NASA Contractor Report 145328

ODIN SYSTEM TECHNOLOGY MODULE LIBRARY, 1972-73

(NASA-CR-145328) ODIN SYSTEM TECHNOLOGY MODULE LIBRARY, 1972 - 1973 Report Study, Mar. 1971 - Jun. 1972 (Aerophysics Research Corp., Bellevue, Wash.) 738 p HC A99/MF A01 CSCL 22B G3/15

N78-20168

Unclas 11906

D. S. HAGUE, D. A. WATSON,
C. R. GLATT, R. T. JONES,
J. GALIPEAU, Y. T. PHOA, AND R. J. WHITE

AEROPHYSICS RESEARCH CORPORATION
BELLEVUE, WA 98009

NAS1-10692

MARCH 1978

National Aeronautics and Space Administration

Langley Research Center
Hampton Virginia 23665

PREFACE

This report was prepared under Contract NAS 1-10692, "Study to Develop a Computer Program for the Synthesis and Optimization of Reusable Launch Vehicles." The study was carried out in the period from March, 1971, to June, 1972. The study was funded by the National Aeronautics and Space Administration, Langley Research Center, and sponsored jointly by the Space System Division and the Flight Dynamics and Control Division. Mr. Jarrell R. Elliott and Mr. Timothy R. Rau of the Flight Dynamics and Control Division served as technical monitors for the study. Development of the ODIN concept was jointly supported not only by NASA but also by Wright-Patterson Air Force Base (Air Force Flight Dynamics Laboratory).

The study resulted in a new, large-scale programming technique called ODIN, for Optimal Design Integration. The use of ODIN involves the linking of independent computer program modules and inter-communication of common information among the programs through an executive program, ODINEX.

This report describes the technology modules and the executive program now available in the ODIN concept.

The authors wish to express their gratitude to Mr. John Decker and his staff of the Space Systems Division for their invaluable contribution in the installation and evaluation of the ODIN system. The authors are also indebted to Mr. Bernard J. Spencer, Jr., of Langley Research Center, for his initial support of the ODIN concept as an aid to the design of reusable launch vehicles. As a result of their efforts, the ODIN system is becoming a working tool at the Langley Research Center computer complex.

ABSTRACT

ODIN/RLV is a digital computing system for the synthesis and optimization of reusable launch vehicle preliminary designs. The system consists of a library of technology modules in the form of independent computer programs and an executive program, ODINEX, which operates on the technology modules.

The *technology module library* contains programs for estimating all major military flight vehicle system characteristics, for example, geometry, aerodynamics, propulsion, inertia and volumetric properties, trajectory and missions, economics, steady-state aeroelasticity and flutter, and stability and control. In addition, a generalized system optimization module, a computer graphics module, and a program precompiler are available as user aids in the ODIN/RLV program technology module library.

The ODINEX executive program controls the design synthesis and optimization by operating on the technology module library under control of a user-specified data input stream. Synthesis procedures in any design simulation are established by the input data. Hence, any set of vehicle component matching and sizing loops can be defined. There is no effective limit on the design sequence "topology" which may be employed in an ODIN/RLV simulation since the sequence is controlled by input data.

The technology module program library has been established by an extensive survey of existing computer programs available to the general aerospace industry. Governmental, industrial, and academic sources for technology module programs were used in construction of the final program library. Individual credit for the program sources is acknowledged where possible in either the technical discussion or the list of references. In certain cases extensive modification of source programs were made. However, many source programs are employed in essentially unmodified form.

It should be noted that the ODIN/RLV program provides the designer with a "building block" approach to vehicle design. The design simulation parallels that now employed in industry; however, the ODIN/RLV permits all interdisciplinary data interchange to be performed within the computer rather than by hand outside the computer. This feature allows the designer to perform more iterations in the vehicle design.

Program operation effectively requires the use of a conventional design team approach. The design team defines all desired information transfers, matching loops and sizing required to achieve a satisfactory vehicle design.

The ODIN/RLV program provides the designer with a tool for automation of the vehicle design process which has the ability to retain the full technical depth associated with current preliminary design analyses.

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

ODIN SYSTEM TECHNOLOGY MODULE LIBRARY, 1972-73

by D. S. Hague, D. A. Watson, C. R. Glatt, R. T. Jones,
J. Galipeau, Y. T. Phoa, and R. J. White

AEROPHYSICS RESEARCH CORPORATION

SECTION 1: INTRODUCTION

Efficient preliminary design of a reusable launch vehicle involves the simultaneous satisfaction of all vehicle operational constraints and optimization of the vehicle's performance. Operational constraints and performance criteria include

a. Landing and take-off performance
b. Payload capability
c. Maximum acceleration and lift coefficient maneuver limits
d. Mach-altitude flight envelope limits
e. Thermodynamic constraints
f. Economics

For vehicles which will operate near civilian population centers, there exists an increasing requirement for satisfying environmental constraints, such as noise and engine pollution. Optimal design of reusable launch vehicles to these performance and constraint characteristics involves a complex system of nonlinear interdisciplinary trade-offs. Technology areas to be considered include

a. Geometry
b. Aerodynamics
c. Propulsion
d. Material stress
e. Weights
f. Aeroelasticity
g. Stability and control
h. Cost

Reusable launch vehicle design teams must carefully integrate the requirements of these multiple disciplines in order to obtain the best vehicle configuration for a specified mission spectrum.

The aerospace industry has continually encountered increases in vehicle and mission complexity. In recent years increasing complexity has tended to force the practicing aerospace engineer into a relatively small area of specialization. Thus, the problem of integrating all significant disciplines entering vehicle design has become a major obstacle to rapid and efficient vehicle design. Efforts to overcome the design integration problem have resulted in increases in vehicle preliminary design staffs of from 20 to 30 working for several months on the earliest supersonic aircraft to many hundred working for several years on more recent supersonic projects.

The increased effort required to integrate modern vehicles has been discussed in Reference 1. For example, Reference 1 indicates that wind tunnel test time required for advanced vehicles is rising exponentially with time, Figure 1-1. This exponential growth of effort is matched by other areas such as man hours and computational effort required. Examples are readily forthcoming. In strength analysis engineer's approximate theories are being replaced by finite element modeling. In aerodynamics and aeroelasticity, strip theory is replaced by the use of finite surface theories. In performance analysis the variational calculus formulation is used in place of conventional flight handbook calculations. In practice experimental effort, manpower, and computational effort to achieve a vehicle design are all rising simultaneously.

The use of more extensive experimental and theoretical analyses can be justified. Thus, when designing a supersonic aircraft the use of variational calculus techniques, Reference 2, will usually produce a performance estimate which improves on flight handbook performance estimates by 15 to 20 per cent, Reference 3. It is pointed out in Reference 4 that the greatly increased computational effort required to define this performance gain and to capitalize upon it in the vehicle design is worthwhile when a significant vehicle production order is anticipated. Similarly, the increased sophistication of analysis in other areas can be justified in terms of ultimate system effectiveness.

However, a major obstacle to the use of more sophisticated analysis emerges in practice. These analyses require increased specialization among the design team members and are generally more costly in terms of dollars and elapsed time. Finally, since each discipline becomes more compartmentalized as a result of increased specialization, *the design integration process itself becomes more complex.*

The increase in design integration complexity is readily visualized in terms of the trajectory analysis. If the vehicle trajectory is fixed, other disciplines can examine the design independent of the trajectory analyst. If each time a vehicle configuration parameter is changed a significant change to the vehicle's optimal flight path results, then the integration problem becomes far more complex. In actuality, efficient modern vehicle design requires the coupling of all major technologies.

The present study was addressed to the technology integration problem, and an optimal design integration procedure (ODIN) has been devised. This procedure is based on computer-aided design concepts. The approximate growth in computational capacities of several representative computers in solution of typical aerospace vehicle analyses is presented in Figure 2. Reference 1 has similarly outlined the total growth of the United States computational power. The ODIN procedure for reusable launch vehicles developed during the present study period and the related study of Ref. 5 is based on the premise that the increased computational capacity which has made today's sophisticated analysis procedures feasible is also capable of greatly improving vehicle design integration procedures.

Achievement of this improvement in design integration procedures has required

a. Creation of a technology computer program library;

b. Construction of an executive program which allows the programs within the technology library to communicate with each other without the necessity for manual intervention in the design analysis;

c. A generalized method for specification of analysis sequence including matching and sizing loops;

d. A method for systematically perturbing design variables to satisfy operating constraints while optimizing system capability.

The ODIN system developed under the present study and the Ref. 5 study contains all the above features. The system and its operation is described in Section 2. Description of the technology program library initially installed at Langley Research Center follow in Section 3 onwards.

This system has also been installed on the CDC CYBERNET system of interlocking computers through the San Francisco and Seattle Data Centers as shown in Fig. 1-3.

The ODIN system described in this report has been applied to a variety of reusable launch vehicle analysis and design integration problems during the study including

a. Orbiter matched subsonic/hypersonic wign design
b. Orbiter hot skin landing problem
c. Advanced transportation system studies

The most comprehensive problem investigated was the synthesis of a matched subsonic/hypersonic orbiter wing, Ref. 7; Fig. 1-4 presents views of this vehicle as produced by the ODIN system. Fig. 1-5 illustrates the complex system of technology modules executed to accomplish the synthesis. Fig. 1-6 presents a wind tunnel photograph of the final ODIN wing design. Lines for this model were produced automatically by the ODIN graphics module and supplied directly to the Langley model manufacturing shop. Fig. 1-7 illustrates the close agreement between experiment and the ODIN aerodynamic estimation modules employed. Fig. 1-8 illustrates the complete set of technology modules available in the ODIN/RLV system.

Development of the ODIN system is continuing with funding supplied by Langley Research Center, Contract NAS 1-12008 and Contract NAS 1-12977; Lyndon B. Johnson Space Center, Contract NAS 9-13584; and the Air Force Flight Dynamics Laboratory, Contract F33615-73-C-3039. The latter reference is reported in Ref. 6. In addition, the ODIN system has been installed at Ames Research Center under Contract NAS 2-7627.

REFERENCES:

1. Harper, C. W., "Prospects in Aeronautics Research and Development," *Journal of Aircraft*, Volume V, Number 5, September-October 1968, Page 417.

2. Hague, D. S., Atmospheric and Near Planet Trajectory Optimization by the Variational Steepest-Descent Method, NASA CR-73365, 1969.

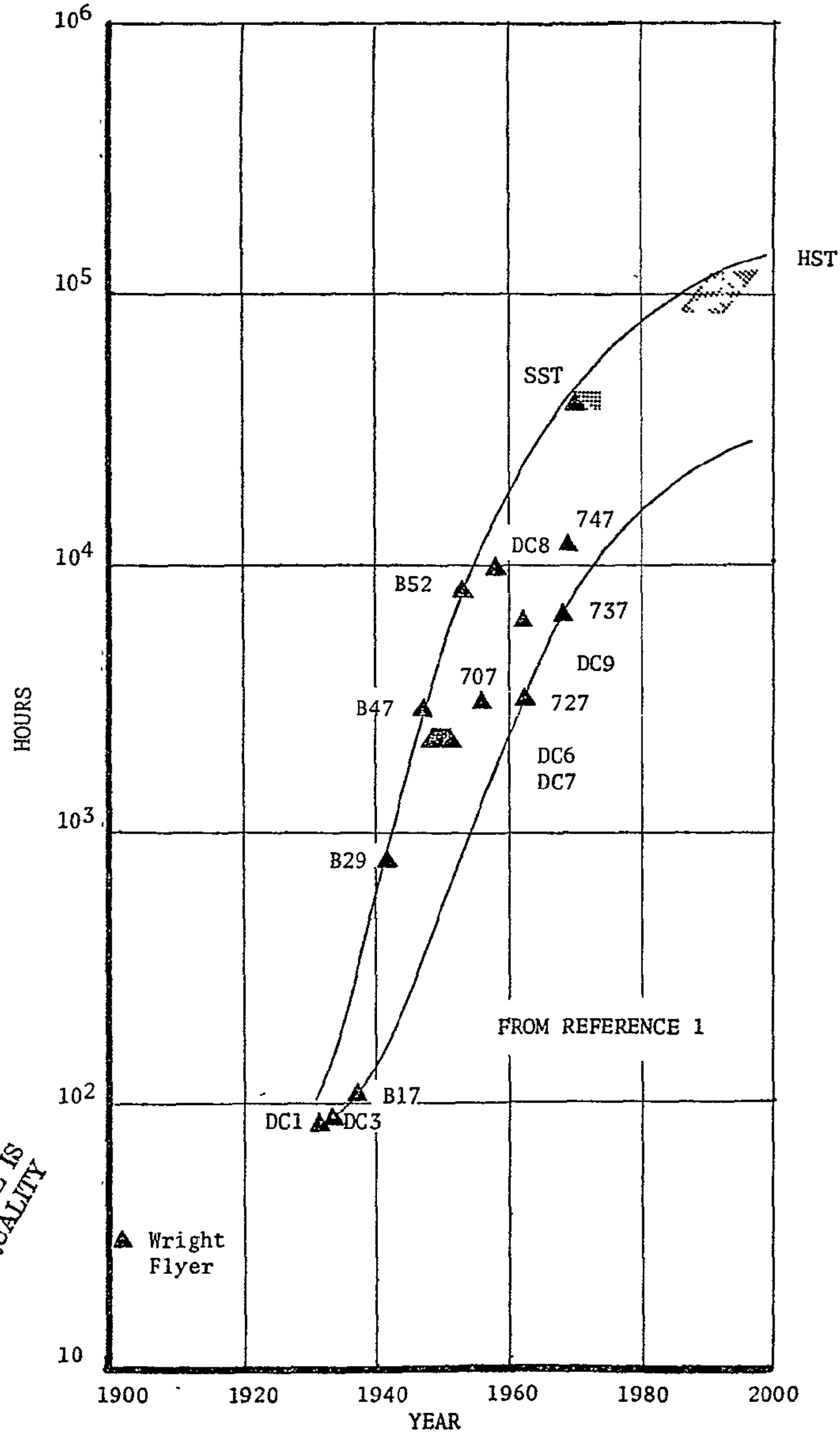
3. Landgraf, S. K., "Some Applications of Performance Optimization Techniques to Aircraft," *Journal of Aircraft*, March-April 1965.

4. Hague, D. S., Jones, R. T., and Glatt, C. R., Combat Optimization and Analysis Program--COAP, Volume I, AFFDL-TR-71-52, May 1971.

5. Hague, D. S. and Glatt, C. R., Optimal Design Integration of Military Flight Vehicles, ODIN/MFV, AFFDL-TR-72-132, December 1972.

6. Hague, D. S., Air to Surface Missile Performance and Design Sensitivity, AFFDL-TR-74-76, July 1974

7. Phillips, W. Pelham; Decker, John P.; Rau, Timothy R.; and Glatt, C. R., Computer-Aided Space Shuttle Orbiter Wing Design Study, NASA TN D-7478, May 1974.



ORIGINAL PAGE IS OF POOR QUALITY

Figure 1-1. Wind Tunnel Hours for Vehicle Design

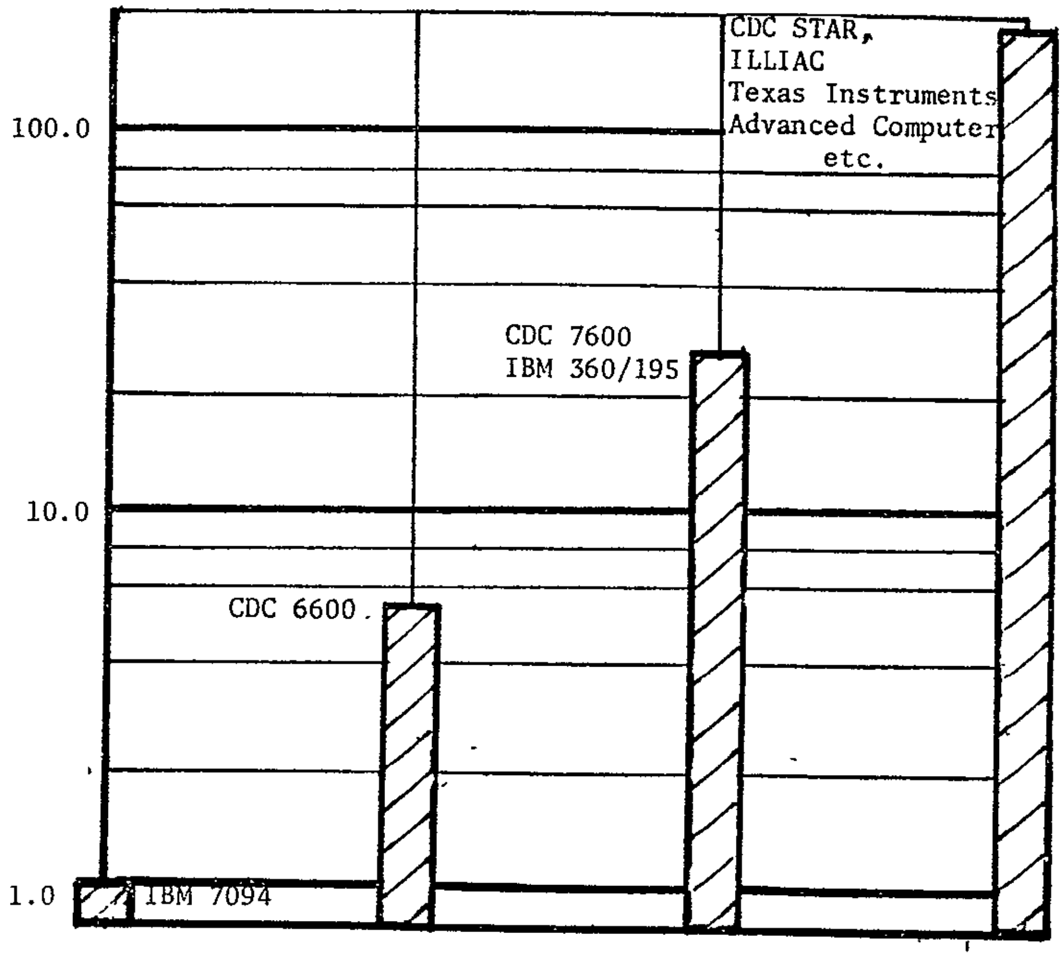


FIGURE 1-2. GROWTH OF AVAILABLE
COMPUTATIONAL POWER

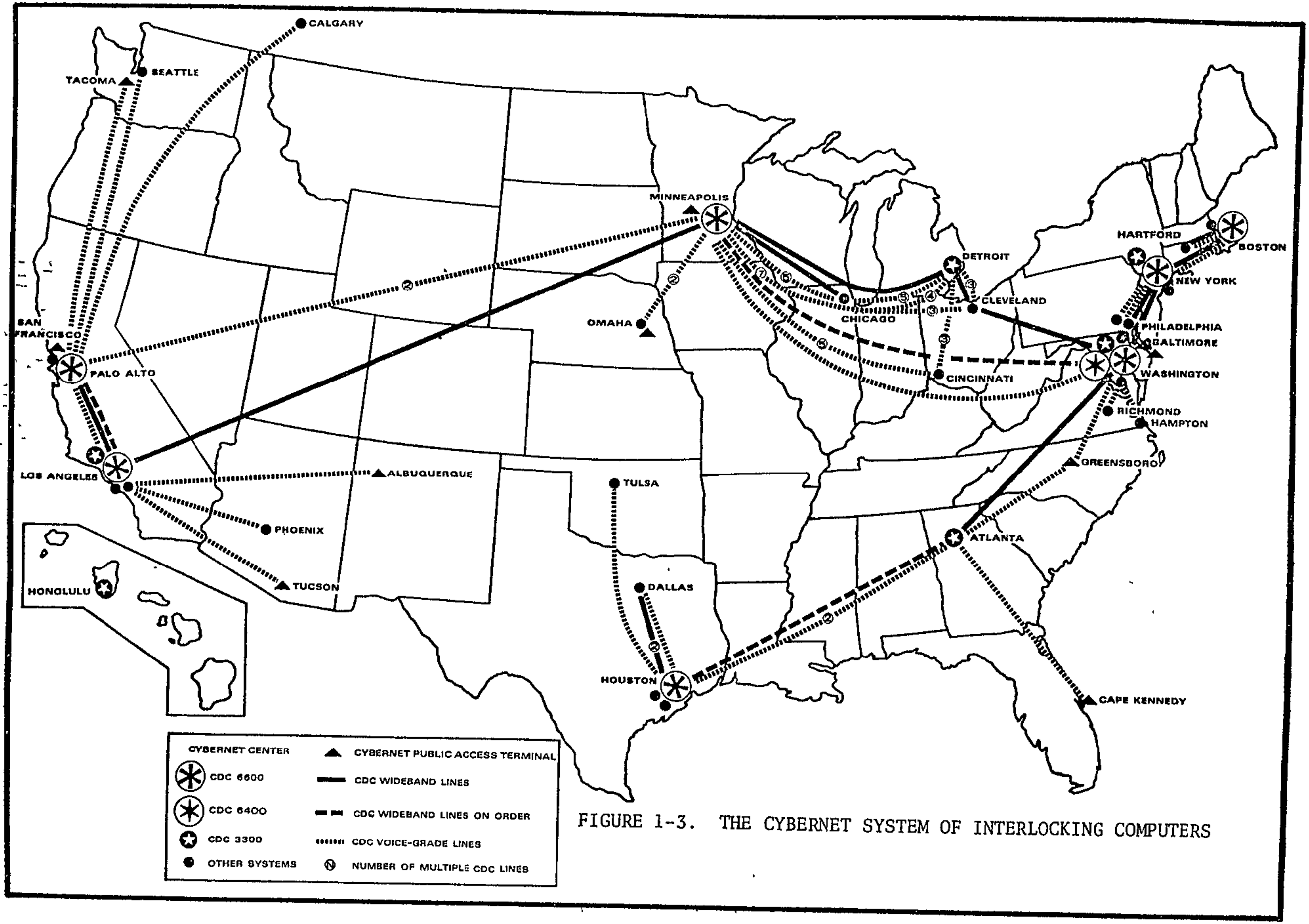
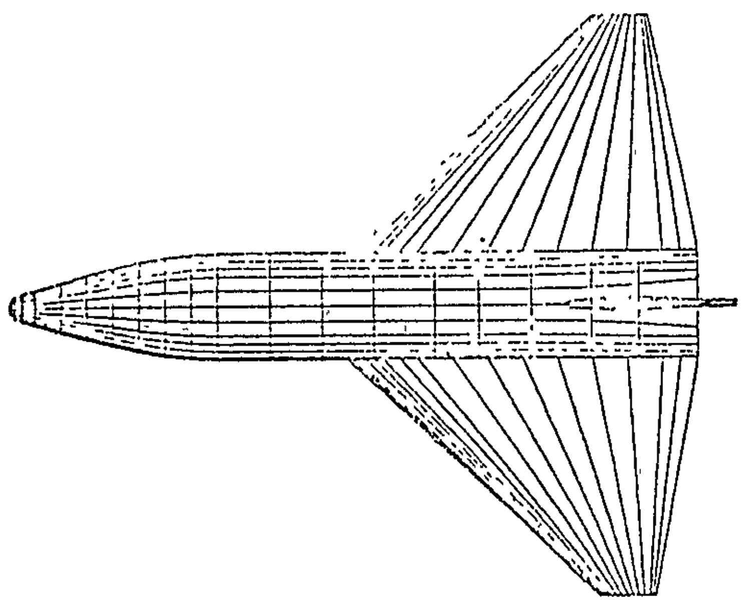


FIGURE 1-3. THE CYBERNET SYSTEM OF INTERLOCKING COMPUTERS

CONFIGURATION SELECTED



$\Lambda_{l.e.} = 47°$

$\Lambda_{t.e.} = -11°$

$S_{WING} = 3387\ FT^2$

$S_{ELEVON} = 679\ FT^2$

$\lambda = .135$

$AR = 2.4$

$i_{WING} = 1.5°$

P/L OUT:

WT	=	159,609 LBS
C.G.	=	.671 l
$C_{m_{C_L}}$	=	$-.028\ \bar{c}$

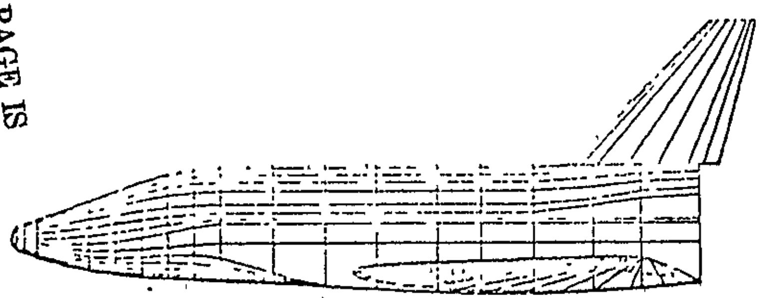
40K P/L:

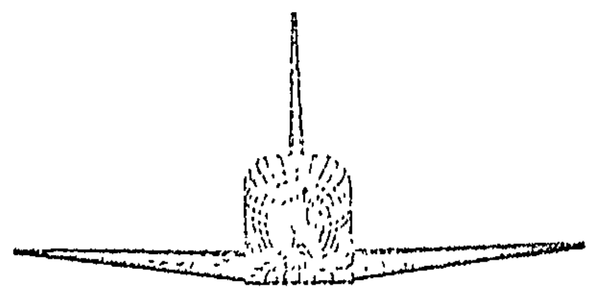
WT	=	199,609 LBS
C.G.	=	.650 l
$C_{m_{C_L}}$	=	$-.080\ \bar{c}$

$V_{M,D}$ (α=17°) = 150 KTS

$V_{M,D}$ (α for $(L/D)_{MAX}$) = 210 KTS

$\alpha_{MAX,TRIM}$ HYPER = 46°





ORIGINAL PAGE IS
OF POOR QUALITY

FIGURE 1-4

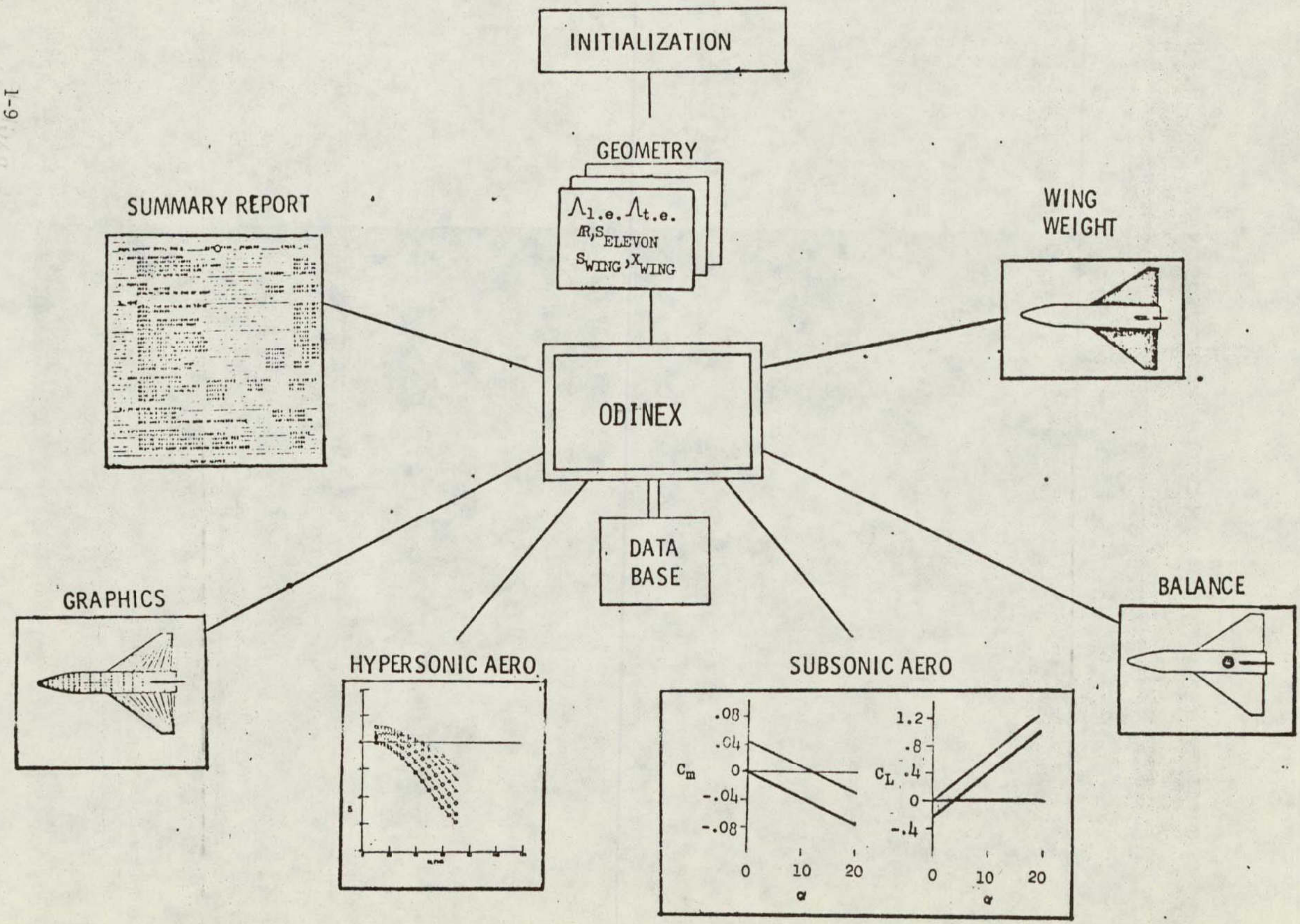
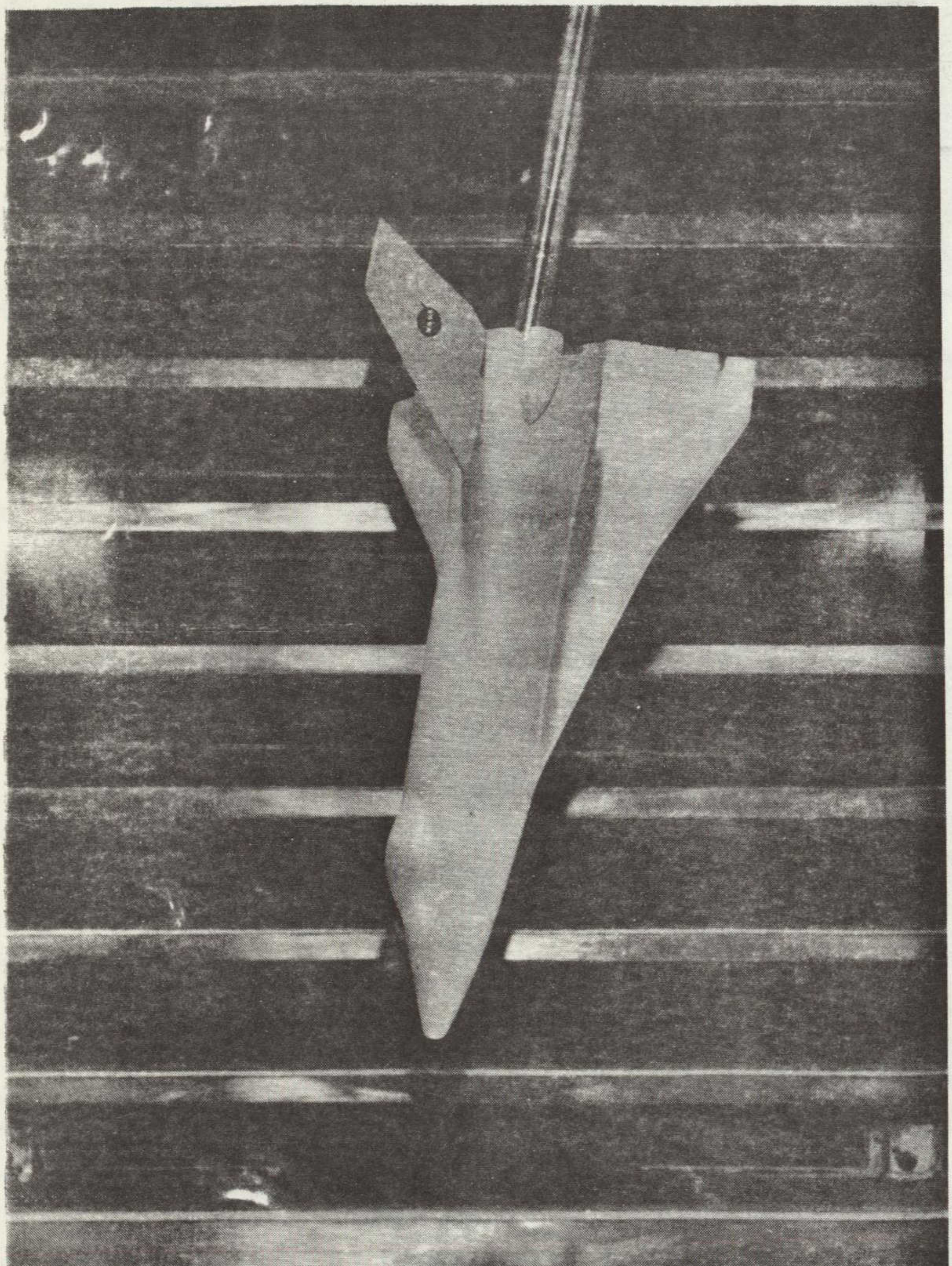


FIGURE 1-5. **ODIN ORBITER WING DESIGN STUDY**



ORBITER SUBSONIC AERODYNAMICS

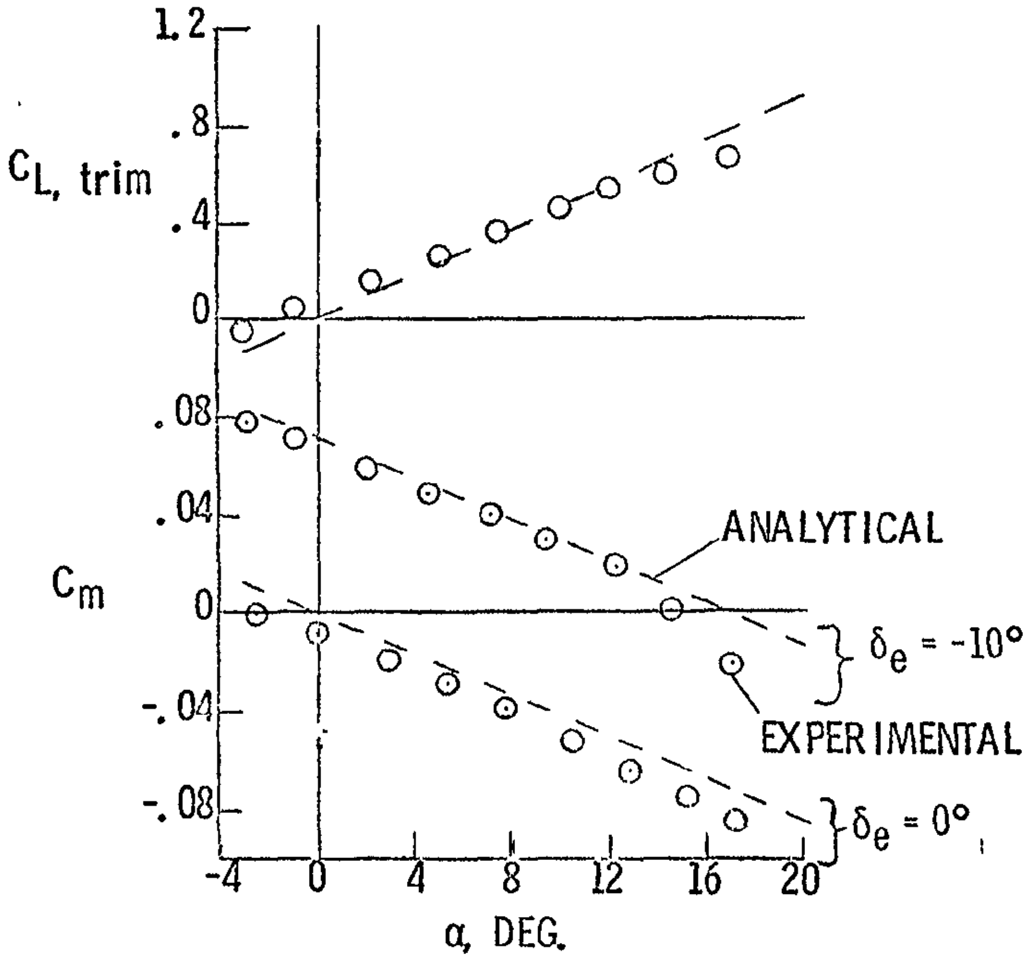


FIGURE 1-7.

OPTIMAL DESIGN INTEGRATION SYSTEM

ODIN

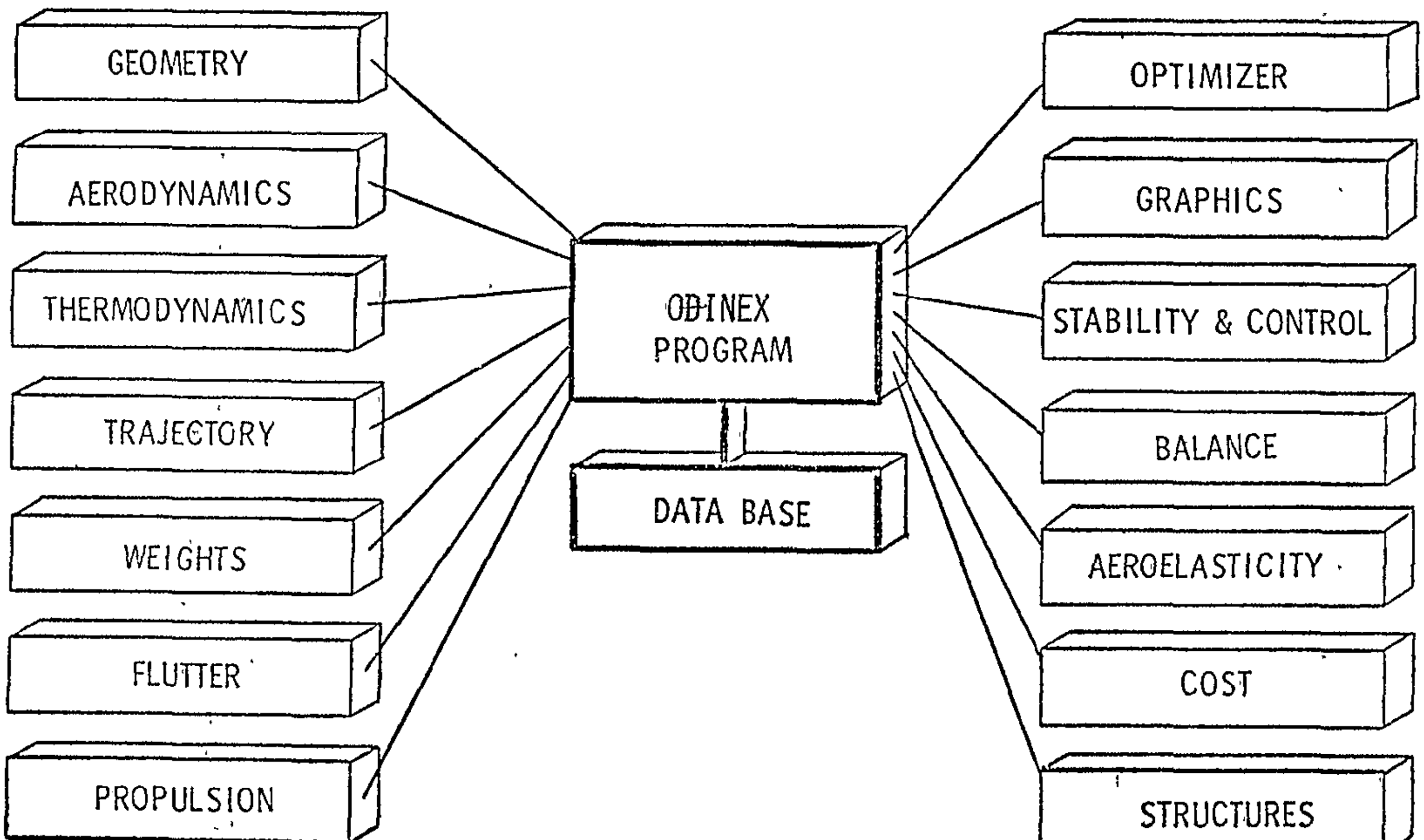


FIGURE 1-8

SECTION 2

STRUCTURE AND OPERATION OF THE OPTIMAL DESIGN INTEGRATION PROCEDURE FOR REUSABLE LAUNCH VEHICLES, ODIN/RLV

This section describes the Optimal Design Integration Procedure for Reusable Launch Vehicles (ODIN/RLV) computational system, its structure, and its application. The ODIN/RLV computational system contains a library of many programs which are used as needed for a given problem. The resultant program run time and core requirements to solve a given problem is, therefore, variable depending upon the programs used. Many of the programs contained in the ODIN/RLV library were developed independently of the present study, several under previous Government funded studies. The ODIN/RLV executive control program which allows the independent programs of the ODIN/RLV library to communicate with each other was developed entirely within the context of the present study, and the related U. S. Air Force supported study of Reference 1.

In developing the ODIN/RLV a survey of existing technology programs and methodologies generally available to the aerospace industry was conducted. Programs surveyed are listed in Reference 1. The ODIN/RLV initial program library was limited to only a few of these programs due to the limited scope of the study effort. Other programs may readily be introduced into the ODIN/RLV by a minor program modification as described later in this section.

REFERENCES:

1. Hague, D. S. and Glatt, C. R., Optimal Design Integration of Military Flight Vehicles, ODIN/MFV, AFFDL-TR-72-132, December 1972.

2.1 STRUCTURE OF THE ODIN/RLV SYSTEM

The components of the ODIN/RLV system are illustrated in Figure 2.1-1; each system component exists in the form of one or more independent computer programs. The system consists of a variety of technology modules including four design service modules plus an executive program. The preliminary design service elements in Figure 2.1-1 consist of

a. Design optimization
b. Plotter program
c. Macro Fortran
d. Report generator

These modules are described in detail in later sections of the report. Briefly, the design optimization element is used to perturb the vehicle design variables in optimization studies. The plotter program element provides the designer with a plot capability for output. The macro Fortran module is a Fortran based pre-compiler of general utility in the manipulation of ODIN/RLV program elements and is available mainly as a programming aid device. The report module enables the user to format his output in any manner he wishes under input data control without program modification.

Since the ODIN/RLV system comprises more than *one quarter of a million Fortran source cards*, some precautions must be taken to provide a usable system capable of interpretation by designer, engineer, and programmer. The major such precaution has been the creation of a system which is truly modular in the sense that it consists of many *independent* computer programs. Any one of these programs can be revised, extended, or replaced without affecting the other program elements of the ODIN/RLV *in any way*. In consequence, the specialist in a given technology area is able to phrase his analysis of the design without regard for the other technologies involved other than for the *interfaces from and to his discipline and other disciplines* entering the design.

The key element in the ODIN system is the executive program ODINEX, of Reference 1. This program controls the execution of all technology modules, the design synthesis, and the interprogram data transfers.

The final element of the ODIN/RLV is the data base, Figure 2.1-1. This data base contains all information to be communicated between program elements. When combined with the nominal input data, it is sufficient to completely define the problem under study.

REFERENCES:

1. Glatt, C. R., Hague, D. S., and Watson, D. A., An Executive Computer Program for Linking Independent Programs, NASA CR-2296, September 1973.

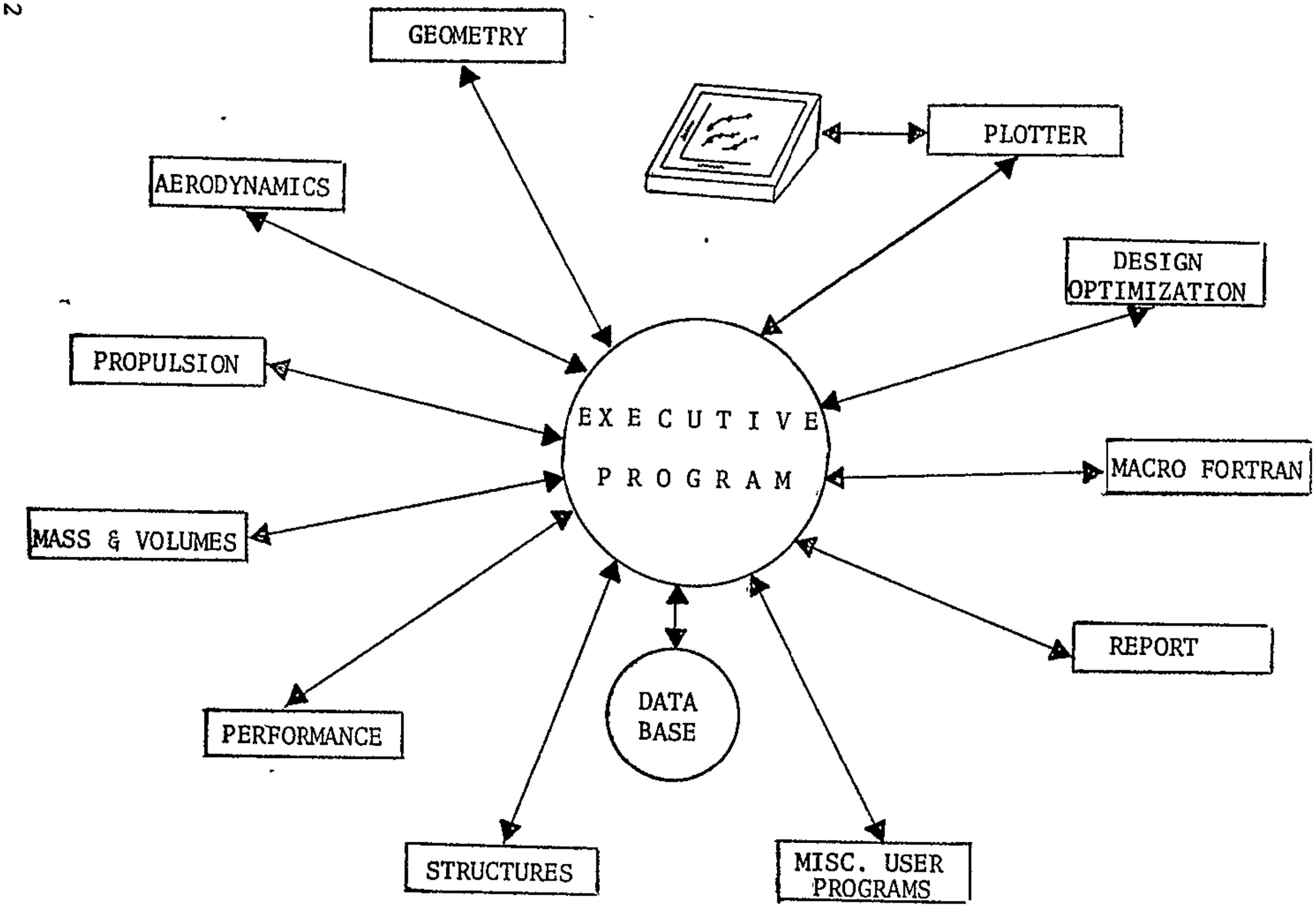


FIGURE 2.1-1. ODIN/RLV SYSTEM

2.2 THE BASIC ODIN/RLV PROGRAM ELEMENTS

The independent program elements which form a basis for the ODIN/RLV system are written in Fortran; although this is not a system restriction. In fact, independent programs written in a variety of languages can be intermingled during an ODIN synthesis, for example, FORTRAN, COMPASS, and COBOL. Each program in the system has been assigned a four to six letter mnemonic for reference purposes and for operation in the ODIN/RLV system. Table 2.2-1 presents a list of the basic ODIN/RLV program library and the mnemonics assigned. When constructing the sequence of analyses which lead to the synthesis and optimization of a reusable launch vehicle, each program must be referred to by its mnemonic code in the ODIN/RLV system. Mnemonic control of the elements in the ODIN/RLV program library is discussed in more detail in later sections and in NASA CR-2296 (see Reference 1, Page 2.1-1).

TABLE 2.2-1. MNEMONICS FOR THE BASIC ODIN/RLV INDEPENDENT PROGRAM LIBRARY

PROGRAM	MNEMONIC	TECHNOLOGY AREA
Executive Control Program	ODINEX	Executive
Geometric Paneling Program	PANEL	Geometry
Hypersonic Arbitrary Body Aerodynamic Computer Program	HABACP	Aerodynamics
Techniques to Evaluate Design Trade-Offs in Lifting Re-entry Vehicles	TREND	Aerodynamics
Skin Friction Drag	LRCSF	Aerodynamics
Zero-Lift Wave Drag	LRCWDZ	Aerodynamics
Zero-Lift Wave Drag	ARPII	Aerodynamics
Wave Drag at Lift	LRCWDL	Aerodynamics
Wetted Areas	LRCWA	Aerodynamics
Configuration Plots	LRCACP	Aerodynamics
Vehicle Synthesis for Advanced Concepts	VSAC	Weights
Atmospheric Trajectory Optimization Program (Version III)	ATOP III	Trajectory
Mission Segment Analysis Program (Version II)	NSEG II	Mission Analysis
Stability and Control Including Linear Control Systems	ACMOTAN	Stability & Control
Development & Production Costs of Aircraft	DAPCA	Economics
Improved Cost Estimation	PRICE	Economics
Volume, Area & Mass Properties	VAMP	Mass Properties
Swept Strip Aeroelastic Model	SSAM	Structures
& Off-Design Performance for single-spool engines	GENENG	Propulsion
Design & Off-Design Performance for Two- and Three-Spool Turbofans with as Many as Three Nozzles	GENENG II	Propulsion
Turbojet Design Point Performance	ENCYCL	Propulsion
Automated Engineering and Analysis	AESOP	Optimization
One-Dimensional Analysis of Three-Layer Ablating Material	ABLATOR	Thermodynamics
Surface Temperature Calculation for Aircraft-Like Vehicles	ATOP *Thermodynamics*	Thermodynamics
Macro-Fortran Language for Development of Precompilers	MAC	Miscellaneous
Independent Plot Program	PLOTTER	Miscellaneous
Quadrilateral Panel Display	IMAGE	Miscellaneous

2.3 INSTALLATION OF THE ODIN/RLV

The ODIN/RLV can be installed on any CDC 6600 computer which has an operating system containing the Appendix 1-b system utility routine CCLINK. Two versions of CCLINK are available in the basic ODIN/RLV library. Since the ODIN/RLV consists of a library of independent programs, the basic program library must be installed on the computer before simulations can begin.

To install the ODIN/RLV program library the sequence of operations depicted in Figure 2.3-1 must be completed. First, all Fortran source program card decks must be compiled. Each independently compiled program is then stored on a tape or disc unit. More than one program may be stored on a given disc or tape, but each such program must be stored as a separate file. When all programs including the executive program are stored in this manner, simulations can begin.

Simulations will involve sequential execution of technology elements in the ODIN/RLV program library. Basic data for each program element must be set up in the usual manner for that program operating independently of the ODIN/RLV. The analyst or team of analysts then defines the sequence of programs to be executed together with the effect of all design variables on the input for each program.

The simulation then commences using nominal design variable values. A common method of running the simulation is to use the optimization module as the final program element executed in the sequence (other than the executive program). This program receives the relevent system characteristics which have been evaluated and stored in the interprogram data base. On the basis of multivariable search algorithms contained within the optimizer, a perturbed set of control variables are defined replacing those residing in the data base, and the complete simulation sequence is repeated. This second simulation defines perturbed system characteristics to predict another set of design variable perturbations for yet another simulation. This process is then repeated, Figure 2.3-2, until the optimum vehicle satisfying all operational constraints is evolved or until further gains in system performance are negligible in magnitude.

During the simulation all information required to fully define the problem at the level of analysis requested is stored in the data base. On problem completion the data base can be interrogated using a stylized report generator program to compose a user-oriented description of the final design. It should be noted that the data base contains all interprogram data and that the flow of all data to or from the data ~~base~~ and the program elements is completely controlled by the executive program.

When a program element is being executed, there is no way that program is "aware" of the fact that it is performing one analysis function in an overall vehicle simulation. This is a key element in the modular structure of the ODIN/RLV. It insures that the analyses function of each program element in the library can be examined independently of the other analysis programs. Without this feature examination of the complex interconnections between analysis modules would become extremely complex and, in view of the ODIN/RLV's system size, of doubtful validity. It should be noted that the ODIN system thus provides much more capability than the simple OVERLAY system of building large scale computer codes. In fact, many of the programs in the ODIN system are themselves quite lengthy.

The manner in which the sequence of program elements to be executed is defined is outlined in Section 2.2. The manner in which interprogram information is passed between program elements and the data base via the ODIN/RLV executive program is outlined in Section 2.4.

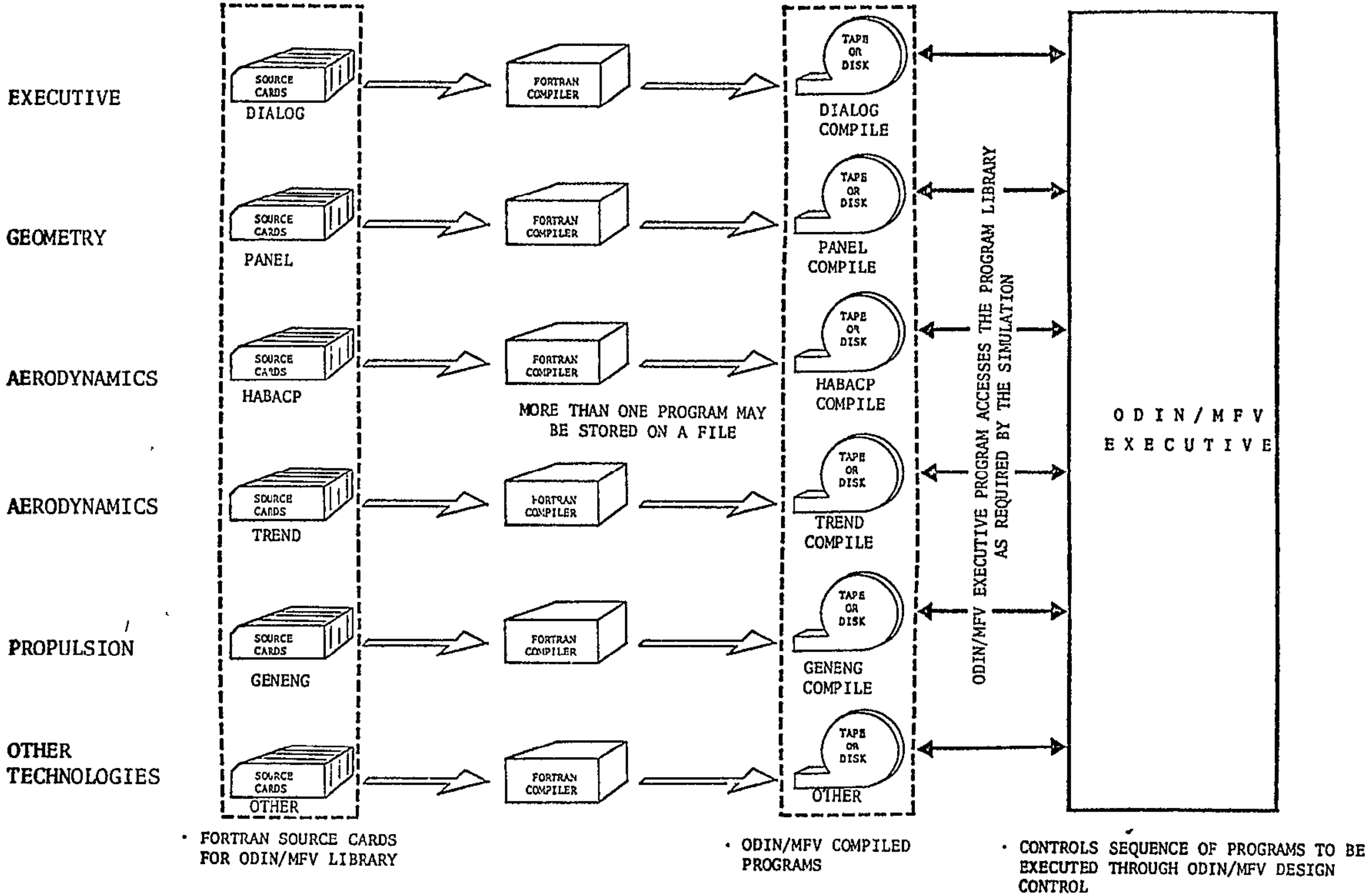


FIGURE 2.3-1 ASSEMBLY OF THE ODIN/RLV PROGRAM SYSTEM

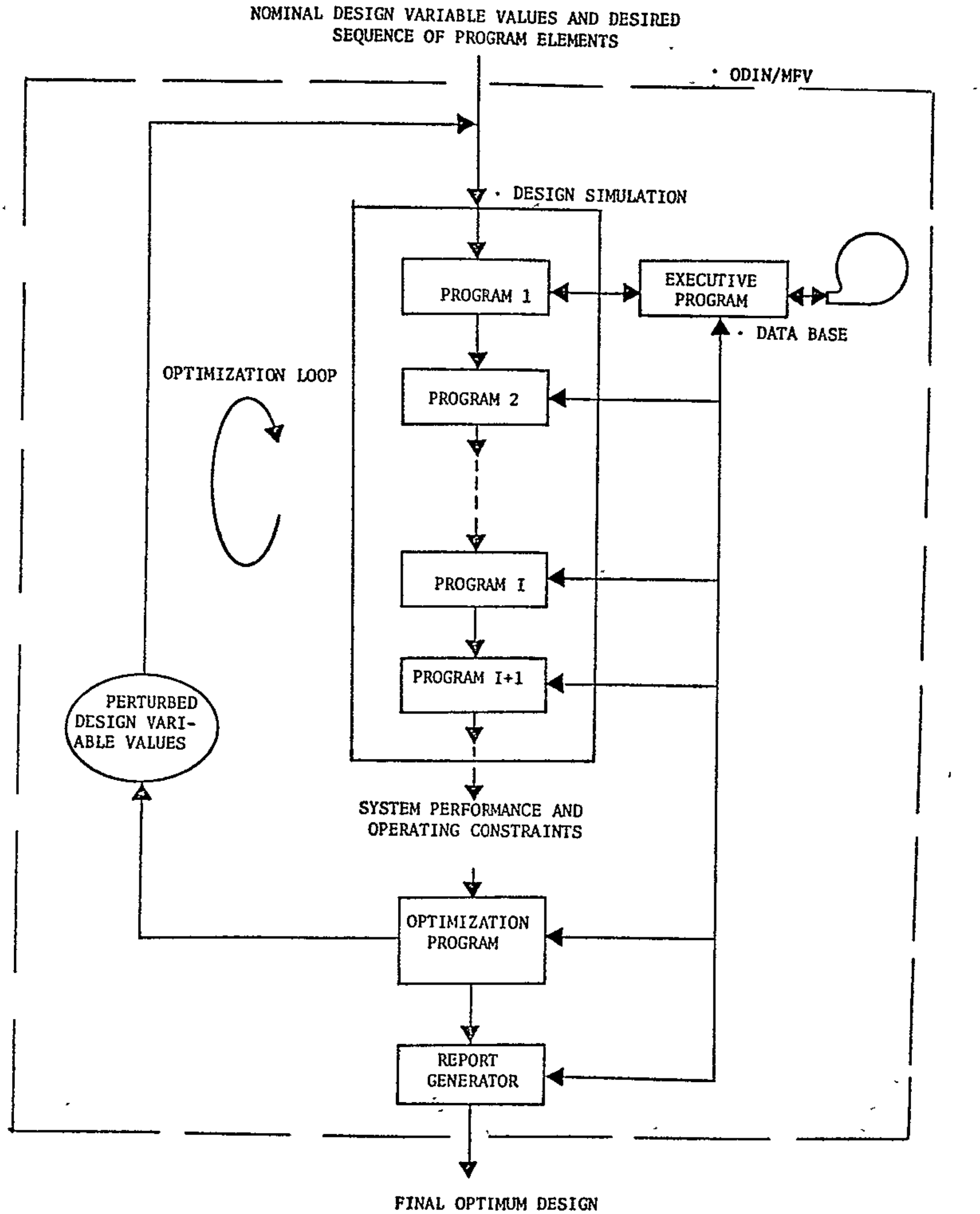


FIGURE 2.3-2 SCHEMATIC OF A VEHICLE DESIGN OPTIMIZATION SIMULATION

2.4 SEQUENTIAL INDEPENDENT PROGRAM EXECUTION

Usually the submission of a computation to a digital computer involves the execution of a single program with possible repetitive evaluation of successive data cases. In the ODIN/RLV system, submission of a computation may involve *the sequential execution of many programs* to obtain a complete vehicle design synthesis. The *sequential execution of many loops through these programs may be required* to obtain an optimal design.

2.4.1 Sequential Execution of More Than One Program

On any digital computer the execution of a single program is governed by a set of control cards which provide instructions to the computer system for compiling and/or loading the specified program. These control cards, the Job Control Language or JCL cards, are peculiar to each computer system and installation. The JCL cards for any computer or installation rarely employ a user-oriented format. For example, Table 2.4-1 presents typical JCL cards for an elementary Fortran compilation and execution of the same program on the CDC 6000 series computer, the IBM 360 series computer, and the UNIVAC 1108. To the user, the JCL, unlike the higher level Fortran language, tends to be incomprehensible. In the remainder of this section details of the JCL cards will be omitted. Collectively, any group of JCL cards necessary to execute a given program (program X) will be referred to as "the JCL to execute program X," and "the JCL cards to compile program X" (JCL_X^E and JCL_X^C).

In actuality to compile and execute the application program X several independent programs must be executed in addition. These other independent programs are all part of the computer operating system. System programs of this type bear a similar relationship to the computer operating system as do the independent technology program elements to the ODIN/RLV executive program, Figure 2.4-1. This analogy forms the basis of the ODIN/RLV:

> "The operating system employs independent system utility programs to compile and execute a given application program. The ODIN/RLV program system employs independent application programs to synthesize a vehicle design."

In this sense, the ODIN/RLV is a newly developed higher order operating system which carries out the analysis function rather than carrying out the program compile and execution function.

Now consider the problem of sequential execution of two applications programs. This can readily be achieved on almost any digital computer. Symbolically,

$$JCL_{(A+B)}^E = JCL_A^E + JCL_B^E$$

where the operator + indicates that the JCL for program B is simply placed behind that of program A and that the operating system operates on the combined JCL cards, $JCL^E_{(A+B)}$.

In general using this notation

$$JCL^E_{(A+B+...+N)} = JCL^E_A + JCL^E_B + \ . \ . \ . + JCL^E_N$$

That is, an arbitrary number of applications programs can be sequentially executed on practically any major digital computer.

This factor forms one basis of the ODIN/RLV; however, in the ODIN/RLV three additional capabilities are required:

a. The sequential JCL cards sets must be controlled by readily understood higher order commands in view of the close requirement for designer interaction. This is achieved by creating an ODIN/RLV Job Control Language which employs commands such as

$$JCL^E_A \equiv \text{EXECUTE A}$$

A readily understood command to the computer, therefore replaces commands such as those in Table 2.4-1.

b. The selected sequence of program JCL cards must be automatically capable of repetition and revision of the sequence as the problem progresses. Symbolically, the following operation must be performed:

$$\begin{aligned}\sum_{i=1,M}(JCL^E_{(A+B+...+N)})_i &= JCL^E_A + JCL^E_B + \ ...+JCL^E_N \\ &+ JCL^E_A + JCL^E_B + \ ...+JCL^E_N \\ &\ . \ . \ . \ . \ . \ . \ . \ . \ . \ . \ . \ . \\ &+ JCL^E_A + JCL^E_B + \ ...+JCL^E_N\end{aligned}$$

where M rows of JCL are to be represented on the right hand side. This capability has been achieved by creating the ability to loop through the ODIN/RLV JCL cards using additional user oriented control commands as illustrated for a five program sequence repeated twenty times in Table 2.4-2. The additional commands are

1. DESIGN POINT I

2. LOOP TO POINT I
IF______.LT.______

which defines Fortran-like instructions for control of the design simulation.

c. The ability to select alternative program execution sequences based on design dependent logic. For example, the symbolic operation

$$JCL_A^E;\ M \leq \bar{M}$$

$$JCL_B^E;\ M \leq \bar{M}$$

This type of operation can readily be carried out with commands of Table 2.4-2 as follows

```
. . . . . . . . . .
. . . . . . . . . .
LOOP TO POINT MA
IF M.GT.MBAR
EXECUTE B
LOOP TO POINT MB
DESIGN POINT MA
EXECUTE A
DESIGN POINT MB
. . . . . . . . . .
. . . . . . . . . .
```

In general, both M and MBAR may be defined in the JCL as in Table 2.4-2 or alternately either may be a variable computed by any of the application programs. In the latter case such variables must be defined in the data base as described in Section 2.5.

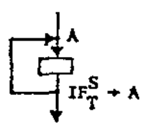
2.4.2 Topology of the General Design Synthesis Calculation

In general the synthesis of military flight vehicles involves a complicated system of analysis loops for satisfying a variety of aerodynamic and propulsive sizing and matching constraints. It is not possible or necessarily desirable to rigidly define the topology of the system of computational loops in the ODIN/RLV. Instead, the analysis sequence to be performed is defined by the ODIN/RLV job control language. This technique allows the vehicle designer complete freedom in specifying the computational sequence; no limit is placed on the complexity of the analysis.

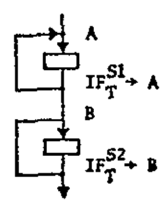
Any number of loops can be created using the LOOP and conditional IF control cards and the associated DESIGN control card. Using the symbolic notation

$$IF_T^S \rightarrow A$$

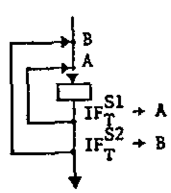
to indicate if the statement S is true go to A, it is apparent that series loops, nested iterative loops, and combined series and nested loops can be constructed. For example:



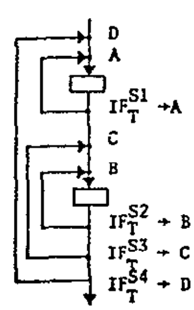
a. SINGLE LOOP



b. TWO SERIES LOOPS



c. TWO NESTED LOOPS



d. TWO SERIES LOOPS AND TWO NESTED LOOPS

Any number of DESIGN POINTS and IF statement ODIN/RLV control cards may be introduced into the computational sequence. Computational time will rise in proportion to the complexity of the computational sequence topology, however. The IF tests employed encompass the standard set of six tests in Fortran; although the form of the ODIN/RLV job control language test differs in form to that of Fortran. The six tests are

```
IF V1.LT.V2 ;   IF(V1 < V2)

IF V1.GT.V2 ;   IF(V1 > V2)

IF V1.LE.V2 ;   IF(V1 ≤ V2)

IF V1.GE.V2 ;   IF(V1 ≥ V2)

IF V1.EQ.V2 ;   IF(V1 = V2)

IF V1.NE.V2 ;   IF(V1 ≠ V2)
```

As noted previously V1 and V2 are two variables constructed in the ODIN/MFV job control language or constructed within any independent program in the synthesis and passed to the data base.

2.4.3 Communication with the Data Base

The data base is an organized system of variable names and the corresponding variable values which are maintained by the ODIN/RLV executive program. Nominally up to 5000 variable names and values may be stored in the data base. This number of variables may be modified by redefining the data base size and recompiling the executive program as discussed in Reference 1.

The data base file of information is dynamically constructed by the executive program as the ODIN/RLV simulation proceeds. The file is resident on disc or tape at the user's option. Construction of the data base involves the following tasks:

a. Search to see if each variable name encountered has been allocated a place in the data base

b. If not define the optimal location in the data base for the variable name and its value

c. Update the variable value

d. Retrieve the variable by name and the associated value.

For example, suppose the vehicle's exposed wing aspect ratio is stored under the name WEXPAR. Let WEXPAR be computed by program A and subsequently used by program B. Schematically, this is illustrated in Figure 2.4-2.

Any number of subsequent programs may access WEXPAR or alternately update this variable. In any given simulation the location of WEXPAR will not change within the data base. In actuality, the programs A, B, C, and D in Figure 2.4-2 *do not* access the data base directly. All access to and from the data base is controlled by the ODIN/RLV executive program, as in Figure 2.4-3.

2.4.4 Data Base Information Transfer System

Data base information transfer is based on a rapid by-name search. Search speed is obtained by the use of "hash" and "collision" methods, Appendix I-A. This approach is more efficient than the more usual linear sequential search which starts with the first name in the table and proceeds sequentially until the desired name is located and the corresponding value is retrieved.

The hash and collision data transfer system operates in the following idealized manner:

1. Take the variable name, say WINGAR, and treat the binary representation of this word as an integer;

2. Find the remainder when the word integer representation is divided by the number of elements in the data base. This is equivalent to the Fortran MOD function which is a very rapid machine operation;

3. Use the remainder as the nominal location of the variable within the data base;

4. Check to see if the location is used since more than one variable name may reduce to this location. If this location has already been used for another variable name store the new variable in the next vacant location and note this location in the data base row originally searched. Figure 2.4-3, line A, illustrates this process with one collision. Line B illustrates a double collision for a name which reduces (hashes) to the same location as B.

5. The retrieval process operates in the same manner. The name is hashed to a given nominal retrieval location. If that location contains the wrong name, the specified alternate location is searched for the desired name, etc. until the desired name is found and the variable value is retrieved.

In numerical experiments with a 2000 word data base filled approximately 75 per cent, it was found that the average name can be retrieved in less than two attempts (fetches). This would compare with 1000 fetches using a linear search for information retrieval. In practice using the ODIN/RLV approximately 9000 values per second are being retrieved on the CDC 6600.

It should be noted that the above description is idealized. Efficient use of core space within the computer requires a more sophisticated packing of information in the data base than the three column diagram of Figure 2.4-4. This is particularly true for arrays which, by definition, have one name but many values. Details of the actual data base structure are provided in Reference 1.

2.4.5 Modifying Program Input to Communicate with the Data Base

Development of the ODIN/RLV program system is based on the principle that independent technology programs without significant modification can be made to communicate with each other through a data base. By following this principle, a method of communicating data base information into each program has been devised. *No modification to the program input data code is required by this method.* The input data prepared by the design team, however, is modified to indicate data base inputs. The modified data input does not affect the technology program; for the ODIN/RLV executive program inspects the data input prior to execution of the technology program and combines the required data base information with the basic program inputs. The executive program then automatically prepares a file containing the modified input format for the technology program and executes that program in the nominal manner. This is illustrated schematically in Figure 2.4-5.

It should be noted that the technology program may still be executed in the normal manner as a stand-alone program independent of the ODIN/RLV system.

2.4.5.1 Data Base Communication through Input

Data base information is entered into the technology program input by means of the special delimiters " ". Any data base variable name may be entered between the delimiters. The executive program will replace the variable name by its value and rewrite a normal card image to replace the modified input cards. The value is placed within the closed region which includes the delimiters. Therefore, namelist-like inputs, rigid format input, and special input procedures can be accomodated by the general input modification.

Examples:

A. NAMELIST

```
A = "WINGAR",
B(1) = "B1"
```

B. RIGID FORMAT

3	"IBAR"	4	18.31	"WINGAR"	6.03	
I5	I6	I5	G12.4	G12.4		

C. SPECIAL INPUT (USED IN ATOP II AND NSEGII, SECTIONS 7.2 and 7.3)

```
AMASS = "SUMMAS "
ATAB01(1) = "AEROTB"
```

2.4.5.2 Algebraic Operations in Data Base Input

The ODIN/RLV system permits the algebraic manipulation of data base information on the modified input cards. Complete details are presented in Appendix I. Some illustrative examples follow.

Examples:

A. CHANGE OF UNITS

AREA = "AREAFT * 144.0"

VKNOTS = "VFPS * 0.593"

This is useful when independent programs employ differing unit systems.

B. VARIABLE COMBINATION

AREA = "BREDTH * WIDTH/2.0"

AMASS = "VOLUME * RHO"

A general ability to perform arithmetic operations involving up to ten operations is available.

The Fortran arithmetic operation precedence convention is not followed. Details are contained in Appendix I, Section 3; basically the calculation proceeds strigtly from left to right. Calculations can be chained by operations such as

A = "B/C + D - E"

F = "A + F . . ., etc. "

Thus, an unlimited arithmetic manipulation capability is present in the ODIN/RLV input procedure.

2.4.5.3 Compiling at the Input Level

When extensive computations are required at the input level or computations involving higher order functions are required, they may be placed in a new program element and compiled at input time. A special ODIN/RLV control card provides this capability. The control card is

EXECUTE COMPILER

This card is followed by the new program which is any normal Fortran program. If desired, the program may include its own subroutine trees. The Fortran source decks present in the input stream are followed by the second ODIN/RLV control card

EXECUTE MYPROGRAM

MYPROGRAM is the file name of the compiled program. The methods of Section 2.3.1 can be used to create a design point structure which insures that the new program is only compiled once and that the compiled program is executed on successive passes through the input stream; for example

```
J = 0
...........
...........
...........

LOOP TO POINT COMPIL
IF J.NE.0
EXECUTE COMPILER

      Fortran Source Deck

J = 1
DESIGN POINT COMPIL
EXECUTE MYPROGRAM
...........
...........
      etc.
```

2.4.6 Communicating Program Output to the Data Base

To communicate selected output of any program to the data base, one modification is required in the technology program. This occurs at the program exit point or points. The modification consists of writing out a Namelist file containing the information to be transferred to the data base. Output file unit is nominally unit 77. For example, to transfer the variables ANAME, BNAME, CNAME, I1, I2, JNAME and these variable values to the data base the following modification is required at the exit point.

```
NAMELIST/DBOUT/ANAME,BNAME,CNAME,I1,I2,JNAME

WRITE(77,DBOUT)
```

The ODIN/RLV executive program interrogates unit 77 after the execution of each technology program to find variable names and values to be entered into the data base. These names and values are entered into the data base as described in Section 2.4.5. A schematic of the output of information to the data base is presented in Figure 2.4-6.

REFERENCES:

1. Glatt, C. R., Hague, D. S., and Watson, D. A., ODINEX: An Executive Computer Program for Lonking Independent Programs, NASA CR-2296, September 1973.

TABLE 2.4-1. TYPICAL JCL TO COMPILE AND EXECUTE A SINGLE PROGRAM ON SEVERAL COMPUTERS

CDC 6600	IBM 360/67	UNIVAC 1108

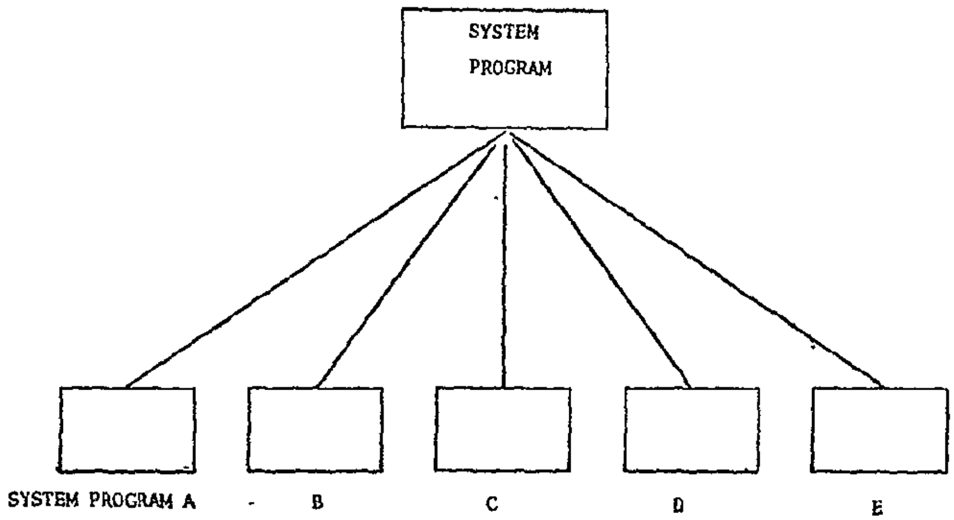
PFL,60000.
FTN,OPT=0.
LGO.
789
{SOURCE DECK}
789
{DATA DECK}
6789.
	// EXEC FORTGCG
	//FORT.SYSIN DD *
	{SOURCE DECK}
	/*
	//GO.SYSIN DD *
	{DATA DECK}
	/*
		▼ FOR MAIN
		{SOURCE DECK}
		▼ XQT MAIN
		{DATA DECK}
		▼FIN

ORIGINAL PAGE IS OF POOR QUALITY

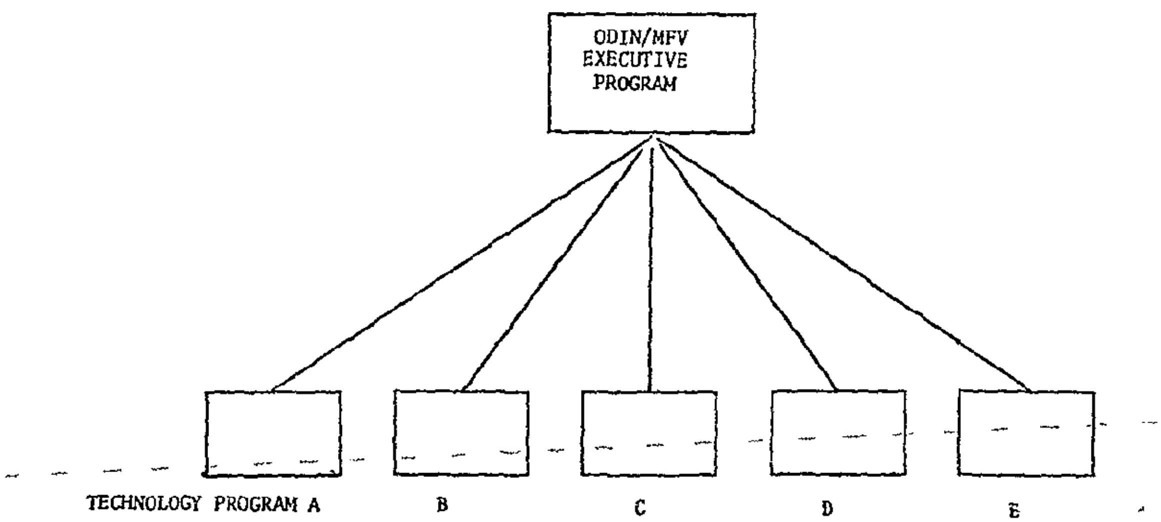
TABLE 2.4-2 USE OF THE ODIN/RLV JOB CONTROL LANGUAGE TO LOOP THROUGH TWENTY SUCCESSIVE EXECUTIONS OF FIVE SEQUENTIAL PROGRAMS

$$\sum_{i=1,20} (JCL^{E}_{A+B+C+D+E})_i =$$

```
COUNT = 0
DESIGN POINT 1
COUNT = COUNT + 1
EXECUTE A
EXECUTE B
EXECUTE C
EXECUTE D
EXECUTE E
LOOP TO POINT 1
IF COUNT .LT. 20
END
```



(a) COMPUTER OPERATING SYSTEM



(b) ODIN/RLV PROGRAM SYSTEM

FIGURE 2 4 -1. ANALOGY BETWEEN OPERATING SYSTEM AND ODIN/RLV SYSTEM

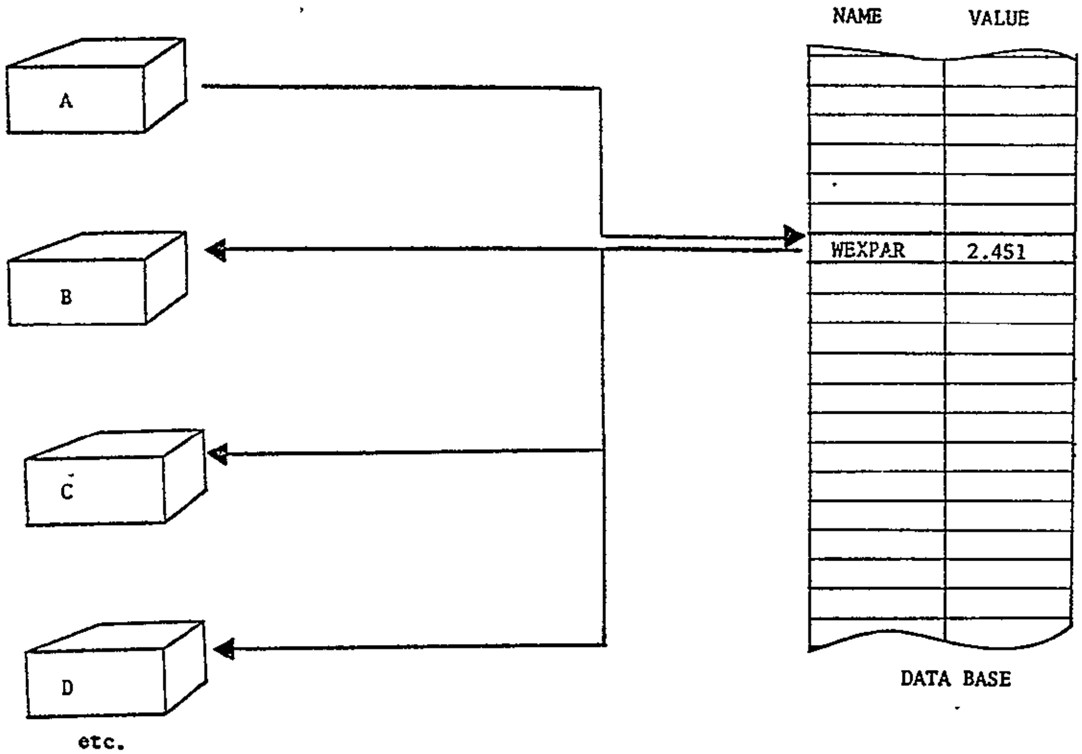


FIGURE 2.4-2. PROGRAM ACCESS TO DATA BASE

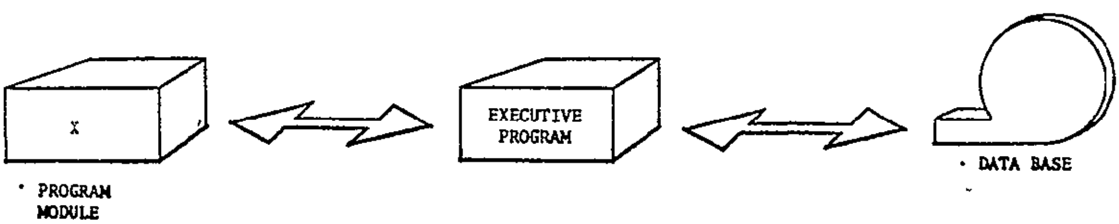


FIGURE 2.4-3. EXECUTIVE PROGRAM CONTROLS ACCESS TO DATA BASE

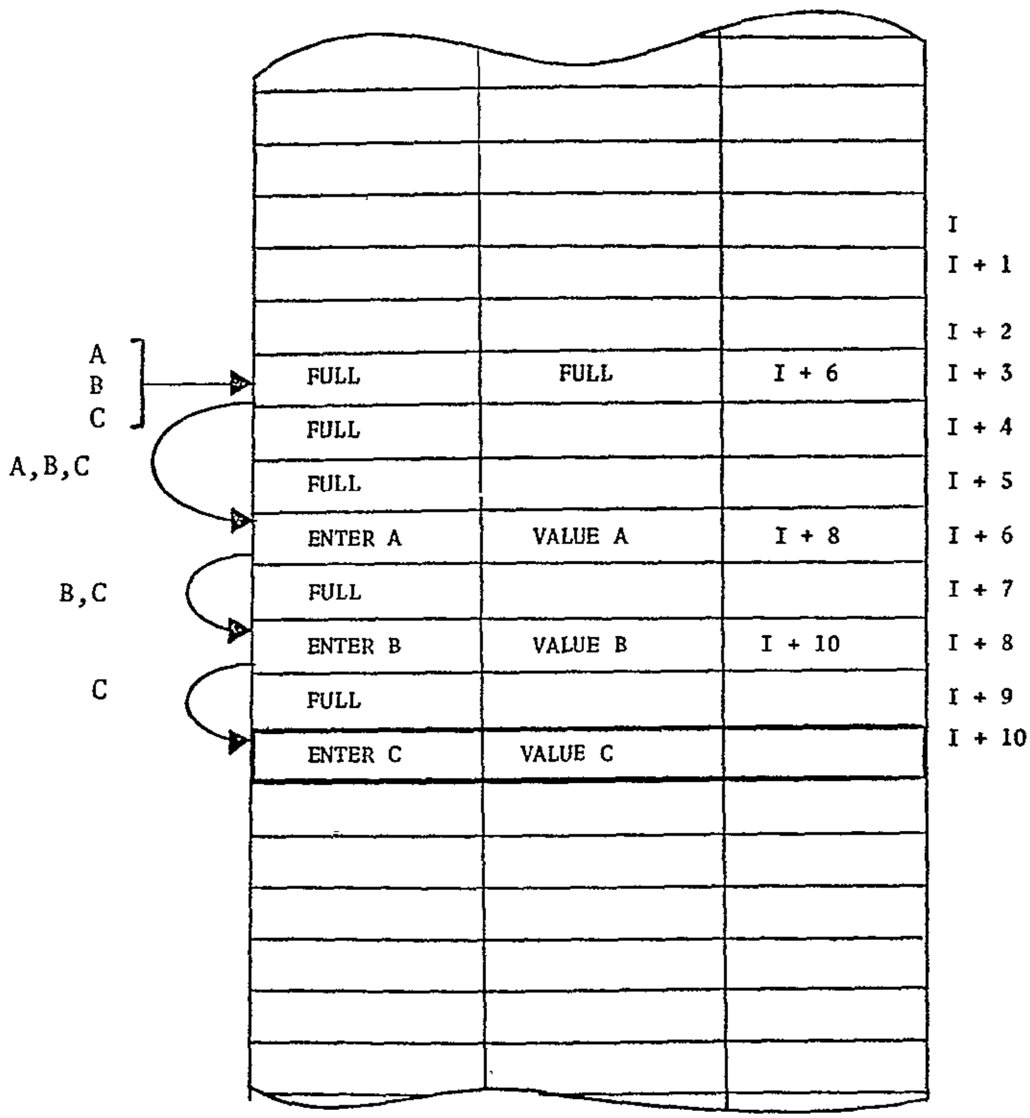


FIGURE 2.4-4. IDEALIZED DATA BASE INFORMATION RETRIEVAL SYSTEM

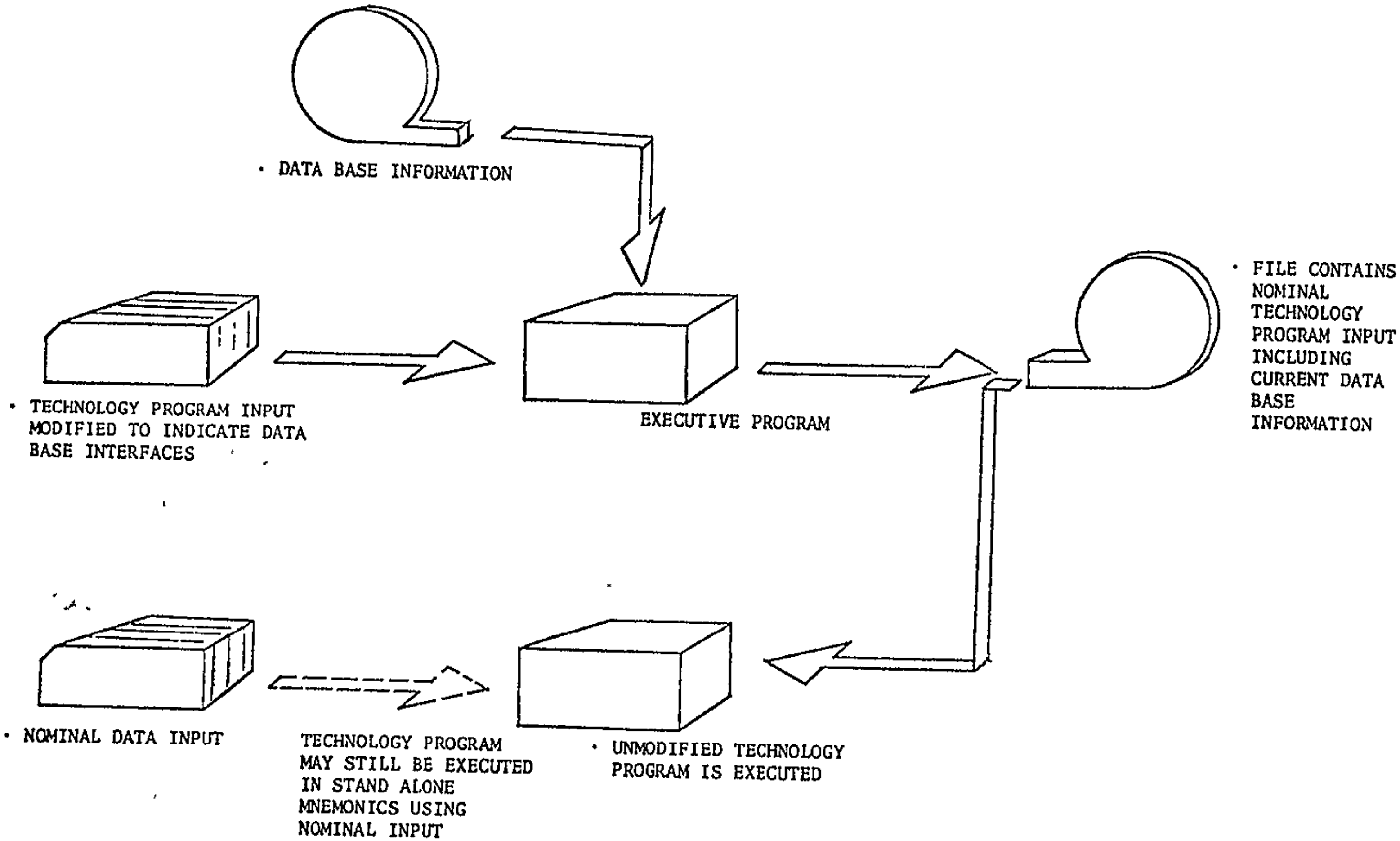


FIGURE 2.4-5 METHOD FOR MODIFYING INPUT DATA WITHOUT PROGRAM MODIFICATION

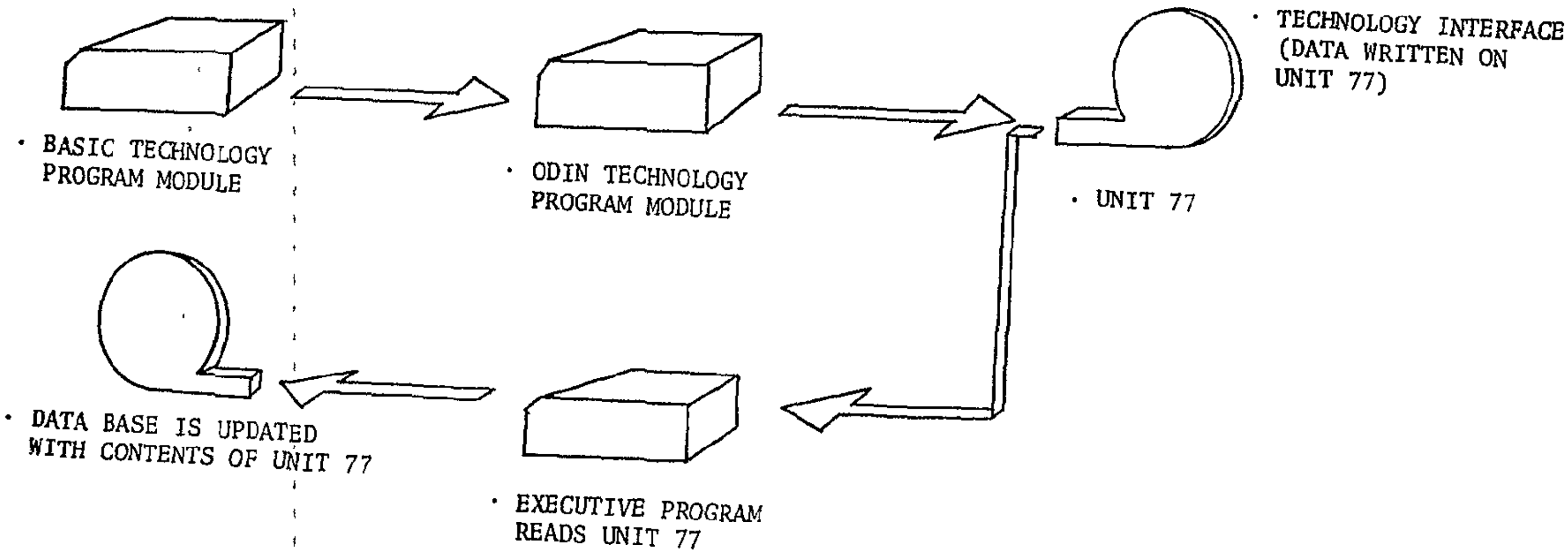


FIGURE 2.4-6. SCHEMATIC OF OUTPUT OF INFORMATION TO DATA BASE

2.5 SUMMARY OF THE ODIN/RLV SYSTEM

The ODIN/RLV System provides a design team with the following capabilities:

1. A basic program library of technology programs for analysis of reusable launch vehicle characteristics
2. The ability to rapidly include additional technology programs in the library
3. A means for automatically transferring and updating information between any technology programs in the library
4. The ability to define an arbitrary sequence of calculations for the analysis of reusable launch vehicle characteristics using the program library including computational loops
5. An automated reusable flight vehicle optimization capability

It follows that the ODIN/RLV has the ability to simulate entirely within the computer the reusable launch vehicle preliminary design procedures now employed in industry. This ability will require the ODIN/RLV design team to have command of all disciplines entering into reusable flight vehicle design.

Description of the ODIN/RLV program elements presented by technology area follow in the next sections. It should be noted that the ODIN system is not specifically limited to RLV simulations. Any computation involving more than one computer program can rapidly be simulated in the ODIN system.

SECTION 3

GEOMETRY

The ODIN/RLV geometry program modules provide three-view, orthographic, and perspective projection graphical descriptions of the vehicle for off-line or cathode ray tube plotting devices. The geometry modules also interface directly with several of the detailed aerodynamic programs of Section 4. Three programs provide the ODIN/RLV geometry capability:

1. PANEL - provides a simplified input for specifying a system of quadrilateral elements which cover the vehicle's surface

2. IMAGE - displays the panelled vehicle surface computed by PANEL on plotting devices

3. LRCACP - is an alternate aircraft configuration surface description and plot package

The first two geometry program modules are closely based on the Gentry hypersonic aerodynamics program, References 1 and 2, geometry package. The third program is a Langley Research Center developed plotting package which interfaces with aerodynamics programs also developed at Langley.

Construction of separate programs for the geometry definition and graphical displays provides a generalized vehicle geometric definition and graphical display available to all technologies. Considerable extension to the computer graphics capability is now being undertaken at Langley Research Center (Contract NAS 1-12977) and the Lyndon B. Johnson Space Center (Contract NAS 9-13584).

TABLE OF CONTENTS FOR SECTION 3.1, PROGRAM PANEL

3.1 PROGRAM PANEL: A COMPUTER CODE FOR GENERATING A PANELED AEROSPACE VEHICLE SURFACE DEFINITION

Program PANEL is a general purpose external geometry definition program developed for use in large scale preliminary design simulations. The PANEL program consists essentially of the geometry subroutines from the Reference 1 hypersonic aerodynamics program converted to the form of an independent program. Complete analytic details are available in Reference 1.

The independent PANEL program produces a vehicle surface definition in the form of a sequence of quadrilateral panels defined by their four corner points. The resulting corner point data is acceptable as input to the original arbitrary hypersonic aerodynamic program of Reference 1. Figure 3.1-1 illustrates the type of surface paneling which is employed in the program. The data may be readily converted to the form required by other technology programs in the ODIN/RLV system. This will require the construction of appropriate interface routines. Alternately, parallel scaling of the PANEL geometry and other program geometric inputs may be employed through the data base.

The program accepts a variety of input data varying from detailed definition of individual panel corner points to a selection of generalized two- and three-dimensional shapes. The two-dimensional section data includes circular, elliptical, and arbitrary cross section definition. A bivariate cubic surface element is included which allows relatively large sections of the vehicle surface to be described by a small amount of input data. With the cubic surface element the input data for the vehicle section is mathematically fitted with boundary matched cubic functions. The cubic function is then reduced to smaller distributed quadrilateral panels.

The unit outward normal vector to each quadrilateral panel is also computed. Since the quadrilateral corner points do not necessarily lie in a plane, a "mean unit normal" is computed. This mean normal is defined by the condition that it is normal to both diagonals of the quadrilateral element and is positioned at the centroid of the mean panel surface.

Some typical aerospace vehicles which have been reduced to quadratic element surface representations are presented in Figure 3.1-2. This figure is reproduced from Reference 1. The PANEL program is outlined below. A more detailed description of the program is contained in Reference 2.

3.1.1 Approach Employed in Program PANEL

This section discusses a collection of techniques suitable for the design of fairly arbitrary geometric solid shapes within the computer. The geometric definition of an aerospace vehicle fuselage, wings, and control surfaces requires the description of a series of surfaces of considerable subtlety and complexity. The geometric definition of such a vehicle is traditionally carried out by manual projective geometry procedures. These procedures are very laborious and entail a large number of graphical iterations in order to assure that the surfaces are

a. Completely described
b. Smooth
c. Satisfy the internal packaging constraints

These graphical iterations involve construction of consistant water lines, buttock lines, and sections by manual methods. The mathematical basis for the surfaces in program PANEL have been devised to automate the surface design process itself. From the designer's standpoint the surface definition process is natural and fairly easy to use. Yet, these definitions provide a geometric description which can be interfaced with other ODIN/RLV technology programs in a unified manner by consistant scaling through the data base.

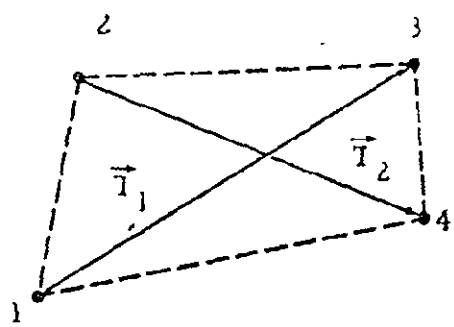
The surface defining mathematics of Reference 1 are straightforward but time consuming for hand calculation. However, the required calculations are rapidly performed on a large scale digital computer.

3.1.2 The Surface Element Geometry Method

The basic geometry method used by the PANEL program is the surface element or quadrilateral method developed in Reference 1. The coordinate system employed is a right handed Cartesian system as shown in Figure 3.1-3. The vehicle is usually positioned with its nose at the coordinate system origin and with the length of the body stretching in the negative X direction. The body surface is represented by a set of points on the body surface. A set of four related points define a quadrilateral panel which locally approximates the vehicle surface. If all such quadrilateral panels are drawn, the vehicle surface shape is revealed as in Figures 3.1-1 and 3.1-2.

It can be seen that different areas of a vehicle require a different organization and spacing of surface points for accurate representation. Each such area or organization of elements is called a section, and each section is independent of all other sections. The division of a vehicle into a given set of sections may also be influenced by another consideration; for example, aerodynamic calculations may obtain the force contributions of each section separately, possibly using different calculation methods.

The geometrical model employed in PANEL is outlined below; more complete details may be obtained from References 1 and 2.



The i^{th} panel corner point coordinates in the reference coordinate system are given by

$$p_k^i = (x, y, z)_k^i \; ; \; k = 1, 2, 3, 4 \qquad (3.1.1)$$

The two diagonal vectors $\vec{T}_1$ and $\vec{T}_2$ components are given by

$$T1_x = x_3^i - x_1^i \qquad T1_y = y_3^i - y_1^i \qquad T1_z = z_3^i - z_1^i \qquad (3.1.2)$$

$$T2_x = x_4^i - x_2^i \qquad T2_y = y_4^i - y_2^i \qquad T2_z = z_4^i - z_2^i$$

An "average" outward normal unit vector to the panel can be obtained from

$$\vec{n} = (\vec{T}_2 \times \vec{T}_1)/|N| \quad ; \quad |N| = |\vec{T}_2 \times \vec{T}_1| \tag{3.1.3}$$

The components of $\vec{n}$ are

$$\begin{aligned} n_x &= (T2_y\ T1_z - T1_y\ T2_z)/|N| \\ n_y &= (T1_x\ T2_z - T2_x\ T1_z)/|N| \\ n_z &= (T2_x\ T1_y - T1_x\ T2_y)/|N| \end{aligned} \tag{3.1.4}$$

Specifying a point in the panel completely defines the panel plane. This point is taken as the point whose coordinates, $\bar{x}$, $\bar{y}$, $\bar{z}$ are the averages of the coordinates of the four input points.

$$\begin{aligned} \bar{x} &= \frac{1}{4}\left(x_1^i + x_2^i + x_3^i + x_4^i\right) \\ \bar{y} &= \frac{1}{4}\left(y_1^i + y_2^i + y_3^i + y_4^i\right) \\ \bar{z} &= \frac{1}{4}\left(z_1^i + z_2^i + z_3^i + z_4^i\right) \end{aligned} \tag{3.1.5}$$

The original panel defining corner points are now projected parallel to $\bar{n}$ onto the panel plane. The resulting quadrilateral completely defines the local vehicle surface representation. It can be shown that all original panel defining points are equidistant from the approximating panel.

Defining the magnitude of the common projection distance by d, the coordinates of the panel corner points in the reference coordinate system are given by

$$\begin{aligned} x_k' &= x_k^i + n_x\ d_k \\ y_k' &= y_k^i + n_y\ d_k \qquad\qquad k = 1,\ 2,\ 3,\ 4 \\ z_k' &= z_k^i + n_z\ d_k \end{aligned} \tag{3.1.6}$$

A local panel element coordinate system is now constructed by defining three mutually perpendicular unit vectors. The unit outward normal vector is taken as one of the unit vectors. One remaining vector is taken as a unit vector $\vec{t}_1$ parallel to the original diagonal vector T_1. The third unit vector which must, by definition, be normal to $\bar{n}$ and $\vec{t}$ is defined by $\vec{t}_2 = \vec{n} \times \vec{t}_1$. The vector $\vec{t}_1$ defines a [illegible] or ξ axis; $\vec{t}_2$ defines the y or η axis, and n defines the z or ζ axis of this coordinate system.

To transform the coordinates of points and the components of vectors between the reference coordinate system and the element coordinate system, a transformation matrix is required. The elements of this matrix are the components of the three basic unit vectors, $\vec{t}_1$, $\vec{t}_2$, and $\vec{n}$. Define

$$
\begin{aligned}
a_{11} &= t1_x & a_{12} &= t1_y & a_{13} &= t1_z \\
a_{21} &= t2_x & a_{22} &= t2_y & a_{23} &= t2_z \\
a_{31} &= n_x & a_{32} &= n_y & a_{33} &= n_z
\end{aligned}
\qquad (3.1.7)
$$

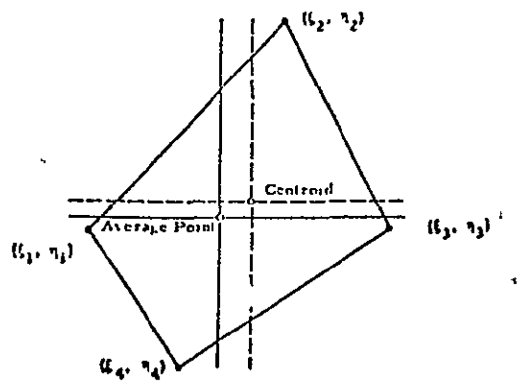
The transformation matrix is

$$
[A] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \qquad (3.1.8)
$$

To transform the coordinates of points from one system to the other, the coordinates of the origin of the element coordinate system in the reference coordinate system are required. Let these be noted x_o, y_o, z_o. Then, if a point has coordinates x', y', z' in the reference coordinate system and coordinates x, y, z in the element coordinate system,

$$
\begin{bmatrix} x \\ y \\ z \end{bmatrix} = [A] \begin{bmatrix} x'-x_o \\ y'-y_o \\ z'-z_o \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = [A] \begin{bmatrix} x & & x_o \\ y & + & y_o \\ z & & z_o \end{bmatrix} \qquad (3.1.9)
$$

The corner points can be transformed into the element coordinate system in the above manner. These points have coordinates x_k', y_k', z_k' in the reference coordinate system. Their coordinates in the element coordinate system are denoted by ξ_k^*, η_k^*, 0. They have a zero, z, or ξ coordinate in the element coordinate system because they lie in the plane of the element. This is illustrated in the diagram below. The origin of the element coordinate system is now transferred to the centroid of the area of the quadrilateral.

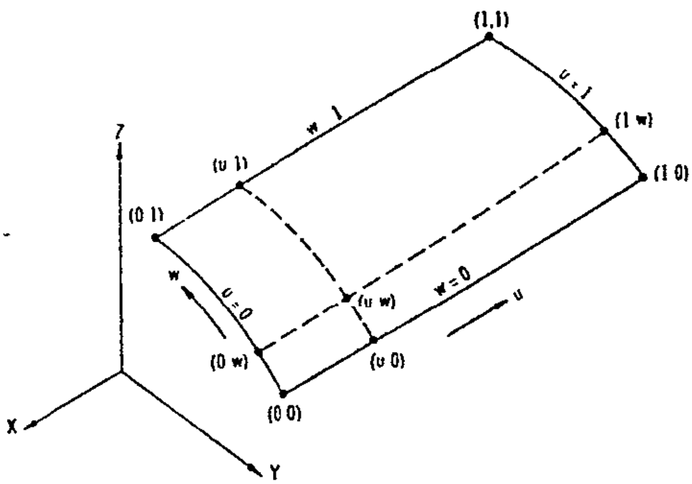


ORIGINAL PAGE IS
OF POOR QUALITY

3.1.3 Parametric Cubic

A second technique for describing three-dimensional curved surfaces is provided within the program. This is a mathematical surface-fit technique identified as the parametric cubic method or cubic patch method. The method is adopted from the formulation given by Coons of MIT, Reference 3. In this method a vehicle shape is also divided into a number of sections or patches. The size and location of each patch depends upon the shape of the surface. Only the surface conditions at the patch corner points are required to completely describe the surface enclosed by the boundary curves of the patch. The basic problem is the determination of all the information required at these corner points, i.e., the surface equation requires corner point surface derivatives with respect to the parametric variables rather than the X, Y, Z coordinates. This has been solved by the use of additional points along the boundary curves.

The geometrical representation of a surface patch is illustrated below.



BOUNDARY CURVE (FOR $u = 0$)

$$X_i(0, w) = Aw^3 + Bw^2 + Cw + D$$

$$A = 2[X_i(0,0) - X_i(0,1)] + \frac{\partial X_i}{\partial w}(0,0) + \frac{\partial X_i}{\partial w}(0,1)$$

$$B = 3[X_i(0,1) - X_i(0,0)] - 2\frac{\partial X_i}{\partial w}(0,0) - \frac{\partial X_i}{\partial w}(0,1)$$

$$C = \frac{\partial X_i}{\partial w}(0,0) \qquad D = X_i(0,0)$$

$$\frac{\partial X_i}{\partial w} = \frac{\partial X_i}{\partial S}\frac{\partial S}{\partial w} \qquad i = 1, 2, 3 \text{ FOR } X, Y, Z$$

BLENDING FUNCTIONS

$$F_1(u) = 3u^2 - 2u^3 \qquad F_1(w) = 3w^2 - 2w^3$$

$$F_0(u) = 1 - F_1(u) \qquad F_0(w) = 1 - F_1(w)$$

SURFACE FORM

$$X_i(u, w) = X_i(0,w)F_0(u) + X_i(1,w)F_1(u) + X_i(u,0)F_0(w)$$
$$+ X_i(u,1)F_1(w) - X_i(0,0)F_0(u)F_0(w)$$
$$- X_i(0,1)F_0(u)F_1(w) - X_i(1,0)F_1(u)F_0(w)$$
$$- X_i(1,1)F_1(u)F_1(w)$$

The basic surface-fit equations and their derivatives are presented in Reference 1 and are outlined in the diagram above.

To summarize, each set of four points is converted into a plane-quadrilateral element. The normal to the quadrilateral is taken as the cross product of two diagonal vectors formed between opposite element points. The order of the input points and the manner of defining the diagonal vectors is used to insure that the cross product gives an outward normal to the body surface. The next step is to define the plane of the element by determining the averages of the coordinates of the original four corner points. These points are then projected parallel to the normal vector into the plane of the element to give the corners of the plane quadrilateral. The corner points of the quadrilateral are equidistant from the four points used to form the element. Additional parameters which may be required for subsequent aerodynamic force calculations, quadrilateral area and centroid, are then calculated.

When using this method, the corner points of the panels are input individually for each panel or in groups of individual panels. This is illustrated in Figure 3.1-4. The points on the body surface are input in rows and columns. The number of panels in the whole section is defined by the number of rows of panels times the number of panels per row. The orientation of the geometric section is optional but two rules must be followed regardless of the orientation:

1. Points along a row are input sequentially upward

2. Rows of points are input sequentially to the right

These rules, illustrated in Figure 3.1-4 apply whether the points are input streamwise, chordwise, or along cross section lines or any other arbitrary orientation.

The cubic patch geometry input option is provided as an alternate method for description of arbitrary shapes. It serves a similar purpose as the surface element input method. In the panel corner point input method, a vehicle's section is described by a large number of surface points organized in panel fashion. In the cubic patch method only points along the boundaries of a patch are input to the program, and the distributed surface points required for the subsequent panel calculations are determined by the program.

The basic features of the cubic patch method are that

1. fewer input points are required to describe a surface

2. the generated panel size is controlled by two input parameters which may be changed to meet the requirements of the problem.

The input consists of coordinate points along each of the four boundaries of a patch. The program calculates the coefficients for a mathematical surface fit equation developed in Reference 1 to provide a description of the interior surface of the patch. This surface is then converted into exactly the same form as the surface panel input data of Section 3.1.2. The panel data generated can be merged with other panel data generated before or after it by any available method.

Figure 3.1-5 illustrates how a section is described by this method. Each of the four boundaries is identified in this figure: two in the w direction and two in the u direction. The user orients the model of the vehicle so that the Number 1 boundary is to the left and the Number 2 boundary is to the right. The order of the points is from the bottom to the top of the patch. Note that a *point must be included outside the patch at either end of the boundary to give proper slopes at the corner points.* Boundaries 3 and 4 are loaded from left to right. *A different number of points may be used to describe each boundary up to a maximum of 20 for each.*

Each boundary curve must be extended by one point on each end to permit the computation of end point derivatives. The second point and the next to the last point on each curve must be common to the adjacent curves, as illustrated in Figure 3.1-5. The program generates equally spaced panels based on arbitrary numbers of rows and columns of panels selected by the user.

NPTS - number of points in each row

NSETS - number of rows or sets of points

XA - array of x points (usually negative) for the current row

YA - array of y points for the current row

ZA - array of z points for current row

LAST - status flag for merging sections
= 0, this section will be merged with the next to form a single section
= 3, this section will not be merged with the next

Note that namelist input parameters which are unchanged from previous values need not be input. Therefore, if the arrays of x points do not change from row to row, for example, they need not be input.

3.1.4 Elliptic Cross Section Method

This method allows the user to generate panel information for partially or completely elliptical cross sections. The surface of the section is described by an ellipse centered at some point off the reference axis and defined by the major and minor axis as shown in Figure 3.1-6. The portion of the reference ellipse used to define the body section is defined by the angular difference between θ_O and θ_L measured from the negative z axis. A sequence of two or more sections describe a surface. The PANEL program generates the panel geometry for an arbitrary number of sections. Each Section can be equally divided into an arbitrary number of diversions.

REFERENCES:

1. Hague, Donald S. and Glatt, C. R., Optimal Design Integration of Military Flight Vehicles, Section 4.1 "Hypersonic Arbitrary Body Aerodynamic Computer Program," AFFDL-TR-72-132, December 1972.
2. Hague, D. S. and Glatt, C. R., Optimal Design Integration of Military Flight Vehicles, Section 3.1, "A Computer Code for Generating Panelled Aerospace Vehicle Surfaces," AFFDL-TR-72-132, December 1972.
3. Coons, Steven A., Surfaces for Computer-Aided Design of Space Forms, MAC-TR-41, Massachusetts Institute of Technology, June 1967.

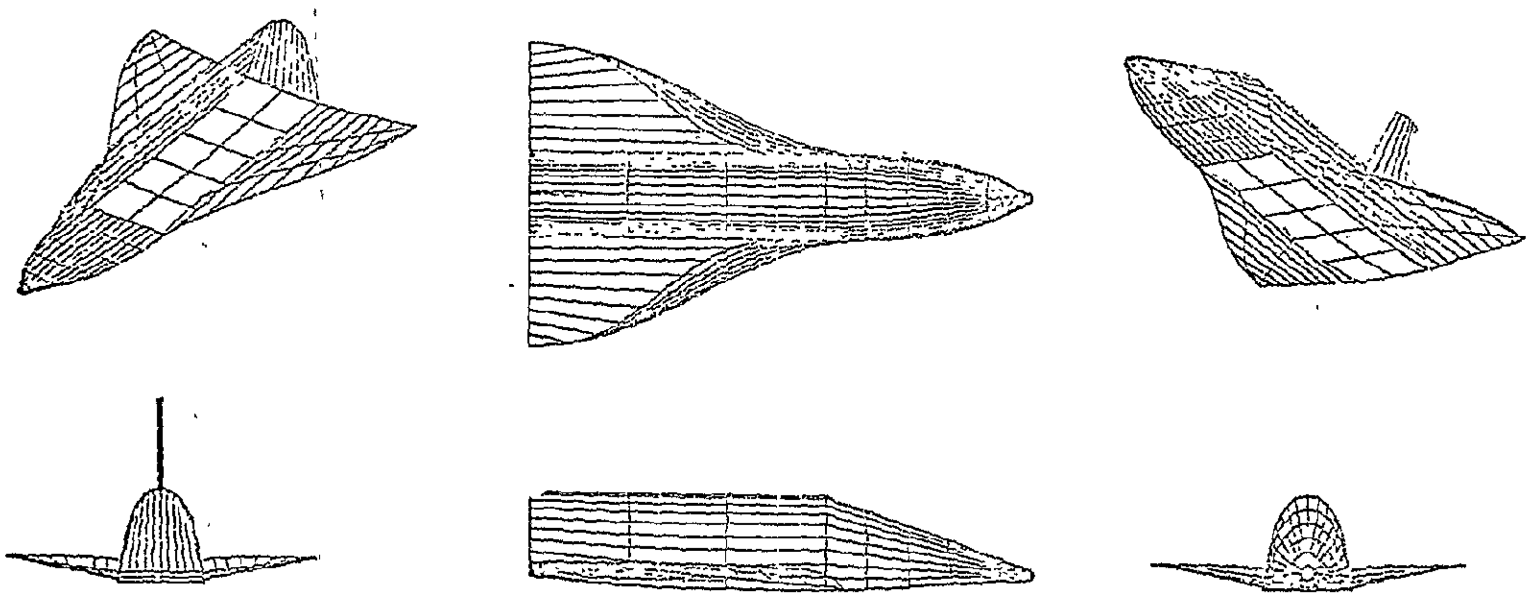


FIGURE 3.1-1. TYPICAL QUADRILATERAL ELEMENT REPRESENTATION OF A VEHICLE SURFACE IN PROGRAM PANEL

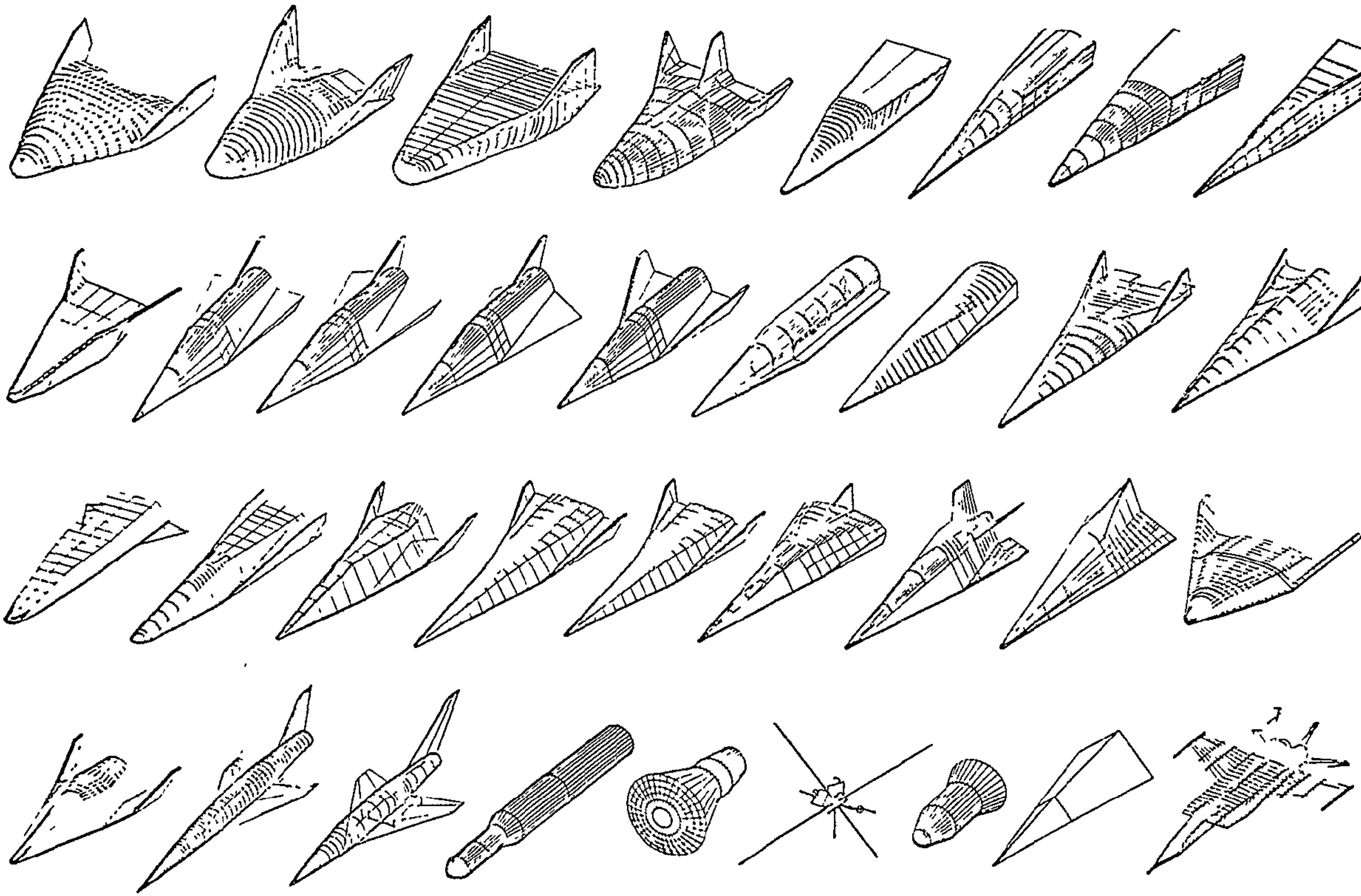
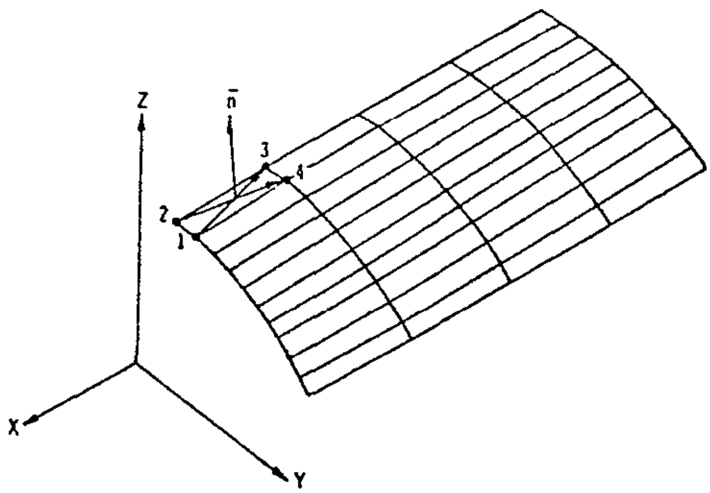


FIGURE 3.1-2 A SELECTION OF CONFIGURATIONS THAT HAVE BEEN STUDIED WITH THE HYPERSONIC ARBITRARY-BODY PROGRAM



DIAGONAL VECTORS $\bar{T}_1$ and $\bar{T}_2$

$$T_{1x} = X_3 - X_1 \quad T_{1y} = Y_3 - Y_1 \quad T_{1z} = Z_3 - Z_1$$
$$T_{2x} = X_4 - X_2 \quad T_{2y} = Y_4 - Y_2 \quad T_{2z} = Z_4 - Z_2$$

UNIT NORMAL $\quad \bar{N} = \bar{T}_2 \times \bar{T}_1$

$$N_x = T_{2y}T_{1z} - T_{1y}T_{2z} \qquad n_x = N_x/N$$
$$N_y = T_{1x}T_{2z} - T_{2x}T_{1z} \qquad n_y = N_y/N$$
$$N_z = T_{2x}T_{1y} - T_{1x}T_{2y} \qquad n_z = N_z/N$$

$$N = \sqrt{N_x^2 + N_y^2 + N_z^2}$$

AVERAGE POINT

$$\bar{x} = \tfrac{1}{4}(X_1 + X_2 + X_3 + X_4)$$
$$\bar{y} = \tfrac{1}{4}(Y_1 + Y_2 + Y_3 + Y_4)$$
$$\bar{z} = \tfrac{1}{4}(Z_1 + Z_2 + Z_3 + Z_4)$$

CORNER POINT PROJECTION DISTANCE

$$d_k = n_x(\bar{x} - X_k) + n_y(\bar{y} - Y_k) + n_z(\bar{z} - Z_k)$$
$$k = 1, 2, 3, 4$$

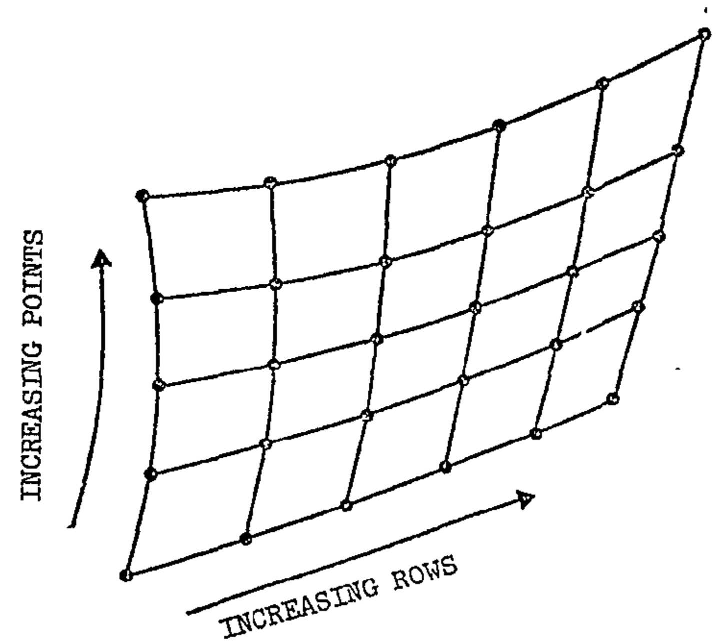
CORNER POINT COORDINATES

$$X_k' = X_k + n_x d_k$$
$$Y_k' = Y_k + n_y d_k$$
$$Z_k' = Z_k + n_z d_k$$

FIGURE 3.1-3. REFERENCE COORDINATE SYSTEM

1. POINTS ALONG A ROW ARE INPUT SEQUENTIALLY UPWARD

2. ROWS ARE INPUT SEQUENTIALLY TO THE RIGHT



• CORNER POINTS

FIGURE 3.1-4. RULES FOR CORNER POINT GEOMETRY INPUT

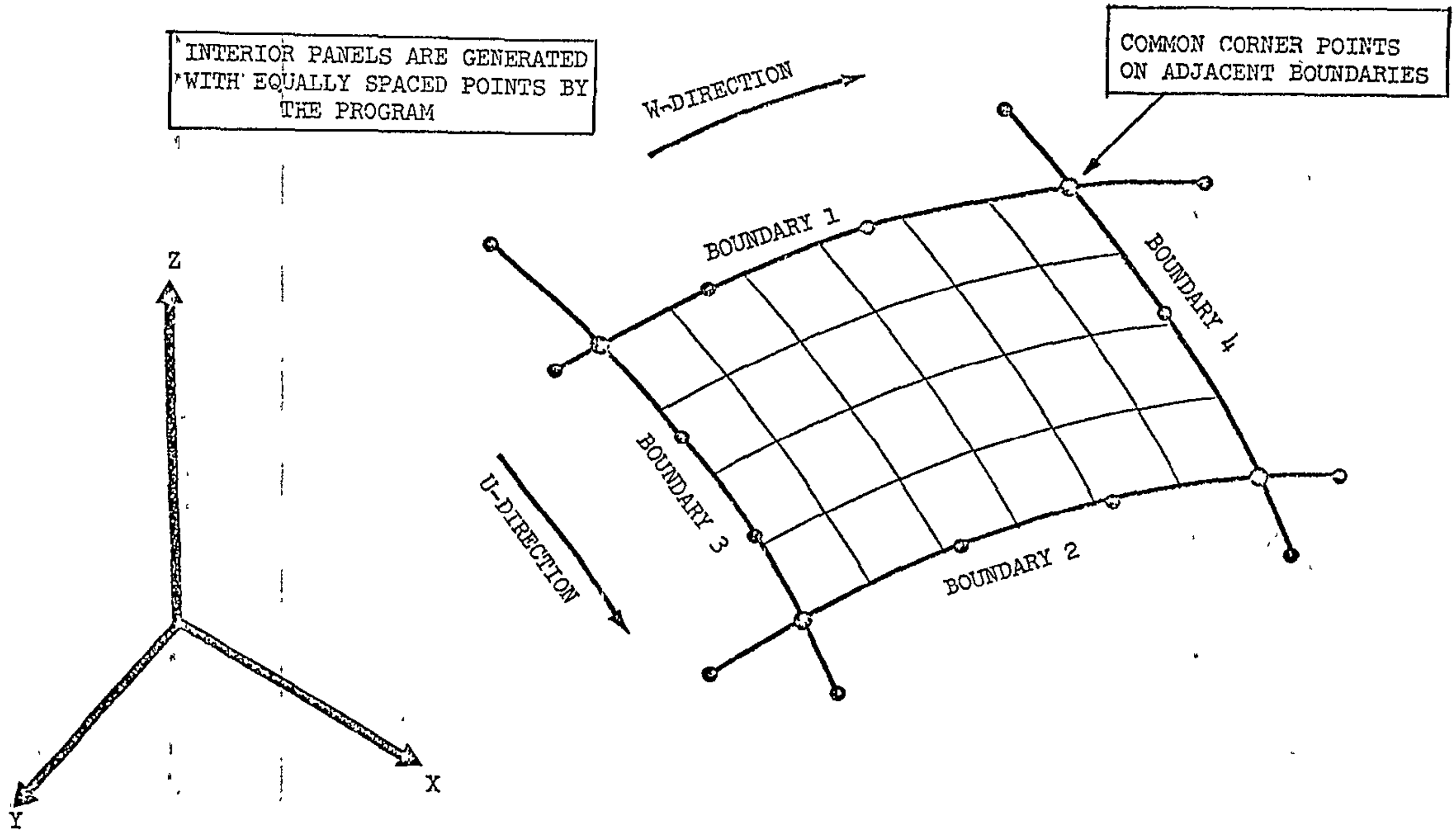


FIGURE 3.1-5. INPUT DESCRIPTION FOR THE CUBIC PATCH TECHNIQUE

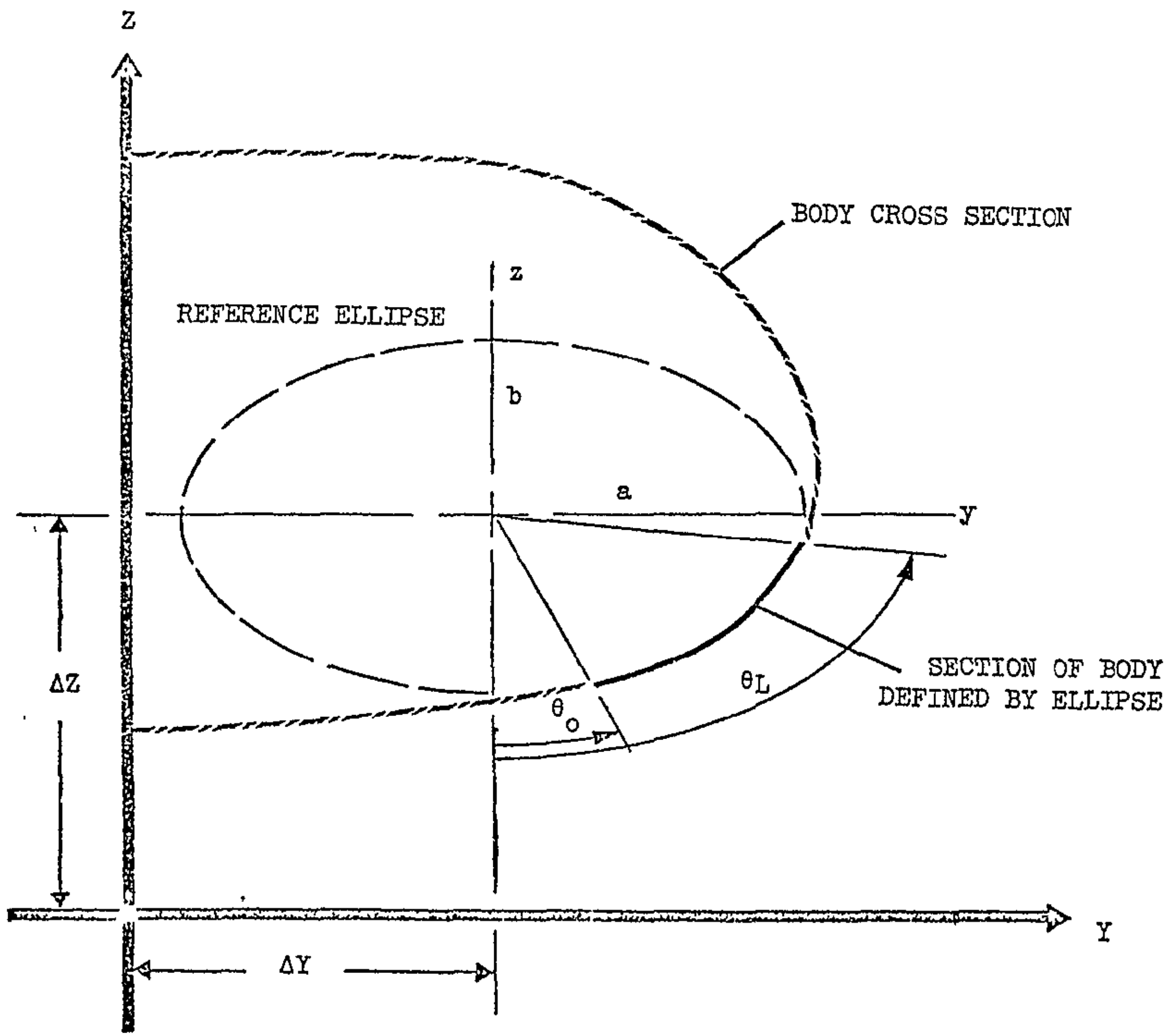


FIGURE 3.1-6. GEOMETRY DEFINITION FOR ELLIPTIC METHOD

TABLE OF CONTENTS FOR PROGRAM IMAGE, SECTION 3.2

3.2 PROGRAM IMAGE: A COMPUTER CODE FOR DISPLAY OF THREE-DIMENSIONAL OBJECTS

Program IMAGE employs the detailed panelled representation of a three-dimensional object to provide an on-line or off-line display of the object's image. The detailed paneling of any three-dimensional object can be accomplished through the PANEL program of Section 3.1. Programs PANEL and IMAGE are completely compatible with each other. On-line displays are presented on cathode ray tube devices; off-line displays may be obtained on CALCOMP, COMPLOT, or SC4020 plotting devices. The three-dimensional image of an object may be rotated to any orientation relative to the viewer. By running a sequence of cases in which the viewing aspect changes the image may be rotated for inspection purposes. By forming the head on, side view, and planform views of the vehicle, a three-view is obtained.

The views displayed may include hidden lines which the viewer cannot see directly. Alternately, on convex objects the hidden lines may be deleted. On non-convex objects only those lines which represent panels whose unit normal faces away from the viewer may be omitted.

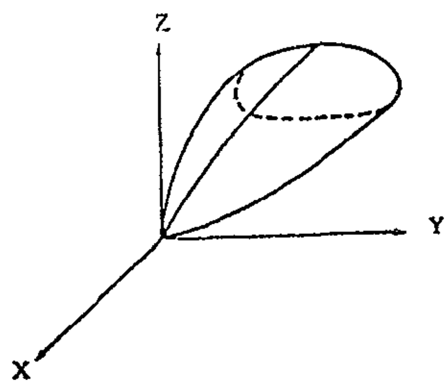
It should be noted that program IMAGE is based on the graphics package of the Reference 1 program and is compatible with the panelled geometry model of Reference 2. The program has been extended to incorporate display capability on CALCOMP, COMPLOT, and certain cathode ray tube displays at installations which have the necessary software and hardware for these devices during the ODIN study. The analytic program basis is unaltered by the type of display device being employed.

3.2.1 Method for Obtaining an Image

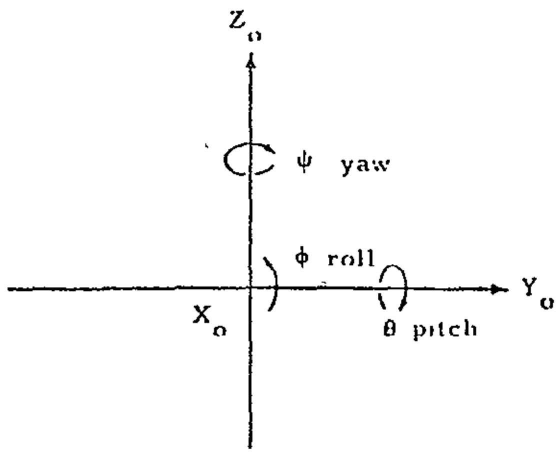
Each point on the surface is described by its coordinates in the body reference coordinate system.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

The body reference coordinate system is assumed to be a conventional right handed Cartesian system as defined in Section 3.1; for example



To create the image each surface point on the body must be rotated to the desired viewing angle and then transformed into a coordinate system in the plane of the paper. With zero rotation angles the body coordinate system is coincident with the fixed system in the plane of the paper.



The rotations of the body and its coordinate system to give a desired viewing angle are specified by a yaw-pitch-roll sequence (ψ, θ, φ). This rotation is given by the following relationship:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = [\phi]\ [\theta]\ [\psi] \begin{bmatrix} X_o \\ Y_o \\ Z_o \end{bmatrix} \tag{3.2.1}$$

Where the rotation matrices are

$$[\psi] = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.2.2}$$

$$[\theta] = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \tag{3.2.3}$$

$$[\phi] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \tag{3.2.4}$$

or

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = [E] \begin{bmatrix} X_o \\ Y_o \\ Z_o \end{bmatrix} \qquad (3.2.5)$$

where

$$[E] = [\phi]\ [\theta]\ [\psi] \qquad (3.2.6)$$

Since each point on the surface is given by its coordinates in the X, Y, Z system, its position in the fixed coordinate system (X_o, Y_o, Z_o) may be found by inverting the above process.

$$\begin{bmatrix} X_o \\ Y_o \\ Z_o \end{bmatrix} = [E]^{-1} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \qquad (3.2.7)$$

Carrying out this operation

$$\begin{bmatrix} X_o \\ Y_o \\ Z_o \end{bmatrix} \left[\begin{array}{c|c|c} \cos\theta\cos\psi & -\sin\psi\ \cos\phi+\sin\theta\cos\psi\ \sin\phi & \sin\psi\ \sin\phi+\sin\theta\cos\psi\ \cos\phi \\ \hline \cos\theta\sin\psi & \cos\psi\ \cos\phi+\sin\theta\sin\psi\ \sin\phi & -\cos\psi\ \sin\phi+\sin\theta\sin\psi\ \cos\phi \\ \hline -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{array}\right] \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \qquad (3.2.8)$$

or

$$X_o = X(\cos\theta\cos\psi) + Y(-\sin\psi\cos\phi+\sin\theta\cos\psi\sin\phi) + Z(\sin\psi\sin\phi+\sin\theta\cos\psi\cos\phi) \qquad (3.2.9)$$

$$Y_o = X(\cos\theta\sin\psi) + Y(\cos\psi\cos\phi+\sin\theta\sin\psi\sin\phi) + Z(-\cos\psi\sin\phi+\sin\theta\sin\psi\cos\phi) \qquad (3.2.10)$$

$$Z_o = X(-\sin\theta) + Y(\cos\theta\sin\phi) + Z(\cos\theta\cos\phi) \qquad (3.2.11)$$

These last two equations are used to transform a given point on the body (X, Y,Z) with a specified set of rotation angles (ψ, ϕ, θ) into the plane of the paper (the Y_o, Z_o system). With the appropriate SC4020, CALCOMP, COMPLOT, or cathode ray tube library subroutines, these data can be plotted, and related points can be connected by straight lines.

In the PANEL program of Section 3.1 the actual surface of an object has been replaced by a set of surface approximating panels. The panel characteristics include the area, centroid, and the direction cosines of the surface unit normal. The surface unit normals may be transformed through the required rotation angles and the component of the unit normal in the X_o direction (out of the plane of the paper) may be found from the following equation.

$$n_{x_o} = n_x(\cos\theta\cos\psi)+n_y(-\sin\psi\cos\phi+\sin\theta\cos\psi\sin\phi)+n_z(\sin\psi\sin\phi+\sin\theta\cos\psi\cos\phi) \quad (3.2.12)$$

where n_x, n_y, n_z are the components of the surface unit normal in the vehicle reference system.

If n_{x_o} is positive, then the surface element is facing the viewer. If n_{x_o} is negative, the element faces away from the plane of the paper. This result is used in the program to provide the capability of deleting most of those elements on a vehicle that normally could not be seen by a viewer. The resulting picture is thus made more realistic,and confusing elements which are on the back side of the vehicle do not appear. No criterion is provided, however, for the deletion of those elements that face the viewer but are blocked by other body components. This may be accomplished by a proper selection of viewing angle or by a physical deletion of the offending section from the input data.

REFERENCES:

1. Hague, D. S. and Glatt, C. R., Optimal Design Integration of Military Flight Vehicles, ODIN/MFV, Section 4.1 "Hypersonic Arbitrary Body Aerodynamic Computer Program," AFFDL-TR-72-132, December 1972.

2. Hague, D. S. and Glatt, C. R., Optimal Design Integration of Military Flight Vehicles, ODIN/MFV, Section 3.1, AFFDL-TR-132, December 1972.

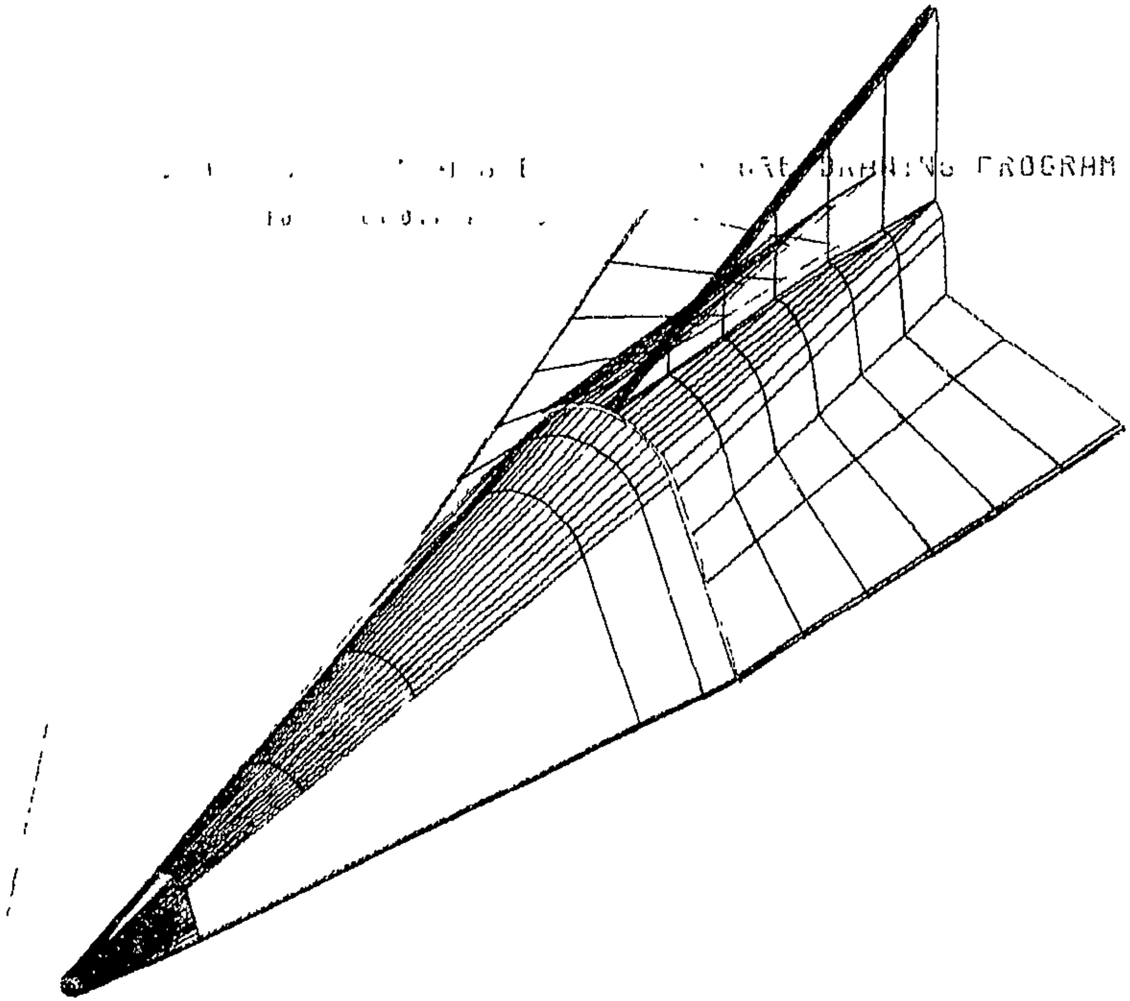


FIGURE 3.2-1 TYPICAL OUTPUT FROM THE PLOTTR PROGRAM

TABLE OF CONTENTS FOR SECTION 3.3 PROGRAM LRCACP

3.3 PROGRAM LRCACP: A CODE FOR PRODUCING AIRCRAFT CONFIGURATION PLOTS

Program LRCACP is the NASA Langley Research Center developed aircraft configuration plot program of Reference 1. The code has a wider range of image drawing options than the combination of programs PANEL and IMAGE. In particular it has the ability to produce views which incorporate true perspective, to produce stereoscopic pair views, and to automatically produce a well laid out three-view.

The LRCACP code interfaces directly to several well established subsonic and supersonic aerodynamic estimation programs. Hence, it complements the program PANEL which is limited to a hypersonic aerodynamic estimation program. The program description presented below is based on that of Reference 1. Since the geometrical methods are similar to the methods of Section 3.1, mathematical detail is omitted.

3.3.1 Method of Producing Vehicle Images

The LRCACP program contains the following types of plotting capability:

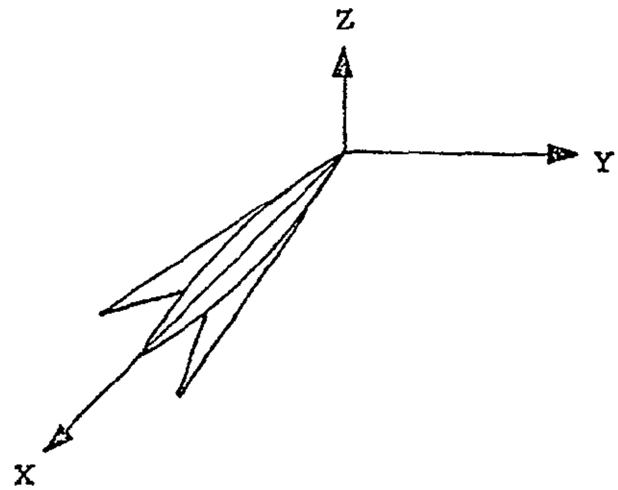
1. Three-views
2. Orthographic, from an arbitrary viewing angle
3. Perspective, from an arbitrary viewing angle
4. Stereoscopic, from an arbitrary viewing angle

The program interfaces through the CDC 6600 to the following types of equipment:

1. On-line cathode ray tube
2. CALCOMP plotter
3. Houston COMPLOT plotter, through an ODIN-developed module
4. Gerber plotter
5. Stereoscope

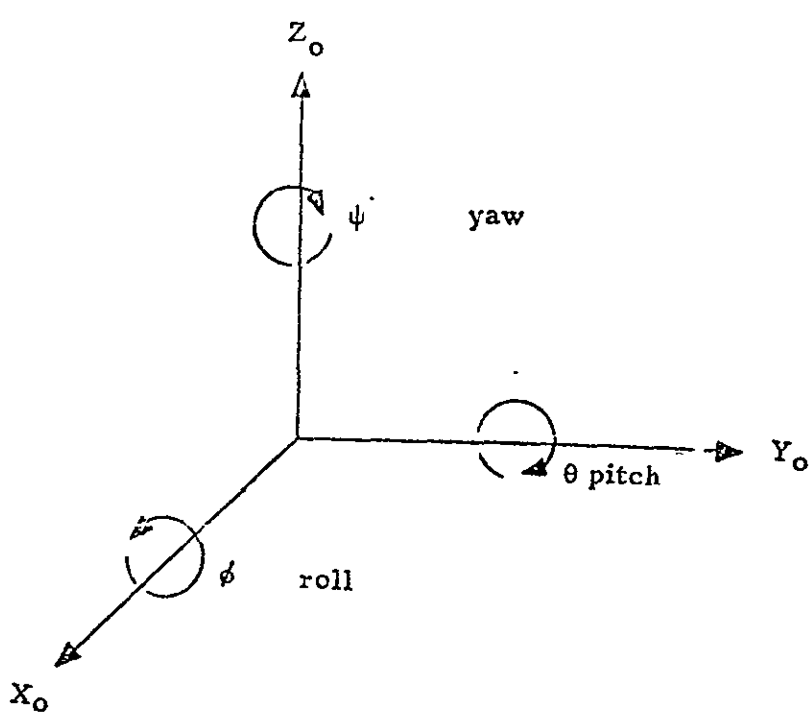
The numerical model of the aircraft configuration may include any combination of components: wing, body, pods, fins, and canards. The wing is made up of airfoil sections; the body is defined by either circular or arbitrary sections. The pods are defined similar to the fuselage, and fins and canards are defined similar to the wings. The vehicle geometric specification is converted into a set of quadrilateral panel elements in a manner similar to that described in Section 3.1

The configuration is usually positioned with its nose at the coordinate system origin and with the length of the body stretching in the positive X direction. The coordinate system used for this program is a right handed Cartesian system as illustrated below.



Related points in the plotted arrays are connected by straight lines; therefore, sufficient points must be given to approximate a desired curve.

Orthographic projections are created by rotating each point on the body surface to the desired viewing angle and then transforming the points into a coordinate system in the plane of the paper. The rotations of the body and its coordinate system to give a desired viewing angle are specified by angles of roll, pitch, and yaw (ϕ, θ, ψ) using the convention below.



The code computes the "average" unit normal vector to each panel. The resulting set of vectors may be used to provide the capability of deleting most elements on the surface of the configuration which would not be seen by a viewer. By this device a user may remove many confusing panel elements. No provision is made for deleting components hidden by other components or for deleting portions of an element at the present time.

When three-views are requested, the plan, front, and side views are provided in a compact and pleasing to the eye arrangement. An option is provided for the orthographic projections of these three-views to be spaced one above the other. A typical three-view obtained in this manner has been presented in Figure 3.3-1.

The perspective views represent the projection of a given three-dimensional array. The two-dimensional view is constructed relative to a viewing point and a focal point specified by coordinate points in the data coordinate system. Data are scaled to the viewer page size automatically by the specification of the viewing field diameter and the viewing field distance. The coordinates of the viewing point determine the position from which the data array will be viewed and the coordinate values of the focal point control the direction and focus. The size of the projection on the viewing plane reflects the distance between the viewing point and the focal point. Data which are within the cone of the viewing plane but not in the immediate range of the focal point may be distorted. Perspective may be eliminated by specifying a large viewing field distance. A typical detailed orthographic projection of a modern fighter aircraft is presented in Figure 3.3-2.

The above explanation of the perspective plots also applies to the stereo views. The use of the stereo option causes the program to be executed twice in setting up two plots for the left and right frames. These frames are suitable for viewing in a stereoscope. A representative stereoscopic pair frame is presented in Figure 3.3-3.

REFERENCES:

1. Craidon, Charlotte B., Description of a Digital Computer Program (D2290) for Aircraft Configuration Plots, NASA TM X-2074, 1972.

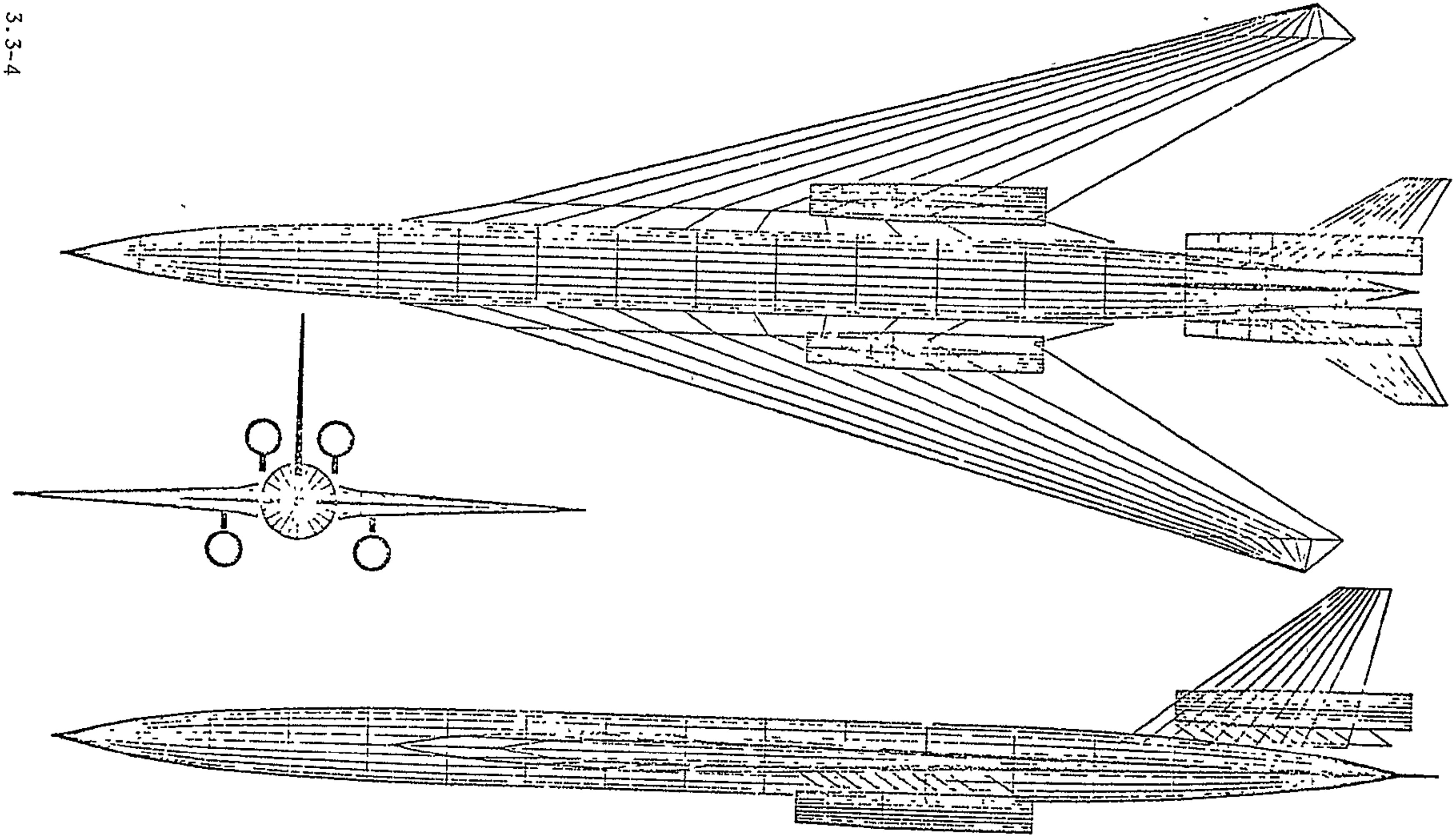


FIGURE 3.3-1. TYPICAL THREE-VIEW ORTHOGRAPHIC REPRESENTATION OF AN AIRCRAFT CONFIGURATION

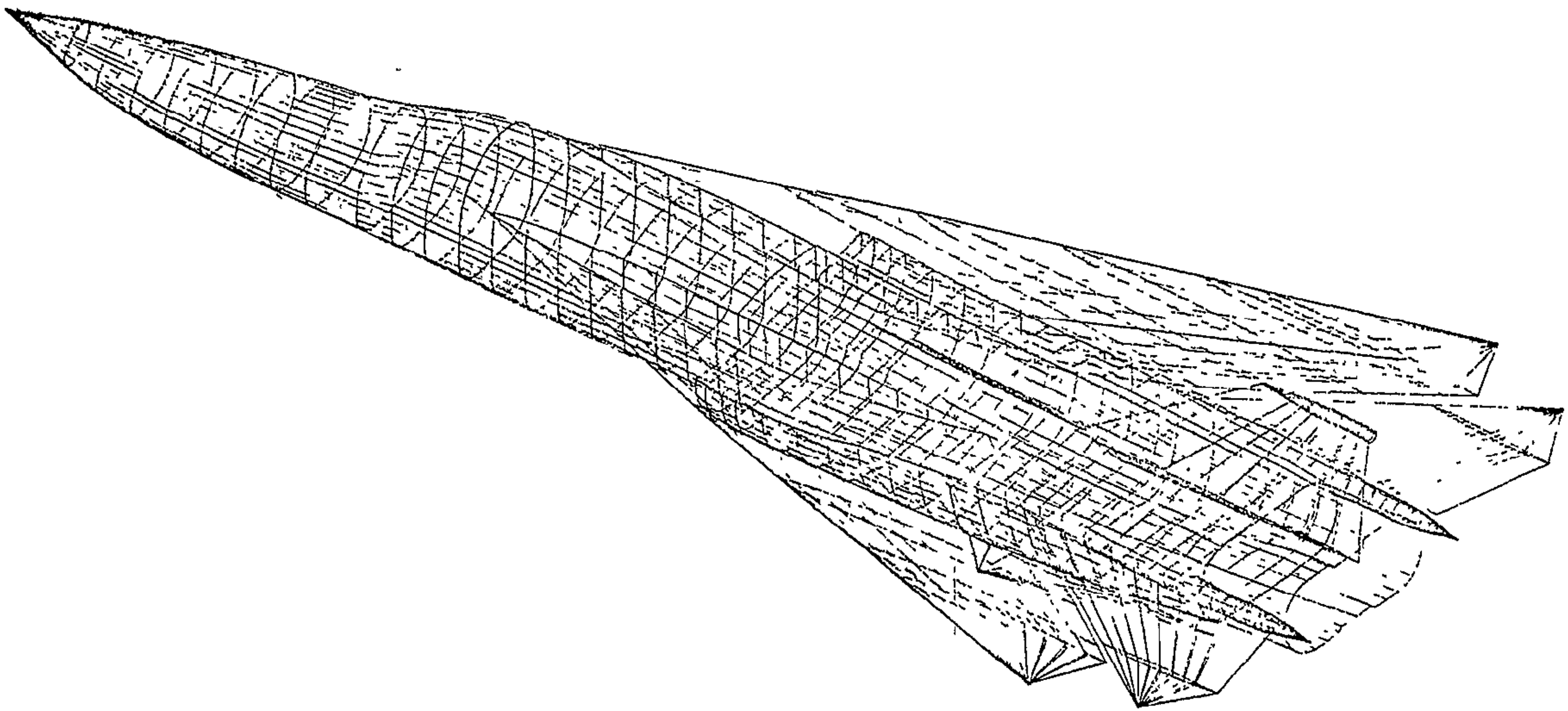


FIGURE 3.3-2. PERSPECTIVE VIEWS OF A REPRESENTATIVE AIRCRAFT CONFIGURATION

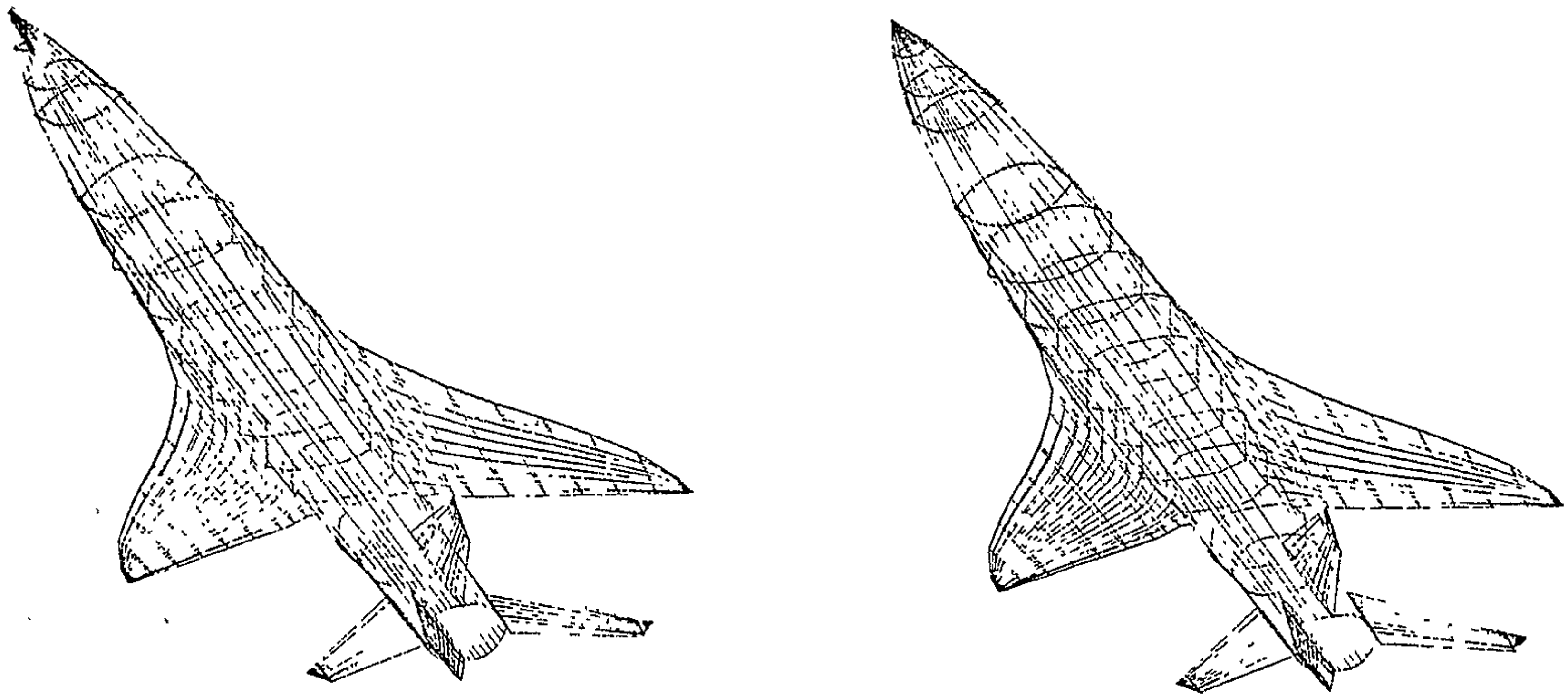


FIGURE 3.3-3. EXAMPLES OF STEREO FRAMES FOR A THREE-DIMENSIONAL VIEW USING AN AIRCRAFT CONFIGURATION

SECTION 4

AERODYNAMICS

The ODIN/RLV program library contains seven well proven independent aerodynamic estimation programs covering flight in subsonic, supersonic, and hypersonic regimes. Program sources include past Air Force Flight Dynamics Laboratory studies, NASA developed programs, and Aerophysics Research Corporation. Programs are provided for

1. Hypersonic viscous and pressure forces
2. Rapid supersonic zero-lift wave drag
3. Detailed supersonic zero-lift wave drag

The ODIN/RLV system, as installed at Langley Research Center, also contains the Reference 1 programs for computing

4. Wave drag at lift
5. Wetted areas
6. Skin friction drag
7. Rapid subsonic, supersonic, and hypersonic aerodynamic trend analyses.

Programs 1 through 3 , now available in the ODIN/RLV, are outlined below; for complete details, reference should be made to the original source documents. Programs 4 through 7 are described in NASA internal documents and in Reference 1.

REFERENCES:

1. Harris, R. V., Jr., An Analysis and Correlation of Aircraft Wave Drag, NASA TM X-947, 1964.

TABLE OF CONTENTS FOR SECTION 4.1, PROGRAM HABACP

4.1 PROGRAM HABACP: THE GENTRY HYPERSONIC ARBITRARY BODY AERODYNAMICS COMPUTATION PROGRAM

Vehicle hypersonic aerodynamic characteristics may be computed by means of the arbitrary hypersonic body program of Reference 1. The description below follows that originally given by Arvel Gentry of McDonnell-Douglas, Long Beach. The program of Reference 1 treats the vehicle surface as a collection of quadrilateral elements oriented tangentially to the local vehicle surface in the manner of Reference 2. Picture drawing capability is provided as in Reference 3. Each individual panel may have its local pressure coefficient specified by any of a variety of pressure calculation methods including modified Newtonian, blunt-body Newtonian, Prandtl-Meyer, tangent-wedge, tangent-cone, boundary layer-induced pressures, free molecular flow, and a number of empirical relationships.

Viscous forces are also calculated and include viscous-inviscid interaction effects. Skin friction options include the Reference Temperature and Reference Enthalpy methods for both laminar and turbulent flow, the Spalding-Chi method (turbulent), and a special blunt body skin friction method. Control surface deflection pressures, including separation effects that may be caused by the deflected surface, are also calculated.

In addition to the above aerodynamic capabilities, the program also contains several other specialized options. Using conventional methods, the program may be used to calculate the dynamic damping derivatives, $C_{m_{\dot{\alpha}}}$ and $C_{m_{\dot{\beta}}}$ for wing-body-tail configurations. Also, since some vehicles may be strongly influenced by other applied force-vector effects (such as those caused by air breathing propulsion systems), capabilities are also provided for including these factors along with the conventionally calculated aerodynamic forces. The program output contains the following parameters as functions of angle of attack and sideslip angle: C_D, C_L, C_A, C_Y, C_N, L/D, C_m, C_ℓ C_n, C_{A_α}, C_{L_α}, C_{N_α}, C_{m_α}, C_{m_q}, C_{A_q}, C_{N_q}, C_{Y_β}, C_{n_β}, C_{ℓ_β}, C_{Y_r}, C_{n_r}, C_{ℓ_r}, C_{m_δ}, C_{ℓ_δ}, C_{y_δ}, C_{n_δ}, C_{N_δ}, $C_{m_{\dot{\alpha}}}$, $C_{Y_{\dot{\beta}}}$, and hinge moments.

The Gentry program was sponsored by the Air Force Flight Dynamics Laboratory. It has seen widespread acceptance throughout the aerospace industry. For the ODIN/RLV it represents a reasonable compromise between preliminary design requirements and the computational complexity of methods such as the three-dimensional method of characteristics program (3DMOC) of References 4 to 6.

4.1.1 Structure of the HABACP Program

The computational structure of the HABACP program is presented in Figure 4.1-1. The program employs a well organized tree of subroutines which follow functional lines. Prime interest in the ODIN/RLV system lies in the analysis method available. An outline of the subroutines which carry out the analysis function is presented below.

4.1.1.1 Control Surface Deflection Subprogram (CONTRL)

This subprogram converts input data for control-surface geometry in the undeflected position to any desired deflected position. The hinge line must be straight. The geometric characteristics of the control surface in the deflected position are stored together with necessary hinge-moment length parameters for subsequent hinge-moment calculations. The geometric characteristics of the control surface in the undeflected position are saved for subsequent calculations. The geometry data for the area in front of the control surface is computed once and saved.

4.1.1.2 Force Calculation Subprogram (FORCE)

FORCE calculates the pressure coefficient on each quadrilateral element, resolves the force in the required body axis system, and sums the contributions of each element to give the vehicle's six aerodynamic coefficients. Some of the force calculation methods require the use of another level of subprograms. The special subroutines provided include oblique-shock compression, Prandtl-Meyer expansion, Newtonian plus Prandtl-Meyer, and flow separation. Several of these subroutines serve a dual purpose since they are also used by the skin friction subprogram. The force subprogram is organized in such a way that it is very easy to modify to include additional force calculation methods.

4.1.1.3 Shock Expansion Subprogram (SHKEXP)

The shock expansion subprogram is capable of performing a shock expansion analysis along a streamwise strip of elements. The local surface pressure, local flow Mach number, and temperature are calculated for each element. The calculation of a shock expansion along a given streamwise strip of elements starts with the determination of the flow properties on the first element in the strip (the section leading edge element). The local properties on this leading edge element may be calculated either by oblique shock relationships, by tangent cone equations, by a delta wing empirical method or, in the case in which the leading element is in shadow flow, by a Prandtl-Meyer expansion from free stream conditions. The calculation of the properties on subsequent elements in a streamwise strip is based on a compression or Prandtl-Meyer expansion from the previous element in that strip.

4.1.1.4 Flow Separation Subprogram (FLOSEP)

Subprogram FLOSEP has the task of determining the effect of flow separation caused by the deflection of a control surface. The subprogram has all the necessary separation criteria built into it to provide the flow separation point on the surface, the flow reattachment position, and the change in vehicle surface pressures caused by the deflected flap and any resulting flow separation effects. The flow separation subroutine also makes use of data obtained from the shock expansion routine and the compression and temperature routines.

4.1.1.5 Skin Friction Subprogram (SKINFR)

SKINFR calculates the viscous forces with the option of using the Reference Temperature, Reference Enthalpy, or Spalding-Chi methods. The vehicle geometry is specified using the same methods as for the pressure calculation geometry model except that a smaller number of elements are used. The wall temperature may be input to the program or the radiation equilibrium value determined by the program. The local properties may be calculated by the tangent wedge, tangent cone, Prandtl-Meyer expansion, or the Newtonian plus Prandtl-Meyer method. The viscous-inviscid interaction effects are calculated by the method of White, Reference The user may specify either laminar or turbulent skin friction data to be added to the vehicle's inviscid forces.

4.1.1.6 Blunt Body Skin Friction Subprogram (BLUNT)

This subprogram calculates the viscous forces on a blunt faced body. The routine is used by the FORCE subprogram in a mode similar to the inviscid pressure calculation options. The vehicle forces calculated, however, account for only the blunt body skin friction shear forces and should be added to previously calculated forces using the data summation option.

4.1.1.7 Atmosphere Subprogram (ATMOS)

This subprogram calculates the atmospheric properties for a given altitude by using U. S. 1962 standard atmosphere. The subprogram uses an inverse square gravitational field and gets results that agree with the COESA document within one per cent at all altitudes up to 700 kilometers. The program is also capable of using input wind tunnel conditions (stagnation pressure and temperature) to determine the properties of the free stream air about a wind tunnel model.

4.1.1.8 Expansion Subprogram (EXPAND)

EXPAND calculates the pressure on a surface by using Prandtl-Meyer relationships. The routine may be called by the FORCE subprogram, by the skin friction subprogram, or by the Newtonian plus Prandtl-Meyer subprogram.

4.1.1.9 Cone Subprogram (CONE)

CONE calculates the surface conditions for a cone using empirical relationships. This routine is used by the force, flow separation, and skin friction routines when the tangent cone option is called for.

4.1.1.10 Compression Subprogram (COMPR)

COMPR calculates the pressure on a surface by using conventional oblique shock relationships (NACA TR 1135). For conditions in which no solution can be found for the oblique shock cubic relationship (for shock detachment conditions) the compression subroutine will then call the Newtonian plus Prandtl-Meyer routine in order to obtain a solution.

4.1.1.11 Blunt Body Newtonian plus Prandtl-Meyer Subprogram (NEWTPM)

This routine calculates the pressure coefficients on a surface by the blunt body Newtonian plus Prandtl-Meyer method. It is used by both the FORCE and the skin friction subprograms. Under oblique shock detachment conditions, it will also be used by the oblique shock compression routine.

This pressure calculation method requires matching the pressure distributions calculated by the modified Newtonian and Prandtl-Meyer expansion methods at the point where their slopes are equal. In the blunt part of the body before this matching point is reached, the pressure is calculated by modified Newtonian theory. When the surface slope has decreased beyond the matching point slope, the pressure is determined by Prandtl-Meyer relationships.

4.1.1.12 Temperature Subprogram (TEMP)

TEMP uses an iterative procedure to calculate the radiation-equilibrium temperature on a surface for use in the skin friction calculations. Options also permit the use of an input wall temperature or the program determined adiabatic wall condition.

4.1.1.13 Convective Heating Function Subprogram (QC)

This routine calculates the aerodynamic convective heating at a given wall temperature for laminar or turbulent flow and for either an ideal gas or a real gas. At the user's option, reference temperature or reference enthalpy methods may be used for both laminar and turbulent flow and, in addition, the Spalding-Chi turbulent method may be selected using either temperature or enthalpy ratios.

4.1.1.14 Fluid Properties Function Subprogram (ROMU)

This subprogram calculates the various fluid properties of equilibrium air required for the real gas viscous calculations. The program has three entries: the first calculates the density-viscosity product at an input pressure and enthalpy; the second calculates the enthalpy corresponding to an input temperature, and the third calculates the density at an input enthalpy and pressure.

4.1.1.15 Plunge Derivative Subprogram (PLUNGE)

PLUNGE is used to calculate the dynamic stability derivatives due to vertical acceleration ($C_{m_{\dot{\alpha}}}$) and horizontal acceleration ($C_{Y_{\dot{\beta}}}$). This is a subprogram used to calculate these derivatives by conventional analysis techniques, and the subprogram includes the calculation of the conventional interference factors for the effect of a wing in the presence of a body and the interference factor for the effect of a body in the presence of wing. The computations for $C_{m_{\dot{\alpha}}}$ involve the application of slender body theory results to the value of $C_{m_{\alpha}}$. This is also true of computations for the parameter $C_{Y_{\dot{\beta}}}$ where the PLUNGE subprogram must make use of the parameter $C_{Y_{\beta}}$ as calculated by the Arbitrary Body Program for the vehicle component involved. Since a particular body may consist of several different components, each of which may have been analyzed separately, it is necessary to wait until the final values of these two parameters ($C_{m_{\alpha}}$, $C_{Y_{\beta}}$) have been obtained.

For this reason, the plunge derivative subprogram should not be called until the user indicates that the necessary vehicle component computations have been completed.

4.1.1.16 Thrust Vector Subprogram (VECTOR)

VECTOR may be used to introduce propulsion system effects into the aerodynamic analysis. This subroutine reads in input data that give the magnitude of each applied force vector, its direction, and its point of application on the vehicle relative to the center of gravity. The subprogram will then convert this information into the required force and moment coefficients for summation with the basic vehicle characteristics. To make the solution more general, any number of input force vectors may be used to account for such things as ram drag, gross thrust, spillage, and other similar forces or moments.

4.1.2 A Comparison of Search Hypersonic Force Estimation Methods

Selection of reasonable force calculation methods in hypersonic flow requires a considerable degree of aerodynamic competence. The available hypersonic aerodynamic methods disagree significantly even on relatively simple shapes. This point is discussed in some detail by Gentry as reported in Reference 1. Typical examples taken from the Gentry discussion are presented below.

4.1.2.1 Analysis Method for Pointed Slender Configurations

Figure 4.1-2 presents some typical pressure coefficient variations with impact angle for analysis techniques generally used on pointed slender components. Also presented for comparison purposes is the modified Newtonian theory with $K = 2.4$. This is the limiting value for wedge type flow as proposed by Lees in Reference 8. Figure 4.1-3 presents the same data over a smaller impact angle range. At $M = 20$ the modified Newtonian and the tangent wedge empirical methods compare favorably with the "exact" oblique shock calculations for impact angles from 0 to over 30 degrees.

4.1.2.2 Analysis Method for Blunt Configurations in Expansion Flow

Figure 4.1-4 presents a comparison of various techniques for both pointed and blunt configurations in expansion flow. It should be noted that the VanDyke unified method for expansion flow has been modified such that if a pressure coefficient of less than $-1/M^2$ is calculated for a given expansion angle, the pressure coefficient is set equal to $-1/M^2$. This limiting value of pressure coefficient has been derived from analysis of experimental data (see References 9 and 10).

Blunt body pressure coefficient calculations are also compared in Figure 4.1-5. The pressure coefficient variation with impact angle is plotted in the form Cp/Cp_{STAG} as suggested by Lees in Reference The calculations for Newtonian, Prandtl-Meyer, and OSU empirical techniques utilized stagnation conditions behind a normal shock in an ideal gas.

4.1.2.3 Free Molecular Flow

Comparison of free molecular flow calculations by the program and data presented in Reference 11 are shown in Figure 4.1-6 to 4.1-8. Flat plate lift and drag coefficients are compared in Figure 4.1-6, assuming specular reflection. Figures 4.1-7 and 4.1-8 present the lift and drag of a flat plate for the more realistic diffuse-reflection assumption. Finally, the drag coefficient for a sphere with both specular and diffuse reflection assumptions is shown in Figure 4.1-9.

4.1.3 Discussion of Modified Newtonian Pressure Methods

A brief review of the important features of some of the modified Newtonian pressure calculation methods in the program is presented in the following discussions. This is the most commonly used method in the Gentry program.

Modified Newtonian is used extensively in hypersonic flow analysis due to its ability to give reasonable answers for a great number of shapes with a very simple calculation technique. This capability depends on the use of the variable K as a function of angle of attack as shown in Figure 4.1-10. The modified Newtonian form permits application of tangent wedge (or tangent cone), an empirically defined equation for a given shape, or an effective K for a complete configuration at a given Mach number. The effect of a real gas may be introduced by variation of K for very blunt bodies. In general, the use of modified Newtonian theory may be divided into two groups for discussion purposes: (1) aerodynamically blunt configurations and (2) aerodynamically sharp configurations.

4.1.3.1 Blunt Bodies

On aerodynamically blunt configurations the impact angle of the nose is greater than that for shock detachment, although the leading edge may be sharp and pointed. In true Newtonian flow ($M = \infty$, $\gamma = 1$) the variable K becomes 2. The most commonly used form of modified Newtonian is to input for K the Cp stagnation derived from normal shock relations into the equation

$$C_p = K \sin^2\delta$$

The effects of a real gas may also be approximated in this manner. A comparison of Newtonian and experimental data is presented in References 12 to 14 for blunt body shapes. In general, modified Newtonian ($C_{p_{STAG}} = K$) agrees with data for spheres if the Mach number is greater than 3. The pressure distribution on cylinders is not as good as on spheres. However, for impact angles of 90 degrees to approximately 60 degrees, the agreement is reasonable but deteriorates as zero impact angle is reached. Nevertheless, for preliminary calculations the induced error in C_N and C_A may be acceptable.

Examples of the comparison of modified Newtonian and experiment for spheres and cylinders are shown in Figure 4.1-11. For curved, shock detached bodies with sharp leading edges of either two- or three-dimensional shape, References 15 and 16 show that $C_p = K \sin^2\delta$ should be modified to the form

$$\frac{C_p}{C_{p_{max}}} = \frac{\sin^2\delta}{\sin^2\delta_{max}} .$$

which is sometimes called the generalized Newtonian theory. Comparison with other bodies is shown in Reference 17.

ORIGINAL PAGE IS
OF POOR QUALITY

4.1.3.2 Sharp Bodies

Many approximations exist for sharp pointed bodies. Figures 4.1-6 and 4.1-9 include one form for the sharp wedge developed by Lees in Reference 8 for large Mach numbers.

$$K = (\gamma + 1)$$

Also shown in the limiting form of the cone

$$K = \frac{2(\gamma + 1)(\gamma + 7)}{(\gamma + 3)^2}$$

For large Mach numbers true Newtonian theory, therefore, closely approximates the limiting case for a cone rather than a wedge.

The main disadvantage of Newtonian theory is its inability to predict the flow field, and, for some shapes, this effect can lead to predicted values which may be in serious disagreement with theory. Seiff in References 18 and 19 presents examples of these shapes and a method for obtaining more realistic results from a Newtonian flow concept.

1. Hague, D. S. and Glatt, C. R., Optimal Design Integration of Military Flight Vehicles, ODIN/MFV, Section 4.1 "Hypersonic Arbitrary-Body Aerodynamic Computer Program," AFFDL-TR-72-132, December 1972.

2. Hague, D. S. and Glatt, C. R., Optimal Design Integration of Military Flight Vehicles, ODIN/MFV, Section 3.1, AFFDL-TR-72-132, December 1972.

3. Hague, D. S. and Glatt, C. R., Optimal Design Integration of Military Flight Vehicles, ODIN/MFV, Section 3.2., AFFDL-TR-72-132, December 1972.

4. Powers, S. A., et al., A Numerical Procedure for Determing the Combined Flow Fields over Generalized Three-Dimensional Bodies, Volume I, AFFDL-TR-67-124, December 1967.

5. Powers, S. A., et al., A Numerical Procedure for Determining the Combined Flow Field over Generalized Three-Dimensional Bodies, Volume II, AFFDL-TR-67-124, December 1967.

6. Powers, S. A. and Beeman, E. R., Flow Fields over Sharp Edged Delta Wings with Attached Shocks, NASA CR-1738, April 1971.

7. White, F.M.,Jr., Hypersonic Laminar Viscous Interactions on Inclined Flat Plates, ARS Journal, May 1962.

8. Lees, L., Hypersonic Flow, Fifth International Aeronautical Conference Proceedings, June 1955.

9. McLellan, C. H., Bertram, M. H., Moore, J. A., An Investigation of Four Wings of Square Planform at a Mach Number of 6.86 in the Langley 11-Inch Hypersonic Tunnel, NACA RML51D17, June 1951.

10. Collingbourne, J. R., An Empirical Prediction Method for the Nonlinear Normal Force on Thin Wings at Supersonic Speeds, Aeronautical Research Council CP-662, (Great Britain), 1963.

11. Hayes, W. D., Probstein, R. F., Hypersonic Flow Theory, Volume 5, Academic Press, 1959.

12. Witcofski, R. D., Marcum, D. C., Jr., Effect of Thickness and Sweep Angle on the Longitudinal Aerodynamic Characteristics of Slab Delta Planforms at a Mach Number of 20, NASA TN D-3459, June 1966.

13. Henderson, A., Jr., Braswell, D. O., Charts for Conical and Two-Dimensional Oblique Shock Flow Parameters in Helium at Mach Numbers from About 1 to 100, NASA TN D-819, June 1961.

14. Gregorek, G. M., Nark, T. C., Lee, J. D., An Experimental Investigation of the Surface Pressure and Laminar Boundary Layer on a Blunt Flat Plate in Hypersonic Flow, Volume I, ASD-TDR-62-792, March 1963.

15. Lee, J. D., Pressures on the Blunt Plate Wing at Supersonic and Hypersonic Speeds, FDL-TDR-64-102, July 1964.

16. Epstein, P. S., On the Air Resistance of Projectiles, Proceedings of the National Academy of Sciences, USA, Volume 17, Pages 532-547, 1931.

17. Love, E. S., Generalized Newtonian Theory, Journal of Aerospace Sciences (Readers Forum), Volume 26, No. 5, May 1959, Pages 314-315.

18. Seiff, A., Secondary Flow Fields Embedded in Hypersonic Shock Layers, NASA TN D-1304, May 1962.

19. Seiff, A., Whiting, E. E., Calculation of Flow Fields from Bow Wave Profiles for the Downstream Region of Blunt Nosed Circular Cylinders in Axial Hypersonic Flight, NASA TN D-1147, 1961.

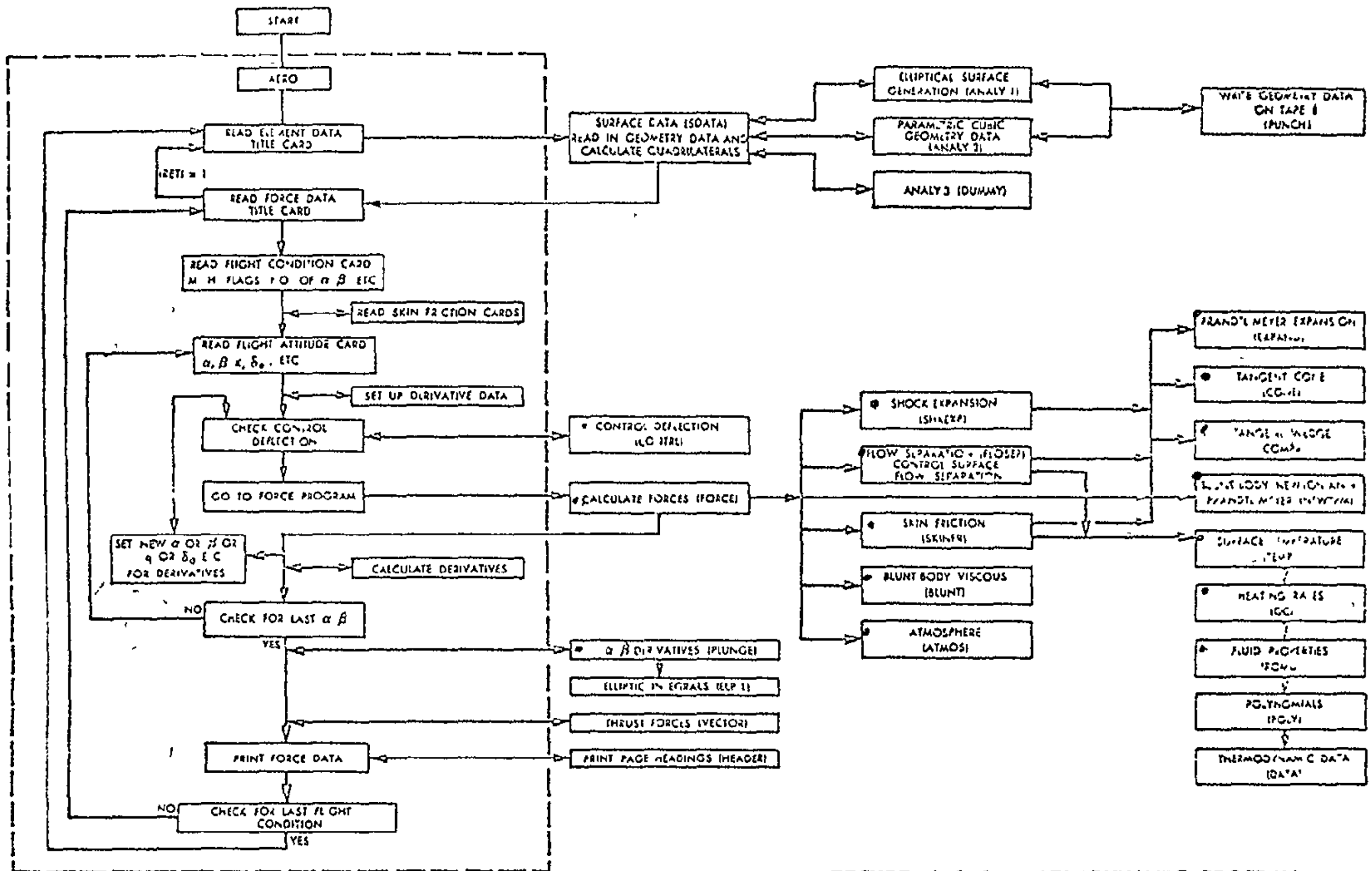


FIGURE 4.1-1. AERODYNAMIC PROGRAM ORGANIZATION

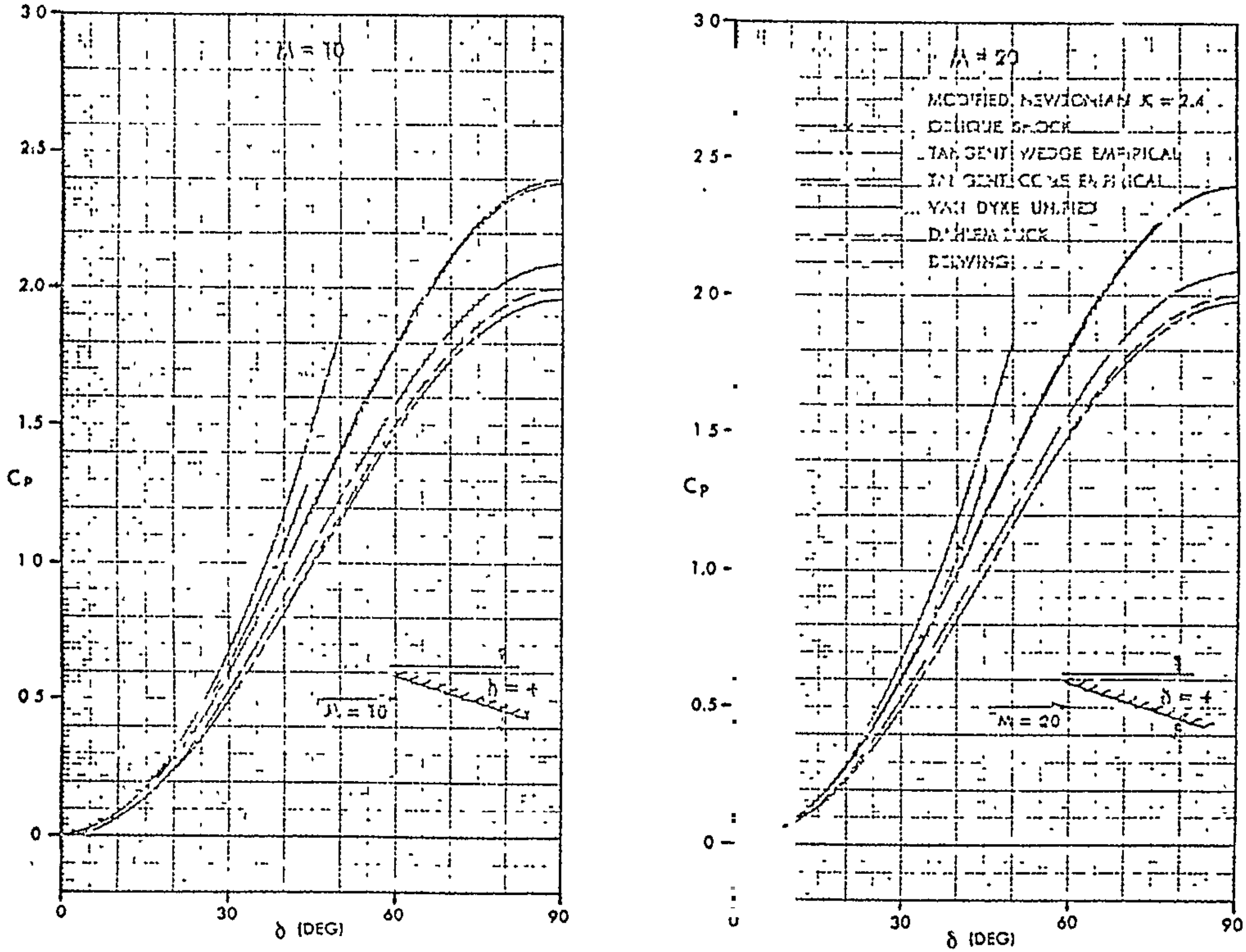


FIGURE 4.1-2. COMPARISON OF PRESSURE PREDICTION METHODS FOR SHARP BODIES IN COMPRESSION TYPE FLOW

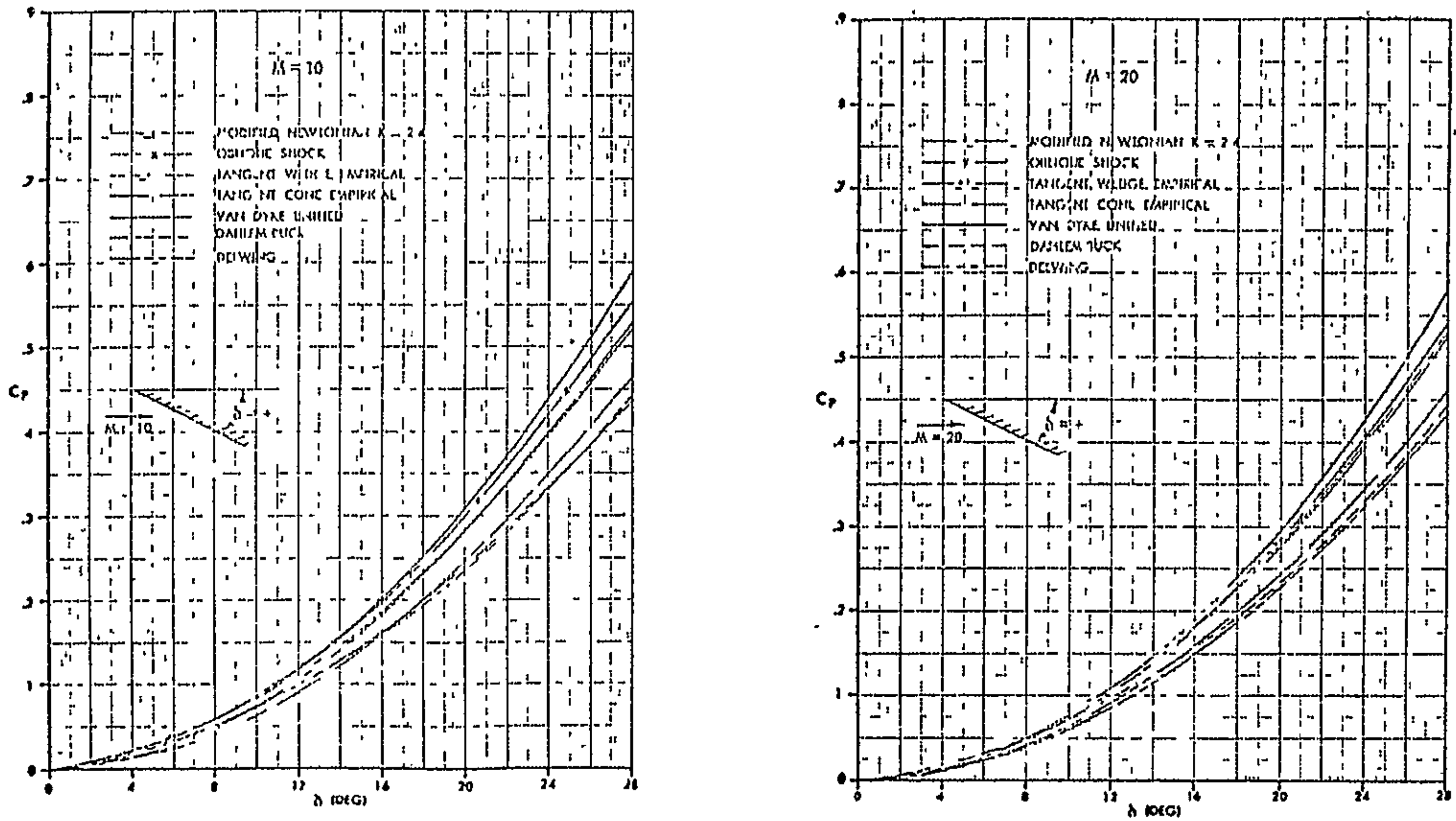


FIGURE 4.1-3. COMPARISON OF PRESSURE PREDICTION METHODS FOR SHARP BODIES IN IN COMPRESSION TYPE FLOW

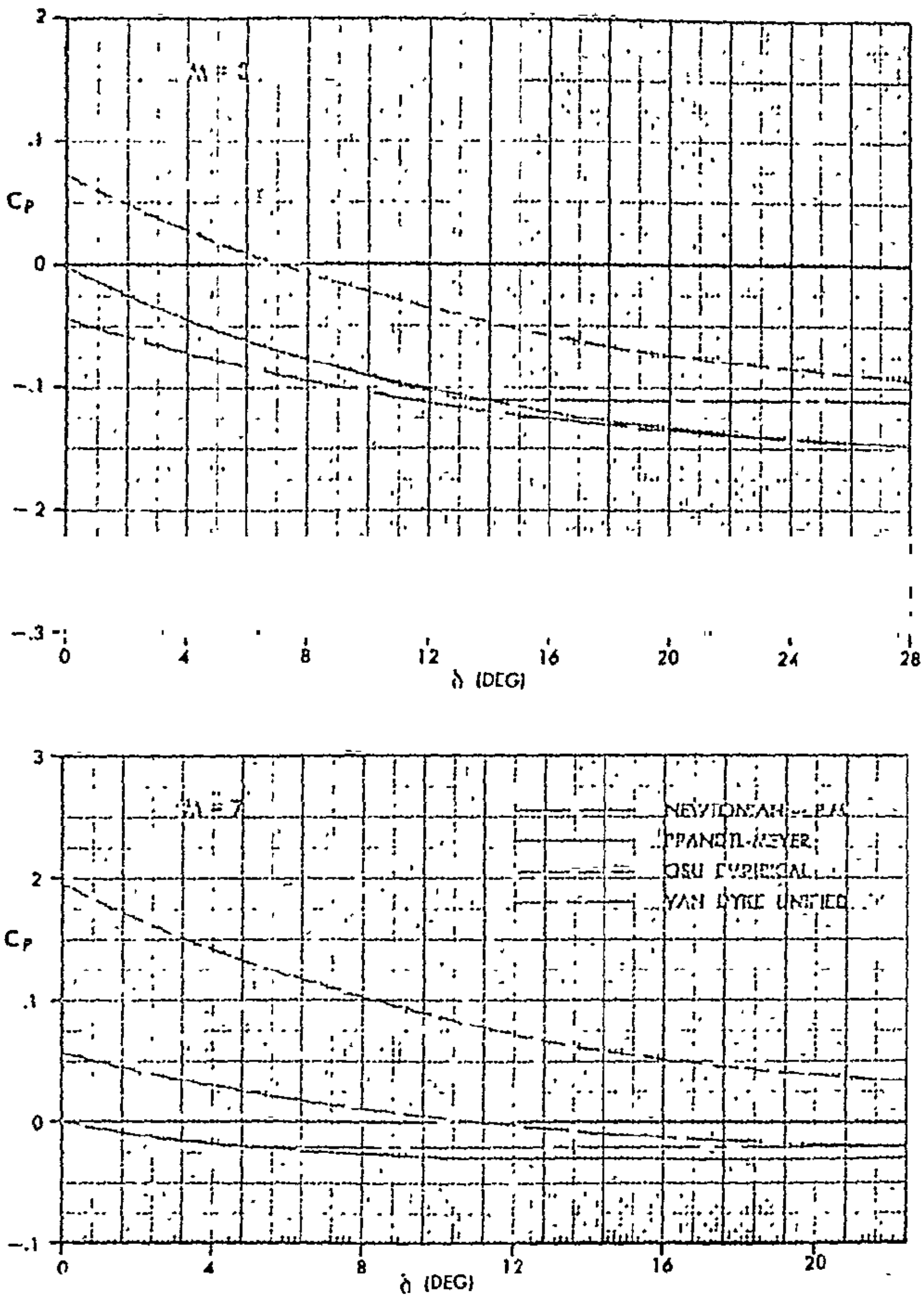


FIGURE 4.1-4. COMPARISON OF PRESSURE PREDICTION TECHNIQUES FOR EXPANSION TYPE FLOW

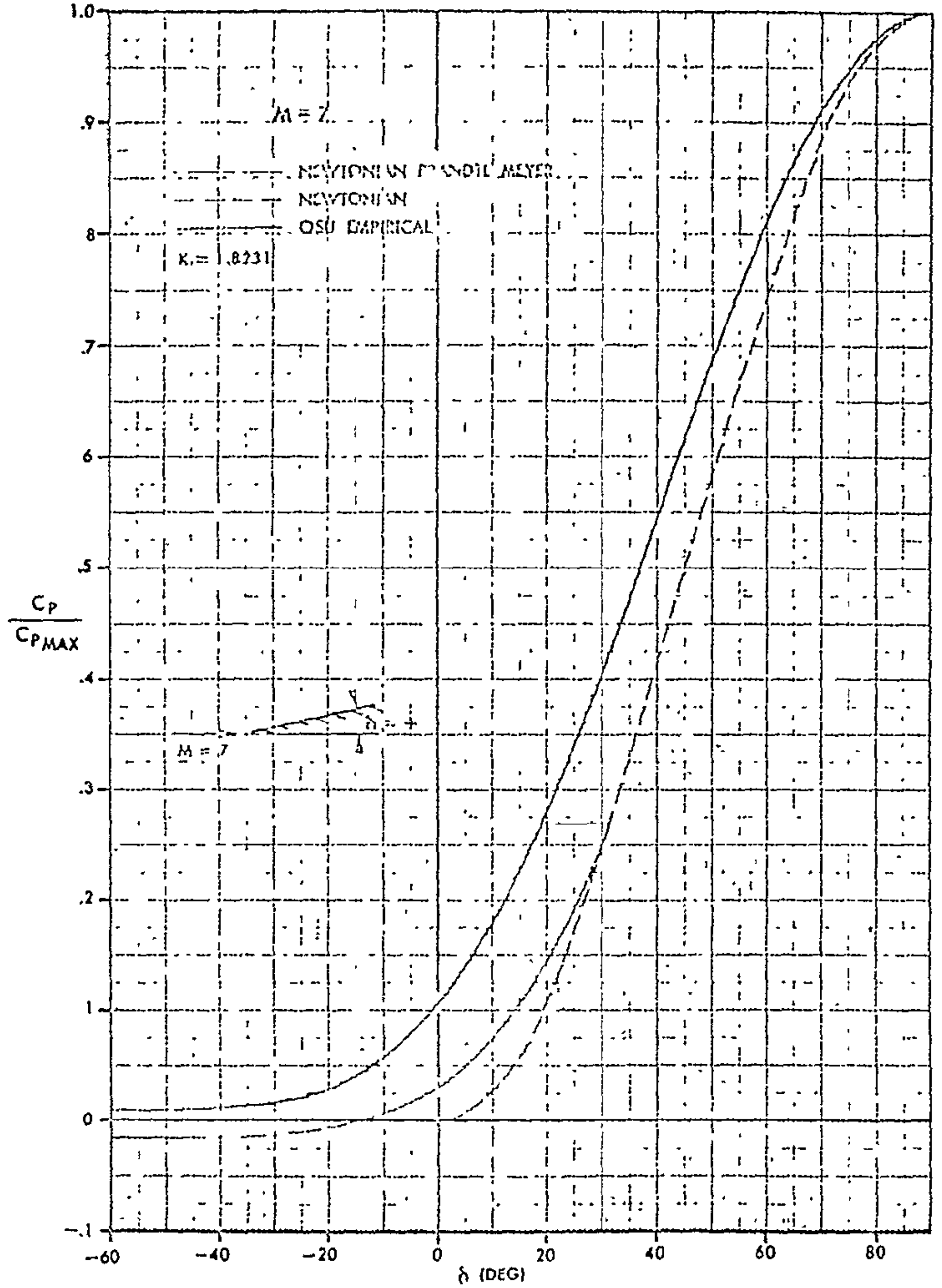


FIGURE 4.1-5. COMPARISON OF BLUNT BODY PRESSURE ESTIMATION TECHNIQUES

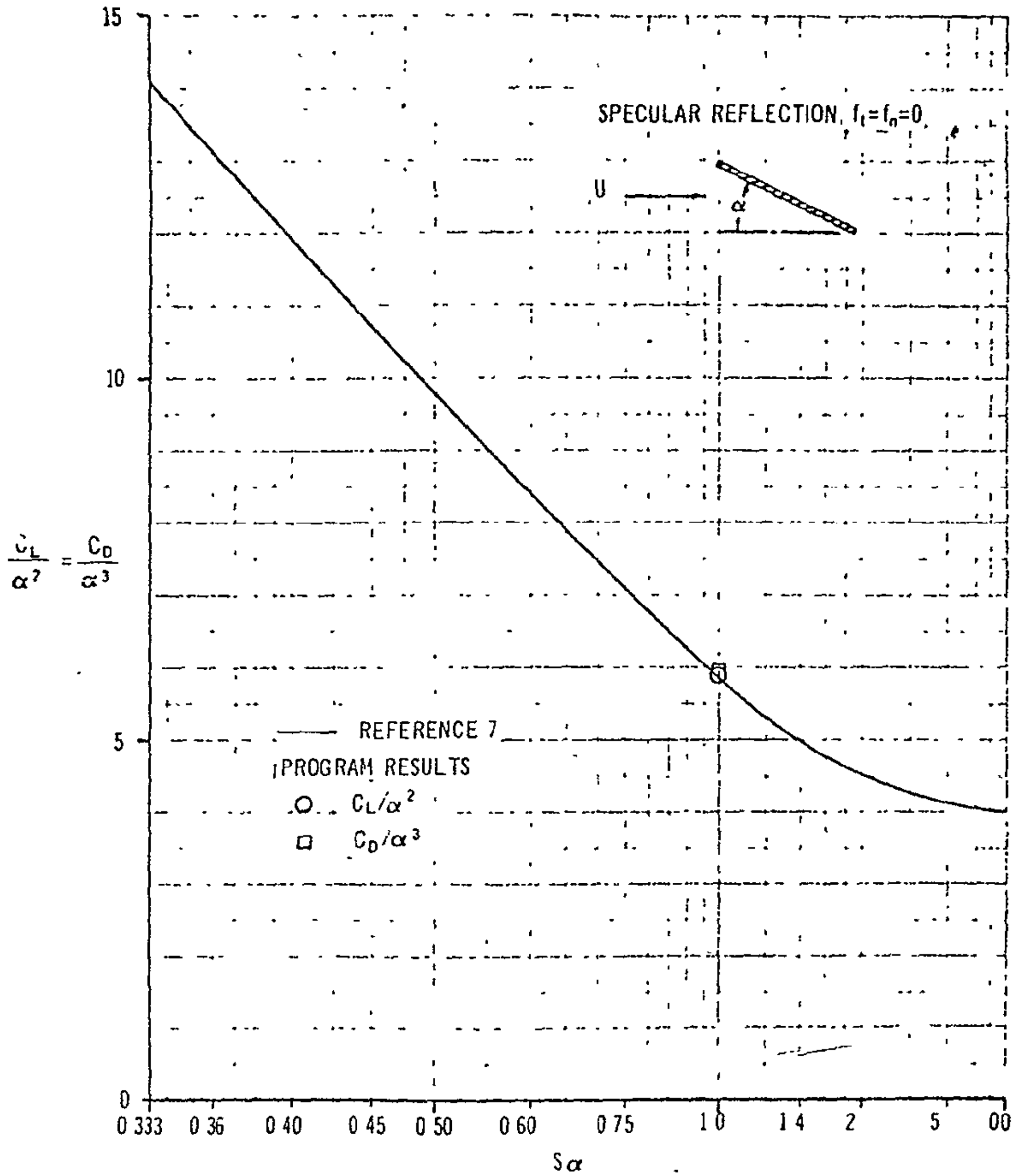


FIGURE 4.1-6. COMPARISON OF FREE MOLECULAR FLOW LIFT AND DRAG FOR A FLAT PLATE WITH SPECULAR REFLECTION

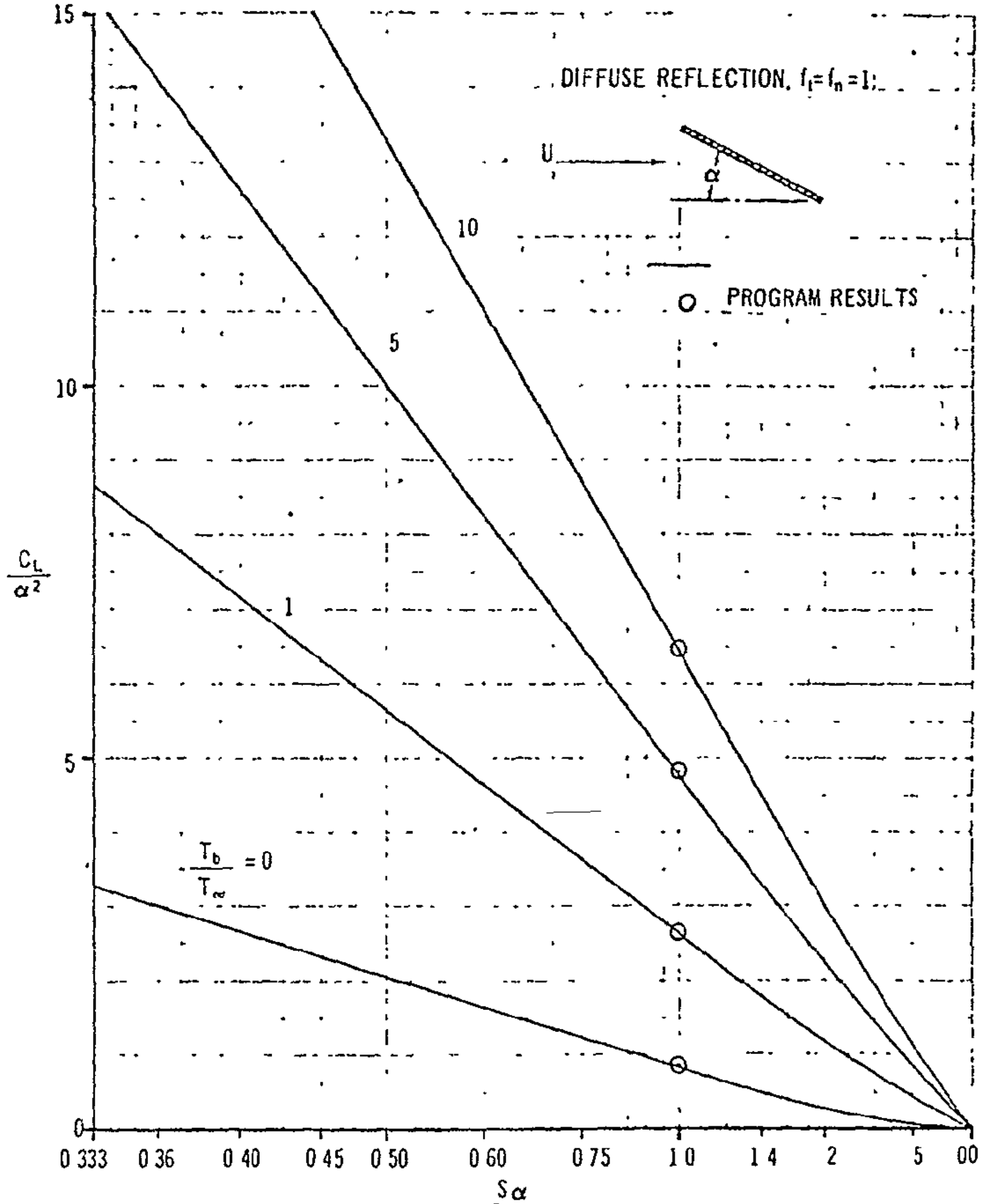


FIGURE 4.1-7. COMPARISON OF FREE MOLECULAR LIFT ON A FLAT PLATE WITH WITH DIFFUSE REFLECTION

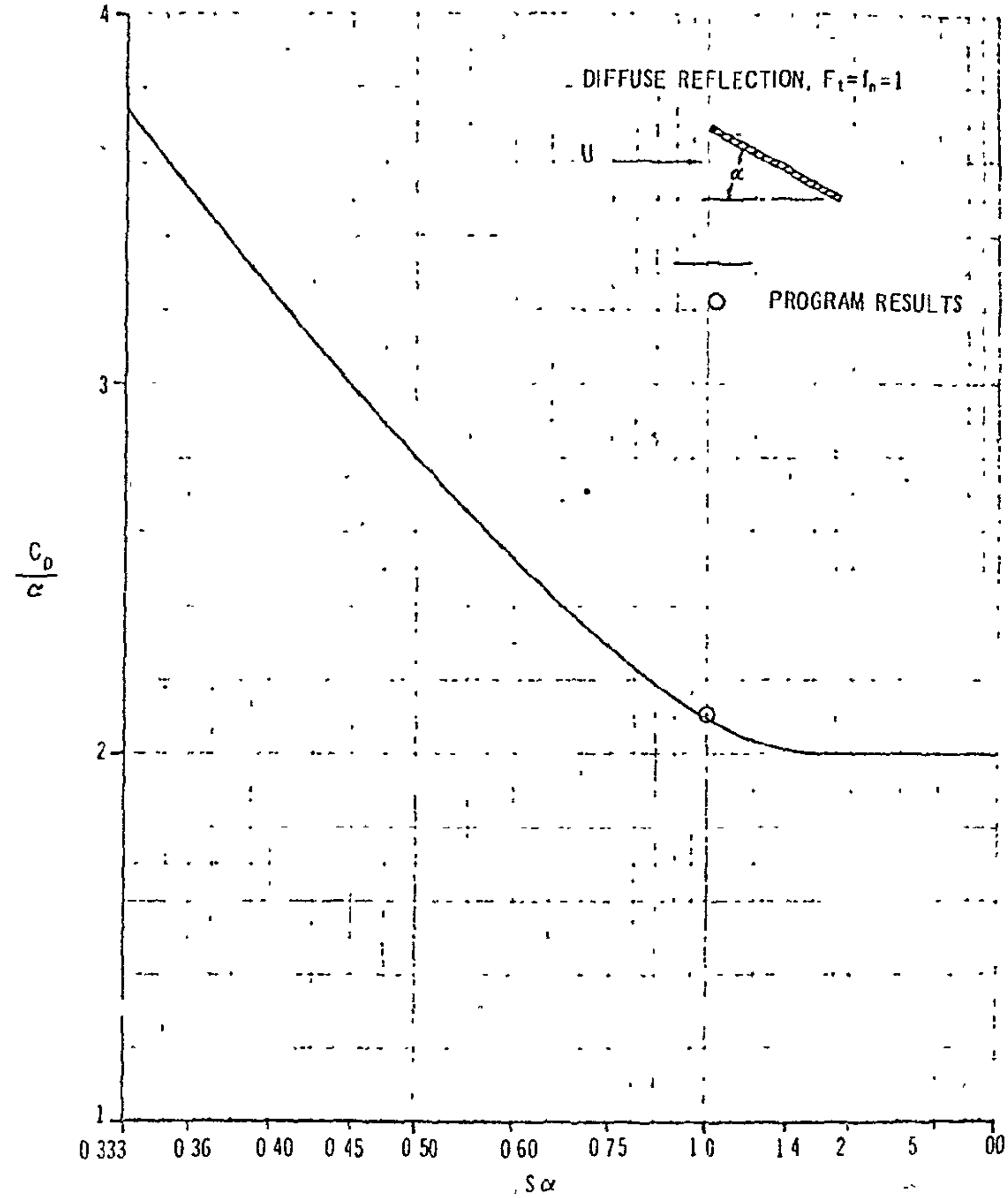


FIGURE 4.1-8. COMPARISON OF FREE MOLECULAR DRAG ON A FLAT PLATE WITH DIFFUSE REFLECTION

C-2

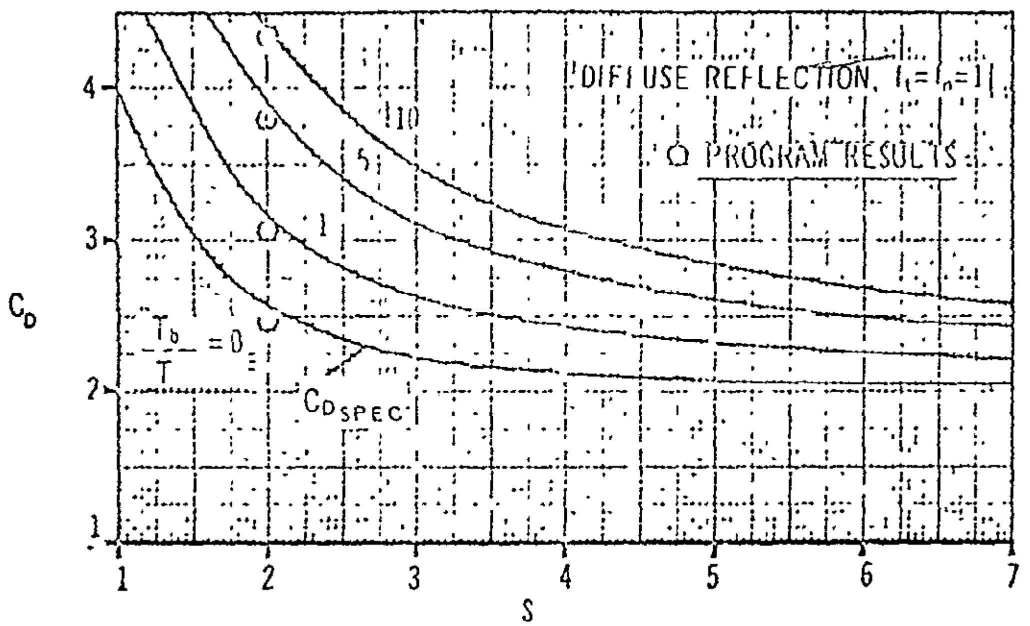


FIGURE 4.1-9. COMPARISON OF FREE MOLECULAR DRAG ON A SPHERE WITH DIFFUSE REFLECTION

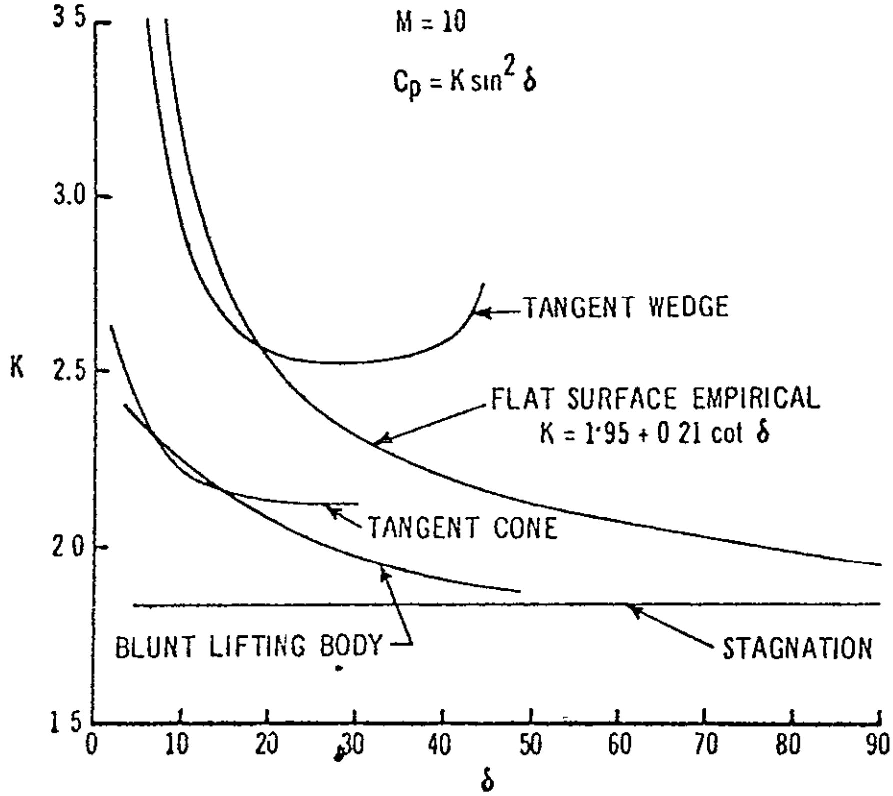
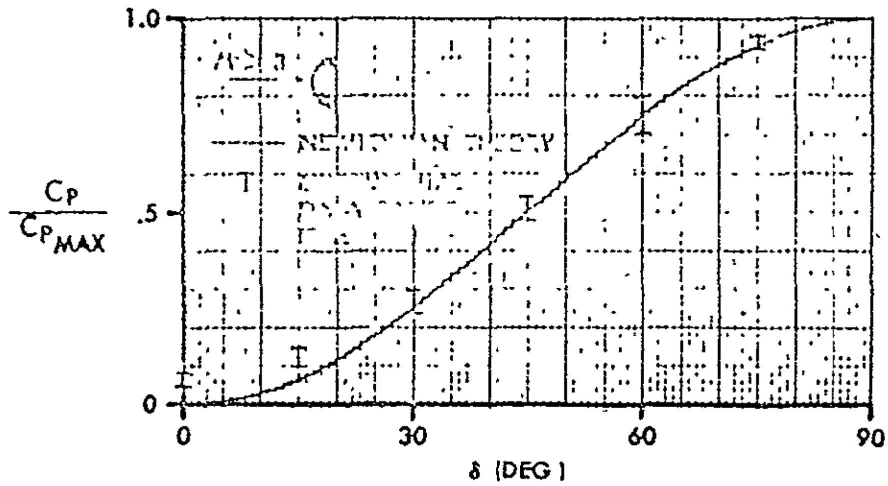


FIGURE 4.1-10. MODIFIED NEWTONIAN CORRELATION FACTORS

SPHERE



CYLINDER

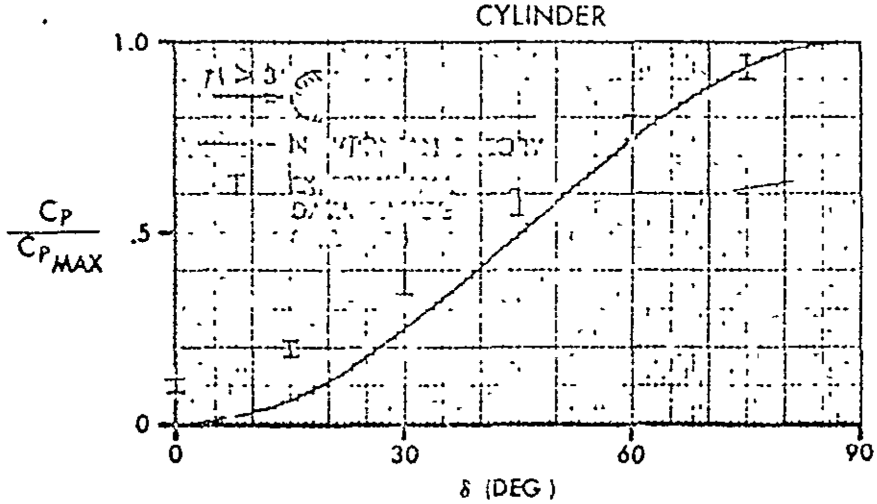


FIGURE 4.1-11. COMPARISON OF EXPERIMENTAL PRESSURES AND MODIFIED NEWTONIAN THEORY FOR SPHERES AND CYLINDERS

TABLE OF CONTENTS FOR SECTION 4.2, PROGRAM ARPII

4.2 PROGRAM ARPII: RAPID SUPERSONIC AREA RULE AERODYNAMIC PROGRAM

The ARPII program is a Fortran version of a supersonic area rule program originally constructed at Avro Aircraft, Reference 1. This program was subsequently updated at McDonnell-Douglas Corporation, St. Louis, Reference 2. Emphasis on the ARPII program lies in obtaining a rapid estimate of the zero lift wave drag component of a vehicle travelling at supersonic speeds. A user's manual for the ARPII program is available, Reference 3.

When considering the design of a supersonic flight vehicle, one of the more significant aerodynamic factors is the configuration drag at zero lift. When the zero lift drag coefficient is plotted against the Mach number for a typical aircraft configuration, a curve similar to that illustrated in Figure 4.2-1 is obtained.

At subsonic speeds any body at zero lift passing through an ideal fluid experiences no net drag force (D'Alambert's Paradox) unless other bodies are also passing through the fluid. If more than one body is passing through the fluid, the net drag force on all the bodies is zero. The individual bodies in the group may have either a thrust or a drag acting on them, but when the thrust and drag are summed over all the bodies, the resulting force will be zero. The drag force which exists on an aircraft flying through a real fluid in subsonic flight must, therefore, have its origin entirely in viscous effects.

At supersonic speeds, the picture changes; in this flight regime a body passing through an ideal fluid creates a system of compression and expansion waves attached to the body. The loss of energy to the wave system causes a drag force to act on a single body even at zero lift. This component force which is known as the wave drag at zero lift is responsible for the supersonic drag rise illustrated in Figure 4.2-1. In supersonic flight, then, the zero lift drag has two components: the viscous drag and the wave drag at zero lift. In order to design an efficient supersonic aircraft an adequate knowledge of both these components is required. The ARPII program presents a rapid method for obtaining the zero lift wave drag component of a wing-body-tail combination at supersonic speeds.

4.2.1 Sonic Area Rule

The method for calculating wave drag outlined in this report is based on the supersonic area rule theory; this theory relates the wave drag of a configuration at zero lift to the development of cross-sectional area of a set of bodies of revolution derived from the basic configuration. The earliest report showing a connection of the above nature seems to be that of Wallace D. Hayes, Reference 4, 1947. This report notes that if the limiting form of the linearized equation for the wave drag of a configuration is taken as $M \rightarrow 1.0$ from above, then the expression

$$\left(\frac{D_W}{q}\right)_{M=1.0} = -\frac{1}{2\Pi}\int_{-L/2}^{L/2}\int_{-L/2}^{L/2} S''(x_1)\ S''(x_2)\ \log\left|x_1-x_2\right| \cdot dx_1\ dx_2 \tag{4.2.1}$$

is obtained. This result is identical with that obtained by Von Karman, Reference 5, for the wave drag of a slender body of revolution at $M = 1.0$. Hayes result was ignored at the time, apparently because of the limitations of linearized theory in the transonic range.

In 1952 Richard T. Whitcomb's report, Reference 6, experimentally established the connection between the transonic drag rise of low aspect ratio thin wings mounted centrally on reasonably slender bodies and that of the body of revolution having the same distribution of cross-sectional area. To illustrate this point, Whitcomb's results for the four basic configurations are shown in Figure 4.2-2. Whitcomb's ideas appear to be based on the phenomenon of stream tube choking at transonic speeds. The invariance of stream tube cross-sectional area to small velocity changes about $M = 1.0$ means that the flux of fluid out of a radius greater than the wing semispan, described on a plane normal to the longitudinal axis, must be the same for both the wing-body combination or the body of revolution having the same distribution of cross-sectional area. This is illustrated in Figure 4.2-3; for if the plane element has thickness δ_x, then the flux out of this disc in both cases will be

$$\delta Q = \frac{dS}{dx} \cdot \delta x \tag{4.2.2}$$

Whitcomb argues that the flow field is such that any radial or circumferential deviations in the disturbances caused by the wing-body combination are rapidly reduced causing the field to tend towards the radially symmetric disturbance produced by a body of revolution. Examination of the shock patterns about a configuration and its equivalent body of revolution provide support for this view of the similarity in the two disturbance fields and, hence, to the similarity in their drag rise characteristics. Whitcomb further reasoned that if the drag of a wing-body combination is similar to that of its equivalent body of revolution, then by indenting the body to account for the cross-sectional area of the wing, the wave drag could be made to approach that of the body alone. The reductions in drag that he obtained in this manner are reproduced from Reference 6 in Figure 4.2-4. In all three cases these tests reveal a considerable improvement in the drag rise characteristics; although, in general, the equivalent body of revolution had a lower drag rise than the indented wing-body combination.

4.2.2 Supersonic Area Rule

The success of Whitcomb's sonic area rule theory in providing a guide to estimating and a means of reducing the zero lift wave drag of a wing-body combination lead to a search for a similar method at supersonic speeds. A method for obtaining the supersonic zero lift wave drag of wings alone in terms of a set of area distributions derived from the basic wing distribution had already been given by Heaslet, Lomax, and Spreiter, Reference 7. In 1953 both R. T. Jones, Reference 8, and Richard T. Whitcomb with T. L. Fischetti, Reference 9, produced reports showing that the supersonic zero lift wave drag of a wing-body combination could be estimated in a similar manner. These methods find the drag of the combinations *as the average drag of a series of equivalent bodies of revolution* constructed in the following way.

With the aircraft rolled through an angle θ (to be definite, let the port wing be raised for positive θ), construct a set of planes which are normal to the horizontal plane from which the roll angle is measured and inclined at the Mach angle μ to the aircraft longitudinal axis. These Mach planes will intersect the wing-body combination and each one in so doing will define an inclined cross-sectional area. The projection of these areas on the yz plane (i.e., the frontal areas of the cross-sections) are used to define the cross-sectional area distribution of the equivalent body of revolution for the particular roll angle θ. This arrangement of planes together with the coordinate system is shown in Figure 4.2-5. The drag of each individual equivalent body of revolution may then be found from Von Karman's formula and the mean taken to find the drag of the wing-body combination

$$\therefore \quad \frac{D_W}{q} = -\frac{1}{4\Pi^2} \int_0^{2\Pi} d\theta \int_{-L/2}^{L/2} \int_{-L/2}^{L/2} S''(x_1, \theta, M)\, S''(x_2, \theta, M)$$

$$\times \operatorname{Log}|x_1 - l_2|\, dx_1 dx_2 \tag{4.2.3}$$

4.2.3 Transfer Rule

In Reference 10, G. N. Ward approaches the problem of the wave drag of wing-body combinations by considering the drag of general distributions of sources in space. He then shows that thin wings and slender bodies, at zero lift, either alone or in combination with each other, can be represented by distributions of sources in the surface of the body with surface density proportional to the local slope of the surface in the x direction. The drag in this particular case reduces to

$$\frac{D_w}{q} = -\frac{1}{2\pi}\iint_{wing} T''(\bar{R}_1)\,T(\bar{R}_2)\cosh^{-1}\left\{\frac{|x_1 - x_2|}{\beta|\bar{r}_1 - \bar{r}_2|}\right\} dS_1\,dS_2$$
$$-\frac{1}{2\pi}\int_{-L/2}^{L/2}\int_{-L/2}^{L/2} A''(x_1)\,A''(x_2)\log|x_1 - x_2|\,dx_1\,dx_2$$
$$-\frac{1}{2\pi}\int_{-L/2}^{L/2}\int_{-L/2}^{L/2}\left[S_F''(x_1) + A''(x_1)\right]\left[S_F''(x_2) + A''(x_2)\right]$$
$$\times \log|x_1 - x_2|\,dx_1\,dx_2 \qquad (4.2.4)$$

where

$$A(x) = \frac{1}{\pi}\int_{\Delta S_w}\frac{T(\bar{R}_1)\,dS_1}{\sqrt{\beta^2 r_1^2 - (x - x_1)^2}} = \frac{1}{\pi}\int_{\Delta S_w}\frac{dV}{\sqrt{\beta^2 r_1^2 - (x - x_1)^2}} \qquad (4.2.5)$$

is the wing transferred area and dS_1, dS_2 = wing elements of area at the points (x_1, y_1, z_1), $x_2, y_2, z_2)$.

$\bar{R}_1 \bar{R}_2$ = the vectors with components (x_1, y_1, z_1), (x_2, y_2, z_2)

$T(R_1)$, $T(R_2)$ = wing thickness at the points $\bar{R}_1$, $\bar{R}_2$

$\bar{r}_1$, $\bar{r}_2$ = the vectors with components $(0, y_1, z_1)$, $(0, y_2, z_2)$

ΔS_w = that portion of the wing in the zone of silence for the point $(x, 0, 0)$, see Figure 4.2-6.

The first term is the drag of the exposed wing panels alone; the second term is the drag of a body of revolution having an area distribution equal to the fuselage area combined with the wing transferred area. The last term is the drag of a body of revolution having an area distribution equal to the transferred area of the wing alone

$$D_{wing-only} = D_{wing} - D\left\{A\right\} - D\left\{S_F + A\right\} \qquad (4.2.6)$$

This relationship will be of note in a later section where the problem of area ruling the fuselage is considered.

The transfer rule is of interest when the optimum fuselage for a given wing is required. It can be shown, Reference 1, for example, that the transferred wing area is the mean wing area versus roll angle determined by the supersonic area rule Mach planes. Hence,the optimum fuselage for a given wing is obtained when the fuselage and mean wing at the Mach number or series of Mach numbers of interest has a minimum drag.

4.2.4 Other Methods

It should be noted that several other methods for computation of wing-body wave drag have been proposed, notably Baldwin and Dickey's moment of area rule, Reference 11, and Faget's method of hoops. The ARPII program, however, is limited to the supersonic area rule method.

4.2.5 Wave Drag in the General Case

The theories of Sections 4.2.2 to 4.2.4 apply to slender bodies and thin wings or their combination provided that no lift is carried over any portion of the planform. There are many configurations which may reasonably be analyzed by linearized theory and yet do not fall into the above class, for example, a cambered thin wing mounted centrally on a slender body. In Reference 12, a generalized area rule correct to the limits of linearized theory for combinations of wings and bodies is obtained; this is

$$\frac{D}{q} = -\frac{1}{4\pi^2}\int_0^{2\pi} d\theta \int_{l_1(\theta)}^{l_2(\theta)} dx_1 \int_{l_1(\theta)}^{l_2(\theta)} \left[S''(x_1,\theta) - \beta\frac{\partial l}{\partial x}(x_1,\theta)\right]$$

$$\times\left[S''(x_2,\theta) - \beta\frac{\partial l}{\partial x}(x_2,\theta)\right]\log|x_1 - x_2|\,dx_2 \qquad (4.2.7)$$

ORIGINAL PAGE IS
OF POOR QUALITY

where $\ell(x, \theta)$ is the component of the force acting on the oblique section resolved in a plane normal to the free stream and resolved again in the θ direction, Figure 4.2-7. It may be noted that at M = 1.0, Equation 4.2.7 reduces to 4.2.1 and the sonic wave drag due to lift vanishes.

In order to use Equation 4.2.7 the distribution of force over the configuration is needed. If this is known the drag can, of course, be found by integration of the force and slopes over the configuration. The ARPII program does not attempt to consider the effects of lift in any way.

4.2.6 Evaluation of the Wave Drag Integral

4.2.6.1 Fourier Series Method

A problem common to the sonic area rule, supersonic area rule and transfer rule is the evaluation of the integral which expresses the wave drag of a body of revolution in terms of its area distribution. This can be written

$$\frac{D}{q} = -\frac{1}{2\pi L^2} \int_0^1 \int_0^1 S''(\xi_1)\, S''(\xi_2) \left|\log\, \xi_1 - \xi_2\right| d\xi_1\, d\xi_2 \qquad (4.2.8)$$

Several methods for evaluation of this integral have been suggested; the earliest method appears to be that of Sears, Reference 13. In Sears' method the transformation

$$x = \frac{1}{2}(1 - \cos\theta) \qquad (4.2.9)$$

is made.

The slope of area dS/dx is now approximated by a Fourier sine series

$$S'(x) = \sum_{n=1}^{\infty} A_n \sin n\theta \qquad (4.2.10)$$

so that

$$A_n = \frac{2}{\pi} \int_0^{\pi} S'(x) \sin \cdot n\theta\, d\theta \qquad (4.2.11)$$

With these assumptions the drag becomes

$$\frac{D}{q} = \frac{\pi}{4} \sum_{n=1} n \cdot A_n^2 \qquad (4.2.12)$$

corresponding to an area distribution of

$$S(x) = S(o) + \frac{A_1}{4}\theta + \frac{1}{4}\sum_{n=1}^{\infty} \frac{1}{n}\left(A_{n+1} - A_{n-1}\right)\sin n\theta \tag{4.2.13}$$

Evaluation of the drag by this method is a tedious process, mainly because of the difficulty of obtaining the slope of area curve from the actual area distribution.

4.2.6.2 Eminton's Method

A different approach is suggested by Eminton, Reference 14. Defining

$$S(o),\ S(1)\ \text{and}\ S(\xi_i) \quad i = 1, 2, \ldots, N \tag{4.2.14}$$

Then, if the drag given Equation 4.2.12 is minimized for the area distribution of Equation 4.2.13 subject to the restraints of 4.2.14, Eminton shows by the method of Lagrangean multipliers the minimum drag is given by

$$\frac{D}{q} = \frac{1}{L^2}\left[\frac{4}{\pi}\left\{S(1) - S(o)\right\}^2 + \pi\sum_{i=1}^{N}\sum_{j=1}^{N} c_i c_j f_{ij}\right] \tag{4.2.15}$$

or in matrix form

$$\frac{D}{q} = \frac{1}{L^2}\left[\frac{4}{\pi}\left(S(1) - S(o)\right)^2 + \pi\lfloor c\rfloor[f]\{c\}\right] \tag{4.2.16}$$

where

$$c_i = S(\xi_i) - S(o) - \left(S(1) - S(o)\right)u_i \tag{4.2.17}$$

$$u_i = \frac{1}{\pi}\left[\cos^{-1}(1-2\xi_i) - 2(1-2\xi_i)\sqrt{\xi_i(1-\xi_i)}\right] \tag{4.2.18}$$

$$[f] = [p]^{-1} \tag{4.2.19}$$

ORIGINAL PAGE IS
OF POOR QUALITY

and

$$p_{ij} = -\frac{1}{2}(\xi_i - \xi_j)^2 \log_e \frac{\xi_i + \xi_j - 2\xi_i\xi_j + 2\sqrt{\xi_i\xi_j(1-\xi_i)(1-\xi_j)}}{\xi_i + \xi_j - 2\xi_i\xi_j - 2\sqrt{\xi_i\xi_j(1-\xi_i)(1-\xi_j)}} \tag{4.2.20}$$

$$+ 2(\xi_i + \xi_j - 2\xi_i\xi_j)\sqrt{\xi_i\xi_j(1-\xi_i)(1-\xi_j)}$$

The corresponding area distribution is given by

$$S(x) = S_o + (S(1) - S(o)) \cdot u(x) + \lfloor P_{xj} \rfloor [f_{ij}] \{c_j\}$$

4.2.6.3 Other Methods

Two other methods of evaluating the wave drag integral have been suggested: that of Cahn and Olstad, reference 15, and that of Holdaway and Mersman, reference 16. The method of Cahn and Olstad uses a numerical technique for evaluating the integral and requires a knowledge of the second derivative of the area distribution.

The remaining method, that of Holdaway and Mersman, also uses a numerical technique, this time in the form of Tchebichef polynomials. By this device it is possible to evaluate the Fourier coefficients of Equation (4.2.1) by working with the area distribution rather than one of its derivatives, a feature common to Eminton's method and the Tchebichef polynomial method. However, as noted in Section 4.2.6.2 the ARP II program uses the N station Eminton method with N=19 being the recommended number of areas as in Eminton's original report.

4.2.7 Applying the Supersonic Area Rule on a Digital Computer

4.2.7.1 Outline

In order to use the supersonic area rule as a preliminary design tool, a rapid method of obtaining the area distributions required by the theory must be employed. For the purpose of determining the wing contribution to such a distribution, it is sufficient to note that to date most aircraft wing surfaces have been generated by a set or sets of straight generator lines. Once the equations of these lines and the equations of the Mach planes for a given Mach number and roll angle are known, it is a straightforward exercise in analytic geometry to find the points at which a particular Mach plane will intersect the wing generator lines and, hence, by integration the wing area defined by that plane. Repeating this for each plane will define the wing contribution to the area distribution for the particular value of Mach number and roll angle. To find the contribution of tanks or the fuselage, sufficient accuracy should be obtained if the point at which a Mach plane intersects their center of area locus is found and the normal cross-sectional area at that point is taken. Once an area distribution has been found, the drag of its equivalent body of revolution must be calculated. In the ARP II program, the method of Eminton is used.

4.2.7.2 Mach Plane Equations

The general equation of a plane is of the form

$$Ax + By + Cz + D = 0 \qquad (4.2.21)$$

It can be shown that the equation of a Mach plane is

$$x - \cot\mu \cos\theta \, y - \cot\mu \sin\theta \, z - x' = 0 \qquad (4.2.22)$$

where

μ - Mach angle
θ - roll angle
x' - plane intercepts on the x axis

4.2.7.3 Wing Generator Lines

In designing an aircraft wing it is customary to specify the section to be used at various spanwise locations, on an aerodynamic basis. When this wing is layed out in the design office the wing surface between any two sections is described by straight lines passing through corresponding percentage chord points on the section profiles. Any wing surface formed in this manner is therefore described by one or more sets of straight generator lines. The equations of these lines is of the form

$$x_i = a_{1_i} y + b_{1_i} \qquad (4.2.23)$$

$$z_i = a_{2_i} y + b_{2_i} \quad \text{where } i = 1, 2, \ldots N \text{ say} \qquad (4.2.24)$$

To find the equation of a generator line passing between any two sections, one need only know the coordinates of its end points. Let the inboard such end point have coordinates x_1, y_1, z_1 and the outboard end point coordinates x_2, y_2, z_2, then

$$a_1 = \frac{x_2 - x_1}{y_2 - y_1} \qquad (4.2.25)$$

$$b_1 = x_1 - a_1 y_1 \qquad (4.2.26)$$

$$a_2 = \frac{z_2 - z_1}{y_2 - y_1} \qquad (4.2.27)$$

$$b_2 = z_1 - a_2 y_1 \qquad (4.2.28)$$

When specifying the generator lines, it may be necessary only to obtain the equation for one side of the wing if a plane of symmetry is present. Similarly, if the wing under consideration has a symmetric section only the equations for the upper surface need be stored in the computer.

4.2.7.4 Wing Contribution to Area Distribution

The first step in finding the wing area defined by a Mach plane is to find the points at which the i^{th} Mach plane intercepts the starboard upper wing generator lines; i.e., where the plane

$$Ax + By + Cz + D_j = 0 \tag{4.2.29}$$

intersects the generator lines

$$x_i = a1_i y + b1_i \tag{4.2.30}$$

$$z_i = a2_i y + b2_i \tag{4.2.31}$$

Substituting the generator equations into the Mach plane equations and solving for the value of the y intercept coordinate

$$\bar{y}_{ij} = \frac{Ab1_i + Cb2_i + D_j}{Aa1_i + B + Ca2_i} \tag{4.2.32}$$

The z coordinate of the intercept is

$$\bar{z}_{ij} = a2_i \bar{y}_{ij} + b2_i \tag{4.2.33}$$

The required area, that is the frontal area of the wing section defined by the Mach plane in passing through the wing is

$$\Delta S_j = \int \bar{z}_{ij}\, d\bar{y}_{ij} \tag{4.2.34}$$

where the integration extends between specified limits.

4.2.7.5 Fuselage and Tank Contribution to the Area Distribution

To find the fuselage contribution to the area distribution, the approximation is made that the frontal area of the fuselage section intercepted by a Mach plane is equal to the normal cross-sectional area at the point where the Mach plane intercepts the fuselage center of area locus.

It can be seen from Figure 4.2.8 that for reasonably slender bodies that the approximation should be reasonable, the area being underestimated on one side of the axis and overestimated on the other. On purely theoretical grounds, it might be concluded that there are no grounds for using the true slant area through the fuselage in any case. However, in Reference 17 the wave drag of bodies which were not so slender was calculated using both the normal area distribution and the frontal projection of the true slant area. It was concluded that greater accuracy is obtained when the slant area method is used. The above approach is equally applicable to external pods mounted on the aircraft. In the case of a fuselage, the x axis is usually placed at or near the locus of the fuselage center of area so that the Mach plane intercept x^1 can be used directly to obtain the point at which the Mach plane intercepts the fuselage distribution. For a tank or pod this is no longer true and the point at which the Mach plane intercepts the tank locus of area (assumed to be represented with sufficient accuracy by a line parallel to the x axis) must be found. Let the y and z coordinates of the tank centroid of area be y_T, z_Y then substituting these values into the Mach plane equation gives the required intercept, i.e.,

$$x_T = x' - (By_T + Cz_T) \qquad (4.2.35)$$

and the normal fuselage area at this point must be used for the tank or pod area contribution.

4.2.8 Configuration Definition by Supersonic Area Rule

Configuration definition involves a balance between internal and external configuration requirements. For example, when laying out a supersonic fighter the forebody geometry involves a trade between aerodynamic drag and radar dish size. Aft of this the fuselage dimensions are determined by a trade between aerodynamic drag and crew mobility and vision constraints. Further aft again the front and rear face of the engines with clearance for other system components at these points tends to size the fuselage cross-section.

The fighter wing t/c is determined by a trade between aerodynamic drag, structural depth, and fuel requirements. The wing planform is determined by a trade between aerodynamic and structural efficiency. Placement of the wing on the body involves a trade between aerodynamic stability and control and supersonic area rule considerations. Supersonic area rule considerations will also tend to govern the longitudinal distribution of fuselage area between the radar dish, crew station, forward engine face, and aft engine face.

A typical configuration layout according to supersonic area rule principles is presented in Figures 4.2-9(a) and (b). These figures present non-area ruled aircraft and a similar design area ruled for Mach 1.4. Figures 4.2-10 (a) to (e) present successively the area distribution for the basic aircraft and M = 1.0, 1.2, 1.4, and the Mach number range 1.0 to 1.4. To obtain these distributions three steps were followed:

First, the minimum fuselage cross-sectional areas at five control points are determined on the basis of crew and subsystem clearances.

Second, the selected wing mean area distribution, averaged for all roll angles at the selected area rule Mach number, is obtained from the supersonic area rule program. In a

ORIGINAL PAGE IS
OF POOR QUALITY

more sophisticated design study the mean wing area may be averaged against both roll angle and Mach number to insure low zero lift wave drag over a specified Mach number range. Example designs of this type are presented in References 1 and 17.

Third, the wing *mean* area is added to the fuselage cross-sectional area obtained from normal cuts to obtain the combined fuselage/mean wing distribution at selected control points. By Ward's transfer of area rule, the sum of the fuselage and mean wing area distribution at the selected Mach number should be a minimum wave drag shape. Minimum wave drag shapes having any number of specified area constraints are given by Eminton in Reference 14. These shapes are available in the supersonic area rule program. The difference between the optimum shape and the mean wing gives the required fuselage area distributions in Figure 4.2-9. The resulting fuselage area distribution is then checked against the minimum required area all along the longitudinal body axis to verify internal clearances. This process may reveal the necessity for additional constraints on the combined fuselage/mean wing distribution. In this case, the fuselage shaping process is repeated again with an additional constraint. The second iteration is usually sufficient to develop a satisfactory fuselage.

Some typical comparisons between area rule indented bodies, wave drag, and wave drag calculations are presented in Figures 4.2-11(a) and (b). The test results and the Tchebichef calculations are taken from Reference 19.

REFERENCES:

1. Hague, Donald S., Estimation of Wing-Body Wave Drag, GEN/THEO AERO/10, Part 1, Avro Aircraft, Ltd., July 1961. (available from Aerophysics Research)

2. Hague, D. S., Engineering Instructions for Use of Supersonic Area Rule Program, Avro Aircraft (Canada) Report P470/THEOR.AERO/7., March 1961.

3. Galipeau, John, User's Manual for Program ARPII, Aerophysics Research Corporation TN-176, 1972.

4. Hayes, Wallace D., Linearized Supersonic Flow, Report No. AL-222, North American Aviation, Inc., June 1947.

5. Von Karman, Th., The Problem of Resistance in Compressible Fluids, Estratto Dugli Atti del V Convegno della Fondazione Allesando Volta, 1935, Rome, Reale A-cademia d'Italia, 1936.

6. Whitcomb, Richard T., A Study of the Drag Rise Characteristics of Wing-Body Combinations near the Speed of Sound, NACA RML 52 H 08, September 1952. (superseded by NACA Rep. 1273, 1956).

7. Heaslet, Max. A, Lomax, Harvard, and Spreiter, John R., Linearized Compressible Flow Theory for Sonic Flight Speeds, NACA Rep. 956, 1950.

8. Jones, Robert T., Theory of Wing-Body Drag at Supersonic Speeds, NACA RM A 53H18a, September 1953. (superseded by NACA Rep. 1284, 1956).

9. Whitcomb, Richard T. and Fischetti, Thomas L., Development of a Supersonic Area Rule and an Application to the Design of a Wing-Body Combination Having High Lift to Drag Ratios. ——

10. Ward, G. N., The Drag of Source Distributions in Linearized Supersonic Flow, College of Aeronautics, Rep. No. 88, February 1955.

11. Baldwin, Barrett S., and Dickey, Robert R., Application of Wing-Body Theory to Drag Reduction at Low Supersonic Speeds, NACA RM A 54 J 19, January 28, 1955.

12. Lomax, Harvard, The Wave Drag of Arbitrary Configurations in Linearized Flow as Determined by Areas and Forces in Oblique Planes, NACA RMA 55 A18, 1955.

13. Sears, W. R., On Projectiles of Minimum Wave Drag, Quarterly of Applied Mathematics, Volume IV, No. 4, January 1947.

14. Eminton, E., On the Minimization and Numerical Evaluation of Wave Drag, R.A.E. Report Aero 2564, November 1955.

15. Cahn, Maurice S. and Olstad, Walter B., A Numerical Method for Evaluating Wave Drag, NACA TN 4258, June 1958.

16. Holdaway, George H. and Mersman, William A., Application of the Tchebichef Form of Harmonic Analysis to the Calculation of Zero Lift Wave Drag of Wing-Body-Tail Combinations, NACA RMA 55 J28, February 1956.

17. Nelson, Robert L. and Welsh, Clement J., Some Examples of the Applications of the Transonic and Supersonic Area Rules to the Prediction of Wave Drag, NASA TN D-446, September 1960.

18. Loving, Donald L., A Transonic Investigation of Changing Indentation Design Mach Number on the Aerodynamic Characteristics of a 45° Swept Back Wing-Body Combination Designed for High Performance, NACA RM L55 J07, January 1956.

19. Holdaway, George H. and Hatfield, Elaine, W., Investigation of Symmetrical Body Indentations Designed to Reduce the Transonic Zero Lift Wave Drag of a 45° Swept Wing with an NACA 64A006 Section and with a Thickened Leading Edge Section, NACA RM A56 K26, March 1957.

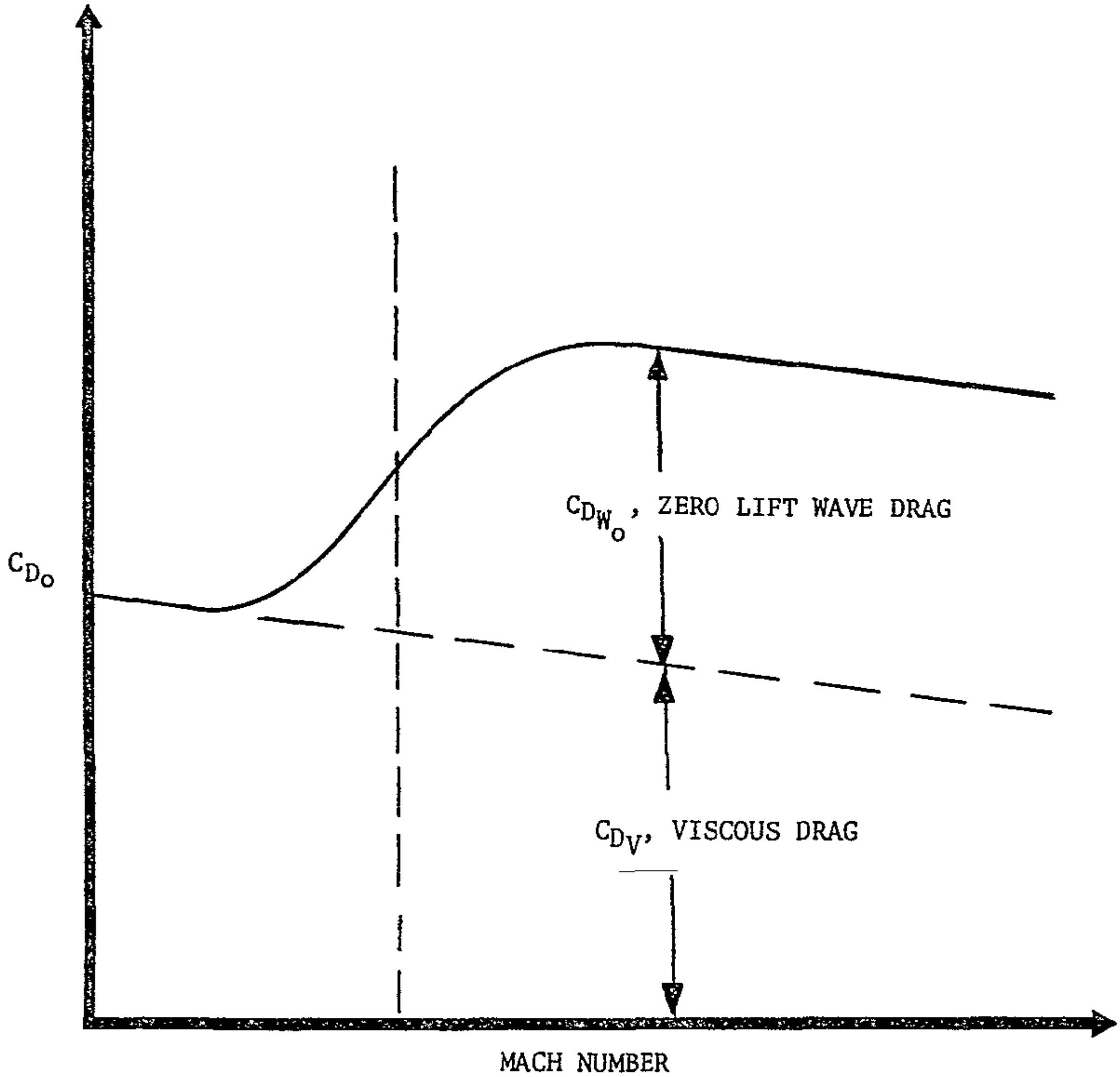


FIGURE 4.2-1. TYPICAL ZERO LIFT DRAG COEFFICIENT

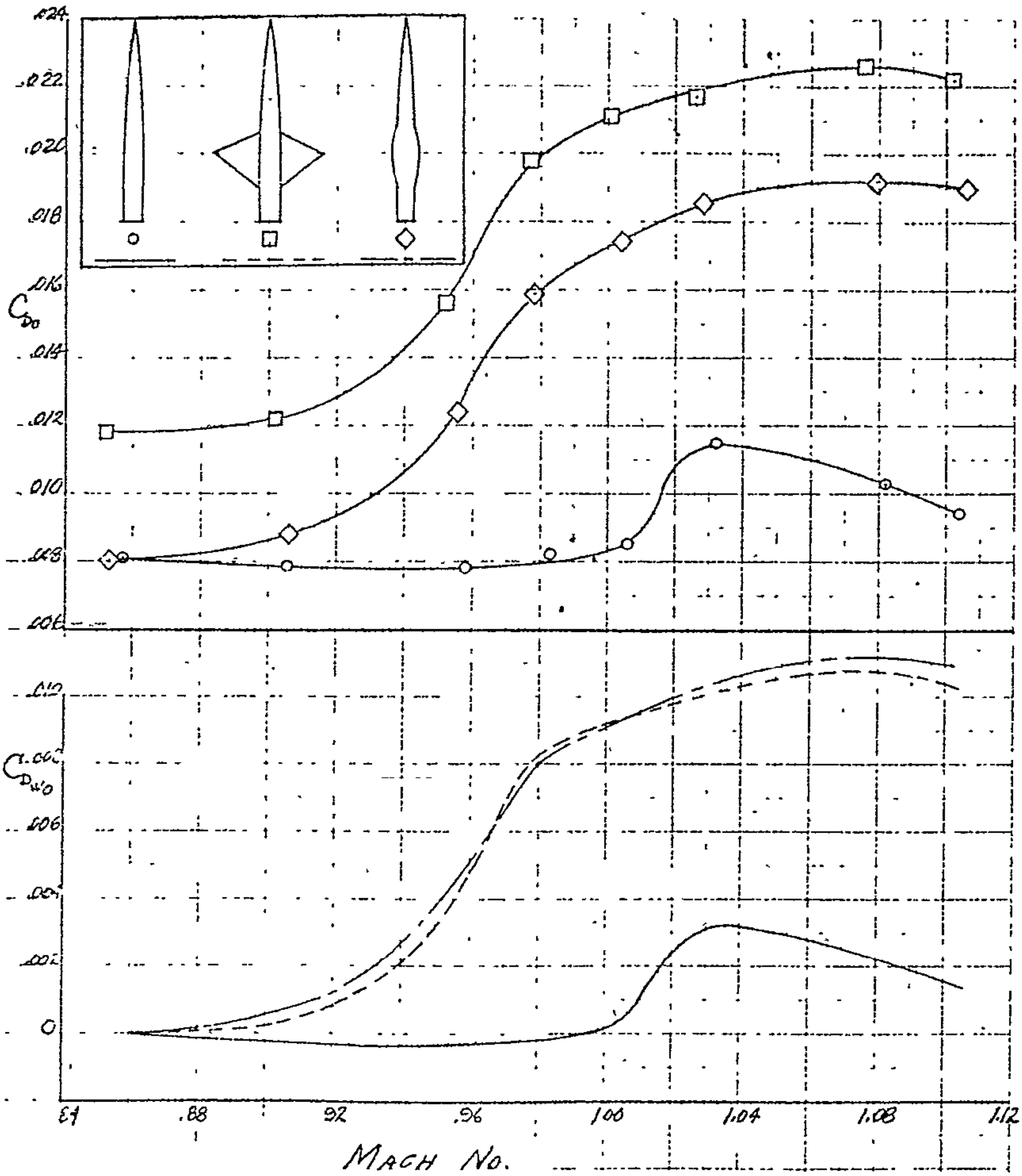


FIGURE 4.2-2(a)

DRAG RISE VERSUS MACH NUMBER FOR CYLINDRICAL BODY, UNSWEPT WING-BODY COMBINATION AND COMPARABLE BODY OF REVOLUTION

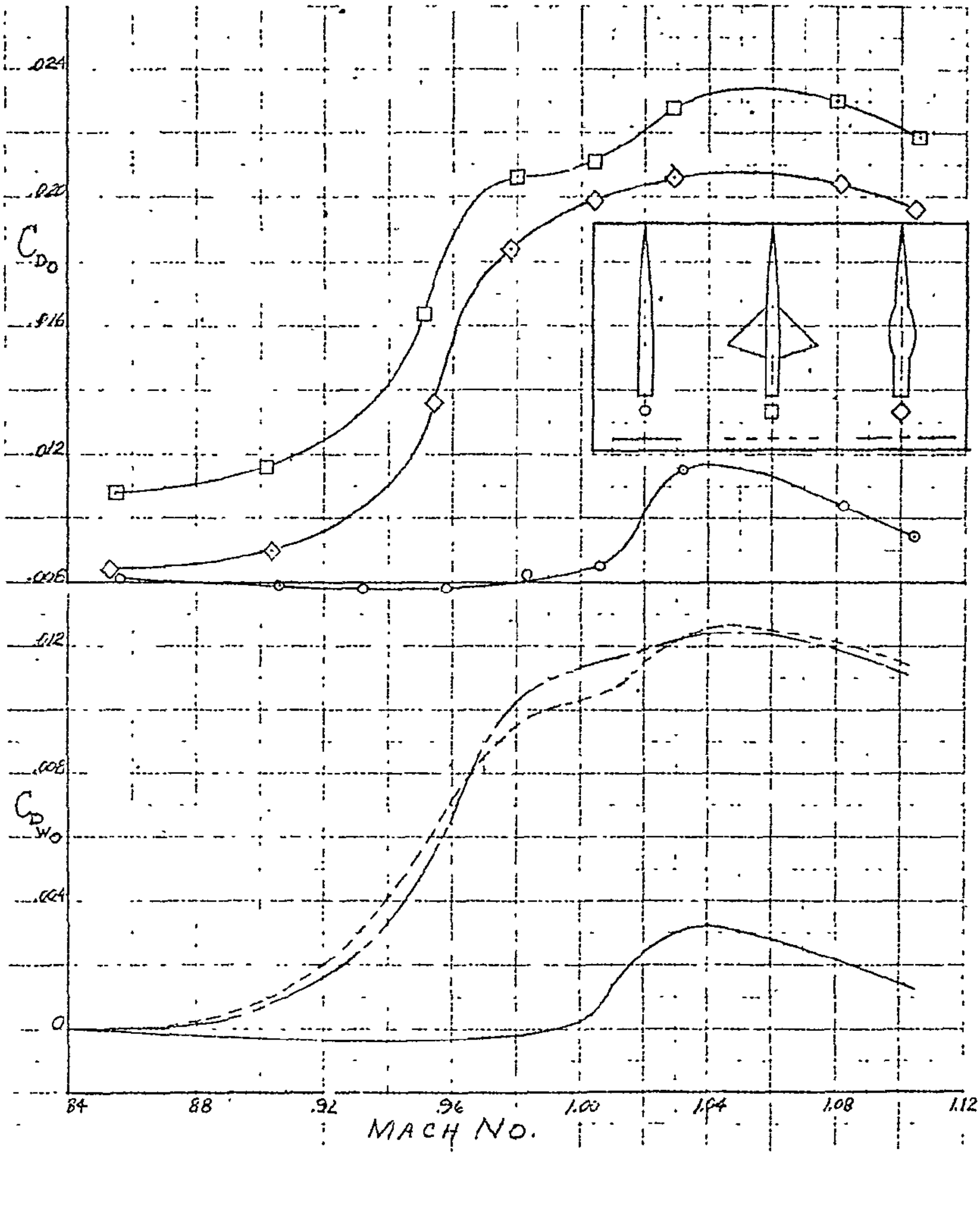


FIGURE 4.2-2(b)

DRAG RISE VERSUS MACH NUMBER FOR CYLINDRICAL BODY, DELTA WING AND BODY COMBINATIONS, AND COMPARABLE BODY OF REVOLUTION

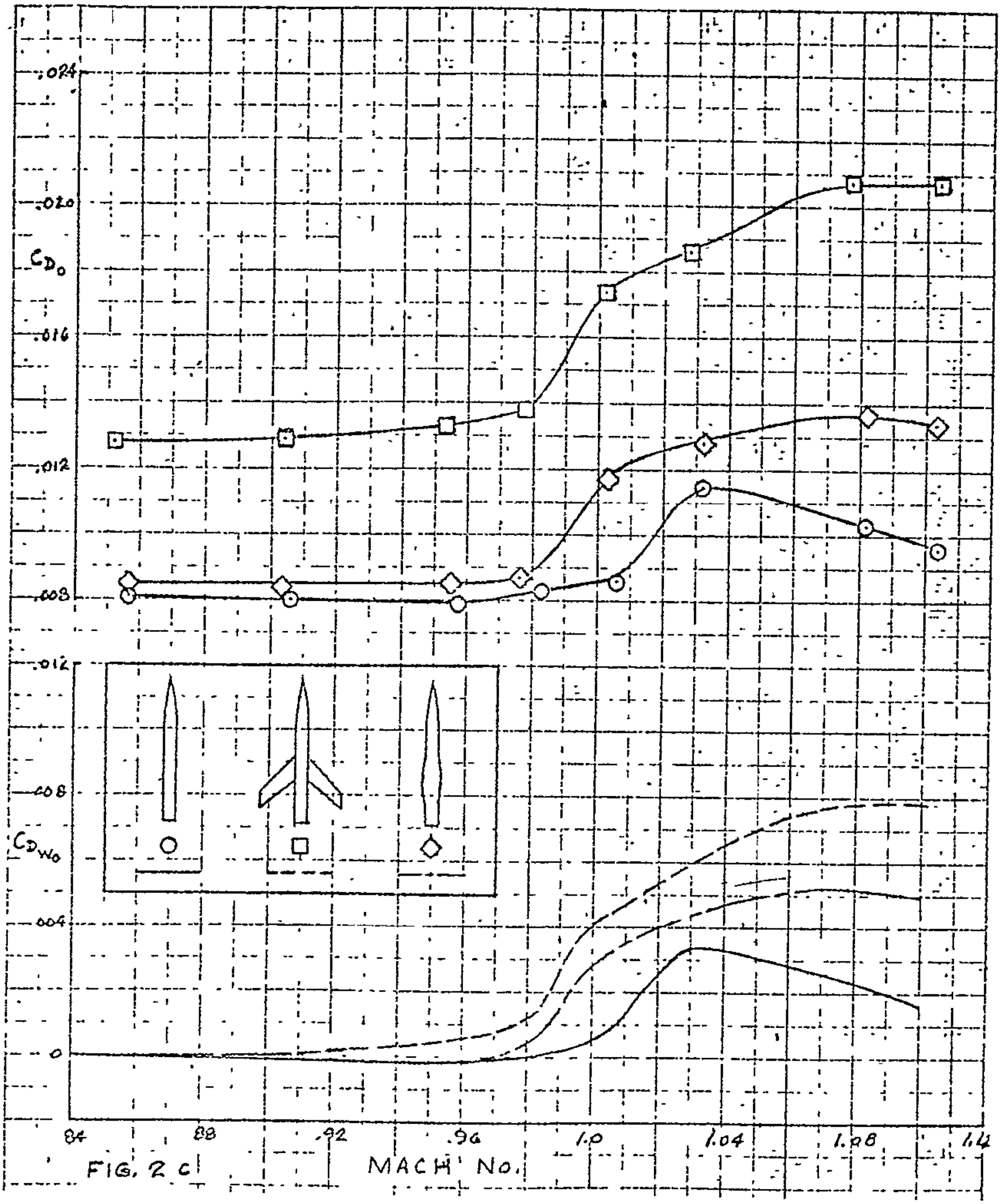


FIGURE 4.2-2 (c)

DRAG RISE VERSUS MACH NUMBER FOR CURVED BODY, SWEPT WING AND CYLINDRICAL BODY COMBINATION, AND COMPARABLE BODY OF REVOLUTION

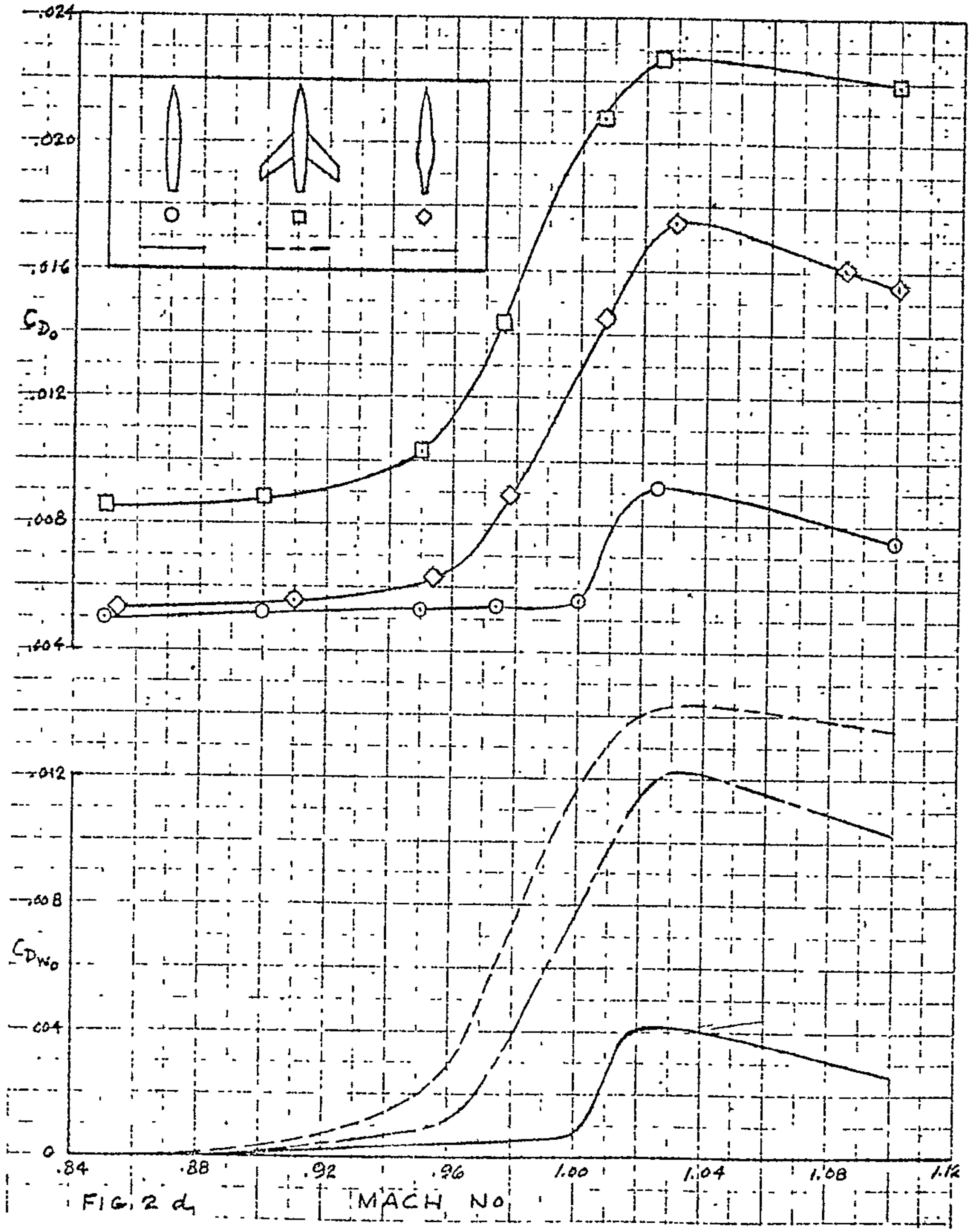


FIGURE 4.2-2(d)

DRAG RISE VERSUS MACH NUMBER FOR CURVED BODY, SWEPT WING AND CURVED BODY COMBINATION, AND COMPARABLE BODY OF REVOLUTION

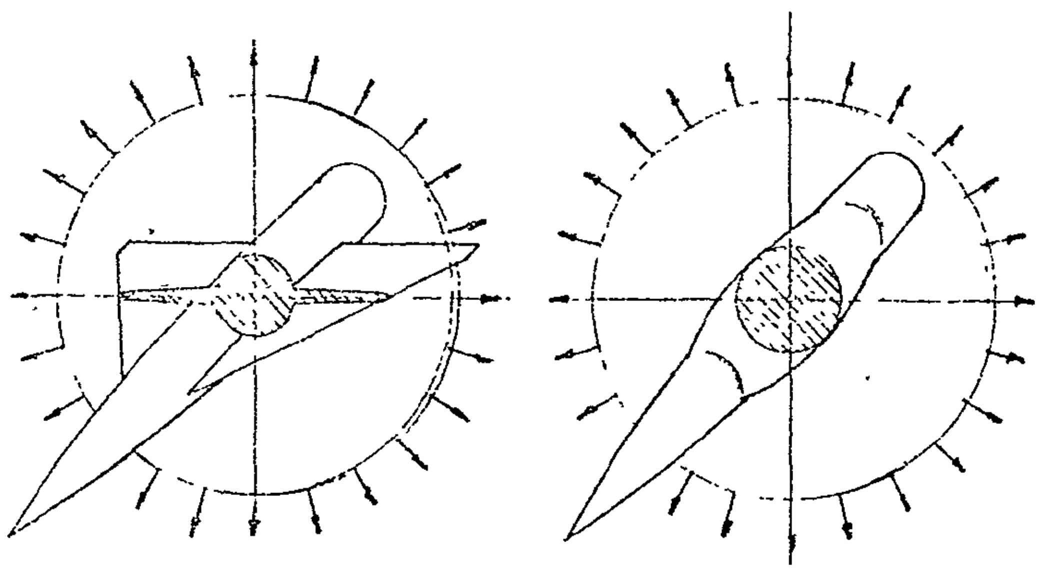


FIGURE 4.2-3. INCOMPRESSIBLE NATURE OF TRANSONIC FLOW FIELD

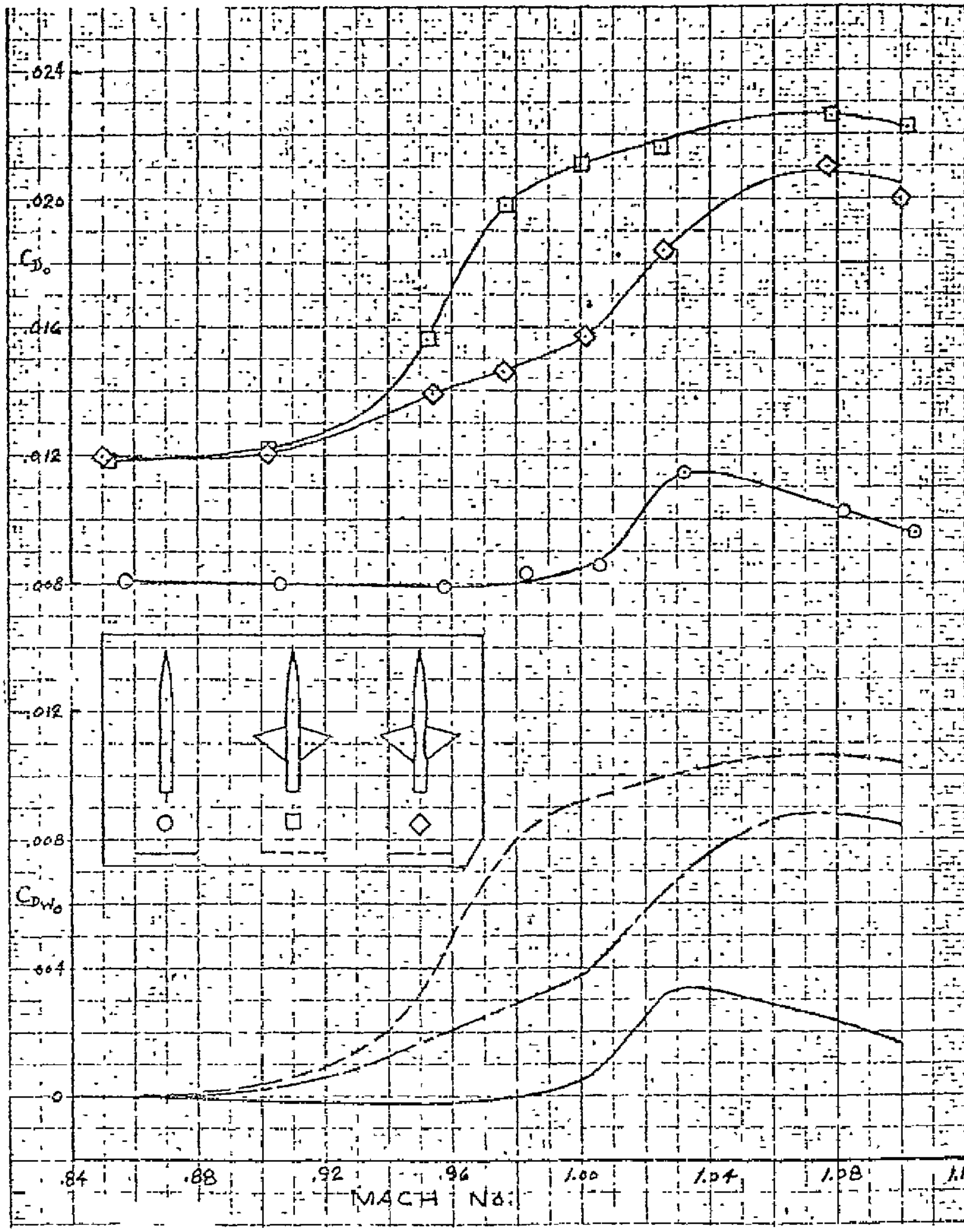


FIGURE 4.2-4(a)

THE EFFECT ON TRANSONIC DRAG OBTAINED BY INDENTING THE BODIES OF WING-BODY COMBINATIONS (UNSWEPT WING)

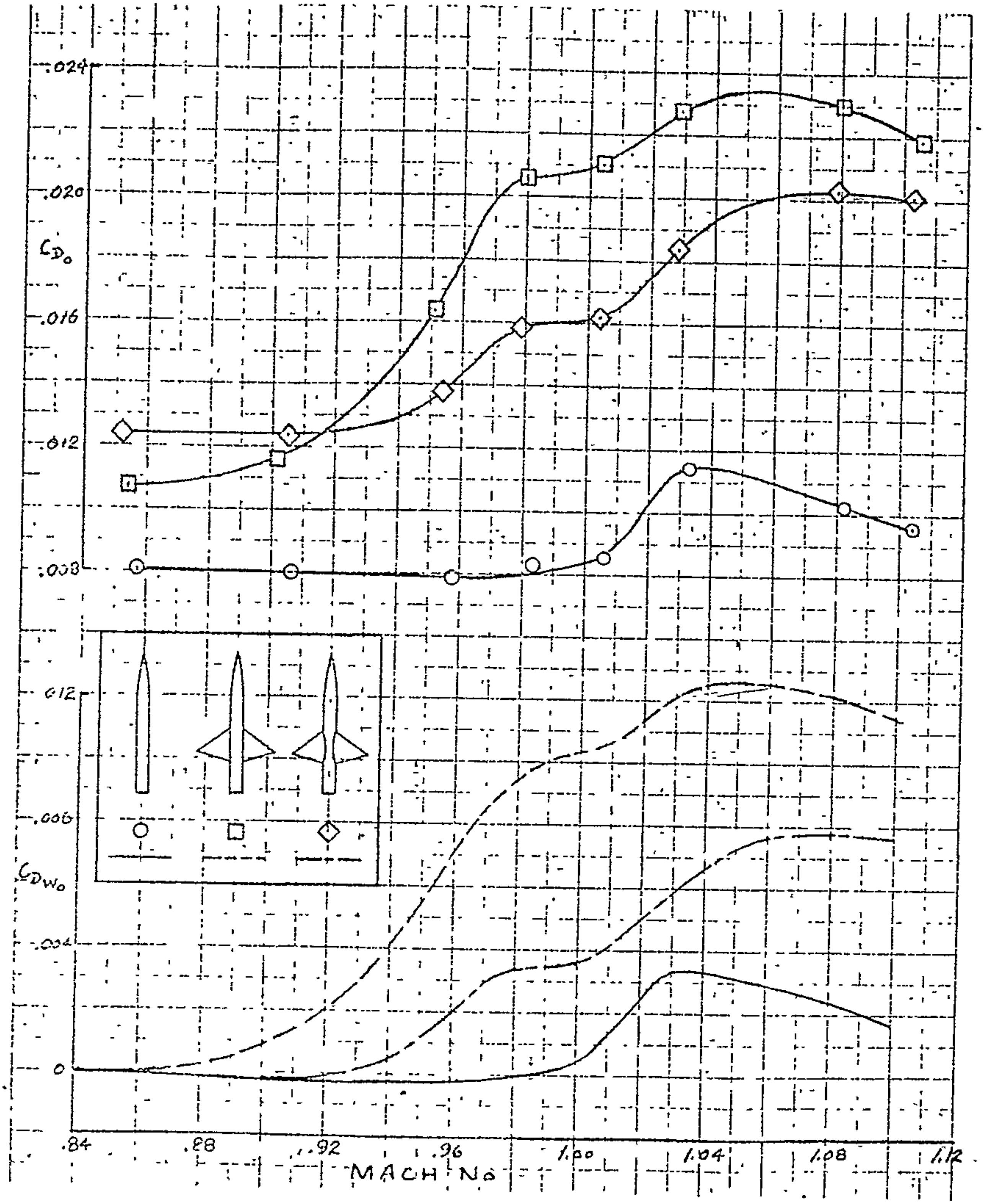


FIGURE 4.2-4(b)

THE EFFECTS ON TRANSONIC DRAG OBTAINED BY INDENTING THE BODIES OF WING-BODY COMBINATIONS (DELTA WING)

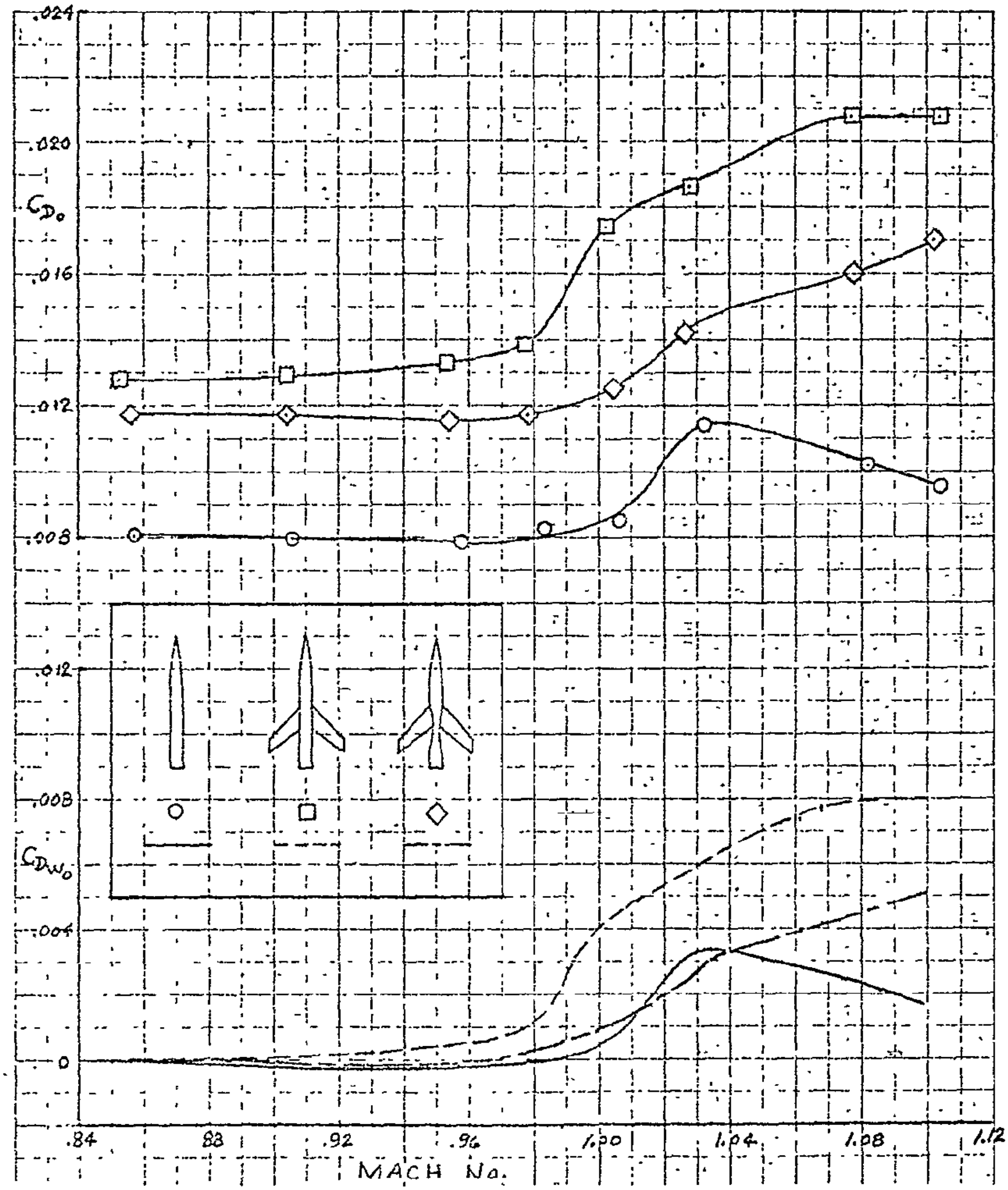


FIGURE 4.2-4(c)

THE EFFECTS OF TRANSONIC DRAG OBTAINED BY INDENTING THE BODIES OF WING-BODY COMBINATIONS (SWEPT WING)

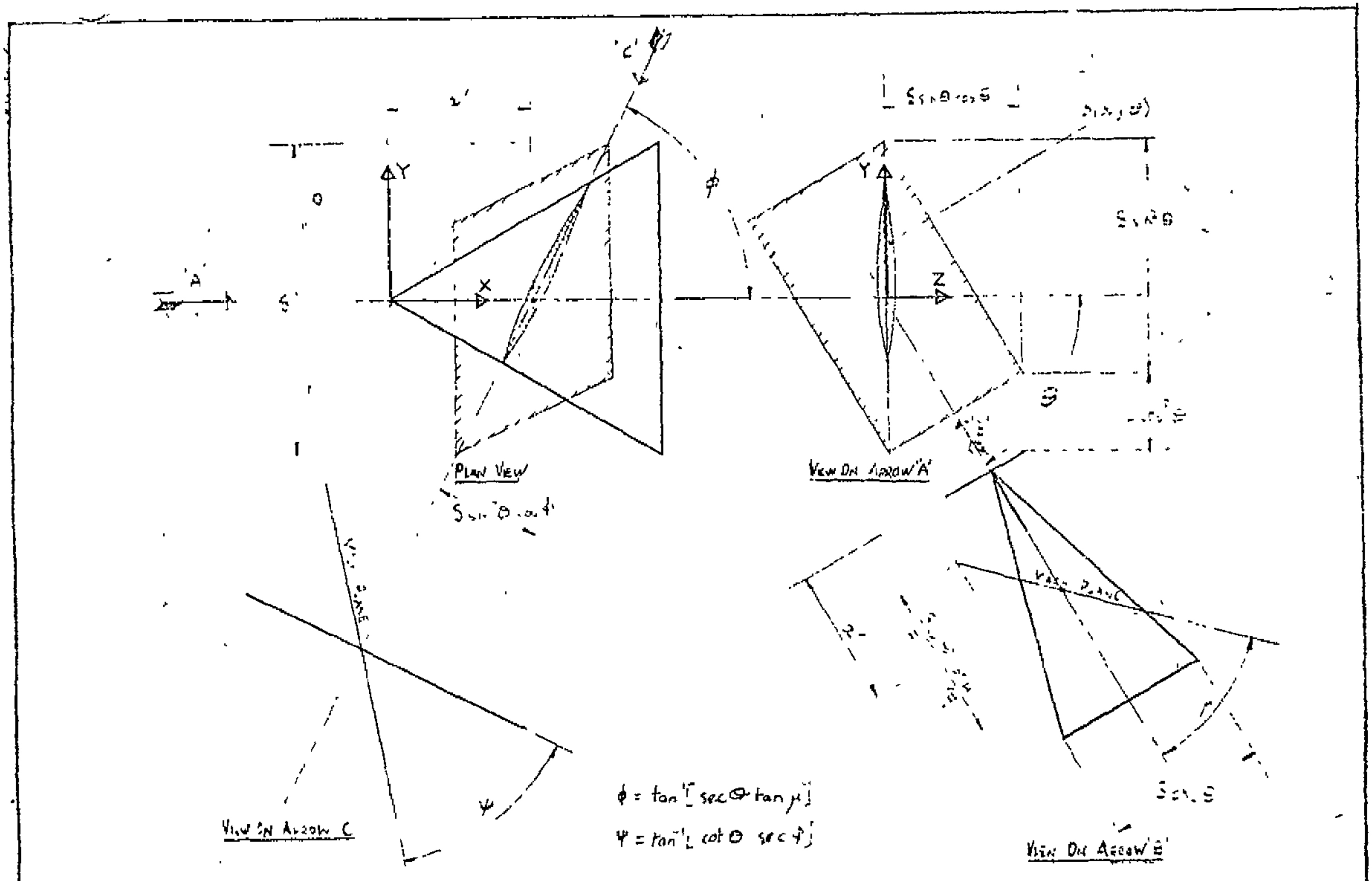


FIGURE 4.2-5. COORDINATE SYSTEM AND MACH PLANES

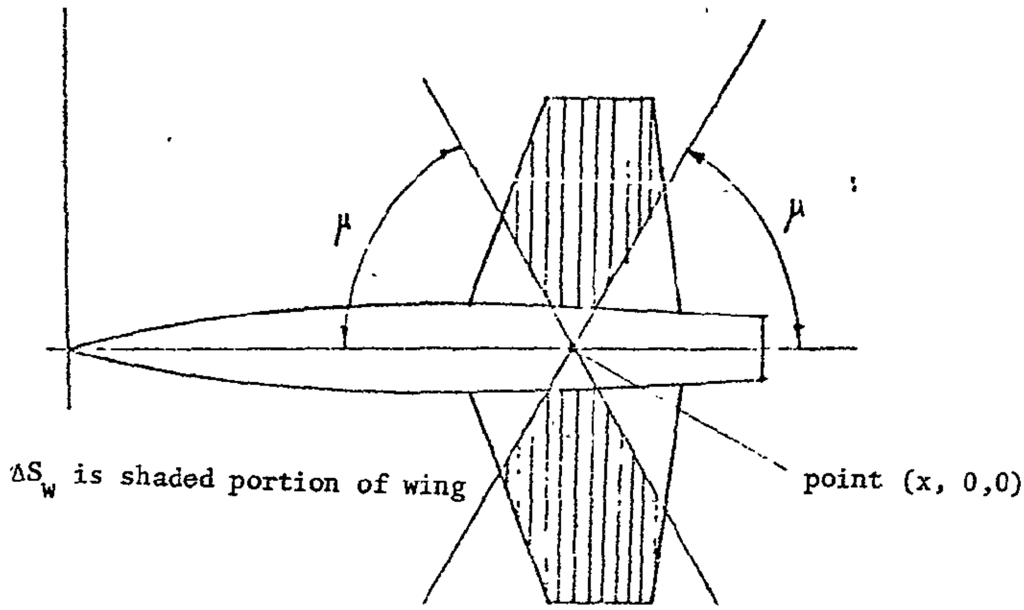


FIGURE 4.2-6. WING REGION FOR OBTAINING A(x).

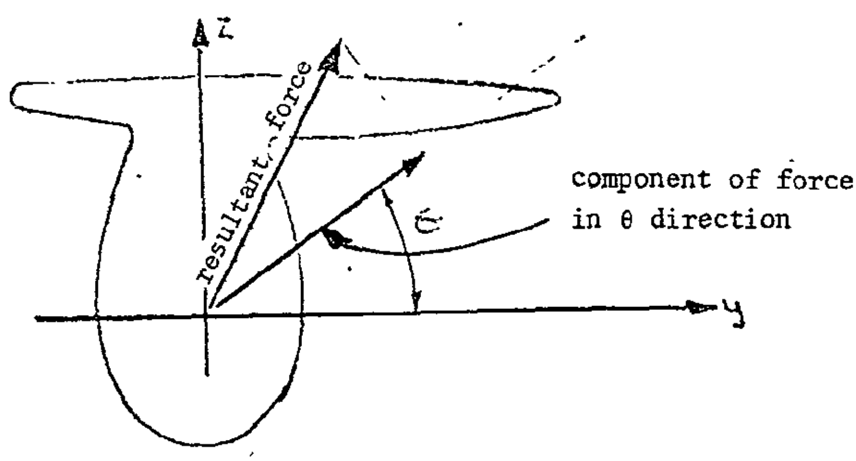


FIGURE 4.2-7. INCLUSION OF LIFT IN SUPERSONIC AREA RULE

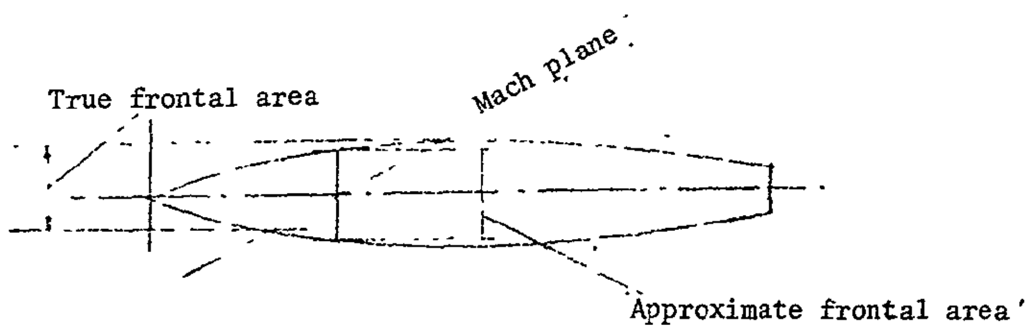


FIGURE 4.2-8. FUSELAGE AREA CONTRIBUTION

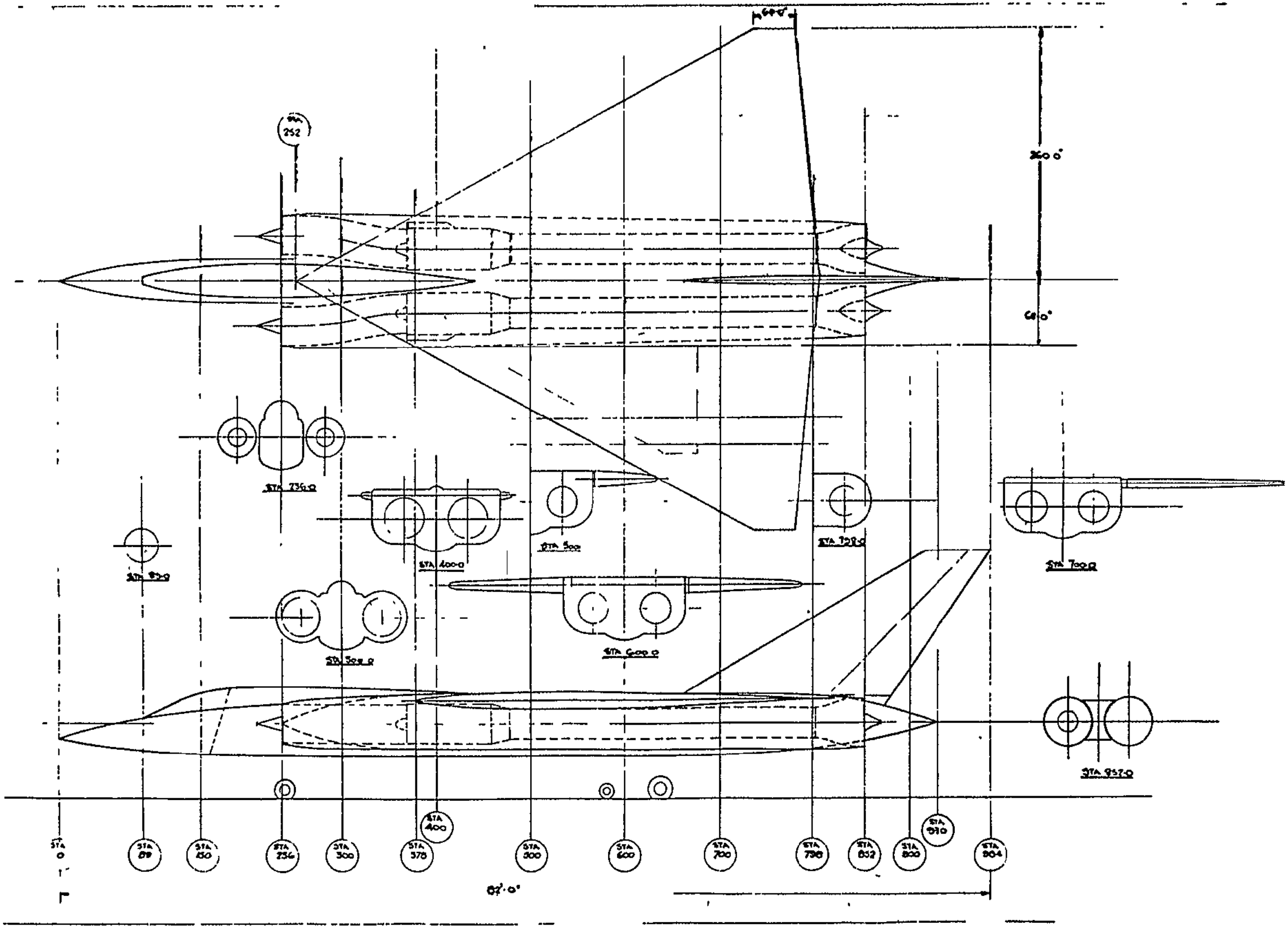


FIGURE 4.2-9(a) LAYOUT OF BASIC EXAMPLE AIRCRAFT

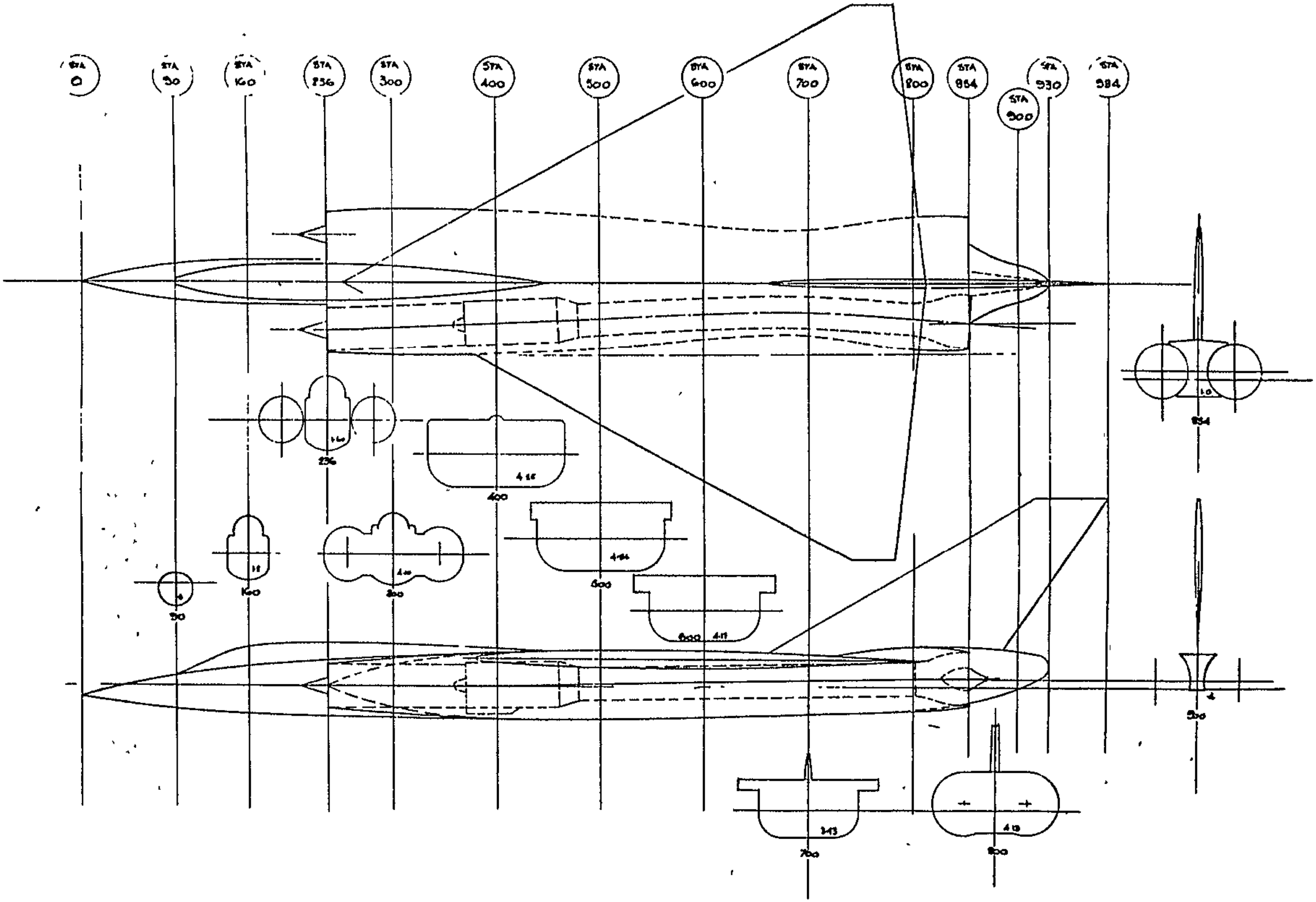


FIGURE 4.2-9(b) LAYOUT OF M = 1.0 AREA RULED EXAMPLE AIRCRAFT

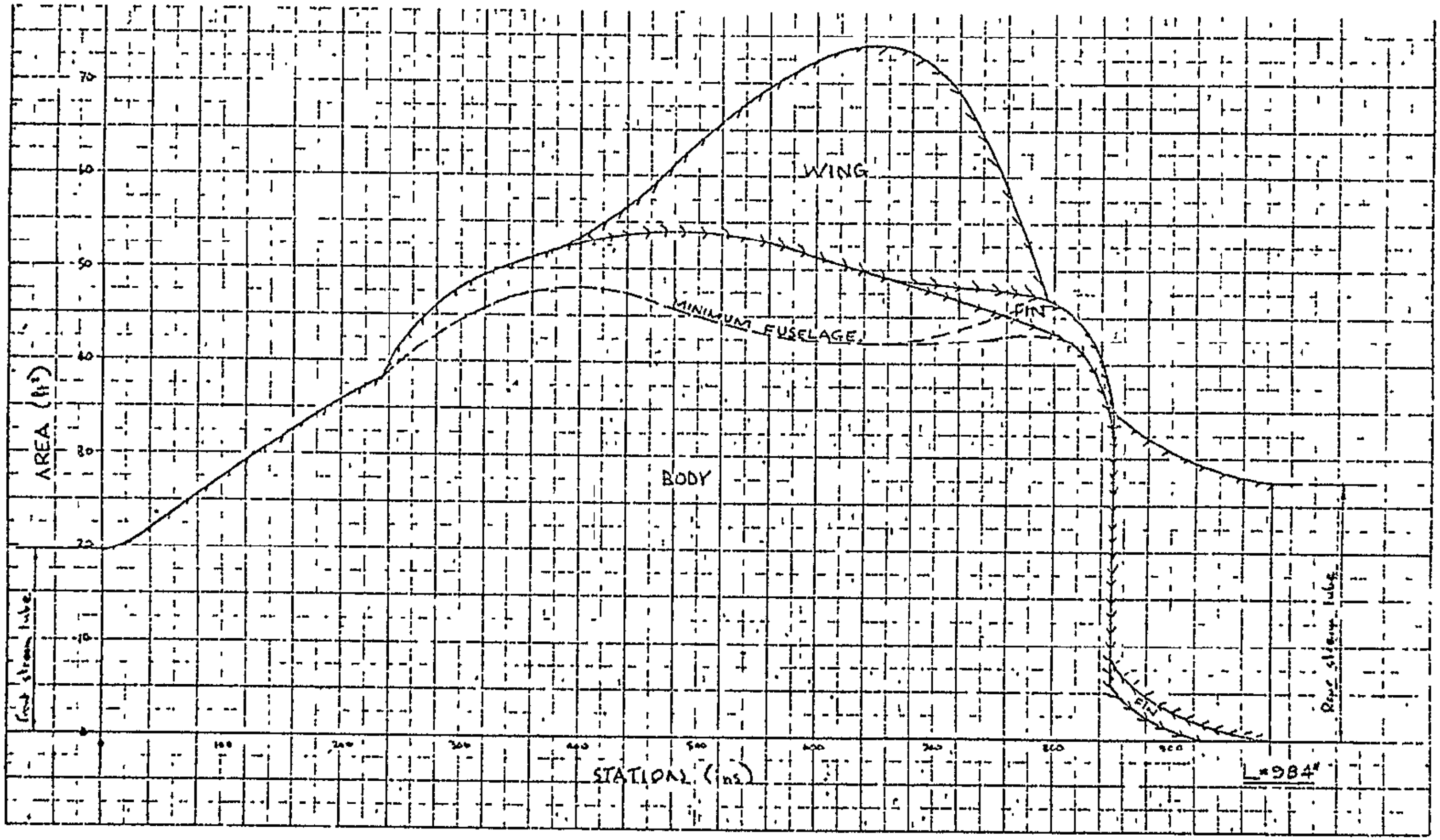


FIGURE 4.2-10(a). AREA DISTRIBUTION OF BASIC A/C

ORIGINAL PAGE IS OF POOR QUALITY

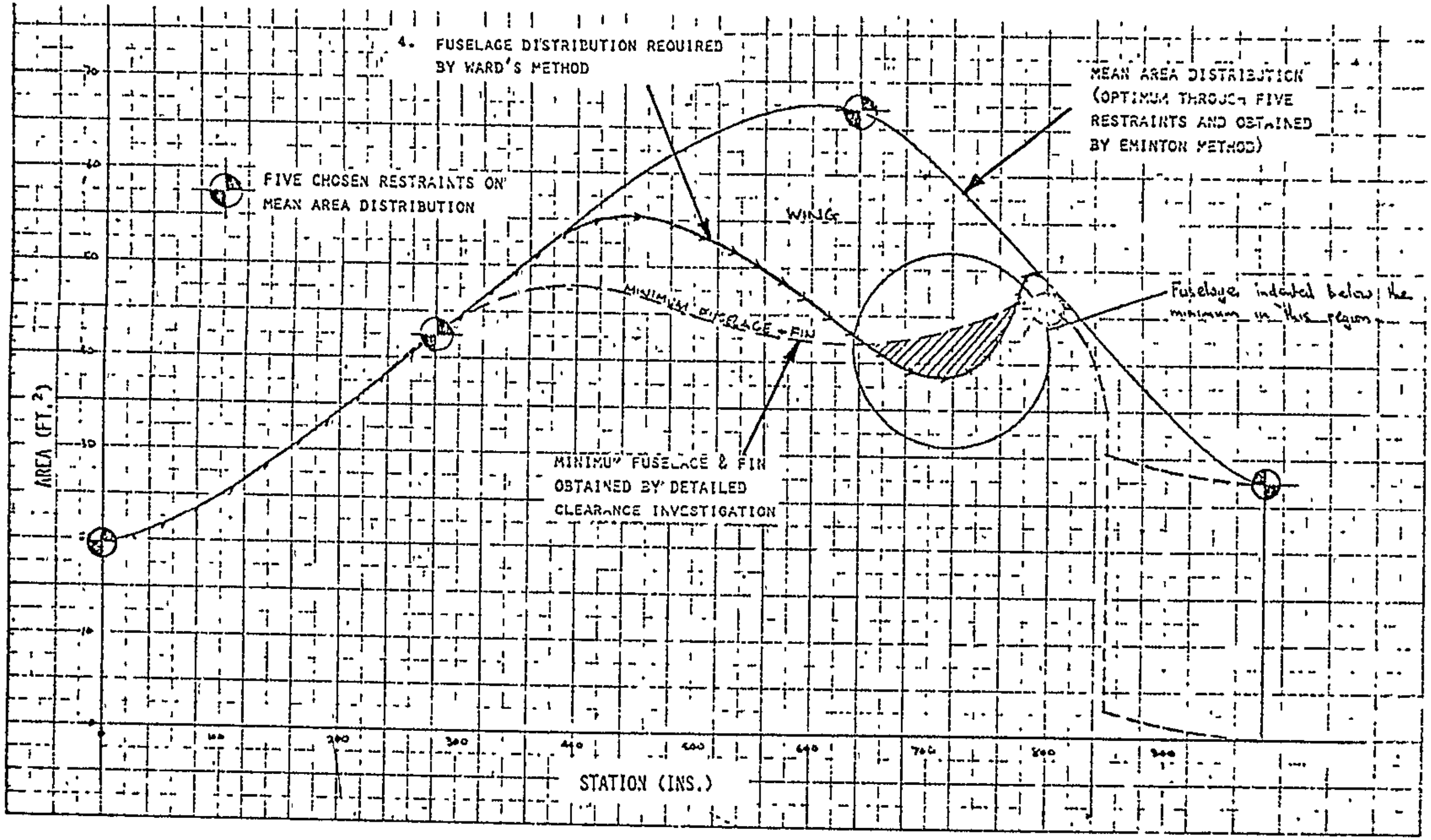


FIGURE 4.2-10(b). AREA DISTRIBUTION OF M = 1.0 AIRCRAFT

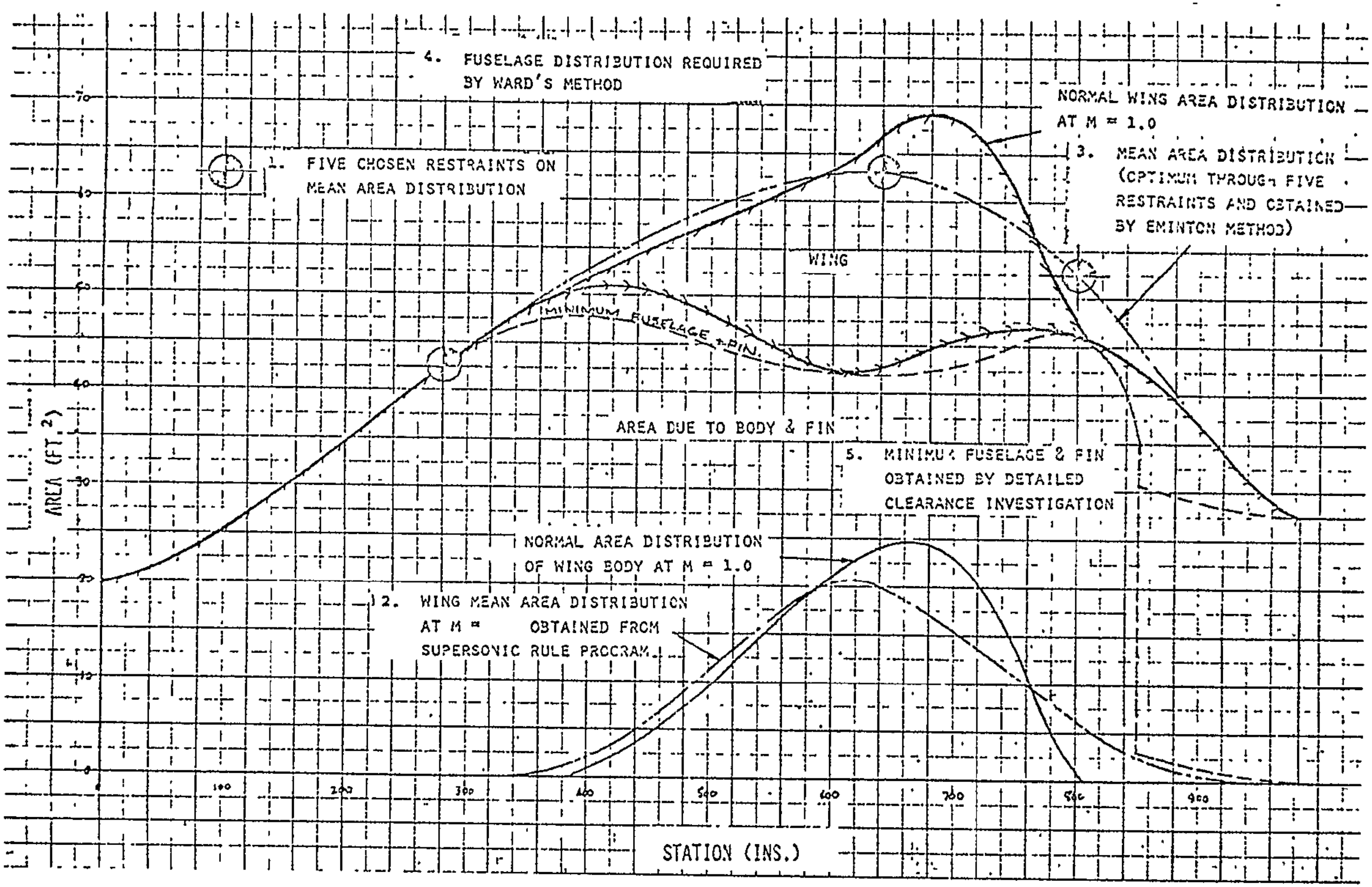


FIGURE 4.2-10(c). AREA DISTRIBUTION OF M = 1.2 AIRCRAFT

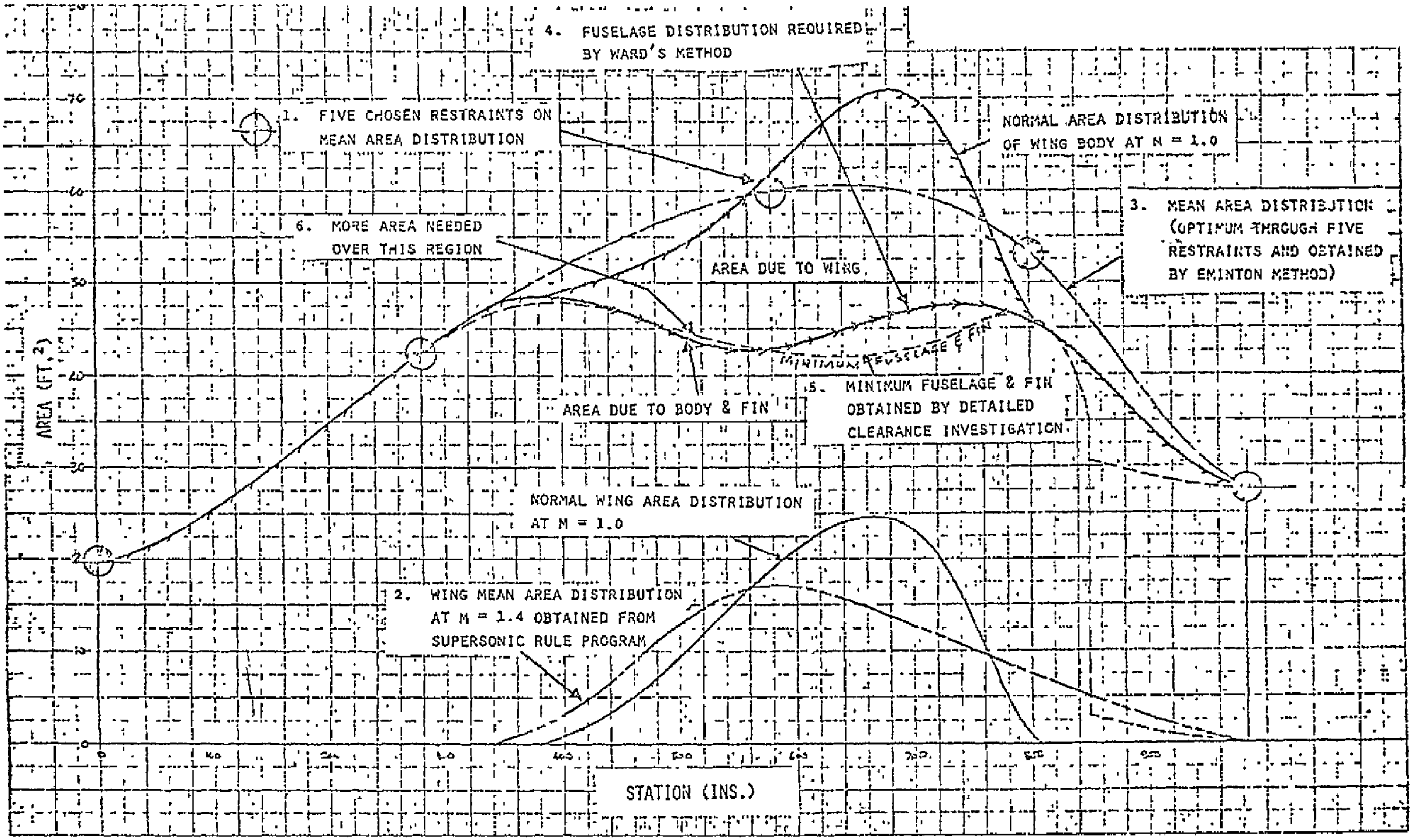


FIGURE 4.2-10(d). AREA DISTRIBUTION OF M = 1.4 AIRCRAFT

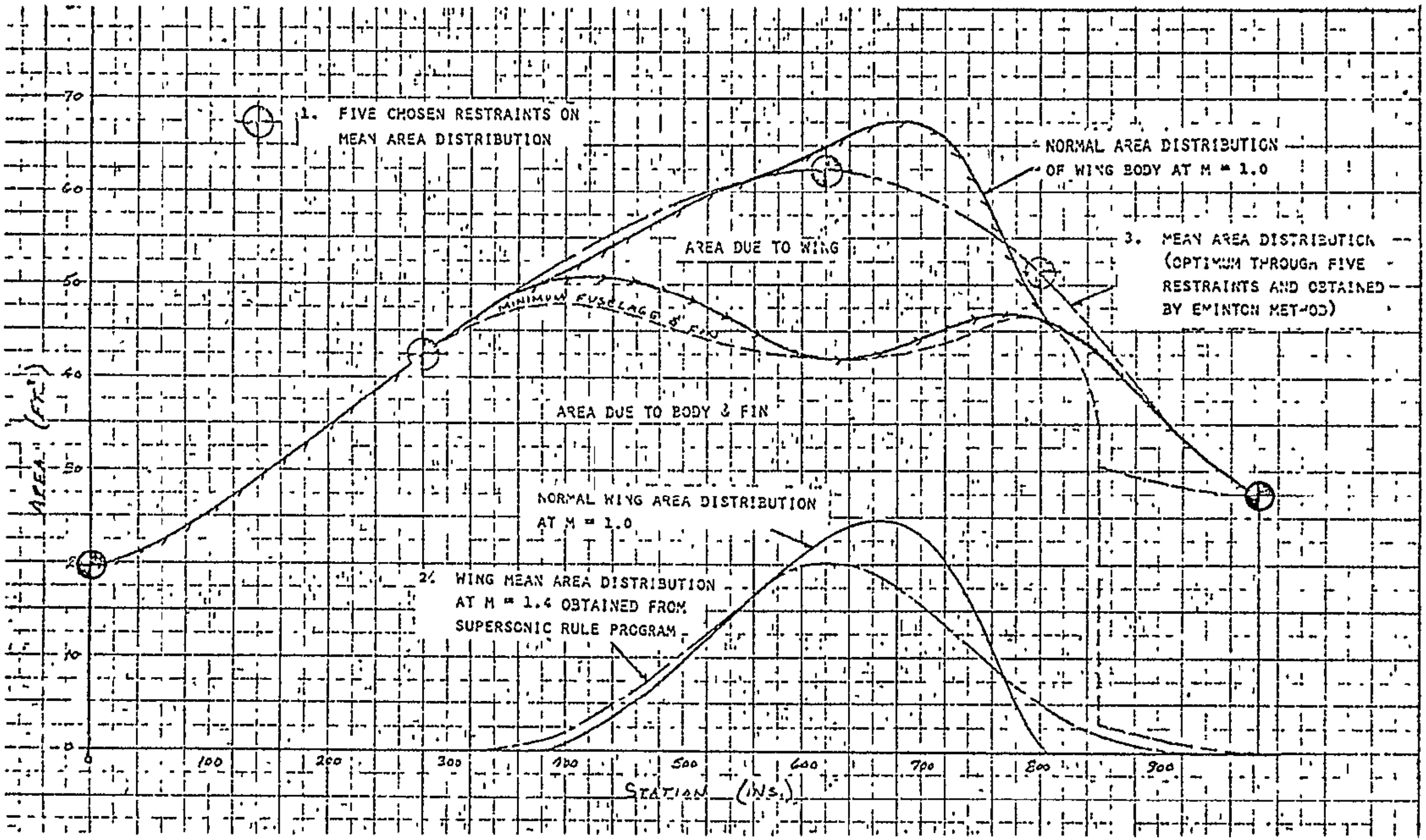


FIGURE 4.2-10(e). AREA DISTRIBUTION OF M = 1.0 TO M = 1.4 AIRCRAFT

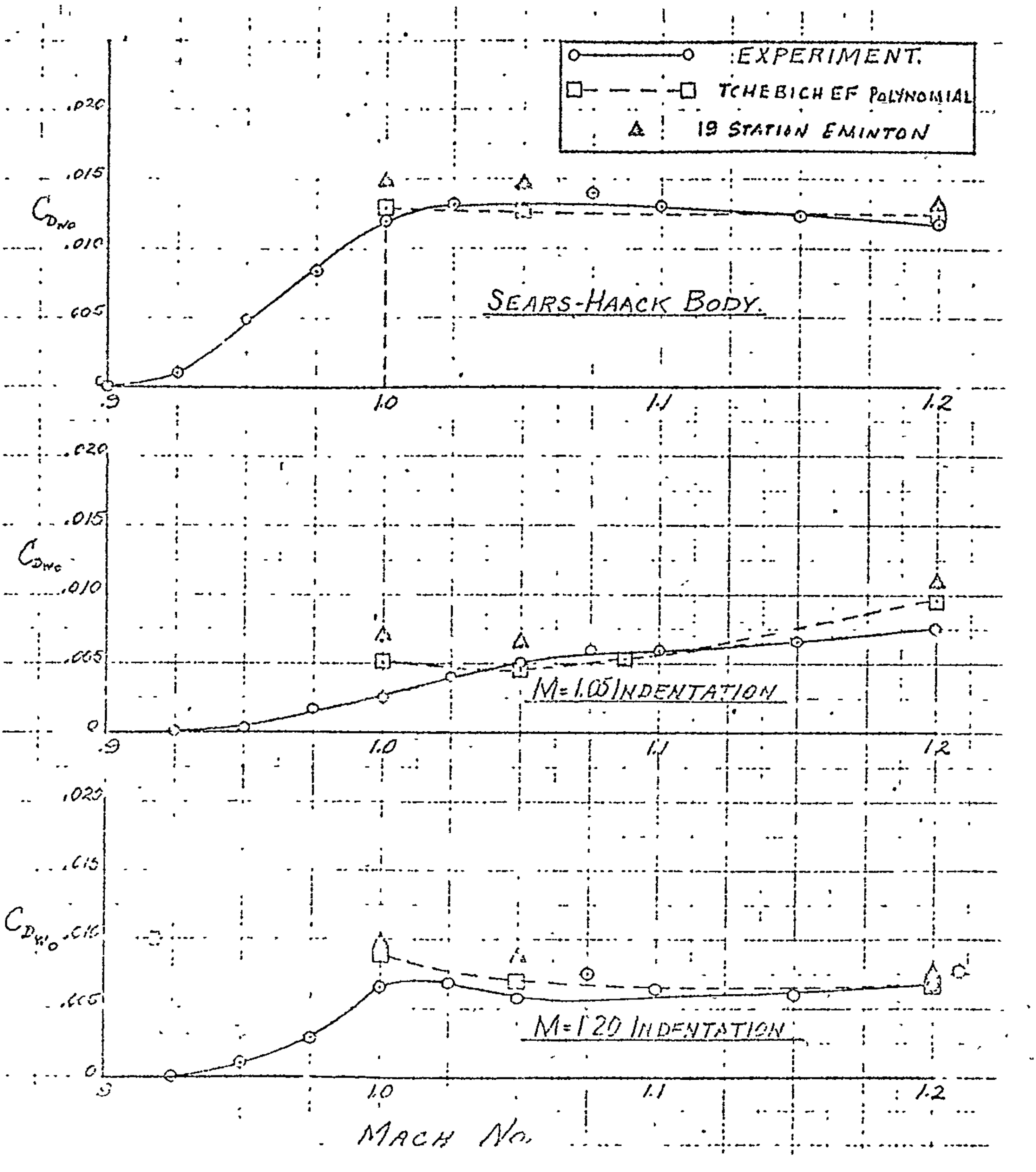


FIGURE 4.2-11(a). EXPERIMENTAL AND COMPUTED ZERO LIFT WAVE DRAG COEFFICIENTS WITH VARIOUS BODIES

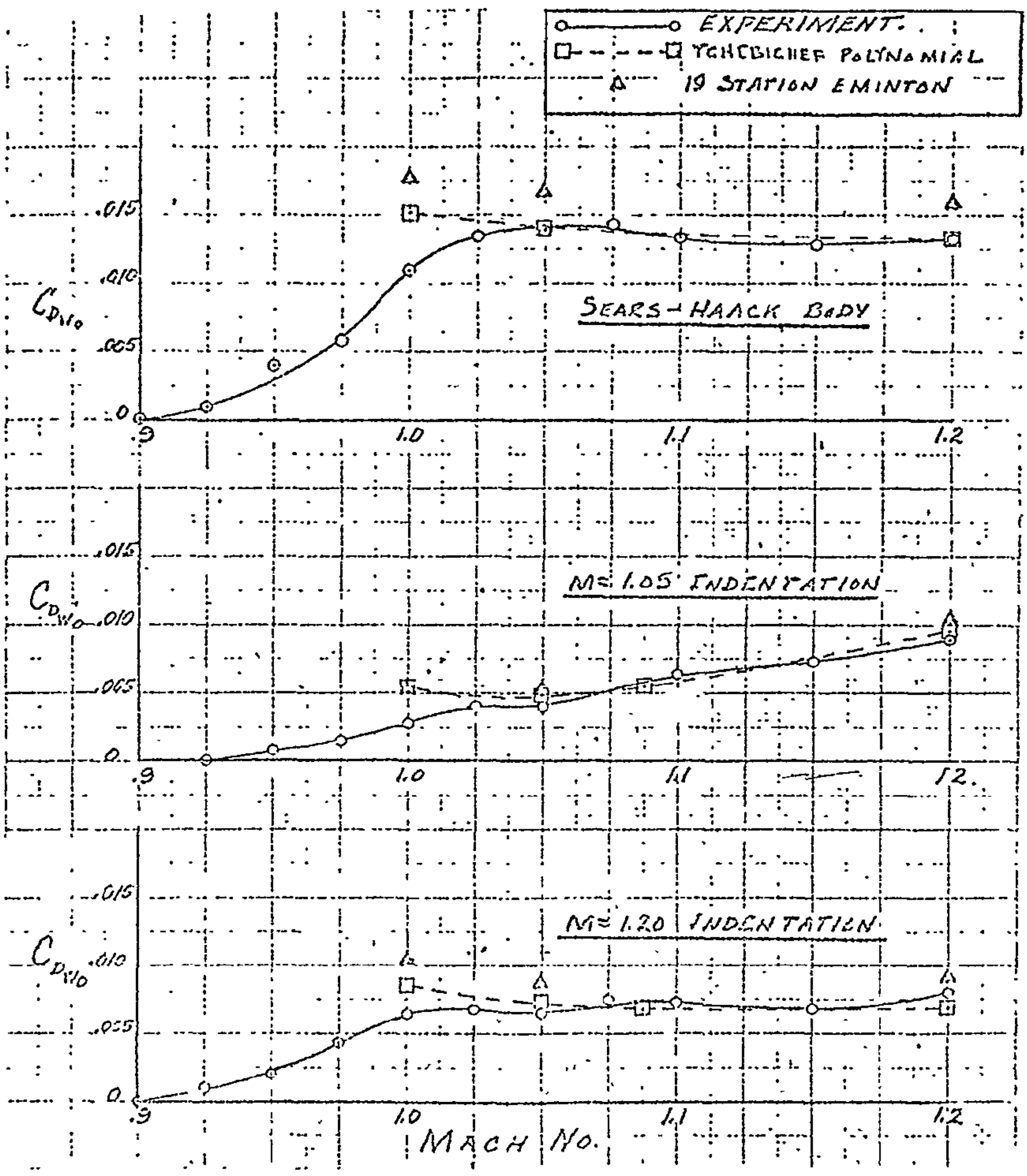


FIGURE 4.2-11(b) EXPERIMENTAL AND COMPUTED ZERO LIFT WAVE DRAG COEFFICIENTS FOR MODIFIED WING WITH VARIOUS BODIES

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE OF CONTENTS FOR SECTION 4.3, PROGRAM TREND

4.3 PROGRAM TREND: A RAPID SUBSONIC/SUPERSONIC/HYPERSONIC AERODYNAMIC TRADE-OFF CODE

Program TREND provides rapid aerodynamic lift, drag and moment estimates in the subsonic, supersonic, and hypersonic flight regimes. The program is primarily designed to estimate high lift-drag reentry vehicle aerodynamic characteristics. The class of vehicles which may be analyzed with the program is of greater range than the primary class of reentry vehicle. However, some program modification may be required as the vehicle shapes diverge from the prime class of vehicles. Program modification may also become desirable where detailed wind tunnel results are available for a specific configuration. When the computed aerodynamic characteristics are matched to the experimental characteristics in this manner program TREND may be used to estimate aerodynamic trade-offs rapidly as the configuration geometry is perturbed.

In the hypersonic flight regime, the program contains an optional aerodynamic heating computation capability. It should be noted that the program does not possess a transonic aerodynamic characteristic estimation capability. The original program TREND was prepared under a previous Air Force Flight Dynamics Laboratory contract. Original detailed program documentation is presented in Reference 1. An example of the use of this program is given in Reference 2.

4.3.1 Basic Configuration Types and Limitations

The basic configuration types which may be analyzed by program TREND are summarized in Table 4.3-1. Typical configurations are illustrated in Figures 4.3-1 and 4.3-2. It should be noted that the Type I configuration may be extended to incorporate a horizontal tail.

Total vehicle lift and drag coefficients for all flight regimes are computed by a *component buildup* that allows the user to select the prediction techniques most applicable to his configuration. In the subsonic and supersonic flight regimes the lift of a composite configuration is computed as the sum of

a. body lift
b. exposed wing lift
c. exposed horizontal tail lift
d. lift increment due to horizontal tail or elevon deflection

The total drag is computed as the sum of the following components:

a. minimum drag
b. drag due to lift
c. trim drag

Lift and drag at hypersonic speeds are obtained from the normal and axial force coefficients computed by the methods presented below.

The limitations imposed in the *subsonic* flight regime are that Mach critical is not reached and that the angle of attack is less than 18 degrees for the wing-body configurations. Methods are included to compute

a. lift above wing stall

b. drag due to lift above polar break up to these limitations for the winged configurations

In the *supersonic* regime the Mach number limits are 1.2 to 3.5, and the angle of attack is limited to 12 degrees for all configurations. This angle of attack limit is well above the angle of attack for maximum lift to drag ratio.

The prediction techniques applicable to the *hypersonic* regime are limited to continuum flow, Mach number greater than 3.5, and an angle of attack range between 5 and 50 degrees.

4.3.2 Sensitivity Analysis Options

Program TREND can be used to obtain vehicle aerodynamic sensitivities to configuration and/or flight path perturbations.

The sensitivity factor analysis provides the partial derivatives of specified aerodynamic performance parameters with respect to the input vehicle geometric or flight path parameters. Derivatives are calculated directly from the sensitivity factor equations or by finite differences. The latter method must be used where sensitivity factor equations are not available. The sensitivity analysis calculates incremental aerodynamic performance parameter values for increment values of geometric and flight path parameters. In general,

$$\Delta P = \frac{\partial P}{\partial V_1}\Delta V1 + \frac{\partial P}{\partial V_2}\Delta V2 + \cdots \frac{\partial P}{\partial V_n}\Delta V_n + \frac{\partial P}{\partial M}\Delta M + \frac{\partial P}{\partial h}\Delta h + \frac{\partial P}{\partial \delta e}\Delta \delta e + \frac{\partial P}{\partial \alpha}\Delta \alpha \qquad (4.3.1)$$

where P = an aerodynamic performance parameter; L, D, or L/D
V = vehicle geometric parameter such as wing sweep or leading edge radius
M = Mach number
h = altitude
δe = elevon deflection
α = angle of attack

Partial derivatives may be computed at either constant angle of attack or constant lift coefficient in Equation 4.3.1.

It should be noted that certain aerothermodynamic sensitivities must be computed by an integration along the vehicle flight path, for example, heat shield mass sensitivity with respect to vehicle geometric parameters and thermal protection system properties. The option to compute such sensitivities along the *equilibrium glide path of the nominal vehicle* is available in TREND.

4.3.3 Subsonic Flight Regime

4.3.3.1 Subsonic Lift

4.3.3.1(a) Modified Body of Revolution.

Subsonic lift of a body or modified body of revolution is based on the DATCOM of Reference 3, Section 4.2.1. The method combines slender-body potential flow predictions with a viscous crossflow force proportional to the square of the angle of attack and modifies the results for noncircular body sections, Reference 4.

$$C_{L_B} = C_{L_P} + C_{L_V} \tag{4.3.2}$$

$$C_{L_P} = (C_{N_\alpha})_b \, (k_2 - k_1) \, F_m \alpha' \, \frac{S_b}{S} \tag{4.3.3}$$

$$C_{L_V} = C_{d_c} \, F_m \alpha^{2\prime} \, \frac{S_p}{S} \tag{4.3.4}$$

where

$(C_{N_\alpha})_b$ = 2.0

$k_2 - k_1$ = reduced mass factor obtained from Figure (4.2.1.1-6a) of DATCOM

F_m = cross-sectional shape parameter, Reference 1

C_{d_c} = $1 + 1.2\,(M\alpha')^3$

S_b = base area

S_p = planform area

S = reference area

α_{LO_B} = body angle of attack at zero lift

The term α' is the effective angle of attack and is calculated from $\alpha' = \alpha - \alpha_{LO_B}$. Lift curve slope is a nonlinear function of angle of attack and is determined from

$$C_{L\alpha_B} = (C_{N_\alpha})_b (k_2 - k_1) \, F_m \frac{S_b}{S} + 2C_{d_c} \, F_m \alpha' \, \frac{S_p}{S} \tag{4.3.5}$$

4.3.3.1(b) Planar Body

This class of bodies is treated as a low aspect ratio wing possessing a linear lift curve slope. The method used to determine C_{L_α} is based on the expression for normal force curve slope at zero lift contained in the subsection 4.8.1 of DATCOM.

$$C_{L_{\alpha_B}} = \left[.54 \left(R_{1/3LE}/b\right)^{-.153} \left(\pi AR/2 + 2A_{Y_2}/S\right) \frac{4}{4+AR} \right] \frac{S_P}{S} \qquad (4.3.6)$$

$$C_{L_B} = C_{L_{\alpha_B}} \alpha' \qquad (4.3.7)$$

The term $.54\ (R_{1/3LE}/b)^{-.153}$ approximates the curve in Figure (4.8.1.2-11) of DATCOM.

4.3.3.1(c) Wing

The lift contribution of exposed wing panels is computed as wing-alone lift modified for wing-body carryover. The wing-alone lift-curve slope is computed and the carryover factors $K_{B(W)}$ and $K_{W(B)}$ are obtained by the method of Pitts, Nielsen, and Kaattari, Reference 5.

$$C_{L_{\alpha_W}} = C_{L_{\alpha_{WE}}} \left[K_{B(W)} + K_{W(B)} \right] \qquad (4.3.8)$$

$$C_{L_{\alpha_{WE}}} = \frac{\pi AR_{WE}\, G}{G + \sqrt{1 + G^2 \left(1 - M^2 AR_{WE}^2/4\right)}} \frac{S_{WE}}{S} \qquad (4.3.9)$$

$$G = 2\cos \Lambda_{c/2\,W} / AR_{WE} \qquad (4.3.10)$$

$$S_{WE} = \frac{(b_W - W)}{2} \left(C_{R_{WE}} + C_{T_W}\right) \qquad (4.3.11)$$

$$AR_{WE} = \left(b_W - W\right)^2 / S_{WE} \qquad (4.3.12)$$

where W is the average body width at the wing junction and the subscript WE pertains to the exposed wing.

4.3.3.1(d) Horizontal Tail

Lift contribution of the exposed part of the horizontal tail is determined by the above exposed wing method modified for downwash and dynamic pressure reduction at the tail.

$$C_{L_{\alpha_H}} = C_{L_{\alpha_{HE}}} \left[K_{B(W)} + K_{W(B)}\right] \left(1 - d\epsilon/d\alpha\right) q_H/q \tag{4.3.13}$$

Increment of lift resulting from a horizontal tail deflection is obtained from the tail effectiveness parameter for lifting surface incidence and modified for dynamic pressure reduction at the tail, Reference 5,

$$\Delta C_{L_H} = C_{L_{\delta_H}} \delta_H \tag{4.3.14}$$

$$C_{L_{\delta_H}} = C_{L_{\alpha_{HE}}} \left[k_{B(W)} + k_{W(B)}\right] q_H/q \tag{4.3.15}$$

For a leading edge down deflection of the horizontal tail, δ is negative.

4.3.3.1 (e) Elevon

The elevon contribution to total vehicle lift at zero deflection is ignored. For large elevon deflections, the elevon lift contribution may become significant and is assumed to be given by

$$\Delta C_{L_e} = C_{L_{\delta_e}} \delta e \tag{4.3.16}$$

where the elevon effectiveness term is obtained by the methods given in Section IV of Reference 1.

4.3.3.1(f) Wing-Body Configurations

The lift of a composite configuration below wing stall is calculated as the sum of the component lift contributions. For a wing-body-tail configuration

$$C_{L_{WB}} = C_{L_{\alpha_{WB}}} \alpha' + \Delta C_{L_H} \tag{4.3.17}$$

$$C_{L_{\alpha_{WB}}} = C_{L_{\alpha_B}} + C_{L_{\alpha_W}} + C_{L_{\alpha_H}} \tag{4.3.18}$$

$$\alpha' = \alpha - \alpha_{LO_{WB}} \tag{4.3.19}$$

At angles of attack above wing stall where the body and tail continue to lift, lift is calculated as the sum of the wing lift at stall plus body and horizontal tail lift. Based on test data for the vehicle class considered, the wing stall angle of attack is established at eight degrees. Lift above the stall coefficient is computed from

$$C_{L_{WB}} = \left(C_{L_{\alpha_B}} + C_{L_{\alpha_H}}\right)\alpha' + C_{L_{\alpha_W}}\left(.13963 - \alpha_{LO_{WB}}\right) + \Delta C_{L_H} \qquad (4.3.20)$$

4.3.3.2 Subsonic Drag

Subsonic total drag coefficient of a configuration is the sum of minimum drag, drag due to lift, and trim drag,

$$C_D = C_{D_{MIN}} + C_{D_L} + C_{D_T} \qquad (4.3.21)$$

4.3.3.2(a) Minimum Drag

Subsonic minimum drag of a configuration is computed as the sum of friction and pressure drag of each component plus body base drag,

$$C_{D_{MIN}} = \Sigma \Delta C_{D_{MIN}} + C_{D_b} \qquad (4.3.22)$$

4.3.2(b) Body Friction and Pressure Drag

Body friction and forebody pressure drag contribution to subsonic minimum drag is computed by the method discussed in subsection 4.2.3 of DATCOM. Assuming the equivalent body fineness ratio is greater than four

$$\Delta C_{D_{MIN_B}} = 1.02\, C_f \, \frac{S_{WET}}{S} \qquad (4.3.23)$$

Compressible mean flat plate friction coefficient, C_f, is given by

$$C_f = C_{f_i} \left[C_f / C_{f_i}\right] \qquad (4.3.24)$$

$$C_{f_i} = 0.455 / \left(\log_{10} RN\right)^{2.58} \qquad (4.3.25)$$

Reynolds number to calculate C_{f_i} is either

(a) Reynolds number based on body length

(b) limiting Reynolds number based on admissible surface roughness

Both Reynolds numbers are computed, and the lowest value is used to determine C_{f_i}. Reynolds number based on length is obtained from

$$RN = \frac{a}{\nu}(M \ell_B) \tag{4.3.26}$$

where a = speed of sound, a function of altitude
ν = kinematic viscosity, a function of altitude
M = free stream Mach number
ℓ_B = body length

The limiting Reynolds number is computed from

$$RN_{lim} = K_1 \left(\frac{\ell_B}{k}\right)^{1.0489} \tag{4.3.27}$$

where K_1 is a function of Mach number, Reference 6.

Mach No.	K1
0	37.587
1	49.320
2	91.692
3	189.640

and k is the average surface roughness height. For the surface of a modern aircraft, a k of .0003 inches is a realistic value. For a body that has experienced the heat of reentry, k is much greater.

4.3.3.2(c) Exposed Surface Friction and Pressure Drag

Friction and pressure drag contribution of exposed surfaces is computed by the method contained in subsection 4.2 of the DATCOM.

$$\Delta C_{D_{MIN}} = 2C_f(1+2\,t/c)\frac{S_E}{S} \tag{4.3.28}$$

where C_f is determined as above with length being replaced by the mean geometric chord in determining Reynolds number

$$\bar{C} = \frac{2}{3}\left[C_{R_E} + C_T - \frac{C_{R_E} \times C_T}{C_{R_E} + C_T}\right] \tag{4.3.29}$$

4.3.3.2(d) Base Drag

The body base drag contribution to minimum drag is computed by the method reported in subsection 4.8.2 of DATCOM

$$C_{D_b} = -C_{P_{B_0}} \frac{S_b}{S} \tag{4.3.30}$$

$$C_{P_{B_0}} = \left(\frac{Ⓟ}{2\sqrt{\pi S_b}}\right)\left[C_{P_{B_0}} \Big/ \left(Ⓟ/2\sqrt{\pi S_b}\right)\right] \tag{4.3.31}$$

where S_b and the required base geometry are obtained from the input. The term $C_{P_{B_0}}/(Ⓟ/2\sqrt{\pi S_b})$ is obtained by table lookup of the information presented in Figure (4.8.2.1-7) of DATCOM.

4.3.3.3 Subsonic Drag Due to Lift

4.3.3.3(a) Subsonic Lifting Body Configurations

The method presented in subsection 4.2.3.2 of DATCOM is used to calculate drag due to lift of *modified bodies of revolution*. Body suction forces are neglected by this method and

$$C_{D_L} = C_L \alpha' \tag{4.3.32}$$

According to test data a *delta planform lifting body configuration* possesses a linear lift-curve slope and a parabolic variation of drag with lift. The drag due to lift is expressed as

$$C_{D_L} = K(C_L - \Delta C_L)^2 \tag{4.3.33}$$

where K is the drag due to lift factor calculated from

$$K = 1/C_{L_\alpha} \tag{4.3.34}$$

and ΔC_L is the polar displacement.

4.3.3.(b) Subsonic Wing-Body Configurations

Test data indicates that the drag polar of wing-body configuration breaks at a lower lift than that at which wing stall occurs. This break in the polar is quite pronounced and occurs near an angle of attack of five degrees for both body types considered by TREND. The C_L where drag polar break occurs is calculated by

$$C_{L_{BR}} = C_{L_\alpha}(.08727 - \alpha_{L0_{WB}}) \tag{4.3.35}$$

Drag due to lift depends on whether the operating C_L is above or below the polar break, C_L. At C_L's below $C_{L_{BR}}$

$$C_{D_L} = K(C_L - \Delta C_{L_{WB}})^2 \tag{4.3.36}$$

The polar displacement parameter, $C_{L_{WB}}$, is arrived at by engineering judgment of the effectiveness of body and wing camber. The drag polar shape factor, K, is obtained from

$$K = (1/\pi AR_T e)\frac{S}{S_T} \tag{4.3.37}$$

$$e = e_w\left[1.0 - (1.18 - .681\, C_T/C_{R_T})\frac{W}{b}\right] \tag{4.3.38}$$

$$e_w = \frac{C_{L_\alpha}}{AR_T}\left[\frac{1}{\pi - .85(\pi - C_{L_\alpha}/AR_T)}\right] \tag{4.3.39}$$

The *span efficiency factor*, e_w, is determined by the method reported in Reference 6. A leading edge suction factor of .85 is used. The *configuration efficiency factor* is obtained by modifying the wing alone factor for body effects, Reference 6. The theoretical wing aspect ratio is obtained by extending the wing to the centerline,

$$AR_T = \frac{b^2}{S_T} \tag{4.3.40}$$

where

$$S_T = \frac{b}{2}(C_{R_T} + C_T) \tag{4.3.41}$$

Drag due to lift at C_L's above $C_{L_{BR}}$ is computed from an empirical expression derived from test data. The expression used to compute drag due to lift above polar break is

$$C_{D_L} = K_{BR}(C_L - \Delta C_L)^2 \tag{4.3.42}$$

$$K_{BR} = K + \left[1.26\left(\frac{S}{S_T}\right)^{1.5}(C_L - C_{L_{BR}})\right]^2 \tag{4.3.43}$$

4.3.3.4 Subsonic Trim Drag

Due to the many possible combinations of pitch control surfaces that could be used, a simple method is employed to account for trim drag. This method relies on the user's knowledge of control surface effectiveness and static margin of the particular configuration being worked. The approach is to modify the polar shape factor by an input trim drag factor, F_{TD}. Suggested values of F_{TD} as a function of static margin and moment arm are contained in Figure 4.3-3.

$$C_{D_T} = C_{D_L}(F_{TD} - 1.0) \tag{4.3.44}$$

$$K_T = K \cdot F_{TD} \tag{4.3.45}$$

4.3.3.5 Subsonic Maximum Lift to Drag Ratio

4.3.3.5(a) Subsonic Lifting-Body Configurations

$(L/D)_{max}$ of the *modified body of revolution* configuration is computed by an iterative procedure. Total lift and drag at increasing angles of attack are sequentially computed until a maximum value is obtained.

$(L/D)_{max}$ of the *planar lifting body configuration* is computed using

$$(L/D)_{MAX} = \frac{1}{2K_T} \quad \frac{1}{C_{L_{(L/D)_{MAX}}} - \Delta C_L} \tag{4.3.46}$$

$$C_{L_{(L/D)_{MAX}}} = \sqrt{(C_{D_{MIN}}/K_T) + \Delta C_L^2} \tag{4.3.47}$$

where K_T is the trimmed value. The C_L for polar break is much higher than C_L for $(L/D)_{max}$ for this type of configuration; therefore, polar break is not considered in calculating $(L/D)_{max}$.

4.3.3.5(b) Subsonic Wing-Body Configurations

The method of determining $(L/D)_{max}$ for the wing-body configurations combines the above two methods because C_L for $(L/D)_{max}$ is near the C_L for polar break. C_L for $(L/D)_{max}$ is computed and compared with the C_L for the polar break. If it is less than the C_L for polar break, equation (4.3.46) is used to calculate $(L/D)_{max}$. If it is greater, then the iteration procedure is used above C_L for polar break.

4.3.4 Supersonic Flight Regime

4.3.4.1 Supersonic Lift

4.3.4.1(a) Supersonic Bodies

The lift generated by modified bodies of revolution is computed by the methods presented in Section 4.3.3.1(a) of the subsonic discussion. The potential normal force curve slope, $(C_{N_\alpha})_b$, is obtained by table lookup of the information in Figure (4.2.1.1-7a) of DATCOM for basic ogive cylinder bodies of infinite fineness ratio and $(k_2 - k_1)$ is equal to unity because the supersonic values of $(C_{N_\alpha})_b$ are semiempirical.

The lift generated by planar lifting bodies is again assumed to vary linearly with angle of attack below C_L for $(L/D)_{max}$. This linear lift curve slope is obtained from the wing linear theory calculation results reported in Figure III.A.1-1a of Reference 6 as a function to trailing edge cutout, leading edge sweep, taper ratio, and Mach number.

4.3.4.1(b) Supersonic Wing and Horizontal Tail

The lift contributions of the wing and horizontal tail exposed panels are computed by the methods discussed in subsections 4.3.3.1(c) and (d). The basic lift-curve slopes that are modified for carryover, dynamic pressure reduction and downwash are obtained from the linear theory calculation results reported in Figure III.A.1-1a of Reference 6 based on the surface area formed by joining the exposed panels.

4.3.4.1(c) Supersonic Wing-Body Configurations

Tests on wing-body configurations of the type analyzed by program TREND do not show a reduction of lift-curve slope up to angles of attack of 12 degrees. This angle is considerably above the angle of attack for $(L/D)_{max}$. For this reason lift above wing stall is not computed as in the subsonic case. At all angles of attack lift is computed by the method discussed in subsection 4.3.3.1(f) for the attached flow case.

4.3.4.2 Supersonic Drag

The supersonic drag of a configuration is computed as the sum of minimum drag, drag due to lift, and trim drag,

$$C_D = C_{D_{MIN}} + C_{D_L} + C_{D_T} \qquad (4.3.21)$$

Drag due to lift beyond polar break is not accounted for in the supersonic flight regime because the polar break C_L is much greater than C_L for $(L/D)_{max}$.

4.3.4.2(a) Supersonic Minimum Drag

The minimum drag of a supersonic configuration is calculated as the sum of the component zero lift wave drag, friction drag, and base drag. The friction and base drag of a component at supersonic Mach numbers are usually treated as isolated parts, but the problem of estimating the zero lift wave drag of a composite configuration is one of properly accounting for the mutual interferences that exist between its components. The supersonic area rule program described in Section 4.2 uses that approach. The approach of program TREND is to isolate the individual components and their interferences. In general, the determination of individual interferences has been unsuccessful. However, for the purpose of program TREND in which the sensitivity of drag to configuration changes is desired, simplicity is required, and the isolated component buildup method which neglects any interference used.

4.3.4.2(b) Supersonic Zero Lift Wave Drag

Body wave drag is computed using the empirical data contained in Figure III.B.10-7 of Reference 6 for ogive bodies of revolution. A table lookup procedure is used to determine the forebody and afterbody contributions

$$C_{D_{W_B}} = \left(C_{D_{P_N}} + C_{D_{P_{BT}}}\right)\frac{A_{YZ}}{S} \tag{4.3.48}$$

where

C_{DP_N} = function of nose fineness ratio and β

$C_{DP_{BT}}$ = function of afterbody fineness ratio, β, and the ratio of maximum diameter to base diameter

The required parameters are determined from an ogive body of revolution that approximates the cross section area distribution of the body to be analyzed. The input quantities needed are Mach number, maximum cross-sectional area, base area, distance from nose to maximum cross section, and distance from maximum cross section to base.

Zero lift wave drag of a wing, horizontal tail, or vertical tail is computed by the methods reported in Reference 6 for round leading-edge airfoils provided that $\beta \cot(\lambda_{.5C})$ is less than or equal to 3.5.

$$C_{D_W} = \left[\beta C_{D_W}/f(a,\lambda)(t/c)^2\right]\left[f(a,\lambda)(t/c)^2/\beta\right]\frac{S_P}{S} \quad (4.3.49)$$

$$\beta C_{D_W}/f(a,\lambda)(t/c)^2 = 0.768\,\beta\,\mathrm{COT}\Lambda_{.5C} + 0.0718(\beta\,\mathrm{COT}\Lambda_{.5C})^2 \quad (4.3.50)$$

The term f(a, λ) is obtained from Figure III.B.10-1 of Reference 6 and the average thickness ratio (t/c) is obtained from the input. The wave drag computed by this method contains wing-alone plus body interference drag because the correlation was achieved by subtracting body-alone wave drag from wing-body wave drag test data.

In the case where the parameter, $\beta \cot\lambda_{.5C}$ exceeds 3.5, an alternate method is included based upon the information contained in subsection 4.1.5.1-18 of DATCOM.

$$C_{D_W} = \left\{7.85\,\mathrm{COS}^2\Lambda_{.25C} - [2.05(M-1)][1-\mathrm{TAN}\Lambda_{.25C}]\right\}(t/c)^2\frac{S_P}{S} \quad (4.3.51)$$

4.3.4.2(c) Supersonic Base Drag

Base drag of the bodies is calculated from

$$C_{D_b} = -C_{P_b}\frac{S_b}{S} \quad (4.3.52)$$

$$C_{P_b} = -.475 + .2M - .025M^2 \quad (4.3.53)$$

The expression for the base pressure coefficient was derived from recent base pressure test data. This value is significantly ~~higher than~~ values predicted by the methods reported in Reference 7.

4.3.4.2(d) Supersonic Friction Drag

The total vehicle friction drag is computed as the sum of the friction drag of each component

$$C_{D_F} = \Sigma\Delta C_{D_F} \quad (4.3.54)$$

$$\Delta C_{D_F} = C_f\frac{S_{WET}}{S} \quad (4.3.55)$$

where C_f is computed by the methods discussed in subsection 4.3.3.2(b) of this report, S_{wet} for bodies is obtained from the input, and S_{wet} for surfaces is computed by multiplying the exposed surface planform area by 2.03.

4.3.4.3 Supersonic Drag Due to Lift

4.3.4.3(a) Supersonic Lifting Body Configurations

The drag due to lift of modified body of revolution types of configurations with control surfaces is calculated by the method presented in 4.3.3.3 of the subsonic section

$$C_{D_L} = C_L \alpha' \tag{4.3.56}$$

Drag due to lift of the planar lifting systems that possess linear lift curve slopes and parabolic drag polars is calculated from

$$C_{D_L} = K(C_L - \Delta C_L)^2 \tag{4.3.57}$$

$$K = 1/C_{L_\alpha} \tag{4.3.58}$$

where ΔC_L is the drag polar lift coefficient displacement obtained from the input, and C_{L_α} is calculated as discussed in subsection 4.3.4.1(c). The drag due to lift factor corresponds to zero leading edge suction and is based on recent test data that showed that although some suction is obtained at low supersonic Mach numbers it is quickly lost as Mach number increases.

4.3.4.3(b) Supersonic Wing-Body Configurations

Drag due to lift of bodies with variable sweep wings is computed using Equation 4.3.57. The polar displacement, $\overline{\Delta C_L}$, is obtained from input, and K for subsonic leading edge conditions is calculated from an expression derived from the correlation of supersonic test data in the WINSTAN program of Reference 8. At Mach numbers where the outer wing panels have supersonic leading edges, the drag due to lift factor is equal to the reciprocal of the lift curve slope,

$$\beta \cot \Lambda \geq 1, \quad K = 1/C_{L_\alpha} \tag{4.3.59}$$

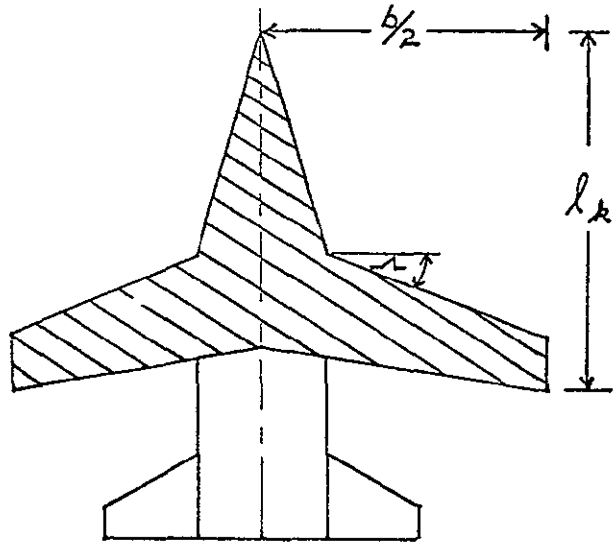
$$\beta \cot \Lambda < 1, \quad K = \frac{P+1}{P(\pi+4)AR}\left[1 + 2\beta^2\left(\frac{b/2}{\ell_A}\right)^2\right] \tag{4.3.60}$$

where

$$P = S_A/b\,\ell_A \tag{4.3.61}$$

$$AR = b^2/S_A \tag{4.3.62}$$

S_k is the shaded area in the sketch below.



4.3.4.4 Supersonic Trim Drag

Trim drag is accounted for by modifying the untrimmed polar shape factor by a trim drag factor obtained from the input as described in subsection 4.3.3.4 of the subsonic discussion.

4.3.4.5 Supersonic Maximum Lift to Drag Ratio

$(L/D)_{max}$ for modified body of revolution configurations without wings is calculated by an iteration procedure discussed in subsection 4.3.3.5 of the subsonic discussion. $(L/D)_{max}$ for all other configurations is computed using equations 4.3.46 and 4.3.47. $(L/D)_{max}$ beyond polar break is not considered because the C_L for polar break is larger than the C_L for $(L/D)_{max}$.

4.3.5 Hypersonic Flight Regime

The analytical expressions of Hankey and Alexander for normal and axial force are used. This permits analytic sensitivity derivatives (partial derivatives) to be included in the computer procedure. These techniques were selected because

1. They are applicable for a wide range of flight and geometric parameters
2. They are derived from simple theories that are modified by empirical relations to improve accuracy and applicability.

A vehicle to be analyzed is defined by the following generalized configuration components:

1. hemispherical nose cap
2. lower flat surface
3. ramp lower flat surface
4. lower flat and ramp cylindrical leading edges
5. flat fin surface with a straight swept leading edge
6. fuselage composed of no more than two truncated cones (or a cone-cylinder combination
7. elevon.

4.3.5.1 Hypersonic Lift and Drag

The equations used in program TREND are listed below.

$$C_L = C_N \cos\alpha - C_A \sin\alpha \tag{4.3.63}$$

$$C_D = C_N \sin\alpha + C_A \cos\alpha \tag{4.3.64}$$

$$C_N = C_{N_N} + C_{N_L} + C_{N_{LE}} + C_{N_{L_R}} + C_{N_{LE_R}} + C_{N_e} + C_{N_{LE_F}} + C_{N_{B_1}} + C_{N_{B_2}} \tag{4.3.65}$$

$$C_A = C_{A_N} + C_{A_{L\ \ell\, or\, t}} + C_{A_{LE}} + C_{A_{L_R}} + C_{A_{LE_R}} + C_{A_e} + C_{A_{LE_F}} + C_{A_{B_1}} + C_{A_{B_2}} \tag{4.3.66}$$

$$C_{N_N} = F_N K_N \sin\alpha\,(1+\cos\alpha) \tag{4.3.67}$$

$$C_{N_L} = K_L\, S_L/S\ \sin^2\alpha \tag{4.3.68}$$

$$C_{N_{LE}} = F_{LE} K_{LE} \sin\alpha\left[\cos(\Lambda_{LE})_E + \cos\Lambda_{LE}\cos\alpha\right] \tag{4.3.69}$$

$$C_{N_{L_R}} = K_{L_R} S_{L_R}/S\ \sin^2(\alpha+\Delta_R)\cos\Delta_R \tag{4.3.70}$$

$$C_{N_{LE_R}} = F_{LE_R} K_{LE} \sin(\alpha+\Delta_R)\cos\Delta_R\left[\cos(\Lambda_{LE_R})_E + \cos\Lambda_{LE_R}\cos(\alpha+\Delta_R)\right] \tag{4.3.71}$$

$$C_{N_e} = K_e\, S_e/S\ \sin^2(\alpha+\Delta_e)\cos\Delta_e \tag{4.3.72}$$

$$C_{N_{LE_F}} = -F_{LE_F} K_{LE} \sin^2(\Gamma-\alpha)\cos\Gamma \tag{4.3.73}$$

$$C_{N_{B_1}} = -0.125\ F_{B_1}\frac{A_{xy_1}}{S} \tag{4.3.74}$$

$$C_{N_{B_2}} = -0.125\ F_{B_2}\frac{A_{xy_2}}{S} \tag{4.3.75}$$

$$C_{A_N} = \tfrac{1}{2} F_N K_N (1+\cos\alpha)^2 \tag{4.3.76}$$

$$C_{A_{L_\ell}} = \frac{G_\ell}{\sqrt{RN}}\ \frac{S_{WET}}{S}\left[0.45\cos\alpha + 4.65\frac{V_\infty}{10{,}000}\sin\alpha\cos^{2.2}\alpha\right] \tag{4.3.77}$$

$$C_{A_{L_t}} = \frac{G_t}{(RN)^{.2}}\ \frac{S_{WET}}{S}\left[0.048\sin(4.5\alpha) + 0.7\frac{V_\infty}{10{,}000}\sin^{1.5}\alpha\cos^{2.25}\alpha\right] \tag{4.3.78}$$

$$C_{A_{LE}} = \tfrac{1}{2} F_{LE} K_{LE}\cos\Lambda_{LE}\left[\cos(\Lambda_{LE})_E + \cos\Lambda_{LE}\cos\alpha\right]^2 \tag{4.3.79}$$

$$C_{A_{L_R}} = K_{L_R} S_{L_R}/S \; Sin^2(\alpha + \Delta_R) \; Sin \; \Delta_R \tag{4.3.80}$$

$$C_{A_{LE_R}} = \tfrac{1}{2} F_{LE_R} K_{LE} Cos \, \Lambda_{LE_R} \; Cos \, \Delta_R \left[Cos(\Lambda_{LE_R})_E + Cos \, \Lambda_{LE_R} Cos(\alpha + \Delta_R) \right]^2 \tag{4.3.81}$$

$$C_{A_e} = K_e S_e/S \; Sin^2(\alpha + \Delta_e) \; Sin \; \Delta_e \tag{4.3.82}$$

$$C_{A_{LE_F}} = F_{LE_F} K_{LE} Sin^2(\Gamma - \alpha) \; Sin \; \Gamma \tag{4.3.83}$$

$$C_{A_{B_1}} = 0.196 \, F_{B_3} \frac{A_{xy_1}}{S} \, Tan \, \Delta_1 \tag{4.2.84a}$$

$$C_{A_{B_2}} = 0.196 \, F_{B_4} \frac{A_{xy_2}}{S} \, Tan \, \Delta_2 \tag{4.3.84b}$$

$$F_N = \pi R_N^2 / 4S \tag{4.3.86}$$

$$F_{LE} = 4 R_{LE} \, l_{LE} / 3S \tag{4.3.87}$$

$$F_{LE_R} = 4 R_{LE_R} \, l_{LE_R} / 3S \tag{4.3.88}$$

$$F_{LE_F} = 4 R_{LE_F} \, l_F / 3S \tag{4.3.89}$$

$$F_{B_1} = \eta_c (1 + 0.845 \Delta_1 - 0.663 \alpha)^5 - 22.93/M^2 \tag{4.3.90}$$

$$F_{B_2} = \eta_c (1 + 0.845 \Delta_2 - 0.663 \alpha)^5 - 22.93/M^2 \tag{4.3.91}$$

$$F_{B_3} = \eta_c (1 + 0.845 \Delta_1 - 0.537 \alpha)^5 - 22.93/M^2 \tag{4.3.92}$$

$$F_{B_4} = \eta_c (1 + 0.845 \Delta_2 - 0.537 \alpha)^5 - 22.93/M^2 \tag{4.3.93}$$

$$RN = \frac{V_\infty l}{\nu} \tag{4.3.94}$$

$$Cos(\Lambda_{LE})_E = \left[1 + Sin^2 \Lambda_{LE} \; Cos^2 \alpha \right]^{1/2} \tag{4.3.95}$$

$$Cos(\Lambda_{LE_R})_E = \left[1 + Sin^2 \Lambda_{LE_R} \; Cos^2(\alpha + \Delta_R) \right] \tag{4.3.96}$$

The following factors are obtained from the input and can be varied: K_N, K_{LE}, G_ℓ, G_t, and η_c.

4.3.6 Stability and Control Sensitivity Analysis

Program TREND provides an approximation to vehicle lift, drag, and aerodynamic center, and the sensitivity of lift, drag, and aerodynamic center to certain aerodynamic and geometric parameters. These sensitivities are obtained in the general form $\partial F/\partial V$ where F is lift, drag, or aerodynamic center or a component lift, drag, or aerodynamic center. The variable V may be angle of attack or any of a variety of vehicle geometric characteristics.

At subsonic and supersonic speeds, the sensitivities are obtained by numerical differentiation of the basic lift and drag equation presented in subsections 4.3.4 and 4.3.5 and related aerodynamic center equations. At hypersonic speeds the Hankey and Alexander formulation permits development of closed form sensitivities.

The aerodynamic characteristics and geometric parameters selected for sensitivity factors are summarized in Tables 4.3-2, 4.3-3, and 4.3-4 for each speed regime. It may be anticipated that certain of the geometric parameters would have negligible influence on some of the ~~stability~~ and control derivatives for certain vehicle configurations. To insure scope of program coverage, however, most major geometric parameters have been included as sensitivity factors. The program can be used, therefore, to determine which geometric changes have the greater influence on the stability and control derivatives and those which have little influence.

A complete discussion of the equations supporting the stability and control portion of program TREND are given in their entirety in the original TREND documentation for subsonic, supersonic, and hypersonic speed regimes. As noted previously, the prediction techniques at subsonic and supersonic speeds are based primarily on the DATCOM while the technique at hypersonic speed is based on the Hankey and Alexander method. In some instances, alterations of the DATCOM method or utilization of other reference material is necessary to adapt the prediction scheme to the intended class of configurations. These deviations are listed below as follows:

1. The terms e_I and e_{II} were included in the body predictions of sideforce at subsonic and supersonic speeds for configurations I and II, respectively, to provide latitude to correct for body cross-sectional shapes. Representative values of these parameters is discussed in subsection 4.2.1.2 of DATCOM.

2. Effect of vertical tail endplating on the aerodynamic center of configuration II is included at subsonic speeds. Necessary parameters for calculating this effect are obtained from the lift and drag section of the program.

3. Configuration II aerodynamic center characteristics are more indicative of a cranked wing configuration than a wing-body combination at subsonic speeds. Therefore, the technology obtained from the F-111 airplane and from the WINSTAN studies, Reference 8, is utilized for this prediction.

4. To reduce procedure complexity and, at the same time maintain an acceptable degree of accuracy, the center of pressure of the horizontal tail and of the elevon has been assumed to be located at the centroid of the surface at supersonic speeds.

5. Additional references required to support the DATCOM techniques for purposes of completeness and ease of handling include those of References 5, 6, and 9.

6. Subsonic elevon effectiveness for configuration II is based on the method of Reference 10.

4.3.7 Aerothermodynamic Techniques in Hypersonic Flight

The trend program contains a variety of aerothermodynamic analysis modules. Correlations have been developed which describe the aerodynamic heating to the five major regions of the vehicle listed below:

1. Nose cap stagnation point
2. Nose cap lower surface interaction point
3. Leading edge stagnation line
4. Leading edge lower surface interaction line
5. Lower surface centerline (fuselage)

The options available are described in detail in section on thermodynamics.

REFERENCES:

1. Fox, M. K., Barnes, K. H., Harrington, L. J., Mauzy, E. L., et al., Investigation of Techniques to Evaluate Design Trade-Offs in Lifting Reentry Vehicles, Volume I, AFFDL-TR-66-77, October 1966.

2. Phillips, W. Pelham; Decker, John P.; Rau, Timothy R.; and Glatt, C. R.: Computer-Aided Space Shuttle Orbiter Wing Design Study, NASA TN D-7478, May 1974.

3. Hoak, D. E. and Carlson, J. W.: Air Force Project Engineers for U. S. Air Force Stability and Control DATCOM, Section 4.4

4. Jorgensen, L. H., Inclined Bodies of Various Cross Sections at Supersonic Speeds," NASA Memo 10-3-58A, November 1958.

5. Pitts, W. C., Nielsen, J. N., and Kaattari, G. E., Lift and Center of Pressure of Wing-Body-Tail Combinations at Subsonic, Transonic, and Supersonic Speeds, NACA Report 1307, 1959.

6. Frost, R. C., Editor, GD/FW Aerospace Handbook, General Dynamics, Fort Worth Report FZA-381, November 1963.

7. Hargis, C. B., Jr., Davison, P. H., and Savage, S. B., Methods for Estimating Base Pressures on Aircraft Configurations, WADC TN 57-28, Wright Air Development Center, Wright-Patterson Air Force Base, July 1957.

8. ———., Aerodynamic Characteristics of Nonstraight-Taper Wings, Air Force Contract 33(615)-2520.

9. Kaattari, G. E., Estimation of Directional Stability Derivatives at Moderate Angles and Supersonic Speeds, NASA Memo 12-1-58A, 1959.

10. Kouri, B. G., Estimation of Longitudinal Center of Pressure for Flap-Type Control Surfaces on Wings of Various Planforms, General Dynamics Convair Division, Report IAR 3.157, December 1959.

TABLE 4.3-1. CONFIGURATION TYPES

TYPE	FLIGHT REGIME
I - Modified body of revolution with horizontal and vertical tails	Subsonic, supersonic, and hypersonic
IB - Type I with a wing	Subsonic and supersonic
II - Planar lifting body with elevons and vertical tail	Subsonic, supersonic, and hypersonic
IIB - Type II with a wing	Subsonic and supersonic

TABLE 4.3-2. AVAILABLE SUBSONIC SENSITIVITIES IN PROGRAM TREND

CONFIGURATION I

INPUT VARIABLE	STABILITY AND CONTROL DERIVATIVE													
	A.C.		C_{m_α}		C_{Y_β}			C_{n_β}			C_{ℓ_β}			$C_{m_{\delta_e}}$
	BODY	WING-BODY-TAIL	BODY	WING-BODY-TAIL	BODY	WING-BODY	VERTICAL TAIL	BODY	WING-BODY	VERTICAL TAIL	BODY	WING-BODY	VERTICAL TAIL	HORIZONTAL TAIL
b_V							X			X			X	
b_W		X		X		X			X			X		
b_H		X		X										X
c_{r_V}							X			X			X	
c_{r_W}		X		X		X			X			X		
c_{r_H}		X		X										X
Γ						X						X		
h_W						X						X		
ℓ_H	X	X	X	X				X	X					
λ_V							X			X			X	
Λ_{LEV}							X			X			X	
λ_W		X		X		X			X			X		
Λ_{LEW}		X		X		X			X			X		
λ_H		X		X										X
Λ_{LEH}		X		X										X
r_V							X			X			X	
w_W												X		
w_H		X		X										X
z_W						X						X		
c_T					X	X		X	X					
$d\epsilon/d\alpha$		X		X										
q_H/q		X		X										X

CONFIGURATION II

INPUT VARIABLE	STABILITY AND CONTROL DERIVATIVE													
	A.C.		C_{m_α}		C_{Y_β}			C_{n_β}			C_{ℓ_β}			$C_{m_{\delta_e}}$
	BODY	WING-BODY	BODY	WING-BODY	BODY	WING-BODY	VERTICAL TAIL	BODY	WING-BODY	VERTICAL TAIL	BODY	WING-BODY	VERTICAL TAIL	ELEVONS
α											X	X		
b_B	X	X	X	X				X	X		X	X		X
b_V							X			X			X	
b_W		X		X		X			X			X		
c_{r_B}	X	X	X	X				X	X		X	X		
c_{r_V}							X			X			X	
c_{r_W}		X		X		X			X			X		
c_{r_e}														X
c_{r_e}														X
δ_{e_A}					X	X		X	X		X	X		
Γ						X						X		
h_W						X								
λ_B	X	X	X	X				X	X		X	X		
Λ_{LEB}	X	X	X	X				X	X		X	X		X
λ_V							X			X			X	
Λ_{LEV}							X			X			X	
λ_W		X		X		X			X			X		
Λ_{LEW}		X		X		X			X			X		
$x_{\frac{1}{2}LE}$					X	X		X	X		X	X		
S_{BS}/S_{BOD}					X	X		X	X		X	X		
S_{2c_r}/S_{BS}											X	X		
Z_W						X								
$FSLE_e$														X

TABLE 4.3-3. AVAILABLE SUPERSONIC SENSITIVITIES IN PROGRAM TREND

CONFIGURATION I

INPUT VARIABLE	STABILITY AND CONTROL DERIVATIVE													
	A.C.		C_{m_α}		C_{Y_β}			C_{n_β}			C_{l_β}			$C_{m_{\delta e}}$
	BODY	WING-BODY-TAIL	BODY	WING-BODY-TAIL	BODY	WING-BODY	VERTICAL TAIL	BODY	WING-BODY	VERTICAL TAIL	BODY	WING-BODY	VERTICAL TAIL	HORIZONTAL TAIL
b_W		X		X		X			X			X		
b_V							X			X			X	
c_{r_W}		X		X		X			X			X		
c_{r_V}							X			X			X	
C_L					X	X		X	X					
Γ												X		
h_a					X	X		X	X					
h_b					X	X		X	X					
h_c						X			X					
ℓ_N	X	X	X	X	X	X		X	X					
ℓ_A	X	X	X	X	X	X		X	X					
λ_W		X		X								X		
Λ_{LEW}		X		X								X		
λ_V							X			X			X	
Λ_{LEV}							X			X			X	
x_V							X			X			X	
V_A	X	X	X	X										
V_N	X	X	X	X		X			X					
w_W		X		X		X			X			X		
z_W												X		
$FSCP_H$		X		X										X
$d\epsilon/d\alpha$		X		X										
q_H/q		X		X										X

CONFIGURATION II

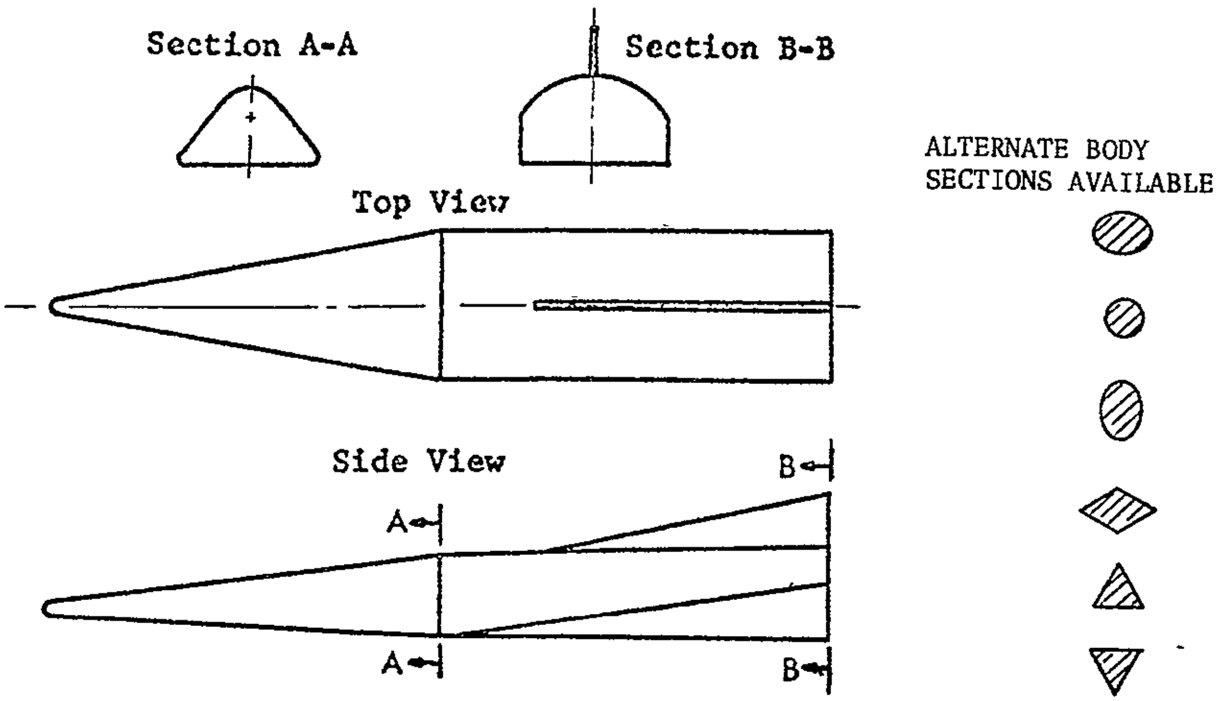
INPUT VARIABLE	STABILITY AND CONTROL DERIVATIVE													
	A.C.		C_{m_α}		C_{Y_β}			C_{n_β}			C_{l_β}			$C_{m_{\delta e}}$
	BODY	WING-BODY	BODY	WING-BODY	BODY	WING-BODY	VERTICAL TAIL	BODY	WING-BODY	VERTICAL TAIL	BODY	WING-BODY	VERTICAL TAIL	ELEVONS
b_B											X	X		
b_W		X		X		X			X			X		
b_V							X			X			X	
c_{r_B}											X	X		
c_{r_W}		X		X		X			X			X		
c_{r_V}							X			X			X	
c_{r_e}														X
c_{t_e}														X
Γ											X	X		
C_L					X	X		X	X					
h_a					X	X		X	X		X	X		
h_N					X	X		X	X		X	X		
h_V						X			X			X		
ℓ_N	X	X	X	X	X	X		X	X		X	X		
ℓ_A	X	X	X	X	X	X		X	X		X	X		
λ_B											X	X		
Λ_{LE_B}											X	X		
λ_W		X		X								X		
Λ_{LE_W}		X		X								X		
λ_V							X			X			X	
Λ_{LEV}							X			X			X	
V_B	X	X	X	X										
V_T	X	X	X	X		X			X					
w_W						X			X			X		
z_W												X		
$FSCP_{\delta e}$														X
b_e														X

TABLE 4.3-4. AVAILABLE HYPERSONIC INPUT SENSITIVITIES IN PROGRAM TREND

STABILITY & CONTROL DERIVATIVE	INPUT VARIABLE / ELEMENT	α	R_N	x	y	z	Λ_{LE}	R_{LE}	ℓ_{LE}	s	Δ	Γ	λ	ϕ	A_{XZ}	A_{XY}
AC	1		X	X												
	2			X			X	X	X							
	3			X						X						
	4			X			X	X	X							
	5			X						X						
	6			X				X	X			X				
	7			X							X					X
	8			X							X					X
C_{m_α}	1	X	X	X		X										
	2	X		X		X	X	X	X							
	3	X		X		X				X						
	4	X		X		X	X	X	X		X					
	5	X		X		X				X	X					
	6	X		X		X		X	X			X				
	7	X		X		X					X					X
	8	X		X		X					X					X
C_{y_β}	1	X	X													
	2	X					X	X	X							
	3	X								X						
	4	X					X	X	X		X					
	5	X								X	X					
	6	X								X			X	X		
	7	X									X				X	
	8	X									X				X	
C_{l_β}	1	X	X			X										
	2	X			X	X	X	X	X							
	3	X				X				X						
	4	X			X	X	X	X	X		X					
	5	X				X				X	X					
	6	X			X	X				X			X	X		
	7	X				X					X				X	
	8	X				X					X				X	
C_{n_β}	1	X	X	X												
	2	X		X	X		X	X	X							
	3	X		X						X						
	4	X		X	X		X	X	X		X					
	5	X		X						X	X					
	6	X		X	X					X			X	X		
	7	X		X							X				X	
	8	X		X							X				X	
$C_{m_{\Delta R}}$	9	X		X		X										
$C_{m_{\Delta E}}$	10	X		X		X										

1 NOSE
2 LOWER SURFACE LEADING EDGE
3 LOWER SURFACE
4 RAMP LEADING EDGE
5 RAMP SURFACE
6 VERTICAL FIN
7 BODY CONE
8 BODY CYLINDER
9 RAMP CONTROL
10 ELEVON CONTROL

CONFIGURATION TYPE I - BODY-ALONE



CONFIGURATION TYPE I - WING-BODY-TAIL

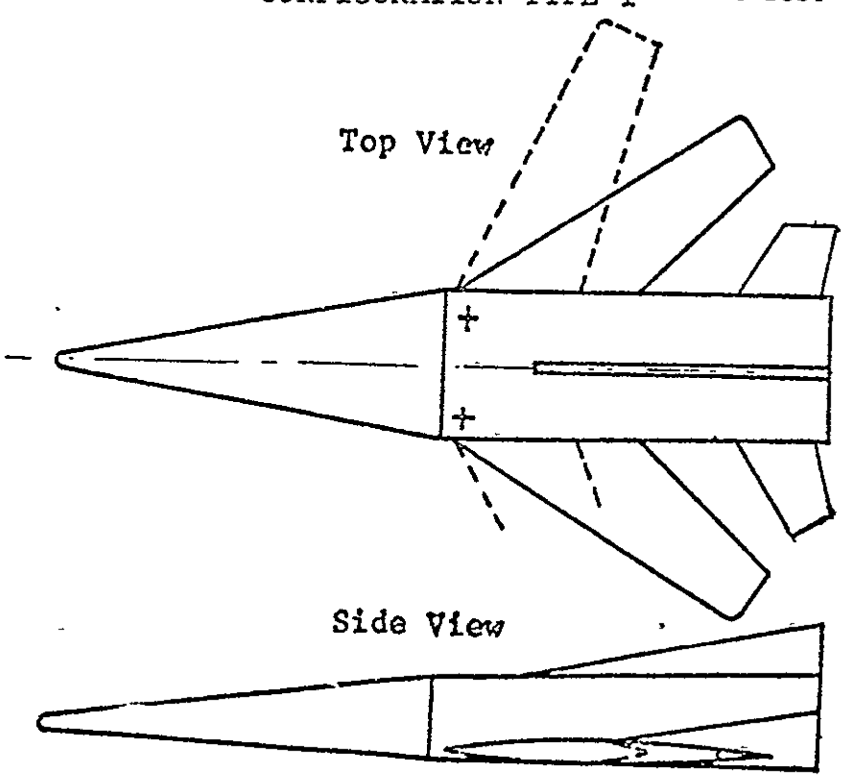
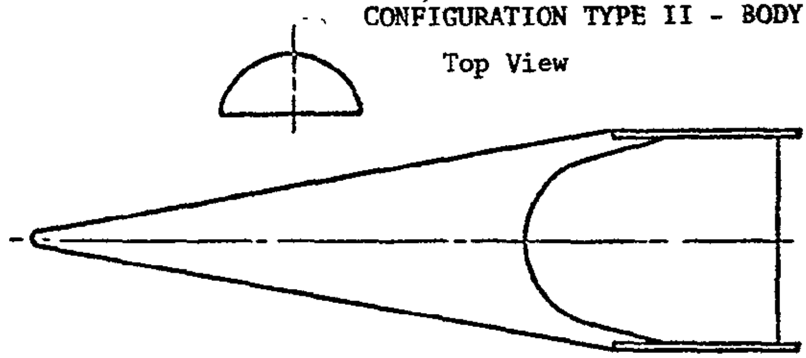


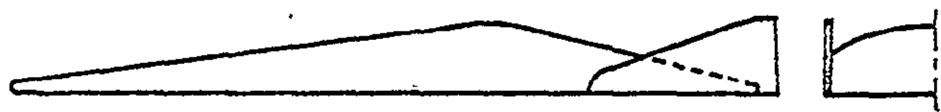
FIGURE 4.3-1. TYPICAL TYPE I CONFIGURATIONS

CONFIGURATION TYPE II - BODY ALONE

Top View

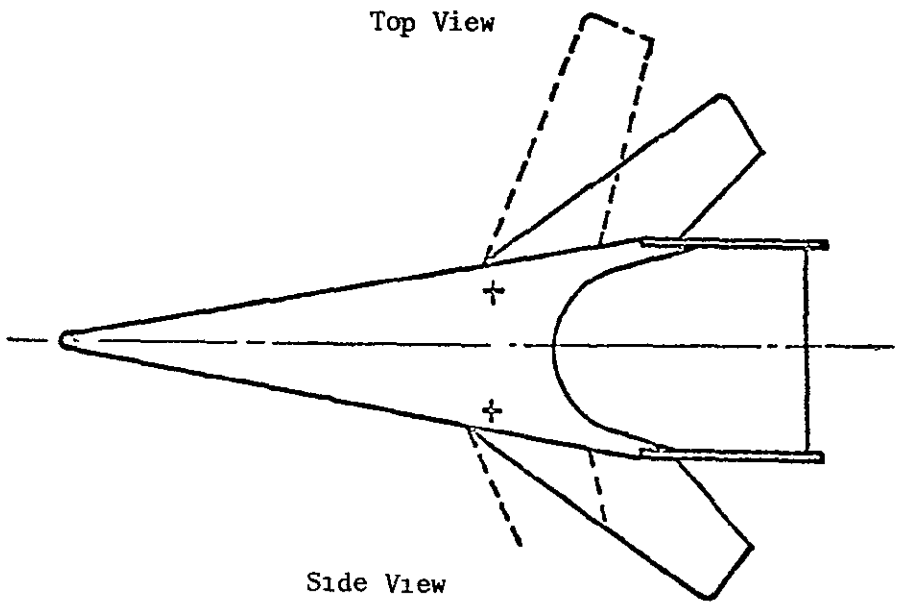


Side View



CONFIGURATION TYPE II - WING-BODY

Top View



Side View

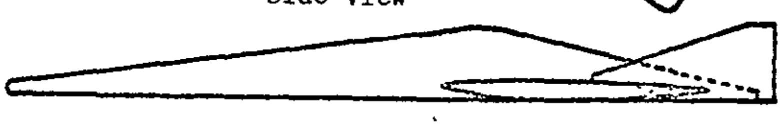


FIGURE 4.3-2. TYPICAL TYPE II CONFIGURATIONS

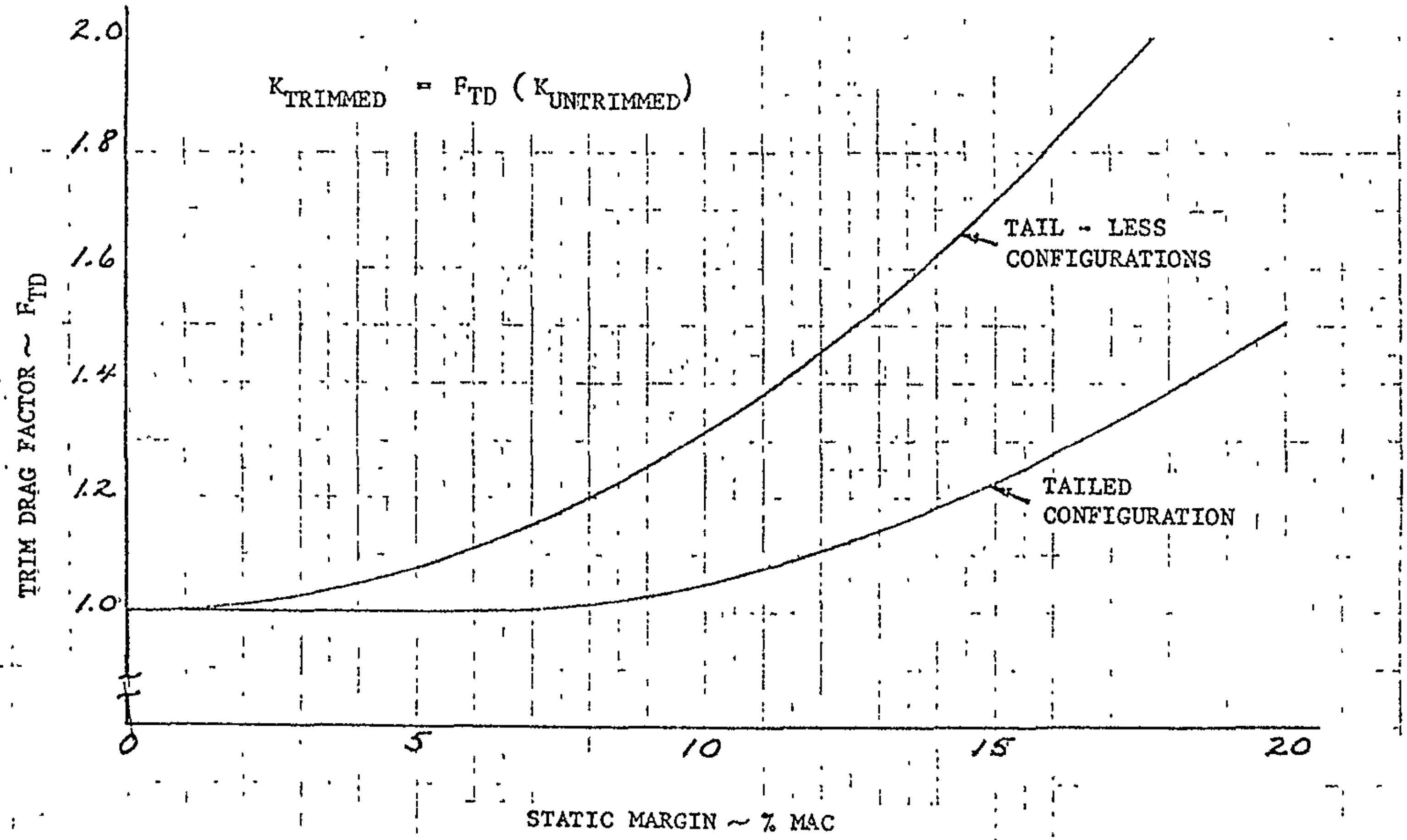


FIGURE 4.3-3 TRIM DRAG CHARACTERISTICS

4.4 USAF STABILITY AND CONTROL DATCOM

The USAF DATCOM is a large scale Air Force Flight Dynamics Laboratory project aimed at the development of a unified and systematic method for predicting vehicle aerodynamic characteristics. The AFFDL project officer is Mr. D. E. Hoak. Over most of DATCOM's development D. E. Ellison of McDonnell-Douglas has served as the contractor's principal investigator. Recently McDonnell-Douglas and TRW Systems have constructed computerized versions of the DATCOM, Reference 1.

Fundamentally, the purpose of the DATCOM (for data compendium) is to provide a systematic summary of methods for estimating basic stability and control derivatives. The DATCOM is organized in such a way that is is self sufficient. For any given flight condition and configuration the complete set of derivatives can be determined without resort to outside information. The book is intended to be used for preliminary design purposes before the acquisition of test data. The use of reliable test data in lieu of the DATCOM is always recommended; however, there are many cases where the DATCOM can be used to advantage in conjunction with test data. For instance, if the lift-curve slope of a wing-body combination is desired, the DATCOM recommends that the lift-curve slopes of the isolated wing and body, respectively, be estimated by methods presented and that appropriate wing-body interference factors (also presented) be applied. If wing-alone test data are available, it is obvious that these test data should be substituted in place of the estimated wing-alone characteristics in determining the lift-curve slope of the combination. Also, if test data are available on a configuration similar to a given configuration, the characteristics of the similar configuration can be corrected to those for the given configuration by judiciously using the DATCOM material.

The various sections of the DATCOM have been numbered with a decimal system which provides the maximum degree of flexibility. A "Section" as referred to in the DATCOM contains information on a single specific item, e.g., wing lift-curve slope. Sections can, in general, be deleted, added, or revised with a minimum disturbance to the remainder of the volume. The numbering system used throughout the DATCOM follows the scheme outlined below:

Section: An orderly decimal system is used consisting of numbers having no more than four digits (see Table of Contents). All sections are listed in the Table of Contents although some consist merely of titles. All sections begin at the top of a right-hand page.

Page: The page number consists of the section number followed by a dash number. (For example, Page 4.1.3.2-4 is the fourth page of Section 4.1.3.2.)

Figures: Figure numbers are the same as the page number. This is a convenient system for referencing purposes. For pages with more than one figure, a lower case letter follows the figure number. Example: Fibure 4.1.3.2-43b is the second figure on Page 4.1.3.2-43. Where a related series of figures appears on more than one page, the figure number is the same as the first page on which the series begins. Example: Figure 4.1.3.2-46d may be found on Page 4.1.3.2-47 and is the fourth in a series of charts. Figures are frequently referred to as "charts" in the text.

Tables: Table numbers consist of the section number followed by an upper case dashed letter. Example: Table 4.1.3.2-A is the first table to appear in Section 4.1.3.2.

Equations: Equation numbers consist of the section number followed by a lower case dashed letter. Example: 4.1.3.2-b is the second equation (of importance) appearing in Section 4.1.3.2. Repeated equations are numbered the same as for the first appearance of the equation but are called out as follows: Equation 4.1.3.2-b.

The major classification of sections in the DATCOM is according to type of stability and control parameter. This classification is summarized below:

Section 1. Guide to DATCOM and Methods Summary (present discussion including the Methods Summary)

Section 2. General information

Section 3. Reserved for future use

Section 4. Characteristics at angle of attack

Section 5. Characteristics in sideslip

Section 6. Characteristics of high-lift and control devices

Section 7. Dynamic derivatives

Section 8. Mass and inertia

Section 9. Characteristics of VTOL-STOL aircraft

The information in Section 2 consists of a complete listing of notation and definitions used in the DATCOM, including the sections in which each symbol is used. It should be noted that definitions are also frequently given in each section where they appear. Insofar as possible, NASA notation has been used. Thus the notation from original source material has frequently been modified for purposes of consistency. Also included in Section 2 is general information used repeatedly by the engineer, such as geometric parameters, airfoil notation, wetted area charts, etc.

Sections 4 and 5 are for configurations with flaps and control surfaces neutral. Flap and control characteristics are given in Section 6 for both symmetric and asymmetric deflections. Section 4 includes effects of engine power and ground plane on the angle of attack parameters.

The DATCOM presents less information on the dynamic derivatives (Section 7) than on the static derivatives, primarily because of the relative scarcity of data, but partly because of the complexities of the theories. Furthermore, the dynamic derivatives are frequently less important than the static derivatives and need not be determined to as great a degree of accuracy. However, the DATCOM does present test data, from over a hundred sources, for a great variety of configurations (Table 7-A).

If more than preliminary design information on mass and inertia (Section 8) is needed, a weights and balance engineer should be consulted.

Section 9 is a unified section covering aerodynamic characteristics of VTOL STOL aircraft, with the exception of ground-effect machines and helicopters. The DATCOM presents less information in this area than that presented for conventional configurations because of the scarcity of data, the complexities of the theories, and the large number of variables involved. In most cases the DATCOM methods of this section are based on theory and/or experimental data such that their use is restricted to first approximations of the aerodynamic characteristics.

In view of the massive documentation required for complete DATCOM documentation, the present section is limited to the above brief outline and the following list of sections available.

4.4.1 DATCOM Summary

Section 1		Guide to DATCOM and Methods Summary
Section 2		General Information
	2.1	General Notation
	2.2	Wing Parameters
	2.2.1	Section Parameters
	2.2.2	Planform Parameters
	2.3	Body Parameters
Section 3		Reserved for Future Use

* Subjects for Future Additions

Section 8 Mass and Inertia
- 8.1 Aircraft Mass and Inertia
- 8.2 Missile Mass and Inertia

Section 9 Characteristics of VTOL-STOL Aircraft
- 9.1 Free Propeller Characteristics
  - 9.1.1 Propeller Thrust Variation with Angle of Attack
  - 9.1.2 Propeller Pitching-Moment Variation with Power and Angle of Attack
  - 9.1.3 Propeller Normal-Force Variation with Power and Angle of Attack
- 9.2 Propeller-Wing Characteristics
  - 9.2.1 Propeller-Wing-Flap Lift Variation with Power and Angle of Attack
  - 9.2.2 *Propeller-Wing-Flap Pitching-Moment Variation with Power and Angle of Attack
  - 9.2.3 Propeller-Wing-Flap Drag Variation with Power and Angle of Attack
- 9.3 Ducted-Propeller Characteristics
  - 9.3.1 Ducted-Propeller Lift Variation with Power and Angle of Attack
  - 9.3.2 Ducted-Propeller Pitching-Moment Variation with Power and Angle of Attack
  - 9.3.3 Ducted-Propeller Drag Variation with Power and Angle of Attack

When used in computer design simulations, the Reference 1 TRW Systems Program is most applicable to reusable launch vehicles. For military aircraft systems, it is recommended that contact be made with the AFFDL Project Officer and the latest USAF computerized version of DATCOM be obtained.

References:

1. ————., Configuration Design Analysis Program (DATCOM), Project Technical Report 20029-H115-RO-00, TRW Systems Applied Mechanics Department and Applied Technology Laboratory, September 1974.

TABLE OF CONTENTS FOR SECTION 5.1, PROGRAM ENCYCL

SECTION 5

PROPULSION

The ODIN/RLV program library contains three independent engine cycle analysis programs. These programs have a combined ability for analysis of single or multi-spool turbojet and turbofan engine cycles with or without afterburners. Two of the programs contain an approximate off-design point analysis capability. All three programs were written at NASA's Lewis Research Center; however, the multi-spool engine programs are derivatives of a U. S. Air Force Aero Propulsion Laboratory, Wright-Patterson Air Force Base engine cycle analysis called SMOTE. Programs are provided for

1. Design-point performance of single spool turbojet and turbofan engines

2. Design and off-design performance of one- and two-spool and turbojet and turbofan engines

3. Design and off-design performance of two- and three-spool turbofan engines

Each program is outlined in the following sections; for complete details, reference should be made to the original source documents.

REFERENCES:

1. Vanco, Michael R., Computer Program for Design-Point Performance of Turbojet and Turbofan Engines, NASA TM X-1340, February 1967.

2. Koenig, Robert W. and Fishback, Laurence H., GENENG - A Program for Calculating Design and Off-Design Performance for Turbojet and Turbofan Engines, NASA TN D-6552, February 1972.

3. Fishback, Laurence H. and Koenig, Robert W., GENENG II - A Program for Calculating Design and Off-Cesign Performance of Two- and Three-Spool Turbofans with as Many as Three Nozzles, NASA TN D-6553, February 1972.

5.1 PROGRAM ENCYCL: COMPUTER PROGRAM FOR DESIGN-POINT PERFORMANCE OF TURBOJET AND TURBOFAN ENGINE CYCLES

Program ENCYCL is a CDC 6600 computer version of the original Reference 1 Vanco program. The program description below is essentially that of the original Vanco document.

5.1.1 Introduction

Advanced aircraft for supersonic flight, high-payload long-range subsonic flight and vertical flight require the development of advanced propulsion systems. Study of these vehicles requires thermodynamic analyses for performance of turbojet and turbofan engine cycles. Program ENCYCL will compute a design point analysis of such engines provided they employ a single spool.

The program requires the following input:

1. airplane Mach number
2. altitude-state conditions
3. turbine-inlet temperature
4. afterburner temperature
5. duct-burner temperature
6. bypass ratio
7. coolant flow
8. component efficiencies
9. component pressure ratios

The thermodynamic properties used are expressed as functions of temperature and fuel to air ratio. The fuel is assumed to be of the form $(CH_2)_n$. Results of the analysis include

1. specific thrust
2. specific fuel consumption
3. engine efficiency
4. several component temperatures and pressures

The equations used in the ENCYCL analysis are presented below and follow Vanco's original report of Reference 1.

5.1.2 Cycle Description

The general engine cycle is shown in Figure 5.1-1. Air enters the diffuser and the major portion of its velocity head is changed into a pressure head. This lower velocity air then enters the fan and is compressed. The air flow then divides into the main flow and the duct flow. The main flow enters the compressor and is further compressed. The major portion of the main flow then enters the combustor and mixes with fuel. Combustion then occurs. The small remaining portion of the main flow bypasses the combustor and is used to cool the turbine. Combustor exit gases are then expanded in the turbine, producing work to drive the compressor and fan. Turbine exit gases and the coolant flow then mix and enter the afterburner with fuel, and further combustion occurs. These hot gases are then expanded in the main nozzle to produce thrust. The duct flow enters the duct-burner with fuel, and combustion occurs. Hot duct gases are then expanded in a nozzle to produce additional thrust.

5.1.3 Derivation of Equations

The equations used in the analysis of the general engine are derived in this section. The thermodynamic properties used for this analysis are functions of temperature and fuel to air ratio. The specific heat of the gas is expressed by a polynomial equation. Appropriate integrations of this equation yield the enthalpy change and the entropy function. The entropy function is herein defined as

$$\Delta\varphi = \int_{T_1}^{T_2} \frac{C_p}{T}\, dT \tag{5.1.1}$$

(All symbols are defined in Section 5.1.5). The derivation of the equations for specific heat, enthalpy change, and entropy function are given in Section 5.1.4. Since the specific heat is a function of temperature and fuel to air ratio, it is expressed as $C_p(T, f)$ in this analysis. Since

$$\Delta h = \int_{T_1}^{T_2} C_p(T, f)\, dT \tag{5.1.2}$$

and

$$\Delta\varphi = \int_{T_1}^{T_2} \frac{C_p(T, f)}{T}\, dT \tag{5.1.3}$$

they are expressed as $\Delta h(T_2, T_1, f)$ and $\Delta\varphi(T_2, T_1, f)$, respectively. When the fuel to air ratio is zero, these quantities will appear with temperatures only. The fuel is assumed to be of the form $(CH_2)_n$.

5.1.3.1 Engine Inlet

The static inlet conditions of the diffuser T_0 and p_0 are a function of the altitude. The inlet velocity is

$$V_0 = M_0 \sqrt{g R_a \gamma T_0} \tag{5.1.4}$$

where M_0 is the Mach number at which the airplane is traveling. The specific heat ratio is

$$\gamma = \frac{1}{1 - \dfrac{R_a}{C_p(T_0)J}} \tag{5.1.5}$$

The total temperature at the inlet is

$$T_0' = T_0 + \frac{V_0^2}{2gJC_p} \tag{5.1.6}$$

where

$$C_p = \frac{\Delta h(T_0', T_0)}{T_0' - T_0} \tag{5.1.7}$$

The correct total temperature is then obtained by an iterative procedure involving equations 5.1.4 to 5.1.7. The total pressure at the diffuser inlet is obtained from

$$\frac{R_a}{J} \ln \frac{P_0'}{P_0} = \Delta\varphi(T_0', T_0) \tag{5.1.8}$$

Therefore,

$$P_0' = P_0 e^{\Delta\varphi(T_0', T_0)J/R_a} \tag{5.1.9}$$

5.1.3.2 Diffuser

Since there is adiabatic flow in the diffuser,

$$T'_1 = T'_0 \tag{5.1.10}$$

The pressure ratio across the diffuser $r_{1,0}$ is an input parameter which is a function of the Mach number. Therefore

$$\frac{P'_1}{P'_0} = r_{1,0} \tag{5.1.11}$$

An example variation of this parameter is presented in Reference 2.

5.1.3.3 Fan

The fan pressure ratio $P'_{1'}/P'_1$ is a variable. To determine the ideal fan exit temperature, the isentropic flow equation is used. Therefore

$$\Delta\varphi_F = \frac{R_a}{J} \ln \frac{P'_{1'}}{P'_1} \tag{5.1.12}$$

and

$$\Delta\varphi(T'_{1',id}, T'_1) = \Delta\varphi_F \tag{5.1.13}$$

Therefore, $T'_{1',id}$ can be determined from Equation 5.1.13. The fan work is

$$\Delta h_F = \frac{\Delta h(T'_{1',id}, T'_1)}{\eta_F} \tag{5.1.14}$$

where fan efficiency is a design parameter. The fan exit temperature $T'_{1'}$ is determined from

$$\Delta h(T'_{1'}, T'_1) = \Delta h_F \tag{5.1.15}$$

The total airflow is

$$W_{tot} = W_{a,D} + W_{a,m} \tag{5.1.16}$$

where $W_{a,D}$ is the duct airflow and $W_{a,m}$ is the main airflow. The ratio of the duct airflow to the main airflow $W_{a,D}/W_{a,m}$ is called the bypass ratio b. Therefore,

$$W_{tot} = (1 + b)W_{a,m} \tag{5.1.17}$$

5.1.3.4 Compressor

The overall pressure ratio of the fan and compressor is a variable. Thus, the compressor pressure ratio is

$$\frac{P'_2}{P'_{1'}} = \frac{P'_2/P'_1}{P'_{1'}/P'_1} \tag{5.1.18}$$

The ideal compressor exit temperature $T'_{2,\,id}$ is obtained from

$$\Delta\varphi(T'_{2,\,id}, T'_{1'}) = \frac{R_a}{J} \ln \frac{P'_2}{P'_{1'}} \tag{5.1.19}$$

The compressor work is

$$\Delta h_C = \frac{\Delta h(T'_{2,\,id}, T'_{1'})}{\eta_C} \tag{5.1.20}$$

where the compressor efficiency is a design parameter. The compressor exit temperature T'_2 is determined from

$$\Delta h(T'_2, T'_{1'}) = \Delta h_C \tag{5.1.21}$$

5.1.3.5 Combustor

An energy balance for the combustor yields

$$W_{f,m}h_f + \eta_B W_f HVF + (W_{a,m} - W_c)h_a = (W_{a,m} - W_c + W_f)h_g \tag{5.1.22}$$

For the enthalpy of the fuel to be equal to zero, the enthalpy reference temperature T_R must be equal to the temperature of the incoming fuel. As discussed in Section 5.1.5, the enthalpy change of the gas can be expressed as

$$\Delta h_g = (\Delta h_a + \Delta h_b f) \frac{1}{1 + f} \tag{5.1.23}$$

Therefore, substituting Equation 5.1.23 into Equation 5.1.22 and dividing by $(W_{a,m} - W_c)$ yields

$$\eta_B \frac{W_{f,m} HVF}{W_{a,m} - W_c} + \Delta h(T_2', T_R) = \Delta h(T_3', T_R) + \frac{W_{f,m}}{W_{a,m} - W_c} \Delta h_b(T_3', T_R)$$

The fuel to air ratio based on combustor airflow is

$$\frac{W_{f,m}}{W_{a,m} - W_c} = \frac{\Delta h(T_3', T_R) - \Delta h(T_2', T_R)}{HVF\eta_B - \Delta h_b(T_3', T_R)} \tag{5.1.24}$$

where the turbine inlet temperature T_3' and the combustor efficiency η_B are design parameters. The fuel to air ratio based on main flow is

$$\frac{W_{f,m}}{W_{a,m}} = \frac{W_{f,m}}{W_{a,m} - W_c}\left(\frac{W_{a,m} - W_c}{W_{a,m}}\right) = \frac{W_{f,m}}{W_{a,m} - W_c}\left(1 - \frac{W_c}{W_{a,m}}\right) \tag{5.1.25}$$

where the coolant-flow ratio $W_c/W_{a,m}$ is a design parameter. The combustor pressure ratio $r_{3,2}$ is an input parameter.

5.1.3.6 Turbine—

The turbine work is equal to the fan work plus the compressor work.

$$(W_{a,m} - W_c + W_{f,m})\Delta h_T = W_{tot}\,\Delta h_F + W_{a,m}\,\Delta h_C \tag{5.1.26}$$

Substituting Equation 5.1.25 into Equation 5.1.26 and solving for turbine enthalpy drop yield

$$\Delta h_T = \frac{(1 + b)\Delta h_F + \Delta h_C}{\left(1 - \frac{W_c}{W_{a,m}}\right)\left(1 + \frac{W_{f,m}}{W_{a,m} - W_c}\right)} \tag{5.1.27}$$

Since

$$\Delta h\left(T_3', T_4', \frac{W_{f,m}}{W_{a,m} - W_c}\right) = \Delta h_T \tag{5.1.28}$$

The turbine exit temperature T'_4 is determined from Equation 5.1.28. To determine the turbine pressure ratio, the ideal turbine-exit temperature, $T'_{4,id}$ is needed. Therefore, the ideal turbine work is

$$\Delta h_{T,id} = \frac{\Delta h_T}{\eta_T} \tag{5.1.29}$$

and

$$\Delta h\left(T'_3, T'_{4,id}, \frac{W_{f,m}}{W_{a,m} - W_c}\right) = \Delta h_{T,id} \tag{5.1.30}$$

where turbine efficiency is a design parameter. The ideal turbine-exit temperature $T'_{4,id}$ is determined from Equation 5.1.30. Thus, the turbine pressure ratio is

$$\frac{P'_3}{P'_4} = \exp\left[\Delta\varphi\left(T_3, T'_{4,id}, \frac{W_{f,m}}{W_{a,m} - W_c}\right)\overline{M}_g \frac{J}{R}\right] \tag{5.1.31}$$

where the equation for the molecular weight of the gas is given in Section 5.1.4.

5.1.3.7 Coolant Mixing

The turbine-exit gas and coolant flow mix just downstream of the turbine. This mixing is assumed to take place without any change in the mainstream total pressure. However, there is a change in total temperature. An energy balance for the mixing section with 0°R as the reference temperature yields

$$W_c\,\Delta h(T'_2, 0) + (W_{a,m} - W_c + W_{f,m})\Delta h\left(T'_4, 0, \frac{W_{f,m}}{W_{a,m} - W_c}\right)$$
$$= (W_{a,m} + W_{f,m})\Delta h\left(T'_{4M}, 0, \frac{W_{f,m}}{W_{a,m}}\right) \tag{5.1.32}$$

Dividing by $W_{a,m}$ yields

$$\frac{W_c}{W_{a,m}}\Delta h(T'_2, 0) + \left(1 - \frac{W_c}{W_{a,m}}\right)\left(1 + \frac{W_{f,m}}{W_{a,m} - W_c}\right)\Delta h\left(T'_4, 0, \frac{W_{f,m}}{W_{a,m} - W_c}\right)$$
$$= \left(1 + \frac{W_{f,m}}{W_{a,m}}\right)\Delta h\left(T'_{4M}, 0, \frac{W_{f,m}}{W_{a,m}}\right) \tag{5.1.33}$$

ORIGINAL PAGE IS
OF POOR QUALITY

Solving for the exit enthalpy yields

$$\Delta h\left(T'_{4M}, 0, \frac{W_{f,m}}{W_{a,m}}\right) = \frac{\frac{W_c}{W_{a,m}}\Delta h(T'_2, 0) + \left(1 - \frac{W_c}{W_{a,m}}\right)\left(1 + \frac{W_{f,m}}{W_{a,m} - W_c}\right)\Delta h\left(T'_4, 0, \frac{W_{f,m}}{W_{a,m} - W_c}\right)}{1 + \frac{W_{f,m}}{W_{a,m}}} \tag{5.1.34}$$

Total exit temperature T'_{4M} is determined from Equation 5.1.34. The total pressure is

$$P'_{4M} = P'_4 \tag{5.1.35}$$

5.1.3.8 Afterburner

Three cases are considered for the afterburner. For case I with no afterburner,

$$P'_5 = P'_4 \qquad T'_5 = T'_{4M} \qquad \frac{W_{f,AB}}{W_{a,m}} = 0 \tag{5.1.36}$$

For case II with the afterburner not lighted,

$$\frac{P'_5}{P'_4} = r_{5,4,n} \qquad T'_5 = T'_{4M} \qquad \frac{W_{f,AB}}{W_{a,m}} = 0 \tag{5.1.37}$$

where $r_{5,4}$ is given. For case III with afterburning, an energy balance on the afterburner yields

$$\eta_{AB}\frac{W_{f,AB}}{W_{a,m}}\mathrm{HVF} + \left(1 + \frac{W_{f,m}}{W_{a,m}}\right)h'_{4M} = \left(1 + \frac{W_{f,m}}{W_{a,m}} + \frac{W_{f,AB}}{W_{a,m}}\right)h'_5 \tag{5.1.38}$$

Solving Equation 5.1.38 for the afterburner fuel to air ratio yields

$$\frac{W_{f,AB}}{W_{a,m}} = \frac{\Delta h(T'_5, T_R) + \Delta h_b(T'_5, T_R)\frac{W_{f,m}}{W_{a,m}} - \left(1 + \frac{W_{f,m}}{W_{a,m}}\right)\Delta h\left(T'_{4M}, T_R, \frac{W_{f,m}}{W_{a,m}}\right)}{\eta_{AB}\mathrm{HVF} - \Delta h_b(T'_5, T_R)} \tag{5.1.39}$$

where the afterburner temperature T'_5 is a design parameter. The total pressure ratio across the afterburner is

$$\frac{P'_5}{P'_4} = r_{5,4} \tag{5.1.40}$$

which is a design parameter. Therefore, the total fuel to air ratio of the mainstream is

$$\frac{W_{f,tot}}{W_{a,m}} = \frac{W_{f,AB}}{W_{a,m}} + \frac{W_{f,m}}{W_{a,m}} \tag{5.1.41}$$

5.1.3.9 Main Nozzle

Full expansion is assumed in the mainstream nozzle. Therefore,

$$\frac{P_6}{P'_5} = \frac{P_0}{P'_5} \tag{5.1.42}$$

and

$$\Delta\varphi_N = \frac{\mathcal{R}}{JM_g} \ln \frac{P_6}{P'_5} \tag{5.1.43}$$

Since

$$\Delta\varphi\left(T_{6,id}, T'_5, \frac{W_{f,tot}}{W_{a,m}}\right) = \Delta\varphi_N \tag{5.1.44}$$

the ideal nozzle exit temperature $T_{6,id}$ can be determined from Equation 5.1.44. The main nozzle exit velocity is

$$V_6 = \psi \sqrt{2gJ\, \Delta h\left(T'_5, T_{6,id}, \frac{W_{f,tot}}{W_{a,m}}\right)} \tag{5.1.45}$$

where ψ is the velocity coefficient and a function of the airplane Mach number, Reference 2.

The equations derived thus far are for the main flow. The equations for the duct flow are now derived.

5.1.3.10 Duct Burner

Three cases are considered for the duct burner. For case I with no duct burner

$$P'_7 = P'_1, \qquad T'_7 = T'_1, \qquad \frac{W_{f,D}}{W_{a,D}} = 0 \tag{5.1.46}$$

ORIGINAL PAGE IS
OF POOR QUALITY

For case II with the duct burner not lighted,

$$\frac{P'_7}{P'_{1'}} = r_{7,1',n} \qquad T'_7 = T'_{1'} \qquad \frac{W_{f,D}}{W_{a,D}} = 0 \tag{5.1.47}$$

For case III with duct burning, an energy balance on the duct burner is

$$W_{a,D}\,\Delta h(T'_{1'}, T_R) + W_{f,DB}(\eta_{DB})(HVF) = (W_{a,D} \cdot W_{f,DB})\Delta h_g \tag{5.1.48}$$

Dividing Equation 5.1.48 by $W_{a,D}$ and solving for the duct-burner fuel to air ratio yield

$$\frac{W_{f,DB}}{W_{a,D}} = \frac{\Delta h(T'_7, T_R) - \Delta h(T'_{1'}, T_R)}{\eta_{DB} HVF - \Delta h_b(T'_7, T_R)} \tag{5.1.49}$$

where the duct-burner temperature T'_7 is a design parameter. The duct-burner fuel to air ratio based on the main flow is

$$\frac{W_{f,DB}}{W_{a,m}} = \frac{W_{f,DB}}{W_{a,D}}\,\frac{W_{a,D}}{W_{a,m}} = \frac{W_{f,DB}}{W_{a,D}}\,b \tag{5.1.50}$$

The total pressure ratio is

$$\frac{P'_7}{P'_{1'}} = r_{7,1'} \tag{5.1.51}$$

which is a design parameter.

5.1.5.11 Duct Nozzle

Full expansion is also assumed in the duct nozzle. Therefore,

$$\frac{P_8}{P'_7} = \frac{P_0}{P'_7} \tag{5.1.52}$$

and

$$\Delta\varphi_{D,N} = \frac{\mathcal{R}}{J\bar{M}_g} \ln \frac{P_8}{P'_7} \tag{5.1 53}$$

Since

$$\Delta\varphi\left(T_{8,id}, T'_7, \frac{W_{f,DB}}{W_{a,D}}\right) = \Delta\varphi_{D,N}$$

(5.1.54)

the ideal exit temperature $T_{8,id}$ is determined from Equation 5.1.54. The duct nozzle exit velocity is

$$V_8 = \psi \sqrt{2gJ\ \Delta h\left(T'_7, T_{8,id}, \frac{W_{f,DB}}{W_{a,D}}\right)}$$

(5.1.55)

where ψ is a function of M_0, Reference 1.

5.1.3.12 Specific Thrust

The specific thrust of the engine is defined as the net thrust divided by the total airflow:

$$ST = \frac{(W_8 V_8 + W_6 V_6 - W_{tot} V_0)}{g W_{tot}}$$

(5.1.56)

Substituting the weight-flow relations yields

$$ST = \frac{(W_{a,D} + W_{f,DB})V_8 + (W_{a,m} + W_{f,tot})V_6 - (1 + b)W_{a,m} V_0}{g(1 + b)W_{a,m}}$$

(5.1.57)

and rearranging Equation 5.1.57 yields

$$ST = \frac{\left(b + \frac{W_{f,DB}}{W_{a,m}}\right)V_8 + \left(1 + \frac{W_{f,tot}}{W_{a,m}}\right)V_6 - (1 + b)V_0}{g(1 + b)}$$

(5.1.58)

5.1.3.13 Specific Fuel Consumption

Specific fuel consumption is defined as the total fuel flow in pounds per hour divided by the net thrust in pounds:

$$SFC = \frac{(W_{f,tot} + W_{f,DB})3600g}{(W_{a,D} + W_{f,DB})V_8 + (W_{a,m} + W_{f,tot})V_6 - (1 + b)W_{a,m}V_0} \tag{5.1.59}$$

Dividing Equation 5.1.59 by $W_{a,m}$ yields

$$SFC = \frac{\left(\frac{W_{f,tot}}{W_{a,m}} + \frac{W_{f,DB}}{W_{a,m}}\right)3600g}{\left(b + \frac{W_{f,DB}}{W_{a,m}}\right)V_8 + \left(1 + \frac{W_{f,tot}}{W_{a,m}}\right)V_6 - (1 + b)V_0} \tag{5.1.60}$$

5.1.3.14 Engine Efficiency

Another performance parameter that is often used is engine efficiency:

$$\eta_e = \frac{\text{Thrust power}}{\text{Heat added}} \tag{5.1.61}$$

The thrust power is equal to the net thrust multiplied by the inlet velocity. The heat added is

$$W_f(HVF)J \tag{5.1.62}$$

Therefore

$$\eta_e = \frac{3600 v_0}{(SFC)(HVF)J} \tag{5.1.63}$$

The method presented for determining the performance of a general jet engine applies to several engines. dry turbojet, afterburning turbojet, dry turbofan, and duct burning turbofan. The performance of any one of these engines is obtained by eliminating the appropriate components of the general engine from the calculation procedure.

The analysis of a dry turbojet is obtained by eliminating the duct equations, setting the fan pressure ratio (P_{1I}'/P_1') equal to 1, the bypass ratio (b) equal to 0, the fan efficiency (η_F) equal to 1.0, and taking case I for the afterburner section The analyses of the afterburning turbojet are the same as the dry turbojet except that case II or III for the afterburner section is used. The analyses of the turbofan engines are obtained by eliminating the afterburner, that is, by setting $P_5' = P_{4M}'$, and $T_5' = T_{4M}'$.

5.1.4 Derivation of Combustion Gas Thermodynamic Property Equations

5.1.4.1 Reaction Stoichiometry

The fuel used was of the form C_nH_{2n}. Therefore, the general combustion equation is

$$C_nH_{2n} + \frac{3}{2} nO_2 \rightarrow nCO_2 + nH_2O \qquad (5.1.64)$$

Eliminating n, the reaction and the formula weights are

$$\underbrace{CH_2}_{14.026} + \underbrace{\frac{3}{2} O_2}_{48.00} \rightarrow \underbrace{CO_2}_{44.010} + \underbrace{H_2O}_{18.016} \qquad (5.1.65)$$

For f pounds of fuel used, the amount of O_2 used is

$$\frac{48.000}{14.026} f = 3.422 f \qquad (5.1.66)$$

the amount of CO_2 formed is

$$\frac{44.010}{14.026} f = 3.138 f \qquad (5.1.67)$$

and the amount of H_2O formed is

$$\frac{18.016}{14.026} f = 1.284 f \qquad (5.1.68)$$

The weights of the components of air per pound of air are oxygen, 0.2314; nitrogen, 0.7552, argon, 0.0129, and carbon dioxide, 0.0005. Therefore, the net amounts of components left after the reaction of f pounds of fuel with one pound of air are

$$\overline{W}_{O_2} = 0.2314 - 3.422 f \qquad (5.1.69)$$

$$\overline{W}_{CO_2} = 0.0005 + 3.138 f \qquad (5.1.70)$$

$$\overline{W}_{H_2O} = 1.284 f \qquad (5.1.71)$$

$$\overline{W}_{N_2} = 0.7552 \qquad (5.1.72)$$

$$\overline{W}_{Ar} = 0.0129 \qquad (5.1.73)$$

and the weight of the gas is 1 + f pounds per pound of air.

ORIGINAL PAGE IS
OF POOR QUALITY

5.1.4.2 Specific Heat

The specific heat of the gas is

$$\left(C_p\right)_g = \frac{\sum\left(\overline{W}C_p\right)_{components}}{1+f} \tag{5.1.74}$$

The specific heat of each component is expressed in the form below, Reference 3.

$$C_p = A + B(T\times10^{-3}) + C\left(T\times10^{-3}\right)^2 + D\left(T\times10^{-3}\right)^3 + E\left(T\times10^{-3}\right)^4 \tag{5.1.75}$$

where C_p is in Btu per pound mass per °R and T is in °R. The coefficients A, B, C, D, and E are obtained from Reference 3. These coefficients and the molecular weight of each component are given in Table 5.1-2.

Therefore, the equation for the specific heat of the combustion products is obtained by substituting Equation 5.1.75 into Equation 5.1.74. The resulting equation is

$$\begin{aligned}\left(C_p\right)_g = \left(\frac{1}{1+f}\right)\Bigg\{&0.24062 - 0.017724\,T\times10^{-3} + 0.038056\times\left(10^{-3}T\right)^2 - 0.012662\times\left(10^{-3}T\right)^3 \\ &+ 0.0013012\times\left(10^{-3}T\right)^4 + \Big[0.22091 + 0.51822\times10^{-3}\,T \\ &- 0.19462\times\left(10^{-3}T\right)^2 + 0.045089\times\left(10^{-3}T\right)^3 - 0.0043275\times\left(10^{-3}T\right)^4\Big]f\Bigg\}\end{aligned} \tag{5.1.76}$$

$$C_p(T,f) = \left(C_p\right)_g \tag{5.1.77}$$

5.1.4.3 Enthalpy Change

The enthalpy change can be expressed as

$$\Delta h = \int_{T_1}^{T_2} C_p \, dT \tag{5.1.78}$$

Substituting Equation 5.1.76 into Equation 5.1.78 and integrating yields

$$\begin{aligned}\Delta h_g = \frac{1}{1+f} \Bigg\{ & 0.24062(T_2 - T_1) - \frac{0.017724\times10^{-3}}{2}\left(T_2^2 - T_1^2\right) + \frac{0.038056\times10^{-6}}{3}\left(T_2^3 - T_1^3\right) \\ & - \frac{0.012662\times10^{-9}}{4}\left(T_2^4 - T_1^4\right) + \frac{0.0013012\times10^{-12}}{5}\left(T_2^5 - T_1^5\right) + \Bigg[0.22091(T_2 - T_1) \\ & + \frac{0.51822\times10^{-3}}{2}\left(T_2^2 - T_1^2\right) - \frac{0.19462\times10^{-6}}{3}\left(T_2^3 - T_1^3\right) + \frac{0.045089\times10^{-9}}{4}\left(T_2^4 - T_1^4\right) \\ & - \frac{0.0043275\times10^{-12}}{5}\left(T_2^5 - T_1^5\right)\Bigg] f \Bigg\}\end{aligned} \tag{5.1.79}$$

$$\Delta h(T_2, T_1, f) = \Delta h_g \tag{5.1.80}$$

The enthalpy change of the gas can also be expressed as

$$\Delta h_g = \frac{1}{1+f}\left(\Delta h_a + \Delta h_b f\right) \tag{5.1.81}$$

where Δh_a is the enthalpy change of o pound of air

$$\Delta h_a = \Delta h(T_2, T_1) \tag{5.1.82}$$

and Δh_b is the correction to the air enthalpy due to combustion and is expressed as

$$\begin{aligned}\Delta h_b = & 0.22091(T_2 - T_1) + \frac{0.51822\times10^{-3}}{2}\left(T_2^2 - T_1^2\right) - \frac{0.19462\times10^{-6}}{3}\left(T_2^3 - T_1^3\right) \\ & + \frac{0.045089\times10^{-9}}{4}\left(T_2^4 - T_1^4\right) - \frac{0.0043275\times10^{-12}}{5}\left(T_2^5 - T_1^5\right)\end{aligned} \tag{5.1.83}$$

5.1.4.4 Entropy Function

For isentropic flow,

$$\frac{R}{J} \ln \frac{P_2}{P_1} = \int_{T_1}^{T_2} \frac{C_p}{T} \, dT \tag{5.1.84}$$

This equation is used in the evaluation of ideal processes in turbomachines. For convenience the right side of Equation 5.1 84 will be called the entropy function. The entropy function is

$$\Delta\varphi = \int_{T_1}^{T_2} \frac{C_p}{T} \, dT \tag{5.1.85}$$

Substituting Equation 5.1.76 into Equation 5.1.85 and integrating yields

$$\begin{aligned}\Delta\varphi_g = \frac{1}{1+f} \Bigg\{ & 0.24062 \ln \frac{T_2}{T_1} - 0.017724\times10^{-3}(T_2 - T_1) + \frac{0.038050\times10^{-6}}{2}\left(T_2^2 - T_1^2\right) \\ & - \frac{0.012062\times10^{-9}}{3}\left(T_2^3 - T_1^3\right) + \frac{0.0013012\times10^{-12}}{4}\left(T_2^4 - T_1^4\right) \\ & + f\Bigg[0.22091 \ln \frac{T_2}{T_1} + 0.31822\times10^{-3}(T_2 - T_1) - \frac{0.19462\times10^{-6}}{2}\left(T_2^2 - T_1^2\right) \\ & + \frac{0.025009\times10^{-9}}{3}\left(T_2^3 - T_1^3\right) - \frac{0.0013275\times10^{-12}}{4}\left(T_2^4 - T_1^4\right)\Bigg]\Bigg\}\end{aligned} \tag{5 1.86}$$

$$\Delta\varphi(T_2, T_1, f) = \Delta\varphi_g \tag{5.1.87}$$

5.1.4 5 Molecular Weight

The molecular weight of the gas is equal to the weight of the gas divided by the total number of moles (sum of the moles of the components). Therefore, molecular weight can be expressed as

$$\overline{M}_g = \frac{1+f}{\sum \left(\frac{W}{\overline{M}}\right)_{components}} \tag{5 1.88}$$

the resulting equation is

$$\overline{M}_g = \frac{1+f}{0.034522 + 0.036648\, f} \tag{5.1.89}$$

C-3

5.1.5 Symbols for Description of Program ENCYCL

b bypass ratio

C_p specific heat, Btu/(lb)(°R)

f fuel to air ratio, (lb fuel)/(lb air)

g gravitational constat, 32.17 ft/sec^2

HVF heating value of fuel, Btu/lb

h enthalpy, Btu/lv

Δh enthalpy change, Btu/lv

Δh_b correction to air enthalpy (Equation 5.1.83), Btu/lb

J mechanical equivalent of heat, 778 ft-lb/Btu

M Mach number

$\bar{M}$ molecular weight

P pressure, psia

R gas constant, ft-lb/(lb)(°R)

$\mathcal{R}$ universal gas constant, ft-lb/(lb mole)(°R)

r pressure ratio

SFC specific fuel consumption, (lb fuel)/(lb thrust)(hr)

ST specific thrust, (lb thrust)/(lb air)

T temperature, °R

V velocity, ft/sec

W weight flow, lb/hr

$\bar{W}$ weight of component per pound of air, lb/lb air

γ ratio of specific heats

η efficiency

$\Delta\phi$ entropy function

ψ velocity coefficient

Subscripts:

AB	afterburner
a	air
B	combustor
C	compressor
c	coolant
D	duct
DB	duct burner
e	engine
F	fan
f	fuel
g	gas
id	ideal
m	main
N	nozzle
n	not lighted
R	reference
T	turbine
t	thrust
tot	total
0	diffuser inlet
1	fan inlet
1'	fan outlet
2	compressor outlet, main burner inlet
3	main burner outlet, turbine inlet
4	turbine outlet
4M	mixing station
5	afterburner outlet
6	nozzle outlet
7	duct burner outlet
8	duct nozzle outlet

Superscript:

(') total, as applied to state points

REFERENCES:

1. Vanco, Michael R., Computer Program for Design-Point Performance of Turbojet and Turbofan Engine Cycles, NASA TM X-1340.

2. Dugan, J. F., Jr., Koenig, R. W., Whitlow, J. B., Jr., and McAuliffe, T.B., Power for the Mach 3 SST, Astron. Aeron., Vol. 2, No. 9, September 1964, Pages 44-51.

3. McBride, Bonnie J., Heimel, Sheldon, Ehlers, Janet G., and Gordon, Sanford, Thermodynamic Properties to 6000°K for 210 Substances Involving the First 18 Elements, NASA SP-3001, 1963.

TABLE 5.1-1. SCHEMATIC DIAGRAM OF GENERAL ENGINE

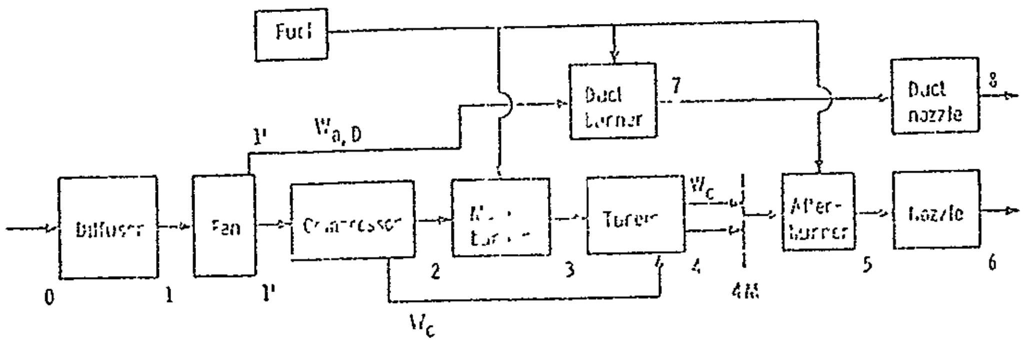


TABLE 5.1-2. COEFFICIENTS AND MOLECULAR WEIGHTS OF COMPONENTS

Components	Coefficients in eq (B4)					Molecular weight, $\overline{M}$
	A	B	C	D	E	
Nitrogen	0.27[illegible]	-[illegible]	[illegible]	0.01[illegible]	[illegible]	28.016
Oxygen	.26[illegible]	[illegible]	[illegible]	-[illegible]	[illegible]	32.000
Carbon dioxide	1[illegible]	21[illegible]	-[illegible]	[illegible]	-[illegible]	44.010
Water vapor	[illegible]	-.00[illegible]	[illegible]1	-[illegible]	[illegible]	18.01
Argon	1[illegible]	0	0	0	0	[illegible]

ORIGINAL PAGE IS OF POOR QUALITY

TABLE OF CONTENTS FOR SECTION 5.2, PROGRAM GENENG

5.2 PROGRAM GENENG: A PROGRAM FOR CALCULATING DESIGN AND OFF-DESIGN PERFORMANCE FOR TURBOJET AND TURBOFAN ENGINES

Program GENENG was developed by Koenig and Fishback of NASA's Lewis Research Center. Documentation of the program is provided in Reference 1. The discussion of program GENENG presented below follows Reference 1.

The original version of the GENENG computer program entitled SMOTE (SiMulation Of Turbofan Engine) and was developed by the Turbine Engine Division of the Air Force Aero Propulsion Laboratory, Wright-Patterson Air Force Base, Ohio. SMOTE is capable of calculating only turbofan design and off-design performance using specific component performance maps. GENENG calculates steady-state design and off-design turbofan and one- and two-spool turbojet engine performance. The Reference 1 version of GENENG was prepared for the IBM 7094 computer. The ODIN/RLV version of GENENG was converted to the CDC 6600 by Robert Leko of the Naval Air Development Center.

5.2.1 Introduction

For preliminary as well as in-depth studies it is necessary to study a broad range of engines operating at both design and off-design conditions in order to find an efficient airframe/engine combination. The spectrum of flight conditions through which an engine must operate will strongly affect the optimum design parameters for that engine.

The SMOTE code discussed in References 2 and 3 provided a computer program having off-design point calculation capability for either existing engines or theoretical ones--a major advance. Theoretical engines are simulated by scaling component performances from existing engines to the design conditions of the theoretical engines.

Program GENENG (GENeralized ENGine), a computer code derived from SMOTE, was written to improve the versatility of SMOTE. Among the changes made are

1. One- and two-spool turbojets can be calculated, as well as turbofans

2. Afterburner performance maps can be used

3. Nozzle performance maps can be used

4. Fan and compressor pressure ratios are automatically redesigned for mixed-flow turbofans if the statis pressure losses are calculated.

5. Duct combustor pressure losses are calculated

6. A new method of entering data into the program is used

A derivative program from GENENG, called GENENG II, is reported in the next section, 5.3; for further detail see Reference 4. GENENG II calculates performance of two- or three-spool front or aft fan turbofan engines with as many as three nozzles (or airstreams).

5.2.2 Thermodynamic Analysis of One- and Two-Spool Engine Types

All thermodynamic properties of air and gas are calculated by considering variable specific heats and no dissociation. Curve fitted air and gas property tables of Reference 5 are used. The engine cycles that can be studied using GENENG are discussed below.

5.2.2.1 Two-Spool Afterburning Turbofan

The basic engine, a two-spool turbofan in shown in Figure 5.2-1. All other engine types are treated as variations of this basic engine. Free-stream conditions exist at Station 1 and are determined by using the U. S. Standard Atmosphere Table of 1962, Reference 6. The conditions at Station 2 are determined by flight speed and inlet recovery.

GENENG compressor maps work with corrected values of airflow. At the entrance to the fan the corrected airflow $WA_{F,c}$ is

$$WA_{F,c} = \frac{WA_F\sqrt{T_2/518.668}}{P_2/P_{SLS}} \tag{5.2.1}$$

where P_2 and P_{SLS} are in atmospheres and P_{SLS} equals 1.0. All symbols are defined in Table 5.2-1. Some symbols are formed as the combination of other symbols; thus WA is airflow; F is for fan; and c when following a compound symbol means corrected. Station numbers are defined on the appropriate figure.

All the fan air WA_F is compressed by the fan giving rise to conditions at station 21. The power required to do this is

$$\text{Fan power} = WA_F \times (H_{21} - H_2) \tag{5.2.2}$$

Some fan air may be lost to the cycle as fan bleed Bl_F, which is expressed as a fraction of the fan airflow

$$Fl_F = PC_{Bl,F} \times WA_F \tag{5.2.3}$$

The corrected airflow into the core compressor is

$$WA_{C,c} = \frac{WA_C\sqrt{T_{21}/518.668}}{P_{21}/1.0} \tag{5.2.4}$$

The remaining air goes through the fan duct where some leakage from the core air may also enter; see Equation 5.2.11

$$WA_D = WA_F - Bl_F - WA_C + Bl_{DU} \tag{5.2.5}$$

The air which may be heated by a duct burner to a temperature T_{24} undergoes a pressure drop

$$P_{25} = P_{24} \times \left[1 - \left(\frac{\Delta P}{P}\right)_{DUCT}\right] \tag{5.2.6}$$

The air would have been heated by the addition of fuel which can be expressed as a fuel-air ratio so that

$$WG_{24} = WA_{23} \times [1 + (f/a)_{23}] \tag{5.2.7}$$

The gas is then either expanded through a nozzle (station 29) to produce thrust or is mixed with the core air as shown in Figure 5.2-2 (mixed flow turbofan). The bypass ratio of the engine is defined by

$$BYPASS = \frac{WA_D}{WA_C} \tag{5.2.8}$$

The air continuing into the core is compressed to conditions at station 3. The power required is

$$\text{Compressor power} = WA_C \times (H_3 - H_{21}) = WA_3 \times (H_3 - H_{21}) \tag{5.2.9}$$

Some core bleed air Bl_C may be used for turbine cooling. Some of the air is put back into the cycle into each of the two turbines, and some is lost to the cycle as overboard bleed or leakage into the fan duct.

$$Bl_C = PC_{Bl,C} \times WA_3 \tag{5.2.10}$$

$$Bl_{DU} = PC_{Bl,DU} \times Bl_C \tag{5.2.11}$$

$$Bl_{OB} = PC_{Bl,OB} \times Bl_C \tag{5.2.12}$$

$$Bl_{HP} = PC_{Bl,HP} \times Bl_C \tag{5.2.13}$$

$$Bl_{LP} = PC_{Bl,LP} \times Bl_C \tag{5.2.14}$$

Since $Bl_{DU} + Bl_{OB} + Bl_{HP} + Bl_{LP} = Bl_C$, the sum of $PC_{Bl,DU} + PC_{Bl,OB} + PC_{Bl,HP} + PC_{Bl,LP}$ must be equal to 1.0. The remaining air is

$$WA_4 = WA_3 - Bl_C \tag{5.2.15}$$

and is heated to a turbine inlet temperature T_4 while undergoing a combustor pressure drop $(\Delta P/P)_{COMB}$. The fuel required to do this is expressed as a fuel-air ratio $(f/a)_4$ so that the weight of the gas entering the first (high) pressure) turbine WG_4 can be expressed as

$$WG_4 = WA_4 \times [1 + (f/a)_4] \quad (5.2.16)$$

This gas is then expanded through the turbine to conditions at Station 5. The enthalpy at Station 5 is first calculated by making a power balance since this turbine drives the compressor and supplies any work extracted (HPEXT). By using Equation 5.2.9

$$WG_4 \times (H_4 - H_5) = WA_3 \times (H_3 - H_{21}) + HPTEXT \quad (5.2.17)$$

In addition, the physical speeds must match

$$N_{HP,TURBINE} = N_{COMP} \quad (5.2.18)$$

If high pressure turbine bleed air Bl_{HP} is added into the cycle at this point, H_5 must be readjusted

$$H_5 = \frac{(Bl_{HP} \times H_3) + WG_4 H_5}{WG_4 + Bl_{HP}} = \frac{(Bl_{HP} \times H_3) + WG_4 H_5}{WG_5} \quad (5.2.19)$$

Similarly,

$$WG_5 \times (H_5 - H_{55}) = WA_F \times (H_{21} - H_2) \quad (5.2.20)$$

$$N_{LP,TURBINE} = N_{FAN} \quad (5.2.21)$$

$$H_{55} = \frac{(Bl_{LP} \times H_3) + WG_5 H_{55}}{WG_5 + Bl_{LP}} = \frac{(Bl_{LP} \times H_3) + WG_5 H_{55}}{WG_{55}} \quad (5.2.22)$$

For non-mixed flow turbofans the gas flow at Station 6, WG_6, is identical with that at Station 55, WG_{55}. For mixed-flow turbofans, the air in the fan duct is added.

$$WG_6 = WG_{55} + WA_D \quad (5.2.23)$$

Mixed-flow turbofans additionally require that the static pressures at Station 25 and at Station 55 (Figure 5.2-2) match.

$$PS_{55} = PS_{25} \quad (5.2.24)$$

The gas flow WG_6 then may be heated by an afterburner to a gas temperature T_7 and may undergo a pressure drop.

$$P_7 = P_6 \, [1 - (\Delta P/P)_{AFTERBURNER}] \quad (5.2.25)$$

And the gas flow rate would be increased by any fuel burned

$$WG_7 = WG_{55} + WFA \quad (5.2.26)$$

The gas is then expanded through the nozzle (Station 9) to produce the remainder of the engine thrust.

5.2.2.2 Two-Spool Turbojet

The two-spool turbojet is equivalent to a two-spool turbofan with a BYPASS of zero. This engine is shown in Figure 5.2-3. In calculating this type of engine, there is no fan duct and the air entering the inner compressor is the same as the air entering the inlet less any bleed.

$$WA_C = WA_F - B1_F \tag{5.2.27}$$

The thermodynamic calculations proceed identically to the previous case, the two-spool turbofan case, except that any equations referring to the fan duct are eliminated.

5.2.2.3 One-Spool Turbojet

The one-spool turbojet is shown in Figure 5.2-4. As can be seen, to simulate this engine the inner compressor and its driving turbine are eliminated. That is, Stations 21 and 3 become identical and Stations 4 and 5 become identical.

The only calculation changes required therefore are (1) to eliminate any thermodynamic equations relating to the fan duct and the inner spool of the two-spool turbofan engine and (2) to add the horsepower extracted to the power requirements of the outer turbine.

5.2.3 Balancing Technique

An off-design engine cycle calculation requires satisfying various matching constraints (rotational speeds, airflows, compressor and turbine work functions, and nozzle flow functions) at each specified operating condition. GENENG internally searches for compressor and turbine operating points that will satisfy the constraints. It does this by generating differential errors caused by small changes in the independent variables. The program then uses a matrix that is loaded with the differential errors to solve for the zero error condition using the Newton-Raphson iteration technique.

For the two-spool turbofan or turbojet engines a solution for a set of *six simultaneous linear algebraic equations* is obtained; for the one-spool turbojet a set of *three simultaneous linear equations* is solved. The six independent variables selected are

(a) ZF - Ratio of pressure ratios of outer compressor (fan) along a speed line

$$ZF = \frac{\text{(Pressure ratio along speed line) - (Low pressure ratio on speed line)}}{\text{(High pressure ratio on speed line) - (Low pressure ratio on speed line)}}$$

(b) PCNF or T4 - Per cent fan speed or turbine inlet temperature

(c) ZC - Ratio of pressure ratios of inner compressor along speed line; calculated same as ZF

(d) PCNC - Per cent compressor speed or turbine inlet temperature or T4

(e) TFFHP - Inner (high pressure) turbine flow function,

$$WG_4\sqrt{T_4}/P_4$$

(f) TFFLP - Outer (low pressure) turbine flow function,.

$$WG_5\sqrt{T_5}/P_5$$

ZC, PCNC, and TFFHP are not used for the one-spool turbojet.

The program initially selects new (perturbed) values for the variables based on the design values. It is then possible to proceed through the entire engine cycle where six (or three) errors are generated. The initial values of the six (or three) variables and six (or three) errors are base values.

From Reference 2, the partial differential equations for E = f(v) are

$$dE_i + \sum_{j=1}^{j_{max}} \frac{\partial E_{ij}}{\partial V_j} dV_j \qquad (5.2.28)$$

for i going from 1 to j_{max} where j_{max} is 6 for two-spool engines or 3 for one spool turbojets, E is an error; V is a variable; and ∂E_{ij} is the change in E_i caused by a change in V_j.

The assumption of a small change in the variable results in the following approximations (B refers to a base value):

$$dE = E - EB \qquad (5.2.29)$$

$$dV = V - VB \qquad (5.2.30)$$

$$\frac{\partial E}{\partial V} = \frac{E}{V} \qquad (5.2.31)$$

With these approximations and the knowledge that E should equal zero for the balanced engine, the set of partial differential equations (Equation 5.2.28) reduces to

$$E_i - EB_i = \sum_{j=1}^{j_{max}} \frac{\partial E_{ij}}{\partial V_j} dV_j = EB_i \qquad (5.2.32)$$

for i going from 1 to j_{max}.

Thus, the calculations made with the perturbed variables are used to compute $\Delta E/\Delta V$, and Equation 5.2.32 is solved for dV_j. The variables V are then given new values from

$$V_j = V_j B + dV_j \tag{5.2.33}$$

If the engine cycle calculations were linear functions, the engine would balance with the new values of the variables. However, the calculations are nonlinear and it usually is necessary to repeat the process of changing each variable by a small amount for each pass. A change in each error because of the small change in the variable is calculated for each pass, where the new values become base values. This process occurs several times before a balance is obtained.

The most often used independent variable and the differential errors for four types of engines that can be run on GENENG are listed in Table 5.2-2.

5.2.4 Choice of Component Maps - Scaling Laws

Many of the engines that are studied using GENENG are theoretical. Therefore, actual component maps for these engines will be nonexistent. The program, however, does require component maps to do off-design calculations. To alleviate this problem, GENENG uses scaling laws to change data from one component map into a new component map. Hopefully, a component map can be found which could be expected to perform in a similar manner to the actual map for the engine type being studied. In fact, many maps are identified as to the range of pressure ratio and the engine component design type for which they are valid (i.e , pressure ratio range, 4 to 8; subsonic compressor or inner compressor). However, it should be noted, for example, that a high bypass ratio, subsonic flight speed, low pressure ratio fan map for a CF6 engine would not properly simulate a low bypass ratio, high pressure ratio, supersonic multistage fan.

5.2.4.1 Compressor Maps

The scaling equations used for the compressor maps are

$$PR = \frac{PR_{design} - 1}{PR_{map,design} - 1} \quad (PR_{map} - 1) + 1 \tag{5.2.34}$$

$$WA = \frac{WA_{design}}{WA_{map,design}} \times WA_{map} \tag{5.2.35}$$

$$ETA = \frac{ETA_{design}}{ETA_{map,design}} \times ETA_{map} \tag{5.2.36}$$

ORIGINAL PAGE IS
OF POOR QUALITY

Similar equations are used for combustor and turbine map scaling. These equations are found in the appropriate GENENG subroutines, Section 5.2.6. The correction factors used in scaling the maps are printed in the GENENG output. The closer these values are to 1.0 (especially pressure ratio, a primary characteristic of a given compressor map), the more reasonable are the simulated maps of the engine. Conversely, however, not being close to 1.0 does not necessarily mean that the simulation is poor since many maps have been shown to be typical over quite a large range of variables.

A typical compressor map which may be employed in the program is presented in Figure 5.2-5. The method of entering such a map into the GENENG program is described in detail, Reference 1.

5.2.4.2 Combustor Maps

The combustor map is a plot of temperature rise across the combustor against efficiency for constant input pressure. Temperature rise and input pressure are the independent variables. Combustor efficiency is the dependent variable. A typical combustor map is presented in Figure 5.2-6. The method of entering a combustor map into GENENG is described in Reference 1.

5.2.4.3 Turbine Maps

Turbine maps are entered into GENENG in a similar manner to fan and compressor maps, Section 5.2.4.1. The parameters of a typical turbine map are illustrated in Figure 5.2-7. Detailed instructions for describing specific turbine maps in the GENENG program are given in Reference 1.

5.2.4.4 Afterburners

Afterburner performance has been programmed in a generalized form in GENENG. The afterburner performance map included in the program is shown in Figure 5.2-8(a). The performance map shows afterburner combustion efficiency ratio as a function of fuel-air ratio. The value of afterburner combustion efficiency correction factor during off-design operation is shown against design afterburner inlet Mach number ratio, Figure 5.2-8(b) and design afterburner inlet total pressure ratio, Figure 5.208(c). Other correction factors or performance maps can be added as desired The afterburner efficiency fuel-air ratio, inlet total pressure, and Mach number are generalized external to the program.

A specific afterburner performance is generalized by dividing the specific off-design values by the design values. The generalized afterburner values are obtained as follows:

$$\text{Efficiency} = \frac{\text{Afterburner Efficiency Off-Design}}{\text{Afterburner Efficiency at Design Point}}$$

$$\text{Fuel-air ratio} = \frac{\text{Fuel-Air Ratio Off-Design}}{\text{Fuel-Air Ratio Design Point}}$$

$$\text{Entrance total Pressure} = \frac{\text{Entrance Total Pressure Off-Design}}{\text{Entrance Total Pressure at Design Point}}$$

$$\text{Entrance Mach number} = \frac{\text{Entrance Mach Number Off-Design}}{\text{Entrance Mach Number Design Point}}$$

To achieve a reasonable accuracy in cycle calculations when using any generalized component map, the usage of the map should be limited within a certain range of the original design values and configuration changes. Therefore, if an afterburner has a design task that differs significantly from an example used, a new generalized performance map should be used in order to simulate the component more accurately.

5.2.4.5 Nozzles

SMOTE, the original code, uses a single-point input for nozzle velocity coefficients when calculating engine performance. GENENG, however, uses a convergent-divergent nozzle velocity coefficient which is input in map form. The velocity coefficient is input as a function of nozzle total pressure ratio (P8/P1 or P28/P1). A typical nozzle performance map is illustrated in Figure 5.2-9. Detailed input routines for nozzle performance maps are presented in Reference 1.

5.2.5 Means of Specifying Mode of Engine Operations

Several methods are available for specifying off-design operation points. The most common one is to select a Mach number, altitude, and turbine inlet temperature other than design values. There are, however, several other possibilities which may be employed. For example, changing the following controls:

MODE = 0, Specify a new turbine inlet temperature T4

MODE = 1, Specify a compressor rotational speed PCNC

MODE = 2, Specify a fuel flow rate WFB

MODE = 3, Specify a fan rotational speed PCNF

If the engine has all its nozzles fixed, an input such as turbine inlet temperature, fuel flow, or speed will set the thrust level. But other means of changing engine operation can be accomplished by varying such nozzle thrust areas as

A8 Main nozzle thrust area

A28 Fan nozzle thrust area

For example, an off-design condition may exist where, in an attempt to satisfy continuity of mass flow (one of the component matching requirements), the fan operating point may lie outside the limits of the data map that was input for the component map. A fan nozzle thrust area change could be used to return the fan operating condition on the map such that a match would occur. This would indicate a possibility exists that variable fan nozzle would be required on this engine for operation at the desired condition.

It should be noted that the GENENG Huff input package is not available in the CDC 6600 program version. However, since the ODIN/RLV executive program permits any set of arithmetic and symbolic operations in the data input to a program, Section 2, the equivalent Huff input operations may be specified in ODIN/RLV simulations.

Input required for operation of the GENENG program is listed in Table 5.2-3.

5.2.6 GENENG Subroutine Functions and Descriptions

A flow chart of the computer program with the subroutines is shown in Figure 5.2-10. The functions of the subroutines are listed here and the purpose of each is described.

GENENG Dummy main program to initiate the calculations and cause the input of the controlled output variables. Because of the looping between subroutines, control is never transferred back to this routine.

ENGBAL Main routine. Controls all engine balancing loops; checks tolerances and number of loops and loads matrix, calls input.

GUESS Determines initial values of independent variable (see Table 5.2-2) at each point.

MATRIX Solves error matrix.

PUTIN Calls input subroutine package. Controls loop on static pressures for mixed flow turbofan.

ZERO Zeros nearly all of common and certain controls.

COINLT Determines ram recovery and performs inlet calculations.

ATMOS 1962 U. S. Standard Atmosphere Table.

RAM Calculates ram recovery defined by MIL-E-5008B specifications.

RAM2 Calculates special cases of input ram recovery as a function of flight Mach number.

COFAN Uses BLOCK DATA to perform outer compressor (fan) calculations.

COCOMP Uses BLOCK DATA to perform inner-compressor calculations (two spools only).

COCOMB Uses BLOCK DATA to perform combustor calculations. May use either T4 or WFB as the main parameter.

COHPTB Uses BLOCK DATA to perform inner turbine calculations (two spools only).

COLPTB Uses BLOCK DATA to perform outer turbine calculations.

CODUCT Performs duct and duct burning calculations for turbofans. May use either T24 or WFD as main parameters.

COMIX Performs gas mixing calculations if in mixed flow mode. At design points it calculates areas either from an input static pressure PS55 or from an input Mach number AM55 if PS55 = 9. At off-design points it calculates static pressures and Mach numbers from the design areas. Rescales pressure ratios for mixed flow turbofans to match duct and core static pressures just prior to mixing. COMIX also calculates afterburner entrance area A6 as a function of afterburner entrance Mach number AM6.

COAFBN Performs the afterburning calculations. May use either T7 or WFA as the main parameters.

FRTOSD Dummy routine to transfer values from common FRONT to common SIDE.

FASTBK Dummy routine to transfer values from common SIDE to common BACK.

COMNOZ Controls the main nozzle.

ERROR Controls all printouts if an error occurs. Prints names of subroutine where error occurred and also prints the values of all variables in the main commons.

SYG Controls printing from UNIT08. Throughout the program and particularly in ENGBAL, certain messages, variables, and matrix values are written on UNIT08 as an aid in determining why an error occurred or why a point did not balance. These values are printed out if subroutine ERROR is called and IDUMP is greater than zero, or after a good point if IDUMP = 2.

PERF Calculates performance after the engine is balanced.

OUTPUT Prints output except for controlled output. Prints the main commons after the design point.

CONOUT Controls and prints the controlled output variables.

THCOMP Performs isentropic calculations for compressors.

PROCOM Calculates thermodynamic gas properties for either air or a fuel-air mixture based on JP-4 using curve fits of the tables of Reference 5.

SEARCH General table lookup and interpolation routine to obtain data from the BLOCK DATA subroutines.

MAPBAC Used when calculations result in values not on the turbine maps. Changes the map value and an independent variable (PCNF, PCNC, or T4) in an attempt to rectify the situation.

CONVRG Performs nozzle calculations for a convergent nozzle.

CONDIV Performs nozzle calculations for a convergent-divergent (C-D) nozzle.

THTURB Performs isentropic calculations for turbines.

THERMO Provides thermodynamic conditions using PROCOM.

AFQUIR General quadratic interpolation routine.

PARABO Parabolic curve-fit routine

BLKFAN Performance data for outer compressor (fan) map (BLOCK DATA).

BLKCMP Performance data for inner compressor map (BLOCK DATA; two-spool engines).

CMBDAT BLOCK DATA for combustor

HPTDAT Performance data for inner turbine map (BLOCK DATA; two-spool engines).

LPTDAT Performance data for outer turbine map (BLOCK DATA).

ETAAB Generalized afterburner performance BLOCK DATA as a function of fuel-air ratio with correction factors for off-design afterburner entrance pressure and Mach number.

FRATIO Convergent-divergent nozzle velocity coefficient (BLOCK DATA input as a function of nozzle pressure ratio and area expansion ratio).

INPUT Package of Huff input subroutines.

REFERENCES:

1. Koenig, Robert W. and Fishbach, Laurence H., GENENG - A Program for Calculating Design and Off-Design Performance for Turbojet and Turbofan Engines, NASA TN D-6552, February 1972.

2. McKinney, John S., Simulation of Turbofan Engine, Part I, Description of Method and Balancing Technique, Report AFAPL-TR-67-125, Air Force Systems Command, November 1967. (Available from DDC as AD-825197).

3. McKinney, John S., Simulation of Turbofan Engine, Part II, User's Manual and Computer Program Listing, Report AFAPL-TR-67-125, Air Force Systems Command, November 1967. (Available from DDC as AD-825198).

4. Fishbach, Laurence H. and Koenig, Robert W., GENENG II - A Program for Calculating Design and Off-Design Performance of Two- and Three-Spool Turbofans with as Many as Three Nozzles, NASA TN D-6553, 1972.

5. Keenan, Joseph H., and Kaye, Joseph, Gas Tables, John Wiley and Sons, Inc., 1948.

6. ————., U. S. Standard Atmosphere, 1962. Prepared under sponsorship of NASA, USAF, and USWB, 1962.

7. Turner, Don N. and Huff, Vearl N., An Input Routine Using Arithmetic Statements for the IBM 704 Digital Computer. NASA TN D-1092, 1961.

TABLE 5.2-1. SYMBOLS

Thermodynamic Properties

AM	Mach number
FAR	fuel-air ratio, f/a
H	enthalpy, Btu/lbm
P	total pressure, atm
PS	static pressure, atm
S	entropy, Btu/^{o}R/lbm
T	total temperature, ^{o}R
TS	static temperature, ^{o}R
V	velocity, ft/sec

Station Numbers

See Figures 5.2-1 to 5.2-4 for each type of engine.

Component Symbols

A, AFT	afterburner
B	burner
C	inner compressor
COM	combustor
D	fan duct
F	first or fan compressor
I	intermediate (middle) compressor
M	core nozzle
MAIN	all but wing
NOZ	nozzle
OB	overboard
T	total
THP	inner (high pressure) turbine
TIP	middle (intermediate pressure) turbine
TLP	outer (low pressure) turbine
WDUCT	wing (third stream) duct
WING, WNG	wing (third stream)

Engine Symbols

BL	bleed, lbm/sec
CN	ratio of corrected speed to design corrected speed
DHT	turbine delta enthalpy, Btu/lbm
DHTC	turbine delta enthalpy (temperature corrected), $(H_{in} - H_{out})/T_{in}$, Btu/^{o}R/lbm
DP	pressure drop, $\Delta P/P$
DT	temperature change, ^{o}R
ETA	efficiency
ETAR	ram recovery, P_2/P_1
HPEXT	horsepower extracted
N	shaft speed
PCBL	fractional bleed
PCN	percent of design shaft speed
PR	pressure ratio
TFF	turbine flow function, lbm $\sqrt{^{o}R}$/(psia)(sec)
WA	airflow, lbm/sec
WF	fuel flow, lbm/sec
WG	gas flow, lbm/sec
Z	ratio of pressure ratios

TABLE 5.2-2. VARIABLES AND ERRORS

	Two-spool turbofan	Mixed-flow turbofan	Two-spool turbojet	One-spool turbojet
Variable 1	ZF	ZF	ZF	ZF
Variable 2	PCNF	PCNF	PCNF	PCNF
Variable 3	ZC	ZC	ZC	TFFLP
Variable 4	PCNC	PCNC	PCNC	----------------
Variable 5	TFFHP	TFFHP	TFFHP	----------------
Variable 6	TFFLP	TFFLP	TFFLP	----------------
Error 1	$\frac{TFHCAL - TFFHP}{TFHCAL}$	$\frac{TFHCAL - TFFHP}{TFHCAL}$	$\frac{TFHCAL - TFFHP}{TFHCAL}$	$\frac{TFLCAL - TFFLP}{TFLCAL}$
Error 2	$\frac{DHTCC - DHTCHP}{DHTCC}$	$\frac{DHTCC - DHTCHP}{DHTCC}$	$\frac{DHTCC - DHTCHP}{DHTCC}$	$\frac{DHTCF - DHTCLP}{DHTCF}$
Error 3	$\frac{TFLCAL - TFFLP}{TFLCAL}$	$\frac{TFLCAL - TFFLP}{TFLCAL}$	$\frac{TFLCAL - TFFLP}{TFLCAL}$	$\frac{P7R - P7}{P7R}$
Error 4	$\frac{DHTCF - DHTCLP}{DHTCF}$	$\frac{DHTCF - DHTCLP}{DHTCF}$	$\frac{DHTCF - DHTCLP}{DHTCF}$	----------------
Error 5	$\frac{P25R - P25}{P25R}$	$\frac{PS25 - PS55}{PS25}$	$\frac{WAF - WAC - BLF}{WAC}$	----------------
Error 6	$\frac{P7R - P7}{P7R}$	$\frac{P7R - P7}{P7R}$	$\frac{P7R - P7}{P7R}$	----------------
Matrix size	6 × 6	6 × 6	6 × 6	3 × 3

TABLE 5.2-3. INPUTS REQUIRED FOR BASIC CYCLES

Variable	Units	Definition	Two-spool turbofan	Mixed-flow turbofan	Two-spool turbojet	One-spool turbojet
PRFDS	----------	Fan pressure ratio	Yes	Yes	Yes	Yes
WAFCDS	lb/sec	Fan corrected airflow				
ETAFDS	----------	Fan efficiency				
ZFDS	----------	Design Z of fan				
PCNFDS	----------	Corrected speed of fan				
PRCDS	----------	Compressor pressure ratio				No
WACCDS	lb/sec	Compressor corrected airflow			No	
ETACDS	----------	Compressor efficiency			Yes	
ZCDS	----------	Design Z of compressor				
PCNCDS	----------	Corrected speed of compressor				
ETABDS	----------	Combustor efficiency				Yes
DPCODS	----------	Combustor pressure drop, $\Delta P/P$				Yes
T4DS	°R	Turbine inlet temperature				Yes
TFHPDS	$\frac{lb\sqrt{°R}}{(sec\ psia)}$	High-pressure-turbine flow function				No
CNHPDS	----------	Corrected speed - high-pressure turbine				No
ETHPDS	----------	Efficiency - high-pressure turbine				No
TFLPDS	$\frac{lb\sqrt{°R}}{(sec\ psia)}$	Low-pressure-turbine flow function				Yes
CNLPDS	----------	Corrected speed - low-pressure turbine				Yes
ETLPDS	----------	Efficiency - low-pressure turbine				Yes
DPDUDS	----------	$\Delta P/P$ of fan duct			No	No
DPAFDS	----------	$\Delta P/P$ of afterburner			Yes	Yes
FAN	----------	Logical variable	TRUE	TRUE	FALSE	FALSE
ISPOOL	----------	Number of spools	2	2	2	1

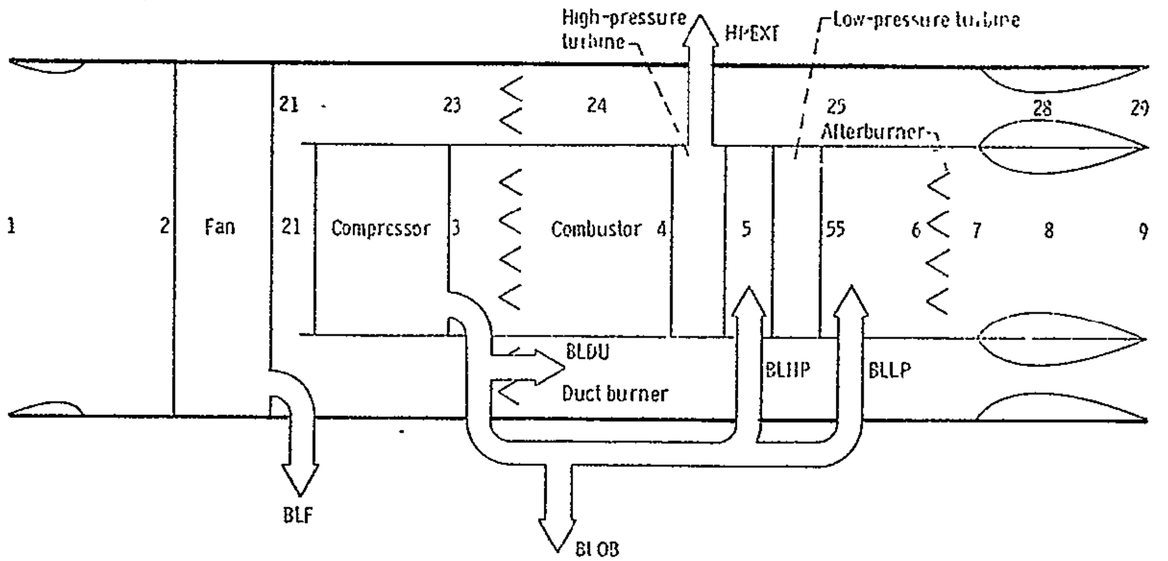


Figure 5.2-1. SCHEMATIC OF NON-MIXED FLOW DUCT BURNING AND/OR AFTERBURNING TURBOFAN

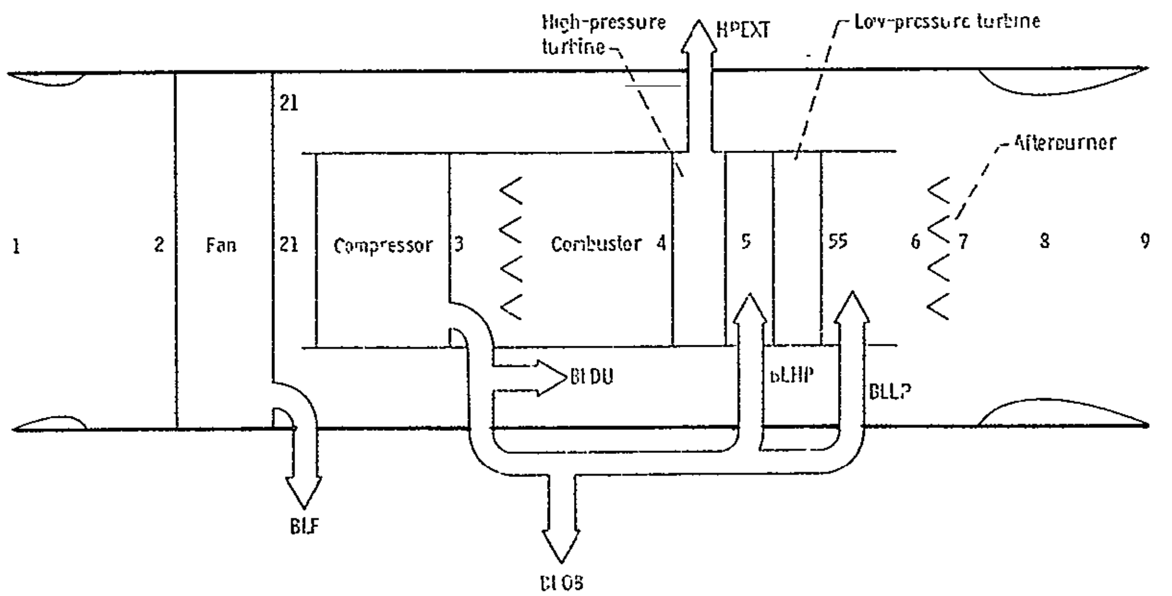


FIGURE 5.2-2. SCHEMATIC OF MIXED FLOW AFTERBURNING TURBOFAN

ORIGINAL PAGE IS OF POOR QUALITY

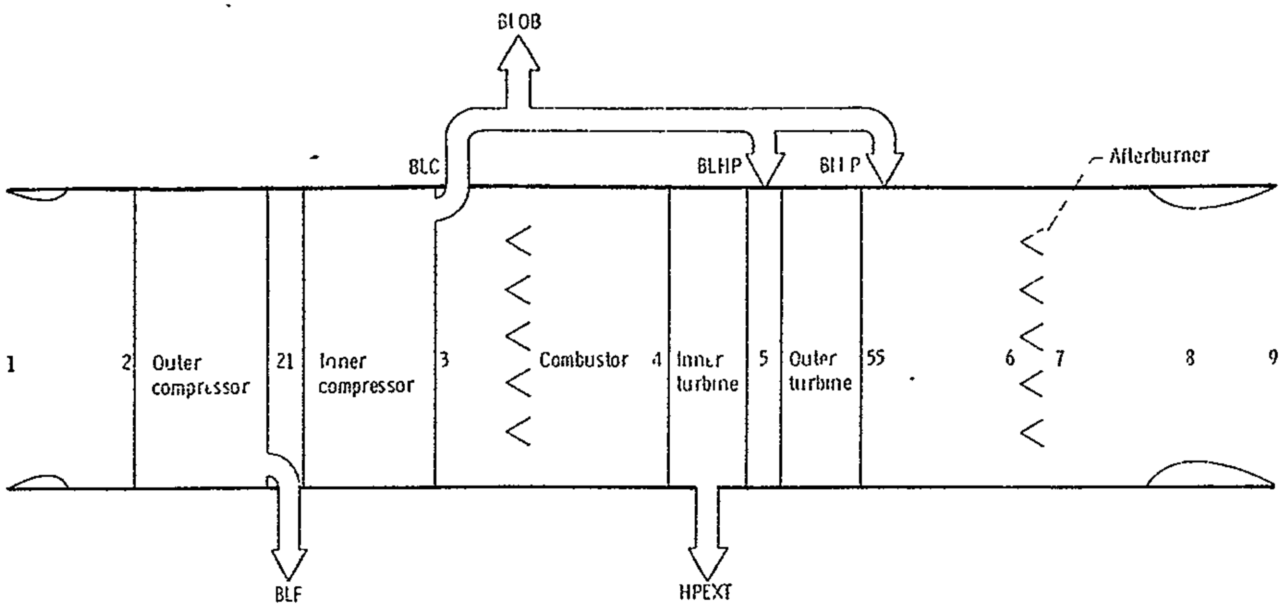


FIGURE 5.2-3. SCHEMATIC OF TWO-SPOOL TURBOJET

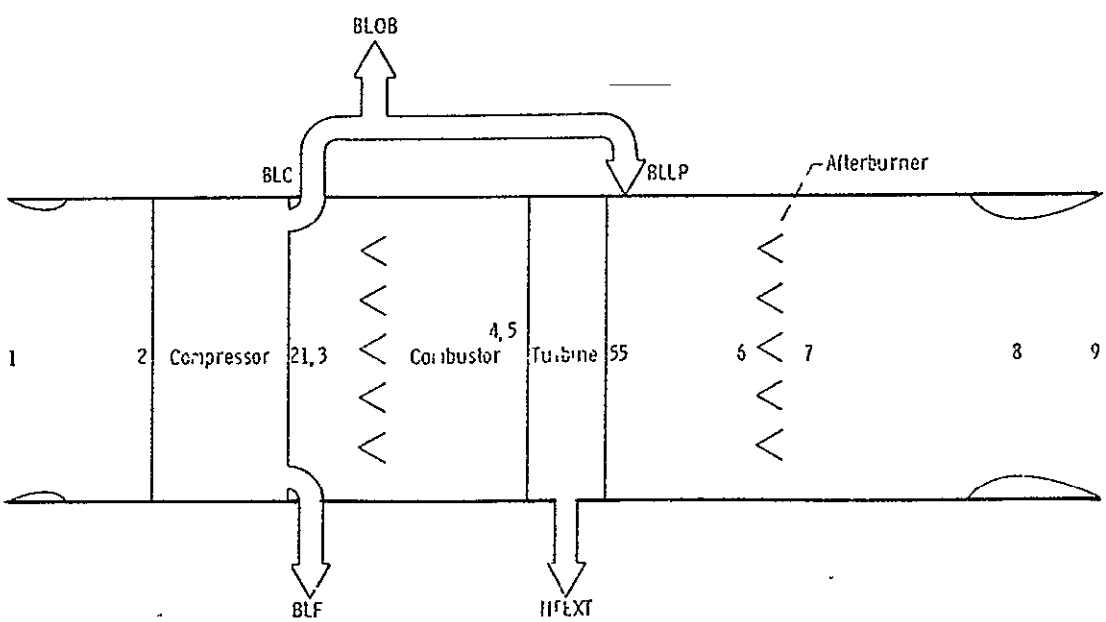


FIGURE 5.2-4. SCHEMATIC OF ONE-SPOOL TURBOJET

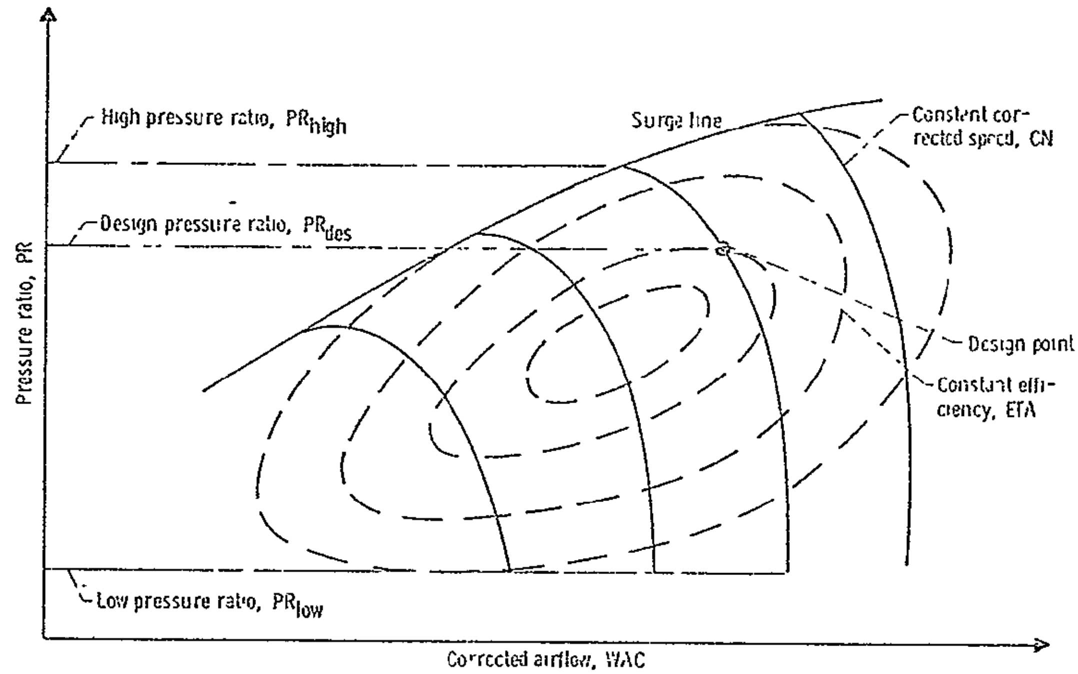


FIGURE 5.2-5. EXAMPLE OF A SPECIFIC FAN-COMPRESSOR MAP

$Z = (PR_{(} - PR_{low})/(PR_{high} - PR_{low})$

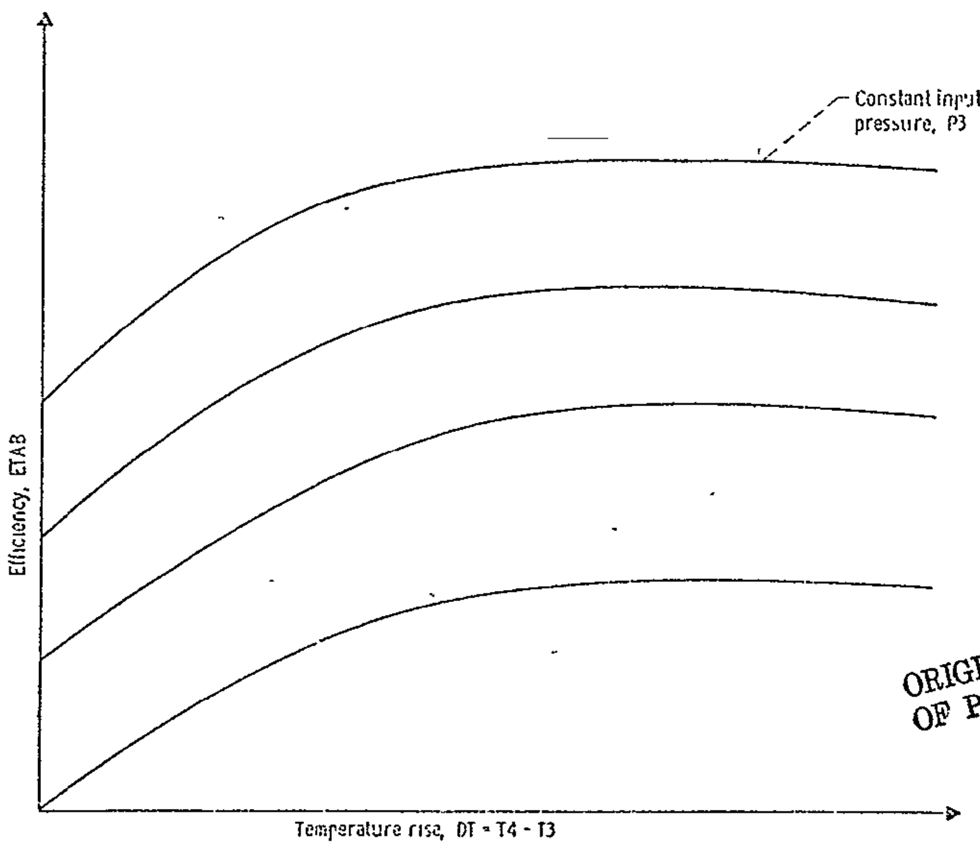


FIGURE 5.2-6. EXAMPLE OF COMBUSTOR MAP

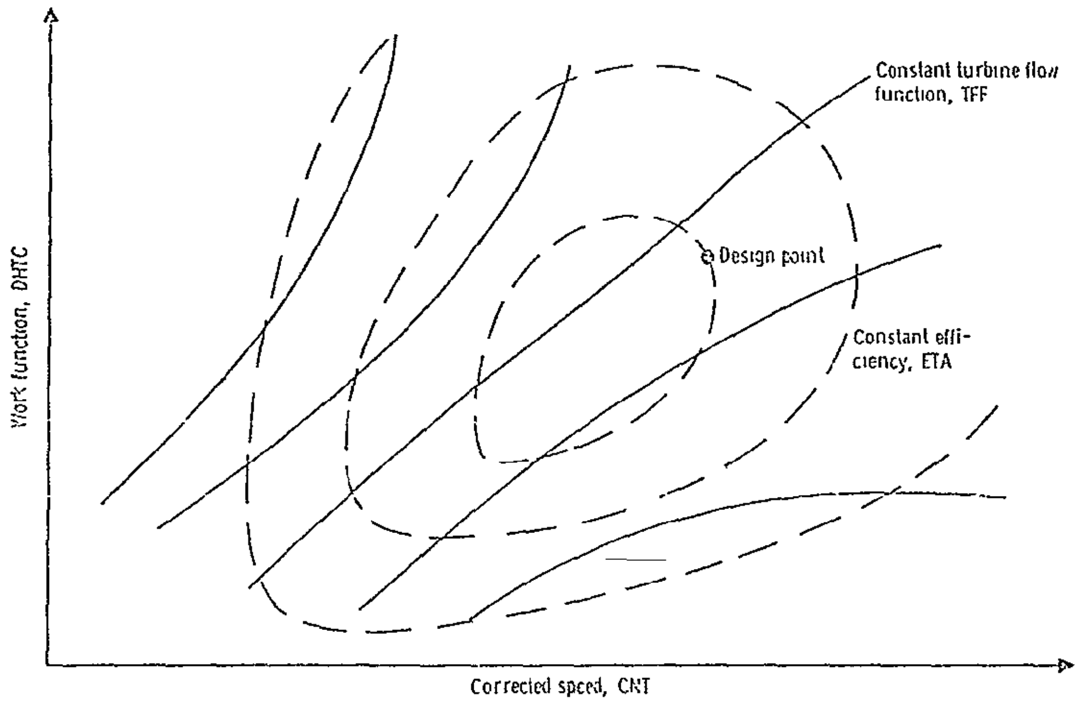
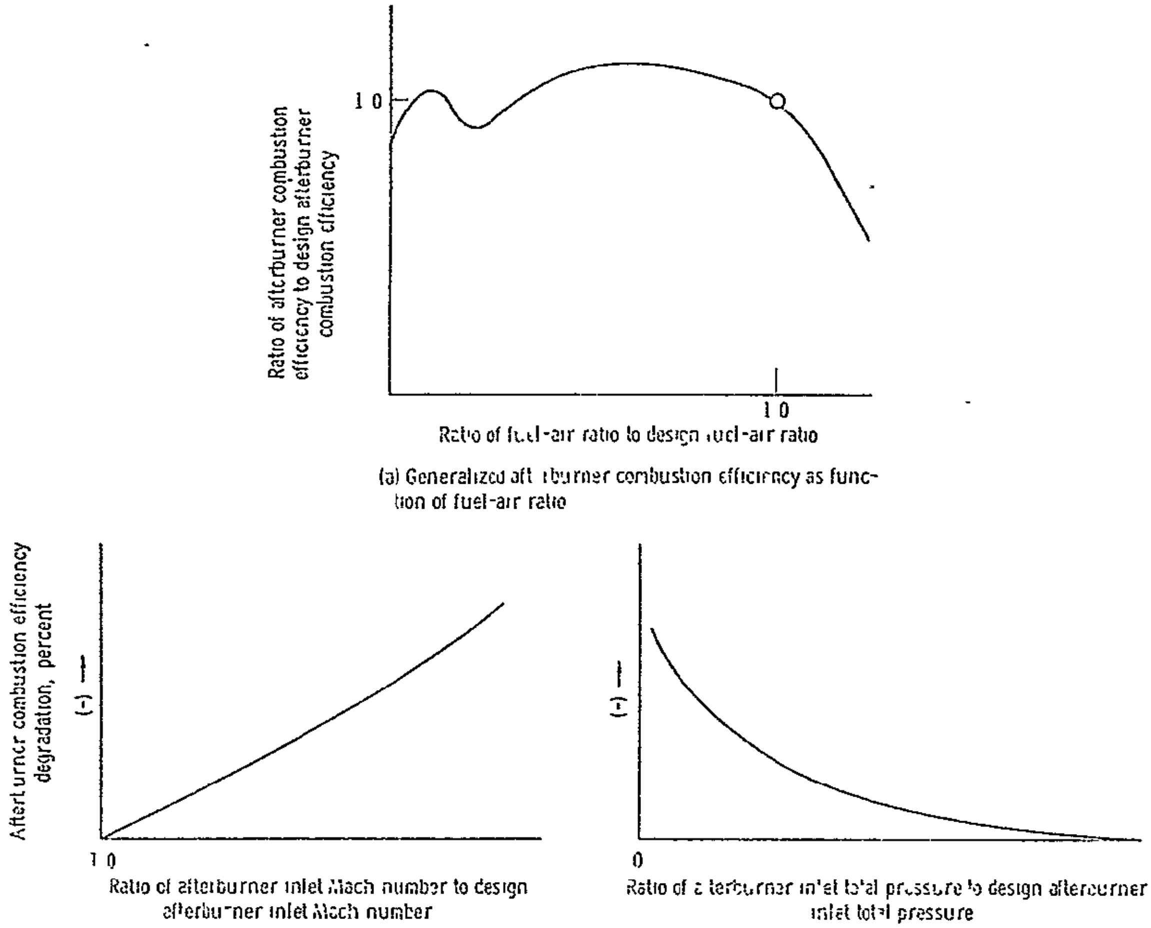


FIGURE 5.2-7. EXAMPLE OF SPECIFIC TURBINE MAP



(a) Generalized afterburner combustion efficiency as function of fuel-air ratio

(b) Efficiency correction factor as function of afterburner inlet Mach number

(c) Efficiency correction factor as function of afterburner inlet total pressure ratio

FIGURE 5.2-8. EXAMPLE OF A GENERALIZED AFTERBURNER COMBUSTION EFFICIENCY PERFORMANCE MAP

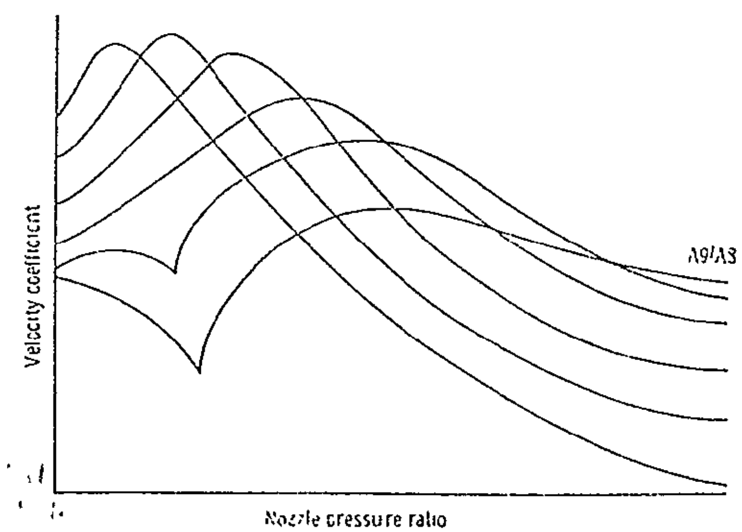


FIGURE 5.2-9. PERFORMANCE MAP FOR NOZZLE, GIVING VELOCITY COEFFICIENT

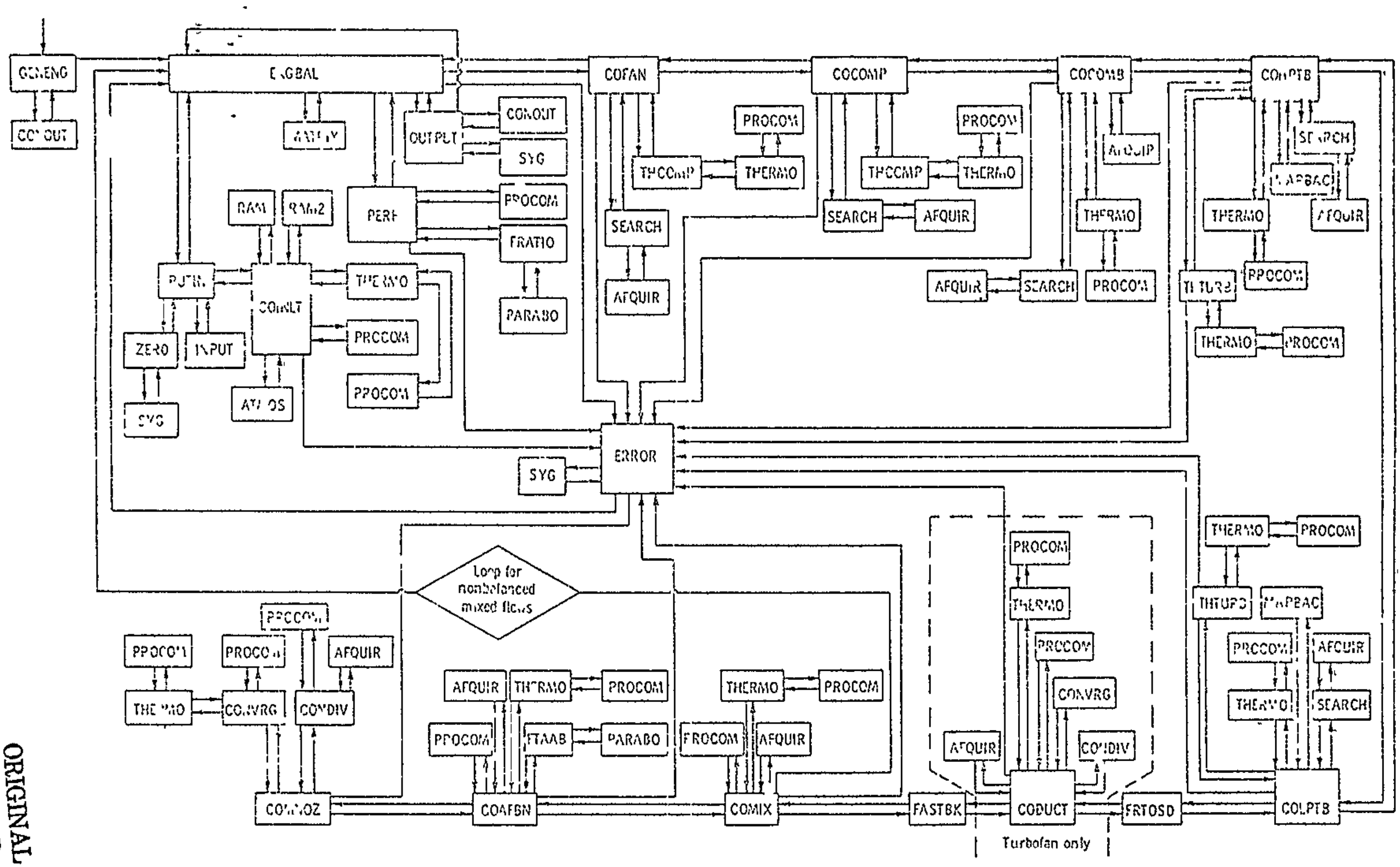


FIGURE 5.2-10. FLOW CHART FOR GENENT COMPUTER PROGRAM

TABLE OF CONTENTS FOR SECTION 5.3, PROGRAM GENENG II

5.3 PROGRAM GENENG II: A PROGRAM FOR CALCULATING DESIGN AND OFF-DESIGN PERFORMANCE OF TWO- AND THREE-SPOOL TURBOFANS WITH AS MANY AS THREE NOZZLES

Program GENENG II was developed by Fishbach and Koenig of NASA's Lewis Research Center. Original program documentation of the program is provided in Reference 1. The discussion of program GENENG II presented below follows Reference 1. The GENENG II Program is a derivative of GENENG (GENeralized ENGine). GENENG, which is capable of calculating steady-state design and off-design performance of turbofan and turbojet engines was evolved from SMOTE (SiMulation Of Turbofan Engine) which was developed by the Turbine Engine Division of the Air Force Aero Propulsion Laboratory, Wright-Patterson Air Force Base, Ohio.

GENENG II calculates design and off-design jet engine performance for existing or theoretical turbofan engines with two or three spools and with one, two, or three nozzles. In addition, aft fan engines can be calculated. Nine basic turbofan engines can be calculated without any programming changes:

1. Three-spool, three-stream engine
2. Two-spool, three-stream boosted fan engine
3. Two-spool, three-stream, supercharged compressor engine
4. Three-spool, two-stream engine
5. Two-spool, two-stream engine
6. Three-spool, three-stream, aft fan engine
7. Two-spool, three-stream, aft fan engine
8. Two-spool, two-stream, aft fan engine
9. Three-spool, two-stream, aft fan engine

The first three of these engines are likely candidates for a STOL aircraft with internally blown flaps. By examining the methods used to simulate these engines, other engine types may be simulated. As examples, a boosted aft fan engine with two streams would simulate a high bypass ratio engine where the core and tip portions of the fan have different component performance maps; a boosted fan, two-stream engine could be simulated (JT9D type); or supercharged compressor, two-stream engines could be studied The number of possibilities are too many to enumerate, being determined by the user's knowledge of program GENENG and the elements of engine design.

5.3.1 Introduction

Program GENENG II is a derivative of program GENENG described in Section 5.2. Program GENENG, in turn, is a derivative of the References 2 and 3 Air Force SMOTE program. GENENG satisfies a need for calculating the performance of two- or three-spool turbofan engines with as many as three nozzles (or airstreams). An example of this type of engine would be one in which a fan is used to compress all the air, some of which is expanded through a separate nozzle to produce thrust. The remaining air passes through a compressor,

ORIGINAL PAGE IS
OF POOR QUALITY

after which some air is put into a wing duct and expelled over the wing flaps (an internally blown flap). The remaining air passes through another compressor into a combustor; is heated and expanded through three turbines, each of which drives one of the compressors; and is then expelled out the third (main) nozzle producing more thrust. This engine type is under consideration for STOL aircraft and until the development of GENENG II, off-design performance calculations were difficult to attain.

GENENG II was developed to provide the capability to study this engine type. Once this capability had been achieved, it was realized (Reference 1) that many other engine types could be simulated by building simple options into the code and modifying the input data to the program. As an example, the fan and first compressor in the engine just described could be physically attached and driven by one turbine (the so-called "boosted turbofan"), or the fan could be put at the rear of the engine (an aft fan). Thus, GENENG II has become a versatile program with many engine design options built in internally. These are described in the next section, Section 5.3.2. The original Fishbach and Koenig GENENG II program was written for the IBM 7094 computer. The GENENG II program contained in the ODIN/RLV program library is a CDC 6600 version constructed at the Naval Air Development Center by Robert Leko.

5.3.2 Engine Types

All thermodynamic properties of air and gas are calculated by considering variable specific heats and no dissociation. Curve fitted air and gas property tables of Reference 5 are used.

5.3.2.1 Type a - Three-Spool, Three-Stream Turbofan

The basic engine, a three-spool, three-stream turbofan, of which all other engine types are treated as variations, is shown in Figure 5.3-1. Free stream conditions exist at Station 1. The conditions at Station 2 are determined by flight conditions and inlet recovery. GENENG compressor maps work with corrected values of airflow. At the entrance to the fan, the corrected airflow, $WA_{F,c}$ is

$$WA_{F,c} = \frac{WA_F\sqrt{T_2/T_{518.668}}}{P_2/P_{SLS}} \tag{5.3.1}$$

where P_2 and P_{SLS} are atmospheres and P_{SLS} equals 1.0. All symbols are defined in Table 5.2-1 of Section 5.2. Some symbols are formed as the combination of other symbols; thus WA is airflow; F is for fan, and c, when following a component symbol means corrected. Station numbers are defined on the appropriate figure.

All the fan air WA_F is compressed by the fan giving rise to conditions at station 22. The power required to do this is

$$\text{Fan power} = WA_F \times (H_{22} - H_2) \qquad (5.3.2)$$

Some fan air may be lost to the cycle as fan bleed Bl_F, which is expressed as a fraction of the fan airflow

$$Bl_F = PC_{Bl,F} \times WA_F \qquad (5.3.3)$$

The corrected airflow into the intermediate compressor is

$$WA_{I,c} = \frac{WA_I\sqrt{T_{22}/T_{518.668}}}{P_{22}/1.0} \qquad (5.3.4)$$

The remaining air goes through the fan duct where some leakage from the core air may also enter; see Equation 5.3.16.

$$WA_D = WA_F - Bl_F - WA_I + Bl_{DU} \qquad (5.3.5)$$

This air, which may be heated by a duct burner to a temperature T_{24}, undergoes a pressure drop

$$P_{25} = P_{24} \times \left[1 - \left(\frac{\Delta P}{P}\right)_{DUCT}\right] \qquad (5.3.6)$$

The air would have been heated by the addition of fuel, which can be expressed as a fuel-air ratio so that

$$WG_{24} = WA_{23} \times [1 + (f/a)_{23}] \qquad (5.3.7)$$

The gas is then expanded through a nozzle (Station 29) to produce thrust. The bypass ratio is defined by

$$BYPASS = \frac{WA_D}{WA_I} \qquad (5.3.8)$$

To this point the analysis is similar to the GENENG discussion of Section 5.2. Now, however, the air going into the intermediate compressor is compressed to the conditions at Station 21. The power required is

$$\text{Intermediate-compressor power} = WA_I \times (H_{21} - H_{22}) \qquad (5.3.9)$$

The conditions at Station 21 are the same as those at Station 32, which is the entrance to the wing duct as the third streampath is called. The airflow entering this duct is called Bl_I, meaning intermediate bleed flow, and is expressed as a fraction $PC_{Bl,I}$ of the total airflow at Station 21.

$$Bl_I = PC_{Bl,I} \times WA_I \qquad (5.3.10)$$

ORIGINAL PAGE IS
OF POOR QUALITY

The remainder of the air enters the core compressor

$$WA_C = WA_I - Bl_I \tag{5.3.11}$$

and

$$WA_{C,c} = \frac{WA_C \times \sqrt{T_{21}/T_{518.668}}}{P_{21}/1.0} \tag{5.3.12}$$

The air entering the wing duct experiences a pressure drop

$$P_{36} = P_{32} \times \left[1 - \left(\frac{\Delta P}{P}\right)_{WING}\right] \tag{5.3.13}$$

and then passes through a nozzle (Station 39) to produce additional thrust. The air continuing on through the core is compressed to conditions at Station 3. The power required is

$$\text{Core compressor power} = WA_C \times (H_3 - H_{21}) = WA_3 \times (H_3 - H_{21}) \tag{5.3.14}$$

Some core bleed air Bl_C may be used for turbine cooling. Some of the air is put back into the cycle into each of the three turbines, and some is lost to the cycle as overboard bleed or leakage into the fan duct.

$$Bl_C = BC_{Bl,C} \times WA_3 \tag{5.3.15}$$

$$Bl_{DU} = PC_{Bl,DU} \times Bl_C \tag{5.3 16}$$

$$Bl_{OB} = PC_{Bl,OB} \times Bl_C \tag{5.3.17}$$

$$Bl_{HP} = PC_{Bl,HP} \times Bl_C \tag{5.3.18}$$

$$Bl_{IP} = PC_{Bl,IP} \times Bl_C \tag{5.3.19}$$

$$Bl_{LP} = PC_{Bl,LP} \times Bl_C \tag{5.3.20}$$

Since $Bl_{DU} + Bl_{OB} + Bl_{HP} + Bl_{IP} + Bl_{LP} = Bl_C$, the sum of $PC_{Bl,DU}$, $PC_{Bl,OB}$, $PC_{Bl,HP}$, $PC_{Bl,IP}$, and $PC_{Bl,LP}$ must be equal to 1. The remaining air is

$$WA_4 = WA_3 - Bl_C \tag{5.3.21}$$

and is heated to a turbine inlet temperature T_4 and goes through a combustor pressure drop $(\Delta P/P)_{COMB}$. The fuel required to do this is expressed as a fuel-air ratio $(f/a)_4$ so that the gas entering the first turbine WG_4 can be expressed as

$$WG_4 = WA_4 \times [1 + (f/a)_4] \tag{5.3.22}$$

This gas is then expanded through this high pressure turbine to conditions at Station 50. The enthalpy at Station 50 is first calculated by making a power balance since this turbine drives the core compressor and supplies any work extracted (HPEXT). By using Equation 5.3.14

$$WG_4 \times (H_4 - H_{50}) = WA_3 \times (H_3 - H_{21}) + HPEXT \tag{5.3.23}$$

In addition, the physical speeds must match

$$N_{HP,TURBINE} = N_{COMP} \tag{5.3.24}$$

If high pressure turbine bleed air Bl_{HP} is added into the cycle at this point, H_{50} must be readjusted

$$H_{50} = \frac{(Bl_{HP} \times H_3) + WG_4 H_{50}}{WG_4 + Bl_{HP}} = \frac{(Bl_{HP} \times H_3) + WG_4 H_{50}}{WG_{50}} \tag{5.3.25}$$

Similarly,

$$WG_{50} \times (H_{50} - H_5) = WA_I \times (H_{21} - H_{22}) \tag{5.3.26}$$

$$N_{IP,TURBINE} = N_{INT\ COMP} \tag{5.3.27}$$

$$H_5 = \frac{(Bl_{IP} \times H_3) + WG_5 H_{55}}{WG_5 + Bl_{LP}} = \frac{(Bl_{LP} \times H_3) + WG_5 H_{55}}{WG_{55}} \tag{5.3.28}$$

$$WG_5 \times (H_5 - H_{55}) = WA \times (H_{22} - H_2) \tag{5.3.29}$$

$$N_{LP,TURBINE} = N_{FAN} \tag{5.3.30}$$

$$H_{55} = \frac{(Bl_{LP} \times H_3) + WG_5 H_{55}}{WG_5 + Bl_{LP}} = \frac{(Bl_{LP} \times H_3) + WG_5 H_{55}}{WG_{55}} \tag{5.3.31}$$

The gas flow WG_{55} then may be heated by an afterburner to a gas temperature T_7 and may undergo a pressure drop.

$$P_7 = P_6 \left[1 - \left(\frac{\Delta P}{P}\right)_{AFTERBURNER}\right] \qquad (5.3.32)$$

The gas flow would be increased by any fuel burned.

$$WG_7 = WG_{55} + WFA \qquad (5.3.33)$$

The gas is then expanded through the nozzle (Station 9) to produce the remainder of the total engine thrust.

5.3.2.2 Type b - Two-Spool, Three-Stream, Boosted Fan Turbofan

From Figure 5.3-2 it is apparent why the three-spool, three-stream engine can be modified to represent the other types presented herein. The only difference between engine b and engine a is that the intermediate compressor is physically attached to the fan in terms of speed and the combination is driven by one turbine (the low pressure turbine). The thermodynamic calculation changes are that the speeds are attached.

$$N_{INT\ COMP} = N_{FAN} \qquad (5.3.34)$$

The power of the low pressure turbine is now

$$WG_{50} \times (H_{50} - H_{55}) = WA_F \times (H_{22} - H_2) + WA_I \times (H_{21} - H_{22}) \qquad (5.3.35)$$

$PC_{B1,IP}$ must be zero and H_{55} is readjusted by

$$H_{55} = \frac{(B1_{LP} \times H_3) + WG_{50} H_{55}}{WG_{50} + B1_{LP}} \qquad (5.3.36)$$

This type of engine is of interest because it might be created by adding a new boosted-fan turbine combination to an existing core. If the third air-stream is deleted (see engine e) and ductburner and afterburner are removed, engine b becomes a two-spool, two-stream turbofan of the type represented by the General Electric CF6 and Pratt and Whitney JT9D turbofan, both of which have booster stages on the fan.

5.3.2.3 Type c - Two-Spool, Three-Stream Supercharged Compressor Turbofan

Engine c is shown in Figure 5.3-3. Here, the intermediate and core compressors have been physically attached. For programming reasons, the combination is driven by the intermediate pressure turbine. The calculation procedure bypasses the routine which calculates high pressure turbine performance but transfers the turbine performance data from this routine into that of the intermediate pressure turbine to represent the turbine performance. Since the intermediate pressure turbine speed is set by the speed of the intermediate compressor which also sets the speed of the combination of the compressors, this procedure was necessary.

$$N_{COMP} = N_{INT\ COMP} \tag{5.3.37}$$

$$WG_{50} \times (H_{50} - H_5) = WA_I \times (H_{21} - H_{22}) + WA_C \times (H_3 - H_{21}) + HPEXT \tag{5.3.38}$$

$PC_{B1,HP}$ must be zero and H_5 is readjusted by

$$H_5 = \frac{(B1_{IP} \times H_3) + WG_{50}H_5}{WG_{50} + B1_{IP}} \tag{5.3.39}$$

5.3.2.4 Type d - Three-Spool, Two-Stream Turbofan

Engine d, shown in Figure 5.3-4, is presently in existence (Rolls Royce RB 211) and differs from the reference engine in that all the air entering the intermediate compressor also enters the inner compressor. For this reason, the only change necessary to run this engine is to set $PC_{B1,I}$ equal to zero.

5.3.2.5 Type e - Two-Spool, Two-Stream Turbofan

Engine e is the typical turbofan and is shown in Figure 5.3-5. To simulate this engine, it is necessary to have the air go through the intermediate compressor at a pressure ratio of 1.0 and an efficiency of 1.0 and to bypass the intermediate pressure turbine calculations. A logical control has been built into the program to do this. At the same time, $PC_{B1,I}$ must be set equal to zero. By using this option, GENENG II can be used to replace its original version GENENG, Reference 4, in calculating turbofan performance. It cannot, however, do turbojet calculations (two-spool, one stream or one-spool, one stream engines). As mentioned earlier, boosted fan, two-spool, two stream engines can be calculated by setting $PC_{B1,I}$ equal to zero in engine b.

5.3.2.6 Type f - Three-Spool, Three-Stream Aft Fan Turbofan

The three-spool, three-stream aft fan engine is shown in Figure 5.3-6. Thermodynamically, the only difference between this and the reference engine is that the intermediate compressor sees the same conditions at its entrance as does the fan (conditions at Station 2; both inlets assumed to have the same performance). This is accomplished by setting a logical control variable AFTFAN to be true. The power of the intermediate pressure turbine would be

$$WG_{50} \times (H_{50} - H_5) = WA_I \times (H_{21} - H_2) \qquad (5.3.40)$$

Each of the aft fan engines has a counterpart in the front fan engines, the only difference being that the intermediate compressor (or in the case of engine h, a two-spool, two-stream aft fan engine, the compressor) sees free-stream conditions. These engines and their counterparts are described in the following sections.

5.3.2.7 Type g - Two-Spool, Three-Stream Aft Fan Turbofan

Engine g, a counterpart of engine c (Figure 5.3-3) is shown in Figure 5.3-7. The power balance would be

$$WG_{50} \times (H_{50} - H_5) = WA_I \times (H_{32} - H_2) + WA_C \times (H_3 - H_{32}) \qquad (5.3.41)$$

5.3.2.8 Type h - Two-Spool, Two-Stream Aft Fan Turbofan

Engine h, a counterpart of engine e, Figure 5.3-5, is shown in Figure 5.3-8. The power balance would be

$$WG_{50} \times (H_{50} - H_5) = WA_C \times (H_3 - H_2) \qquad (5.3.42)$$

5.3.2.9 Type i - Three-Spool, Two-Stream Aft Fan Turbofan

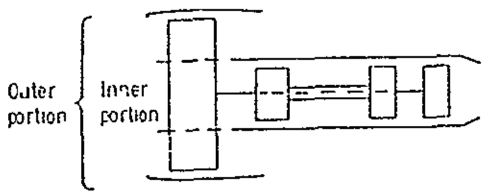
Engine i, a counterpart of engine d, Figure 5.3-4, is shown in Figure 5.3-9. The power balance would be

$$WG_{50} \times (H_{50} - H_5) = WA_I \times (H_{21} - H_2) \qquad (5.3.43)$$

5.3.2.10 Other Engines

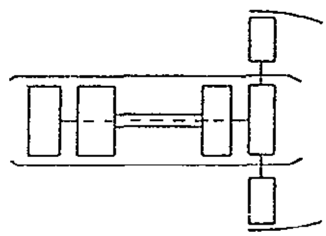
By using imagination in conjunction with the engines illustrated, the reader can determine other engine types which can be simulated. An obvious one is a supercharged compressor, two-stream turbofan which is a derivative of engine c. the only change necessary being setting $PC_{B1,I} = 0$ In addition, all engines illustrated could be run as mixed-flow engines eliminating the fan duct nozzle, Reference 4.

An interesting engine more difficult to be simulated is a high bypass ratio turbofan (two streams) where the outer and inner portions of the fan are represented by different performance maps. As can be seen by the following sketches, this engine can be simulated by a boosted aft fan engine. When AFTFAN is true, the second spool sees free-stream conditions. When the fan and intermediate spool are attached, the physical rotational speeds of the aft fan (outer portion of fan) and the second spool (inner portion of fan) will be the same. Both are driven off the same turbine.



(a)

The high bypass ratio turbofan (sketch a) can be simulated by a boosted aft fan engine (sketch b).



(b)

5.3.3 Balancing Technique

An off-design engine cycle calculation requires satisfying various matching constraints (rotational speeds, airflows, compressor and turbine work functions and nozzle flow functions) at each specified operating condition. GENENG II internally searches for compressor and turbine operating points that will satisfy the constraints. It does this by generating differential errors caused by small changes in the independent variables. The program then uses a matrix that is loaded with the differential errors to solve for the zero error condition. The procedure employed is the Newton-Raphson iteration technique.

For a three-spool engine, a solution for a set of nine simultaneous linear equations is obtained, for other types, fewer equations are used. The nine independent variables selected are

1. ZF - Ratio of pressure ratios of fan compressor along a speed line,

$$ZF = \frac{\text{(Pressure ratio along speed line) - (Low pressure ratio on speed line)}}{\text{(High pressure ratio on speed line)-(Low pressure ratio on speed line)}}$$

2. PCNF or T4 - Per cent fan speed or turbine inlet temperature

3. ZI - Ratio of Pressure ratios of intermediate compressor along a speed line (Calculated the same as ZF)

4. PCNI - Per cent intermediate compressor speed

5. ZC - Ratio of pressure ratios of inner compressor along a speed line (calculated same as ZF)

6. PCNC or T4 - Per cent inner compressor speed or turbine inlet temperature

7. TFFHP - High pressure turbine flow function $WG_4\sqrt{T_4/P_4}$

8. TFFIP - Intermediate pressure turbine flow function, $WG_{50}\sqrt{T_{50}/P_{50}}$

9. TFFLP - Low pressure turbine flow function, $WG_5\sqrt{T_5/P_5}$

The program initially selects new (perturbed) values for the variables, based on the design values. It is then possible to proceed through the entire engine cycle calculations, where up to nine errors are generated. The initial values of the nine (or less) variables and nine (or less) errors are base values. Solution method is outlined in Section 5.2. The most often used independent variables and the differential errors for each of the nine engine types capable of being run on GENENG II are listed in Table 5.3-1.

5.3.4 Choice of Component Maps - Scaling Laws

The component maps and scaling lows follow the techniques employed in GENENG, Section 5.2. A discussion of these techniques has been presented in Section 5.2.4.

5.3.5 Means of Specifying Mode of Engine Operations

The methods for specifying mode of engine operations is similar to that described in Section 5.2.5 of the GENENG discussion. However, A38, the wing nozzle throat area is available as a parameter for changing engine operation in GENENG II. Again as in GENENG the Huff input routine is not available on the CDC 6600 program. The use of DIALOG, Section 2, does permit all arithmetic and symbolic input operations available in the Huff input package when GENENG II is employed in an ODIN/RLV simulation.

Inputs for operating the GENENG II program are summarized in Table 5.3-2.

5.3.6 GENENG II Subroutine Functions and Descriptions

A flow chart of the computer program with the subroutines is shown in Figure 5.3-10. The functions of the GENENG II subroutines have been mostly described in Section 5.2.6. Several additional subroutines are present in the GENENG II program, as described below.

GEN2 Dummy main program to initiate the calculations and cause the input of the controlled output variables. Because of the looping between subroutines, control is never transferred back to this routine.

COINTC Uses BLOCK DATA to perform intermediate compressor calculations.

INTDUM Makes intermediate compressor not change air conditions for engines e and h.

WDUCT Performs third-stream (wing) duct calculations (not used in two-stream engines).

COIPTB Uses BLOCK DATA to perform intermediate turbine calculations (not used in engines b, e, and h).

OVELAY DUMMY routine to restore working part of program to core when using overlay

IPTDAT Performance data for intermediate turbine map (BLOCK DATA)

5.3.7 Symbols

Symbols for the GENENG II discussion are the same as those symbols employed in the GENENG discussion, and are listed in Table 5.2-1 of Section 5.2.

REFERENCES:

1. Fishbach, Laurence H. and Koenig, Robert W., GENENG II - A Program for Calculating Design and Off-Design Performance of Two- and Three-Spool Turbofans with as Many as Three Nozzles, NASA TN D-6553, 1972.

2. McKinney, John S., Simulation of Turbofan Engine, Part I, Description of Method and Balancing Technique, Report AFAPL-TR-67-125, Air Force Systems Command, November 1967. (Available from DDC as AD-825197).

3. McKinney, John S., Simulation of Turbofan Engine, Part II, User's Manual and Computer Program Listing, Report AFAPL-TR-67-125, Air Force Systems Command, November 1967. (Available from DDC as AD-825198).

4. Koenig, Robert W. and Fishbach, Laurence H., GENENG - A Program for Calculating Design and Off-Design Performance for Turbojet and Turbofan Engines, NASA TN D-6552, February 1972.

5. Keenan, Joseph H. and Kaye, Joseph, Gas Tables, John Wiley and Sons, Inc., 1948.

TABLE 5.3-1. VARIABLES AND ERRORS

	Engine designation								
	a	b	c	d	e	f	g	h	i
	Number of spools								
	3	2	2	3	2	3	2	2	3
	Number of streams								
	3	3	3	2	2	3	3	2	2
	Turbofan	Boosted fan	Supercharged compressor	Turbofan		Aft fan	Supercharged compressor	Aft fan	
Variable 1	ZF	ZF	ZF	ZF	ZF	ZF	ZF	ZF	ZF
Variable 2	PCNF	PCNF	PCNF	PCNF	PCNF	PCNF	PCNF	PCNF	PCNF
Variable 3	ZC	ZC	ZC	ZC	ZC	ZC	ZC	ZC	ZC
Variable 4	PCNC	PCNC	PCNI	PCNC	PCNC	PCNC	PCNI	PCNC	PCNC
Variable 5	TFFHP	TFFHP	TFFIP	TFFHP	TFFHP	TFFHP	TFFIP	TFFHP	TFFHP
Variable 6	TFFLP	TFFLP	TFFLP	TFFLP	TFFLP	TFFLP	TFFLP	TFFLP	TFFLP
Variable 7	ZI	ZI	ZI	ZI	------	ZI	ZI	------	ZI
Variable 8	PCNI	------	----------------	PCNI	------	PCNI	------------	------	PCNI
Variable 9	TFFIP	------	----------------	TFFIP	------	TFFIP	------------	------	TFFIP
Error 1	$\frac{\text{TFHCAL - TFFHP}}{\text{TFHCAL}}$	(a)	$\frac{\text{TFICAL - TFFIP}}{\text{TFICAL}}$	(a)	(a)	(a)	(b)	(a)	(a)
Error 2	$\frac{\text{DHTCC - DHTCHP}}{\text{DHTCC}}$	(a)	$\frac{\text{DHTIC - DHTCIP}}{\text{DHTIC}}$	(a)	(a)	(a)	(b)	(a)	(a)
Error 3	$\frac{\text{TFLCAL - TFFLP}}{\text{TFLCAL}}$	(a)	(a)	(a)	(a)	(a)	(a)	(a)	(a)
Error 4	$\frac{\text{DHTCF - DHTCLP}}{\text{DHTCF}}$	(a)	(a)	(a)	(a)	(a)	(a)	(a)	(a)
Error 5	$\frac{\text{P25R - P25}}{\text{P25R}}$	(a)	(a)	(a)	(a)	(a)	(a)	(a)	(a)
Error 6	$\frac{\text{P7R - P7}}{\text{P7R}}$	(a)	(a)	(a)	(a)	(a)	(a)	(a)	(a)
Error 7	$\frac{\text{P38R - P38}}{\text{P38R}}$	(a)	(a)	$\frac{\text{WAC - WAI}}{\text{WAC}}$	------	(a)	(a)	------	(c)
Error 8	$\frac{\text{TFICAL - TFFIP}}{\text{TFICAL}}$	------	----------------	(a)	------	(a)	------------	------	(a)
Error 9	$\frac{\text{DHTIC - DHTCIP}}{\text{DHTIC}}$	------	----------------	(a)	------	(a)	------------	------	(a)
Matrix size	9 × 9	7 × 7	7 × 7	9 × 9	6 × 6	9 × 9	7 × 7	6 × 6	9 × 9

[a]Same as error for engine a
[b]Same as error for engine c
[c]Same as error for engine d

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE 5.3-2. INPUTS REQUIRED FOR BASIC CYCLES

Variable	Units or type	Definition	a	b	c	d	e	f	g	h	i
			Engine designation								
			Number of spools								
			3	2	2	3	2	3	2	2	3
			Number of streams								
			3	3	3	2	2	3	3	2	2
			Turbo-fan	Boosted fan	Super-charged com-pressor	Turbofan		Aft fan	Super-charged com-pressor	Aft fan*	
PRFDS	------------	Fan pressure ratio	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
WAFCDS	lb/sec	Fan corrected airflow									
ETAFDS	------------	Fan efficiency									
ZFDS	------------	Design Z of fan									
PCNFDS	------------	Corrected speed of fan									
PRIDS	------------	Intermediate pressure ratio					No			No	
WAICDS	lb/sec	Intermediate corrected airflow					Yes			Yes	
ETAIDS	------------	Intermediate efficiency					No			No	
ZIDS	------------	Design Z of intermediate compressor					No			No	
PCNIDS	------------	Corrected speed of intermediate compressor		No			No			No	
PRCDS	------------	Compressor pressure ratio		Yes			Yes			Yes	
PCBLIDS	------------	Fraction of air into third duct				Zero	Zero			Zero	Zero
ETACDS	------------	Compressor efficiency				Yes	Yes			Yes	Yes
ZCDS	------------	Design Z of compressor									
PCNCDS	------------	Corrected speed of compressor			No				No		
ETABDS	------------	Combustor efficiency			Yes				Yes		
DPCODS	------------	Combustor pressure drop, ΔP, P			Yes				Yes		
T4DS	°R	Turbine inlet temperature			Yes				Yes		
TFHPDS	$\frac{lb\sqrt{°R}}{(sec)(psia)}$	High-pressure-turbine flow function			No				No		
CNHPDS	------------	High-pressure-turbine corrected speed			No				No		
ETHPDS	------------	High-pressure-turbine efficiency			No				No		
TFIPDS	$\frac{lb\sqrt{°R}}{(sec)(psia)}$	Intermediate-turbine work function		No	Yes		No		Yes	No	
CNIPDS	------------	Intermediate-pressure-turbine corrected speed		No			No			No	
ETIPDS	------------	Intermediate-pressure-turbine efficiency		No			No			No	
TFLPDS	$\frac{lb\sqrt{°R}}{(sec)(psia)}$	Low-pressure-turbine flow function		Yes			Yes			Yes	
CNLPDS	------------	Intermediate-pressure-turbine corrected speed									
ETLPDS	------------	Intermediate-pressure-turbine efficiency									
DPDUCS	------------	Fan pressure drop ΔP/P									
DPWGDS	------------	Wing duct pressure drop ΔP/P				No	No			No	No
DPAFDS	------------	Afterburner pressure drop ΔP/P				Yes	Yes			Yes	Yes
FINFANTOMIDDLE	Logical	Boosted fans	F	F	F	F	F	F	F.	F	F.
FANMID[illegible]TOCOMP	Logical	Supercharged compressors		F	T			F	F	F	F
DU[illegible]SPOOL	Logical	No intermediate spool		F	F			F	F	T	F
AFTFAN	Logical	Aft-fan engines		F	F			T	T	T	T

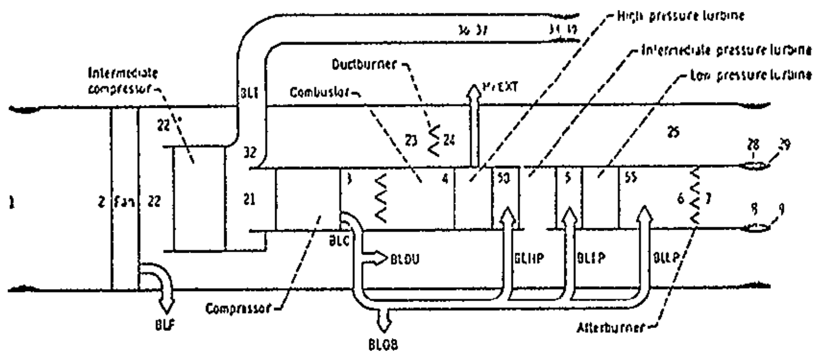


Figure 5.3-1. Three-Spool, Three-Stream Turbofan Engine (Type a)

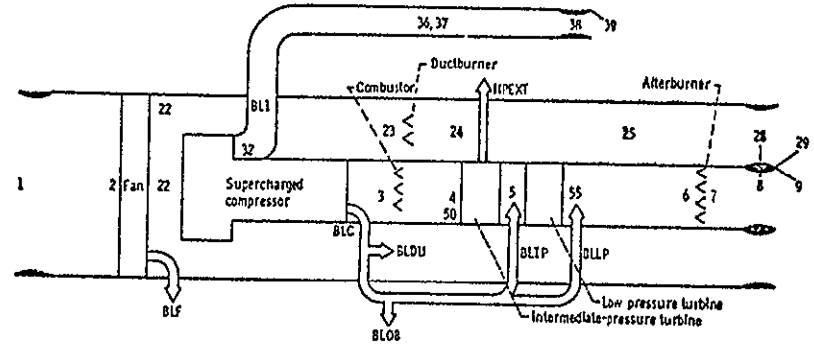


Figure 5.3-3. Two-Spool, Three-Stream, Supercharged Compressor Engine (Type c)

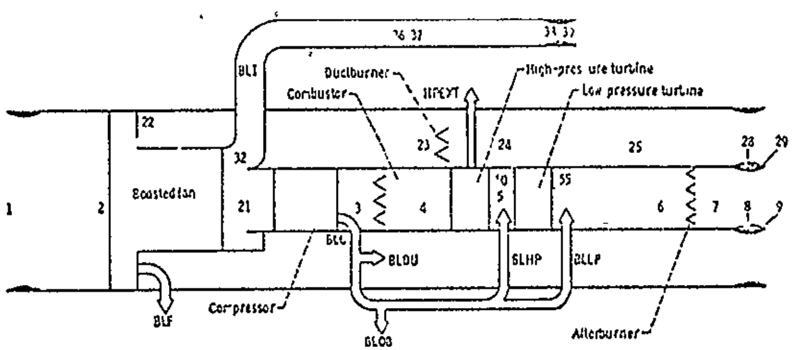


Figure 5.3-2. Two-Spool, Three-Stream Boosted Fan Engine (Type b)

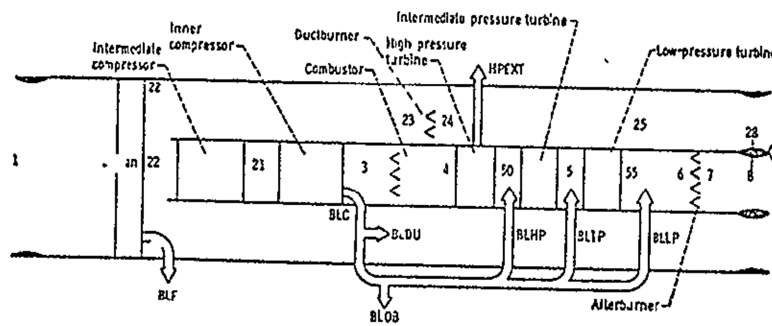


Figure 5.3-4. Three-Spool, Two-Stream Engine (Type d)

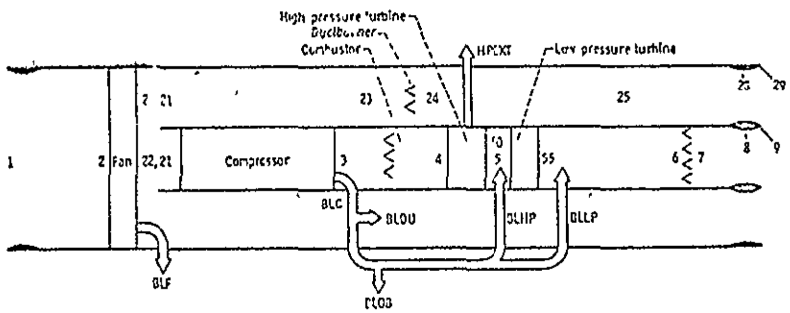


Figure 5.3-5. Two-Spool, Two-Stream Turbofan Engine (Type e)

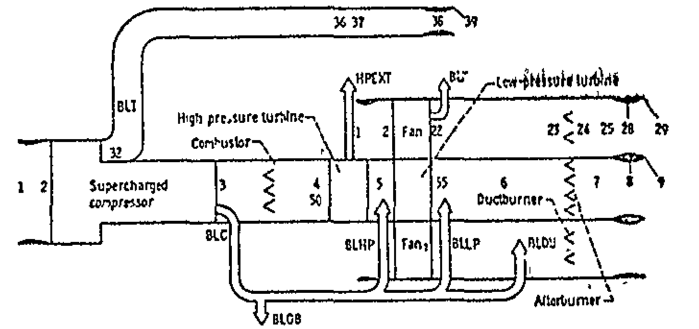


Figure 5.3-7. Two-Spool, Three-Stream, Aft Fan Engine (Type g)

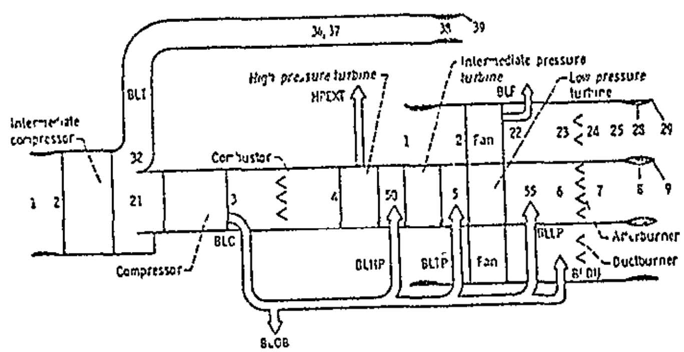


Figure 5.3-6. Three-Spool, Three-Stream, Aft Fan Engine (Type f)

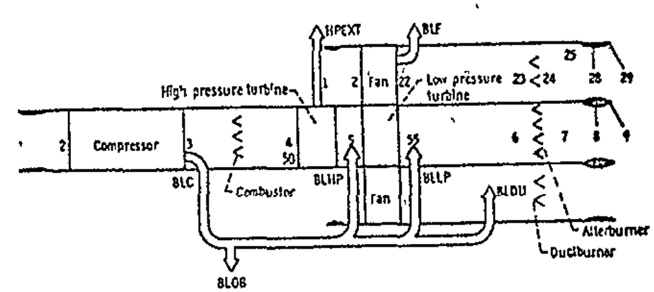


Figure 5.3-8. Two-Spool, Two-Stream, Aft Fan Engine (Type h)

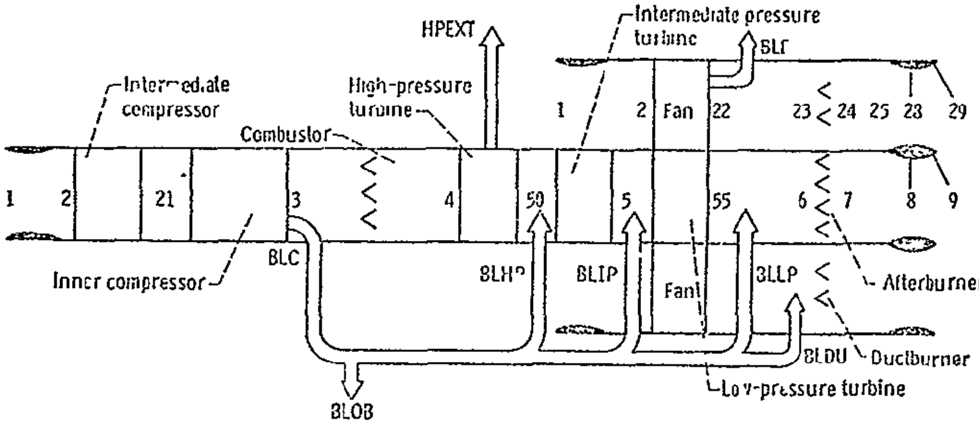


Figure 5.3-9. Three-Spool, Two-Stream, Aft Fan Engine (Type i)

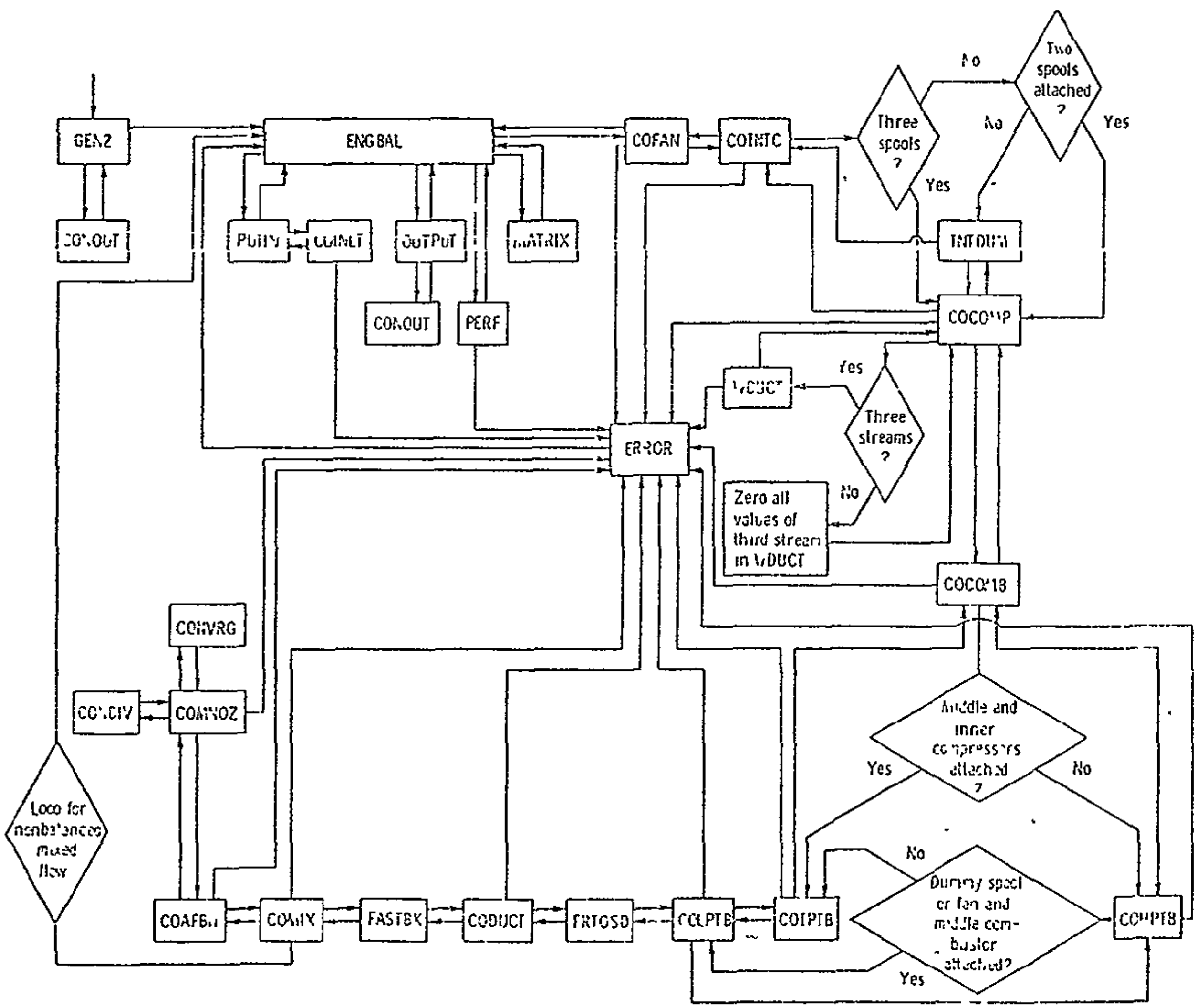


Figure 5.3-10. Flow Chart for GENENG II

SECTION 6

MASS AND VOLUMETRIC PROPERTIES

The ODIN/RLV program library contains two independent programs for estimation of reusable launch vehicles' mass/volume properties. These programs were developed under previous National Aeronautics and Space Administration and Air Force Flight Dynamics Laboratory-funded studies. Programs are provided for

1. Approximate mass and volume properties based on statistical past flight vehicle designs

2. Detailed mass and volume properties based on component representations of flight vehicles

Both programs are outlined in the following sections. For complete details reference should be made to the original source documents, References 1 through 7.

The approximate volume and mass properties routine is taken from the Air Force Flight Dynamics Laboratory's vehicle synthesis for advanced concepts program, VSAC. This program provides a self-contained vehicle synthesis capability for certain classes of flight vehicle.

Complete program details including options for

1. aerodynamics

2. propulsion

3. performance

4. volume and mass properties

are given in References 1 and 2. ODIN/RLV usage to date has been limited to the volume and mass properties routines.

It should be noted that the weight equations of the VSAC program include the References 4 to 7 SSSP program's weight equations as options.

TABLE OF CONTENTS FOR SECTION 6.1, PROGRAM VSAC

6.1 PROGRAM VSAC: APPROXIMATE AIRCRAFT MASS PROPERTIES AND VOLUME ANALYSIS

This program computes approximate military flight vehicle mass and volumetric properties based on the statistics of past designs. This technique is based on (a) correlation of past vehicle mass and volume properties against physically significant parameters and (b) regression analysis of the correlations to provide an analytic model for military flight vehicle mass and volume properties.

The program operates at the subsystem and major component level. The subsystem breakdown employed is

1. aerodynamic surfaces
2. body structure
3. induced environment protection
4. launch and recovery
5. main propulsion
6. orientation controls and separation
7. power supply, conversion and distribution
8. avionics
9. crew systems
10. design reserve
11. personnel
12. payload
13. propellants

Each subsystem is broken down into major components. For example, aerodynamic surfaces is broken down into four components:

1. wings
2. vertical fin
3. horizontal stabilizer
4. fairings, shrouds, and associated structure

Each subsystem and subsystem component weight and volume weight estimation relationship used in program VSAC is presented below.

Weight analysis is based entirely on the weight and volume subroutines in the Air Force Flight Dynamics Laboratory VSAC program, References 1 and 2. For complete details regarding the analytic basis of the weight model, reference should be made to the original VSAC documentation. An outline of the VSAC program capability follows.

It should be noted that an extended weight analysis code which incorporates this analysis is now available in Reference 3. This code, WAATS, has eliminated all VSAC calculations which are extraneous to the weight analysis function. The resulting code fits into 20000 machine locations.

6.1.1 Aerodynamic Surfaces

The total weight of the aerodynamic surface group is given by

WSURF = WWING + WVERT + WHORZ + WFAIR (6.1.1)

where

WWING = wing weight
WVERT = vertical fin weight
WHORZ = horizontal tail weight
WFAIR = aerodynamic fairing weight

Expressions for each of these component weights are presented below.

6.1.1.1(a) Wing

The wing weight equation calculates an installed structural wing weight including control surfaces and carry through. The weight is calculated as a function of load and geometry:

```
WWING = AC(1) * (WTO*XLF*STSPAN*SWING/TROOT)**AC(78)/1000

     + AC(2) * SWING + AC(3)                              (6.1.2)
```

where

WWING = total structural wing weight, lbs.
WTO = gross weight, lbs.
XLF = ultimate load factor
STSPAN= structural span (along .5 chord), ft.
SWING = gross wing area, $ft.^2$
TROOT = theoretical root thickness, ft.
AC(1) = wing weight coefficient (intercept)
AC(78)= wing weight coefficient (slope)
AC(2) = wing weight coefficient (f(gross area)), lbs/ft^2
AC(3) = fixed wing weight, lbs.

The data in Figures 6.1-1 and 6.1-2 represent wings that are basically constructed of aluminum and wings that are basically constructed of high temperature materials (steel and inconel), respectively. The latter data is also representative of *supersonic wings with t/c values in the order of 3 to 3-1/2%* . For variable sweep wing designs the various wing input terms should be based on the fully swept position. The C(1) coefficient should then be increased by 15 to 20 per cent to account for the structural penalty for sweeping the wing forward. The user has an option of adding or removing a wing weight penalty on the basic wing calculation. An example would be to add a fixed weight per square foot for thermal protection system structure or high temperature resistant coatings. The coefficient C(3) is to input a fixed weight to the wing calculation.

6.1.1.2 Vertical Fin

The vertical fin weight includes the weight of the control surface. The weight is calculated as a logarithmic function of surface area. The equation for vertical fin weight is

$$WVERT = AC(4) * SVERT ** AC(89) + AC(5) \qquad (6.1.3)$$

where
WVERT = total vertical fin weight, lbs
SVERT = vertical fin planform area, ft^2
AC(4) = vertical fin weight coefficient
AC(89)=vertical fin weight coefficient (slope)
AC(5) = fixed vertical fin weight, lbs.

The data of Figure 6.1-3 is based on Mach 2-type airplanes. They include aluminum, steel and inconel fin materials. Figure 6.1-3 is assumed to be representative of the best type construction for the Mach 0.6 to 2.0 range. The data, as shown, does not include allowances for thermal protection system weight.

6.1.1.3 Horizontal Stabilizer

The horizontal stabilizer weight includes the weight of the control surface. The weight is calculated as a function of wing loading, stabilizer planform area and dynamic pressure. The equation for horizontal stabilizer weight is

$$WHORZ = AC(6) * ((WTO/SWING) ** .6 * SHORZ ** 1.2 * QMAX **.8)$$

$$**AC(90) + AC(7) \qquad (6.1.4)$$

where
WHORZ = total horizontal stabilizer weight, lbs.
WTO = gross weight, lbs.
SWING = gross wing area, $ft.^2$
SHORZ = horizontal stabilizer planform area, $ft.^2$
QMAX = maximum dynamic pressure, $lbs/ft.^2$
AC(6) = horizontal stabilizer weight coefficient (intercept)
AC(90) = horizontal stabilizer weight coefficient (slope)
AC(7) = fixed horizontal stabilizer weight, lbs.

The data includes aluminum and inconel stabilizer materials. The data, as shown, does not include allowances for thermal protection system weight Figure 6.1.4.

6.1.1.4 Fairings, Shrouds, and Associated Structure

The type of aerodynamic structures included in this section are aerodynamic shrouds, equipment, dorsal, landing gear, and canopy fairings. The canopy fairing is the structure aft of the canopy that is required to fair the canopy to the body. The weight of the canopy proper is included in Section 6.1.2.2. Wing to body fairings are included in the wing weights. Horizontal or vertical surface to body fairings are included in either the horizontal or vertical surface weight.

Fairing and shroud weight may be determined from their surface area and the operating environment and is given in the program as

$$WFAIR = AC(8) * SFAIR + AC(9) \tag{6.1.5}$$

where
WFAIR = total weight of fairings or shrouds, lbs.
SFAIR = total fairing or shroud surface area, $ft.^2$
AC(8) = unit weight of fairing or shroud, $lbs./ft.^2$
AC(9) = fixed weight of fairing or shroud, lbs.

If the design loads and the fairing geometry is known, the weight in $lbs./ft.^2$ (i.e., the coefficient AC(8)) can be found by calculation. In most cases, however, empirical or statistical data has to be used. The coefficient AC(8) can be found by multiplying an empirical unit weight WF by a factor to account for dynamic pressure and temperature differences.

$$AC(8) = WF \cdot KQ \cdot KT \tag{6.1.6}$$

where
WF = fairing weight factor, Table 6.1-1
KQ = fairing dynamic pressure coefficient, Figure 6.1-5
KT = fairing temperature coefficient, Figure 6.1-6

The factor KQ is shown plotted against dynamic pressure in Figure 6.1-5. The factor KT is shown plotted versus temperature in Figure 6.1-6. The unit weight os typical fairings, WF, is shown in Table 6.1-1.

6.1.2 Aircraft Body Structure

The total weight of the aircraft body group is given by

$$WBODY = WBASIC + WSECST + WTHRST \tag{6.1.7}$$

where
WBASIC = basic body weight
WSECST = secondary structure weight
WTHRST = thrust structure weight

Expressions for each component weight are given below. The weight of *booster* body structures is presented in Section 6.1.2.

6.1.2.1 Basic Aircraft Body

The vehicle body weight equation is based upon correlating the actual weight of existing hardware with significant load, geometry, and environmental parameters. For vehicles of an advanced nature, modifying factors based upon design studies of cruise vehicles are applied to the basic data to account for the expected advances in technology and more severe environment. Equations derived from existing data includes non-optimum factors which are difficult to justify by analytical procedures. These non-optimum factors are important weight items, as shown by the weight growth of many vehicles between the initial concept and the finished hardware.

The equation used for basic body weight is

$$WBASIC = AC(14) * SBODY + AC(15)((ELBODY*XLF/HBODY) **.15 * QMAX ** .16 * SBODY ** AC(81) + AC(16) \qquad (6.1.8)$$

where
WBASIC = total weight of basic body, lbs.
SBODY = total body wetted area, $ft.^2$
XLF = ultimate load factor
ELBODY = body length, ft.
QMAX = maximum dynamic pressure, $lbs./ft.^2$
HBODY = body height, ft.
AC(14) = basic body unit weight, $lbs./ft.^2$
AC(15) = basic body weight coefficient (intercept)
AC(81) = basic body weight coefficient (slope)
AC(16) = fixed basic body weight, lbs.

The primary function of the first part of the basic body equation, AC(14) * SBODY allows a weight penalty based upon a constant unit weight of structural area without involving the parameters used in the second part of the overall equation. The second part of the equation obtains the basic body weight using design and geometry parameters. The basic body weight data is shown in Figure 6.1-7. Since the data is for aluminum structure, operating at temperatures of 250°F, a modifying factor must be used with AC(15) for other materials and temperatures. The modifying factor (MF) is obtained from Figure 6.1-8. The AC(15) obtained from Figure 6.1-7 is multiplied by the modifying factor (MF) to obtain the input for aluminum, titanium or Rene' 41 at elevated temperatures.

$$AC(15)_{actual} = AC(15)_{fig.6.1-7} \times MF \qquad (6.1.9)$$

6.1.2.2 Aircraft Body Secondary Structure

Secondary structure includes windshields, canopy, landing gear doors, flight opening doors and speed brakes. If a weight estimate based upon analysis is available, it should be used in lieu of the following data.

The equation for calculating secondary structure is

$$WSECST = AC(17) * SBODY + AC(18) \qquad (6.1.10)$$

where
WSECST = weight of body secondary structure, lbs.
SBODY = total body wetted area, $ft.^2$
AC(17) = secondary structure unit weight, $lbs./ft.^2$
AC(18) = fixed secondary structure weight, lbs.

The body secondary weight coefficient AC(17) varies from 0.58 to 1.38. If specific design detail is not available, an average value of 0.98 may be used for the AC(17) coefficient. However, if any design detail is available, the coefficient should be tailored using the data shown in Table 6.1-2 as a guideline.

6.1.2.3 Aircraft Thrust Structure

The thrust structure weights are a function of the total vacuum thrust of the engines. The equation used for thrust structure weight is

$$WTHRST = AC(19) * TTOT + AC(20) \qquad (6.1.11)$$

where
WTHRST = weight of thrust structure, lbs.
TTOT = total stage vacuum thrust, lbs.
AC(19) = thrust structure weight coefficient
AC(20) = fixed thrust structure weight, lbs.

The aircraft thrust structures are required to mount airbreathing engines and rocket engines. The airbreathing thrust structure weight coefficients AC(19) and AC(20) are obtained from Figure 6.1-9. The input for rocket engine thrust structure weight is obtained from Figure 6.1-10. The rocket engine thrust structure assumed for this data is a cone or barrel structure attached to a bulkhead.

6.1.3 Booster Body Structure

The total weight of the booster body group is given by

$$BWBODY = BWINFT + BWINOT + BWBASC + BNSSTR + BWTRST \qquad (6.1.12)$$

where
BWINFT = integral fuel tank weight
BWINOT = integral oxidizer tank weight
BWBASC = basic body structure weight
BWSSTR = secondary structure weight
BWTRST = thrust structure weight

Expressions for each component weight are given below. The weight of *aircraft* body structures has been presented in Section 6.1.2.

6.1.3.1 Booster Integral Fuel Tanks

The integral fuel tanks are sized as a function of total tank volume, including ullage and residual volume. The input coefficients are based on historical data from the Saturn family of LO_2/LH_2 vehicles. The equation for integral fuel tank weight is

$$BWINFT = BC(10) * BVFUTK + BC(11) \qquad (6.1.13)$$

where
BWINFT = weight of integral fuel tank, lbs.
BVFUTK = total volume of fuel tank, $ft.^3$
BC(10) = integral fuel tank weight coefficient, $lbs./ft.^3$
BC(11) = fixed integral fuel tank weight, lbs.

The integral fuel tank weight coefficients BC(10) and BC(11) are obtained from Figure 6.1-11. When a non-Saturn type tank configuration is utilized, the coefficient BC(10) should be multiplied by a configuration factor.

6.1.3.2 Booster Integral Oxidizer Tanks

The integral oxidizer tanks are sized as a function of total tank volume, including ullage and residual volume. The input coefficients are based on historical data from the Saturn family of LO_2/LH_2 vehicles. The equation for integral oxidizer tank weight is

$$\text{BWINOT} = \text{BC}(12) * \text{BVOXTK} + \text{BC}(13) \qquad (6.1.14)$$

where
BWINOT = weight of integral oxidizer tank, lbs.
BVOXTK = total volume of oxidizer tank, $ft.^3$
BC(12) = integral oxidizer tank weight coefficient, $lbs./ft.^3$
BC(13) = fixed integral oxidizer tank weight, lbs.

The integral oxidizer tank weight coefficients BC(12) and BC(13) are obtained from Figure 6.1-12. When a non-Saturn type tank configuration is utilized, the coefficient BC(12) should be multiplied by a configuration factor.

6.1.3.3 Booster Basic Body Structure

The basic body weight includes the structure forward, aft and in between the integral tanks but does not include the secondary structure or thrust structure. The equation for basic body structure weight is

$$\text{BWBASC} = \text{BC}(14) * \text{BSBODY} + \text{BC}(15) * \text{BVBODY} + \text{BC}(16) \qquad (6.1.15)$$

where
BWBASC = total weight of basic body, lbs.
BSBODY = total body wetted area, $ft.^2$
BVBODY = total body volume, $ft.^3$
BC(14) = basic body weight coefficient (F(area)), $lbs./ft.^2$
BC(15) = basic body weight coefficient (f(volume)), $lbs./ft.^3$
BC(16) = fixed basic body weight, lbs.

The equation is programmed to accept a coefficient input as a function of wetted area or volume. The coefficient BC(14) is a function of area and is derived as follows.

$$\text{BC}(14) = 4.0 * \frac{\text{Basic Body Wetted Area}}{\text{Total Body Wetted Area}}$$

The coefficient BC(15) is a function of volume. Input data for this coefficient has not been derived in the original study of Reference 1.

6.1.3.4 Booster Secondary Structure

The secondary structure includes access doors, non-structural fairings, etc. The secondary structure is minimal for the type of booster designs involved in the VSAC study of References 1 and 2. The equation for booster secondary structure weight is

$$\text{BWSSTR} = \text{BC}(17) * \text{BSBODY} + \text{BC}(18) \qquad (6.1.16)$$

where
BWSSTR = total weight of body secondary structure, lbs.

BSBODY = total body wetted area, $ft.^2$
BC(17) = secondary structure weight coefficient, $lbs./ft.^2$
BC(18) = fixed secondary structure weight, lbs.

The weight coefficient BC(17) is used to scale the secondary structure weight as a function of body wetted area. When possible, the coefficient should be derived from design data. However, during the early phase of a study, this is not always practical. A first cut value of 0.05 to 0.1 may be used for BC(17) until design data is available.

6.1.3.5 Booster Thrust Structure

The weight of the rocket engine thrust structure is a function of total vacuum thrust and type of attachment utilized. However, for this study the type of attachment has been restricted to a cone or barrel structure attached to the aft bulkhead. With this design criteria, the effect of attachment geometry is built into the BC(19) coefficient. The equation for booster thrust structure is

$$BWTRST = BC(19) * BTTOT + BC(20) \quad (6.1.17)$$

where
BWTRST = total weight of thrust structure, lbs.
BTTOT = total stage vacuum thrust, lbs.
BC(19) = thrust structure weight coefficient
BC(20) = fixed thrust structure weight, lbs.

The weight coefficient BC(19) is used to scale the thrust structure as a function of total stage vacuum thrust. When specific design data is not available, a typical preliminary design value of BC(19) = 0.0025 will provide a realistic thrust structure weight for a cone or barrel design concept. This coefficient input value does not include the aft skirt weight.

6.1.4 Aircraft Induced Environment Protection

The total weight of the aircraft induced environment protection group is given by

$$WTPS = WINSUL + WCOVER \quad (6.1.18)$$

where
WINSUL = insulation weight
WCOVER = cover plate weight

The inputs for a specific design concept are normally obtained by a thermal analysis. This method should be used when specific design conditions are known, as it yields the most accurate results accounting for all the features of a particular design. When detailed knowledge of a design is not available, generalized data is given based upon the results of prior design studies. The data presented is simplified for use in generalized aircraft weight/sizing. The results do not replace a detailed thermal analysis.

A radiative protection system to hold structural temperatures within acceptable limits is the type of vehicle thermal protection system considered for this study. This system utilizes radiative cover panels with or without insulation.

6.1.4.1 Aircraft Insulation

When insulation is used, it assumes that the structural temperature is held to approximately 200°F. The insulation must then be protected from the flight conditions by radiative cover panels. The equation for the insulation weight is

$$WINSUL = AC(21) * STPS + AC(76) \qquad (6.1.19)$$

where

WINSUL = total weight of TPS insulation, lbs.
STPS = total TPS surface area, $ft.^2$
AC(21) = insulation unit weight, $lbs./ft.^2$
AC(76) = fixed insulation weight, lbs.

The coefficient AC(21) is an insulation unit weight that may be obtained as a function of surface temperature from Figure 6.1-13. The user must estimate the surface temperature that will be encountered in order to input the coefficient AC(21). The data shown in Figure 6.1-13 is based on microquartz insulation for a *1.0 hour time duration*. The three curves represent allowable heating rates of 100, 400, and 700 $Btu/ft.^2$ with the structural temperature being held to approximately 200°F. The area of the aircraft which is to be covered by insulation is specified in the input data.

The coefficient AC(76) is a fixed input weight to the insulation calculation. A typical example of the use of this coefficient would be to add a fixed insulation weight for localized hot spots.

6.1.4.2 Aircraft Cover Panels

When the design concept utilizes insulation panels to hold the structural temperature within acceptable limits, the insulation must be protected from flight conditions. This protection is provided by cover panels. The equation for the cover panel weight is

$$WCOVER = AC(22) * STPS + AC(77) \qquad (6.1.20)$$

where

WCOVER = total weight of TPS cover panels, lbs.
STPS = total TPS surface area, $ft.^2$
AC(22) = cover panel unit weight, $lbs./ft.^2$
(AC77) = fixed cover panel weight, lbs.

Cover panels used in recent studies have varied greatly in design features and materials. The generalized equation used in this program must be input from point design data if a specific design is to be properly represented. A range of input values are included to provide the user with a weight that will be representative of the cover panel designs used in recent studies.

The coefficient will vary from AC(22) = 0.8 to 1.5 if insulation is used in conjunction with the cover panels. If insulation panels are not utilized, the input will vary from AC(22) = 1.25 to 2.0. The lower values are representative of efficient attachment capability and the higher value requiring deep frame or standoff's for attachment. The values shown are average unit weights to be used with the total body wetted area.

6.1.5 Booster Induced Environment Protection

The total weight of the booster induced environment protection group is given by

$$BWTPS = BWINSL + BWCOVR \qquad (6.1.21)$$

where
BWINSL = insulation weight
BWCOVER= cover plate weight

A radiative protection system is used to hold structural temperatures within acceptable limits in the VSAC study. The comments in Section 6.1.4 apply equally to boosters.

6.1.5.1 Booster Insulation Weight

The equation for the insulation weight is

$$BWINSL = BC(21) * BSBODY + BC(76) \qquad (6.1.22)$$

where
BWINSL = total weight of TPS insulation, lbs.
BSBODY = total body wetted area, $ft.^2$
BC(21) = insulation unit weight, $lbs./ft.^2$
BC(76) = fixed insulation weight, lbs.

The coefficient BC(21) is an insulation unit weight that may be obtained as a function of surface temperature from Figure 6.1-14. The user must estimate the surface temperature that will be encountered on the initial case in order to input the coefficient BC(21). The data shown in Figure 6.1-14 is based on microquartz insulation for a *one-half hour time duration.* The three curves represent allowable heating rates of 100, 400, and 700 $Btu/ft.^2$ with the structural temperature being held to approximately 200°F.

The equation for booster stage insulation computes the weight as a function of total body wetted area. If only a percentage of the body is actually covered by insulation, the input coefficient BC(21) must be modified by that percentage value to account for the weight. The coefficient BC(76) is a fixed input weight to the insulation calculation. A typical example of the use of this coefficient would be to add a fixed insulation weight for localized hot spots.

When the design concept utilizes insulation panels to hold the structural temperature within acceptable limits, the insulation must be protected from flight conditions. This protection is provided by cover panels. The equation for cover panel weight is

BWCOVR = BC(22) * BSBODY + BC(77) (6.1.23)

where
BWCOVR = total weight of TPS cover panels, lbs.
BSBODY = total body wetted area, $ft.^2$
BC(22) = cover panel unit weight, $lbs./ft.^2$
BC(77) = fixed cover panel weight, lbs.

The cover panels that have been used in recent studies have varied greatly in design features and materials. The discussion regarding AC(22) in Section 6.1.4.2 also applies to values for BC(22) above.

6.1.6 Aircraft Launch and Recovery

The total weight of the aircraft launch and recovery gear is given by

WGEAR = WLANCH + WLG (6.1.24)

where
WLANCH = launch system weight (if any)
WLG = landing gear weight

Expressions for these component weights are given below.

6.1.6.1 Launch Gear

The launch gear equation is used for the support structure and devices associated with aircraft that are used to attach to a hover ship. This includes struts, pads, sequencing devices, controls, etc. The equation for launch gear is

WLANCH = AC(23) * WTO + AC(24) (6.1.25)

where
WLANCH = total weight of launch gear, lbs.
WTO = gross weight, lbs.
AC(23) = launch gear weight coefficient
AC(24) = fixed launch gear weight, lbs.

The weight coefficient AC(23) is a proportion of the computed gross weight. A typical value for preliminary design purposes, would be AC(23)= 0.0025.

6.1.6.2 Landing Gear

The landing gear equation has been developed from data correlation of existing aircraft. This data included the nose gear, main gear and controls. The equation for calculating landing gear (including controls) is

WLG = AC(25) * WTO ** AC(101) + AC(26) * WLAND + AC(27)   (6.1.26)

where

WLG = total weight of landing gear and controls, lbs.
WTO = gross weight, lbs.
WLAND = maximum landing weight, lbs.
AC(25) = landing gear weight coefficient (intercept $f_{(WTO)}$)
AC(101)= landing gear weight coefficient (slope $f_{(WTO)}$)
AC(26) = landing gear weight coefficient ($f_{(WLAND)}$)
AC(27) = fixed landing gear weight, lbs.

The landing gear weight coefficients are shown in Figure 6.1-15. These coefficients should be used when the landing gear is to be scaled as a function of gross weight. When the coefficients AC(25) and AC(101) are used, the coefficient AC(26) should be zero.

The weight coefficient AC(26) is used for vehicles whose gear is used only for landing. Gear weight will then vary with the landing weight instead of gross weight. For first estimates the coefficient AC(26) should range between 0.03 for 11 feet per second sink rate and 0.05 for 25 feet per second. When the coefficient AC(26) is used, the coefficient AC(25) should be set to zero.

6.1.7 Aircraft Main Propulsion

The total weight of the aircraft main propulsion group is given by

WPROPU = WABENG + WRENGS + WFUNCT + WOXCNT + WINSFT + WINSOT + WFUSYS
+ WOXSYS + WPRSYS + WINLET   (6.1.27)

where

WABENG = airbreathing engine weight including engine mounts
WRENGS = rocket engine weight, including engine mounts
WFUNCT = fuel tank weight
WOXCNT = oxidizer tank weight, rocket engines only
WINSFT = fuel tank insulation weight
WINSOT = oxidizer tank weight, rocket engines only
WFUSYS = weight of storable propellant fuel system, less tanks
WOXSYS = crogenic propellant oxidizer system weight
WPRSYS = propellant pressurization system weight
WINLET = inlet system weight

Expressions for each component weight are presented below.

6.1.7.1 Aircraft Main Propulsion Engines, Turboramjet, Ramjet, and Rocket

The main engines are used to propel the vehicle. This includes either airbreathing or rocket propulsion systems. The airbreathing engines considered in this study are the turboramjet and ramjet.

6.1.7.1(a) Turboramjet

The turboramjet data is for the GE 12/JZ8 engine. The equation for turboramjet follows.

$$WABENG = (AC(32) * e ** (AC(33) * WA) * ((PT2-PHIGH)/(PLOW-PHIGH) + AC(34) * e ** (AC(35) * WA) * ((PT2-PLOW)/(PHIGH-PLOW)) * ENGINS + AC(91) * ENGINS + WENGMT \quad (6.1.28)$$

where

WABENG	=	total weight of airbreathing engines, lbs.
WA	=	calculated turboramjet engine air flow rate, lbs./sec.
PT2	=	calculated turboramjet engine inlet pressure, psi.
PHIGH	=	turboramjet engine inlet pressure (upper design curve), psi
PLOW	=	turboramjet engine inlet pressure (lower design curve), psi
ENGINS	=	total number of engines per stage
WENGMT	=	weight of engine mounts, lbs.
AC(32)	=	turboramjet engine weight coefficient (lower design point)
AC(33)	=	turboramjet engine weight coefficient (lower design point)
AC(34)	=	turboramjet engine weight coefficient (upper design point)
AC(35)	=	turboramjet engine weight coefficient (upper design point)
AC(91)	=	fixed turboramjet engine weight, lbs.

The weight coefficients, AC(32), AC(33), AC(34) and AC(35) are used to scale the turboramjet engine weight as a function of engine air flow rate and pressure. The input values for these coefficients may be obtained from Figure 6.1-16. The data presented is for two design conditions of the GE 14/JZ8 engine. The data in the lower curve represents an engine for Mach 4.5 with a pressure of 46 psia at a cruise altitude of 90,000 feet. The data in the upper curve represents an engine for Mach 4.5 with a pressure of 176 psia at a cruise altitude of 61,600 feet. The ratio of calculated pressure (PT2) to the pressure for the upper curve (PHIGH = 176 psia) and the pressure for the lower curve (PLOW = 46 psia) allows a scaling capability around the two design conditions.

6.1.7.1(b) Ramjet

The ramjet engine is sized as a function of thrust. The equation for ramjet engine weight is

$$WABENG = AC(82) * TTOT + AC(83) + WENGMT \quad (6.1.29)$$

where

WABENG	=	total weight of airbreathing engines, lbs.
TTOT	=	total stage vacuum thrust, lbs.
AC(82)	=	ramjet engine weight coefficient
AC(83)	=	fixed ramjet engine weight, lbs.
WENGMT	=	weight of engine mounts, lbs.; see Section 6.1.7.2.

An input value of AC(82) = 0.01 is representative of a low volume ramjet engine with a thrust to calculated weight ratio equal to 100:1. Figure 6.1-17 shows ramjet engine weight versus thrust for an AC(82) value of 0.01.

6.1.7.1(c) Rocket

The rocket engine data is based on the LR-129 LO_2/LH_2 engine. The weight is scaled as a function of total stage vacuum thrust and area ratio. The equation for rocket engine weight is

$$WRENGS = AC(28) * TTOT + AC(29) * TTOT * ARATIO ** AC(30) + AC(31) * ENGINS + WENGMT \quad (6.1.30)$$

where

WRENGS = total weight of rocket engine installation, lbs.
TTOT = total stage vacuum thrust, lbs.
ARATIO = rocket engine area ratio
ENGINS = total number of engines per stage
WENGMT = weight of engine mounts, lbs.; see Section 6.1.7.2
AC(28) = rocket engine weight coefficient ($f_{(Thrust)}$)
AC(29) = rocket engine weight coefficient ($f_{(Thrust\ and\ area\ ratio)}$)
AC(30) = rocket engine area ratio exponent
AC(31) = fixed rocket engine weight, lbs.

The weight coefficients AC(28), AC(29), and AC(30) are obtained from Figure 6.1-18. The engine data presented does not include allowances for PVC ducts or gimbal system. The gimbal system weight equation is presented in Section 6.1.9.1. The assumption has been made that PVC ducts are not required on the type vehicles used for this study.

6.1.7.2 Aircraft Engine Mounts

The weight equation for engine mounts is

$$WENGMT = AC(102) * TTOT + AC(103) \quad (6.1.31)$$

where

WENGMT = weight of engine mounts, lbs.
TTOT = total stage vacuum thrust, lbs.
AC(102) = engine mount weight coefficient
AC(103) = fixed engine mount weight, lbs.

The expression AC(102) * TTOT is the penalty for engine mounts attached to the engine. The engine mounting penalty associated with the body is included in basic body structure. A typical value used in design studies is AC(102) = 0.004 for airbreathing engine installations and AC(102)= 0.0001 for rocket engines.

6.1.7.3 Aircraft Fuel and Oxidizer Tanks

The type of fuel and oxidizer tank construction include non self-sealing (bladder), self-sealing, and integral. The configuration concepts that utilize airbreathing engines with JP-4 and JP-5 type fuel may use any one of the three type fuel tank constructions discussed. However, when airbreathing engines are used with liquid hydrogen fuel the tanks are assumed to be an integral design based on the X-15 concept. The configuration concepts that utilize a rocket engine installation are assumed to have an integral tank design for both fuel and oxidizer that is based on the X-15 design concept.

6.1.7.3(a) JP-4 and JP-5 Type Fuel

The non self-sealing and self-sealing fuel tank weights for JP-4 and JP-5 type fuel are derived by the equation

WFUNCT = AC(36) * (GAL/Tanks) ** .6 * TANKS + AC(37) (6.1.32)

where
WFUNCT = total weight of fuel tank, lbs.
GAS = total gallons of fuel
TANKS = number of fuselage fuel tanks
AC(36) = fuel tank weight coefficient (=0, for integral tanks)
AC(37) = fixed fuel tank weight, lbs. (=0, for integral tanks)

The weight coefficient AC(36) is obtained from Figure 6.1-19. The weight for these tanks include supports and backing boards. Existing airplanes that utilize integral fuel tank are the F-102, F-106, and F-111. The F-4 and A-7 also utilize this concept in the wings but not in the fuselage.

6.1.7.3(b) Liquid Hydrogen Fuel and Rockets

The aircraft stages that use either airbreathing engines with liquid hydrogen fuel or rocket engines are assumed to have propellant tanks that are integral and based on the X-15 design concept. The equation for fuel tank weight is

WFUNCT = AC(36) * VFUTK + AC(37) (6.1.33)

where
WFUNCT = total weight of fuel tank, lbs.
VFUTK = total volume of fuel tank, $ft.^3$
AC(36) = fuel tank weight coefficient, $lbs./ft.^3$
AC(37) = fixed fuel tank weight, lbs.

The weight coefficient AC(36) is obtained from Figure 6.1-20. The equation for oxidizer tank weight is

WOXCNT = AC(38) * VOXTK + AC(39) (6.1.34)

where
WOXCNT = total weight of oxidizer tank, lbs.
VOXTK = total volume of oxidizer tank, $ft.^3$
AC(38) = oxidizer tank weight coefficient, $lbs./ft.^3$ (=0, for airbreather)
AC(39) = fixed oxidizer tank weight, lbs. (=0, for airbreather)

The weight coefficient AC(38) is obtained from Figure 6.1-20.

6.1.7.4 Aircraft Fuel Tank Insulation

This section presents the data to obtain a weight penalty associated with protection required to prevent excessive boil-off from cryogenic propellant tanks. The insulation penalty is in terms of lbs./ft.^{2} of tank area.

The equation for fuel tank insulation weight is

$$WINSFT = AC(40) * SFUTK + AC(41) \qquad (6.1.35)$$

where
WINSFT = total weight of fuel tank insulation, lbs.
SFUTK = total fuel tank wetted area, ft.^{2}
AC(40) = fuel tank insulation unit weight, lbs./ft.^{2}
AC(41) = fixed fuel tank insulation weight, lbs.

The weight coefficient AC(40) is obtained from Figure 6.1-21. The fuel tank insulation unit weight is a function of radiating temperature. A typical radiating temperature of 500°F may be assumed for preliminary runs if other data is not available for making a specific selection.

The AC(40) value obtained from Figure 6.1-21 is for a total flight duration time of 5000 seconds. When other flight times are anticipated, the AC(40) value should be modified by multiplying it by the time correction factor (Tcorr.) obtained from Figure 6.1-22.

6.1.7.5 Oxidizer Tank Insulation

It is assumed that the cryogenic oxidizer may be based upon general data of Section 6.1.7.5 No requirement for the insulation of the main oxidizer tanks has been necessary in past design studies because storage times have been relatively low. However, an equation and input data is provided for cases where oxidizer tank insulation is required. The equation for oxidizer tank insulation weight is

$$WINSOT = AC(42) * SOXTK + AC(43) \qquad (6.1.36)$$

where
WINSOT = total weight of oxidizer tank insulation, lbs.
SOXTK = total oxidizer tank wetted area, ft.^{2}
AC(42) = oxidizer tank insulation unit weight, lbs/ft^2
AC(43) = fixed oxidizer tank insulation weight, lbs.

The weight coefficient AC(42) is obtained from Figures 6.1-21 and 6.1-22. The selection criteria used to obtain AC(42) is the same as that used for AC(40).

6.1.7.6 Aircraft Storable Propellant Fuel System

The weight of the storable propellant fuel system is given by the following equation:

$$WFUSYS = WBPUMP + WDIST1 + WDIST2 + WFCONT + WREFUL + WDRANS + WSEAL \quad (6.1.37)$$

where

WBPUMP	= boost and transfer pump weight
WDIST1	= weight of fuel lines, supports, fittings, etc from reservoir tank to engines
WDIST2	= weight of fuel lines, supports, fittings, etc. between tanks
WFCONT	= fuel system control weight
WREFUL	= tank refueling system weight
WDRANS	= dump and drain system weight
WSEAL	= sealing weight

Expressions for each component weight are provided below.

6.1.7.6(a) Boost and Transfer Pumps

The weight of the boost and transfer pumps is a function of the engine thrust and the number of engines. The equation for boost and transfer pumps is

$$WBPUMP = \frac{TTOT}{1000} * (1.75 + 0.266 * ENGINS) \quad (6.1.38)$$

where

WBPUMP	= total weight of boost and transfer pumps, lbs.
TTOT	= total stage vacuum thrust, lbs.
ENGINS	= total number of engines per stage

6.1.7.6(b) Fuel Distribution, Reservoir to Engine

The fuel distribution system, Part I, is the total of all fuel lines, supports, fittings, etc. to provide fuel flow from a reservoir tank to the engines. The equation for the fuel distribution Part I weight is

$$WDIST1 = ENGINS * AC(104) * (TTOT/ENGINS) ** .5 \quad (6.1.39)$$

where

WDITS1	= total weight of fuel distribution system Part I, lbs.
ENGINS	= total number of engines per stage
TTOT	= total stage vacuum thrust, lbs.
AC(104)	= weight coefficient for fuel distribution system Part I

The weight coefficient AC(104) is used to differentiate between a non-afterburning and afterburning engine. The value of AC(104) is obtained from Figure 6.1-23.

6.1.7.6(c) Fuel Distribution, Inter-Tank

The fuel distribution system, Part II, is the total of all fuel lines, fittings, supports, etc. to provide flow between various tanks within the system. The equation for the fuel distribution system Part II weight is

$$WDIST2 = 0.255 * GAL ** .7 * TANKS ** .25 \qquad (6.1.40)$$

where

WDIST2	= total weight of fuel distribution system Part II, lbs.
GAL	= total gallons of fuel
TANKS	= number of fuselage fuel tanks

6.1.7.6 (d) Fuel System Controls

The fuel system controls is the total of all valves and valve operating equipment such as wiring, relays, cables, etc. The equation for the fuel system controls weight is

$$WFCONT = 0.169 * TANKS * GAL ** .5 \qquad (6.1.41)$$

where

WFCONT	= total weight of fuel system controls, lbs.
TANKS	= number of fuselage fuel tanks
GAL	= total gallons of fuel

6.1.7.6(e) Refueling System

The fuel tank refueling system includes the ducts and valves necessary to fill the fuel tanks. The equation for fuel tank refueling system weight is

$$WREFUL = TANKS * (3.0 + 0.45 * GAL ** .333) \qquad (6.1.42)$$

where

WREFUL	= total weight of fuel tank refueling system, lbs.
TANKS	= number of fuselage fuel tanks
GAL	= total gallons of fuel

6.1.7.6(f) Dump and Drain System

The fuel tank dump and drain system is the total valves and plumbing necessary to dump and drain the fuel system. The equation for fuel tank dump and drain system weight is

$$WDRANS = 0.159 * GAL **.65 \qquad (6.1.43)$$

where

WDRANS	= total weight of fuel tank dump and drain system, lbs.
GAL	= total gallons of fuel

6.1.7.6(g) Sealing

The fuel tank bay sealing is the total weight of sealing compound and structure required to provide a fuel tight compartment. This sealing is used with a bladder tank to prevent fuel leakage and it is used to seal off a structural compartment to provide an integral tank concept. The equation for fuel tank bay sealing weight is

$$WSEAL = 0.045 * TANKS\ 8\ (GAL/TANKS)\ **.75 \qquad (6.1.44)$$

where

WSEAL = total fuel tank bay sealing weight, lbs.
TANKS = number of fuselage fuel tanks
GAL = total gallons of fuel

6.1.7.7 Aircraft Cryogenic Propellant Fuel System

The equation for cryogenic propellant fuel system weight is used for airbreathing engines that utilize liquid hydrogen fuel and with rocket engine installations. This system weight includes the pumps, lines, valves, supports, etc. associated with the cryogenic fuel system. It is divided into the components that are thrust dependent and the components that are primarily length dependent. The equation for the cryogenic fuel system weight is

$$WFUSYS = AC(44) * TTOT + AC(45) * ELBODY + AC(46) \qquad (6.1.45)$$

where

WFUSYS = total weight of fuel system, lbs.
TTOT = total stage vacuum thrust, lbs.
ELBODY = body length, ft.
AC(44) = cryogenic fuel system weight coefficient ($f_{(Thrust)}$)
AC(45) = cryogenic fuel system weight coefficient ($f_{(Length)}$), lbs./ft.
AC(46) = fixed cryogenic fuel system weight, lbs.

The thrust dependent weight coefficient AC(44) is obtained from the upper curve in Figure 6.1-24 and the length dependent weight coefficient AC(45) is obtained from the lower curve.

6.1.7.8 Aircraft Cryogenic Propellant Oxidizer System

The equation for cryogenic propellant oxidizer system weight is used with rocket engine installations. This system weight includes the pumps, lines, valves, supports, etc. associated with the cryogenic oxidizer system. It is divided into the components that are thrust dependent and the components that are primarily length dependent. The equation for the cryogenic oxidizer system weight is

$$WOXSYS = AC(47) * TTOT + AC(48) * ELBODY + AC(49) \qquad (6.1.46)$$

where

WOXSYS = total weight of oxidizer system, lbs.
TTOT = total stage vacuum thrust, lbs.
ELBODY = body length, ft.
AC(47) = cryogenic oxidizer system weight coefficient ($f_{(Thrust)}$)

AC(48) = cryogenic oxidizer system weight coefficient ($f_{(length)}$), lbs./ft.

AC(49) = fixed cryogenic oxidizer system weight, lbs.

The thrust dependent weight coefficient AC(47) is obtained from the upper curve in Figure 6.1-25 and the length dependent weight coefficient AC(48) is obtained from the lower curve. When an airbreathing engine installation is used with liquid hydrogen fuel the coefficients AC(47), AC(48), and AC(49) must be set to zero.

6.1.7.9 Aircraft Storable Propellant Pressurization System

The pressurization system for storable propellants includes the bottles, valves, plumbing and supports. This system is used on the aircraft stage with airbreathing engines. The equation for storable propellant pressurization system weight is

$$WPRSYS = 0.0009 * TTOT * TANKS \qquad (6.1.47)$$

where

WPRSYS = weight of pressurization system, lbs.

TTOT = total stage vacuum thrust, lbs.

TANKS = number of fuselage fuel tanks

6.1.7.10 Aircraft Cryogenic Propellant Pressurization System

The cryogenic propellant pressurization system is based on the X-15 concept. The system weight includes the storage bottles, stored gas, and system components. The weight equation inputs are based on the fuel and oxidizer tank volumes. The equation for cryogenic propellant pressurization system weight is

$$WPRSYS = AC(50) * VFUTK + AC(51) * VOXTK + AC(52) \qquad (6.1.48)$$

where

WPRSYS = weight of pressurization system, lbs.

VFUTK = total volume of fuel tank, ft.^{3}

VOXTK = total volume of oxidizer tank, ft.^{3}

AC(50) = fuel tank pressure system weight coefficient, lbs./ft.^{3}

AC(51) = oxidizer tank pressure system weight coefficient, lbs./ft.^{3}

AC(52) = fixed pressurization system weight, lbs.

The coefficients AC(50) and AC(51) are fuel and oxidizer dependent, respectively, for the pressurization system weights. The input value for these coefficients are obtained from Figure 6.1-26. When an airbreathing engine is used with liquid hydrogen fuel, the coefficient AC(51) must be set to zero.

6.1.7.11 Aircraft Inlet System

The weight of the inlet system is given by

$$WINLET = WIDUCT + WVRAMP + WSPIKE \tag{6.1.49}$$

where
WIDUCT = internal duct weight
WVRAMP = ramp and ramp control weight
WSPIKE = spike weight

Expressions for each component weight are given below.

6.1.7.11(a) Internal Duct

The equation for inlet internal duct weight is

```
WIDUCT = AC(53) * ((ELNLET*XINLET) ** .5 *(AICAPT/XINLET) ** .3334

              * PT2 **.6667 * GEOFCT * FCTMOK) ** AC(54) + AC(105)
```

(6.1.50)

where
WIDUCT = weight of inlet internal duct, lbs.
ELNLET = length of duct (lip to engine face), ft.
XINLET = number of inlets
AICAPT = total inlet capture area, ft.^{2}
PT2 = calculated engine inlet pressure, psia
GEOFCT = geometrical out of round factor
1.0 for round or one flat side
1.33 for two or more flat sides
FCTMOK = Mach number factor
1.0 for Mach $\leq$ 1.4
1.5 for Mach $>$ 1.4
AC(53) = inlet internal duct weight coefficient (intercept)
AC(54) = inlet internal duct weight coefficient (slope)
AC(105) = fixed internal duct weight, lbs.

The inlet internal duct weight coefficients AC(53) and AC(54) are available from Figure 6.1-27.

6.1.7.11(b) Ramp

The weight for variable ramps, actuators and controls is dependent on temperature as the design Mach number increases. The equation for the temperature correction factor follows.

$$TMPFCT = \begin{cases} 1.0, & \text{Mach number} < 3.0 \\ 0.203 * DM + 0.4, & \text{Mach number} \geq 3.0 \end{cases} \quad (6.1.51)$$

where

TMPFCT = temperature correction factor
DM = design Mach number

The design Mach number of 3.0 gives a temperature correction factor of 1.0 and should be considered as a minimum input.

The equation for variable ramps, actuators, and controls is

$$WVRAMP = AC(106) * (ELRAMP * XINLET * (AICAPT/XINLET) ** .5 * TMPFCT) ** AC(107) + AC(108) \quad (6.1.52)$$

where

WVRAMP = weight of inlet variable ramps, actuators and controls, lbs.
ELRAMP = total length of ramp, ft.
XINLET = number of inlets
AICAPT = total inlet capture area, $ft.^2$
TMPFCT = temperature correction factor
AC(106) = variable ramps, actuators and controls weight coefficient (intercept)
AC(107) = variable ramps, actuators and controls weight coefficient (slope)
AC(108) = fixed weight for variable ramps, actuators and controls, lbs.

The variable ramps, actuators, and controls weight coefficients, AC(106) and AC(107) are given in Figure 6.1-28.

6.1.7.11(c) Spike

The weight of the spike is a fixed input which depends on the type of spike used. The equation for total spike weight is

$$WSPIKE = AC(109) * XINLET \quad (6.1.53)$$

where

WSPIKE = total weight of spikes, lbs.
XINLET = number of inlets
AC(109) = spike weight coefficient, lbs.

The weight coefficient AC(109) is obtained from Table 6.1-3.

6.1.8 Booster Main Propulsion

The total weight of the booster main propulsion group is given by

$$BWPRPL = BWRENG + BWFCNT + BWOCNT + BWINSF + BWINSO + BWFUSY + BWDXSY + BWPRSY \quad (6.1.54)$$

where
BWRENG = main engine weight including mounts
BWFCNT = non-structural fuel container weight
BWOCNT = non-structural oxidizer container weight
BWINSF = fuel tank insulation weight
BWINSO = oxidizer tank insulation weight
BWFUSY = cryogenic fuel system weight
BWDXSY = cryogenic oxidizer system weight
BWPRSY = cryogenic propellant pressurization system weight

Expressions for each component weight are given below.

6.1.8.1 Booster Main Engines

The rocket engine data is based on the LR-129 LO_2/LH_2 engine. The weight is scaled as a function of total stage vacuum thrust and area ratio. The equation for rocket engine weight is

$$BWRENG = BC(28) * BTTOT + BC(29) * BTTOT * BARATO ** BC(30) + BC(31) * ENGINS + WBENMT \quad (6.1.55)$$

where
BWRENG = total weight of rocket engine installation, lbs.
BTTOT = total stage vacuum thrust, lbs.
BARATO = rocket engine area ratio
ENGINS = total number of engines per stage
WBENMT = weight of engine mounts, lbs; section 6.1.8.2
BC(28) = rocket engine weight coefficient ($f_{(Thrust)}$)
BC(29) = rocket engine weight coefficient ($f_{(Thrust\ and\ Area\ Ratio)}$)
BC(30) = rocket engine area ratio exponent
BC(31) = fixed rocket engine weight, lbs.

The weight coefficients BC(28), BC(29), and BC(30) are obtained from Figure 6.1-29. The area ratio is set by the user and its effect on engine weight is shown in Figure 6.1-29. The engine data presented does not include allowances for PVC ducts or gimbal system. The gimbal system weight equation is presented in Section 6.1.9.1. An assumption has been made that PVC ducts are not required on the type of vehicles used for this study so data has not been developed to account for them. The coefficient BC(31) is used to input the fixed engine weight that does not scale with size. This input is obtained from Figure 6.1-29.

6.1.8.2 Booster Engine Mounts

The weight equation for engine mounts is

$$WBENMT = BC(102) * BTTOT + BC(103) \qquad (6.1.56)$$

where
WBENMT = weight of engine mounts, lbs.
BTTOT = total stage vacuum thrust, lbs.
BC(102) = engine mount weight coefficient
BC(103) = fixed engine mount weight, lbs.

The expression BC(102)*BTTOT is the penalty for engine mounts attached to the engine. The engine mounting penalty associated with the body is included in the basic body structure. A typical value used in design studies is BC(102) = 0.0001.

6.1.8.3 Booster Non-Structural Propellant Containers

The program contains scaling equations for non-structural fuel and oxidizer containers that are sized as a function of total fuel tank volume and total oxidizer tank volume, respectively. The equation for non-structural fuel container weight is

$$BWFCNT = BC(36) * BVFUTK + BC(37) \qquad (6.1.57)$$

where
BWFCNT = weight of non-structural fuel tank, lbs.
BVFUTK = total volume of fuel tank, $ft.^3$
BC(36) = fuel tank weight coefficient (non-structural), $lbs./ft.^3$
BC(37) = fixed fuel tank weight (non-structural), lbs.

The equation for non-structural oxidizer container weight is

$$BWOCNT = BC(38) * BVOXTK + BC(39) \qquad (6.1.58)$$

where
BWOCNT = weight of non-structural oxidizer tank. lbs.
BVOXTK = total volume of oxidizer tank, $ft.^3$
BC(38) = oxidizer tank weight coefficient (non-structural), $lbs./ft.^3$
BC(39) = fixed oxidizer tank weight (non-structural), lbs.

6.1.8.4 Booster Fuel Tank Insulation

This section presents data to obtain a weight penalty associated with protection required to prevent excessive boiloff from cryogenic propellant tanks. The insulation penalty is in terms of $lbs./ft.^2$ of tank area which varies in the sizing routine according to tank volume which, in turn, varies with a number of other design parameters. The equation for fuel tank insulation weight is

$$BWINSF = BC(40) * BSFUTK + BC(41) \qquad (6.1.59)$$

where

BWINSF = total weight of fuel tank insulation, lbs.
BSFUTK = total fuel tank wetted area, $ft.^2$
BC(40) = fuel tank insulation unit weight, $lbs./ft.^2$
BC(41) = fixed fuel tank insulation weight, lbs.

The weight coefficient BC(40) is obtained from Figure 6.1-21 with BC(40) and BC(41) replacing AC(40) and AC(41). The fuel tank insulation unit weight is a function of radiation temperature. A typical radiating temperature of 500°F may be assumed for preliminary runs if other data is not available for making a specific selection. The BC(40) value obtained from Figure 6.1-21 is for a total flight duration time of 5,000 seconds. When other flight times are anticipated, the BC(40) value should be modified by multiplying it by the time correction factor (Tcorr.) obtained from Figure 6.1-22.

6.1.8.5 Booster Oxidizer Tank Insulation

It is assumed that the cryogenic oxidizer may be based upon the general data of Section 6.1.8.4. No requirement for the insulation of the main oxidizer tanks has been necessary in past design studies because storage times have been relatively low. However, an equation and input data is provided for cases in which oxidizer tank insulation is required.

$$BWINSO = BC(42) * BSOXTK + BC(43) \quad (6.1.60)$$

where

BWINSO = total weight of oxidizer tank insulation, lbs.
BSOXTK = total oxidizer tank wetted area, $ft.^2$
BC(42) = oxidizer tank insulation unit weight, $lbs./ft.^2$
BC(43) = fixed oxidizer tank insulation weight, lbs.

The weight coefficient BC(42) is obtained from Figure 6.1-21. The selection criteria used to obtain BC(42) is the same as that used for BC(40). The coefficient BC(42) obtained from Figure 6.1-21 is for a total flight time of 5000 seconds. When other flight times are anticipated, the BC(42) value should be modified by multiplying it by the time correction factor, Tcorr., obtained from Figure 6.1-22.

6.1.8.6 Booster Cryogenic Propellant Fuel System

The equation for cryogenic propellant fuel system weight includes the pumps, lines, valves, supports, etc. associated with the cryogenic fuel cryogenic fuel system. It is divided into the components that are thrust dependent and the components that are primarily length dependent. The equation for the cryogenic fuel system weight is

$$BWFUSY = BC(44) * BTTOT + BC(45) * BLBODY + BC(46) \quad (6.1.61)$$

where

BWFUSY = total cryogenic fuel system weight, lbs.

BTTOT = total stage vacuum thrust, lbs.
BLBODY = body length, ft.
BC(44) = cryogenic fuel system weight coefficient ($f_{(Thrust)}$)
BC(45) = cryogenic fuel system weight coefficient ($f_{(Length)}$), lbs./ft.
BC(46) = fixed cryogenic fuel system weight, lbs.

The thrust dependent weight coefficient BC(44) is obtained from the upper curve in Figure 6.1-30 and the length dependent weight coefficient BC(45) is obtained from the lower curve.

6.1.8.7 Booster Cryogenic Propellant Oxidizer System

The equation for cryogenic propellant oxidizer system weight is used with rocket engine installations. This system weight includes the pumps, lines, valves, supports, etc. associated with the cryogenic oxidizer system. It is divided into the components that are thrust dependent and the components that are primarily length dependent. The equation for the cryogenic oxidizer system weight is

$$BWOXSY = BC(47) * BTTOT + BC(48) * BLBODY + BC(49) \quad (6.1.62)$$

where
BWOXSY = total cryogenic oxidizer system weight, lbs.
BTTOT = total stage vacuum thrust, lbs.
BLBODY = body length, lbs.
BC(47) = cryogenic oxidizer system weight coefficient ($f_{(Thrust)}$)
BC(48) = cryogenic oxidizer system weight coefficient ($f_{(Length)}$), lbs./ft.
BC(49) = fixed cryogenic oxidizer system weight, lbs.

The thrust dependent weight coefficient BC(47) is obtained from the upper curve in Figure 6.1-31 and the length dependent weight coefficient BC(48) is obtained from the lower curve.

6.1.8.8 Booster Cryogenic Propellant Pressurization System

The cryogenic propellant pressurization system is representative of a stored high pressure helium system. The two major parameters used to obtain input are the main tank pressure and the helium storage temperature. The system weight includes the storage bottles, stored gas, and system components. The weight equation inputs are based on the fuel and oxidizer tank volumes. The equation for cryogenic propellant pressurization system weight is

$$BWPRSY = BC(50) * BVFUTK + BC(51) * BVOXTK + BC(52) \quad (6.1.63)$$

where
BWPRSY = weight of pressurization system, lbs.
BVFUTK - total volume of fuel tank, $ft.^3$
BVOXTK = total volume of oxidizer tank, $ft.^3$
BC(50) = fuel tank pressure system weight coefficient, $lbs./ft.^3$
BC(51) = oxidizer tank pressure system weight coefficient, $lbs./ft.^3$
BC(52) = fixed pressurization system weight, lbs.

The coefficients BC(50) and BC(51) are fuel and oxidizer dependent, respectively, for the pressurization system weights. The input value for these coefficients are obtained from Figure 6.1-32.

6.1.9 Aircraft Orientation Controls and Separation

The total weight of the aircraft orientation controls and separation group is given by

$$WORNT = WGIMBL + WACS + WACSTK + WAERO + WSEP \tag{6.1.64}$$

where

WGIMBL	= gimbal system weight
WACS	= attitude control system weight
WACSTK	= attitude control system tank weight
WAERO	= aerodynamic control system weight
WSEP	= separation system weight

Expressions for each component weight are given below.

6.1.9.1 Aircraft Gimbal System

The gimbal (thrust-vector-control) actuation system is utilized on the aircraft configuration when a rocket engine is used for main impulse. The data in Figures 6.1-33 and 6.1-34 is for an electrical system consisting of a silver-zinc primary battery, a d.c. electric motor and a gear train, two magnetic partical clutches and ball-screw actuators. Reference 1 also discussed a pneumatic actuation system. Both systems were competitive from a weight standpoint with a slight advantage for electrical systems for the longer operating times (≈1200 seconds) and for all torque levels greater than 1000 lb-in.

The system weight is expressed in parametric form as a function of delivered torque, maximum deflection rate of nozzle and operating time. The range of significant operational requirements and conditions for the data presented are given in Table 6.1-4. The system assumes pitch and yaw control for single engine and pitch, yaw and roll control for multiple engines. The equation for delivered torque is

$$TDEL = 750 * (TTOT/ENGINS/PCHAM) ** 1.25 \tag{6.1.65}$$

where

TDEL	= gimbal system delivered torque, lb-in
TTOT	= total stage vacuum thrust, lbs.
ENGINS	= total number of engines per stage
PCHAM	= rocket engine chamber pressure, psia

The delivered torque calculation assumes a maximum nozzle deflection of 10 degrees. The calculated delivered torque is then used in the gimbal system weight equation which is

WGIMBL = AC(55) * TDEL ** AC(110) + AC(56) (6.1.66)

where

WGIMBL	= weight of engine gimbal system, lbs.
TDEL	= gimbal system delivered torque, lb-in
AC(55)	= gimbal system weight coefficient (intercept)
AC(110)	= gimbal system weight coefficient (slope).
AC(56)	= fixed gimbal system weight, lbs.

The weight coefficients AC(55) and AC(110) are obtained from Figures 6.1-33 and 6.1-34. The data in Figure 6.1-33 represents a gimbal system with a maximum nozzle deflection rate of 20 degrees per second and Figure 6.1-34 is for five degrees per second. Both figures are for maximum deflections of 10 degrees and operating times of 100 and 1200 seconds. When the airplane configuration utilizes airbreathing engines for main impulse, a gimbal system is not required. Directional control will be accomplished through the use of aerodynamic surfaces.

6.1.9.2 Aircraft Spatial Attitude Control System

This subsystem includes the weight of the attitude control system which includes engines, valves, pressurant and residual propellants. It does not include the propellants and their associated tankage.

The system includes 4-pitch, 4-yaw, and 4-roll engines with each of the pitch and yaw engines having identical thrust levels, the thrust of the roll engines being half that of a pitch or yaw engine. All the engines are radiation cooled with a pitch and yaw thrust range from 30 to 100 lbs. The equation for attitude control system weight is

WACS = AC(57) * WTO ** AC(58) + AC(59) (6.1.67)

where

WACS	= weight of attitude control system, lbs.
WTO	= gross weight, lbs.
AC(57)	= ACS system weight coefficient (intercept)
AC(58)	= ACS system weight coefficient (slope)
AC(59)	= fixed ACS system weight, lbs.

The weight coefficients AC(57) and AC(58) represents the intercept and slope, respectively, for the data shown in Figure 6.1-35. The curves in Figure 6.1-35 represent three different size systems with total impulse ranges of 100,000; 200,000; and 300,000 lb-sec. When design data is not available to base a total impulse estimate on,the user may input AC(57) and AC(58) on the 200,000 lb-sec. curve. The X-15 had 235,000 lb-sec as a comparative bases.

6.1.9.3 Aircraft Attitude Control System Tankage

The attitude control system tankage weight includes the bladders, insulation, mounting, etc., but does not include the propellants. The tankage system assumes storable monopropellants, helium pressurization and titanium tank material. The equation for attitude control system tankage weight is

$$WACSTK = AC(64) * (WACSFU + WACSOX) + AC(65) \qquad (6.1.68)$$

where

WACSTK	= weight of attitude control system tankage, lbs.
WACSFU	= weight of ACS fuel, lbs.
WACSOX	= weight of ACS oxidizer, lbs.
AC(64)	= ACS tank weight coefficient
AC(65)	= fixed ACS tank weight, lbs.

The weight coefficient AC(64) is a ratio of tankage weight to propellant weight. A typical predesign value for AC(64) is 0.10.

6.1.9.4 Aircraft Aerodynamic Controls

The weight of this subsystem includes the total weight of the aerodynamic control system. It includes all control levers, push-pull rods, cables, and actuators from the control station up to but not including the aerodynamic surfaces. It will also include the autopilot if it is not integral with the navigation system. This weight does not include the hydraulic/pneumatic system weight. The aerodynamic controls data for straight and swept wing aircraft has been separated from the delta wing aircraft data. The basic equation for aerodynamic controls system weight is

$$WAERO = AC(60) * (WTO ** .666 * (ELBODY + GSPAN) ** .25) ** AC(111) + AC(61) \qquad (6.1.69)$$

where

WAERO	= weight of aerodynamic controls, lbs.
WTO	= gross weight, lbs.
ELBODY	= body length, ft.
GSPAN	= geometric wing span, ft.
AC(60)	= aerodynamic control system weight coefficient (intercept)
AC(111)	= aerodynamic control system weight coefficient (slope)
AC(61)	= fixed aerodynamic control system weight, lbs.

The weight coefficients AC(60) and AC(111) are obtained from Figure 6.1-36.

6.1.9.5 Aircraft Separation System

The separation system weight includes the system and attachments on the airplane for separating the two stages from each other. The equation for the separation system weight is

$$WSEP = AC(62) * WTO + AC(63) \qquad (6.1.70)$$

where

WSEP = weight of separation system, lbs.

WTO = gross weight, lbs.

AC(62) = separation system weight coefficient

AC(63) = fixed separation system weight, lbs.

The coefficient AC(62) is a constant that will scale the separation system weight as a function of gross weight. If design data is not available, and it is assumed that the major loads are reacted by the booster, a preliminary design value of AC(62) = 0.003 may be used.

6.1.10 Booster Orientation Controls and Separation

The total weight of the booster orientation controls and separation group is given by

$$BWORNT = BWGIMB + BWSEP \qquad (6.1.71)$$

where

BWGIMB = gimbal system weight

BWSEP = separation system weight

Expressions for each component weight are given below.

6.1.10.1 Booster Gimbal System

The booster gimbal (thrust-vector-control) actuation system data is identical to the aircraft gimbal system of Section 6.1.9.1. The gimbal system weight equation is

$$BWGIMB = BC(55) * BTDEL ** BC(110) + BC(56) \qquad (6.1.72)$$

where

BWGIMB = weight of engine gimbal system, lbs.

BTDEL = gimbal system delivered torque, lb-in., Section 6.1.9.1

BC(55) = gimbal system weight coefficient (intercept)

BC(110) = gimbal system weight coefficient (slope)

BC(56) = fixed gimbal system weight, lbs.

The weight coefficients BC(55) and BC(110) are obtained from Figures 6.1-33 and 6.1-34 as the aircraft system.

6.1.10.2 Booster Separation System

The separation system weight includes the system and attachments that are on the booster for separating the two stages from each other. The equation for the separation system weight is

$$BWSEP = BC(62) * BWTO + BC(63) \qquad (6.1.73)$$

where

BWSEP = weight of separation system, lbs.
BWTO = gross weight, lbs.
BC(62) = separation system weight coefficient
BC(63) = fixed separation system weight, lbs.

The coefficient BC(62) is a constant that will scale the separation system weight as a function of gross weight. If design data is not available, and if it is assumed that the major loads are reacted by the booster, a preliminary design value of BC(62) = 0.0005 may be used.

6.1.11 Aircraft Power Supply, Conversion and Distribution

The total weight of the aircraft power supply, conversion and distribution group is given by

$$WPWRSY = WELECT + WHYPNU \qquad (6.1.74)$$

where

WELECT = electrical system weight
WHYPNU = hydraulic/pneumatic system weight

Expressions for each component weight are given below.

6.1.11.1 Aircraft Electrical System

This subsystem includes the weight for the items required to generate, convert and distribute electrical power required to operate the various vehicle subsystems. Subsystems requiring electrical power are mainly electronics equipment, life support, environmental control equipment, lights, heaters, and blower motors. The electrical load varies with flight conditions and flight phase depending upon the demands of each subsystem. The electrical system data presented provides a preliminary weight representative of high speed fighter aircraft.

Major components represented in the system weight are batteries and AC generators, transformer rectifier units, control equipment and power distribution system. The equation for electrical system weight is

$$WELECT = AC(66) * (WTO **.5 * ELBODY **.25) ** AC(112) + AC(67) \qquad (6.1.75)$$

where

WELECT = weight of electrical system, lbs.
WTO = gross weight, lbs.
ELBODY = body length, ft.
AC(66) = electrical system weight coefficient (intercept)
AC(112) = electrical system weight coefficient (slope)
AC(67) = fixed electrical system weight, lbs.

The weight coefficients AC(66) and AC(112) are obtained from Figure 6.1-37.

6.1.11.2 AIRCRAFT HYDRAULIC/PNEUMATIC SYSTEM

The hydraulic/pneumatic system is comprised of the system components to produce fluid or pneumatic pressure, control equipment, storage vessels, hydraulic fluid, and a distribution system up to but not including the various functional branches, actuators, etc. The equation for hydraulic/pneumatic system weight is

$$\text{WHYPNU} = \text{AC}(68)\left[\left((\text{SWING}+\text{SHORZ}+\text{SVERT}) * \text{QMAX}/1000\right) ** 0.334 + (\text{ELBODY} + \text{STSPAN}) ** 0.5 * \text{TYTAIL}\right] ** \text{AC}(113) + \text{AC}(69) \tag{6.1.76}$$

where

WHYPNU = weight of hydraulic/pneumatic system, lbs.
SWING = gross wing area, ft.^2
SHORZ = horizontal stabilizer planform area, ft.^2
SVERT = vertical fin planform area, ft.^2
QMAX = maximum dynamic pressure, lbs./ft.^2
ELBODY = body length, ft.
STSPAN = structural span (along .5 chord), ft.^2
TYTAIL = type tail coefficient
1.0 for conventional tail
1.25 for delta planform
1.5 for all moving horizontal and/or vertical
AC(68) = hydraulic/pneumatic system weight coefficient (intercept)
AC(113) = hydraulic/pneumatic system weight coefficient (slope)
AC(69) = fixed hydraulic/pneumatic system weight, lbs.

The weight coefficients AC(68) and AC(113) are obtained from 6.1-38.

6.1.12 Booster Power Supply, Conversion and Distribution

Total weight of the booster power supply, conversion, and distribution system is given by

$$\text{BWPWSY} = \text{BWELEC} + \text{BWHYPN} \tag{6.1.77}$$

where

BWELEC = electrical system weight
BWHYPN = hydraulic/pneumatic system weight

Expressions for each component weight are given below.

6.1.12.1 Booster Electrical System

The electrical system consists of a distribution system only. The booster electrical system is assumed to be a function of body length and the scaling equation is

$$\text{BWELEC} = \text{BC}(66) * \text{BLBODY} + \text{BC}(67) \tag{6.1.78}$$

where
BWELEC = weight of electrical system, lbs.
BLBODY = body length, ft.
BC(66) = electrical system weight coefficient, lbs./ft.
BC(67) = fixed electrical system weight, lbs.

If design data is not available, a preliminary design value of BC(66) = 2.0 may be used.

6.1.12.2 Booster Hydraulic/Pneumatic System

The hydraulic/pneumatic system for the booster consists of control valves and distribution system. The hydraulic/pneumatic power generation will be obtained from the aircraft system. The equation for booster hydraulic/pneumatic system weight is

$$BWHYPN = BC(68) * BLBODY + BC(69) \qquad (6.1.79)$$

where
BWHYPN = weight of hydraulic/pneumatic system, lbs.
BLBODY = body length, ft.
BC(68) = hydraulic/pneumatic system weight coefficient, lbs./ft.
BC(69) = fixed hydraulic/pneumatic system weight, lbs.

If design data is not available, a preliminary design value of BC(68) = 4.0 may be used.

6.1.13 Aircraft Avionics

The avionic system includes the guidance and navigation system, the instrumentation, and the communications system.

The *guidance and navigation system* includes those items necessary to insure that the vehicle position and its trajectory is known at all times. This system also generates commands for the flight control system for changing or correcting the vehicle heading.

The *instrumentation system* provides for a weight allocation assigned to the basic instruments normally required for sensing and readout of the normal flight parameters needed for monitoring a flight program. In addition to this basic system there are many possible mission oriented instrumentation functions that may be required. Weight allocation for the instrumentation system is normally part of a design study for a particular vehicle design and mission requirement.

The *communication system* weight allocation is for all equipment necessary to provide for the communication between vehicle and air or ground stations including communication within the vehicle itself.

The equation for avionic system weight is

$$WAVONC = AC(70) * WTO ** AC(114) + AC(71) \qquad (6.1.80)$$

where
WAVONC = weight of avionics system, lbs.
WTO = gross weight, lbs.
AC(70) = avionic system weight coefficient (intercept)
AC(114) = avionic system weight coefficient (slope)
AC(71) = fixed avionic system weight, lbs.

The weight coefficients AC(70) and AC(114) are obtained from Figure 6.1-39. This data represents systems of advanced capability with significant fire control capability (F-111 and B-58 type).

6.1.14 Aircraft Crew Systems

The crew provisions include the equipment and personnel environment control system, crew compartment insulation, personnel accommodations, fixed life support equipment, emergency equipment, crew station controls and panels.

The *equipment environmental control system* is used to maintain the correct operating conditions for vehicle system equipment. The function of the *personnel environmental control system* is to provide an acceptable environmental condition for the crew. This includes temperature, atmosphere and pressurization equipment and supports. The *compartment insulation* is required for controlling environment in conjunction with the overall active environmental system. The *accommodations for personnel* includes seats, supports, restraints, shock absorbers, ejection mechanisms, etc. The *fixed lift support system* includes food containers, waste management, hygiene equipment, etc. The *fixed emergency equipment* includes a built-in fire extinguishing system, life rafts, etc. The *crew station control and panels* is for installation of crew station flight controls, instrument panels, control pedestals and stands.

The crew provisions are a combined function of gross weight, crew size, and fixed weights. Therefore, the weight penalty may be represented by one equation and the various inputs collected and summed from Table 6.1-5. The equation for crew provisions weight is

$$WCPROV = AC(74) * WTO + AC(80) * CREW + AC(75) \qquad (6.1.81)$$

where
WCPROV = weight of crew provisions, lbs.
WTO = gross weight, lbs.
CREW = number of crew members
AC(74) = equipment ECS weight coefficient
AC(80) = crew provisions weight coefficient
AC(75) = fixed crew provisions weight, lbs.

6.1.15 Aircraft Design Reserve

The input for contingency and growth permits a proportion of dry weight and/or a fixed weight to be set aside for growth allowance, design unknowns, etc. The aircraft dry weight is summed by the equation:

$$WDRY = WSURF + WBODY + WTPS + WGEAR + WPROPU + WORNT + WPWRSY + WAVONC + WCPROV \quad (6.1.82)$$

This value for dry weight is then used in the equation for contingency and growth which is

$$WCONT = AC(98) * WDRY + AC(99) \quad (6.1.83)$$

where
WCONT = weight of contingency and growth, lbs.
WDRY = stage dry weight, lbs.
AC(98) = contingency and growth coefficient
AC(99) = fixed contingency and growth weight, lbs.

The aircraft weight empty is summed by the equation

$$WEMPTY = WDRY + WCONT \quad (6.1.84)$$

6.1.16 Booster Design Reserve

The input for contingency and growth permits a proportion of dry weight and/or a fixed weight to be set aside for growth allowance, design unknowns, etc. The booster dry weight is summed by equation (6.1.85)

$$BWDRY = BWBODY + BWTPS + BWPRPL + BWORNT + BWPWSY \quad (6.1.85)$$

This value for dry weight is then used in the equation for contingency and growth which is

$$BWCONT = BC(98) * BWDRY + BC(99) \quad (6.1.86)$$

where
BWCONT = weight of contingency and growth, lbs.
BWDRY = stage dry weight, lbs.
BC(98) = contingency and growth coefficient
BC(99) = fixed contingency and growth weight, lbs.

The booster weight empty is summed by the equation

$$BWEMTY = BWDRY + BWCONT \quad (6.1.87)$$

6.1.17 Aircraft Crew and Crew Life Support

This section includes the crew, gear and accessories as well as the crew life support. The *crew, gear, and accessories* includes crew, constant wear and protection garments, pressure suits, head gear, belt packs, personal parachutes, portable hygienic equipment, maps, manuals, log books, portable fire extinguishers, maintenance tools, etc. The *crew life support* includes food, water, portable containers, medical equipment, survival kits, etc. The equation for crew and crew life support weight is

$$WCREW = AC(72) * CREW + AC(73) \qquad (6.1.88)$$

where

WCREW = weight of crew, gear, and crew life support, lbs.
CREW = number of crew members
AC(72) = crew weight coefficient
AC(73) = fixed crew weight, lbs.

Typical values for the crew dependent weight is shown in Table 6.1-6. The input coefficient AC(73) is used for fixed crew life support weight. A typical input for AC(73) is shown in Table 6.1-6. This coefficient may also be used to input a fixed weight for crew and crew life support. When AC(73) is used for this purpose the coefficient AC(72) must be set to zero.

6.1.18 Payload

6.1.18.1 Aircraft Payload

The aircraft payload weight is input as WPAYLD. The value is determined by the user.

6.1.18.2 Booster Payload

The booster payload consists of the upper stage

6.1.19 Aircraft Propellants

6.1.19.1 Aircraft Trapped Propellants

The equation for *trapped fuel* weight is

$$WFTRAP = AC(92) * WFUEL + AC(93) \qquad (6.1.89)$$

where

WFTRAP = weight of fuel trapped in tank and lines, lbs.
WFUEL = weight of main impulse plus reserve fuel, lbs.

AC(92) = trapped fuel weight coefficient
AC(93) = fixed trapped fuel weight, lbs.

A typical input value for AC(92) will vary from 0.005 to 0.03.

The equation for *trapped oxidizer* weight is

$$WOTRAP = AC(94) * WOXID + AC(95) \tag{6.1.90}$$

where
WOTRAP = weight of oxidizer trapped in tank and lines, lbs.
WOXID = weight of main impulse plus reserve oxidizer, lbs.
AC(94) = trapped oxidizer weight coefficient
AC(95) = fixed trapped oxidizer weight, lbs.

A tppical input value for AC(94) will vary from 0.005 to 0.03

6.1.19.2 Aircraft Reserve Propellant

The equation for *reserve fuel* weight is

$$WFRESV = AC(84) * WFUELM + AC(85) \tag{6.1.91}$$

where
WFRESV = weight of fuel reserve, lbs.
WFUELM = weight of main impulse fuel, lbs.
AC(84) = reserve fuel weight coefficient
AC(85) = fixed reserve fuel weight, lbs.

The equation for *reserve oxidizer* weight is

$$WORESV = AC(86) * WOXIDM + AC(87) \tag{6.1.92}$$

where
WORESV = weight of oxidizer reserve, lbs.
WOXIDM = weight of main impulse oxidizer, lbs.
AC(86) = reserve oxidizer weight coefficient
AC(87) = fixed reserve oxidizer weight, lbs.

A typical input value for AC(84) and AC(86) will vary from 0.01 to 0.20.

6.1.19.3 Attitude Control System (ACS) Propellants (In-Flight Losses)

The attitude control system is based on a monopropellant system. The equations for ACS fuel and oxidizer weight are

$$WACSFU = AC(96) * WTO + AC(97) \tag{6.1.93}$$

and

$$WACSOX = WACSFU * OFACS \tag{6.1.94}$$

where

WACSFU = weight of ACS fuel, lbs.
WTO = gross weight, lbs.
AC(96) = ACS fuel weight coefficient
AC(97) = fixed ACS fuel weight, lbs.
WACSOX = weight of ACS oxodizer, lbs.
OFACS = ACS oxidizer to fuel mixture ratio by weight

A predesign value for AC(96) from 0.001 to 0.005 may be used.

6.1.19.4 Main Propellants

The main impulse propellant equations are

$$WFUELM = WPMAIN/(1. + OF) \tag{6.1.95}$$

and

$$WOXIDM = WFUELM * OF \tag{6.1.96}$$

where

WFUELM = weight of main impulse fuel, lbs.
WPMAIN = weight of main impulse propellant, lbs.
OF = main oxidizer to fuel mixture ratio by weight
WOXIDM = weight of main impulse oxidizer, lbs.

6.1.20 Aircraft Weight Summary

The total weight of main impulse plus reserve fuel and the total weight of main impulse plus reserve oxidizer are summed by the equations:

$$WFUEL = WFUELM + WFRESV \tag{6.1.97}$$

$$WOXID = WOXIDM + WORESV \tag{6.1.98}$$

The total weight of fuel and oxidizer in the tanks are summed by the equations

$$WFUTOT = WFUEL + WFTRAP \tag{6.1.99}$$

$$WOXTOT = WOXID + WOTRAP \tag{6.1.100}$$

The total weight of propellant tanked is summed by the equation

$$WP - WFUTOT + WOXTOT \tag{6.1.101}$$

The operating weight empty is summed by the equation

$$WOPMTY = WEMPTY + WRESID + WCREW + WACSFU + WACSOX \tag{6.1.102}$$

The zero fuel weight is summed by the equation

$$WZROFU = WOPMTY + WPAYLD \tag{6.1.103}$$

The gross weight is summed by the equation

$$WTO = WZROFU + WPMAIN + WFRESV + WORESV \quad (6.1.104)$$

The landing weight is calculated by the equation

$$WLAND = WTO - AC(100) * WPMAIN \quad (6.1.105)$$

6.1.21 Booster Propellants

6.1.21.1 Booster Trapped Propellant

The equation for *trapped fuel* weight is

$$BWFTRP = BC(92) * BWFUEL + BC(93) \quad (6.1.106)$$

where
BWFTRP = weight of fuel trapped in tank and lines, lbs.
BWFUEL = weight of main impulse plus reserve fuel, lbs.
BC(92) = trapped fuel weight coefficient
BC(93) = fixed trapped fuel weight, lbs.

A typical input value for BC(92) will vary from 0.005 to 0.3

The equation for *trapped oxidizer* weight is

$$BWOTRP = BC(94) * BWOXID + BC(95) \quad (6.1.107)$$

where
BWOTRP = weight of oxidizer trapped in tank and lines, lbs.
BWOXID = weight of main impulse plus reserve oxidizer, lbs.
BC(94) = trapped oxidizer weight coefficient
BC(95) = fixed trapped oxidizer weight, lbs.

A typical input value for BC(94) will vary from 0.005 to 0.03.

6.1.21.2 Booster Reserve Propellant

The booster propellant residuals are summed by the equation

$$BWRESD = BWFTRP + BWOTRP \quad (6.1.108)$$

The equation for *reserve fuel* weight is

$$BWFRES = BC(84) * BWFULM + BC(85) \quad (6.1.109)$$

where
BWFRES = weight of fuel reserve, lbs.

BWFULM = main impulse fuel weight, lbs.
BC(84) = reserve fuel weight coefficient
BC(85) = fixed reserve fuel weight, lbs.

A typical input value for BC(84) will vary from 0.01 to 0.20

The equation for *reserve oxidizer* weight is

$$BWORES = BC(86) * BWOXM + BC(87) \qquad (6.1.110)$$

where
BWORES = weight of oxidizer reserve, lbs.
BWOXM = main impulse oxidizer weight, lbs.
BC(86) = reserve oxidizer weight coefficient
BC(87) = fixed reserve oxidizer weight, lbs.

A typical input value for BC(86) will vary from 0.01 to 0.20

6.1.21.3 Main Propellants

If a mass ratio and mixture ratio are input, the propellants are calculated by the following equations:

$$BWPMAN = BWTO * (BMASRT -1.)/BMASRT \qquad (6.1.111)$$

$$BWFULM = BWPMAN/(BMIXRT +1.) \qquad (6.1.112)$$

$$BWOXM = BWPMAN - BWFULM \qquad (6.1.113)$$

where
BWPMAN = weight of main impulse propellant, lbs.
BWTO = gross weight, lbs.
BMASRT = stage mass ratio
BWFULM = main impulse fuel, lbs.
BMIXRT = oxidizer to fuel mixture ratio by weight
BWOXM = main impulse oxidizer, lbs.

If the main impulse propellant is calculated and input as BWPMAN, the weight of main impulse fuel and oxidizer are then calculated by the other two equations.

6.1.22 Booster Weight Summary

The weight of the main impulse and reserve fuel and oxidizer are summed by the equations

$$BWFUEL = BWFULM + BWFRES \qquad (6.1.114)$$

$$BWOXID = BWOXM + BWORES \qquad (6.1.115)$$

The total weight of the fuel and oxidizer in the tank are summed by the equations:

$$BWFUTL = BWFUEL + BWFTRP \quad (6.1.116)$$

$$BWOXTL = BWOXID + BWOTRP \quad (6.1.117)$$

The booster operating weight empty is summed by the equation

$$BWOPMT = BWEMPTY + BWRESD \quad (7.1.118)$$

The booster zero fuel weight (or burnout) is summed by the equation

$$BWZROF = BWOPMT + WTO \quad (6.1.119)$$

The booster gross weight is summed by the equation

$$BWTO = BWZROF + BWPMAN + BWFRES + BWORES \quad (6.1.120)$$

6.1.23 Volume and Geometry Calculations

The References 1 and 2 VSAC program contains several geometric scaling options, all of which are based on straightforward magnification or diminution of the nominal configuration. These scaling options are generally inadequate for realistic configuration perturbations. Hence, in the ODIN/RLV alternate sources of geometric perturbations must be employed such as the geometry programs of Section 3.

REFERENCES:

1. Oman, B.: Vehicle Synthesis for High Speed Aircraft, Volume I - Analysis Techniques and User Instructions, AFFDL-TR-71-40, June 1971.

2. Oman, B.: Vehicle Synthesis for High Speed Aircraft, Volume II - Weights and Geometry Analysis, AFFDL-TR-71-40, June 1971.

3. Hague, D. S.: Weight Analysis for Advanced Aerospace Vehicle Systems and Users Guide to the Program WAAVS II, Aerophysics Research Corporation, to be issued as a high number NASA contractor report.

4. Anon.: Space Shuttle Synthesis Program (SSSP), Volume I, Part 1 - Engineering and Programming Discussion, General Dynamics Report GDC-DB70-002, December 1970

5. Anon.: Space Shuttle Synthesis Program (SSSP), Volume I, Part 2 - Program Operating Instructions, General Dynamics Report GDC-DB70-002, December 1970.

TABLE 6.1-1 TYPICAL FAIRING WEIGHTS

No.	Airplane	Nose Gear Door	Windshield and Canopy	Main Gear Doors	Flight Opening Doors	Speed Brakes	Total Secondary Structure	Body Wetted Area	AC(17)
1	T-33	20	366	42	0	53	481	533	0.90
2	F-104A	21	168	197	0	17	403	669	0.60
3	XF-88	32	174	177	0	31	414	715	0.58
4	F-105B	41	293	40	384	402	1160	1030	1.13
5	F-105D	35	278	169	480	402	1364	991	1.38
6	F-101C	27	251	136	407	174	995	1036	0.96
7	F-101B	28	376	127	272	150	953	827	1.15
8	F-102A	30	302	166	526	35	1059	991	1.07
9	F-106A	70	662	171	632	72	1607	1222	1.32
10	B-58A	85	486	235	0	0	806	1373	0.59

TABLE 6.1-2 SECONDARY STRUCTURE DATA

Fairing Type	WF at Q = 400 lbs/ft^2 and T = 400^{o}F
Aerodynamic Shroud	4.80
Canopy Fairing	4.00
Equipment Fairing	1.50
Dorsal Fairing	2.00
Cable Fairing	1.50
Landing Gear Fairing	2.00

TABLE 6.1-3 TYPICAL SPIKE WEIGHTS

TYPE OF SPIKE	AC(109)
1/2 ROUND - FIXED	35
FULL ROUND - TRANSLATING	70
FULL TRANSLATING AND EXPANDING	290

TABLE 6.1-4 AIRCRAFT SYSTEM GIMBAL PARAMETERS

Delivered Torque	6,000 to 3,000,000 lb-in
Nozzle Deflection	2 to 20 degrees
Nozzle Deflection Rate	5 to 25 degrees/second
Operating Time	50 to 1200 seconds
Thermal Environment	-420 to +400 ^{o}F
Acceleration	2.5 to 15g

TABLE 6.1-5 TYPICAL AIRCRAFT CREW PROVISION INPUTS

SYSTEM DESCRIPTION	AC(74)	AC(80)	AC(75)
Equipment Environmental Control	0.0005	-	100
Personnel Environmental Control	-	10	250
Compartment Insulation	-	50	-
Accommodations for Personnel			
B-70 Type Encapsulated Seat	-	570	-
X-15 Ejection Seat	-	300	-
Gemini Ejection Seat	-	220	-
Lightweight Ejection Seat	-	100	-
Conventional Crew Seat	-	50-120	-
Fixed Life Support	-	10	-
Fixed Emergency Equipment	-	50	-
Crew Station Controls and Panels	-	40	50

TABLE 6.1-6 TYPICAL INPUTS FOR CREW AND CREW LIFT SUPPORT

DESCRIPTION	AC(72)	AC(73)
Crew, Gear and Accessories	220-290	---
Crew Life Support	2-5	25-50

ORIGINAL PAGE IS
OF POOR QUALITY

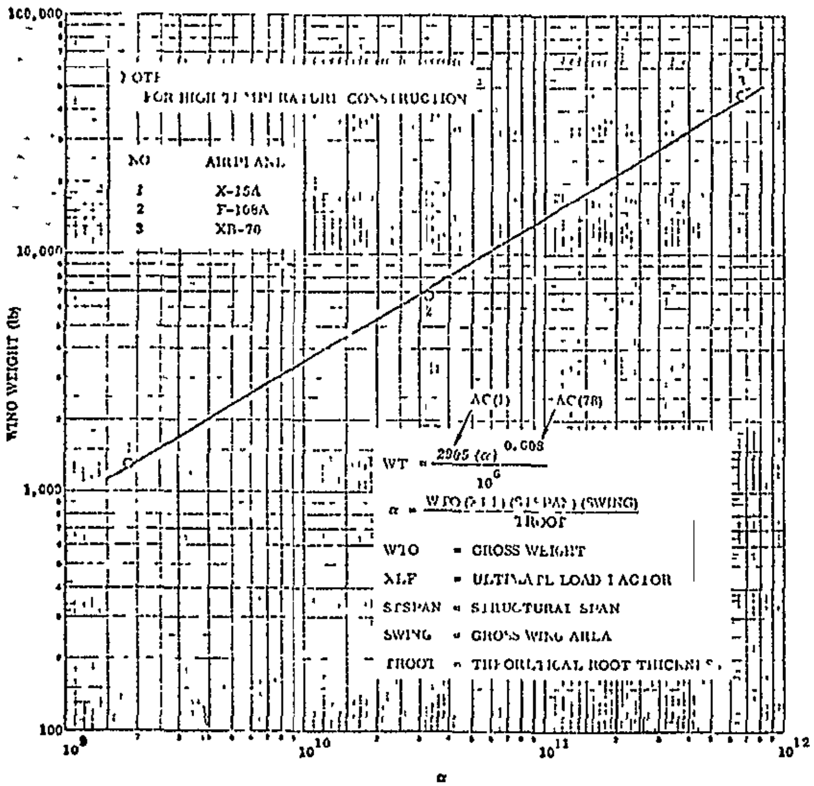


FIGURE 6.1-1. ALUMINUM WING WEIGHTS (NO VARIABLE SWEEP PENALTY INCLUDED)

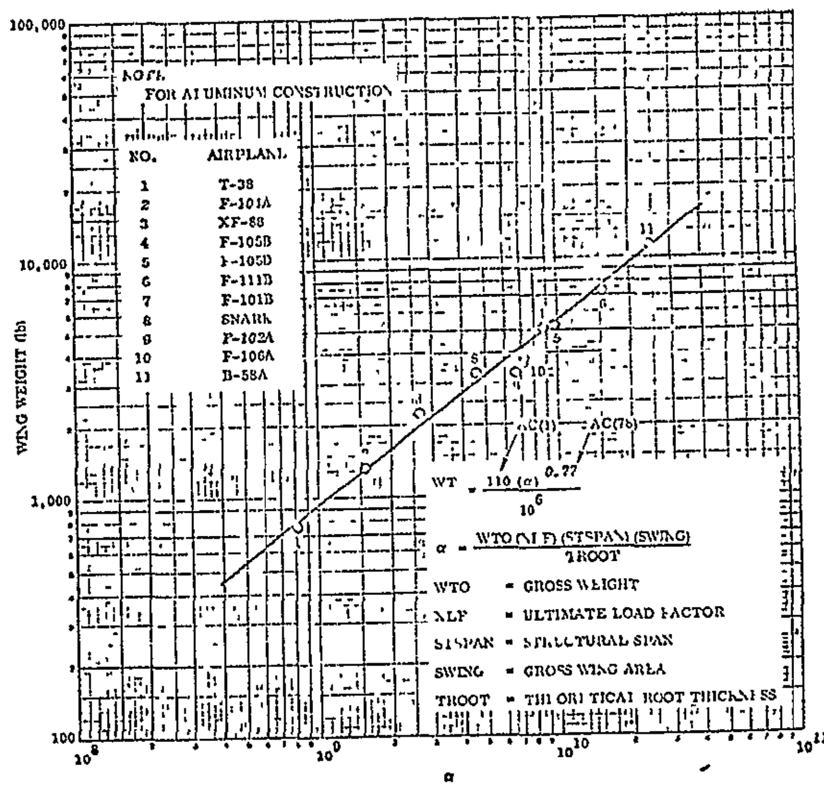


FIGURE 6.1-2. HIGH TEMPERATURE AND THIN SUPERSONIC WING WEIGHTS

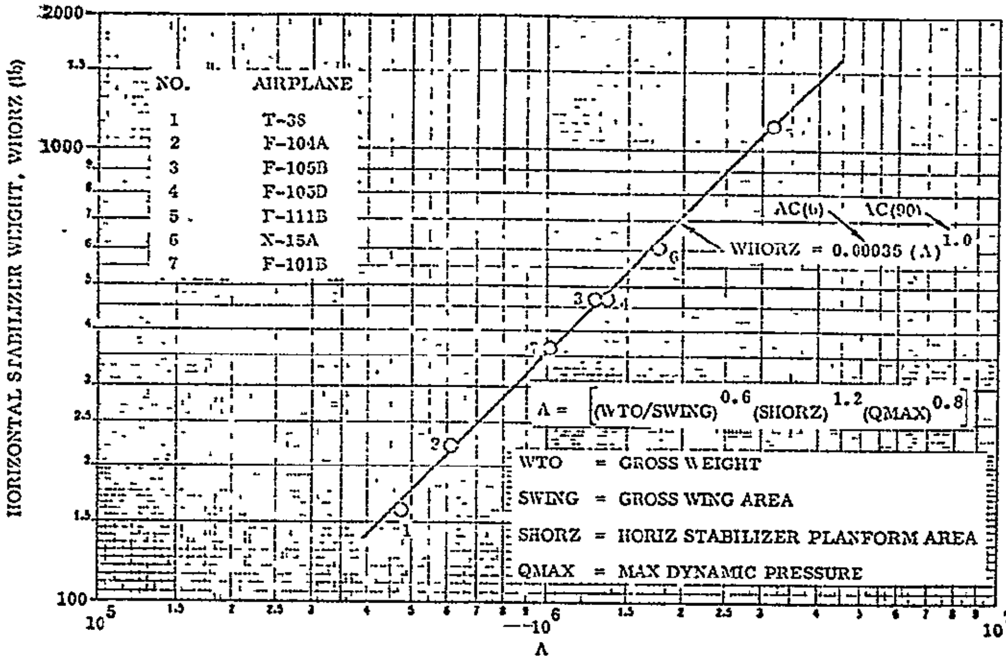


FIGURE 6.1-3. VERTICAL FIN WEIGHTS

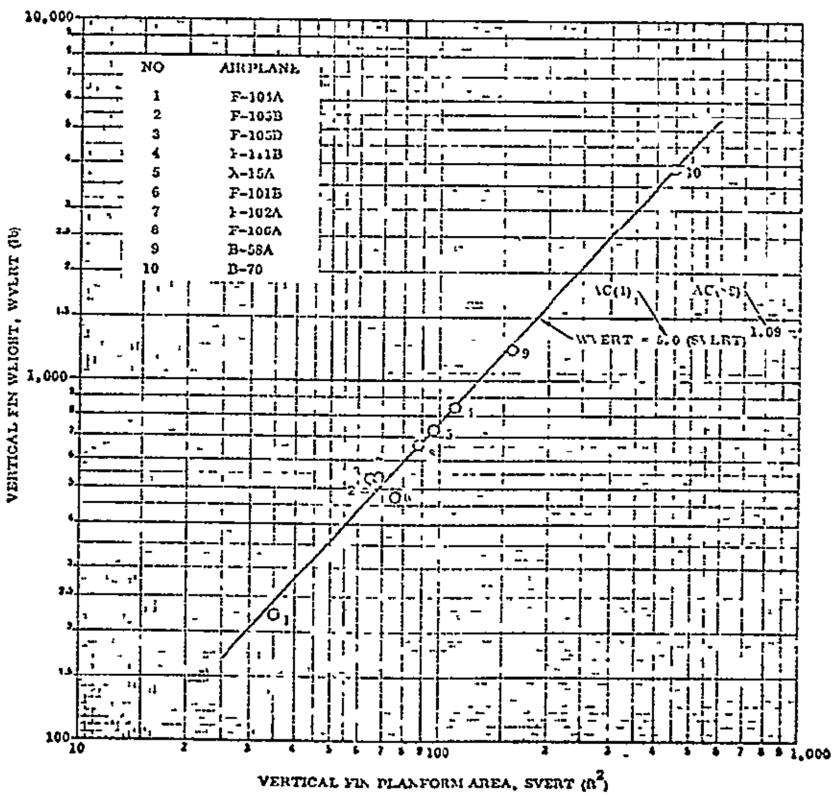


FIGURE 6.1-4. HORIZONTAL STABILIZER WEIGHTS

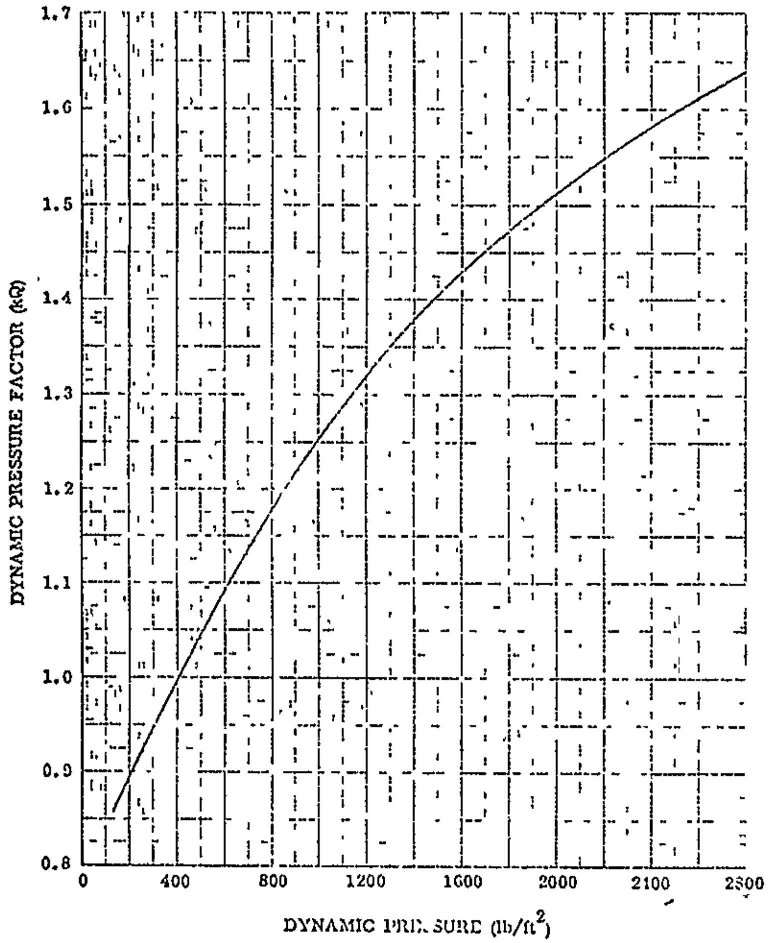


FIGURE 6.1-5. FAIRING WEIGHT DYNAMIC PRESSURE FACTOR

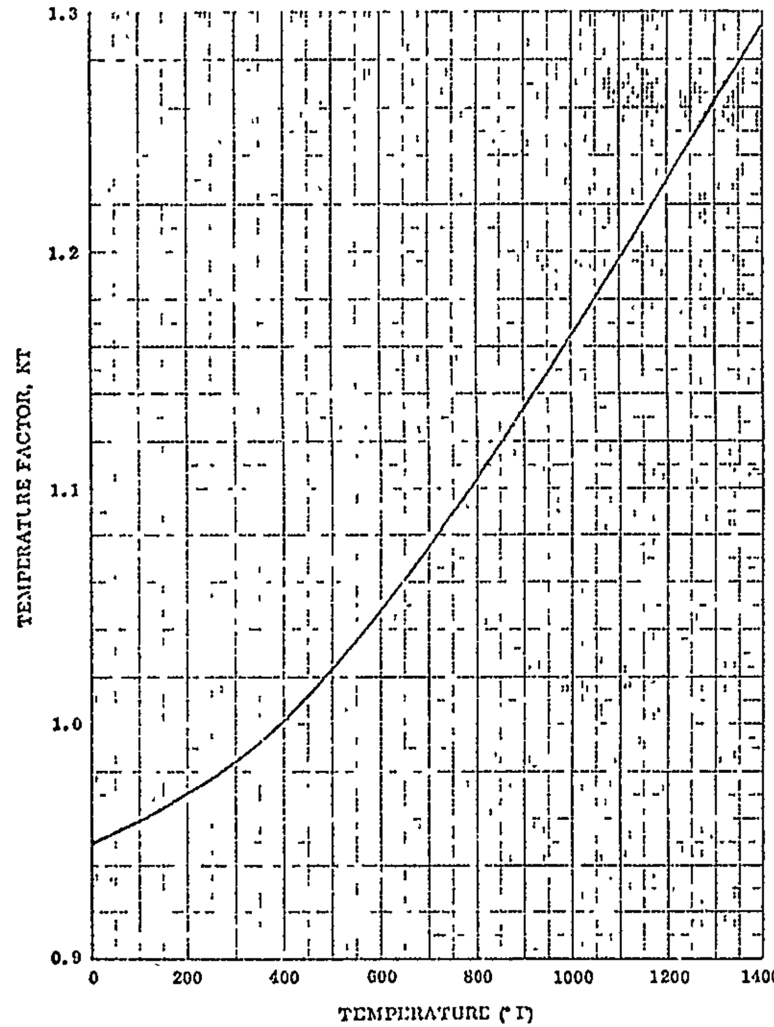


FIGURE 6.1-6. FAIRING WEIGHT TEMPERATURE FACTOR

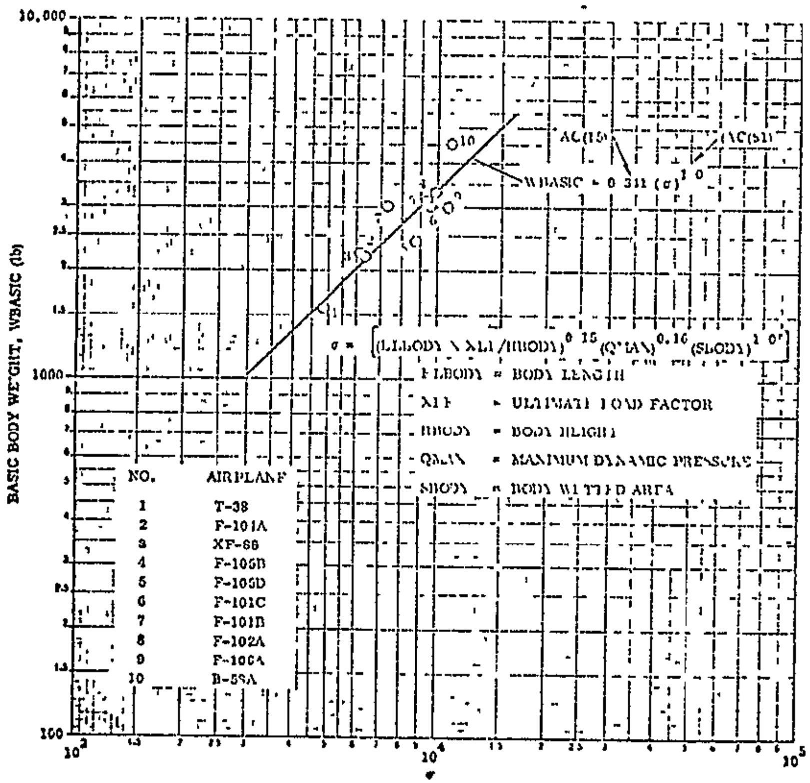


FIGURE 6.1-7. BASIC AIRCRAFT BODY WEIGHTS

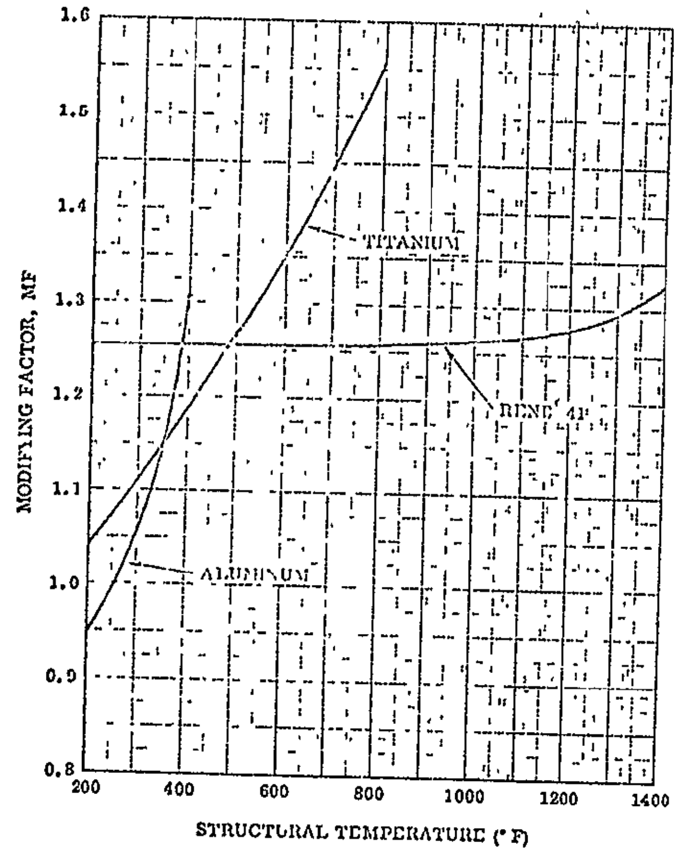


FIGURE 6.1-8. BASIC AIRCRAFT BODY
WEIGHT TEMPERATURE FACTOR

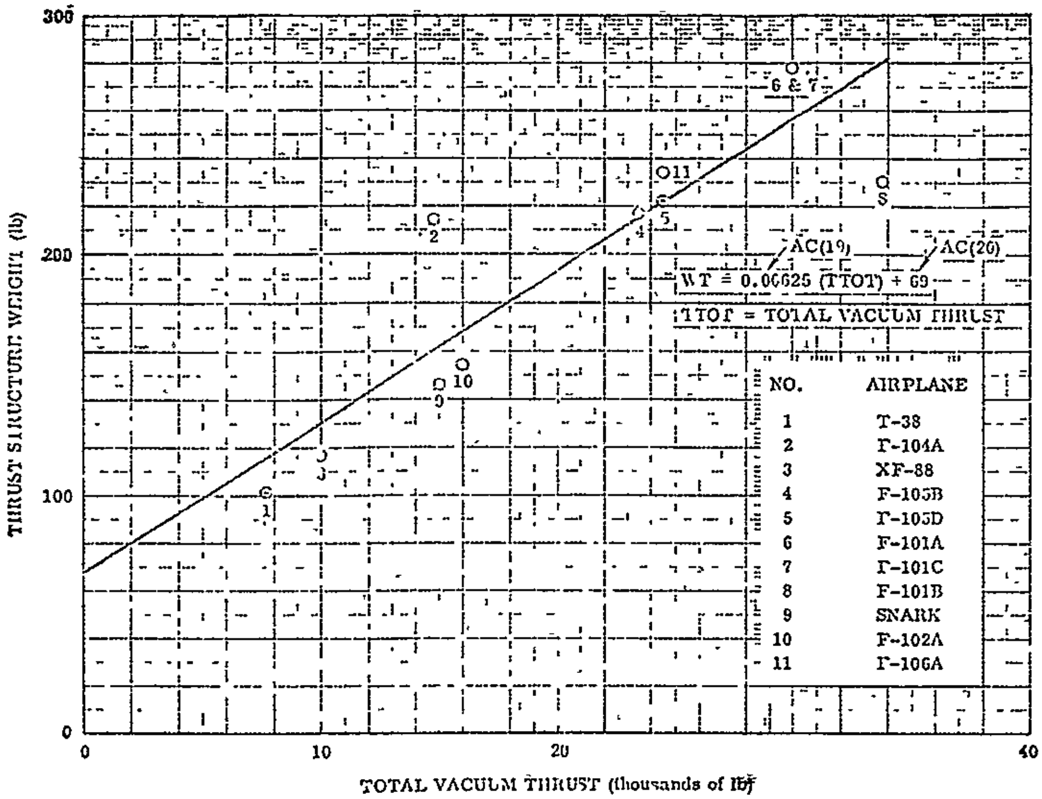


FIGURE 6.1-9 AIRBREATHING THRUST STRUCTURE WEIGHT

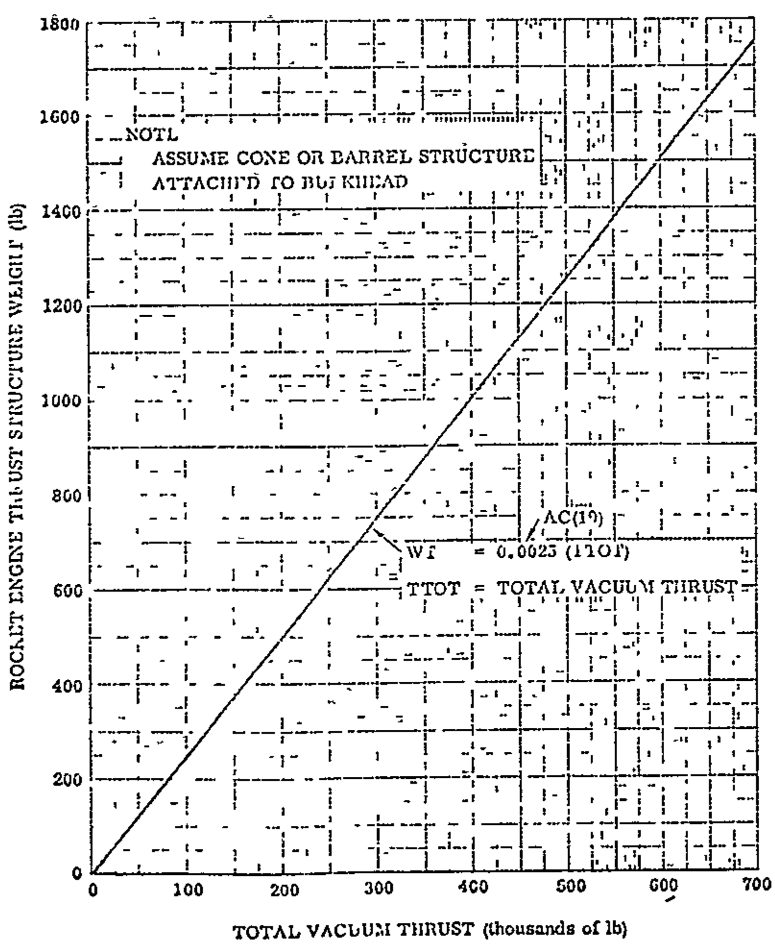


FIGURE 6.1-10 ROCKET ENGINE THRUST STRUCTURE

C-4

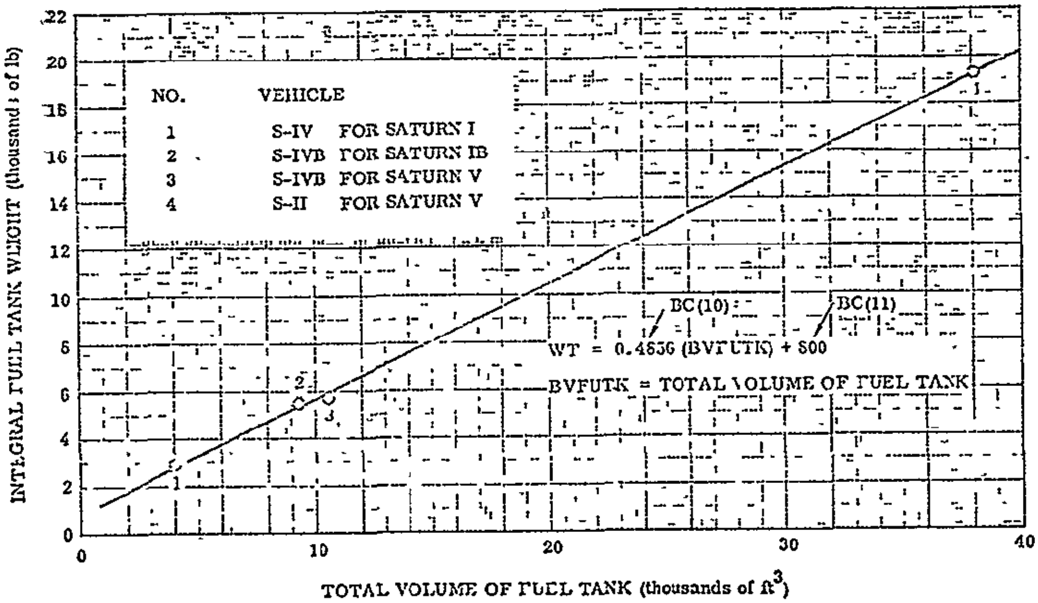


FIGURE 6.1-11. BOOSTER INTEGRAL FUEL TANK WEIGHTS

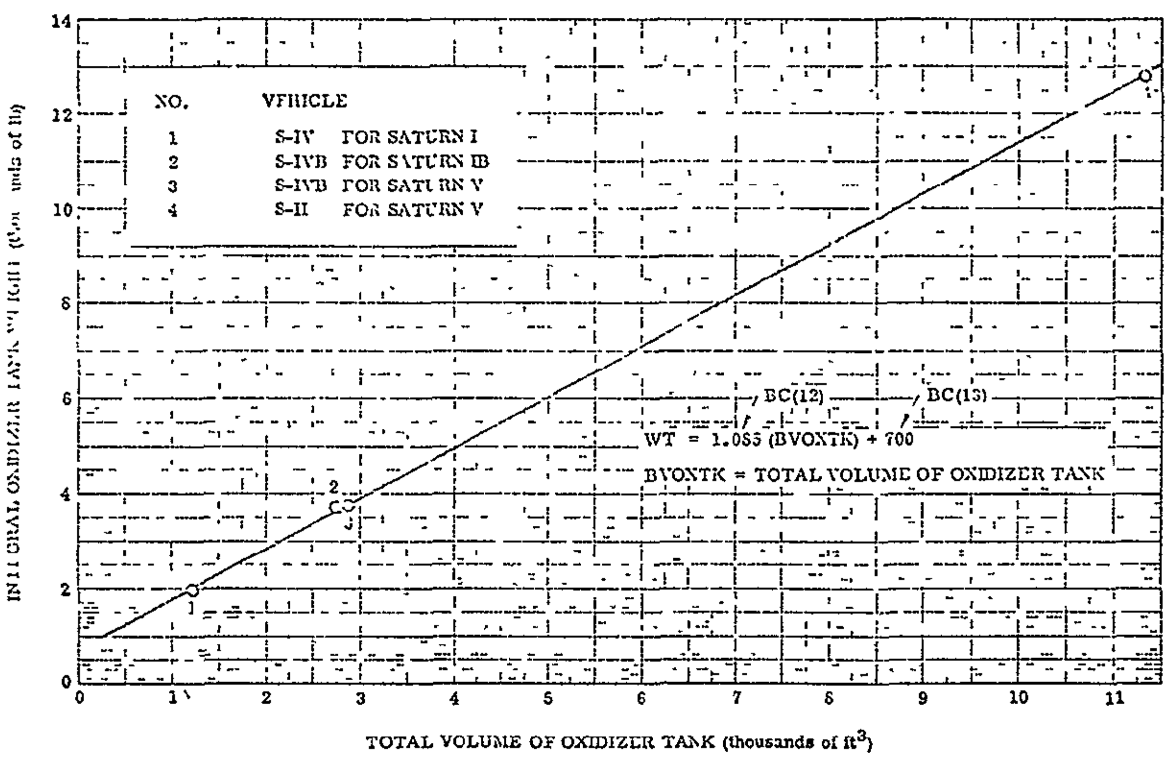


FIGURE 6.1-12. BOOSTER INTEGRAL OXIDIZER TANK WEIGHTS.

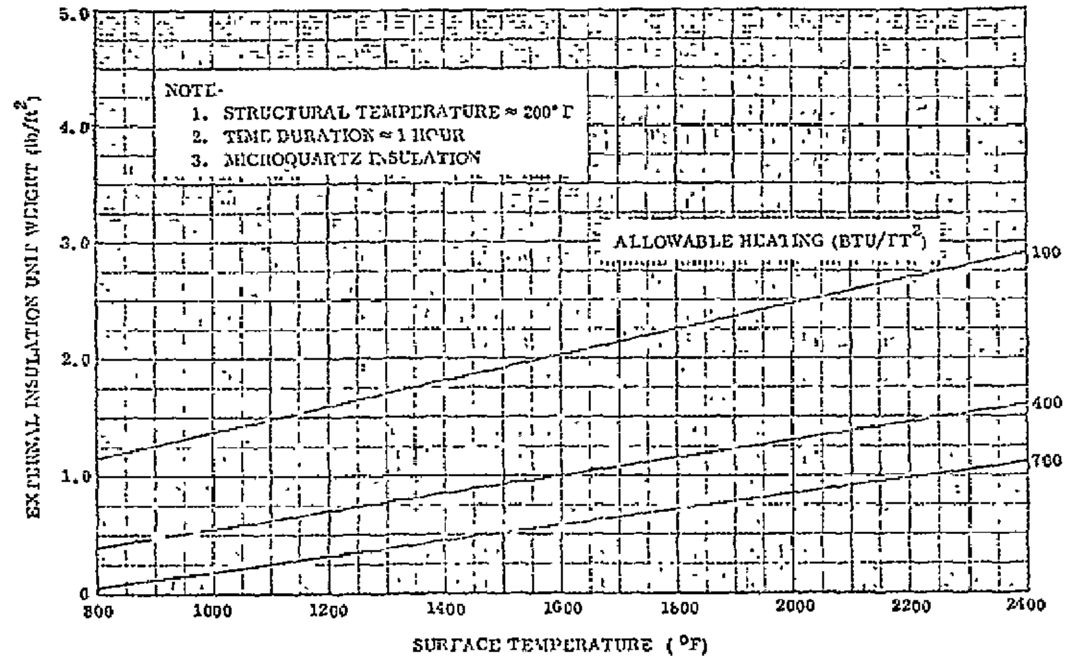


FIGURE 6.1-13. AIRCRAFT INSULATION WEIGHTS

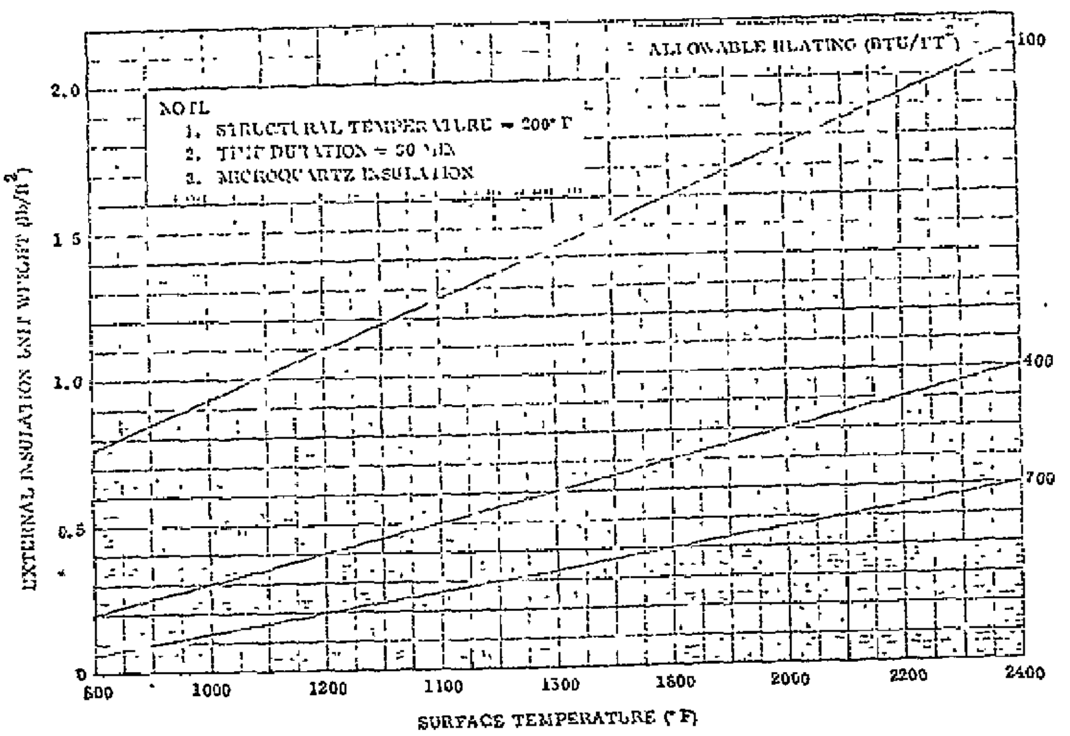


FIGURE 6.1-14. BOOSTER INSULATION WEIGHT

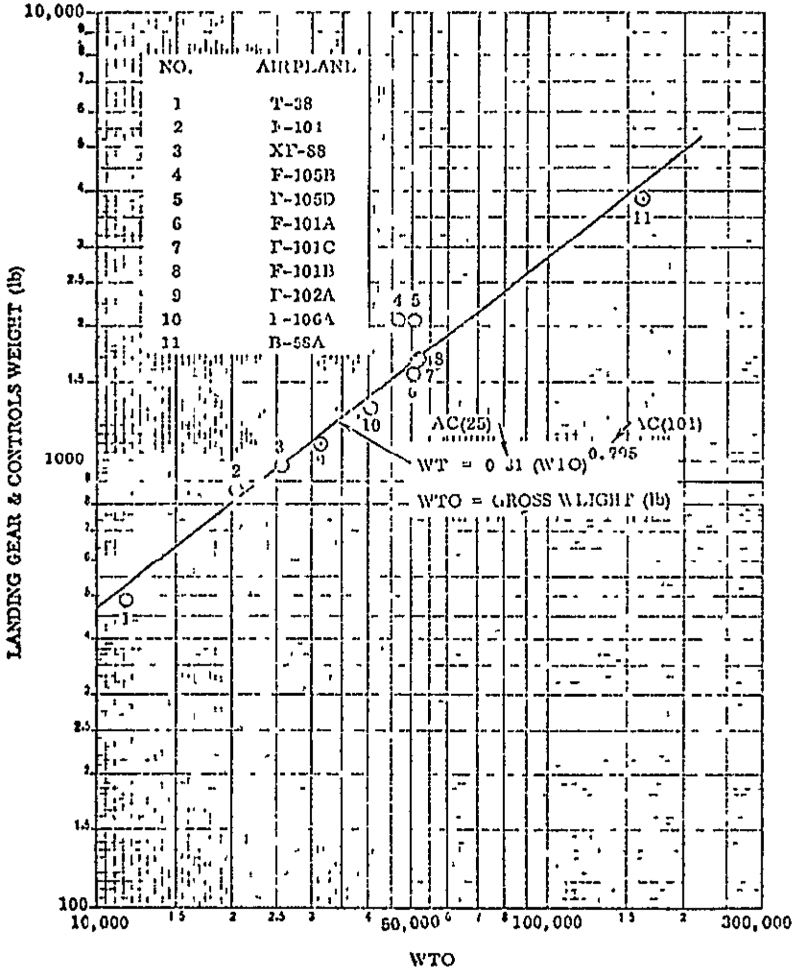


FIGURE 6.1-15. AIRCRAFT LANDING GEAR WEIGHT

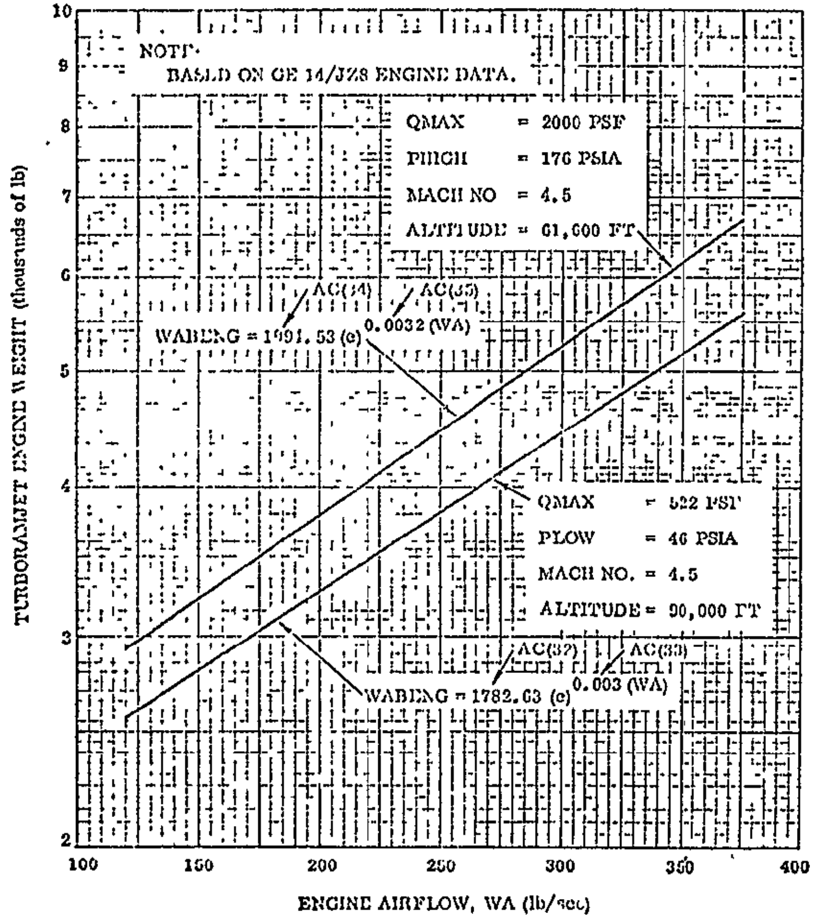


FIGURE 6.1-16. AIRCRAFT TURBORAMJET ENGINE WEIGHT

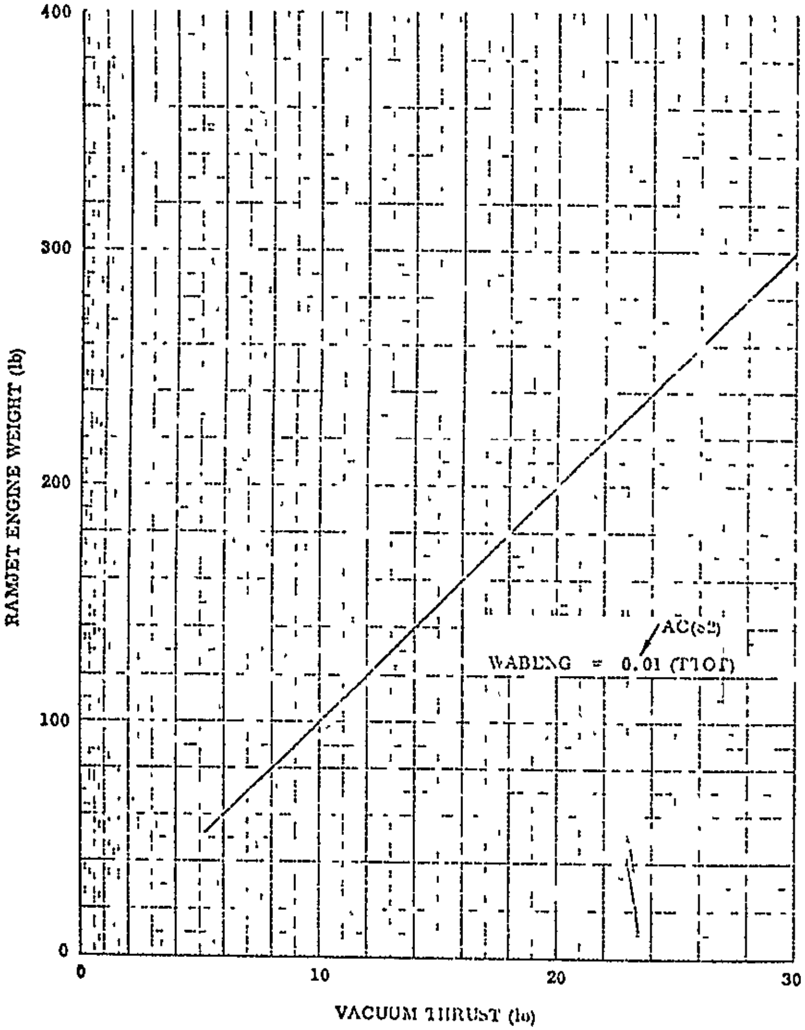


FIGURE 6.1-17. AIRCRAFT RAMJET ENGINE WEIGHT

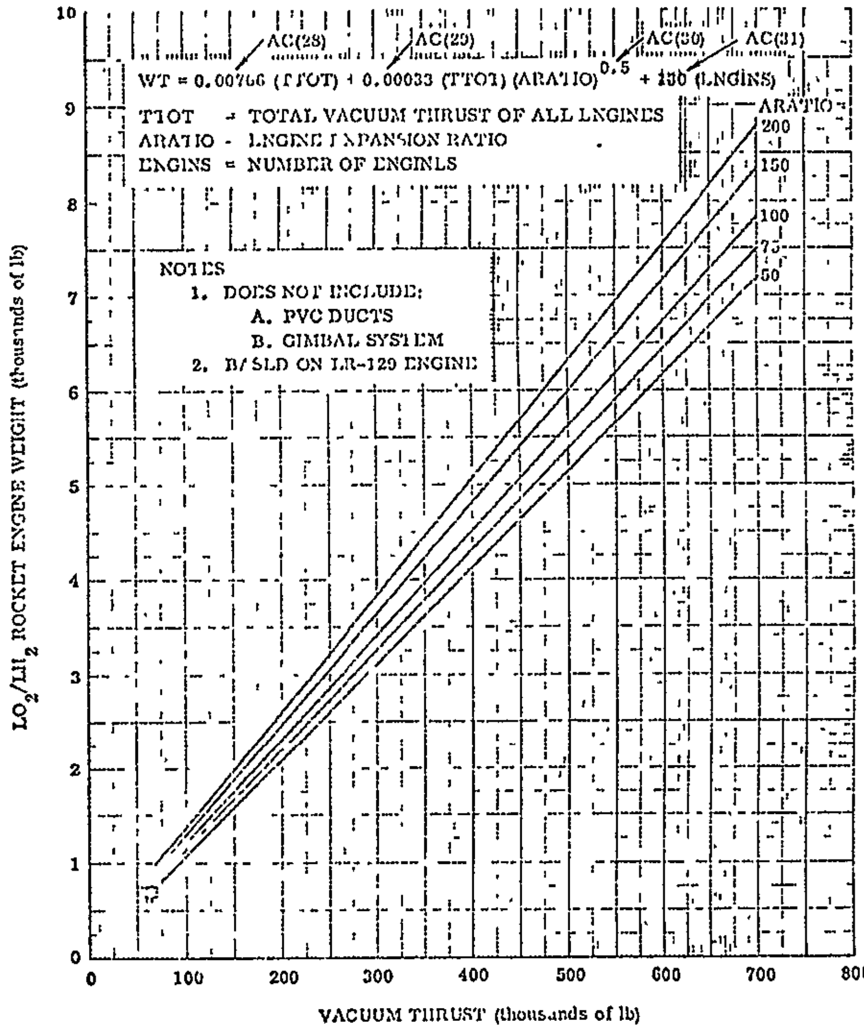
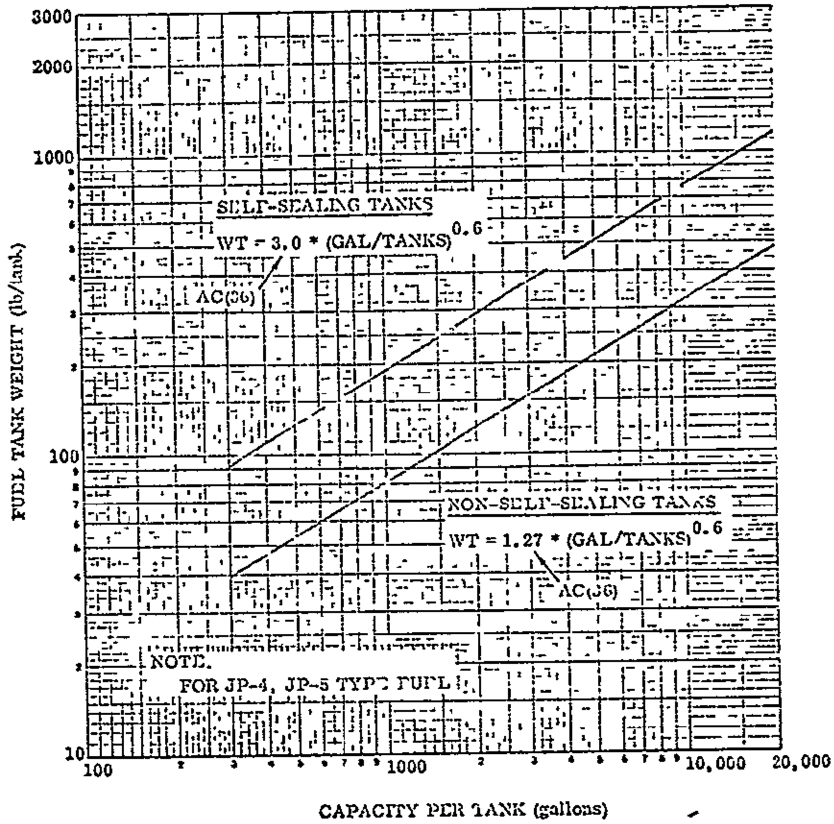


FIGURE 6.1-18. AIRCRAFT ROCKET ENGINE WEIGHT



ORIGINAL PAGE IS OF POOR QUALITY

FIGURE 6.1-19. AIRCRAFT FUEL TANK WEIGHT

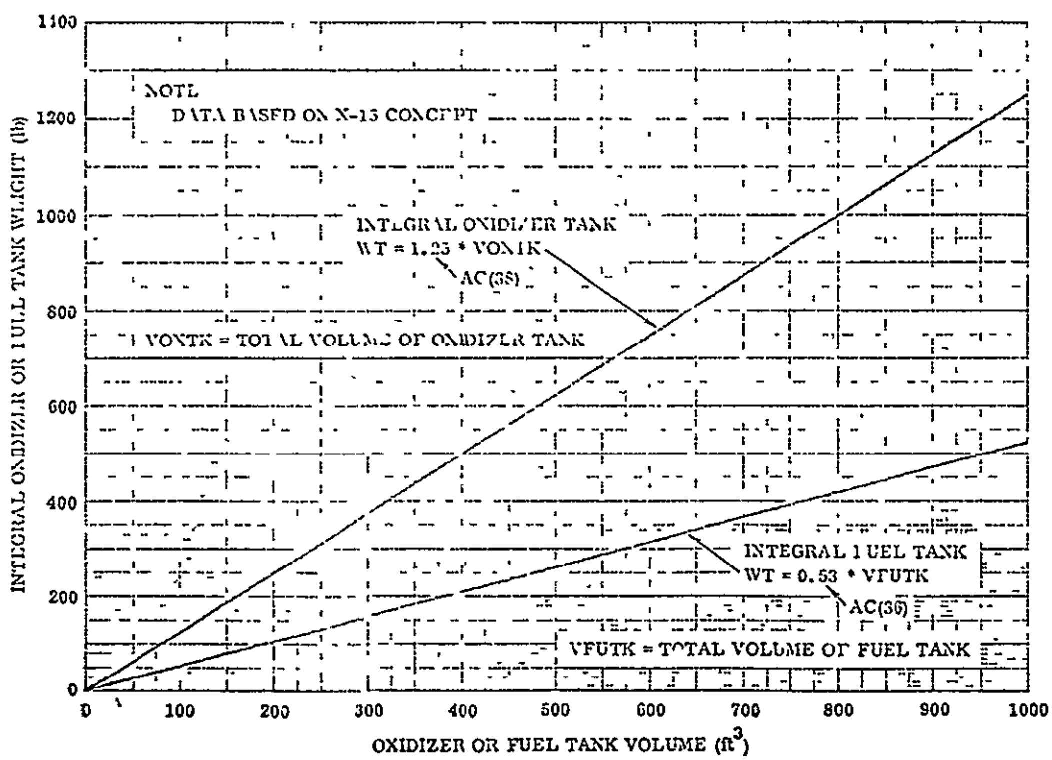


FIGURE 6.1-20. X15 CONCEPT TANK WEIGHTS

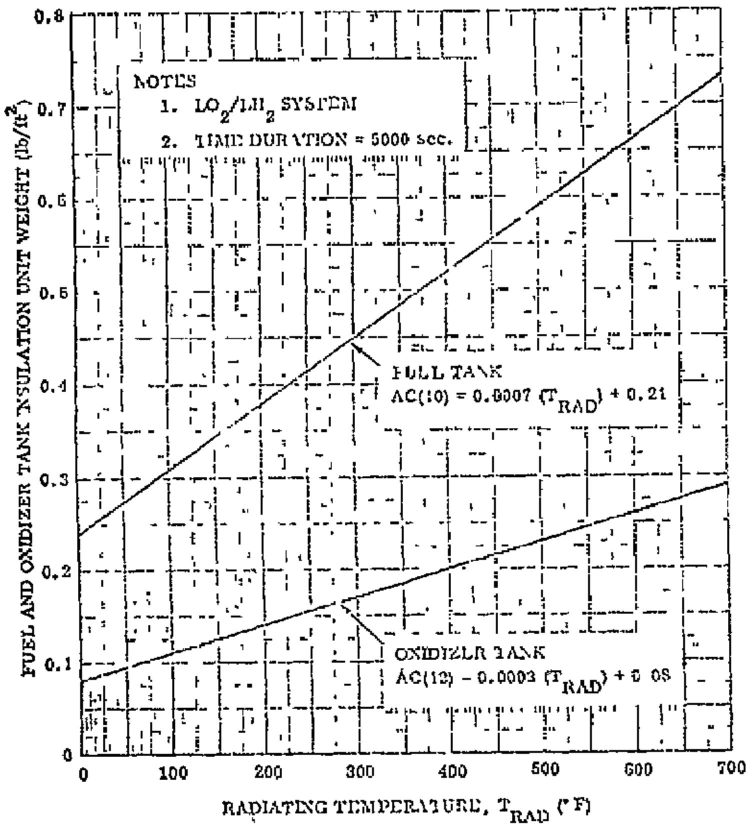


FIGURE 6.1-21. AIRCRAFT AND BOOSTER TANK INSULATION WEIGHTS

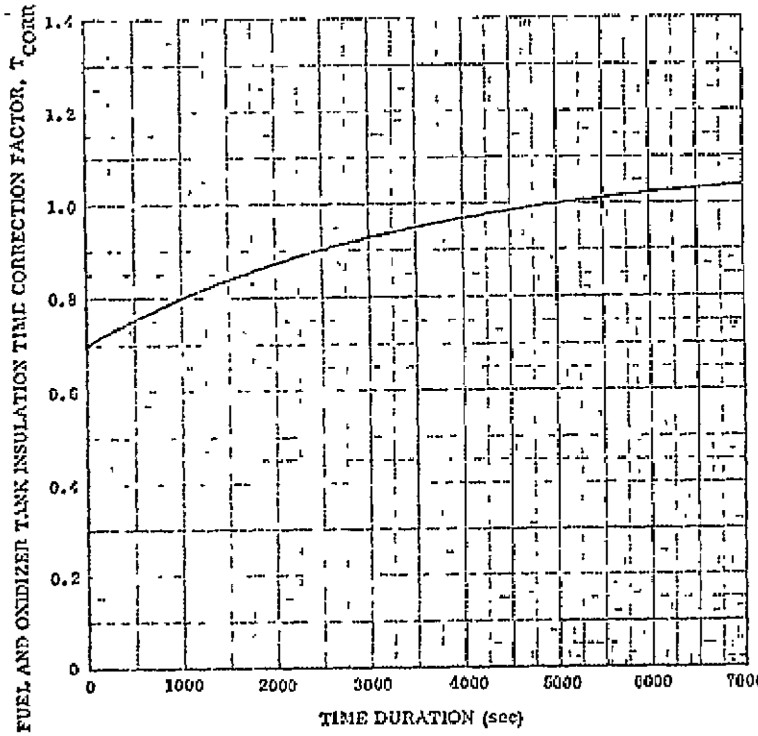


FIGURE 6.1-22. AIRCRAFT AND BOOSTER TANK INSULATION FLIGHT TIME CORRECTION FACTOR

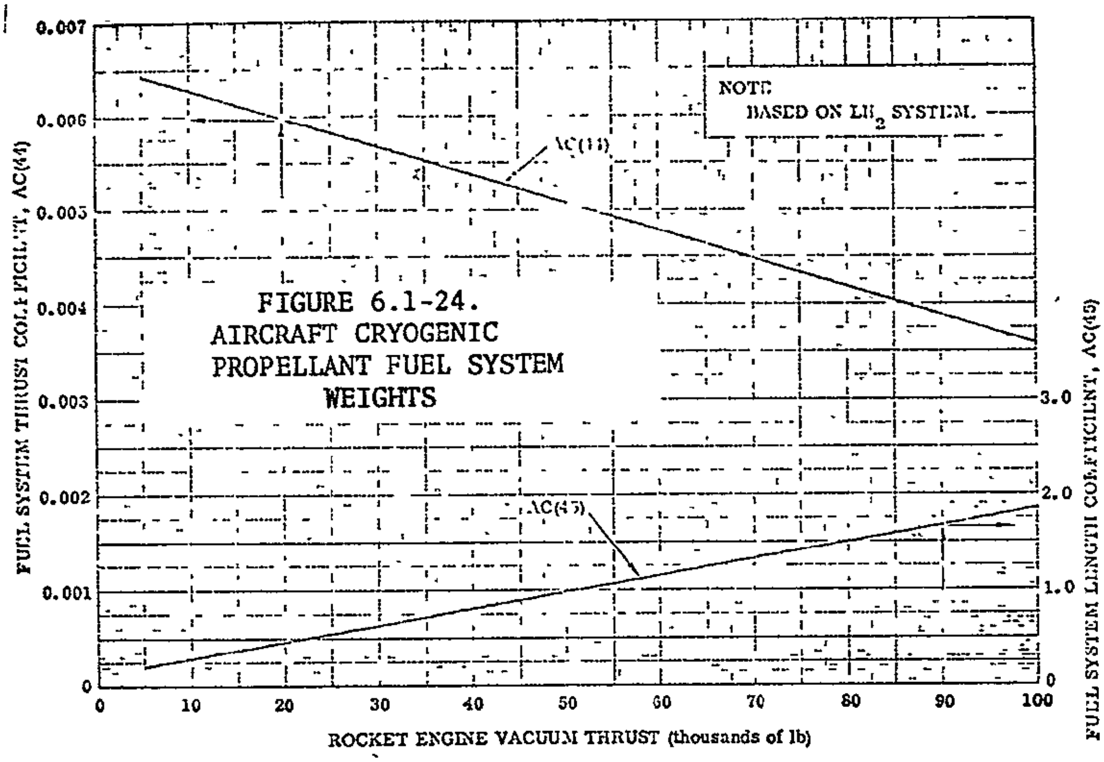


FIGURE 6.1-24.
AIRCRAFT CRYOGENIC
PROPELLANT FUEL SYSTEM
WEIGHTS

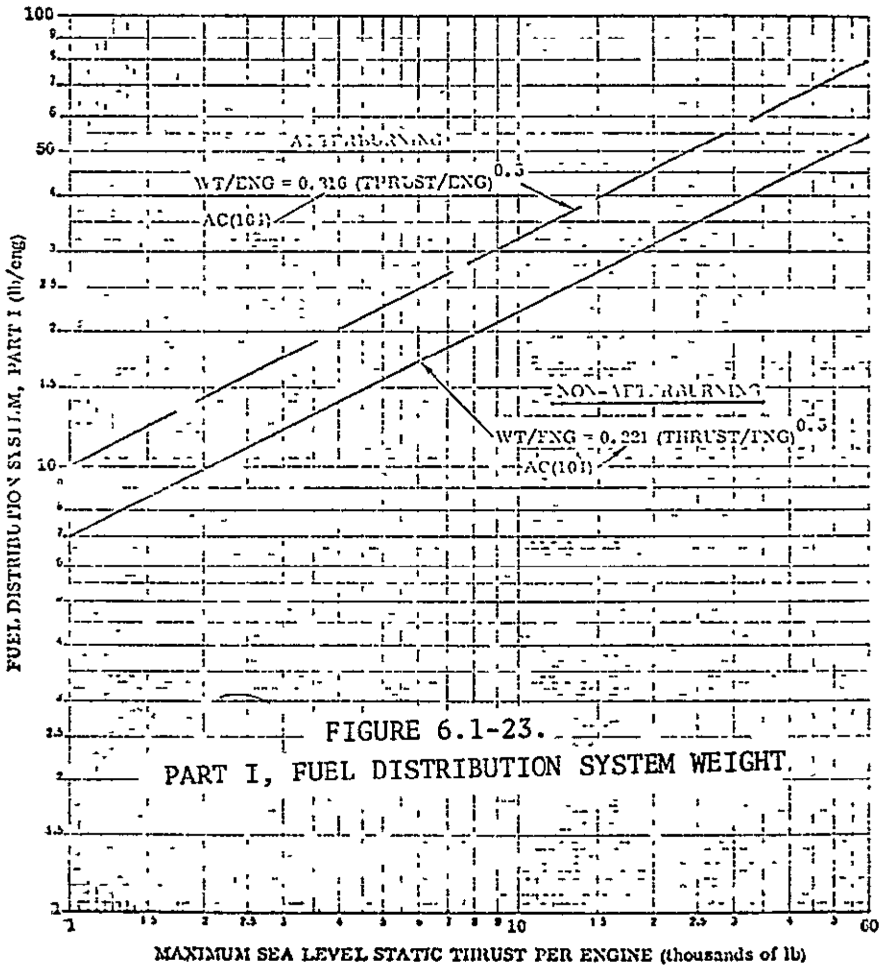


FIGURE 6.1-23.
PART I, FUEL DISTRIBUTION SYSTEM WEIGHT.

ORIGINAL PAGE IS
OF POOR QUALITY

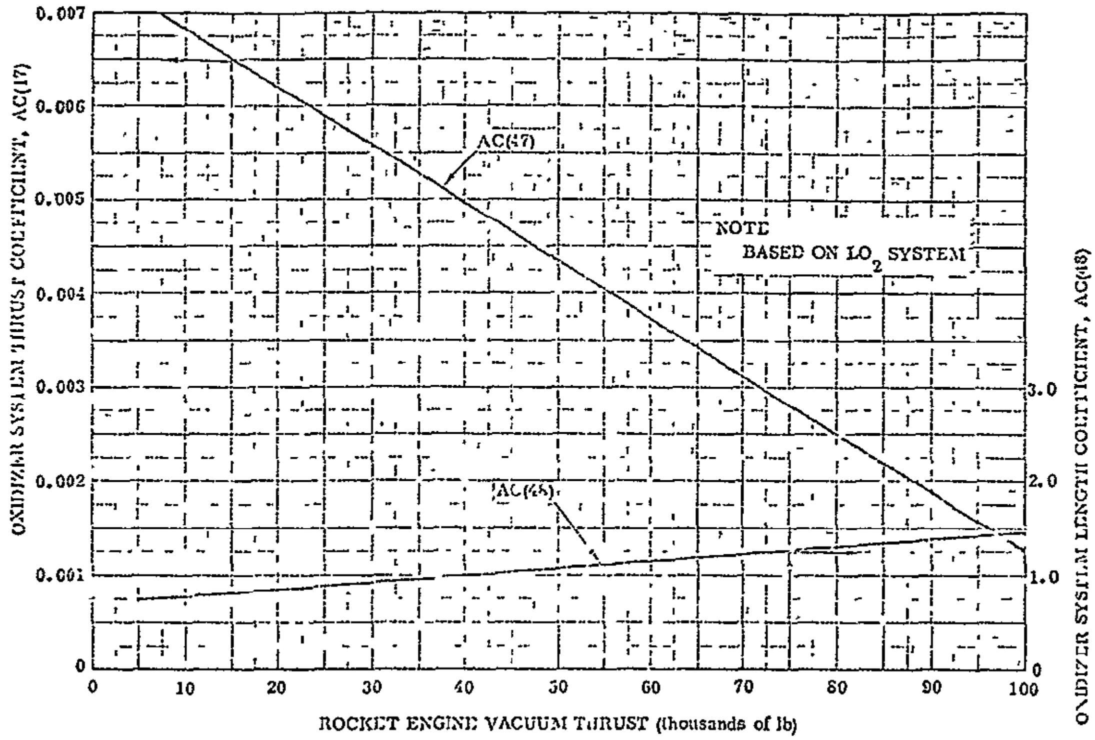


FIGURE 6.1-25. AIRCRAFT CRYOGENIC PROPELLANT OXIDIZER SYSTEM WEIGHTS

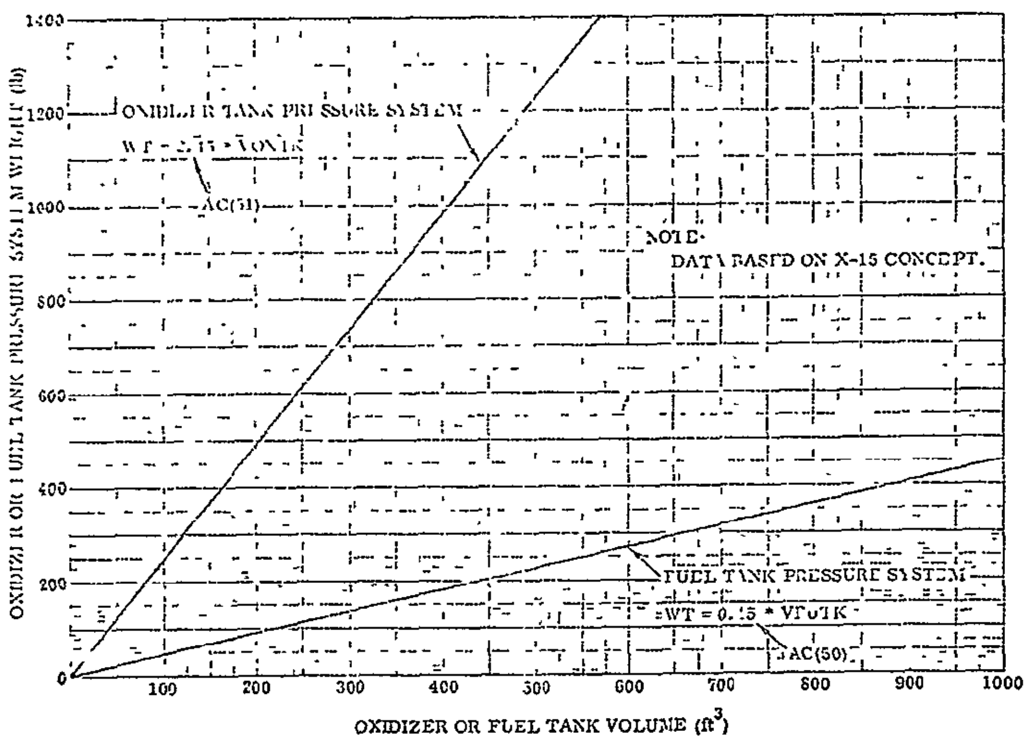


FIGURE 6.1-26. AIRCRAFT CRYOGENIC PROPELLANT PRESSURIZATION SYSTEM WEIGHTS

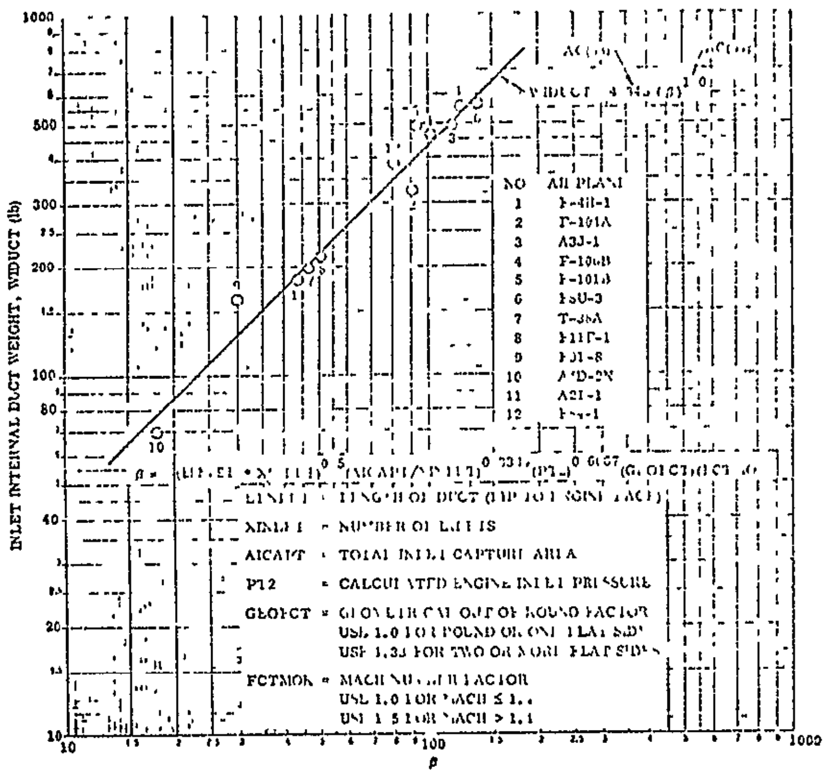


FIGURE 6.1-27. AIRCRAFT INTERNAL DUCT WEIGHT

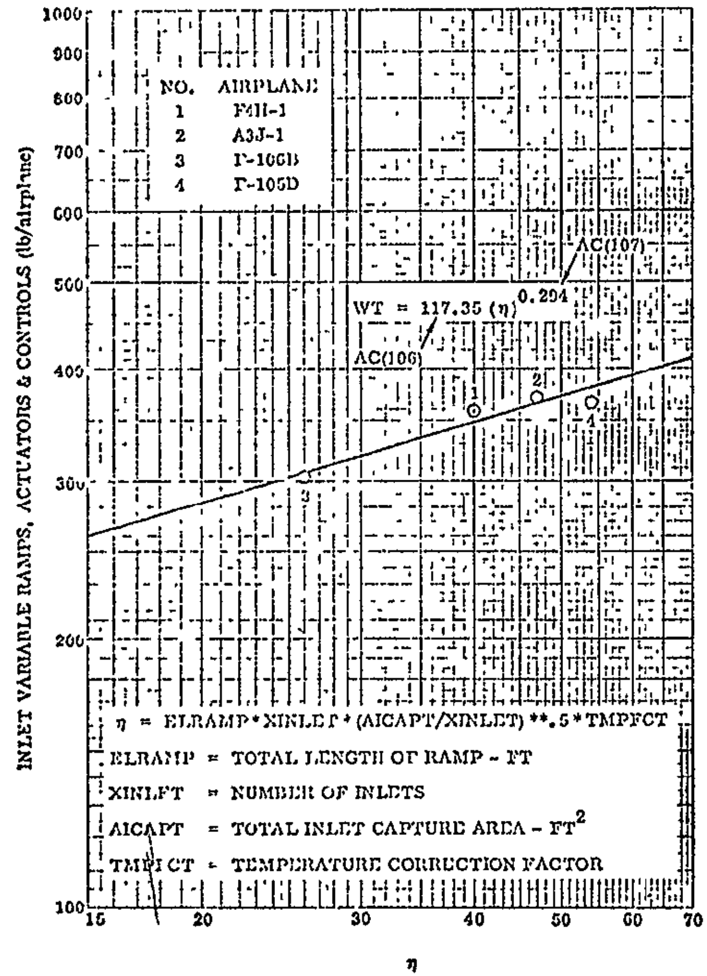


FIGURE 6.1-28. AIRCRAFT INLET VARIABLE RAMPS, ACTUATORS, AND CONTROL WEIGHTS

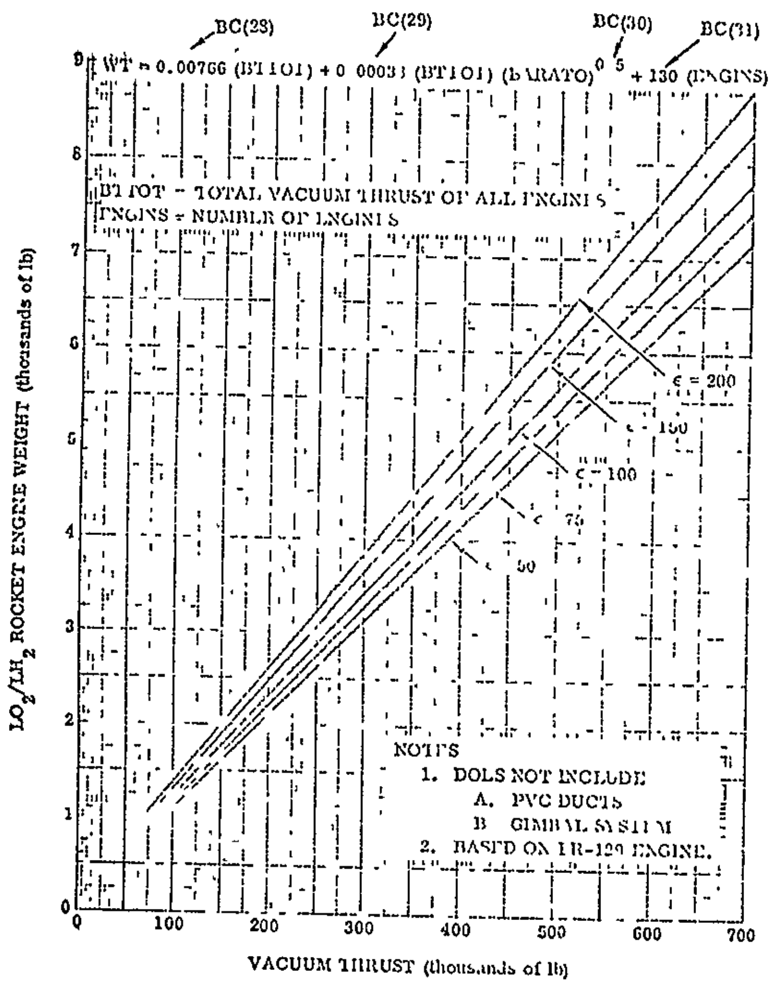


FIGURE 6.1-29. BOOSTER MAIN PROPULSION WEIGHTS

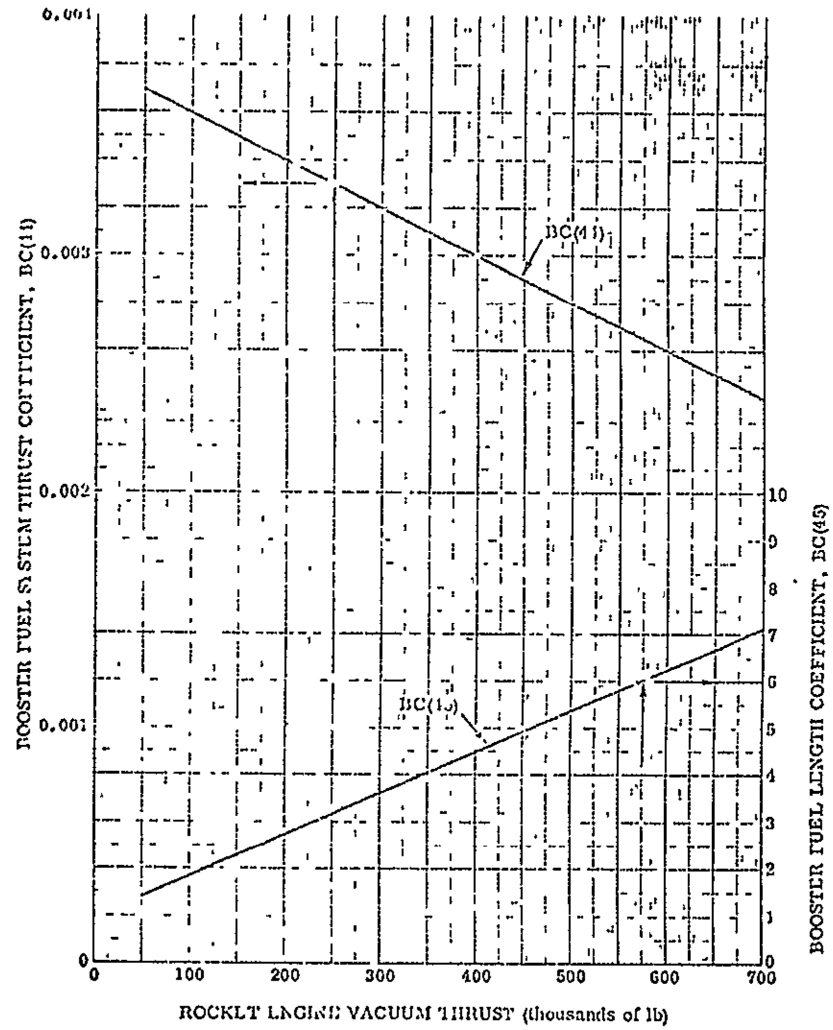


FIGURE 6.1-30. BOOSTER CRYOGENIC PROPELLANT FUEL SYSTEM WEIGHTS

ORIGINAL PAGE IS OF POOR QUALITY

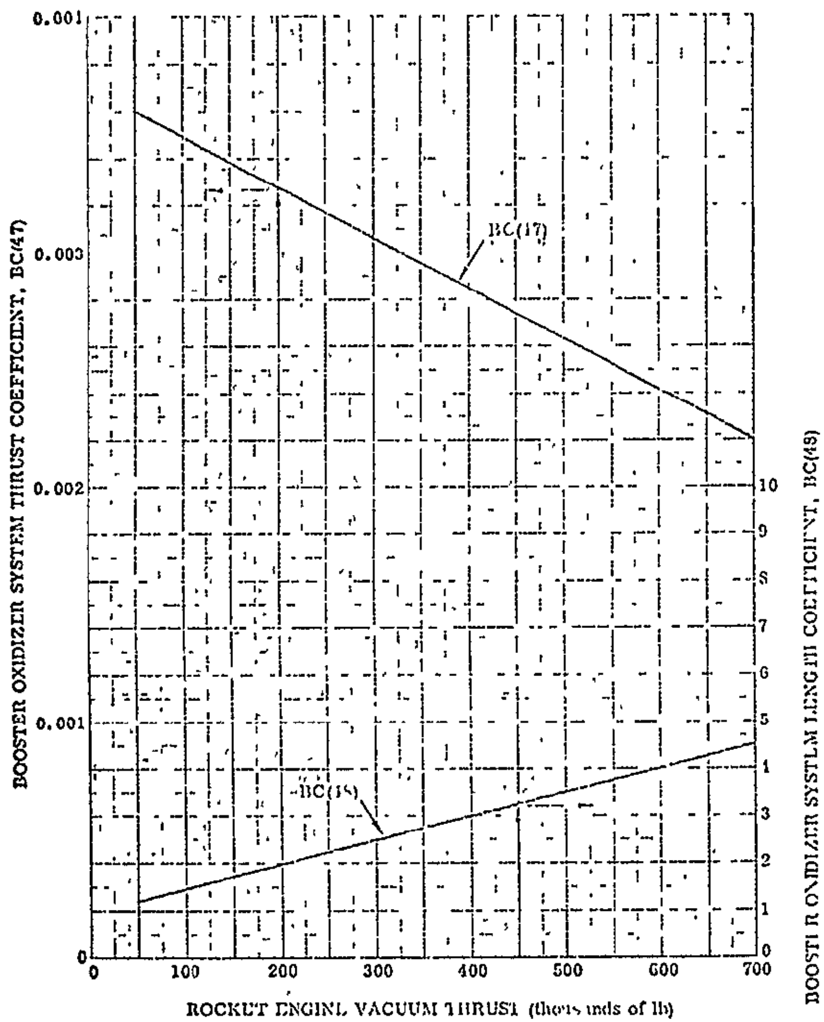


FIGURE 6.1-31. BOOSTER CRYOGENIC PROPELLANT OXIDIZER SYSTEM WEIGHTS

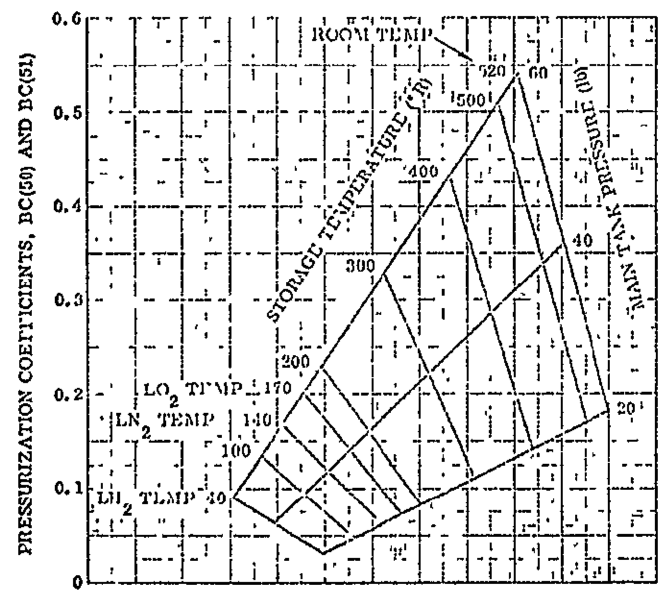


FIGURE 6.1-32. BOOSTER CRYOGENIC PROPELLANT PRESSURIZATION SYSTEM WEIGHTS

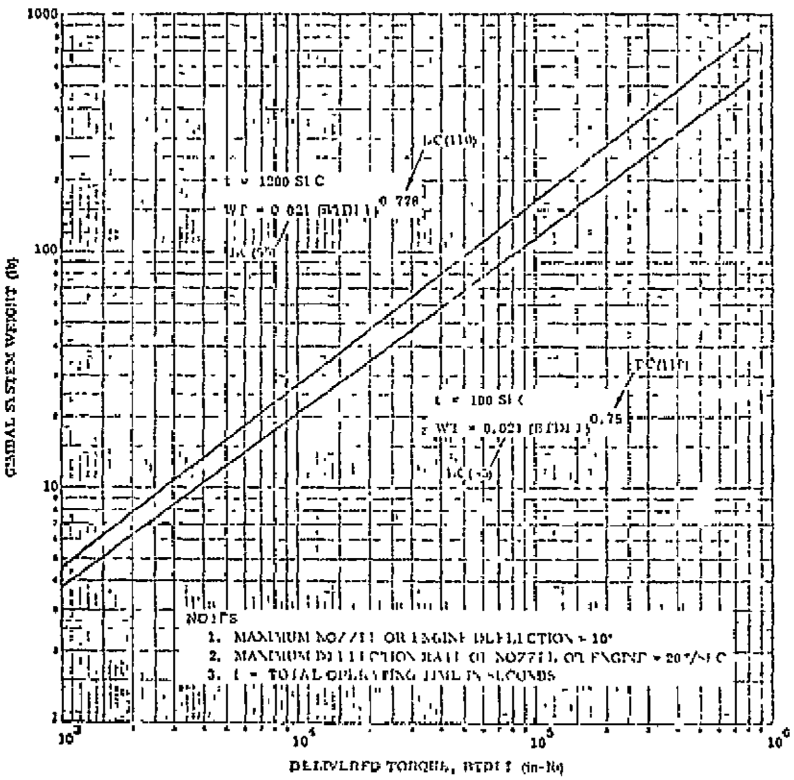


FIGURE 6.1-33. AIRCRAFT AND BOOSTER GIMBAL SYSTEM WEIGHTS (Twenty Degrees Per Second)

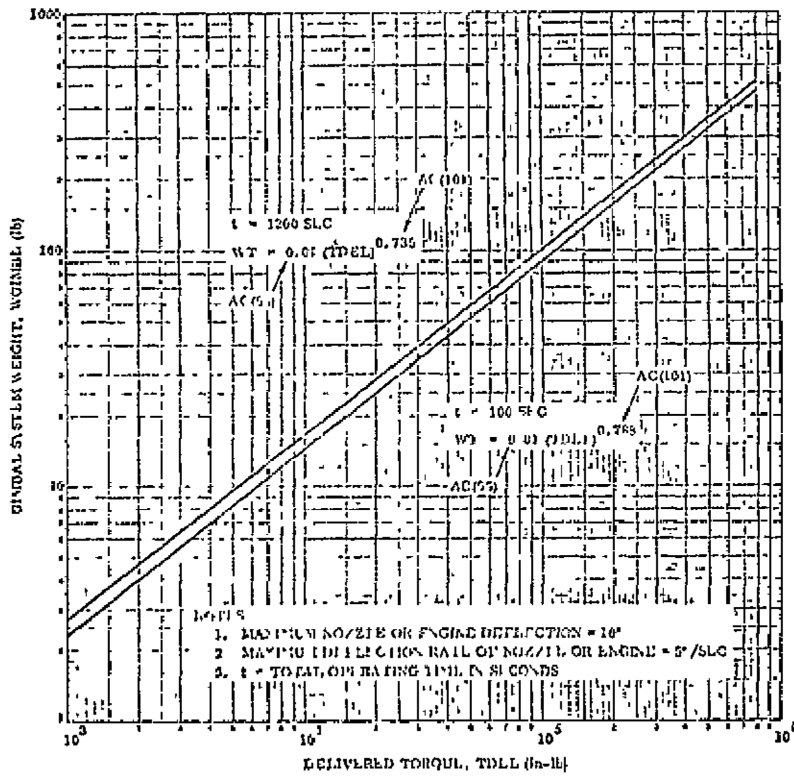


FIGURE 6.1-34. AIRCRAFT GIMBAT SYSTEM WEIGHT (Five Degrees Per Second)

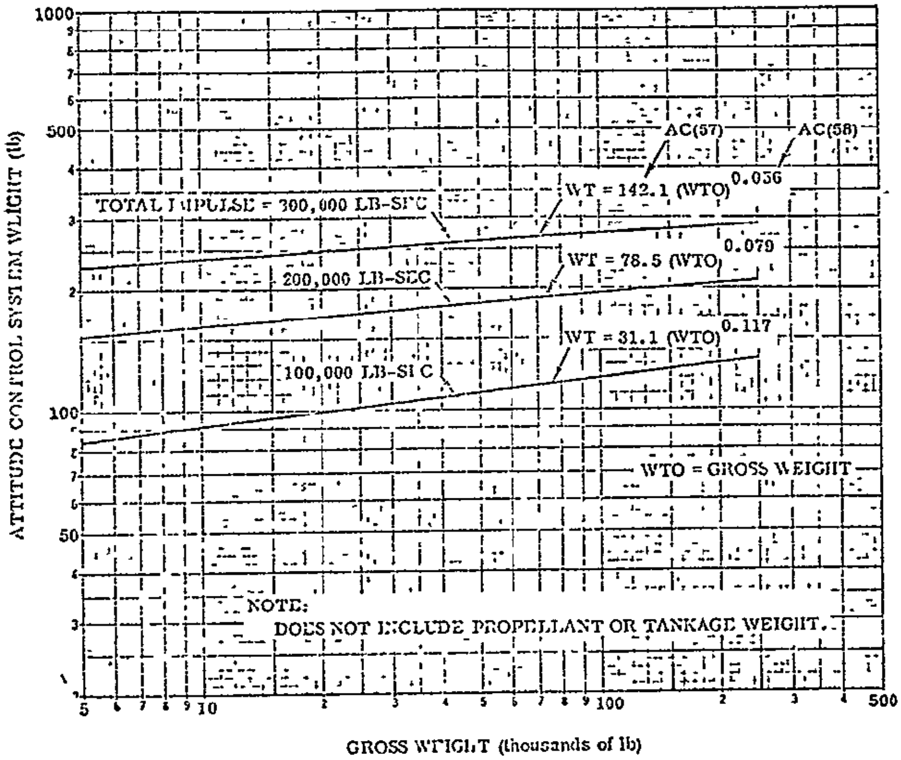


FIGURE 6.1-35. AIRCRAFT ATTITUDE CONTROL SYSTEM WEIGHT

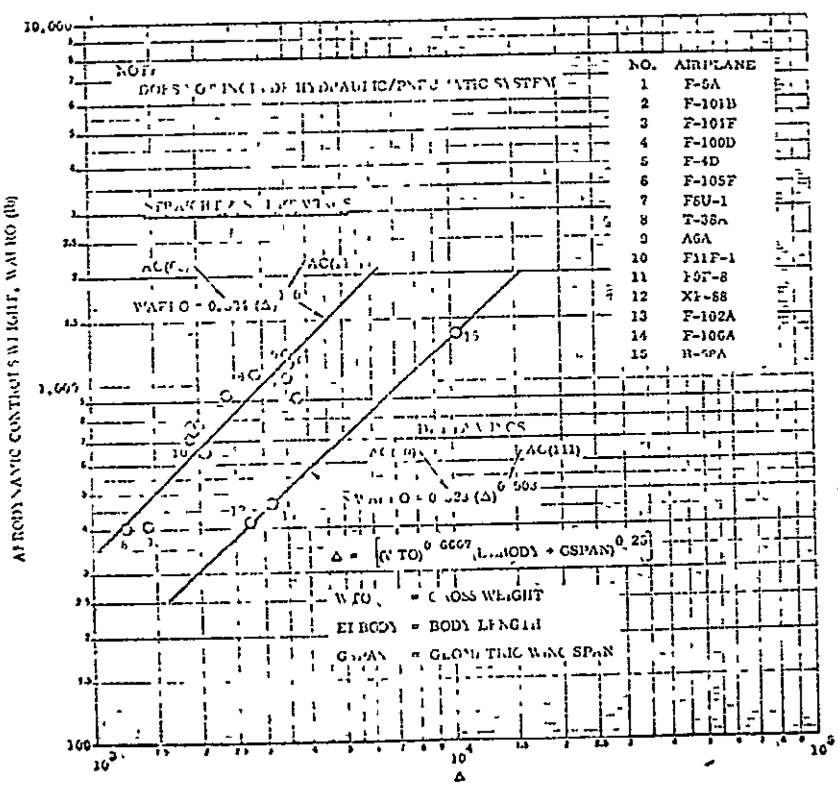


FIGURE 6.1-36. AIRCRAFT AERODYNAMIC CONTROL SYSTEM WEIGHT

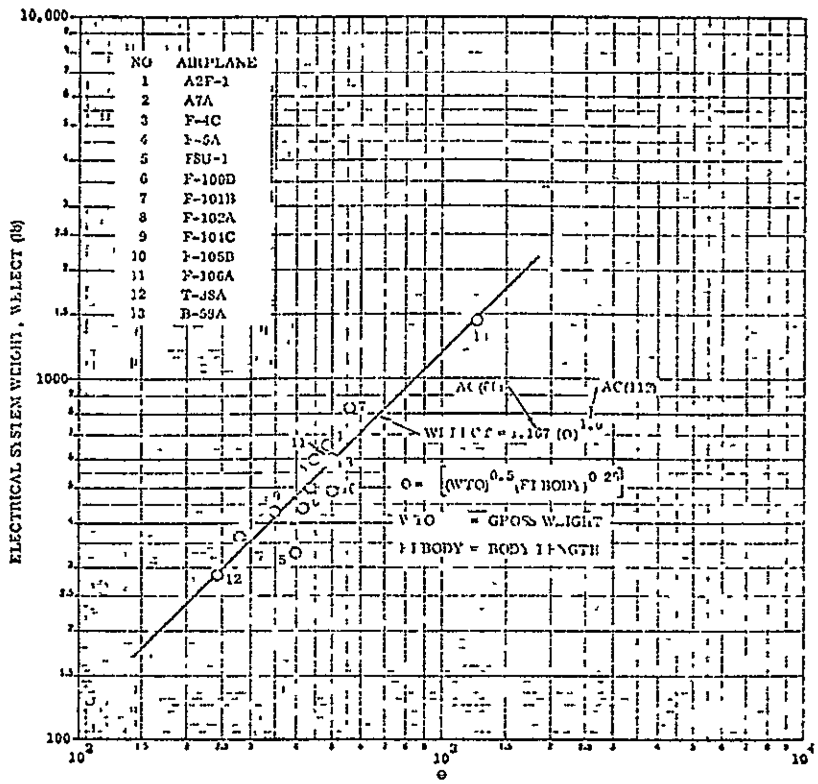


FIGURE 6.1-37. AIRCRAFT ELECTRICAL SYSTEM WEIGHT

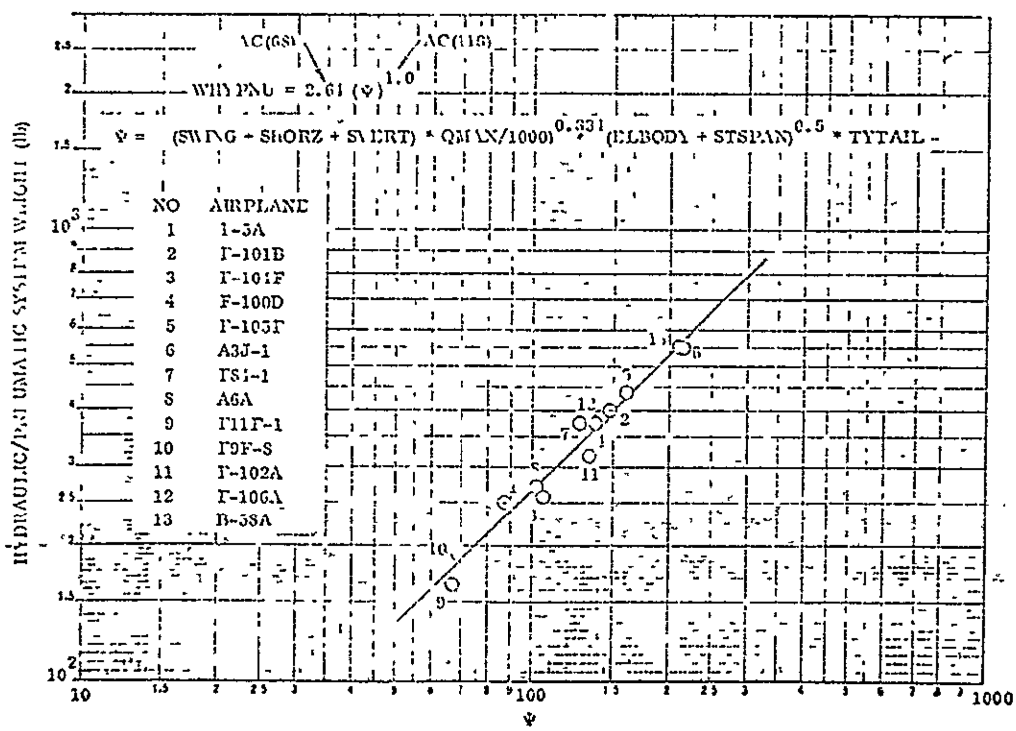


 FIGURE 6.1-38. AIRCRAFT HYDRAULIC/PNEUMATIC SYSTEM WEIGHT

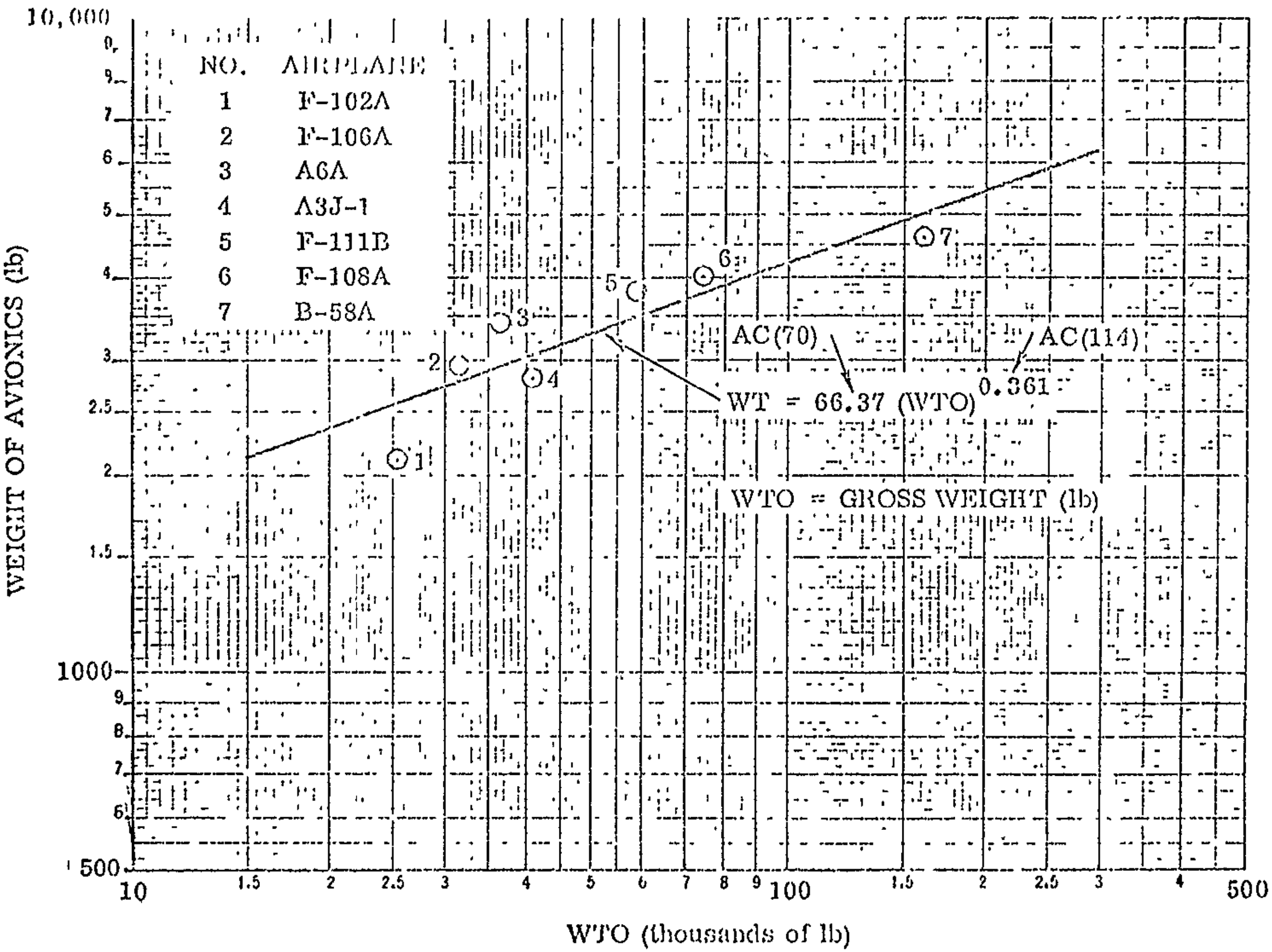


FIGURE 6.1-39. AIRCRAFT AVIONICS WEIGHT

TABLE OF CONTENTS FOR SECTION 6.2; PROGRAM VAMP

6.2 PROGRAM VAMP: A DETAILED VOLUME, AREA, AND MASS PROPERTIES CODE

VAMP is a general purpose program which estimates the volume, area, and mass properties of closed shell surfaces. It has the ability to incorporate any number of point inertia sources into the properties of the shell. By repetitive analysis, the combined properties of any number of shell surfaces and point inertias may be obtained.

Complex shells are represented by a distribution of quadrilateral panel elements over the shell surface. Each such quadrilateral element is decomposed into two triangular elements. This paneling technique differs from that employed in the programs of Section 3.1 and 4.1. These programs employ a mean planar representation of each surface panel.

In the VAMP code a mass per unit area or volume is associated with each quadrilateral element. Thus, the mass properties of a complex shell structure such as a vehicle skin, tank, or duct surface can readily be computed. Volumetric properties of solid bodies can also be obtained by means of Gauss's divergence theorem applied over the solid's surface. This theorem relates an integral through a volume to an integral over a surface. The surface employed is the closed system of quadrilateral panels distributed on the solid's surface. If the volume of an open shell is required, a suitable closing surface must be defined for the volume calculation.

Program VAMP computes the following properties of a closed shell surface:

1. volume enclosed
2. center of volume
3. volume of skin (from a finite thickness specified over each quadrilateral element)
4. skin surface area
5. planform area
6. total skin weight (from a surface density specified over each quadrilateral element)
7. Skin center of gravity locations
8. skin inertias

Program VAMP contains interfaces to the geometry program LRCACP described in Section 3.3 of the present report. This interface provides a general plot visualization capability as discussed in Section 3.3.

An outline of program VAMP is provided below This discussion is based on that provided by Norton in Reference 1. Detailed information regarding this program is contained in the original NASA Langley Research Center documentation of Reference 1.

6.2.1 Introduction

Program VAMP computes the mass properties, c.g. location, enclosed volume, wetted area, and planform area of aerospace vehicles. It is applicable to any closed structure that has a plane of symmetry, e.g., fuselage, stiffened fuel tank, etc. The program may be applied to non-symmetric structures in the ODIN/RLV simulation by appropriate data base manipulations, Section 2.4.5.2. The vehicle is described to the program by ordered sets of X,Y,Z coordinates of points on its surface. These data are input in the form of cross sections normal to the longitudinal axis. Y = 0 is the plane of symmetry.

The surface is approximated by quadrilaterals generated between the ordered set of points. Since the four corners of a quadrilateral are not necessarily coplanar, each quadrilateral is analyzed as two triangles. The mass properties of each quadrilateral may be computed from

a. thickness and density input for each quadrilateral
b. weight per unit area input at each point
c. combination of both

The weight per unit area can be a composite of the shell structural will including skin, insulation, ribs, stringers, standoffs, brackets, etc. Computed mass properties contain all contributions from the distributed mass in the vehicle surface wall. Additional point mass sources may be added by specifying each one's center of gravity (c.g.) location and the mass properties about this c.g. These point masses may lie inside or outside the surface and do not have to be symmetrical with respect to Y = 0.

Program VAMP provides a means for combining detailed shape and mass data to produce the overall mass vehicle properties. The mass properties of well defined subsections are also produced; hence, the vehicle mass distribution can be obtained. The program is applicable to any closed structure that has a plane of symmetry; e.g., aircraft fuselage, stiffened fuel tank, etc. The program is only aware of the existance of the quadrilateral representation of the structure's surface and the mass in or near this surface. Mass contributions from bulkheads, floors, cargo, tanks, fuel avionics, engines, etc. which lie within the surface may be added as point mass sources. The program may be used to compute the properties of such point sources in a separate calculation. The component mass, area, and volume properties may be merged through the ODIN/RLV data base. It should be noted that certain mass properties of a fuel tank may be computed by VAMP. The results can be used as a point source input in a fuselage or wing analysis. Fuel tank output will contain the center of volume. This is also the fuel's center of gravity. The center of gravity travel of the fuel tank as the tank empties may be obtained by an appropriate sequence of analyses.

6.2.2 The Surface Model

The surfaces to be considered here can be of fairly arbitrary shape, for example, a noncircular aircraft fuselage including fins, and wings, Figure 6.2-1. The figure also illustrates the basic coordinate reference system.

employed in VAMP. The X axis is the body longitudinal axis, positive aft. Positive Z is up and positive Y is starboard. An X = constant plane through the body is a station plane and the curve formed by the intersection of the surface and a station plane is a station contour. The body is divided into segments. Each segment is bounded by two station planes. Additional station planes may lie within a segment. The surface points associated with a given segment form a separate group of points.

Successive station contours are specified for increasing values of X. Each station contour is specified on the right hand half plane ($Y \geq 0.0$) from the bottom of the contour to the top of the contour, Figure 6.2-2. The number of points on all station contours within a given segment must be the same. Analytically, the station contour on the i^{th} segment is given by

$$X_k = X^i$$

$$Y_k = Y^i{}_k$$

$$Z_k = Z^i{}_k \qquad k = 1, 2, \ldots, N_i$$

where N_i is the number of points on each contour in the i^{th} segment. This is illustrated in Figure 6.2-2.

A portion of the quadrilateral grid formed from the input points is illustrated in Figure 6.2-3. The quadrilaterals are the surface elements used in the analysis. It has been previously noted that the four corners of these quadrilaterals are not necessarily coplanar and that each is analyzed as two triangular surface elements. The diagonals in Figure 6.2-3 show the triangles. The points P(X,Y,Z) define the shape and location of all triangles. The mass of each triangular surface element is based on

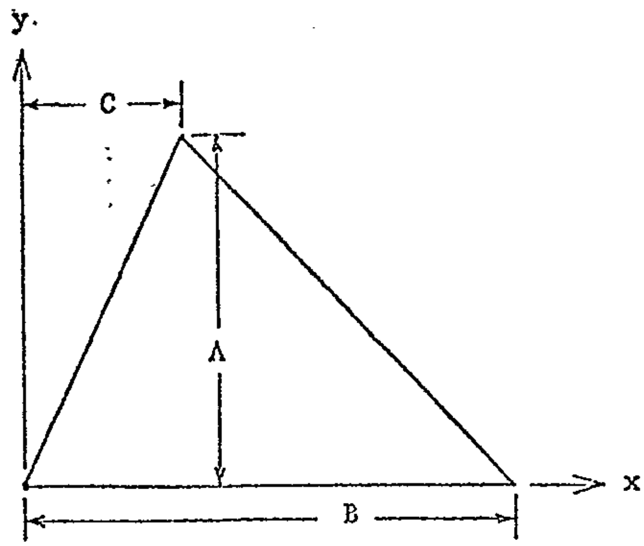
1. thickness and density which is specified for each quadrilateral
2. weight per unit area specified at each point in the mesh
3. from a combination of both

The thickness and density are assumed *constant over any one quadrilateral*. A different value may be input for each quadrilateral. The contribution from the *weight per unit area* is assumed constant over any one triangle. A mean weight per unit area is obtained for each triangle by linear interpolation between the inputs at the three points which define the triangle.

6.2.3 The Elemental Triangle

An elemental triangle is illustrated below. The origin of the coordinate system is the lower left corner. If the triangle is of uniform thickness and density and z = 0 is the triangle's midplane, then the following products of inertia and moment of area about z are zero.

$$I_{yz} = I_{xz} = S_z = 0.0 \qquad (6.2.1)$$



For a thin triangle of thickness t

$$I_{xx} = \iiint \rho \ (y^2+z^2)dx\ dy\ dz \approx \rho t \iint y^2\ dx\ dy \qquad (6.2.2)$$

$$I_{yy} = \rho t \iint x^2\ dx\ dy \qquad (6.2.3)$$

$$I_{zz} \quad I_{xx} + I_{yy} \qquad (6.2.4)$$

It can be shown that the following relationships hold

Area, $\quad S = \int_0^A \int_{x_L}^{x_R} dx\ dy = AB/2 \qquad (6.2.5)$

x moment, $\quad S_x = \iint x\ dx\ dy = S(B+C)/3 \qquad (6.2.6)$

y moment, $\quad S_y = \iint y\ dx\ dy = SA/3 \qquad (6.2.7)$

z moment of inertia, $\quad I_{yy} = \iint x^2 dx\ dy = S(B^2+BC+C^2)/6 \qquad (6.2.8)$

x moment of inertia, $\quad I_{xx} = \iint y^2\ dx\ dy = SA^2/6 \qquad (6.2.9)$

xy product of inertia, $\quad I_{xy} = \iint xy\ dx\ dy = SA(B+2C)/12 \qquad (6.2.10)$

x coordinate of centroid, $\quad \bar{x} = S_x/S = (B+C)/3 \qquad (6.2.11)$

y coordinate of centroid, $\quad \bar{y} = S_y/S = A/3 \qquad (6.2.12)$

Let the c.g. be located at $(\bar{x},\bar{y})$. The inertias and products of inertia about the $(\bar{x},\bar{y})$ are

$$\bar{I}_{xx} = SA^2/18 \tag{6.2.13}$$

$$\bar{I}_{yy} = S(B^2-BC+C^2)/18 \tag{6.2.14}$$

$$\bar{I}_{xy} = SA(2C-B)/36 \tag{6.2.15}$$

$$\bar{I}_{zz} = \bar{I}_{xx} + \bar{I}_{yy} \tag{6.2.16}$$

These properties of area are multiplied by the mass per unit area to obtain mass properties. The mass or weight per unit area, t, is specified for each quadrilateral. Each quadrilateral is analyzed as two triangles, Section 6.2.2. An additional weight per unit area, W may be defined at every point on the surface. These points establish the quadrilateral and triangle geometries. On any elemental triangle, the mean additional W from the three triangle vertices is employed.

6.2.4 Center of Volume

The closed surface center of volume is computed by Gauss's divergence theorem.

$$\iiint_V \vec{\nabla} \cdot \vec{F} dV = \iint_S \vec{F} \cdot \vec{n} dS \tag{6.2.17}$$

where

$\vec{\nabla}$ - the grad operator $\frac{\partial}{\partial x} \cdot \vec{i} + \frac{\partial}{\partial y} \cdot \vec{j} + \frac{\partial}{\partial z} \cdot \vec{k}$, Reference 2.

$\vec{F}$ - a vector function of position

$\vec{n}$ - the unit outward normal to the surface S

The divergence theorem can also be expressed in terms of a scalar function of position, ϕ, where

$$\vec{F} = \nabla\phi \tag{6.2.18}$$

Substituting Equation (6.2.18) into (6.2.17)

$$\iiint_V \nabla^2 \cdot \phi dV = \iint_S \vec{\nabla}\phi \cdot \vec{n} d S \tag{6.2.19}$$

Selecting

$$\phi = x^2/2 \tag{6.2.20}$$

results in

$$\iiint_V dV = \iint_S x\vec{i} \cdot \vec{n} dS \tag{6.2.21}$$

ORIGINAL PAGE IS OF POOR QUALITY

or

$$V = \iint_S ax \, dS \tag{6.2.22}$$

where

$$\vec{n} = a\vec{i} + b\vec{j} + c\vec{k} \tag{6.2.23}$$

Equation (6.2.22) defines the enclosed volume. The volume centroid may be found by substituting the generating function

$$\phi = \frac{x^3}{6} \tag{6.2.24}$$

into divergenge theorem, Equation (6.2.19). This gives

$$\iiint_V x dV = \iint_S (x^2/2)\vec{i} \cdot \vec{n} dS = \iint_S ax^2/2 \cdot dS \tag{6.2.25}$$

By definition

$$\bar{x} \cdot V = \iiint_V x dV \tag{6.2.26}$$

where $\bar{x}$ is the volume centroid along the x axis. From equations (6.2.25) and (6.2.26)

$$\bar{x} = \frac{1}{2V} \iint_S ax^2 dS \tag{6.2.27}$$

Equations (6.2.22) and (6.2.27) are integrated over each elemental surface triangle and summed over the surface to obtain the volume and center of volume, respectively.

REFERENCES:

1. Norton, Patrick J., Volume, Area, and Mass Properties and Configuration Plots of Aerospace Vehicles (Program VAMP), to be published as a NASA Contractor Report.

2. Hague, B.: An Introduction to Vector Analysis, Methuen's Monographs on Physical Subjects, John Wiley, and Sons, Inc. 1961.

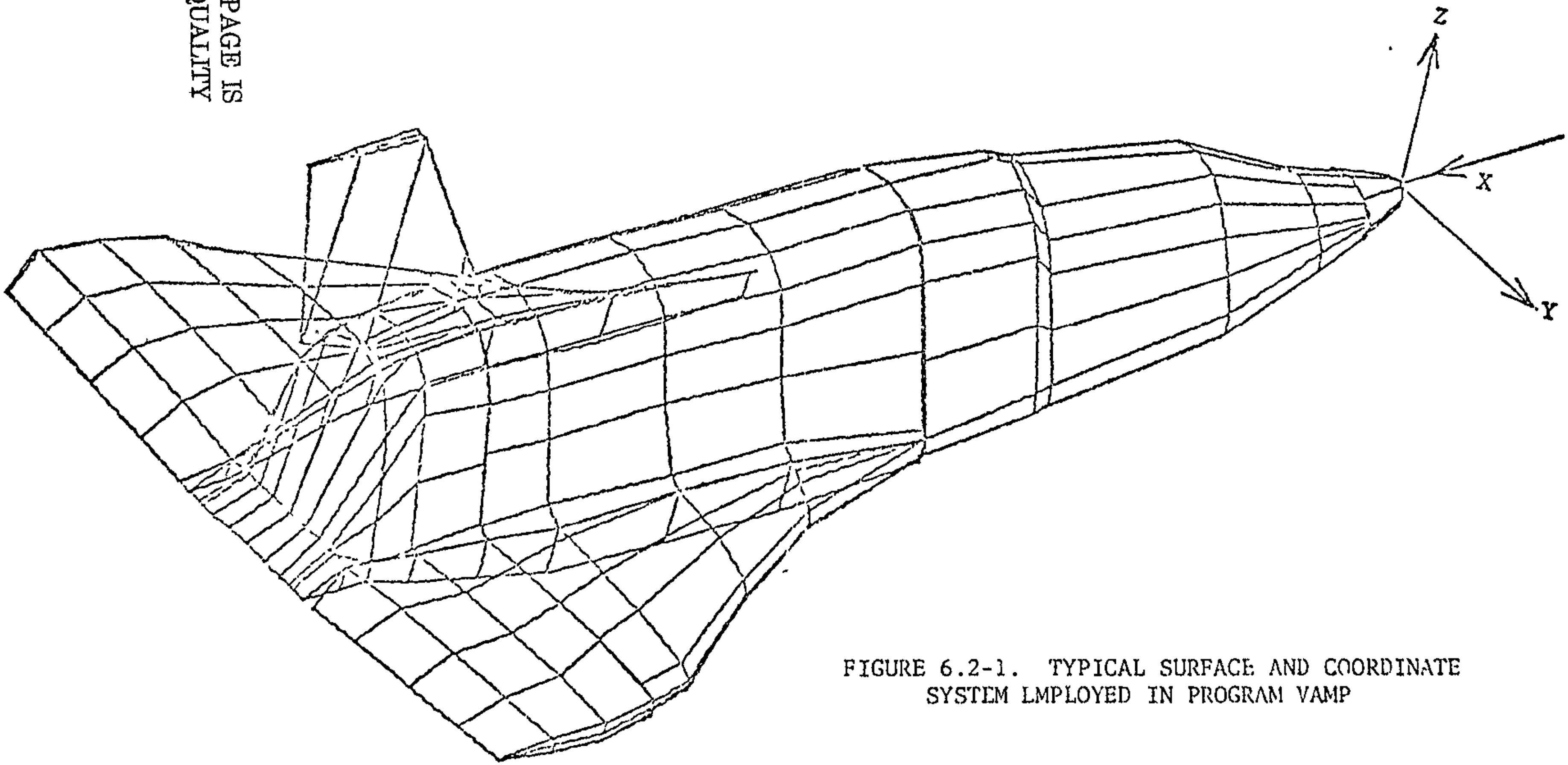


FIGURE 6.2-1. TYPICAL SURFACE AND COORDINATE SYSTEM EMPLOYED IN PROGRAM VAMP

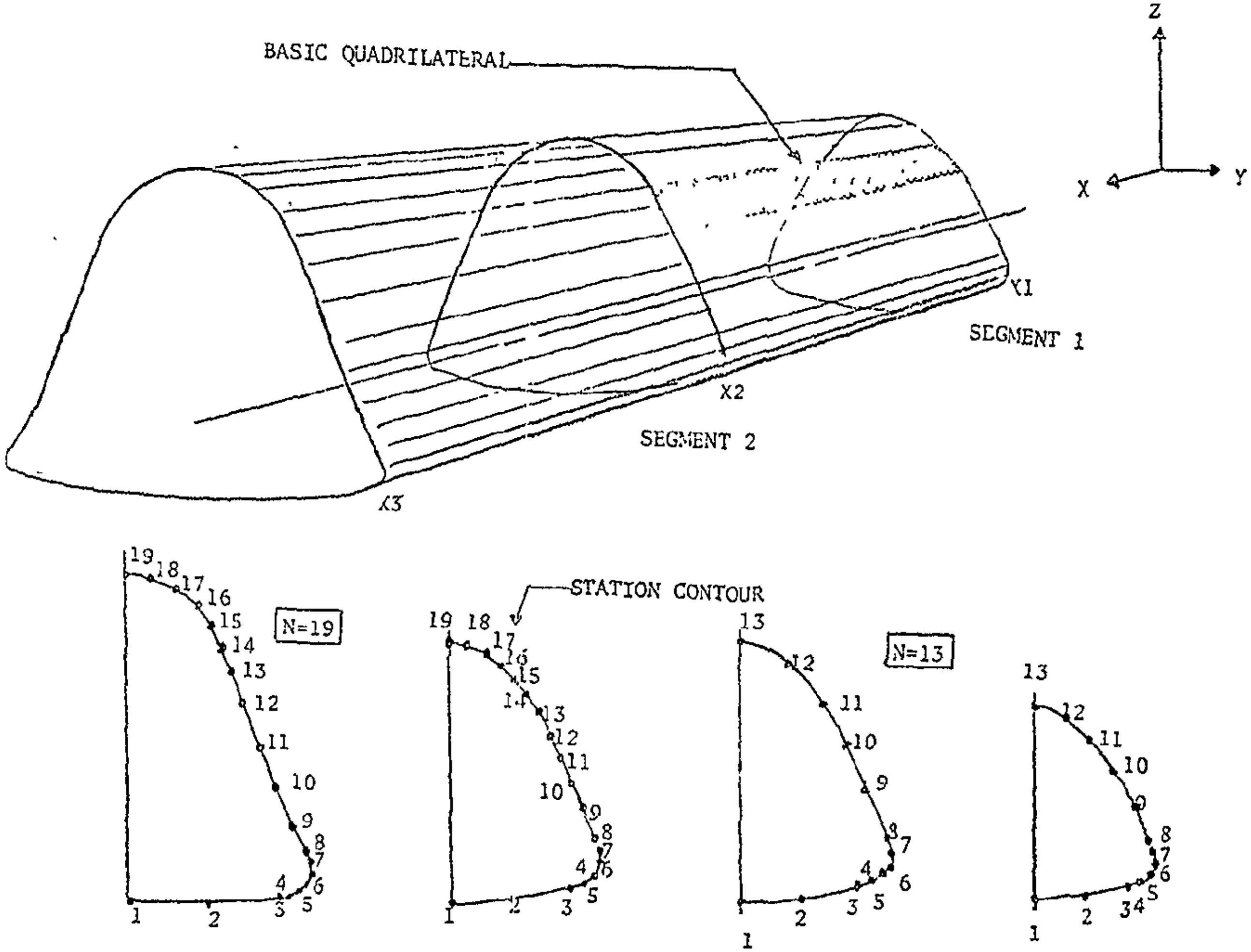


FIGURE 6.2-2. STATION CONTOUR DEFINITION

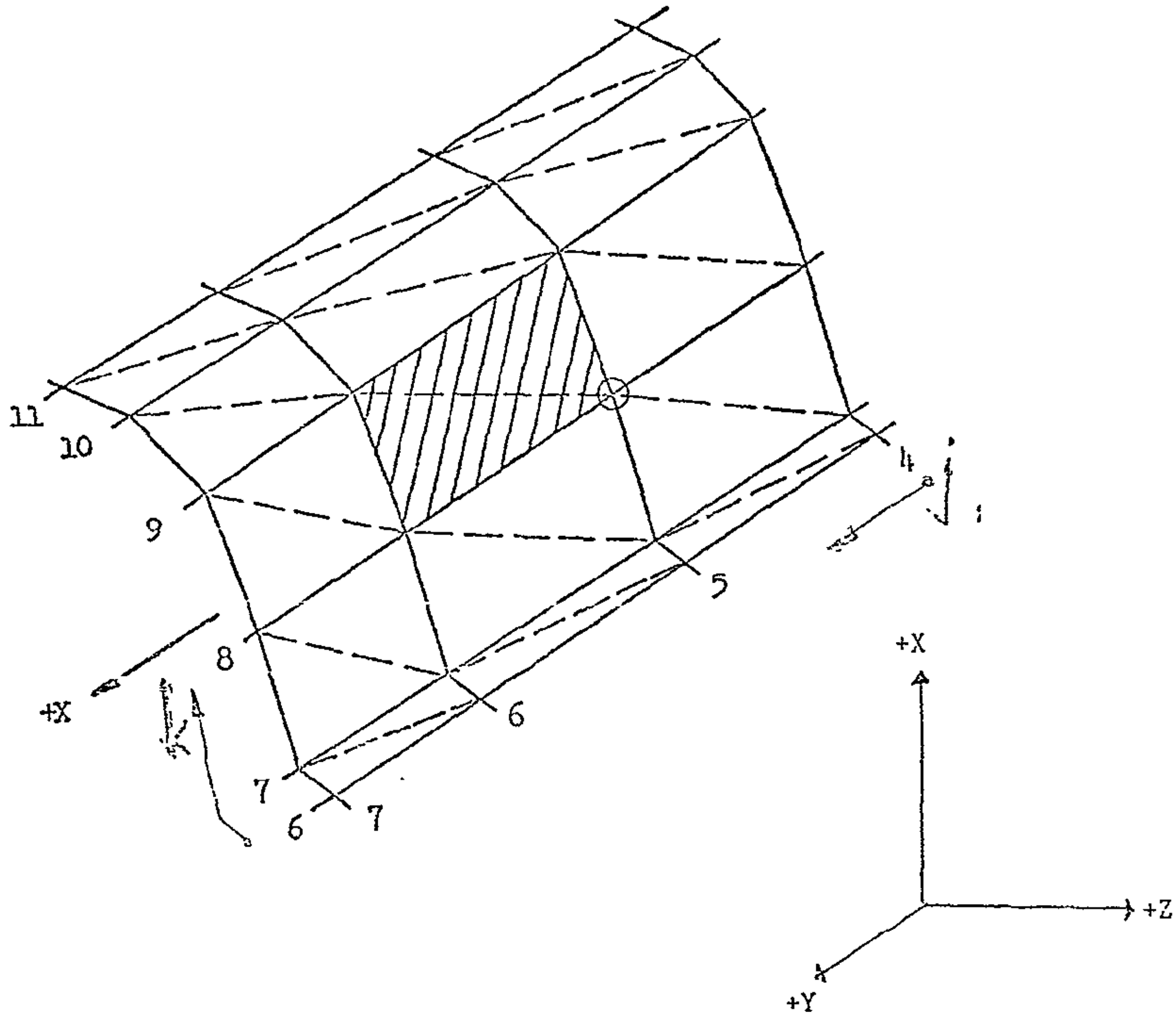


FIGURE 6.2-3. TRIANGULAR SURFACE ELEMENTS EMPLOYED IN VOLUME, AREA, AND MASS PROPERTIES ANALYSIS

SECTION 7

PERFORMANCE

The ODIN/RLV program library contains four performance estimation programs. Program sources are previous Air Force Flight Dynamics Laboratory studies and in-house Air Force and NASA programming. Programs are provided for

1. Simplified take-off and landing analysis
2. Approximate segmented mission analysis
3. Three-degree-of-freedom flight path optimization
4. Combat performance optimization and analysis

Each program is outlined in the following sections. For complete details, reference should be made to the original source documentation. At the present time the simplified take-off and landing analysis code is an integral part of the approximate segmented mission analysis code.

TABLE OF CONTENTS FOR SECTION 7.1, PROGRAM TOLAND

7.1 PROGRAM TOLAND: A SIMPLIFIED TAKE-OFF AND LANDING ANALYSIS CODE

Program TOLAND was originally constructed by Mr. Louis J. Williams of NASA's Advanced Concepts and Missions Division, OART. The program provides

1. Simplified high lift aerodynamics based on Reference 1
2. A ground roll analysis
3. Rotation logic
4. Climb out to clear a 50 foot obstacle

TOLAND, as presently installed in the ODIN/RLV does not exist as an independent code; rather it is an option in the Section 7.~~2-NSE~~G II program.

7.1.1 Take-Off High Lift Aerodynamics

Program TOLAND uses a self-contained aerodynamics package based primarily on the Reference 1 DATCOM methods. Angle of attack in the ground run and rotation maneuvers is determined from the vehicle geometry. In the *ground roll*

$$\alpha_G = \alpha_{BG} + \alpha_{WB} \tag{7.1.1}$$

where

α_G = wing incidence in ground roll
α_{BG}= body incidence in ground roll
α_{WB}= wing incidence relative to body

In the rotated attitude

$$\alpha_R = \alpha_{B_{MAX}} - 1.0 + \alpha_{WB} \tag{7.1.2}$$

The additional symbols are

α_R = wing incidence following rotation
$\alpha_{B_{MAX}}$ = maximum body rotation, usually determined by the tail dragging condition

7.1.1.1 Take-Off Lift and Drag

7.1.1.1(a) Maximum Lift and Drag

The wing maximum lift coefficient is given by

$$C_{L_{MAX}} = (C_{L_{MAX}})_{BASE} + \Delta C_{L_{MAX}} + \Delta C_{L_{FLAP}} \tag{7.1.3}$$

with a corresponding angle of attack

$$\alpha' = (\alpha C_{L_{MAX}})_{BASE} \tag{7.1.4a}$$

During take-off the maximum angle of attack, α_{MAX}, is limited to

$$\alpha_{MAX} = 0.8 \cdot \alpha'_{MAX} \tag{7.1.4b}$$

In these two expressions

$C_{L_{MAX}}$ =wing lift coefficient at the first peak, Figure 7.1-1

$(C_{L_{MAX}})_{BASE}$ =basic wing maximum lift coefficient

$C_{L_{MAX}}$ =maximum lift coefficient increment due to taper and sweep

$C_{L_{FLAP}}$ =maximum lift coefficient increment from flap deflection

$(\alpha_{MAX})_{BASE}$ =basic wing angle of attack at maximum lift coefficient based on linear C_{L_α} term. Nonlinear α increment is ignored in TOLAND.

The high lift aerodynamic model is a simplified DATCOM method for subsonic low aspect ratio, untwisted, symmetric section wings. Due to the low speeds encountered in take-off and landing, the DATCOM method is modified by the approximation

$$\beta = \sqrt{1-M^2} = 1.0 \tag{7.1.5}$$

Clean wing contributions to Equations (7.1.3) and (7.1.4) are obtained from Figures 7.1-2 to 7.1-3. Figure 7.1-2 provides $(C_{L_{MAX}})_{BASE}$; Figure 7.1-3 gives $\Delta C_{L_{MAX}}$. The wing taper ratio correction factors C1 and C2 of Figures 7.1-2 and 7.1-3 are obtained from Figure 7.1-4. In Figure 7.1-2, program TOLAND is limited to the lowest curve, and the curve for $M \leqslant 0.2$ is used in Figure 7.1-3. Angle of attack at maximum lift coefficient is obtained from Figure 7.1-5. (The charts employed are Figures 4.1.3.4-16b to 4.1.3.4-18a of the Reference 1 DATCOM).

Flap maximum lift coefficient increment is based on the expression

$$\Delta C_{L_{FLAP}} = 10.95(C_{L_A})_{BASE}\left[1.5\left(\frac{B_F}{B_{WE}}\right) - 0.5\left(\frac{B_F}{B_{WE}}\right)^2\right] \times \left(\frac{\bar{C}_F}{\bar{C}_{WE}}\right)^{1/2}\left[.0625\int_F - .000472\int_F{}^2\right] \tag{7.1.6}$$

where

$(C_{L_A})_{BASE}$ = linear lift coefficient slope/degrees

B_F = flap span

B_{WE} = exposed wing span

$\bar{C}_F$ - average flap chord

$\bar{C}_{WE}$ = average exposed wing chord

$\int_F$ = flap deflection

7.1.1.1(b) Ground Roll Lift and Drag

During the ground roll, the lift coefficient is determined by

$$C_{L_G} = 57.29\, C_{L_\alpha} \cdot \sin(\alpha_G)\, \cos^2(\alpha_G) + \left[\frac{C_{L_{MAX}}}{\cos(\alpha_{MAX})} - \frac{57.29}{2} C_{L_\alpha} \sin(\alpha_{MAX})\right]$$

$$= F(\alpha_G) \quad \sin^2(\alpha_G)\, \cos(\alpha_G)/\sin^2(\alpha_{MAX}) \qquad (7.1.7)$$

where C_{L_α} is the linear lift curve slope. The corresponding ground roll drag is taken as

$$C_{D_G} = C_{D_o} + k \cdot C_{L_G}[C_{L_G} - C_{L_o} - \Delta C_{L_{FLAP}}] + C_{D_{LG}} \qquad (7.1.8)$$

where

C_{D_o}	=	zero lift drag coefficient
k	=	induced drag factor
C_{L_o}	=	lift coefficient at zero wing incidence
$C_{D_{LG}}$	=	landing gear drag coefficient

7.1.1.(c) Rotation Lift and Drag

The lift coefficient after rotation, C_{L_R}, is given by Equation (7.1.7) with α_R replacing α_G; that is,

$$C_{L_R} = F(\alpha_G) \qquad (7.1.9)$$

The lift coefficient is subject to the condition that

$$C_{L_R} \leq (C_{L_{MAX}})/(1.1)^2 \qquad (7.1.10)$$

This inequality constraint is imposed to prevent buffet or pitch-up problems.

The drag coefficient after rotation is given by

$$C_{D_R} = C_{D_o} + kC_{L_R}[C_{L_R} - C_{L_o} - \Delta C_{L_{FLAP}}] + C_{D_{LG}} \qquad (7.1.11)$$

7.1.1.1(d) Lift and Drag at 50 Foot Obstacle

The lift coefficient at a 50 foot obstacle is based on the rotation lift coefficient.

$$C_{L_{50}} = C_{L_R}/(1.1)^2 \qquad (7.1.12)$$

The corresponding drag is given by

$$C_{D_{50}} = C_{D_o} + k \cdot C_{L_{50}}[C_{L_{50}} - C_{L_o} - \Delta C_{L_{FLAP}}] + C_{D_{LG}} \qquad (7.1.13)$$

7.1.2 Ground Roll and Rotation

The ground roll distance, X_G, is based on the expression

$$X_G = \frac{13.07\left(\frac{W_{TO}}{S \cdot C_{L_R}}\right)}{\frac{F_{TO}}{W_{TO}} - \mu_G - \frac{1}{2}\left(\frac{C_{D_{GR}}}{C_{L_R}}\right)} \qquad (7.1.14)$$

where

$$C_{D_{GR}} = \frac{1}{2}(C_{D_G} + C_{D_R}) \qquad (7.1.15)$$

and

F_{TO} = take-off thrust

W_{TO} = take-off weight

μ_G = vehicle rolling friction coefficient

Time to reach the rotation point is given by

$$T_G = 1.1842\ X_G/V_R \qquad (7.1.16)$$

where the velocity at rotation, V_R, is given by

$$V_R = 17.16\sqrt{\frac{W_{TO}}{SC_{L_R}}} \qquad (7.1.17)$$

Rotation is assumed to occur instantaneously.

7.1.3 Flight to Clear 50 Foot Obstacle

The average drag coefficient between rotation and 50 foot obstacle clearance points is assumed to be

$$C_{D_{R50}} = \frac{1}{2}(C_{D_R} + C_{D_{50}}) \qquad (7.1.18)$$

The distance covered in clearing the obstacle is given by

$$X_{50} = \frac{50.0+2.745\left(\frac{W_{TO}}{S \cdot C_{L_R}}\right)}{\left(\frac{F_{TO}}{W_{TO}}\right) - 1.105\left(\frac{C_{D_{R50}}}{C_{L_R}}\right)} \qquad (7.1.19$$

Time to clear the obstacle after rotation is

$$T_{50} = X_{50}/(1.6889 \times V_R) \qquad (7.1.20)$$

Thus, total distance for take-off over 50 foot obstacle is

$$X_{TO} = X_G + X_{50} \tag{7.1.21}$$

The elapsed time is

$$T_{TO} = T_G + T_{50} \tag{7.1.22}$$

Total fuel used is

$$W_{F_{TO}} = F_{TO} \cdot T_{TO}/(I_{SP_{TO}}) \tag{7.1.23}$$

At the 50 foot obstacle the flight path angle is obtained from

$$\sin(\gamma_{50}) = \frac{F_{TO}}{W_{50}} - \frac{C_{D50}}{C_{L50}} \tag{7.1.24}$$

where

$$W_{50} = W_{TO} - W_{F_{TO}} \tag{7.1.25}$$

The corresponding rate of climb is given by

$$RC_{50} = 1.6889\ V_{50} \sin(\gamma_{50}) \tag{7.1.26}$$

7.1.4 Landing High Lift Aerodynamics

The landing analysis closely follows the take-off analysis but in reverse sequence starting from the 50 foot obstacle. The angle of attack at touch down is

$$\alpha_{TD} = \alpha_{BTD} - 1.0 + \alpha_{WB} \tag{7.1.27}$$

and in the subsequent ground roll

$$\alpha_{LR} = \alpha_{BLR} + \alpha_{WB} \tag{7.1.28}$$

where

α_{TD} = wing incidence at touch down

α_{BTD} = body incidence at touch down

α_{LR} = wing incidence during landing ground roll

α_{BLR} = body incidence in landing ground roll

7.1.5 Landing Lift and Drag

The wing maximum lift coefficient during landing and the corresponding angle of attack are given by Equations (7.1.3) and (7.1.4). Flap incremental lift is given by Equation (7.1.6). It should be noted that the landing configuration parameters such as flap angle and permissible body angle of attack will normally differ significantly between the take-off and landing configurations. At the 50 foot obstacle, configuration lift is assumed to be

$$C_{L_{L50}} = C_{L_{TD}}/(1.1)^2 \qquad (7.1.29)$$

At touchdown, $C_{L_{TD}}$, is based on Equation (7.1.7) using α_{TD}; that is

$$C_{L_{TD}} = F(\alpha_{TD}) + \Delta C_{L_{FLAP}} \qquad (7.1.30)$$

where $\Delta C_{L_{FLAP}}$ is given by Equation (7.1.6) using the landing flap setting. The inequality

$$C_{L_{TD}} < C_{L_{MAX}}/(1.1)^2 \qquad (7.1.31)$$

is used.

Similarly, during the subsequent landing ground roll,

$$C_{L_{LR}} = F(\alpha_{LR}) \qquad (7.1.32)$$

Drag coefficient at the 50 foot obstacle, $C_{D_{L50}}$, is given by Equation (7.1.8) using appropriate landing coefficients. Drag at touchdown, $C_{D_{TD}}$, is given by Equation (7.1.8) using touchdown coefficients. Drag during the landing ground roll is given by

$$C_{D_{LR}} = C_{D_o} + k \cdot C_{L_{LR}}(C_{L_{LR}} - C_{L_o}) + C_{D_{LG}} + C_{D_{CHUT}} \qquad (7.1.33)$$

where

$C_{D_{CHUT}}$ = landing parachute drag

All other symbols are defined in Section 7.1.1.

7.1.6 Flight from 50 Foot Obstacle to Touchdown

Velocity at touchdown is assumed to be

$$V_{TD} = 17.16 \sqrt{\frac{W_L}{S \cdot C_{L_{TD}}}} \qquad (7.1.34)$$

The ground distance covered from 50 foot obstacle to touchdown is

$$C_{L50} = \frac{50.0 + 2.745\left(\frac{W_L}{S \cdot C_{L_{TD}}}\right)}{1.105\left(\frac{C_{D_{TD50}}}{C_{L_{TD}}}\right) - \left(\frac{F_L}{W_L}\right)} \tag{7.1.35}$$

where

X_{L50} = flight distance from 50 foot obstacle to touchdown

W_L = landing weight

$C_{D_{TD50}}$ = $\frac{1}{2}(C_{D_{50}} + C_{D_{TD}})$, the average drag coefficient

F_L = approach thrust

Rate of sink at the 50 foot obstacle is

$$RS_{50} = 1.69\ V_{50}\left[\left(\frac{C_{D_{L50}}}{C_{L_{L50}}}\right) - \left(\frac{F_{L50}}{W_L}\right)\right] \tag{7.1.36}$$

Flight path angle at the 50 foot obstacle is given by

$$\sin(\gamma_{L50}) = \left(\frac{C_{D_{L50}}}{C_{L_{L50}}}\right) - \left(\frac{F_{L50}}{W_L}\right) \tag{7.1.37}$$

The ground roll distance is given by

$$X_{GL} = \frac{13.07\left(\frac{W_L}{S \cdot C_{L_{TD}}}\right)}{\mu + \frac{1}{2}(C_{D_{LG}} - \mu \cdot C_{L_{LR}})/C_{L_{TD}}} \tag{7.1.38}$$

Total landing distance is

$$X_L = X_{L50} + X_{GL} \tag{7.1.39}$$

REFERENCES:

1. Hague, D. S., Optimal Design Integration of Reusable Launch Vehicles, ODIN/RLV, Section 4.4, "U. S. Air Force Stability and Control DATCOM,"

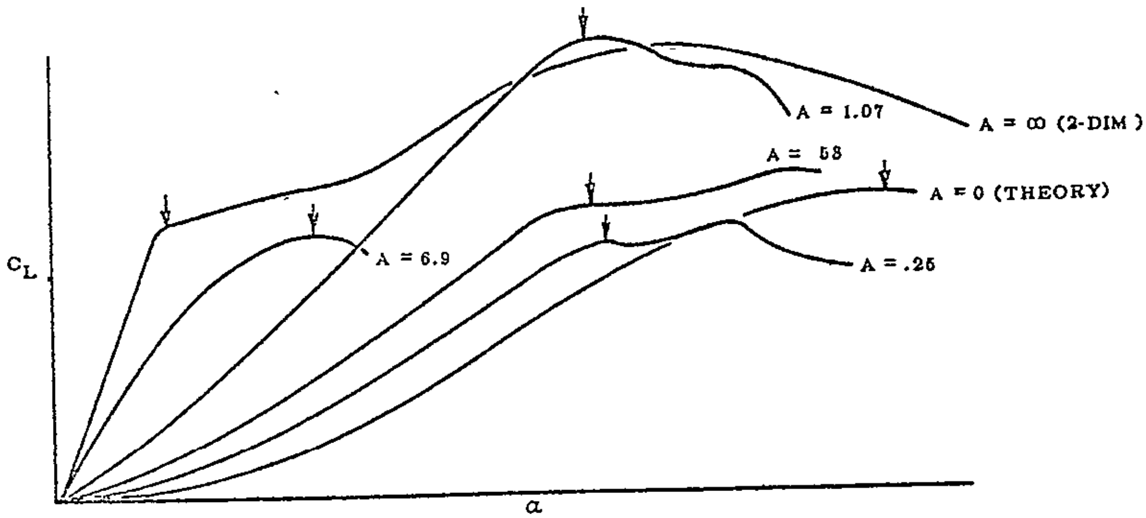


FIGURE 7.1-1. TYPICAL FIRST PEAKS IN LIFT COEFFICIENT VERSUS ANGLE OF ATTACK

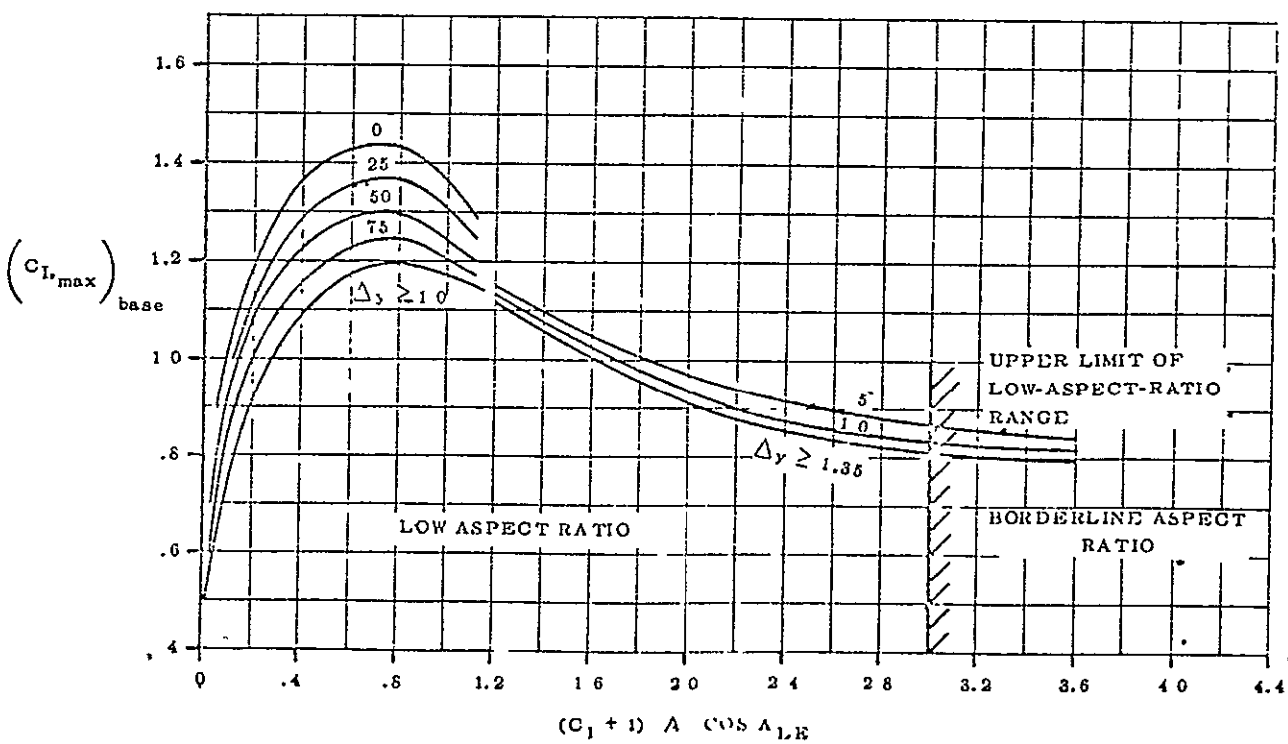


FIGURE 7.1-2. SUBSONIC MAXIMUM LIFT OF WINGS WITH POSITION OF MAXIMUM THICKNESS BETWEEN 35 AND 50 PER CENT CHORD

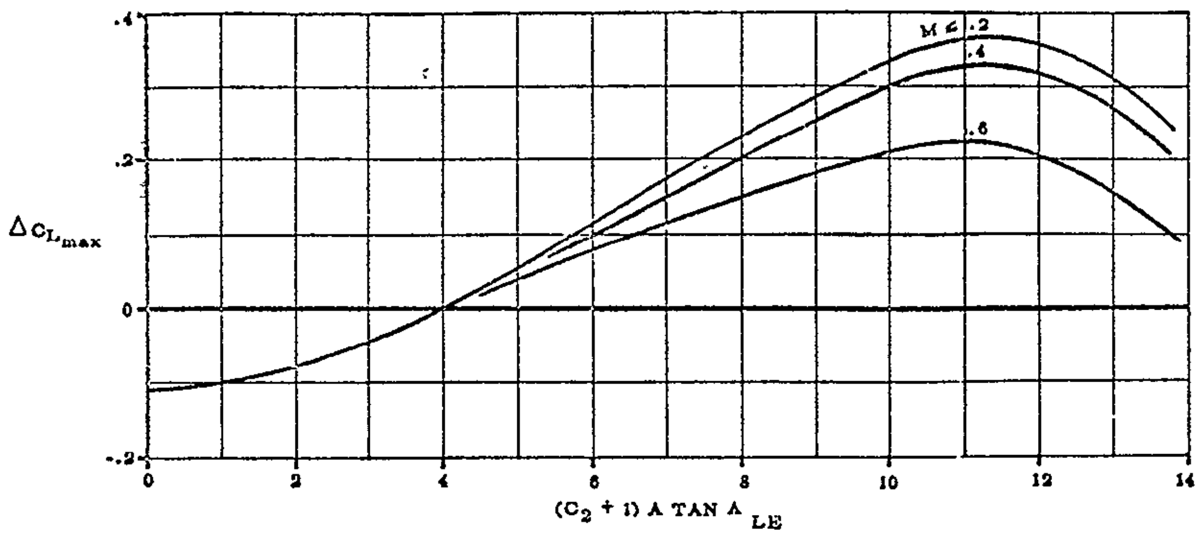


FIGURE 7.1-3. SUBSONIC MAXIMUM LIFT INCREMENT FOR LOW ASPECT RATIO WINGS

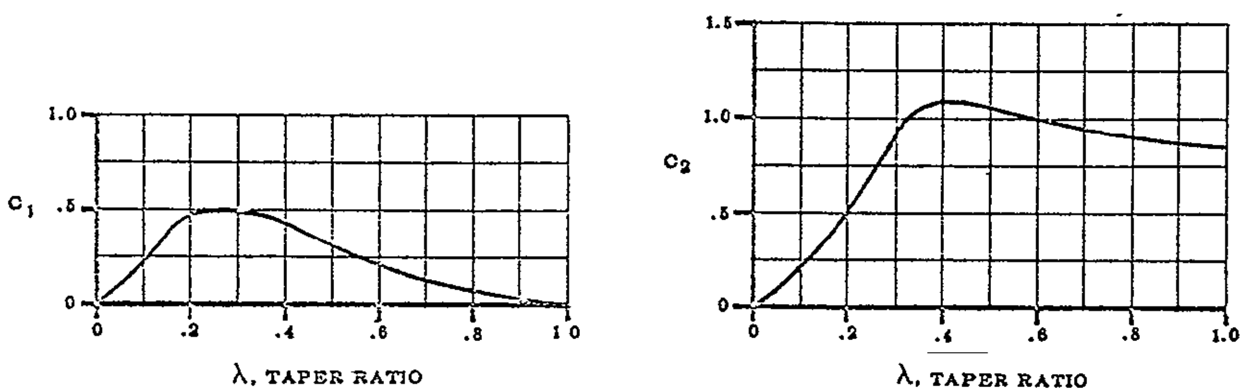


FIGURE 7.1-4. TAPER RATIO CORRECTION FACTORS

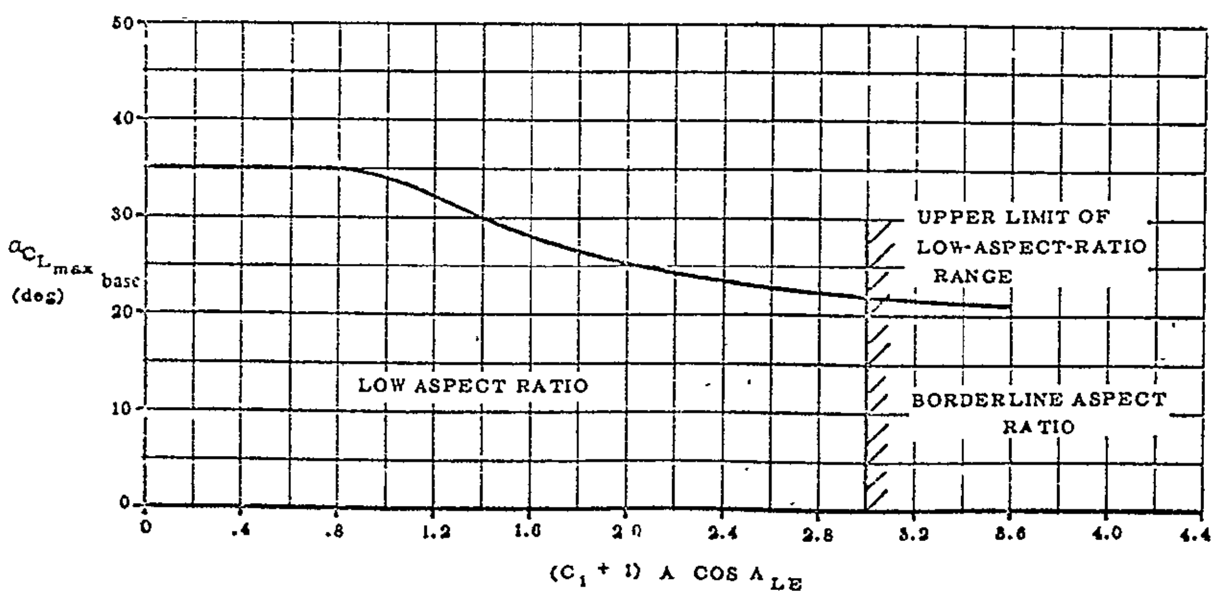


FIGURE 7.1-5. ANGLE OF ATTACK FOR SUBSONIC MAXIMUM LIFT OF LOW ASPECT RATIO WINGS

TABLE OF CONTENTS FOR SECTION 7.2, PROGRAM NSEG II

7.2 PROGRAM NSEG II: A RAPID APPROXIMATE SEGMENTED MISSION PERFORMANCE ANALYSIS CODE

The NSEG II program is an extended version of the Air Force-developed NSEG program, Reference 1. The extended code was constructed under contract F33615-71-C-1480, the ODIN/MFV study. Major changes to NSEG incorporated in NSEG II are a complete code reorganization, more general vehicle specification, addition of energy maneuverability concepts, addition of plotting capability, and development of new data input procedures.

NSEG II provides a generalized mission performance analysis capability based on approximate equations of motion for the state components.

$$\{X_i\} = \{V, h, \gamma, W, R, t\} \tag{7.2.1}$$

In all flight modes the equations of motion

$$\{\dot{X}_i\} = \{f_i(V, h, \gamma, W, R, t; \alpha, B_A, N)\} \tag{7.2.2}$$

are of an approximate nature. For example, in climbs, $\dot{\gamma}$ is neglected.

Approximate equations of motion are available for

1. Take-off
2. Acceleration
3. Climb
4. Cruises and loiters
5. Descents
6. Deceleration
7. Landing

Any number of mission segments may be pieced together to form a complete mission. Segments may be flown in either forward or reverse direction in any sequence specified by the user.

A typical complete mission profile is illustrated in Figure 7.2-1. The program may also be used to generate performance contour plots of the type illustrated in Figure 7.2-2. NSEG II contains a variety of operating modes to aid in mission analysis which include

1. *Point performance* characteristic evaluation where given $\{\bar{X}\} = \{X_i\}$, the function

$$\phi = \phi(X_i) \tag{7.2.3}$$

is evaluated.

2. *Vector performance* evaluation where given

$$\{\bar{X}\} = \{\ldots\ldots, X_i, X_j, \ldots\ldots\} \qquad j = 1, 2, \ldots, N_j \qquad (7.2.4)$$

the vector

$$\{\phi\} = \{\phi_j\} = \{f\ldots, X_i, X_j, \ldots\}\ j = 1, 2, \ldots, N_j \qquad (7.2.5)$$

is evaluated and the maximum or minimum value of ϕ in the region

$$X_{j_L} < X_j < X_{j_H} \qquad (7.2.6)$$

is found by interpolation. That is,

$$\phi_j^* = f(\ldots, X_i, X_j^*, \ldots) \qquad (7.2.7)$$

3. *Map performance* evaluation where given

$$\{\bar{X}\}_{ij} = \{\ldots, X_i, X_j, \ldots\} \qquad i = 1, 2, \ldots, N_i$$
$$j = 1, 2, \ldots, N_j \qquad (7.2.8)$$

the performance array

$$[\phi_{ij}] = [f(\ldots, X_i, X_j, \ldots)] \qquad (7.2.9)$$

is evaluated over a rectangular mesh of points in the (X_i, X_j) plane and the resulting contours obtained in the manner of Figure 7.2-2.

4. *Mission segment performance* where given a state $\{\bar{X}\}_1$, an approximate state equation, and a segment termination criteria, the state transformation

$$\{X\}_1 \rightarrow T_{12} \rightarrow \{\bar{X}\}_2 \qquad (7.2.10)$$

is accomplished.

5. Mission performance where given a sequence of mission segments, the successive state transformations

$$\{\bar{X}\}_1 \rightarrow \{\bar{X}\}_2 \rightarrow \quad \ldots \quad \rightarrow \{\bar{X}_{N-1}\} \rightarrow \{\bar{X}_N\} \qquad (7.2.11)$$

are completed.

The analytic basis of program NSEG II is presented below.

7.2.1 Vehicle Aerodynamic Representation

All aerodynamic representations compute the vehicle drag given a flight condition and lift coefficient. Vehicle lift coefficient required is determined internally by NSEG II on the basis of instantaneous flight conditions.

7.2.1.1 Clean Aircraft

Either of two aerodynamic representations may be employed for the clean aircraft as described below.

7.2.1.1(a) General Form

The clean aircraft drag is computed in the form

$$C_D = C_{D_o} + C_{D_i} \tag{7.2.12}$$

where C_{D_o} is the zero lift drag, and C_{D_i} is the induced drag.

Both C_{D_o} qnd C_{D_i} may be computed by three component summation; that is,

$$C_{D_o} = C_{D_{o1}} + C_{D_{o2}} + C_{D_{o3}} \tag{7.2.13}$$

$$C_{D_i} = C_{D_{i1}} + C_{D_{i2}} + C_{D_{i3}} \tag{7.2.14}$$

In Equation (7.2.13) each of the three component zero lift drags must be of the form

$$C_{D_{oj}} = C_{D_{oj}}(h, M); \; j = 1, 2, 3 \tag{7.2.15}$$

Similarly, in Equation (7.2.14) each induced drag component must be of the form

$$C_{D_{ij}} = C_{D_{ij}}(C_L, M); \; j = 1, 2, 3 \tag{7.2.16}$$

7.2.1.1(b) Polynomial Form

In this aerodynamic option the drag is computed in the component summation form

$$C_D = C_{D_1} + C_{D_2} + C_{D_3} \tag{7.2.17}$$

where

$$C_{D_j} = C_{D_{o_j}} + k_{1_j} \cdot C_L^2 + k_{2_j}(C_L - C_{L_{MIN}})^2 + k_{3_j} C_L^3$$

$$j = 1, 2, 3 \tag{7.2.18}$$

and

$$C_{D_{o_j}} = C_{D_{o_j}}(M) \tag{7.2.19}$$

$$k_{1_j} = k_{1_j}(M) \tag{7.2.20}$$

$$k_{2_j} = k_{2_j}(M) \tag{7.2.21}$$

$$k_{3_j} = k_{3_j}(M) \tag{7.2.22}$$

$$C_{L_{MIN}} = C_{L_{MIN}}(M), \text{ the minimum drag lift coefficient} \tag{7.2.23}$$

7.2.1.2 Store and Pylon Drag

Store and pylon drag is computed in the form

$$C_{D_S} = C_{D_1} + C_{D_2} + C_{D_3} \tag{7.2.24}$$

where

$$C_{D_j} = C_{D_{S_j}} \cdot N_{S_j} + C_{D_{SP_j}} \cdot N_{SP_j} \qquad j = 1, 2, 3 \tag{7.2.25}$$

In Equation (7.2.25) the drag of a single type j store pair is

$$C_{D_{S_j}} = C_{D_{S_j}}(M) \tag{7.2.26}$$

The number of type j store pairs is N_{S_j}. The drag of a single type j store pylon pair is

$$C_{D_{SP_j}} = C_{D_{SP_j}}(M) \tag{7.2.27}$$

The number of type j store pylon pairs is N_{SP_j}

7.2.1.3 Tank and Pylon Drag

Tank and pylon drag is computed in the form

$$C_{D_T} = C_{D_1} + C_{D_2} + C_{D_3} \tag{7.2.28}$$

where

$$C_{D_j} = C_{D_{T_j}} \cdot N_{T_j} + C_{D_{TP_j}} \cdot N_{TP_j} \qquad j = 1, 2, 3 \tag{7.2.29}$$

In Equation (7.2.29) the drag of a single type j tank pair is

$$C_{D_{T_j}} = C_{D_{T_j}}(M) \tag{7.2.30}$$

The number of type j tank pairs is N_{T_j}. The drag of a single type j tank pylon pair is

$$C_{DTP_j} = C_{D_{TP_j}}(M) \qquad (7.2.31)$$

The number of type j tank pylon pairs is N_{TP_j}.

7.2.2 Vehicle Propulsive Representation

All propulsive representations compute the vehicle fuel flow rate given a flight condition and the required thrust. Vehicle required thrust is computed internally by NSEG II on the basis of instantaneous flight conditions.

The maximum thrust, T_{max}, is given by

$$T_{max} = T_{max1},\ T_{max2},\ \text{or}\ T_{max3} \qquad (7.2.32)$$

where

$$T_{maxj} = T_{maxj}(M, h) \qquad (7.2.33)$$

A throttle parameter, N, is determined by

$$N = T_{reqd}/T_{maxj} \qquad j = 1, 2, 3 \qquad (7.2.34)$$

where T_{reqd} is the required thrust. Fuel flow is given by

$$\dot{W}_1 = k \cdot \dot{W}_1(N, M, h) \qquad (7.2.35)$$

or

$$\dot{W}_2 = k \cdot \dot{W}_2(N, M, h) \qquad (7.2.36)$$

or

$$\dot{W}_3 = k \cdot \dot{W}_3(M, h) \qquad (7.2.37)$$

The parameter k is a scalar for adjusting fuel flow to meet various specification requirements. Vehicle thrust, T, is determined by

$$T = T_{reqd},\ T_{max},\ \text{or}\ \bar{N} \cdot T_{max} \qquad (7.2.38)$$

where $\bar{N}$ is a specified throttle setting.

7.2.3 Vehicle Mass Representation

7.2.3.1 Overall Weight Empty

The vehicle overall weight empty is given by the following equation:

ORIGINAL PAGE IS
OF POOR QUALITY

$$W_{OWE} = W_{BARE} + \sum_{i=1}^{3}(N_{S_i} \cdot W_{S_i})_{fixed} + \sum_{i=1}^{3}(N_{SP_i} \cdot W_{SP_i})_{fixed}$$

$$+ \sum_{i=1}^{3}(N_{T_i} \cdot W_{T_i})_{fixed} + \sum_{i=1}^{3}(N_{TP_i} \cdot W_{TP_i})_{fixed} \qquad (7.2.39)$$

where

W_{OWE} = overall weight empty

W_{BARE} = bare weight without stores, tanks, or pylons

N_{S_i} = number of store pairs type i

W_{S_i} = weight of one store par type i

N_{SP_i} = number of store pylon pairs type i

W_{SP_i} = weight of one store pylon pair type i

N_{T_i} = number of tank pairs type i

W_{T_i} = weight of one tank pair type i

N_{TP_i} = number of tank pylon pairs type i

W_{TP_i} = weight of a tank pylon pair type i

The suffix fixed indicates that only fixed tanks, stores or pylons which are not included in the payload must be included in the summations.

7.2.3.2 Fuel Load

The total vehicle fuel load is given by

$$W_{FUEL} = W_{FINT} + \sum_{i=1}^{3}(N_{T_i} \cdot W_{FT_i})_{usable} \qquad (7.2.40)$$

where

W_{FUEL} = total useable fuel weight

W_{FINT} = weight of internal fuel

N_{T_i} = number of tank pairs type i

W_{FT_i} = weight of fuel in one tank pair type i

The suffix usable indicates that the summation only extends over tank pairs which are not included in the payload.

7.2.3.3 Fuel Load

The total non-payload fuel on board at mission initiation is given by

$$W_{FT_o} = W_{F_{INT}} + \sum_{i=1}^{3} (N_{T_i} \cdot W_{FT_i})_{NPL} \qquad (7.2.41)$$

where

W_{FT_o} = initial fuel load

$W_{F_{INT}}$ = initial internal fuel load

N_{T_i} = number of tank pairs type i

W_{FT_i} = weight of fuel in one tank pair type i

and the suffix NPL indicates the summation extends only over tanks which are not assigned to payload. It should be noted that the total fuel is specified directly by data input, and the internal fuel load is a computed quantity.

7.2.3.4 Payload

The total vehicle payload on board at mission initiation is given by

$$W_{PL} = W_{PL_{int}} + \sum_{i=1}^{3} (N_{S_i} \cdot W_{S_i})_{drop} + \sum_{i=1}^{3} (N_{SP_i} \cdot W_{SP_i})_{drop}$$

$$+ \sum_{i=1}^{3} (N_{T_i} \cdot W_{T_i})_{drop} + \sum_{i=1}^{3} (N_{ST_i} \cdot W_{TP_i})_{drop} \qquad (7.2.42)$$

where

W_{PL} = total payload

$W_{PL_{int}}$ = total internal payload

and the remaining quantities in Equation (7.2.42) are defined in Section 7.2.3.1

7.2.4 Planetary Representation

A flat earth planetary model is employed. The gravitational force is a simple inverse square field. A layered atmosphere provides the following options:

1. Tabular 1962 U. S. Standard Atmosphere
2. Analytic 1962 U. S. Standard Atmosphere
3. 1963 Patrick Air Force Base Atmosphere
4. 1959 U. S. Standard Atmosphere
5. January 1966 NASA Atmosphere, 30° North
6. July 1966, NASA Atmosphere, 30° North
7. Arbitrary Atmosphere Generated from Temperature and/or Pressure Variation with Altitude

The NSEG II program basically computes a planar flight path. However, time to turn calculations are available; hence, a three-dimensional path can be analyzed by "folding" the path into a plane, Figure 7.2-3.

7.2.5 Flight Path Analysis

Flight path analysis for take-off, climb, cruise, descent, and landing are included in NSEG II. The analyses are all based on relatively rapid approximate methods. Each flight path analysis model employed is described below.

7.2.5.1 Take-Off

The take-off analysis of the independent program TOLAND, Section 7.1, is also available within the NSEG II program. The take-off analysis performs the transfer

$$\{X\}_{TO} \rightarrow \{X\}_{50} \tag{7.2.43}$$

where the suffix TO indicates state at beginning of take-off, and the suffix 50 indicates state at the 50 foot obstacle.

7.2.5.2 Acceleration at Constant Altitude

The level flight acceleration segment performs the operation

$$\{X\}_2 = \{X\}_1 + \int_{M_1}^{M_2} \{\dot{X}\}dM \simeq \{X_1\} + \sum_i \{\Delta X\}_i \tag{7.2.44}$$

where $\{\Delta X\}_1$ is the state change in accelerating from M_1 to $M_1 + \Delta M$

Given $\{X\}_i = \{V, h, \gamma, W, R, t\}_i$, T_i and D_i, then the velocity change is

$$V = a_s \Delta M \tag{7.2.45}$$

where a_s is the speed of sound at the acceleration altitude and

$$\dot{V}_i = g_o \left[\frac{T-D}{W}\right]_i \tag{7.2.46}$$

The approximate time to accelerate from M_i to $M_i + \Delta M$ is

$$\Delta t' = \Delta V/\dot{V}_i \tag{7.2.47}$$

The corresponding approximate weight change is

$$\Delta W' = \dot{W}_i \, \Delta t' \tag{7.2.48}$$

and

$$W_{i+1} = W_i - \Delta W' \tag{7.2.49}$$

and to the first order

$$\dot{V}_{i+1} = g_o \left[\frac{T-D}{W}\right]_{i+1} \tag{7.2.50}$$

$\dot{W}_{i+1}$ can be obtained at the new Mach number. The mean acceleration is now

$$\bar{\dot{V}} = \frac{1}{2}(\dot{V}_i + \dot{V}_{i+1}) \tag{7.2.51}$$

The time to accelerate is

$$\Delta t = \Delta V/\bar{\dot{V}} \tag{7.2.52}$$

which gives a weight change of

$$\Delta W = \frac{1}{2}(\dot{W}_i + \dot{W}_{i+1})\Delta t \tag{7.2.53}$$

and a range increment

$$\Delta R = \frac{1}{2}(V_i + V_{i+1})\Delta t \tag{7.2.54}$$

The state incremental vector $\{\Delta X\}_i$ is therefore given by

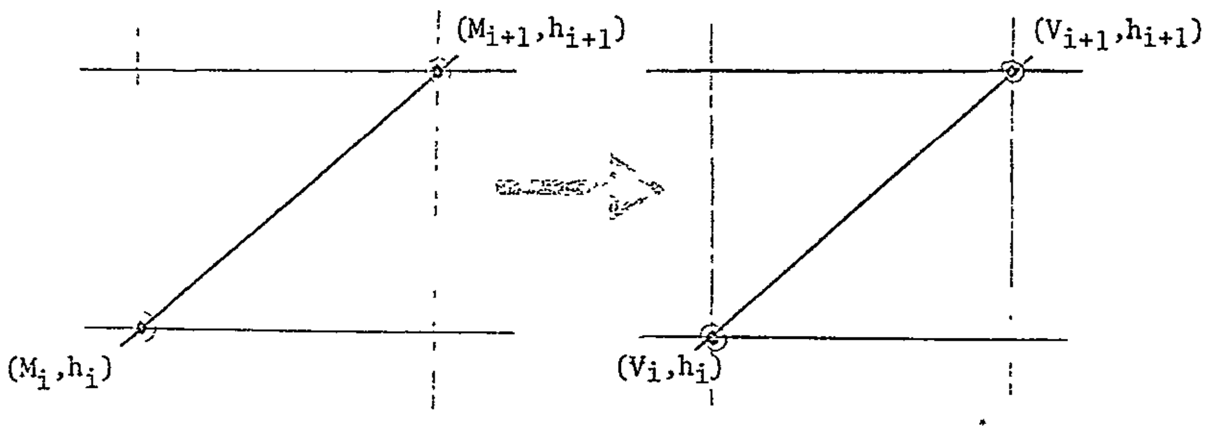
$$\begin{bmatrix} \Delta V \\ \Delta h \\ \Delta \gamma \\ \Delta R \\ \Delta W \\ \Delta t \end{bmatrix}_i = \begin{bmatrix} a_s \Delta M \\ 0 \\ 0 \\ \frac{1}{2}(\dot{W}_i + \dot{W}_{i+1})\Delta t \\ \frac{1}{2}(V_i + V_{i+1})\Delta t \\ \dfrac{2\Delta V}{(\dot{V}_i + \dot{V}_{i+1})} \end{bmatrix}_i \tag{7.2.55}$$

7.2.5.3 Accelerating Climbs

All accelerating climb paths are formed by a sequence of elemental straight line arcs in the Mach-altitude plane. On any arc the vehicle flies from (M_i, h_i) to (M_{i+1}, h_{i+1}). Since the vehicle is climbing

$$h_{i+1} > h_i \tag{7.2.56}$$

The typical arc for a climb path is shown below. The Mach-altitude plane



can be transformed into the velocity-altitude plane as follows:

$$V = V(h, M) \tag{7.2.57}$$

so that

$$\Delta V = \frac{\partial V}{\partial h} \cdot \delta h + \frac{\partial V}{\partial M} \delta M \tag{7.2.58}$$

or

$$\frac{dV}{dh} = \frac{\partial V}{\partial h} + \frac{\partial V}{\partial M} \cdot \frac{\partial M}{\partial h} \tag{7.2.59}$$

$$= \frac{\partial V}{\partial h} + a \frac{dM}{dh} \tag{7.2.60}$$

where a is the local speed of sound

$$V = aM \tag{7.2.61}$$

Now $\partial V/\partial h$ is the change in velocity with altitude at constant Mach number, and from Equation (7.2.61) with M constant

$$\frac{\partial V}{\partial h} = M \frac{\partial a}{\partial h} \tag{7.2.62}$$

$$= M \frac{\partial a}{\partial T_R} \cdot \frac{dT_R}{dh} \tag{7.2.63}$$

where T_R is temperature ratio, T/T_{SL}, so that

$$\frac{\partial V}{\partial h} = \left(\frac{V}{a}\right) \cdot \frac{\partial a}{\partial T_R} \cdot \frac{dT_R}{dh} \qquad (7.2.64)$$

and from the atmospheric model

$$a = 1116.45(T_R)^{1/2} \qquad (7.2.65)$$

$$\frac{\partial a}{\partial T_R} = \frac{a}{2T_R} \qquad (7.2.66)$$

Substituting into Equation (7.2.64)

$$\frac{\partial V}{\partial h} = \frac{V}{2T_R} \cdot \frac{dT_R}{dh} \qquad (7.2.67)$$

Substituting Equation (7.2.67) into Equation (7.2.60)

$$\frac{dV}{dh} = \frac{V}{2T_R} \cdot \frac{dT_R}{dh} + a\,\frac{dM}{dh} \qquad (7.2.68)$$

Equation (7.2.68) is used to define the required variation of velocity with altitude over an elemental climbing arc.

Now the rate of climb is

$$\frac{dh}{dt} = RC \qquad (7.2.69)$$

or

$$\frac{dh}{RC} = dt \qquad (7.2.70)$$

Assuming rate of climb varies linearly with altitude in the elemental arc

$$RC = a + bh \qquad (7.2.71)$$

Substituting into Equation (7.2.70) and integrating

$$\int_{h_1}^{h_2} \frac{dh}{(a+bh)} = \int_{t_1}^{t_2} dt \qquad (7.2.72)$$

or

$$t_{i+1} - t_i = \left[\frac{h_{i+1}-h_i}{RC_{i+1}-RC_i}\right] \log\left(\frac{RC_{i+1}}{RC_i}\right) \qquad (7.2.73)$$

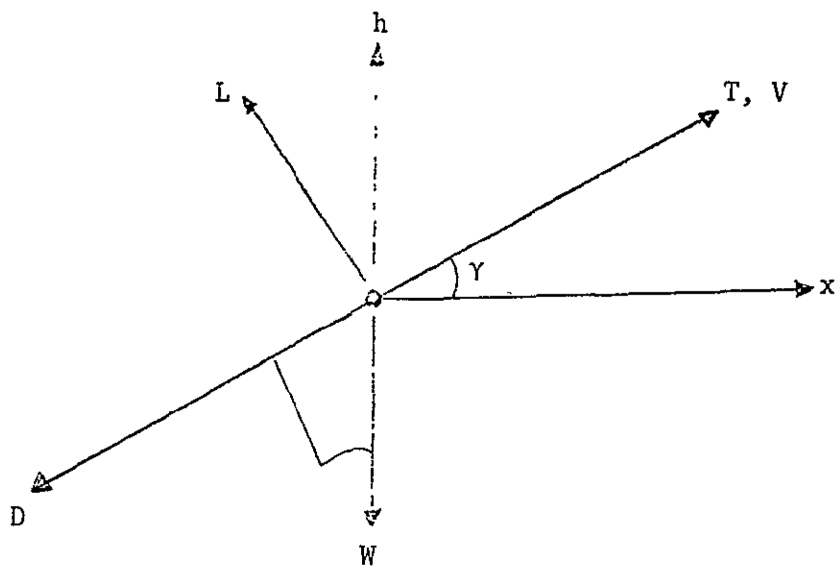
ORIGINAL PAGE IS
OF POOR QUALITY

where

$$a = RC_i \tag{7.2.74}$$

$$b = \frac{RC_{i+1}-RC_i}{h_{i+1}-h_i} \tag{7.2.75}$$

The vehicle rate of climb is computed under the assumption that thrust is aligned along the velocity vector as shown below.



Now

$$m\dot{V} = (T - D) - W \sin\gamma \tag{7.2.76}$$

but

$$m\dot{V} = m\,\frac{dV}{dh}\cdot\frac{dh}{dt} = \frac{W}{g}\,\frac{dV}{dh}\cdot V \sin\gamma \tag{7.2.77}$$

Combining Equations (7.2.76) and (7.2.77)

$$\sin\gamma = \frac{T-D}{W[\frac{V}{g}\frac{dV}{dh}+1.0]} \tag{7.2.78}$$

and

$$\cos\gamma = \sqrt{1-\sin^2\gamma} \tag{7.2.79}$$

so that

$$RC = V\sin\gamma = \left[\frac{(T-D)V}{W}\right] \div \left[\frac{V}{g}\,\frac{dV}{dh} + 1.0\right] \tag{7.2.80}$$

Equation (7.2.80) can be evaluated at each end of the elemental arc to obtain RC_{i+1} and RC_i. Hence, Δt, the time to traverse the elemental arc, is given by Equation (7.2.73). Similarly, the flight path angle at each end of the arc can be obtained from Equation (7.2.78). It should be noted that if $\sin\gamma$, Equation (7.2.79), is greater than 1.0, the approximate climb analysis is invalid. If this condition occurs, the thrust is reduced to produce a climb along the elemental arc at 89.5 degrees.

Summarizing, the state incremental vector for an accelerating climb is given by

$$\begin{bmatrix} \Delta V \\ \Delta h \\ \Delta\gamma \\ \Delta R \\ \Delta W \\ \Delta t \end{bmatrix} = \begin{bmatrix} V_{i+1} - V_i \\ h_{i+1} - h_i \\ \gamma_{i+1} - \gamma_i \\ \frac{1}{2}[V_{i+1}\cos\gamma_{i+1} + V_i\cos\gamma_i] \\ \frac{1}{2}[\dot{W}_{i+1} + \dot{W}_i]\Delta t \\ \frac{h_{i+1}-h_i}{RC_{i+1}-RC_i}\log\left[\frac{RC_{i+1}}{RC_i}\right] \end{bmatrix} \tag{7.2.81}$$

7.2.5.4 Cruise Flight

Cruise flight performance is computed by the Bruguet equation. With constant velocity the distance travelled in time Δt is

$$\Delta R = V \cdot \Delta t \tag{7.2.82}$$

Now

$$SFC = \frac{\dot{W}}{T} \tag{7.2.83}$$

so that

$$\Delta t = \frac{\Delta W}{(SFC)\cdot T} \tag{7.2.84}$$

Substituting in Equation (7.2.82)

$$\Delta R = \frac{V}{(SFC)\cdot T} \cdot \Delta W \tag{7.2.85}$$

In cruise flight

$$\frac{L}{D} \simeq \frac{W}{T} \tag{7.2.86}$$

and

$$\Delta R = \frac{V}{SFC} \left(\frac{L}{D}\right) \frac{\Delta W}{W} \qquad (7.2.87)$$

On integrating

$$R_{i+1} - R_i = \frac{V}{SFC} \left(\frac{L}{D}\right) \log\left(\frac{W_{i+1}}{W_i}\right) = (RF) \log\left(\frac{W_{i+1}}{W_i}\right) \quad . \qquad (7.2.88)$$

Where the range factor RF is usually a slowly changing function of weight. NSEG II uses the inverse relationship to compute the weight change given a range increment

$$W_{i+1} = W_i e^{\left[\frac{R_{i+1}-R_i}{RF}\right]} = W_i e^{\left[\frac{(R_{i+1}-R_i)\cdot(SFC)}{V(L/D)}\right]} \qquad (7.2.89)$$

Alternatively, the program can be used with a time increment Δt by using the relationship

$$W_{i+1} = W_i e^{\left[\frac{\Delta t \cdot SFC}{(L/D)}\right]} \qquad (7.2.90)$$

Several cruise modes are contained in the program including

1. Constant altitude, constant Mach number cruise
2. Constant altitude, constant C_L cruise
3. Constant Mach number, constant C_L

Each of the three cruise modes may be performed in the manner

1. From R_i to $R_{i+1} = \Delta R_i$
2. From T_i to $T_{i+1} = \Delta T_i$
3. From W_i to $W_{i+1} = \Delta W_i$

A cruise flight is computed by summing over N_i steps. Thus,

$$\Delta R_{cruise} = \sum_i \Delta R_i$$

$$\Delta T_{cruise} = \sum_i \Delta T_i$$

or

$$\Delta W_{cruise} = \sum_i \Delta W_i$$

ORIGINAL PAGE IS
OF POOR QUALITY

In all cases the total state increments are summed in the manner

$$\{\Delta X\}_{cruise} = \sum_i \{\Delta X\}_i \tag{7.2.91}$$

A mean range factor is used in all cruise calculations. The mean range factor, $(\overline{RC}_i)$, in each elemental arc bounded by $\{X\}_i$ and $\{X\}_{i+1}$ is determined by an appropriate weighting of the range factors RC_i and RC_{i+1} which bound the arc.

7.2.5.5 Descent

The climb analysis of Section 7.2.5.2 is also used for the descent analysis. If the size of the flight path angle becomes too small ($\sin\gamma < -1$), the engine is throttled back to maintain a realistic flight path approximation.

7.2.5.6 Level Flight Acceleration

The approximate time to accelerate from M_i to M_{i+1} in level flight is

$$\Delta t_i' = a_s(M_{i+1} - M_i)/\dot{V}_i \tag{7.2.92}$$

with a corresponding weight change

$$\Delta W' = \dot{W}_i \, \Delta t' \tag{7.2.93}$$

so that

$$W_{i+1}' = W_i - \dot{W}_i \, \Delta t' \tag{7.2.94}$$

Therefore, to the first order

$$\dot{V}_{i+1} = g_o \left[\frac{T-D}{W}\right] \tag{7.2.95}$$

The fuel flow at this point, $\dot{W}_{i+1}$, can be obtained from the vehicle aerodynamic and propulsion representation.

$$\bar{\dot{V}}_i = \frac{1}{2}(\dot{V}_i + \dot{V}_{i+1}) \tag{7.2.96}$$

and an improved estimate of the time to accelerate from M_i to M_{i+1} is

$$\Delta t_i = a_s(M_{i+1} - M_i)/\bar{\dot{V}}_i \tag{7.2.97}$$

This gives an improved estimate of the weight change

$$\Delta W_i = \frac{1}{2}(\dot{W}_i + \dot{W}_{i+1})\ \Delta t_i \tag{7.2.98}$$

and the corresponding range change

$$\Delta R_i = \frac{1}{2}(V_i + V_{i+1})\ \Delta t_i \tag{7.2.99}$$

Summarizing, the level acceleration state increment is

$$\begin{bmatrix} \Delta h \\ \Delta V \\ \Delta \gamma \\ \delta W \\ \delta R \\ \Delta t_i \end{bmatrix} = \begin{bmatrix} 0.0 \\ a_s(M_{i+1} - M_i) \\ 0.0 \\ \frac{1}{2}(\dot{W}_i + \dot{W}_{i+1})\ \Delta t_i \\ \frac{a_s}{2}(M_{i+1} + M_i)\ \Delta t_i \\ \frac{a_s}{2g_o}\ (M_{i+1} - M_i)/[(\frac{T-D}{W})_i + (\frac{T-D}{W})_{i+1}] \end{bmatrix} \tag{7.2.100}$$

7.2.5.7 Landing

The NSEG II landing analysis is that described in the independent take-off and landing program TOLAND of Section 7.1.

7.2.6 Mission Segments

The state incremental methods of Section 7.2.5 are used to create a variety of optional mission segments in NSEG II. Each available mission segment option is briefly described below. All mission climbs, cruises, accelerations, and decelerations may be performed in forward or reverse direction. Each mission segment described below is performed as a distinct option in NSEG II. There is some degree of overlapping capability in the available mission options. The mission option within NSEG II is indicated for each mission segment for reference purposes.

7.2.6.1 Linear Climb (Mission Option 1)

This option climbs linearly from (M_1, h_1) to (M_2, h_2) using a specified number of linear climb steps from (M_i, h_i) to (M_{i+1}, h_{i+1}). The path is illustrated in Figure 7.2-4.

7.2.6.2 Climb at Specified Dynamic Pressure (Mission Option 1)

This option climbs along a specified dynamic pressure line from $(M_1\ h_1)$ to $(M_2\ h_2)$ with appropriate terminal maneuvers. Along the constant dynamic pressure line a specified number of linear Mach-altitude segments are flown. Appropriate initial and final maneuvers are used when $(M_1\ h_1)$ or $(M_2\ h_2)$ do not lie on the specified dynamic pressure line. The user may specify a climb at the terminal end point dynamic pressure. In this case, the final maneuver is not required. The various path types which may be generated by this climb mission segment option are illustrated in Figure 7.2-4.

7.2.6.3 Rutowski Climb (Mission Option 1)

The Rutowski climb, Reference 2, flies from $(M_1\ h_1)$ to $(M_2\ h_2)$ along the path which most rapidly builds up specific energy. If either of the points $(M_1\ h_1)$ and $(M_2\ h_2)$ do not lie on this path, an appropriate terminal maneuver is employed. The Rutowski path is found by the following procedure.

1. Compute the initial point specific energy

$$E_1 = V_1^2/2g + h_1 \tag{7.2.101}$$

   and find specific energy at end point

$$E_2 = V_2^2/2g + h_2 \tag{7.2.102}$$

   and divide the energy change $(E_2 - E_1)$ into N equal increments

2. Search at each incremental energy level

$$E_i = E_1 + i \cdot \Delta E \qquad i = 1, 2, \ldots, N \tag{7.2.103}$$

   to find the point of maximum specific energy derivative, (M_i, h_i) where

$$\dot{E}_i = (T_i - D_i)V_i/W_i \tag{7.2.104}$$

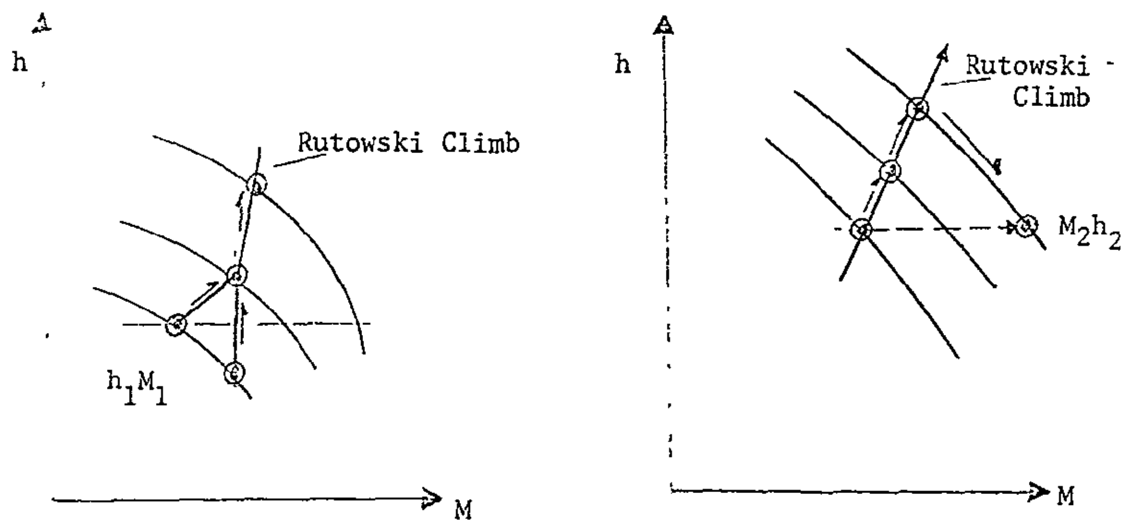
   The $\dot{E}$ calculation is carried out for specified weight and load factor.

3. Fly a sequence of linear Mach-altitude flight increments joining the point (M_i h_1) and (M_{i+1} h_{i+1})

A typical Rutowski path obtained from the program is illustrated in Figure 7.2-5. The initial acquisition of the Rutowski path at

$$h = h_1 + \Delta h \tag{7.2.105}$$

takes a vehicle from its initial condition to the Rutowski path with a velocity loss if this is required. The final maneuver may be either a transfer along a constant energy line from the Rutowski point at the final energy to the point M_2h_2. Alternatively, an altitude limit may be placed on the path such that when a Rutowski point lies above the final point, a transfer to the final point M_2h_2 occurs. These terminal maneuvers are sketched below.



(a) Initial Maneuvers (b) Final Maneuvers

The Rutowski path will observe both $C_{L_{max}}$ and maximum dynamic pressure constraints at the user's option. The thrust levels, vehicle weight, and load factors employed in the $\dot{E}$ calculation are specified by the user. Further details of this mission segment option may be found in Reference 3.

7.2.6.4 Maximum Rate of Climb (Mission Option 1)

A maximum rate of climb path between M_1 h_1 and M_2 h_2 is generated in a similar manner to the Rutowski path of Section 7.2.6.3. However, in the maximum rate of climb path the search for maximum $\dot{E}$ is carried out at the constant altitudes

$$h = h_1 + i \cdot \Delta h \qquad i = 1, 2, \ldots, N \qquad (7.2.106)$$

where the altitude differential (h_2-h_1) has been divided into N equal increments. A typical maximum rate of climb path has been added to Figure 7.2-5.

7.2.6.5 Maximum Acceleration (Mission Option 1)

A maximum acceleration path between M_1 h_1 and M_2 h_2 is generated in a similar manner to the Rutowski path of Section 7.2.6.3. However, in the maximum acceleration path the search for maximum $\dot{E}$ is carried out at the constant Mach numbers.

$$M = M_1 + i \cdot \Delta M \qquad i = 1, 2, \ldots, N \qquad (7.2.107)$$

where the Mach number differential (M_2 - M_1) has been divided into N equal increments. A typical maximum acceleration path has been added to Figure 7.2-5. The maximum acceleration path satisfies the $C_{L_{max}}$ and maximum dynamic pressure constraints of the Rutowski path. In addition, the condition

$$\Delta E_{i+1} \geqslant \Delta E_i \qquad (7.2.108)$$

is imposed. That is, the sequence of points, M_i h_i used in the acceleration will never produce a loss of specific energy. This is illustrated in Figure 7.2-6.

7.2.6.6 Minimum Fuel Paths (Mission Option 1)

Minimum fuel path for given energy, altitude, and Mach number are obtained in a manner similar to Sections 7.2.6.3 through 7.2.6.5, respectively. However, the search optimization criteria on $\dot{E}$ is replaced by the criteria

$$\varphi = \text{Maximum } [\dot{E}/\dot{W}] = \text{Max}\left[\frac{dE/dt}{dW/dt}\right] = \text{Max}\left[\frac{dE}{dW}\right] \qquad (7.2.109)$$

When the search is carried out along lines of constant energy, the minimum fuel energy build up is found. When the search occurs at constant altitude, the minimum fuel climb is found. When the search occurs at constant Mach number the minimum fuel acceleration is found. All appropriate terminal maneuvers and constraints described in Sections 7.2.6.3 to 7.2.6.5 are included in the minimum fuel paths. Some typical paths obtained from the NSEG II program are illustrated in Figure 7.2-7.

7.2.6.7 Maximum Range Glide (Mission Option 1)

The maximum range glide path is obtained when the vehicle flies along the laws of the L/D contours tangency points to an appropriate path generating surface such as constant energy, constant altitude, or constant Mach number. The maneuvers are thus similar to those of Sections 7.2.6.3 to 7.2.6.5 using the optimization criteria .

$$\phi = \text{Maximum } [L/D] \qquad (7.2.110)$$

When the search is carried out along lines of constant energy, the maximum range glide for a given energy loss is found, Reference 4. When the search occurs at constant altitude, the maximum range glide for a given altitude loss is found. When the search occurs at constant Mach number, the maximum range glide for a given velocity loss is found. Some typical paths obtained from the NSEG II program are illustrated in Figure 7.2-8.

7.2.6.8 Range Biased Ascents (Mission Option 1)

Range biased ascents can be obtained when the vehicle flies along the locus of the T/(L-D) contours tangency points to an appropriate path generating surface. This can be seen as follows:

$$E = h + V^2/2g \qquad (7.2.111)$$

and

$$m\left(\frac{dv}{dt}\right) = T - D - W \sin\gamma \qquad (7.2.112)$$

Now

$$R = \int dR = \int \frac{dR}{dE}\, dE = \int \frac{dR}{dt} \cdot \frac{dt}{dE} \cdot dE \qquad (7.2.113)$$

There from Equation (7.2.111)

$$R = \int \frac{V \cos\gamma \cdot dE}{\frac{dh}{dt} + \frac{V}{g} \cdot \frac{dV}{dt}}$$

$$= \int \frac{\cos\gamma \cdot dE}{\sin\gamma + \frac{1}{g}\frac{dV}{dt}} \qquad (7.2.114)$$

But from Equation (7.2.112)

$$\frac{1}{g} \cdot \left(\frac{dV}{dt}\right) = \frac{T-D}{W} - \sin\gamma \qquad (7.2.115)$$

ORIGINAL PAGE IS
OF POOR QUALITY

So that

$$R = \int \frac{W \cos\gamma \cdot dE}{T-D} \qquad (7.2.116)$$

Assuming that range biased ascents occur at small flight path angles with $L \simeq W$, Equation (7.2.116) becomes

$$R \simeq \int \frac{L}{T-D} \; dE \qquad (7.2.117)$$

Therefore, an energy-like approximation for a range biased ascent is to fly is a maximum at each energy level. It should be noted that when $T - D = 0$, no energy gain is possible; therefore, this singular condition must be avoided. In NSEG II the per cent excess of thrust over drag which is acceptable is a program input.

In a manner similar to Sections 7.2.6.3 to 7.2.6.5 a range biased ascent between two energy levels occurs when the points of tangency between constant energy and T/(L-D) contours is flown. A range biased climb between two altitudes will fly the points of tangency between constant altitude and constant T/(L-D) contours. A range biased acceleration will fly the points of tangency between constant Mach number and constant ~~T/(L~~-D) contours.

7.2.6.9 Range Biased Ascents Based On Range Factor (Mission Option 1)

A second series of range biased ascents can be found on the basis of the range factor contours. These ascents are similar to those of Section 7.2.6.8 with range factor replacing T/(L-D).

7.2.6.10 Maximum Lift Coefficient Climb or Descent (Mission Option 2)

The maximum lift coefficient path climbs from M_1 h_1 to M_2 h_2 in N increments of altitude

$$h_i = h_1 + i \cdot \Delta h \qquad i = 1, 2, \ldots, N \qquad (7.2.118)$$

At each altitude the Mach number for maximum rate of climb using the angle of attack for $C_{L_{MAX}}$ is found

$$M_i = M_{i_{MAX\ RC}} \qquad (7.2.119)$$

The vehicle uses the linear Mach-altitude path path-follower to climb between $M_i h_i$ and M_{i+1} h_{i+1}.

Descents follow the same procedure as climbs, but in reverse order.

7.2.6.11 Radius Adjustment (Mission Option 3)

This mission segment option performs an iteration on the range of one cruise segment to make the total range over the combined mission segments S_j; $j = J_1, J_2 \, . \, . \, ., J_N$ equal to the total range over the combined mission segments S_k; $k = K_1, K_2, \, . \, . \, ., K_N$. That is

$$R = \sum_j R_j = \sum_k R_k \qquad (7.2.120)$$

where one of the ΔR_j, and only one, is being adjusted to satisfy the range equality.

7.2.6.12 Cruise Climb to Specified Weight (Mission Option 4)

As an aircraft cruises at the Mach number and altitude for maximum range factor, Equation (7.2.88), the weight reduces. As the weight changes, the altitude for best range factor changes while the Mach number remains approximately constant. The altitude change results from the requirement to maintain the angle of attack for maximum lift coefficient. Thus, as the cruise progresses the altitude increases.

The cruise may be performed in one step or it may be reduced to a sequence of five steps between latter case $W = W_1$ and $W = W_2$, Section 7.2.5.3. At the start of the i^{th} segment in this

$$W_i = W_1 + i \cdot \Delta W \qquad i = 1, 2, \, . \, . \, ., 5 \qquad (7.2.121)$$

Each segment is flown at constant Mach number and lift coefficient and, hence, involves a climbing cruise. At the beginning of each cruise step the weight is instantaneously adjusted to the best altitude for the current weight.

7.2.6.13 Cruise Climb for Specified Distance or Time (Mission Option 5)

This mission segment option performs a cruise climb, Section 7.2.5.3, for specified distance or time. The cruise may be performed with or without range credit. This form of cruise flight is performed in one step.

7.2.6.14 Constant Altitude Cruise Between Two Weights, (Mission Option 6)

This mission segment performs either

1. Constant altitude, constant Mach number cruise
2. Constant altitude, constant lift coefficient cruise

between two weights W_1 and W_2. The cruise is performed in one step, see Section 7.2.5.3.

7.2.6.15 Constant Altitude Cruise for Given Distance (Mission Option 7)

This mission segment performs either

1. Constant altitude, constant Mach number cruise
2. Constant altitude, constant lift coefficient cruise

between two distances R_1 and R_2. The cruise is performed in one step, see Section 7.2.5.3.

7.2.6.16 Constant Altitude Cruise for Given Time (Mission Option 8)

This mission segment performs either

1. Constant altitude, constant Mach number cruise
2. Constant altitude, constant lift coefficient cruise

between two times T_1 and T_2. The cruise is performed in one step, see Section 7.2.5.3. This segment may be performed with or without range credit.

7.2.6.17 Buddy Refuel Cruise (Mission Option 9)

This mission segment determines the optimum in-flight refuelling point and how much fuel will be transferred. The tanker fuel off load capability is specified at three range/fuel combinations and a parabolic variation in available fuel as a function of range is assumed. That is,

$$W_f = a + bR + cR^2 \tag{7.2.122}$$

Cruise flight is assumed in any one of the three forms

1. Constant Mach number, constant lift coefficient
2. Constant Mach number, constant altitude cruise
3. Constant lift coefficient, constant altitude cruise

A maximum range for refuelling may be specified. Refuelling will occur at any point on the segment where

1. Fuel receivable is greater than or equal to fuel available

$$W_{F_R} \geq W_{F_A} \tag{7.2.123}$$

2. Distance flow is equal to maximum refuelling range

3. Minimum in-flight weight of the vehicle receiving fuel is reached where

$$W_{MIN} = W_{OWE} + W_{PL} + W_F \cdot k_F \qquad (7.2.124)$$

where k_F is the unusable residual fuel in the non-payload fuel. For refuelling purposes the maximum weight is taken to be the take-off weight

$$W_{MAX} = W_{TO} \qquad (7.2.125)$$

7.2.6.18 Mach-Altitude-Weight Transfer (Mission Option 10)

This mission segment option retrieves state components at the end of flight segment i and makes them available as the initial conditions for flight segment j. The initial conditions for segment j are thus a linear transformation of the final condition of segment i,

$$\{X\}_j = [P]_{ij} \{X\}_i \qquad (7.2.126)$$

Currently, the NSEG II program is limited to a simple state component transfer on any combination of the three components: Mach number, altitude, or weight.

7.2.6.19 Alternate Mission Selection Option (Mission Option 11)

This mission option retrieves either of two mission segments on the basis of terminal Mach number, altitude or weight. Retrieval criteria may be based on any one of six possibilities:

$$\phi = \text{Min}\ [M_1, M_2] \qquad (7.2.127)$$

$$\phi = \text{Max}\ [M_1, M_2] \qquad (7.2.128)$$

$$\phi = \text{Min}\ [h_1, h_2] \qquad (7.2.129)$$

$$\phi = \text{Max}\ [h_1, h_2] \qquad (7.2.130)$$

$$\phi = \text{Min}\ [W_1, W_2] \qquad (7.2.131)$$

$$\phi = \text{Max}\ [W_1, W_2] \qquad (7.2.132)$$

The segment to be retained is the one which satisfies the selected performance criteria.

7.2.6.20 Instantaneous Weight Change (Mission Option 12)

This mission segment option permits an instantaneous change in vehicle weight, ΔW. The operation

$$W_{i+1} = W_i - \Delta W \tag{7.2.133}$$

is performed.

7.2.6.21 Instantaneous Mach/Altitude Change (Mission Option 13)

This mission segment option provides an instantaneous change in vehicle Mach number, ΔM, and an instantaneous altitude change, Δh. The new Mach number, M_{i+1}, and altitude, h_{i+1}, are specified directly; thus

$$\Delta M = M_{i+1} - M_i \tag{7.2.134}$$

$$\Delta h = h_{i+1} - h_i \tag{7.2.135}$$

7.2.6.22 General Purpose and Point Condition Calculation (Mission Option 14)

This mission segment option provides any of a variety of calculations described below:

1. *Best cruise altitude for given Mach number and weight based on range factor*

$$\underset{h}{\mathrm{Max}}[RF;\ M,\ W] \tag{7.2.136}$$

2. *Ceiling for a specified rate of climb at given Mach number and weight*

$$\underset{h}{\mathrm{Max}}[RC;\ M,\ W] \tag{7.2.137}$$

3. *Mach number for maximum lift coefficient at given weight and altitude*

$$\underset{M}{\mathrm{Max}}[C_L;\ W,\ h] \tag{7.2.138}$$

4. *Mach number for specified lift coefficient given weight and altitude*

$$\underset{M}{\mathrm{Find}}[C_L;\ W,\ h] \tag{7.2.139}$$

5. *Maximum endurance Mach number given altitude, weight, and maximum lift coefficient*

$$\underset{M}{\mathrm{Min}}[\dot{W};\ W,\ h,\ C_{L_{max}}] \qquad (7.2.140)$$

6. *Maximum Mach number at given weight and altitude*

$$\underset{M}{\mathrm{Max}}[M;\ W,\ h] \qquad (7.2.141)$$

7a. *Mach number for maximum rate of climb at given weight and altitude*

$$\underset{M}{\mathrm{Max}}[RC;\ W,\ h] \qquad (7.2.142)$$

7b. *Mach number for maximum rate of climb per pound of fuel at given weight and altitude*

$$\underset{M}{\mathrm{Max}}[dh/dW;\ W,\ h] \qquad (7.2.143)$$

8. *Approximate Mach number and altitude for maximum range factor given weight*

$$\underset{(M,h)}{\mathrm{Max}}[RF;\ W] \qquad (7.2.144)$$

9. *Mach number for maximum range factor given altitude and weight*

$$\underset{M}{\mathrm{Max}}(RF;\ h,\ W] \qquad (7.2.145)$$

10. *Various energy maneuverability parameters at specified load factor given Mach, altitude, and weight*

   a. The required lift coefficient

   b. Specific excess power

$$P_S = \dot{E} = (T-D)V/W$$

   c. Specific excess power divided by fuel flow

$$P_S/\dot{W} = \dot{E}/\dot{W} = (T-D)V/(W \quad \dot{W}) \qquad (7.2.146)$$

   d. Specific excess power divided by fuel flow and multiplied by fuel remaining (ΔE capability) measure

$$P_S \frac{W_F}{\dot{W}} = \frac{\dot{E}\ W_F}{\dot{W}} = \frac{dE}{dW} \cdot W_F \approx \Delta E \qquad (7.2.147)$$

e. Specific energy

$$E_S = h + V^2/2g \qquad (7.2.148)$$

f. Load factor at P_S=0.0

g. Steady state turn radius computed as follows:

$C_L = C_L$, for given load factor

Now for given bank angle, B_A

$$W = qSC_L \cdot \cos(B_A) \qquad (7.2.149)$$

and the centrifugal force is

$$L\sin(B_A) = \frac{WV^2}{Rg} = \frac{L\cos(B_A)V^2}{Rg} \qquad (7.2.150)$$

$$\therefore \quad R = \frac{V^2}{g\tan \cdot B_A} \qquad (7.2.151)$$

but from Equation (7.2.149)

$$\cos B_A = \frac{W}{gSC_L} \qquad (7.2.152)$$

$$\therefore \quad \tan B_A = \sqrt{\left(\frac{qSC_L}{W}\right)^2 - 1.0} \qquad (7.2.153)$$

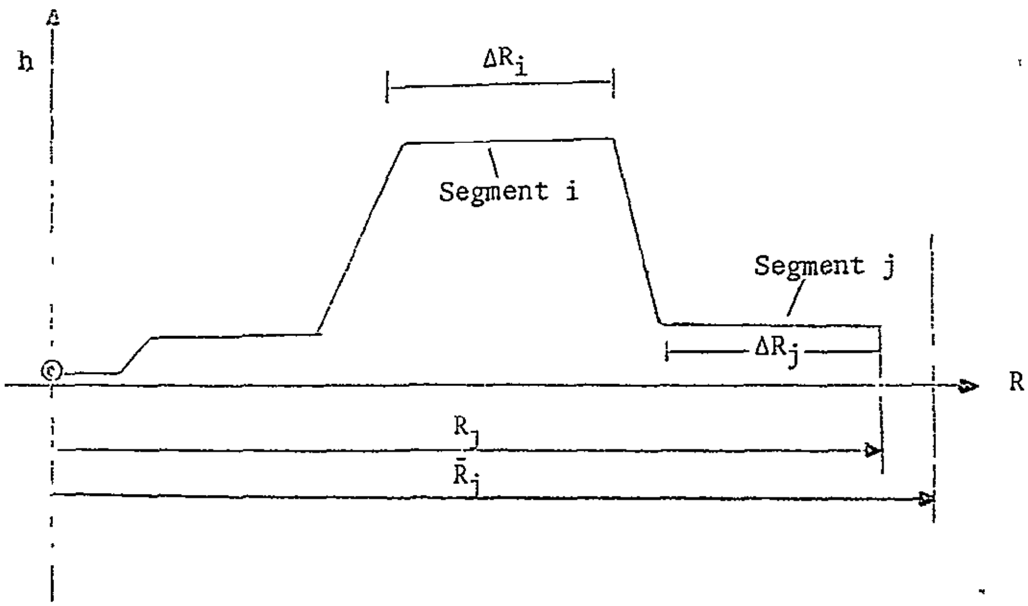
Substituting Equation (7.2.153) and (7.2.151)

$$R = \frac{V^2}{g}\sqrt{\frac{1.0}{\left(\frac{qSC_L}{W}\right)^2 - 1.0}} \qquad (7.2.154)$$

It should be noted that this mission segment option may employ directly specified value of Mach number, altitude, and weight or these state components may be picked up from the previous mission segment termination. The ability to reset Mach number, altitude, and weight from any previous segment termination is also available within the option.

7.2.6.23 Iteration to Fly a Specified Distance (Mission Option 15)

This mission segment option perturbs the range increment in segment i to provide a specified total range (from mission initiation) in segment j



This is illustrated above where ΔR_i is perturbed to satisfy the condition

$$R_j = \bar{R}_j$$

within an error of one nautical mile.

7.2.6.24 Climb or Accelerate (Mission Option 16)

This mission segment option provides a climb or acceleration between two Mach number-altitude points (M_1 h_1) and (M_2 h_2). These two flight conditions must be defined in two mission segments, segment i and segment j. The climb or acceleration will then join the two points Climb or acceleration paths may be performed in either a forward or reverse time direction. *Descents* are not permitted. The mission segment option may be performed with or without range credit.

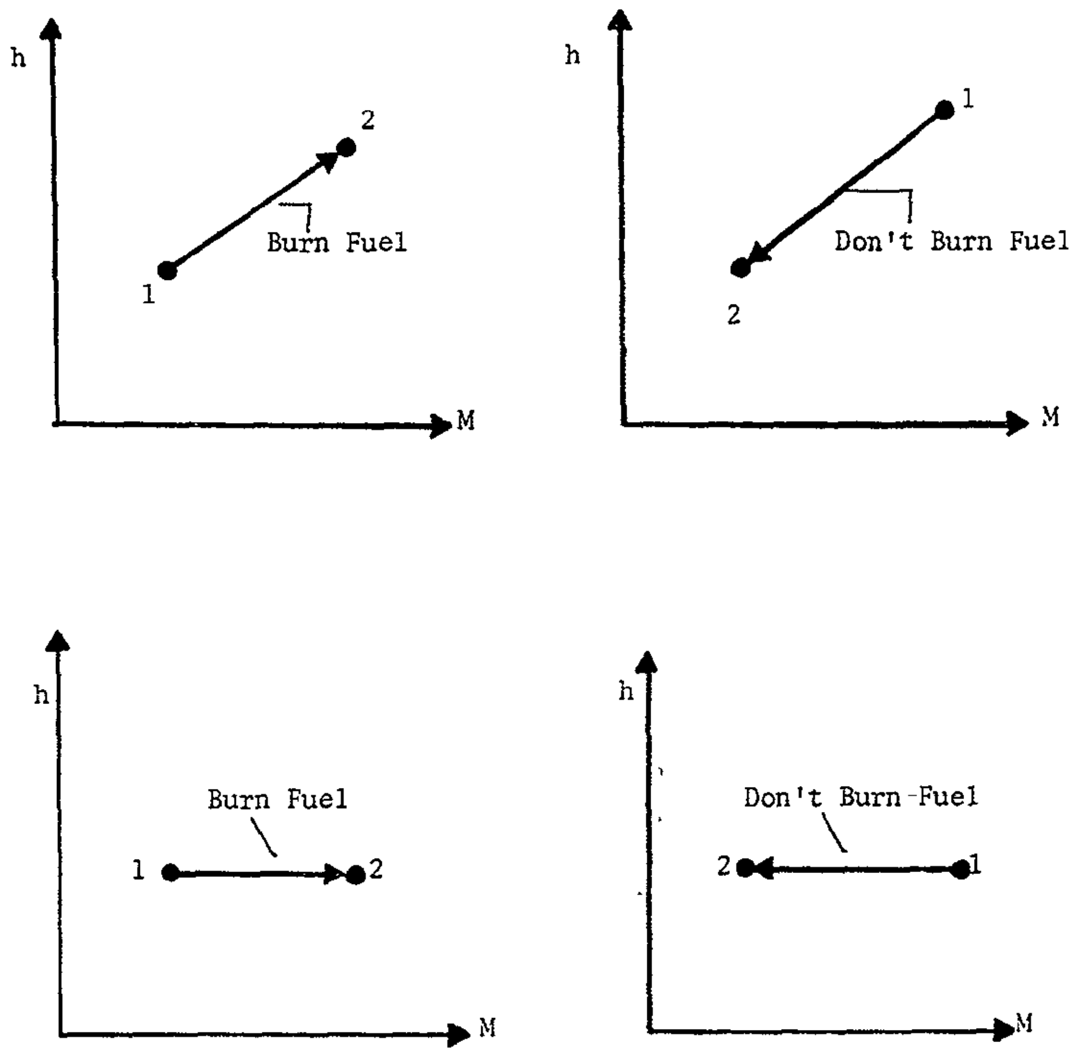
Fuel burning decisions are made according to Mil-C rules while going from *condition 1 to 2*. Thus, fuel is burned if

$$h_2 > h_1$$

or if

$$M_2 > M_1 \quad \text{and } h_2 = h_1$$

This behavior is illustrated below.



ORIGINAL PAGE IS
OF POOR QUALITY

7.2.6.25 Fuel Weight Change (Mission Option 17)

A computed or specified fuel weight change is introduced through this mission segment option. The operation performed is

$$W_{i+1} = W_i - \Delta W \qquad (7.2.155)$$

$$T_{i+1} = T_i + \Delta T \qquad (7.2.156)$$

The option can be used to compute

1. Loiter fuel requirements
2. Warm up and take-off fuel
3. Combat fuel

Take-off fuel when computed is carried out through program TOLAND of Section 7.1. If a detailed take-off analysis is not required the option of Section 7.2.6.26 is used. *Warm up* fuel calculation is computed for given power setting and time. *Loiter* fuel calculation is for flight at specified Mach number, altitude, weight, and a given time. *Combat* fuel calculation is for specified time or degrees of turn at a given load factor. If the degree of turn option is used, the following calculation is performed.

$$L = \bar{n} \cdot W \qquad (7.2.157)$$

where $\bar{n}$ is the load factor. The centrifugal force is

$$F_R = \sqrt{L^2 - W^2} \qquad (7.2.158)$$

and the turn radius is

$$R = \frac{WV^2}{g\sqrt{L2-W2}} \qquad (7.2.159)$$

The thrust force is set to drag at the turn C_L

$$T = D$$

7.2.6.26 Fuel Allowance (Mission Option 20)

This mission segment option computes the fuel allowance for a specified time at

1. Given power setting
2. Given thrust/weight

7.2.7 Thrust Specification in Mission Segment Options

The vehicle propulsive representations have been discussed in Section 7.2.2. There are three available maximum thrust tables, T_{maxj}; j = 1, 2, 3. These tables are referenced as follows:

T_{max1} = maximum dry thrust

T_{max2} = maximum wet thrust

T_{max3} = maximum power

Throttling may only be used for T_{max1} and T_{max2}. In using the various mission segment options an appropriate choice of thrust must be made. The options are

1. T = D
2. T = maximum dry
3. T = maximum wet
4. T = maximum power
5. T = thrust for given power setting; dry.

7.2.8 Flight Envelope Calculations

Several gross flight envelope calculations may be performed. All flight envelope computations are subject to the conditions

$C_L \leq C_{L_{lim}}$, lift coefficient limit

$M \leq M_{lim}$, Mach number limit

$q \leq q_{lim}$, dynamic pressure limit.

Propulsive and aerodynamic characteristics must be specified.

7.2.8.1 Climb Path History

Given an initial weight, warm up, and take-off fuel allowance, a maximum rate of climb path is performed from

$$P_1 = P_1(M_1, 0.0) \qquad (7.2.160)$$

ORIGINAL PAGE IS OF POOR QUALITY

to

$$P_2 = P_2(M_2, h_{MAX\ RF}) \qquad (7.2.161)$$

where

$$h_{MAX\ RF} = \text{altitude for best range factor at } M_2$$

Alternatively, P_2 may be selected as

$$P_2 = P_2(M_{MAX\ RF}, h_{MAX\ RF}) \qquad (7.2.162)$$

The calculation is performed in ten equal altitude increments from P_1 to P_2. Climb paths are generated for N distinct weights

$$W_i = W_o + i \cdot \Delta W; \quad i = 0, 1, \ldots, N-1 \qquad (7.2.163)$$

7.2.8.2 Endurance versus Weight at Various Altitudes

The endurance is calculated at a given altitude for the weights $W_i = W_o + i \cdot \Delta W$; $i = 1, 1, \ldots, N-1$. Mach number selected is for best endurance.

The calculation may be repeated for any number of altitudes, $h = h_o + i \cdot \Delta h$; $i = 0, 1, \ldots$.

7.2.8.3 Optimum Cruise Climb at Various Mach Numbers

An optimum cruise climb between W_1 and W_2 in a specified number of weight increments. The path is repeated for an array of Mach numbers and altitudes

$$M_i = M_o + i \cdot \Delta M; \quad i = 0, 1, 2, \ldots\ldots \qquad (7.2.164)$$

$$h_j = h_o + j \cdot \Delta h; \quad j = 0, 1, 2, \ldots\ldots \qquad (7.2.165)$$

7.2.9 Contour Presentation Capabilities

A set of point calculations (vehicle capability at given flight conditions) are carried out over a two-dimensional array of Mach-altitudes, M_i, h_j. The resulting matrix of capabilities, F^k_{ij}, is then supplied automatically to the CONPLOT routine of Reference 5, and the contours of the function F^k in the Mach-altitude plane are obtained in the form of CALCOMP, Houston plotter, or CRT display device output. At the present time twelve functions, F^1 to F^{12}, may be output in contour form. Each contour plot is described briefly below.

7.2.9.1 Specific Energy Time Derivative, $\dot{E}$, (INDMAP=1)

The specific energy time derivative is computed according to the expression

$$\dot{E}(M, h) - (T - D)V/W \tag{7.2.166}$$

where

$\dot{E}$ = energy total time derivative

T = thrust obtained at a specified power setting or at T = D; wet, dry, or maximum power options are available

D = drag computed for a specified load factor

V = flight velocity

W = aircraft weight

Some typical energy derivative contours for a large four-engine transport are presented in Figure 7.2-9. The minimum contour shown is for the condition T - D = 0. Hence, the flight envelope is a by-product of the $\dot{E}$ map when suitable constraints such as $C_{L_{max}}$, and dynamic pressure limits are added.

7.2.9.2 Specific Energy/Fuel Flow, $\dot{E}/\dot{m}$, (INDMAP=2)

The $\dot{E}/\dot{m}$ contour presents the specific energy time derivative over the fuel flow map. Since

$$\dot{E}/\dot{m} = \frac{dF/dt}{dm/dt} = \frac{dE}{dm} \tag{7.2.167}$$

The map illustrates an aircraft's ability to convert fuel into energy at specified flight conditions.

The point calculation employed is

$$\dot{E}/\dot{m} = (T - D)V/(W\dot{m}) \tag{7.2.168}$$

where $\dot{m}$ is the fuel flow rate. The various thrust and drag options discussed in Sections 7.2.1 and 7.2.2 may be employed to produce a family of maps. A typical example for the large subsonic transport at maximum thrust and 1g flight is shown in Figure 7.2-10.

7.2.9.3 Lift/Drag, L/D, (INDMAP=3)

Lift/drag contours present a measure of the airplane's aerodynamic efficiency. The L/D maps indicate its range capability in unpowered ~~flight~~ flight and partially reflect the cruise range capability. Mass can be produced for any specified load factor. A typical contour for the large subsonic transport in level flight is presented in Figure 7.2-11.

7.2.9.4 Range Factor, R_F, (INDMAP=4)

Range factor contours present a measure of vehicle cruise range capability. Maps are produced for level flight with thrust equal to drag at a specified aircraft weight.

$$R_F = \left(\frac{V}{SFC}\right)\left(\frac{L}{D}\right) \tag{7.2.169}$$

where SFC is the specific fuel consumption. The user may elect to construct maps for other than level unaccelerated flight. However, the interpretation of these charts is not clear. A typical unaccelerated flight range factor contour map for the large subsonic aircraft is presented in Figure 7.2-12.

7.2.9.5 Thrust (INDMAP=5)

The thrust map is available as a device for examining thrust input data or the thrust component of other mapped functions. The map can be obtained for wet, dry, maximum, or throttled power setting. The maximum power thrust map for the large subsonic transport is presented in Figure 7.2-13.

7.2.9.6 Drag Map (INDMAP=6)

The drag map provides a device for inspecting drag data input or the drag component of any other map. Drag maps are produced for a specified load factor. A 1g drag map for the large subsonic transport is presented in Figure 7.2-14.

7.2.9.7 Specific Fuel Consumption, SFC, (INDMAP=7)

Specific fuel consumption maps are provided as a data input inspection device or as an aid to visualizing the specific fuel consumption component of other maps. Maps may be obtained for wet, dry, maximum, or throttled power settings. Maximum power specific fuel consumption of the large subsonic transport is presented in Figure 7.2-15.

7.2.9.8 Fuel Flow Rate, $\dot{m}$, (INDMAP=8)

The fuel flow maps are provided as a data input inspection device or as an aid to visualizing the fuel flow component of other maps. Maps may be obtained for wet, dry, maximum or throttled power settings. Maximum power fuel flow for the large subsonic transport in level unaccelerated flight is presented in Figure 7.2-16.

7.2.9.9 Specific Energy (INDMAP=9)

The specific energy map

$$E = h + V^2/2g \tag{7.2.170}$$

is provided as a user's convenience in visualizing the trajectory points between constant energy lines and any other set of contours. An example is presented in Figure 7.2-17.

7.2.9.10 Lift/(Thrust - Drag), L/(T-D) (INDMAP=10)

The lift/(thrust - drag) contours are useful for determination of maximum range powered flight.

Assuming that maximum range flight occurs at small flight path angles

$$R \simeq \int \frac{L}{T-D}\, dE \tag{7.2.171}$$

Therefore, the energy-like approximation to maximum range flight occurs when L/(T-D) is a maximum at each energy level. It should be noted that when T - D = 0, no energy gain is possible, therefore, this singular condition must be avoided. In NSEG II the per cent excess of thrust over drag which is acceptable is a program input. A typical L/(T-D) map for the large subsonic transport is presented in Figure 7.2-18.

7.2.9.11 Turn Radius (INDMAP-11)

Turn radius maps give a gross indication of aircraft's combat capability. Turn radius is computed by equating the aircraft's lift capability in steady state of decelerating flight using the following expression

$$R = w \left(\frac{V^2}{g}\right) \left[\frac{1.0}{(qSC_L)^2 - W^2}\right]^{1/2} \qquad (7.2.172)$$

where C_L is determined so that (a) thrust equals drag for steady state flight and (b) C_L equals C_L maximum for minimum instantaneous turn radius.

Typical radius of turn map for the subsonic transport are presented in Figure 7.2-19.

7.2.9.12 Time to Turn (INDMAP=12)

Time to turn through 180 degrees is presented as a supplement to the turn radius map. When the minimum instantaneous turn radius calculation is employed, the maps do not give a true time to turn. They merely indicate how long a time the aircraft would take to turn *if it could maintain its current turn rate.* When steady state turns are considered, true time to turn is obtained *which will frequently be much longer than is required for a decelerating turn.* Typical time to turn maps for the subsonic transport are illustrated in Figure 7.2-20.

REFERENCES:

1. Leet, L. H., Contents and Capabilities of NSEG, Unpublished notes written at the Office for Development Planning, ASD, Wright-Patterson Air Force Base, October 1968, (available from Aerophysics Research Corporation).

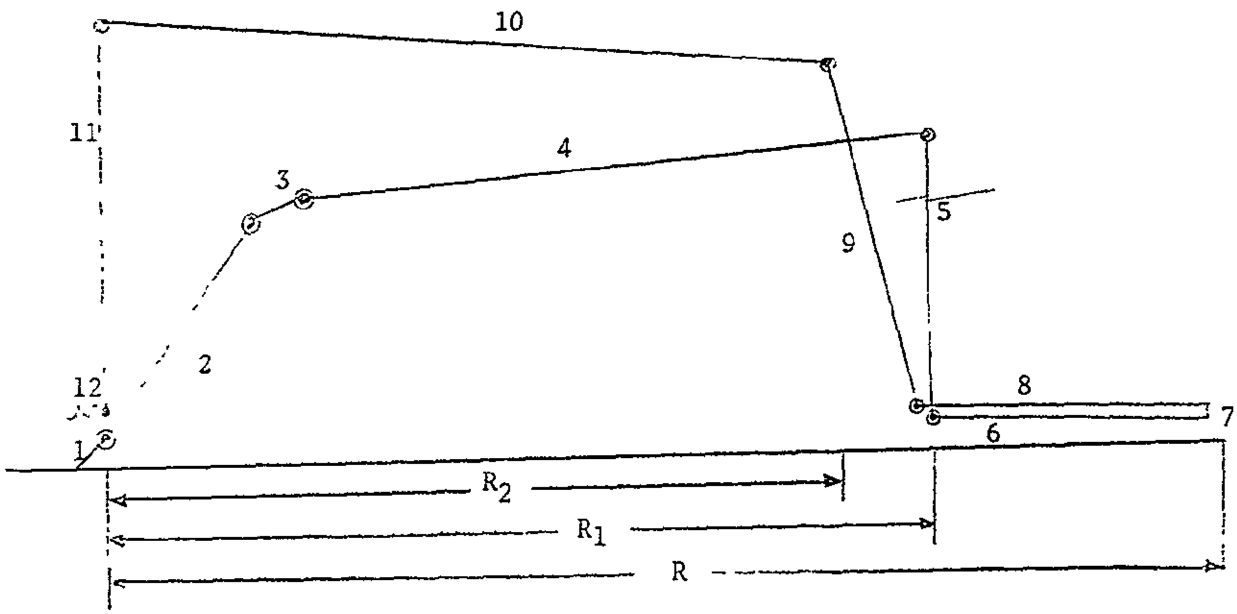
2. Rutowski, E. R., "Energy Approach to the General Aircraft Performance Problem," *Journal of the Aeronautical Sciences*, March 1954.

3. Raybould, B., "Approximate Trajectories for Maximum Range," *Journal of the Aeronautical Society*, June 1963.

4. Hague, D. S. and Jones, R. T., Energy Maneuverability Options for Mission Segment Analysis Program-NSEG, TN-149, Aerophysics Research Corporation, April 1972.

5. Hague, D. S., CONPLOT: A Rapid Code for Production of Three-Dimensional Contour Plots, TN-140, Aerophysics Research Corporation, March 1972.

Hi-Lo-Lo-Hi Radius Mission



1. Take-off
2. Maximum rate of climb to best cruise altitude given weight and Mach number
3. Constant C_L climb to best cruise altitude for new weight and Mach number
4. Breguet cruise to given range, R
5. Instantaneous state change to dash Mach number and altitude
6. Constant Mach number-altitude cruise to total range, R
7. Drop ordnance, instantaneous weight change
8. Constant Mach number-altitude return cruise to given weight
9. Maximum rate of climb to given Mach number-altitude
10. Breguet cruise to given weight
11. Instantaneous state change to best endurance Mach number for given altitude and weight
12. Loiter for given time

FIGURE 7.2-1. TYPICAL NSEG II MISSION PROFILE

ORIGINAL PAGE IS
OF POOR QUALITY

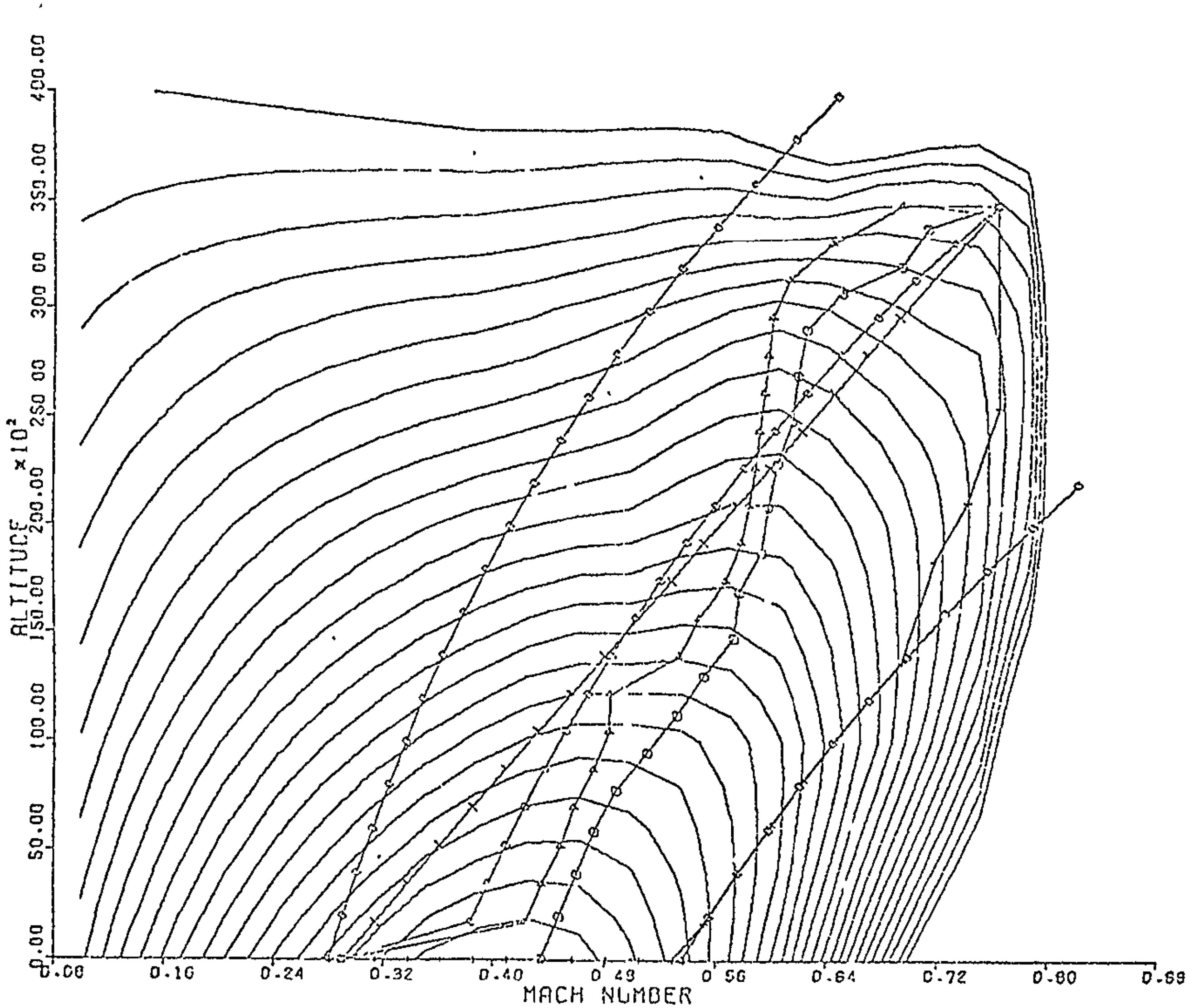


FIGURE 7.2-2. TYPICAL FLIGHT POINT PERFORMANCE MAPS
(SPECIFIC ENERGY AT VARIOUS LOAD FACTORS, WEIGHTS, AND THRUSTS)

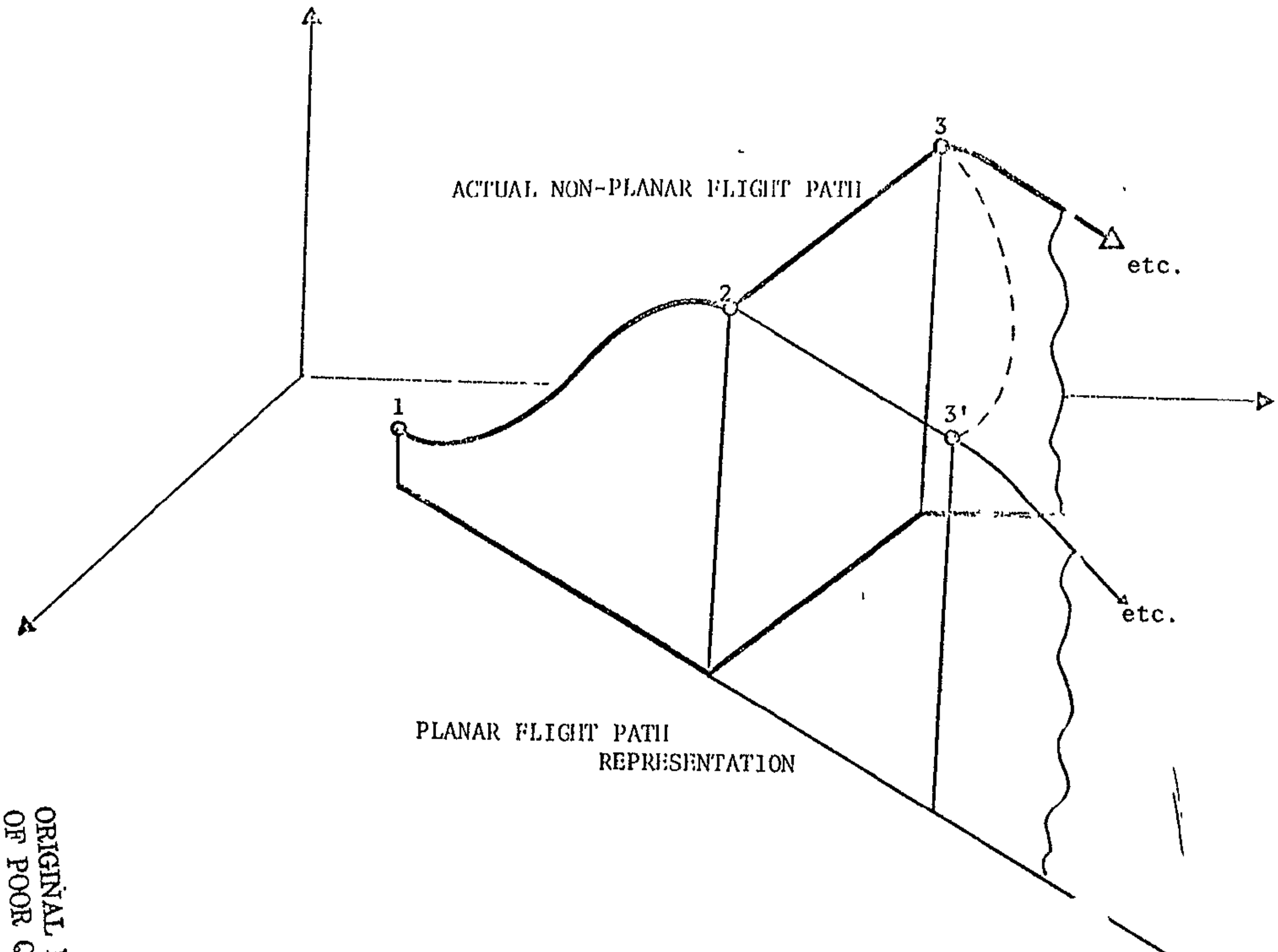


FIGURE 7.2-3 PLANAR REPRESENTATION OF THREE-DIMENSIONAL FLIGHT PATH

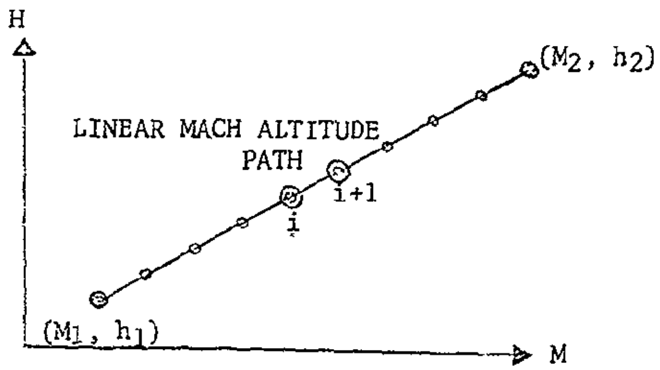


FIGURE 7.2-4 LINEAR MACH ALTITUDE MACH SEGMENT

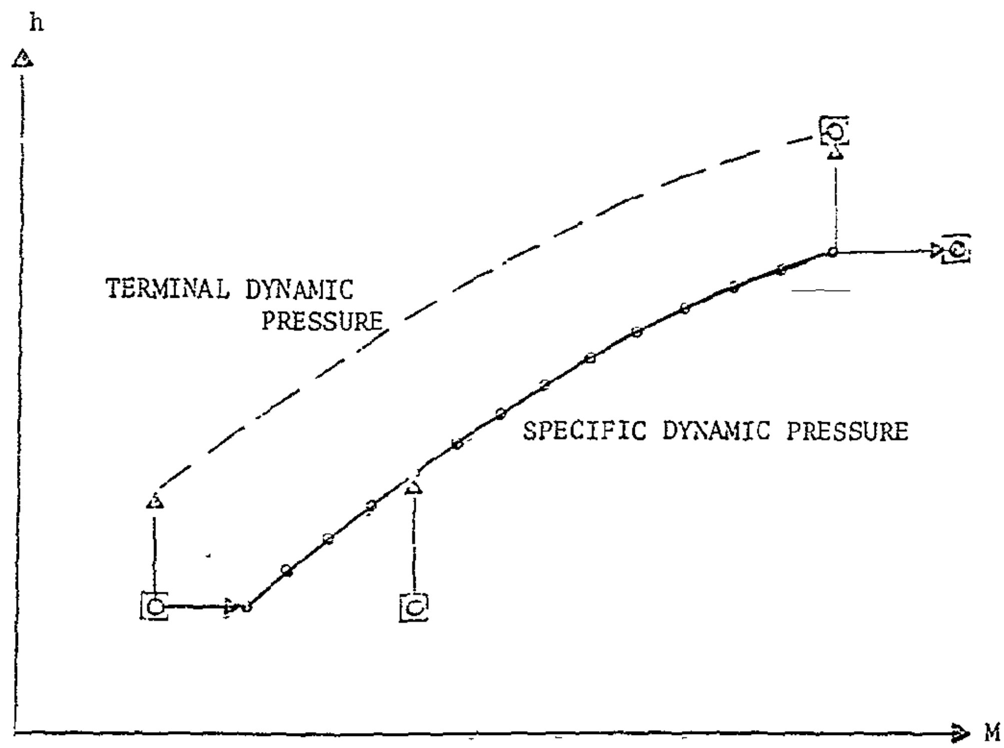


FIGURE 7.2-5 CONSTANT DYNAMIC PRESSURE SEGMENT

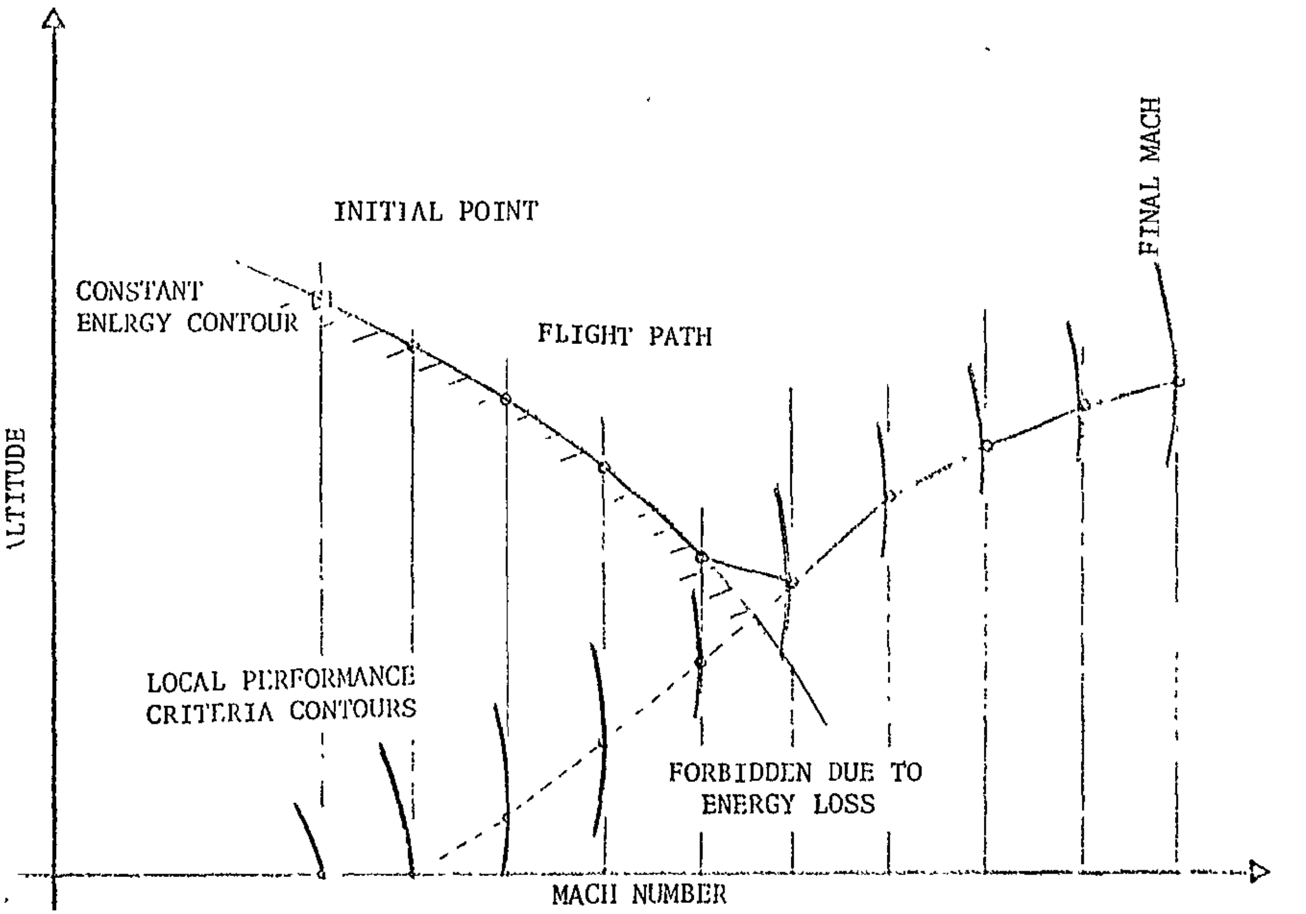


FIGURE 7.2-6. TYPICAL NSEG II ACCELERATION PATH

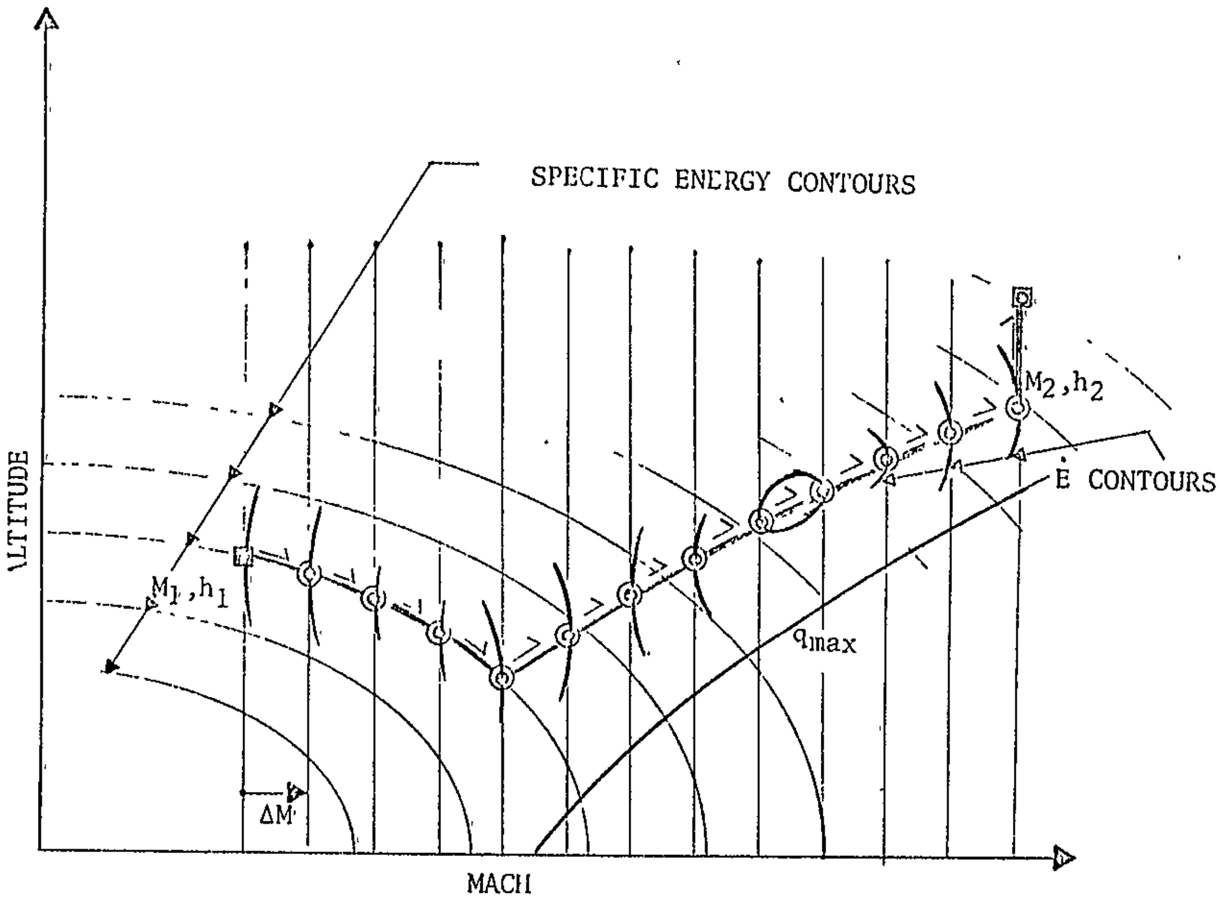


FIGURE 7 2-7 ENERGY CONSTRAINT IMPOSED ON MAXIMUM ACCELERATION PATH

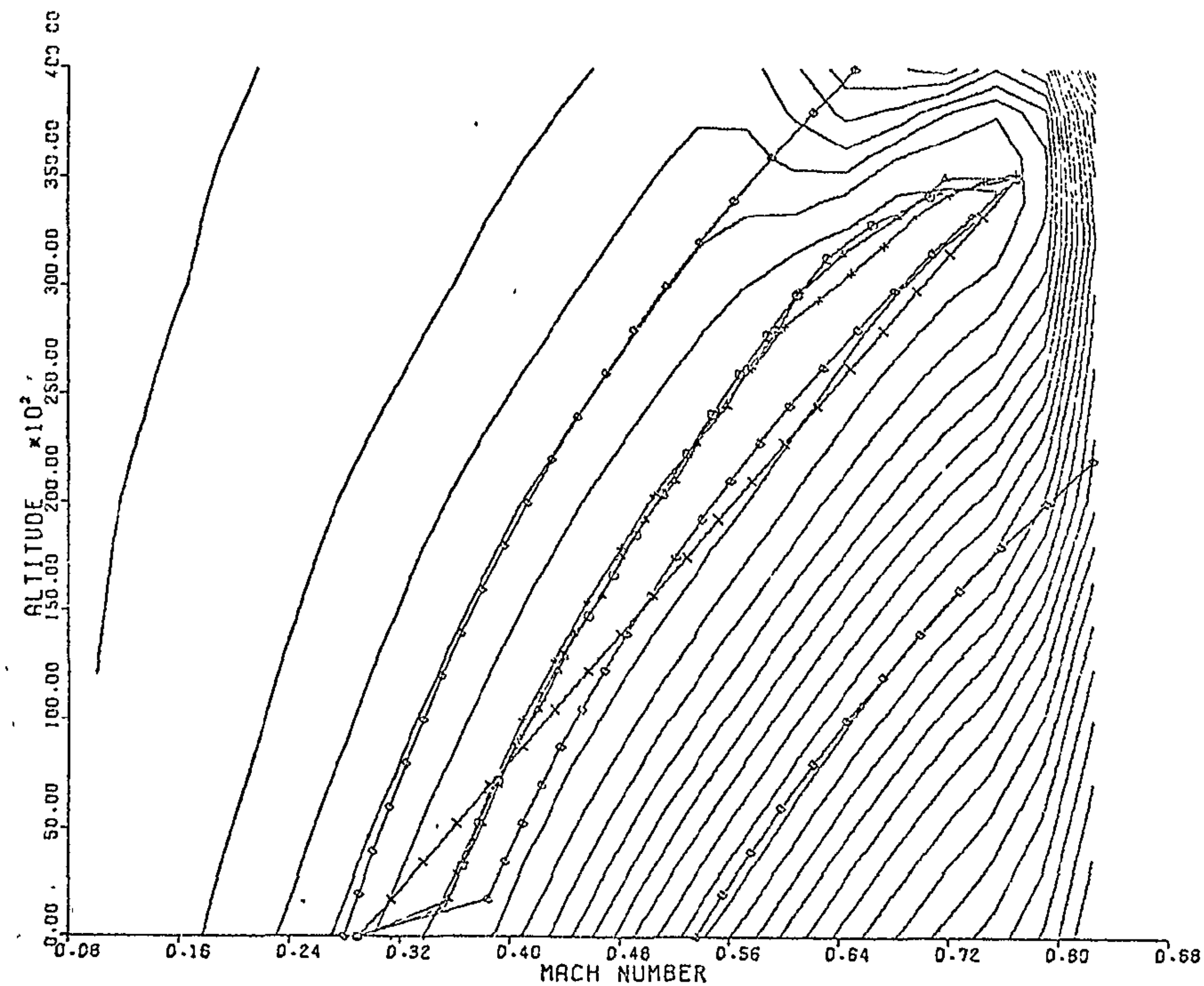


FIGURE 7.2-8. LIFT/DRAG MAP

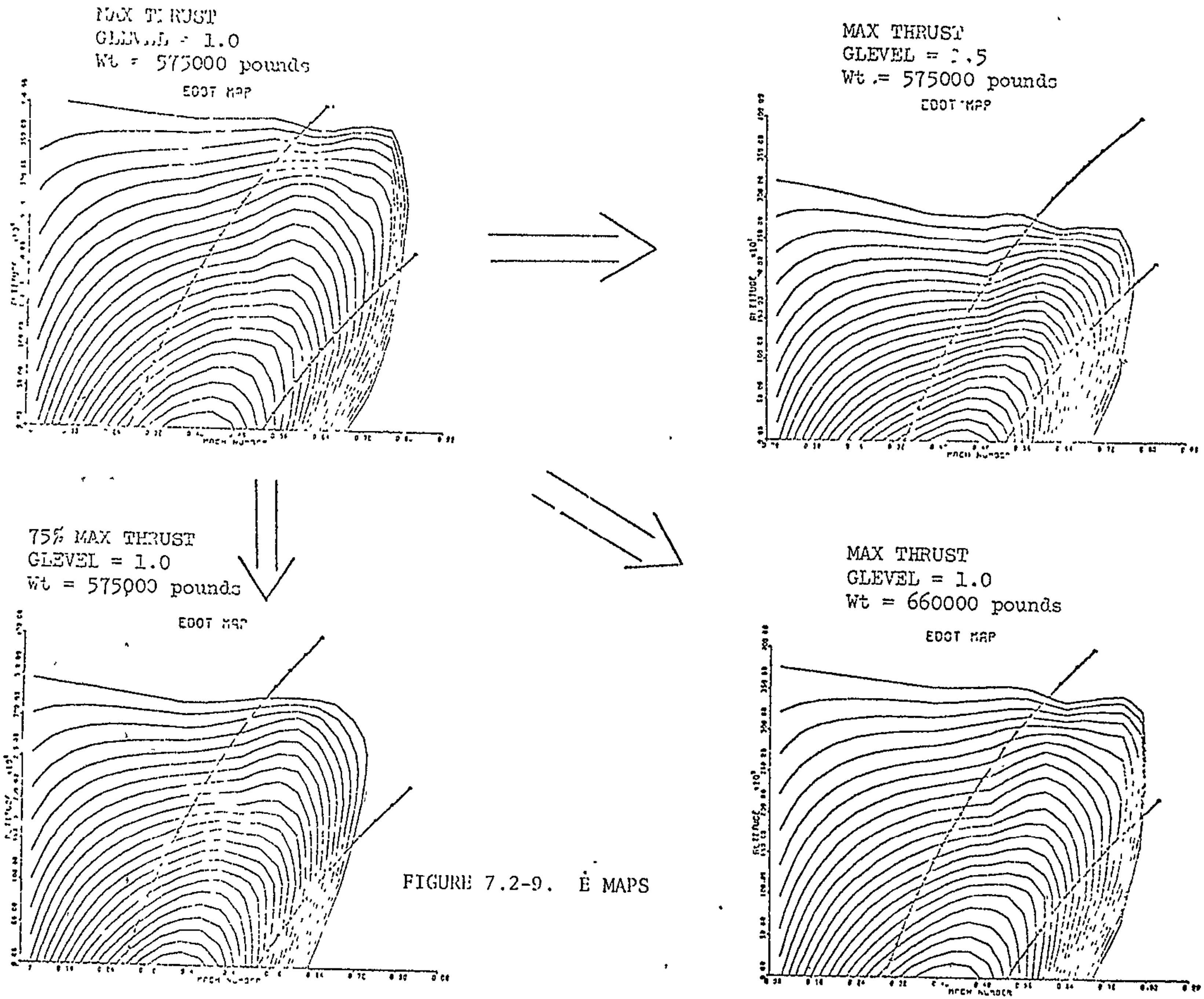


FIGURE 7.2-9. Ė MAPS

FIGURE 7.2-10

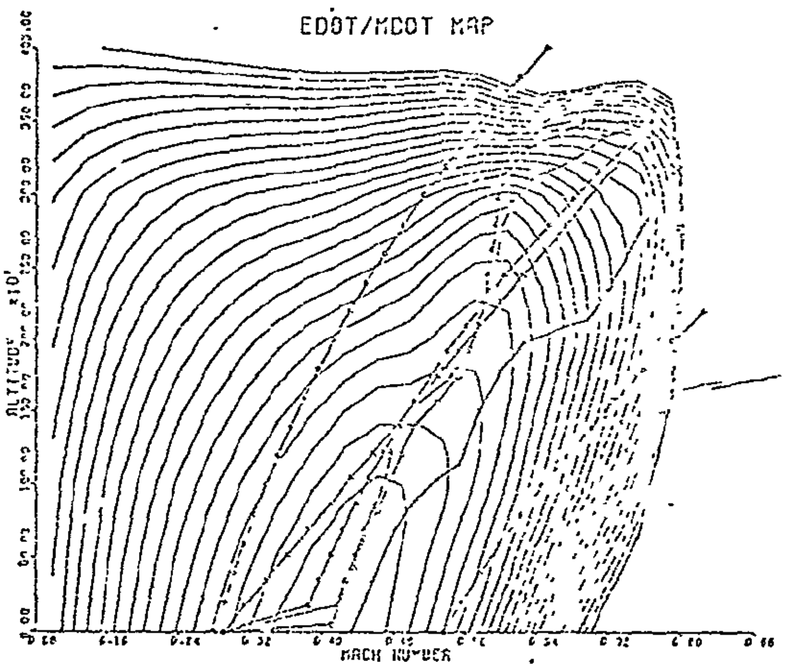


FIGURE 7.2-11

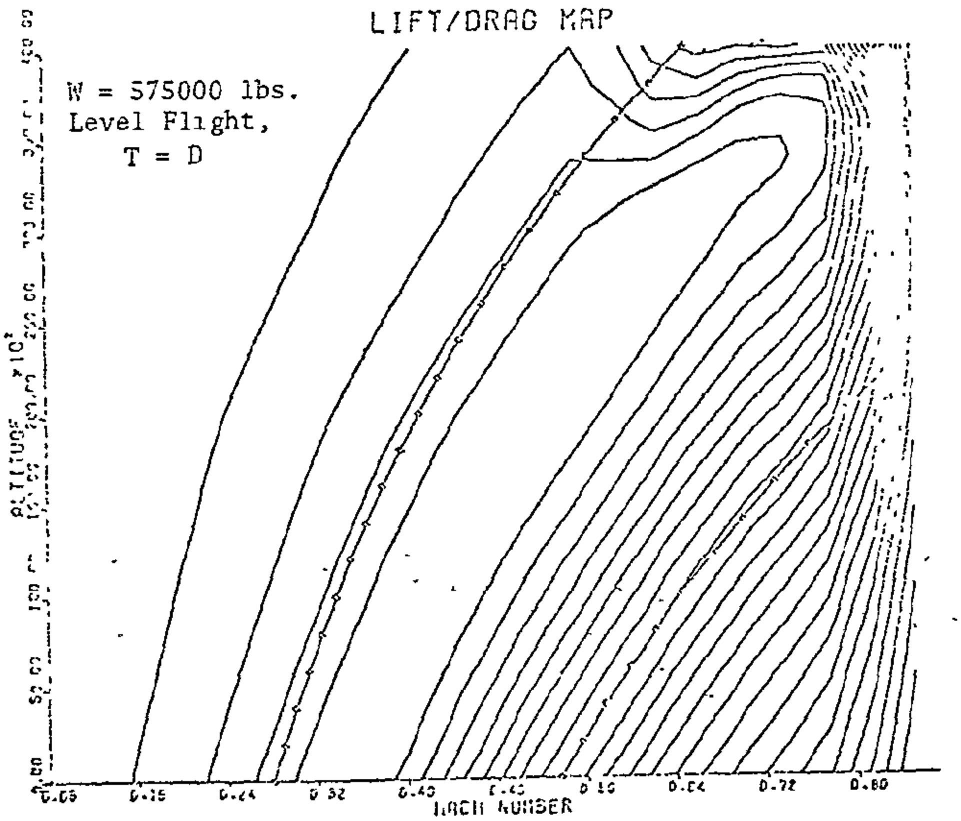


FIGURE 7.2-12.

RANGE FACTOR MAP

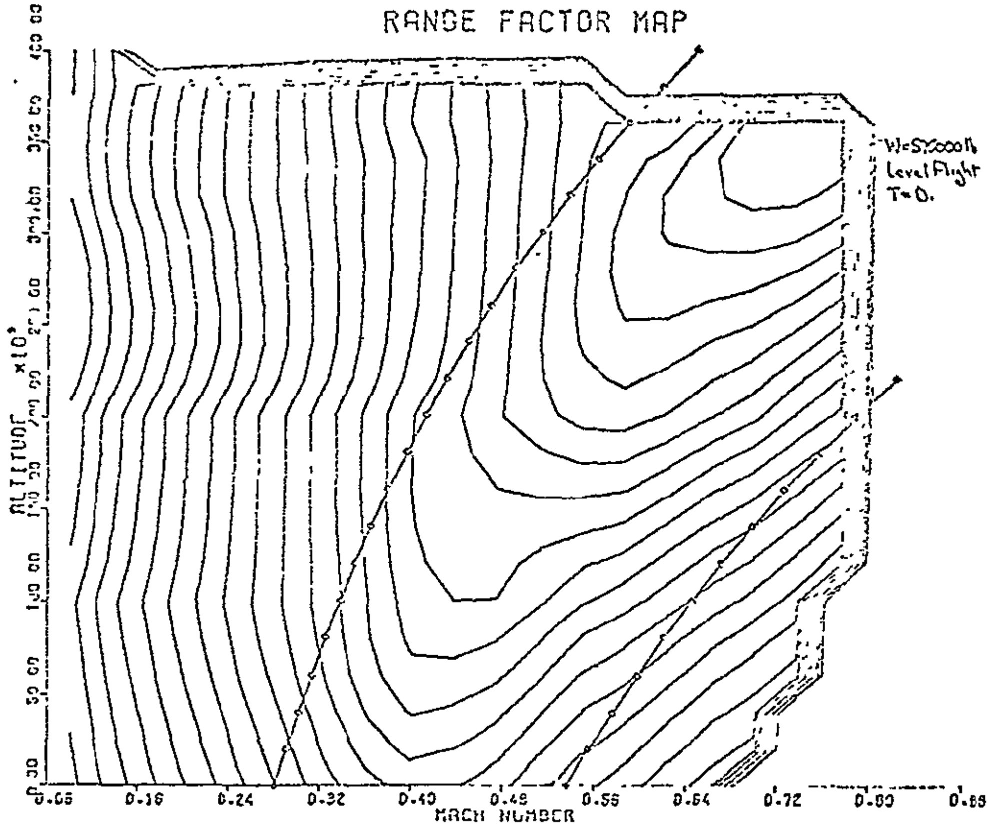


FIGURE 7.2-13

THRUST MAP

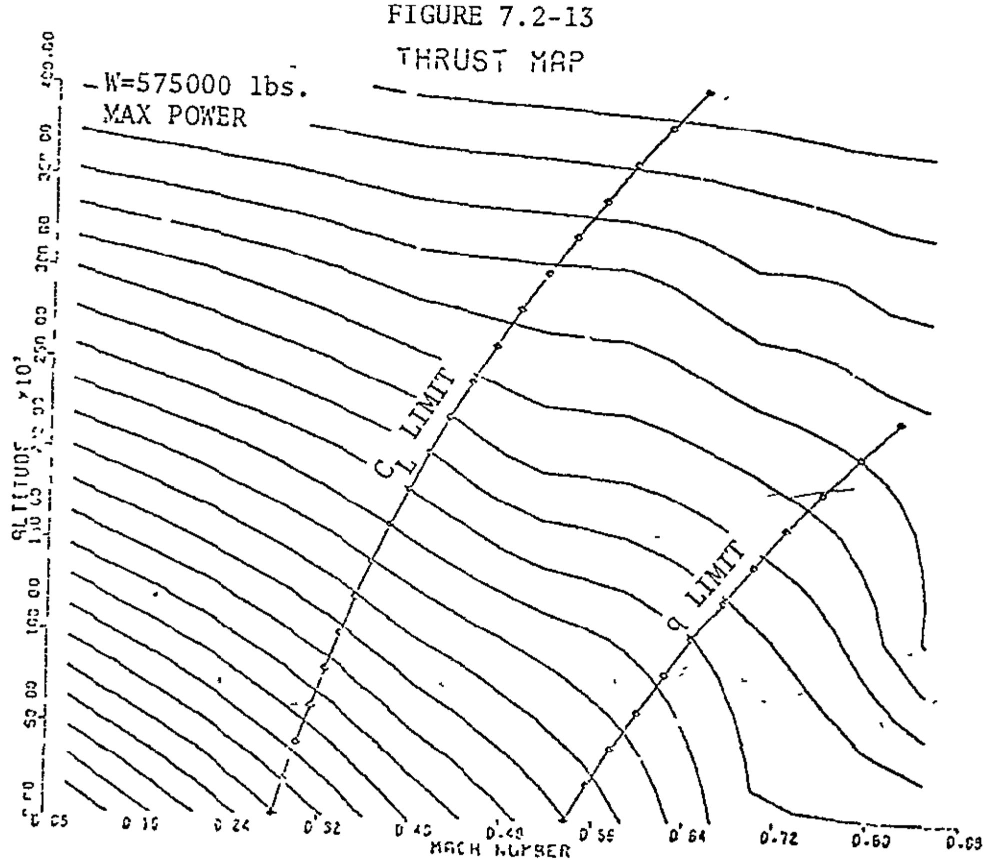


FIGURE 7.2-14. DRAG MAP

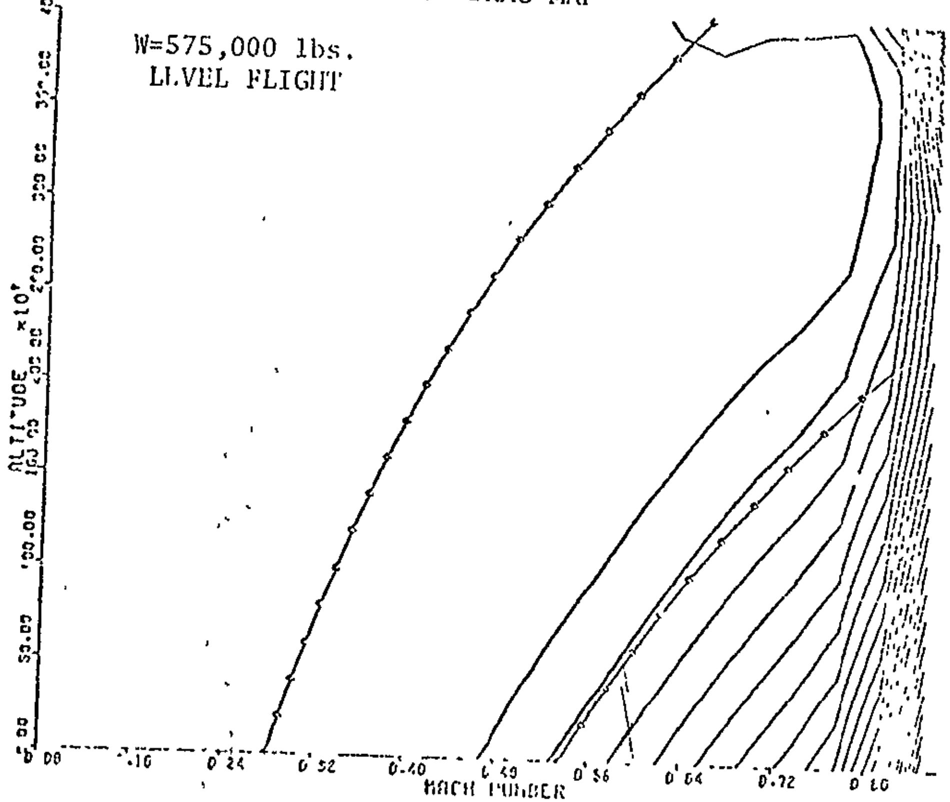


FIGURE 7.2-15. SPECIFIC FUEL CONSUMPTION MAP

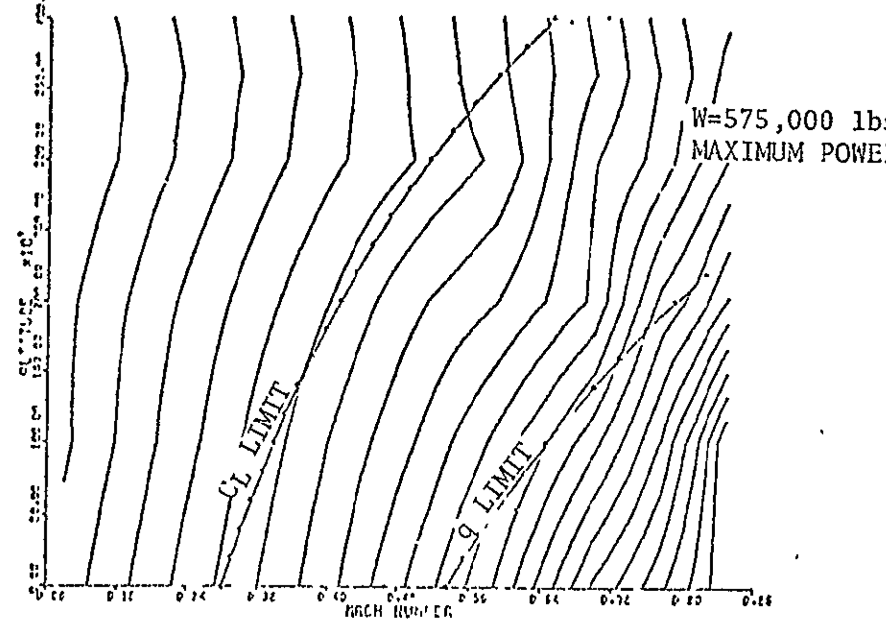
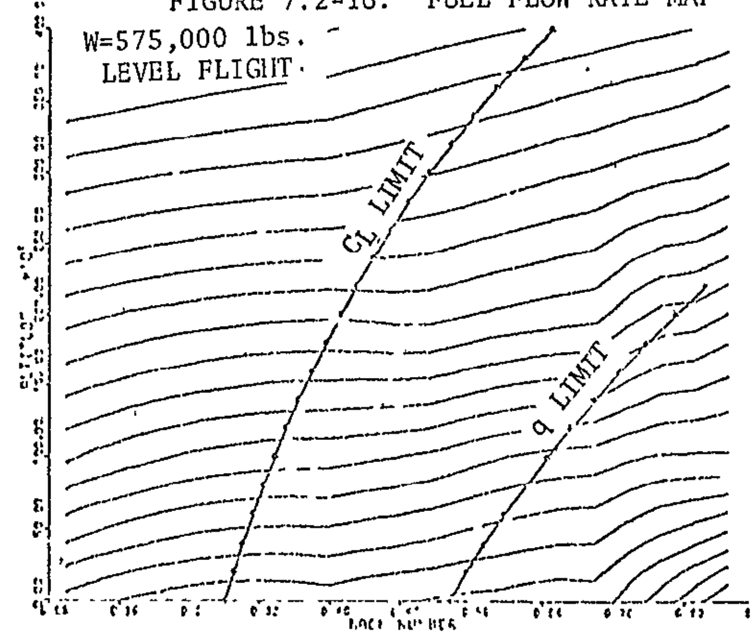


FIGURE 7.2-16. FUEL FLOW RATE MAP



ORIGINAL PAGE IS
OF POOR QUALITY

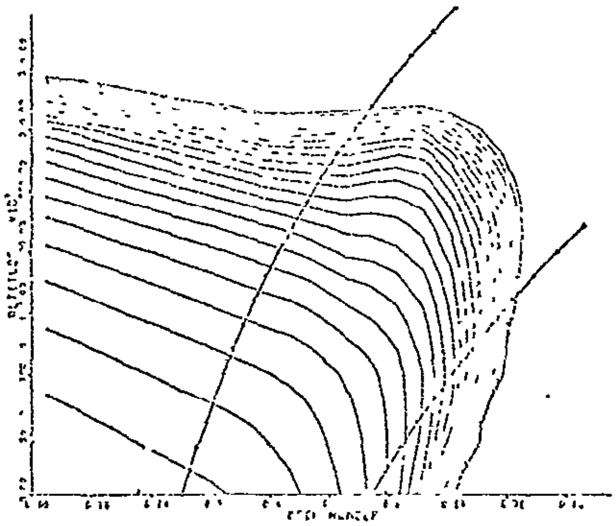


FIGURE 7.2-17 SPECIFIC ENERGY MAP

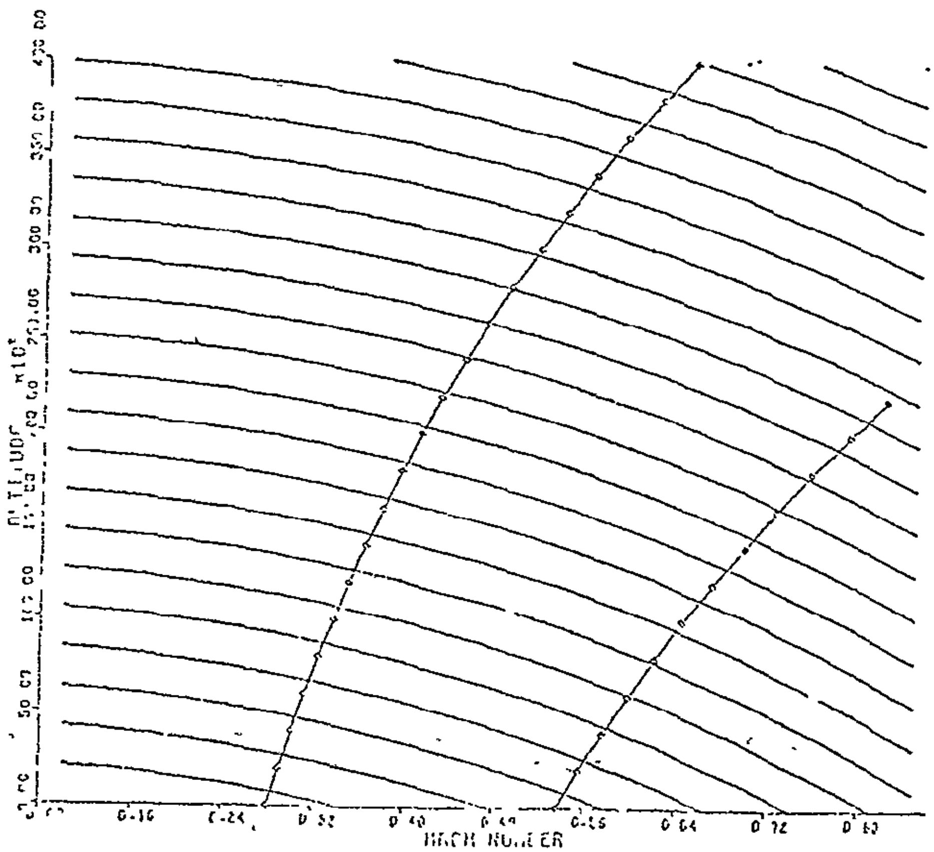


FIGURE 7.2-18 LIFT/(THRUST-DRAG)

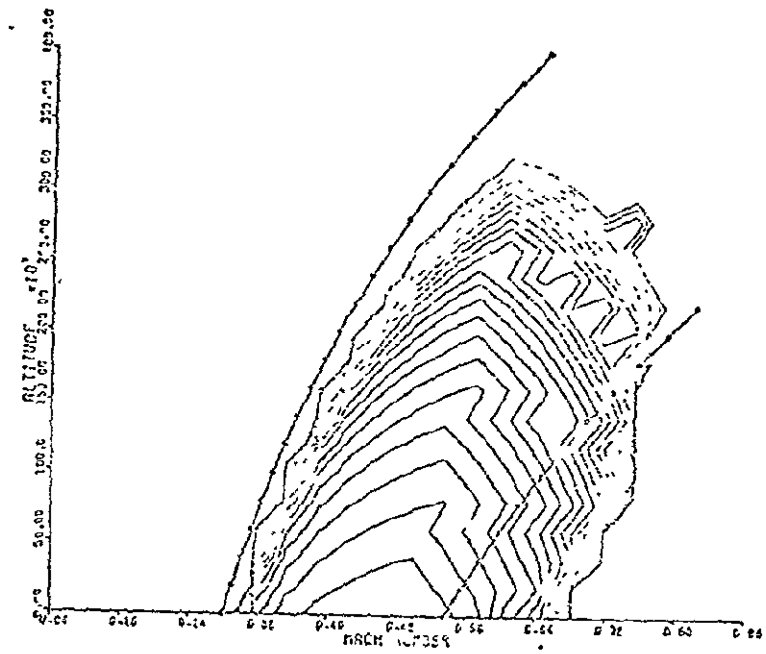


FIGURE 7.2-19 RADIUS OF TURN MAP

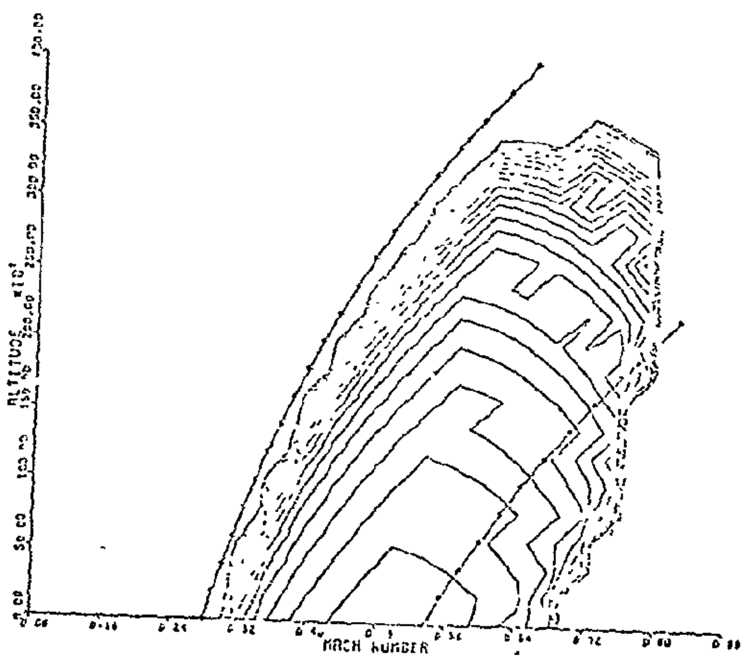


FIGURE 7.2-20 MAP OF TIME TO TURN 180 DEGREES

TABLE OF CONTENTS FOR SECTION 7.3, PROGRAM ATOP II

7.3 ATOP II: ATMOSPHERIC TRAJECTORY OPTIMIZATION PROGRAM

Trajectory optimization by the steepest-descent method is now a routine performance estimation at several Government research establishments and major aerospace concerns. The computer program utilized for trajectory optimization studies in this report is capable of determining optimal three-dimensional flight paths for a wide variety of vehicles in the vicinity of a single planet. Atmospheric effects may be included, if desired. Past program applications include flight path optimization of

a. high performance supersonic aircraft
b. spacecraft orbital transfer rendezvous and re-entry
c. multi-stage booster ascent trajectories
d. boost-glide re-entry vehicles
e. advanced hypersonic cruise aircraft
f. air-to-ground missiles

Optimal control can be determined for any combination of the time varying variables

a. angle of attack (or pitch angle)
b. bank angle
c. side slip
d. throttle
e. two thrust orientation angles

All the commonly employed terminal performance and constraint criteria may be specified. Inequality constraints may be imposed along the vehicle flight path.

Several options are available for specification of vehicle aerodynamic and propulsive options. Data and vehicle characteristics option can be modified at preselected stage points. An arbitrary number of stage points may be specified.

Planetary characteristics are nominally set to those of the earth. Up to four gravitational harmonics may be specified. Nominal planetary atmosphere employed is the 1959 ARDC. A variety of wind specification options are available. An ellipsoidal planetary shape may be specified.

The original trajectory optimization program is described in References 1 and 2. Equations of motion employed are described in References 3 and 4. Some past applications are described in References 5 and 6. An extension of program capability is described in Reference 7. An extension to simultaneously determine both optimal time varying control and discrete stage points together with some applications are described in References 8 and 9. A guidance and control application, the so called lambda guidance scheme, is reported in Reference 10.

The optimization program of References 1 and 2 employs a second-order prediction scheme and several control variable "weighting matrix" options to assist convergence of the steepest-descent algorithm. These two features have also been included in a recently developed trajectory optimization, Reference 11. They are also retained as convergence options in an extended version of the References 1 and 2 program which has multiple arc (branched trajectory) capability as reported in Reference 12.

The remainder of this section is devoted to an outline of the three-degree-of-freedom equations used in ATOP II. The variational optimization formulation employed in ATOP II is described in Section 10.1. It should be noted that the ATOP II program also contains a multivariable optimization capability for applications in which the time varying control can be parameterized. **Examples of this approach are contained in Reference 13. The multivariable search capability is described in Section 10.2.**

7.3.1 Point Mass Trajectory Equations

Several suitable coordinate systems are available for point mass trajectory computations. The basic set of coordinates used in the present analysis is a rectangular set rotating with the earth, (X_e, Y_e, Z_e). This coordinate system is illustrated in Figure 7.3-1.

The X_e and Y_e axes lie in the equatorial plane, the positive X_e axis being initially chosen as the intersection of this plane with the vehicle longitudinal plane at $t = t_o$. Y_e is 90 degrees to the west of X_e, and Z_e is positive through the South Pole. The radius vector magnitude from the center of the earth to the vehicle is given by

$$|R| = \sqrt{X_e^2 + Y_e^2 + Z_e^2} \tag{7.3.1}$$

The angle between R and the North pole is given by

$$\varphi' = 90 - \phi_L \tag{7.3.2}$$

where ϕ_L is the latitude of the vehicle.

As a result of the earth's rotation, an observer in the (X_e, Y_e, Z_e) system would detect an apparent motion of the point mass. In the rotating system Newton's law can be written in the vector form, References 1 and 2,

$$\frac{F}{m} = \left(\frac{d^2r}{dt^2}\right)_e + 2\omega_p \times \left(\frac{dr}{dt}\right)_e + \omega_p \times \omega_p \times R \tag{7.3.3}$$

where F is the total force acting on the vehicle; m is the vehicle mass, and ω is the planet's rotation rate. This vector equation can be expressed in component form using the relationships

$$\frac{F_{x_e}}{m} = \ddot{X}_e + 2\omega_p\dot{Y}_e - \omega_p^2 X_e \tag{7.3.4}$$

$$\frac{F_{y_e}}{m} = \ddot{Y}_e - 2\omega_p\dot{X}_e - \omega_p^2 Y_e \tag{7.3.5}$$

$$\frac{F_{z_e}}{m} = \ddot{Z}_e \tag{7.3.6}$$

Alternatively, by introducing the vehicle velocity components, Equations (7.3.4) to (7.3.6) can be reduced to the first order form

$$\dot{X}_e = u_e \tag{7.3.7}$$

$$\dot{Y}_e = v_e \tag{7.3.8}$$

$$\dot{Z}_e = w_e \tag{7.3.9}$$

$$\dot{u}_e = \frac{F_{x_e}}{m} - 2\omega_p v_e + \omega_p^2 X_e \tag{7.3.10}$$

$$\dot{v}_e = \frac{F_{y_e}}{m} + 2\omega_p u_e + \omega_p^2 Y_e \tag{7.3.11}$$

$$\dot{w}_e = \frac{F_{z_e}}{m} \tag{7.3.12}$$

The vehicle state equations are completed by adding the mass rate of change equation to the equations of motion. The mass rate of change is assumed to be of the general form

$$\dot{m} = \dot{m}(\bar{x}_n(t), \bar{\alpha}_m(t), t) \tag{7.3.13}$$

where $\bar{x}_n(t)$ is the time varying vehicle state vector having components X_e, Y_e, Z_e, u_e, v_e, w_e, and m; and $\bar{\alpha}(t)$ is the time varying control vector having the components discussed in the following section.

7.3.2 Control Variables

The total force acting on the vehicle has three distinct sources: aerodynamic force as a result of interaction between the vehicle surfaces and the planetary atmosphere; second, gravitational force as a result of vehicle and planetary mass interaction; and finally, thrust forces introduced by the vehicle propulsion system.

Aerodynamic force components in the basic (X_e, Y_e, Z_e) rotating coordinate system are functions of the vehicle orientation with respect to the velocity vector. Three angular control variables determine these force components as discussed below.

Angle of attack, α, is the angle between the velocity vector and the vehicle reference axis when viewed in the vehicle side elevation. That is, in a rectangular body axis coordinate system, x,y,z with x along the vehicle reference axis, positive forward, y perpendicular to the vehicle plane of symmetry, positive to starboard, and z completing a right hand system. A view normal to the x-z plane is considered. If u, v, w are the components of the vehicle velocity with respect to the atmosphere in this body axis system

$$\alpha = \tan^{-1} \left(\frac{w}{u}\right) \tag{7.3.14}$$

Sideslip angle, β, is the angle between the velocity vector and the reference axis when looking down on the vehicle planform, that is, along the z axis. In this case,

$$\beta = \tan^{-1} \left(\frac{v}{u}\right) \tag{7.3.15}$$

The third angle required to establish vehicle orientation in space is a rotation about the velocity vector. This angle, bank angle (B_A) is taken as zero when the vehicle plane of symmetry is vertical and when the vehicle is upright. Positive bank angle is a positive rotation about the velocity vector as in Figure 7.3-4. With the above three angles used to describe vehicle attitude, velocity vector known, and a given atmosphere, the aerodynamic forces are completely satisfied.

Thrust from the propulsion system involves the atmospheric properties either due to the atmospheric back pressure degrading the vacuum thrust or by virtue of the atmospheric fluid used in the combustion process which creates thrust. The propulsion unit efficiency may be affected by Mach number and, hence, velocity so that thrust forces depend on the state variable components of position and velocity. If the propulsion system force has a fixed orientation along the x body axis, the control variables introduced to describe aerodynamic forces suffice to describe thrust forces also. It may be, however, that the propulsion unit has a fixed or variable orientation within the vehicle. In this case, additional control variables describe the relative position of the propulsion unit force with respect to the body axes.

Two additional angles are sufficient to orient the thrust. These are the cone angle from the reference axis, λ_T, and the inclination about the reference axis, ϕ_T. This latter angle is measured positively about the reference axis and is zero when the thrust force is perpendicular to the port side of the vehicle plane of symmetry, as illustrated in Figure 7.3-5.

One other control variable for thrust must be specified; this is the throttle setting, N, which serves to determine the propulsion unit power setting on variable thrust engines.

In all, then, to specify the forces acting on a point mass vehicle with a single propulsion unit, six control variables, α, β, B_A, λ_T, ϕ_T, and N are required. If there is more than one independently controllable propulsion unit, additional control variables, λ_{T_i}, ϕ_{T_i}, and N_i, are defined.

7.3.3 Coordinates and Coordinate Transformations

7.3.3.1 Local Geocentric-Horizon Coordinates

Components of the planet-referenced acceleration are integrated to obtain the planet-referenced velocity components ($\dot{X}_e$, $\dot{Y}_e$, $\dot{Z}_e$). Vehicle position in this coordinate system is determined by integration of these velocities. Vehicle position in the planet-referenced spherical coordinate system will now be determined. The spherical coordinates are longitude, geocentric latitude, and distance from the center of the planet. Angle "C" represents the change in vehicle longitude and may be written

$$C = \theta_{L_0} - \theta_L \tag{7.3.16}$$

Angle C is related to the vehicle position by the expression

$$C = \mathrm{Tan}^{-1} \left(\frac{Y_e}{X_e}\right) \tag{7.3.17}$$

The relationships are illustrated in Figure 7.3-6.

To describe body motion relative to the planet, a local-geocentric-horizon coordinate system is employed. The Z_g axis of this system is along a radial line passing through the body center of gravity and is positive toward the center planet. The X_g axis of this system is normal to the Z_g axis and is positive northward; Y_g forms a right handed system. Figure 7.3-6 shows the relation of this coordinate system to the other systems employed.

To locate the X_g-Y_g-Z_g axes with respect to the X_e-Y_e-Z_e axes, rotate about Z_e by an angle $(180° \mp C)$, then rotate about Y_g through the angle $(90° - \phi_L)$. The complete transformation can be reduced to the single transformation matrix

$$\begin{vmatrix} \bar{1}_{X_g} \\ \bar{1}_{Y_g} \\ \bar{1}_{Z_g} \end{vmatrix} = \begin{vmatrix} -\mathrm{Sin}\,\phi_L\,\mathrm{Cos}\,C & -\mathrm{Sin}\,\phi_L\,\mathrm{Sin}\,C & -\mathrm{Cos}\,\phi_L \\ \mathrm{Sin}\,C & -\mathrm{Cos}\,C & 0 \\ -\mathrm{Cos}\,\phi_L\,\mathrm{Cos}\,C & -\mathrm{Cos}\,\phi_L\,\mathrm{Sin}\,C & \mathrm{Sin}\,\phi_L \end{vmatrix} \begin{vmatrix} \bar{1}_{X_e} \\ \bar{1}_{Y_e} \\ \bar{1}_{Z_e} \end{vmatrix} \tag{7.3.18}$$

which defines a direction cosine set (i, j, k) by the equation

$$\begin{vmatrix} \bar{1}_{X_g} \\ \bar{1}_{Y_g} \\ \bar{1}_{Z_g} \end{vmatrix} = \begin{vmatrix} i_1 & j_1 & k_1 \\ i_2 & j_2 & k_2 \\ i_3 & j_3 & k_3 \end{vmatrix} \begin{vmatrix} \bar{1}_{X_e} \\ \bar{1}_{Y_e} \\ \bar{1}_{Z_e} \end{vmatrix} \tag{7.3.19}$$

Planet referenced velocity in the local-geocentric coordinate system is given by

$$\begin{vmatrix} \dot{X}_g \\ \dot{Y}_g \\ \dot{Z}_g \end{vmatrix} = \begin{vmatrix} i_1 & j_1 & k_1 \\ i_2 & j_2 & k_2 \\ i_3 & j_3 & k_3 \end{vmatrix} \cdot \begin{vmatrix} \dot{X}_e \\ \dot{Y}_e \\ \dot{Z}_e \end{vmatrix} \tag{7.3.20}$$

and

$$V_g = \sqrt{\dot{X}^2_g + \dot{Y}^2_g + \dot{Z}^2_g}. \tag{7.3.21}$$

Flight path angles are computed by

$$\sigma = \tan^{-1} \left(\frac{\dot{Y}_g}{\dot{X}_g}\right) \tag{7.3.22}$$

and

$$\gamma = \sin^{-1} \left(\frac{-\dot{Z}_g}{V_g}\right) \tag{7.3.23}$$

Here σ is the heading angle, and λ is the flight path angle.

7.3.3.2 Wind Axis Coordinates

Aerodynamic and thrust forces for point mass problems are conveniently summed in a wind-axis coordinate system (X_A, Y_A, Z_A). The equations of motion are solved in (X_e, Y_e, Z_e) coordinate system; the wind-axis components of force must therefore be resolved into this basic coordinate system.

When winds are defined by atmospheric velocity components along the local geocentric axes, vehicle velocity relative to the atmosphere is the vector difference of vehicle geocentric velocity and wind velocity. The wind axis system is then determined by the vehicle airspeed, V_A, and the flight path angles relative to the atmosphere λ_A and σ_A. If wind velocity is zero, $V_A = V_g$, $\lambda_A = \lambda$ and $\sigma_A = \sigma$. If there is a wind, with velocity components (λ_{gw}, $\dot{Y}_{gw}$, Z_{gw}), then

$$V_A = \sqrt{(\dot{X}_g-\dot{X}_{gw})^2 + (\dot{Y}_g-\dot{Y}_{gw})^2 + (\dot{Z}_g-\dot{Z}_{gw})^2} \tag{7.3.24}$$

$$\gamma_A = \sin^{-1} \left[-(\dot{X}_g-\dot{X}_{gw})/V_A\right]. \tag{7.3.25}$$

$$\sigma_A = \tan^{-1} \left[(\dot{Y}_g-\dot{Y}_{gw})/(\dot{X}_g-\dot{X}_{gw})\right] \tag{7.3.26}$$

Forces are first resolved from wind axes to the local geocentric coordinates. The wind axes are defined relative to the local geocentric axes by three angles: heading, σ_A; flight path attitude, γ_A, (defined above); and bank angle, B_A , Figure 7.3-9.

The complete transformation from local geocentric horizon coordinates to wind axes is

$$\begin{vmatrix} X_A \\ Y_A \\ Z_A \end{vmatrix} = \begin{vmatrix} \cos\gamma_A\cos\sigma_A & \cos\gamma_A\sin\sigma_A & -\sin\gamma_A \\ -\sin\sigma_A\cos B_A + \sin\gamma_A\cos\sigma_A\sin B_A & \cos\sigma_A\cos B_A + \sin\gamma_A\sin\sigma_A\sin B_A & \cos\gamma_A\sin B_A \\ \sin\sigma_A\sin B_A + \sin\gamma_A\cos\sigma_A\cos B_A & -\cos\sigma_A\sin B_A + \sin\gamma_A\sin\sigma_A\cos B_A & \cos\gamma_A\cos B_A \end{vmatrix} \begin{vmatrix} X_g \\ Y_g \\ Z_g \end{vmatrix} \quad (7.3.27)$$

which defines a direction cosine set

$$\begin{vmatrix} X_A \\ Y_A \\ Z_A \end{vmatrix} = \begin{vmatrix} r_1 & s_1 & t_1 \\ r_2 & s_2 & t_2 \\ r_3 & s_3 & t_3 \end{vmatrix} \begin{vmatrix} X_g \\ Y_g \\ Z_g \end{vmatrix} \quad (7.3.28)$$

The resolution of forces from wind axes to local geocentric then becomes

$$\begin{vmatrix} F_{X_g} \\ F_{Y_g} \\ F_{Z_g} \end{vmatrix} = \begin{vmatrix} r_1 & r_2 & r_3 \\ s_1 & s_2 & s_3 \\ t_1 & t_2 & t_3 \end{vmatrix} \begin{vmatrix} F_{X_A} \\ F_{Y_A} \\ F_{Z_A} \end{vmatrix} \quad (7.3.29)$$

For the rotating planet, the local geocentric components must be resolved into the X_e -Y_e -Z_e system. The required direction cosines are given by Equation (7.3.20)

$$\begin{vmatrix} F_{X_e} \\ F_{Y_e} \\ F_{Z_e} \end{vmatrix} = \begin{vmatrix} i_1 & i_2 & i_3 \\ j_1 & j_2 & j_3 \\ k_1 & k_2 & k_3 \end{vmatrix} \begin{vmatrix} F_{X_g} \\ F_{Y_g} \\ F_{Z_g} \end{vmatrix} \quad (7.3.30)$$

The combined transformation from wind axes to local geocentric can be defined as a single matrix transformation [o,p,q]. Adding in the gravitational force component, the total force in the (X_e, Y_e, Z_c) coordinate system becomes

$$\begin{vmatrix} F_{X_e} \\ F_{Y_e} \\ F_{Z_e} \end{vmatrix} = \begin{vmatrix} o_1 & -o_2 & o_3 \\ p_1 & p_2 & p_3 \\ q_1 & q_2 & q_3 \end{vmatrix} \begin{vmatrix} F_{X_A} \\ F_{Y_A} \\ F_{Z_A} \end{vmatrix} + \begin{vmatrix} mg_{X_e} \\ mg_{Y_e} \\ mg_{Z_e} \end{vmatrix} \qquad (7.3.31)$$

7.3.3.3 Body Axis Coordinates

Origin of this system is the vehicle center of gravity with x axis along the geometric longitudinal axis of the body. Positive direction of the x axis is from center of gravity to the front of the body. The y axis is positive to starboard extending from the center of gravity in a water line plane. The z axis forms a right handed orthogonal system. To permit the use of body (x,y,z) axes aerodynamic data and to convert the body axes components of thrust to the wind axes system, a coordinate transformation must be made. The coordinate transformation shown in Figure 7.3-7. involves rotation first through angle of attack, α, then through an auxiliary angle, β'.

The complete transformation is

$$\begin{vmatrix} X_A \\ Y_A \\ Z_A \end{vmatrix} = \begin{vmatrix} \cos\beta'\cos\alpha & \sin\beta' & \cos\beta'\sin\alpha \\ -\sin\beta'\cos\alpha & \cos\beta' & -\sin\beta'\sin\alpha \\ -\sin\alpha & 0 & \cos\alpha \end{vmatrix} \begin{vmatrix} x \\ y \\ z \end{vmatrix} \qquad (7.3.32)$$

which defines the (u, v, ω) direction cosines

$$\begin{vmatrix} X_A \\ Y_A \\ Z_A \end{vmatrix} = \begin{vmatrix} u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \\ w_1 & w_2 & w_3 \end{vmatrix} \begin{vmatrix} x \\ y \\ z \end{vmatrix} \qquad (7.3.33)$$

and the force coefficient transformation

$$\begin{vmatrix} -C_D \\ C_{Y_A} \\ -C_L \end{vmatrix} = \begin{vmatrix} u_1 & u_2 & u_3 \\ v_1 & u_2 & u_3 \\ w_1 & w_2 & w_3 \end{vmatrix} \begin{vmatrix} -C_A \\ C_y \\ -C_N \end{vmatrix} \qquad (7.3.34)$$

The relationship between body and wind axes aerodynamic coefficients is now established.

7.3.3.4 Inertial Coordinates

The selected inertial coordinates coincide with the earth references (X_e, Y_e, Z_e) system at time zero. At a later time they differ by the rotation of the earth, $\omega_p t$.

The transformation from planet referenced velocities to inertial velocities is

$$\begin{vmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{vmatrix} = \begin{vmatrix} \cos \omega_p t & \sin \omega_p t & 0 \\ -\sin \omega_p t & \cos \omega_p t & 0 \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} \dot{X}_e + \omega_p Y_e \\ \dot{Y}_e - \omega_p X_e \\ \dot{Z}_e \end{vmatrix} \tag{7.3.35}$$

The components of inertial velocities are used to calculate the inertial speed of the body as

$$V_I = \sqrt{\dot{X}^2 + \dot{Y}^2 + \dot{Z}^2} \tag{7.3.36}$$

7.3.3.5 Local Geocentric to Geodetic Coordinates

Positions on the planet are specified in terms of geodetic latitude and altitude (for a given longitude) while the motion of the body is computed in a planetocentric system which is independent of the surface. In the computer program flight path angle γ and heading angle σ are calculated with respect to the local geocentric coordinates. By definition γ_D and σ_D are angles measured with respect to the local geodetic. Although the maximum difference that can exist between the two coordinate systems is 11 minutes of arc, it may be desirable to know γ_D and σ_D more accurately than is obtained when measured from the local geocentric.

It is necessary to resolve the geocentric latitude to geodetic latitude for an accurate determination of position. Figure 7.3-8 presents the geometry required for describing the position of a point in a meridian plane of a planet shaped in the form of an oblate spheroid

$$\left(\frac{X}{R_e}\right)^2 + \left(\frac{Z}{R_p}\right)^2 = 1 \tag{7.3.37}$$

It is apparent from Figure 7.3-8 that the most significant difference between the geocentric referenced position and the geodetic position is the distance $\overline{AB}$ on the surface of the reference spheroid. The distance can be defined by a knowledge of the angle ϕ_L; the geocentric latitude, ϕ_g; the geodetic latitude; the corresponding radii; and the distance $\overline{OC}$.

The flight path and heading angles corrected to the local geodetic latitude are computed by

2-5

$$\bar{\gamma}_D = \text{Sin}^{-1}\left(\frac{-\dot{Z}_g - \{\dot{X}_g(\phi_g - \phi_L)\}}{V_g}\right)$$

(7.3.38)

and

$$\sigma_D = \text{Sin}^{-1}\left(\frac{\dot{Y}_g}{\sqrt{\{\dot{X}_g + \dot{Z}_g(\phi_g - \phi_L)\}^2 + \dot{Y}_g^{\,2}}}\right) \tag{7.3.39}$$

where ϕ_g is computed by an iterative scheme described in References 1 and 2.

7.3.4 Auxiliary Computations

In addition to the computations which can be made from the problem formulation as presented in preceding sections, several other quantities are available as optional calculations

a. Planet-surface referenced range, R_D

b. Great circle range, R_g

c. Down- and cross-range, X_D and Y_D

d. Theoretical burnout velocity, V_{theo}

e. Velocity losses, V_P, V_{grav}, V_D, and V_{ML}

f. Orbital variables and satellite target

7.3.4.1 Planet Surfaced Referenced Range

The total distance traveled over the surface of the planet is computed as the integrated surface range. The curvilinear planet surface referenced range is

$$R_D = \int_{t_1}^{t_2} \frac{R_{\phi_L}}{R} V_g \text{ Cos } \gamma \, dt \tag{7.3.40}$$

The flight path angle, γ, is referenced to local geocentric coordinates for this computation.

7.3.4.2 Great Circle Range

Great circle distance from the launch point to the instantaneous vehicle position, R_g, may also be required, Figure 7.3-10. The surface referenced great circle range from the launch point to the vehicle is approximated by

$$R_g = \left[\frac{R_{\phi_L} + R_{\phi_{L_o}}}{2}\right] \text{Cos}^{-1} \left[\text{Sin}\phi_L \text{ Sin } \phi_{L_o} + \text{Cos } \phi_L \text{ Cos } \phi_{L_o} \text{ Cos}(\theta_L - \theta_{L_o})\right] \tag{7.3.41}$$

7.3.4.3 Down and Cross Range

Down and cross range from the initial great circle can be determined. The initial great circle is determined from the input quantities, σ_o, ϕL_o, and θ_{L_o}, Figure 7.3-11. Then the cross range of a particular trajectory point is defined as the perpendicular distance from the point to the initial great circle. The downrange is then the distance along the initial great circle from the initial point to the point P at which the cross range is measured. From the spherical triangle, Figure 7.3-11, the great circle range LF to the point F is computed by Equation (7.3.41).

The right spherical triangle LPF is solved for the downrange, X_D, and the cross range, Y_D.

$$X_D = \left(\frac{R_{\phi L} + R_{\phi L_o}}{2}\right) \cdot \cos^{-1}\left(\frac{\text{Cos LF}}{\text{Cos}(\sin^{-1}(\text{sinLF } \sin\xi))}\right) \tag{7.3.42}$$

$$Y_D = \left(\frac{R_{\phi L} + R_{\phi L_o}}{2}\right) \sin^{-1} (\text{sinLF } \sin\xi) \tag{7.3.43}$$

where

$$\xi = \zeta - \sigma_p \tag{7.3.44}$$

7.3.4.4 Theoretical Burnout Velocity and Losses

For trajectory and performance optimization studies, it is convenient to know the theoretical burnout velocity possible and the velocity losses due to gravity, aerodynamic drag, and atmospheric back pressure upon the engine nozzle. These quantities may be computed as follows:

Theoretical Velocity:

$$V_{theo} = \int_{t_1}^{t_2} \frac{T_{VAC}}{m} \, dt \tag{7.3.45}$$

Speed Loss Due to Gravity:

$$V_{grav} = \int_{t_1}^{t_2} -g_{Zg} \sin \gamma \, dt \tag{7.3.46}$$

Speed Loss Due to Aerodynamic Drag:

$$V_D = \int_{t_1}^{t_2} \frac{D}{m} \, dt \tag{7.3.47}$$

Speed Loss Due to Atmospher Back Pressure Upon the Engine Nozzle:

$$V_P = \int_{t_1}^{t_2} \left(- \frac{PA_e}{m} \right) dt \tag{7.3.48}$$

Maneuvering Losses:

$$V_{ML} = \int_{t_1}^{t_2} \left(\frac{T_{VAC} - PA_e}{m} \right) \left(\cos \alpha - 1 \right) dt \tag{7.3.49}$$

The resultant velocity $V_g'(t_2)$ is obtained by adding the components computed to the initial value $V_g'(t_1)$

$$V_g'(t_2) = V_g'(t_1) + V_{theo} + V_{grav} + V_D + V_P \tag{7.3.50}$$

The maneuvering losses are valid only if λ_T is zero for the engine.

7.3.4.5 Orbital Variables and Satellite Target

Orbital variable calculations follow the calculation of vehicle inertial velocity. Flight path angles in inertial space are computed from the expressions

$$\sigma_I = \tan^{-1} \left(\frac{\dot{Y}_g + \omega_p |R| \cos\phi_L}{\dot{X}_g} \right) \tag{7.3.51}$$

$$\gamma_I = \sin^{-1} \left(\frac{\dot{Z}_g}{|V_I|} \right) \tag{7.3.52}$$

The inclination angle, i, is the angle between the plane containing the velocity vector and the center of the earth, and the equatorial plane.

Applying spherical trigonometry to Figure 7.3-12, we obtain the relationship

$$\cos i = \cos \phi_L \sin \sigma_I \tag{7.3.53}$$

The difference in longitude between the vehicle and the ascending node, ν, is given by

$$\tan \nu = \sin \phi_L \tan \sigma_I \tag{7.3.54}$$

The inertial longitude is given by

$$\theta_I = \theta_L - \omega_p \tau \tag{7.3.55}$$

and the inertial longitude of the ascending node by

$$\Omega = \theta_I - \nu \tag{7.3.56}$$

It is convenient to know the central angle, u, in the orbital plane. Measuring from the ascending node,

$$\tan u = \frac{\tan \phi_L}{\cos \sigma_I} \tag{7.3.57}$$

The orbital variable calculation introduces positional and velocity information from a second body. This body is a satellite considered in a circular orbit about the earth. Its orbital height, h_s, is specified and remains constant. Position in the orbit is computed from an initial central angle, ϕ_{s_o}, by the expression

$$\phi_s = \phi_{s_o} + \omega_s t \tag{7.3.58}$$

The satellite angular velocity is obtained from the satellite inertial velocity, V_{c_s}, where

$$V_{c_s} = \sqrt{\frac{\mu_g}{(R_e+h_s)}} \tag{7.3.59}$$

where μ_g is the gravitational potential constant and R_e is the earth's radius. It should be noted that Equation (7.3.59) assumes a spherical earth; for the earth's radius is taken as constant, and none of the higher order gravitational harmonics are included. Knowing V_{c_s}, it follows that

$$\omega_s = \frac{V_{c_s}}{R_e+h_s} \tag{7.3.60}$$

7.3.5 Vehicle Characteristics

Methods by which the aerodynamic, propulsive, and physical characteristics of a vehicle are introduced into the computer program are presented in this section. Form and preparation of the input data are discussed together with methods by which stages and staging may be used to increase the effective data storage area allotted to a description of the vehicle's properties.

7.3.5.1 Aerodynamic Forces

Aerodynamic forces are defined by three mutually perpendicular forces: lift (L), drag (D), and side force (Y). Lift force is perpendicular to the velocity vector in a vertical plane; drag force is measured along the velocity vector but in opposite direction; side force is measured in the horizontal plane, positive toward the right, provided the bank angle is zero. If the bank angle is not zero, L and Y will be rotated by $-B_A$ about the velocity vector.

Aerodynamic forces are expressed in the form

$$L = q(V,h)\, SC_L(V,h,\alpha,\beta) \tag{7.3.61}$$

$$D = q(V,h)\, SC_D(V,h,\alpha,\beta) \tag{7.3.62}$$

$$Y = q(V,h)\, SC_Y(V,h,\alpha,\beta) \tag{7.3.63}$$

where q is the dynamic pressure and S is a convenient reference area. The aerodynamic coefficients C_L, C_D, and C_Y may be expressed in terms of the aerodynamic derivatives.

$$C_L = C_{L_o} + C_{L_\alpha}\alpha + C_{L_\alpha 2}\,\alpha|\alpha| + C_{L_\beta}|\beta| + C_{L_\beta 2}\beta^2 + C_{L_{\alpha\beta}}\,\alpha|\beta| \tag{7.3.64}$$

$$C_D = C_{D_o} + C_{D_\alpha}|\alpha| + C_{D_\alpha 2}\,\alpha^2 + C_{D_\beta}|\beta| + C_{D_\beta 2}\,\beta^2 + C_{D_{\alpha\beta}}|\alpha||\beta| \tag{7.3.65}$$

$$C_Y = C_{Y_o} + C_{Y_\alpha}|\alpha| + C_{Y_\alpha 2}\,\alpha^2 + C_{Y_\beta}\beta + C_{Y_\beta 2}\,\beta|\beta| + C_{Y_{\alpha\beta}}|\alpha|\beta \tag{7.3.66}$$

Alternatively, the aerodynamic derivatives may be expressed as tabular variables of independent variables such as Mach number (M_N), altitude (h), α, and β, that is, functions of the state variables and the control variables.

It may be convenient to measure the aerodynamic forces in the body axis coordinate system introduced in Section 7.3.3.3. In this case, normal force (n_f) is measured along the -z axis; side force (y) along the y axis, and axial force (a) along the -x axis. The specification of forces in the body axis system is similar to that in the wind axis system.

7.3.5.2 Thrust and Fuel Flow Data

The techniques employed to introduce thrust and fuel-flow data into the equations of motion are developed in an approach similar to that employed for aerodynamic data. An n-dimensional tabular listing and interpolation technique is used with the independent variables being defined by the type of propulsion unit being considered. The propulsion units are grouped into the following options: (1) rocket and (2) airbreathing engines.

7.3.5.2.1 Propulsion Option (1) Rocket. The thrust of a rocket motor is assumed variable with stage time, altitude, and, if the rocket is controllable, with throttle setting. The altitude effect is determined by the exit area of the nozzle, A_e, and the ambient atmospheric pressure, P. If the thrust is specified for some constant ambient air pressure, the altitude correction can be calculated within the subprogram. If the rocket motor is uncontrolled, the vacuum thrust (in pounds) will be introduced by a tabular listing as a function of time (in seconds) and corrected as follows:

$$T = \text{Max} \quad [T_{vac} - PA_e, 0] \tag{7.3.67}$$

The propellant consumption rate is specified by a tabular listing in slugs per second as a function of time (in seconds) for the single-engine options, or computed from the thrust and the engine specific impulse, I_{SP}, for the multiple engine options.

If the rocket is controlled, the propellant mass flow rate $\dot{m}_f$, is introduced by a tabular listing as a function of throttle setting. The thrust is then specified by a tabular listing as a function of mass flow rate.

7.3.5.2.2 Propulsion Option (2) Airbreathing Engines. An airbreathing engine is strongly affected by the environmental conditions under which it is operating. Engines which would be grouped in this classification are turbojets, ramjets, pulsejets, turboprops, and reciprocating machines. The parameters considered significant in the program are

a. Altitude (h-ft)
b. Mach number (M_N)
c. Angle of attack (α-degrees)
d. Throttle setting (N-units defined by problem)

Both the thrust and fuel flow are functions of these variables. In order to accommodate these variables, a five-dimensional tabular listing and interpolation are used to obtain both thrust and fuel flow. The thrust has no further correction as the effects of all parameters are assumed included in the interpolated value.

7.3.5.2.3 Engine Perturbation Factors. The engine options include provision for two data scaling factors for use in parametric studies; these are in the form

$$T = \varepsilon_{13}\, T_{VAC} + \varepsilon_{14} \tag{7.3.68}$$

7.3.5.2.4 Components of the Thrust Vector. The equations used to reduce the thrust vector to its components along the body axes are

$$T_x = T \cos\lambda_T \tag{7.3.69}$$

$$T_y = -T \sin\lambda_T \cos\phi_T \tag{7.3.70}$$

and

$$T_z = -T \sin\lambda_T \sin\phi_T \tag{7.3.71}$$

λ_T and φ_T are defined in Section 7.3.2.

7.3.5.2.5 Reference Weight and Propellant Consumed. Rate of change of vehicle mass, $\dot{m}$, is set equal to the negative of the total mass flow rate, $-\dot{m}_t$. $\dot{m}$ is integrated to give variation of vehicle mass, m. The instantaneous mass is used in the computation of the body motion. The reference weight is obtained by an auxiliary calculation

$$W_T' = 32.174 \cdot m \tag{7.3.72}$$

The propellant consumed is computed as

$$m_f = m_o - m \tag{7.3.73}$$

where m_o is a reference mass input equal to the initial vehicle mass.

7.3.5.3 Stages and Staging

A problem common in missile performance analyses and encountered frequently in airplane performance work is that of staging or the release of discrete masses from the continuing airframe. The effect of dropping a booster rocket or fuel tanks is often great enough to require that the complete set of aerodynamic data be changed. Configuration changes at constant weight, such as extending drag brakes or turning on afterburners, may also require revising the aerodynamic or physical characteristics of the vehicle. At each stage point the equation of motion integration is stopped on a given stage cut-off function with precision. The next stage integration is then restarted following specification of the revised vehicle configuration.

7.3.6 Vehicle Environment

The models for simulating the environment in which a vehicle will operate are presented in this section. This environment includes the atmosphere properties, wind velocity, and the field associated with the planet over which the vehicle is moving. The shape of the planet and the conversion from geodetic to geocentric latitudes are also considered. In the discussions which follow, the descriptions of vehicle environment pertain to the planet Earth. The environmental simulation may be extended to any planet by replacing appropriate constants in the describing equations.

7.3.6.1 Atmosphere

Two atmospheres are considered in this program: the 1959 ARDC Model Atmosphere and the 1962 ARDC Model Atmosphere. The 1959 ARDC Model Atmosphere is specified in layers assuming either isothermal or linear temperature lapse-rate sections. This construction makes it very convenient to incorporate other atmospheres either from specifications for design purposes or for other planets. The relations which mathematically specify the 1959 ARDC Model Atmosphere are as follows: the 1959 ARDC Model Atmosphere is divided into 11 layers as noted in the table below.

Layer	H_b-Lower Altitude (Geopotential Meters	Upper Altitude (Geopotential) Meters
1	0	11,000
2	11,000	25,000
3	25,000	47,000
4	47,000	53,000
5	53,000	79,000
6	79,000	90,000
7	90,000	105,000
8	105,000	160,000
9	160,000	170,000
10	170,000	200,000
11	200,000	700,000

For layers 1, 3, 5, 7, 8, 9, 10, and 11 a linear molecular scale temperature lapse rate is assumed and the following equations are used:

$$H_{gp} = \frac{.3048h}{1+.3048h/6356766} \qquad \text{Meters} \qquad (7.3.74)$$

$$T_m = (T_M)_b[1 + K_1(H_{gp}-H_b)] \qquad °R \qquad (7.3.75)$$

$$T = T_M[A - B \tan^{-1} (\frac{H_{gp}-C}{D})] \qquad °R \qquad (7.3.76)$$

$$P = P_b[1 + K_1(H_{gp}-H_b)]^{-K_2} \qquad Lb/Ft^2 \qquad (7.3.77)$$

$$\rho = \rho_b[1 + K_1(H_{gp} - H_b)]^{-(1+K_2)} \qquad Slugs/Ft^3 \qquad (7.3.78)$$

$$V_s = 49.021175(T_M)^{1/2} \qquad Ft/sec \qquad (7.3.79)$$

$$\nu = 2.269681 \times 10^{-8} \left[\frac{T^{3/2}}{(T+198.72)\rho}\right] \qquad Ft^2/sec \qquad (7.3.80)$$

For the isothermal layers 2, 4, and 6, the following changes are made

$$P = P_b e^{-K_3(H_{gp}-H_b)} \qquad (7.3.81)$$

$$\rho = \rho_b e^{-K_3(H_{gp}-H_b)} \qquad (7.3.82)$$

Values of the temperature, pressure, density, and altitude at the base of each altitude layer are listed together with the appropriate values K_1, K_2, and K_3 in References 1 and 2.

7.3.6.2 Winds Aloft

The winds aloft subprogram provides for three separate methods of introducing the wind vector: as a function of altitude, a function of range, and a function of time. This facilitates the investigation of wind effects for the conventional performance studies. The wind vector is approximated by a series of straight line segments for each of the methods mentioned above.

Four options are used to define the wind vector in the computer program. The three components of the wind vector in a geodetic horizon coordinate system can be specified as tabular listings with linear interpolations (curve reads) in the following options:

Wind options (0). In this option the wind vector is zero throughout the problem. This allows the analyst the option of evaluating performance without the effects of wind. This option causes the winds-aloft computations to be bypassed.

Wind option (1). In this option the components of the wind vector are specified as a function of time. Wind speeds are specified in feet per second and time in seconds.

Wind option (2). The three components of the wind vector are introduced as a function of altitude in this option. Wind speed is specified in feet per second and altitude in feet.

Wind option (3). In this option the components of the wind vector are introduced as a function of range. Wind speed is specified in feet per second and range in nautical miles. The range utilized in this computation is the great circle range.

By staging of the wind option, it is possible to switch from one method of reading wind data to another during the computer run.

7.3.6.3 Gravity

Spherical harmonics are normally used to define the gravity potential field of the Earth, References 3 and 4. Each harmonic term in the potential is due to a deviation of the potential from that of a uniform sphere. In the present analysis the second-, third-, and fourth-order terms are considered. The first-order term, which would account for the error introduced by assuming that the mass center of the Earth is at the origin of the geocentric coordinate system is assumed to be zero. With this assumption

$$U = \frac{\mu}{R}\left[1 + \frac{J}{3}\left(\frac{R_e}{R}\right)^2 P_2 + \frac{H}{5}\left(\frac{R_e}{R}\right)^3 P_3 + \frac{K}{30}\left(\frac{R_e}{R}\right)^4 P_4 + \dots\right] \quad (7.3.83)$$

where P_2, P_3, and P_4 are Legendre functions of geocentric latitude ϕ_L expressed as

$$\begin{aligned} P_2 &= 1 - 3\sin^2\phi_L \\ P_3 &= 3\sin\phi_L - 5\sin^3\phi_L \\ P_4 &= 3 - 30\sin^2\phi_L + 35\sin^4\phi_L \end{aligned} \quad (7.3.84)$$

The gravitational acceleration along any line is the partial derivative of U along that line; in particular:

$$g_{Z_g} = \frac{\mu_g}{R^2}\left[1 + J\left(\frac{R_e}{R}\right)^2 P_2 + \frac{4H}{5}\left(\frac{R_e}{R}\right)^3 P_3 + \frac{K}{6}\left(\frac{R_e}{R}\right)^4 P_4\right] \quad (7.3.85)$$

$$g_{X_g} = \frac{\mu_g}{R^2}\left[-2J\left(\frac{R_e}{R}\right)^2 P_5 + \frac{3H}{5}\left(\frac{R_e}{R}\right)^3 P_6 + \frac{2K}{3}\left(\frac{R_e}{R}\right)^4 P_7\right] \quad (7.3.86)$$

and

$$g_{Y_g} = 0.0 \quad (7.3.87)$$

where

$$P_5 = \sin\phi_L \cos\phi_L$$

$$P_6 = \cos\phi_L \,(1-5 \sin^2 \phi_L)$$

$$P_7 = \sin\phi_L \cos\phi_L \,(-3 + 7 \sin^2 \phi_L) \quad (7.3.88)$$

Equations (7.3.85) and (7.3.86) are used in the gravity subroutine with the following values recommended for the constants.

$$\mu_g = 1.407698 \times 10^{16} \quad \text{ft /sec}$$

$$R_e = 20{,}925{,}631. \quad \text{ft.}$$

$$J = 1623.41 \times 10^{-6}$$

$$K = 6.37 \times 10^{-6} \quad (7.3.89)$$

It should be noted that these constants and equations pertain to the planet Earth; however, it is possible to use these same quations for any other planet once the appropriate constants from that planet are known.

REFERENCES:

1. Hague, D. S. and Glatt, C. R., Optimal Design Integration of Military Flight Vehicles, Section 7.3, "Atmospheric Trajectory Optimization Program," AFFDL-TR-72-132, 1972.

2. Mobley, R. L. and Vorwald, R. R., Three-Degree-of-Freedom Problem Optimization Formulation--User's Manual, Part 2, Volume III, FDL-TDR-64-1 October 1964.

3. Brown, Robert C., Brulle, R. V., Combs, A. E., and Griffin, G. D., Six Degree-of-Freedom Flight Path Study Generalized Computer Program, Part 1 Volume I, FDL-TDR-64-1, October 1964.

4. Seubert, F. W. and Usher, Newell E., Six Degree-of-Freedom Flight Path Study Generalized Computer Program--User's Manual, Part 2, WADD Technical Report 60-781, McDonnell-Douglas Corporation, May 1961.

5. Landgraf, S. K., Some Practical Applications of Performance Optimization Techniques to High Performance Aircraft, AIAA Paper 64-288, July 1964.

6. Hague, D. S., An Outline and Operating Instructions for the Steepest-Descent Trajectory Optimization Program - STOP, Aerodynamics Methods Note 1, McDonnell-Douglas Corporation, March 1963.

7. Hague, D. S., Geib, Ken, Ballew, L., and Witherspoon, J., Two Vehicle Optimization--Theoretical Outline and Program User's Manual, Report B983, McDonnell-Douglas Corporation, 1965.

8. Hague, D. S.,"The Optimization of Multiple-Arc Trajectories by the Steepest-Descent Method,"Recent Advances in Optimization Techniques, Lavi and Vogl (editors), John Wiley and Sons, Inc., 1966, Pages 489-517.

9. Hague, D. S. and Glatt, C. R., Study of Navigation and Guidance of Launch Vehicles Having Cruise Capability, Volume II, Boeing Document D2-113016-5, The Boeing Company, April 1967.

10. Retka, J., Harder, D., Hague, D. S., Glatt, C. R., Seavoy, T., and Minden, D., Study of Navigation and Guidance of Launch Vehicles Having Cruise Capability, Volume IV, Boeing Document D2-113016-7, April 1967.

11. Stein, L. H., Mathews, M. L., and Frenck, J. W., STOP: A Computer Program for Supersonic Transport Trajectory Optimization, NASA CR-793, May 1967.

12. Rozendaal, H. L., A General Branched Trajectory Optimization Algorithm with Applications to Space Shuttle Vehicle Mission Design, AAS/AAIA Astrodynamics Specialist Conference, August 17-19, 1971.

13. Hague, D. S., Glatt, C. R., and Jones, R. T., Integration of Aerospace Vehicle Performance and Design Optimization, AIAA Paper No. 72-948, Presented at AIAA Second Atmospheric Flight Mechanics Conference, Palo Alto, California, September 11-13, 1972.

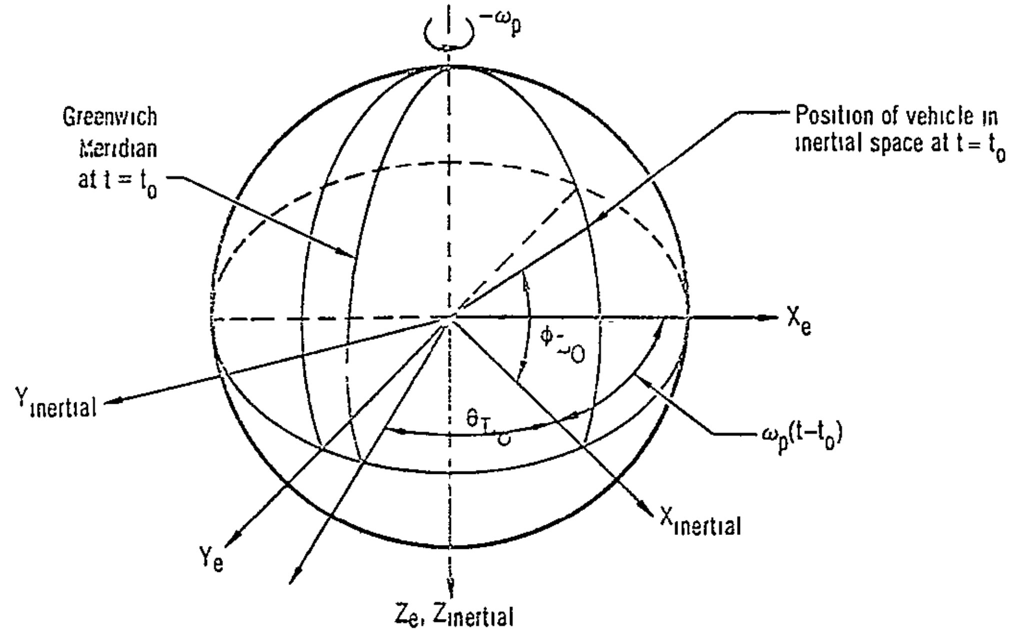


FIGURE 7.3-1. BASIC COORDINATE SYSTEM

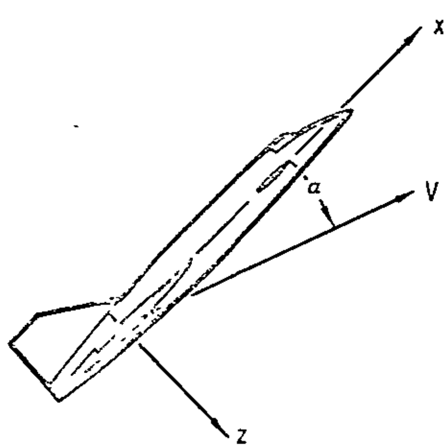
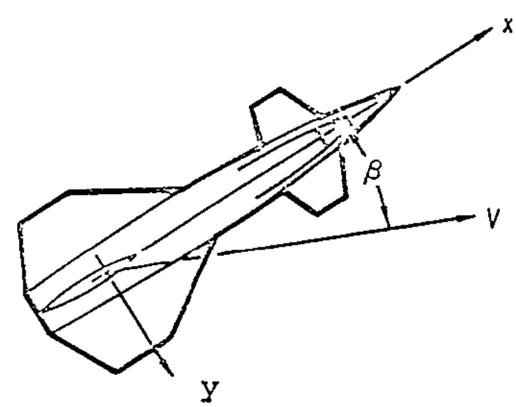


FIGURE 7.3-2. ANGLE OF ATTACK



FIGURL 7.3-3. SIDESLIP ANGLE

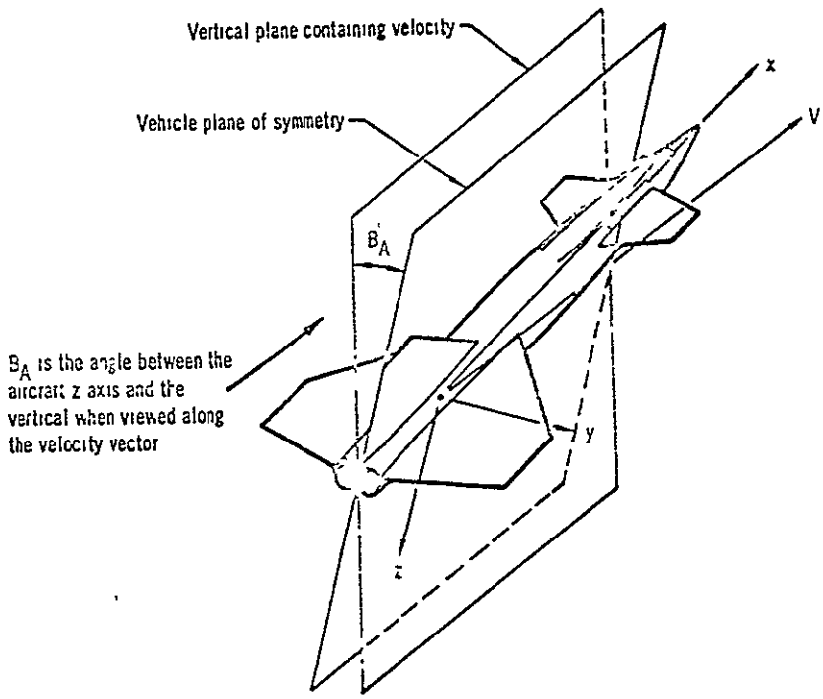


FIGURE 7.3-4. BANK ANGLE

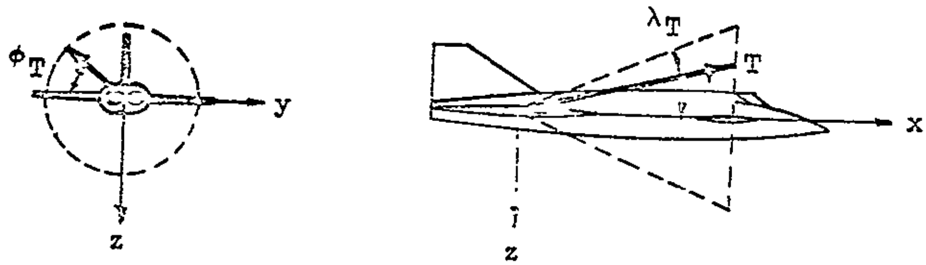
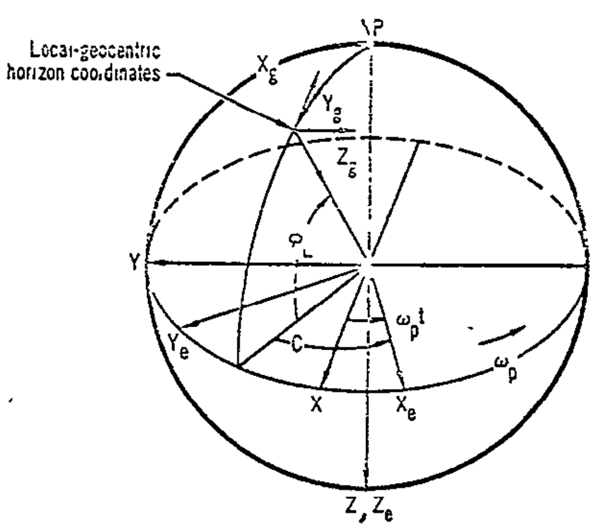


FIGURE 7.3-5. THRUST ANGLES



ORIGINAL PAGE IS
OF POOR QUALITY

FIGURE 7.3-6. RELATION BETWEEN LOCAL-GEOCENTRIC, INERTIAL AND EARTH-REFERENCED COORDINATES FOR POINT MASS PROBLEMS

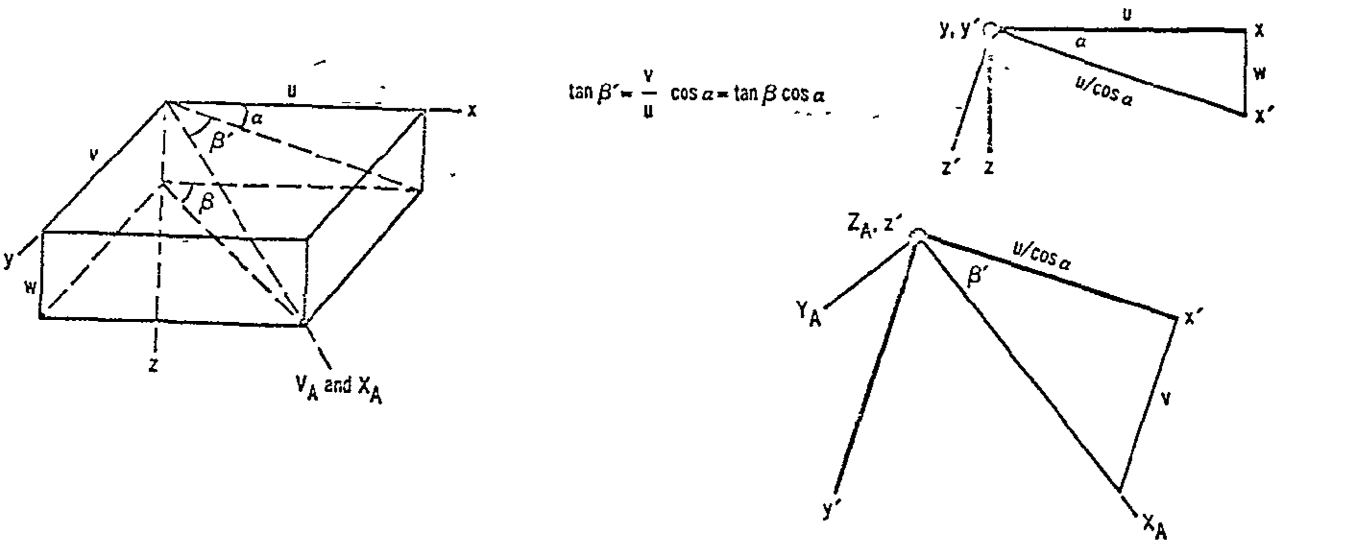
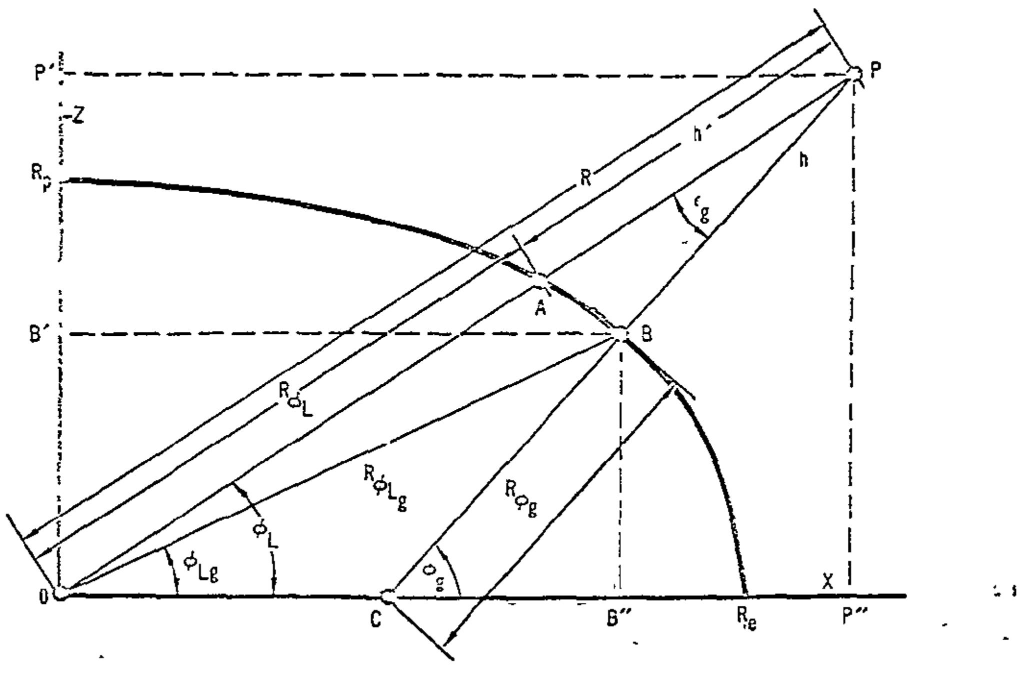


FIGURE 7.3-7. RELATIONSHIP BETWEEN BODY AXES AND WIND AXES



FIGURF 7.3- 8. PLANET-OBLATENESS EFFECT ON LATITUDE AND ALTITUDE

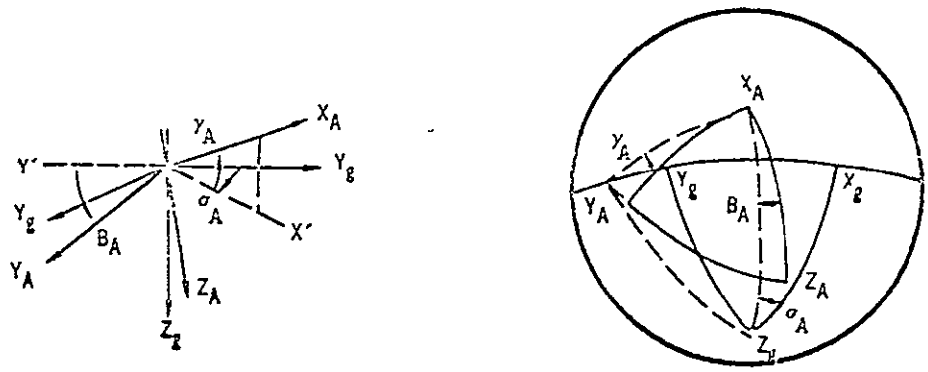


FIGURE 7.3-9. RELATIONSHIP BETWEEN LOCAL-GEOCENTRIC AXES AND WIND AXES

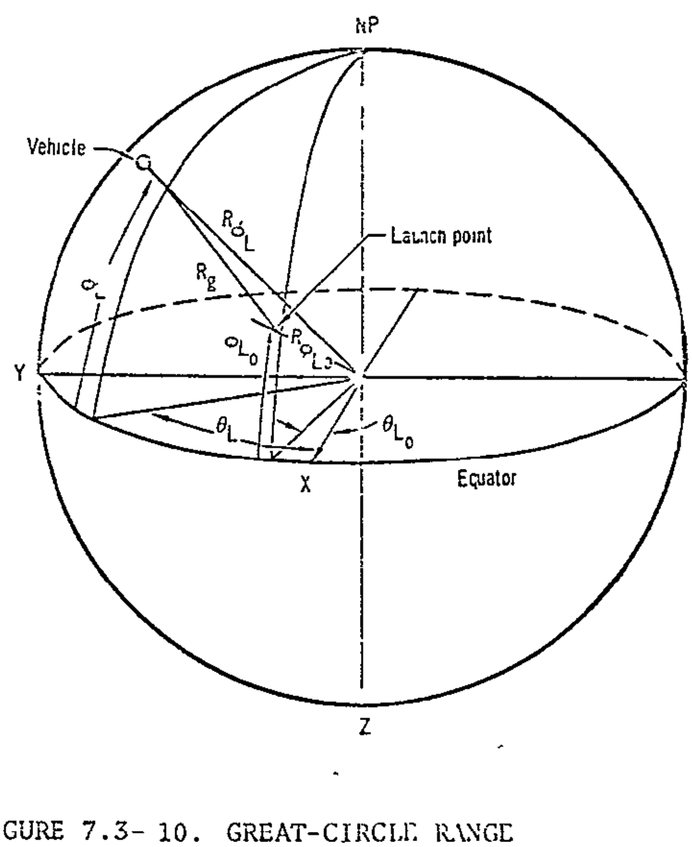


FIGURE 7.3- 10. GREAT-CIRCLE RANGE

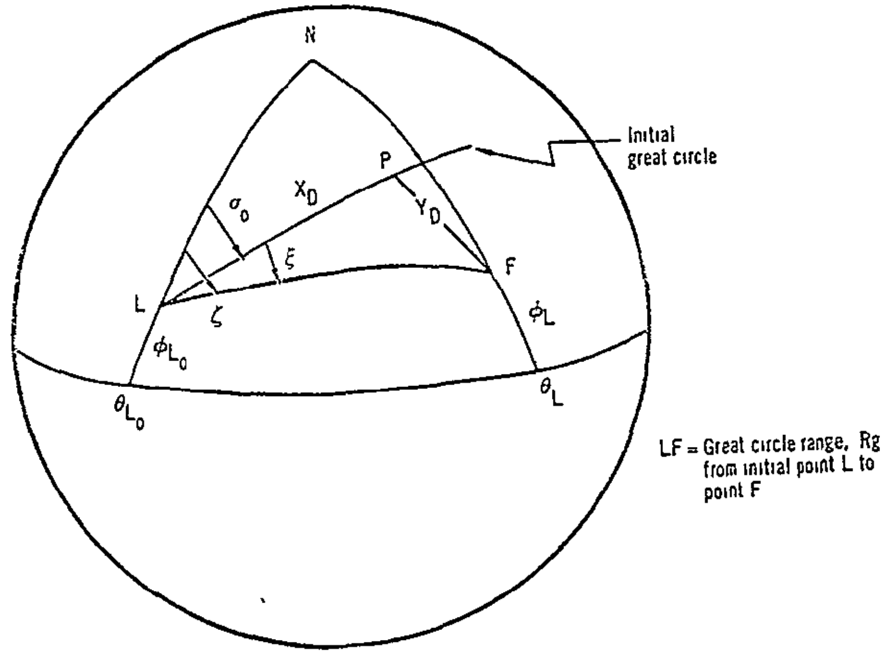


FIGURE 7.3-11. DOWNRANGE AND CROSSRANGE GEOMETRY

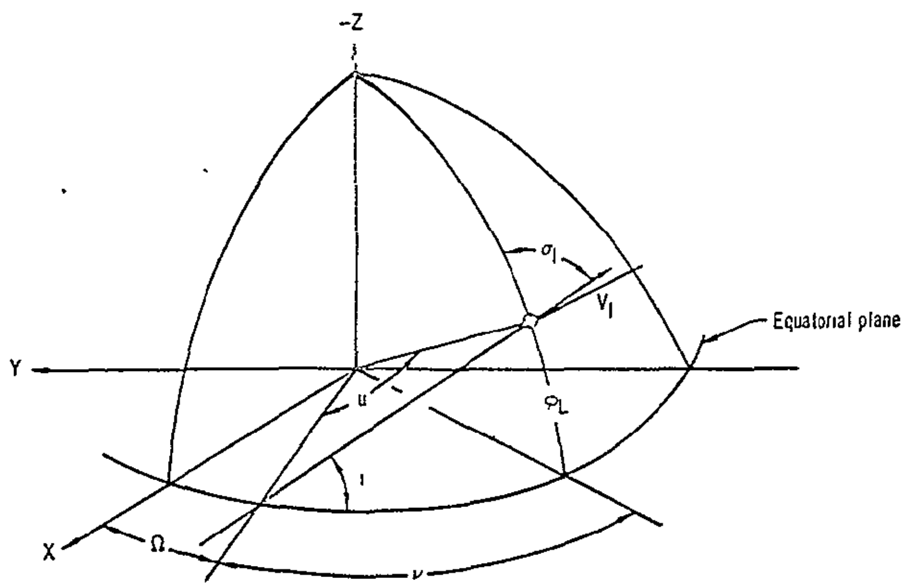


FIGURE 7.3-12. ORBITAL PLANE GEOMETRY

TABLE OF CONTENTS FOR SECTION 7.4, PROGRAM COAP

7.4 PROGRAM COAP: COMBAT OPTIMIZATION AND ANALYSIS PROGRAM

This program was constructed by Aerophysics Research Corporation under Air Force contract F33615-70-C-1036, References 1 to 4. It is an extension of the Section 7.3 ATOP program. The COAP program utilizes two complete three-dimensional equations of motion sets to simulate a one-on-one combative encounter between two military flight vehicles. The flight vehicle aerodynamic and propulsion representations are sufficiently general to permit the simulation of both current and proposed vehicles by data input. The program is written for the CDC 6600 computer but will run on any modern large scale computer with minor modification. Generalized rotating planetary and atmospheric models permit simulation of either aircraft, missile, or spacecraft encounters. Combat *roles* for each vehicle (attacker, defender, etc.) are automatically defined on the basis of vehicle relative positions, headings, and velocities. Depending on the vehicle role selected, any one of a set of *tactics* designed to satisfy the role requirements is executed. These tactics vary in nature from straightforward stylized maneuvers, such as the split S or barrel roll, to three-dimensional lag or lead pursuit paths.

Combat optimization capability may be introduced by repetitive simulation using parameterization of the combat guidance parameters and the application of multivariable search techniques. Alternately, the variational calculus may be employed to define optimal continuous control against a reacting opponent. In the parameter optimization mode, the option to determine a "mini-max" solution is available.

Air-to-air combat imposes severe design requirements on fighter aircraft. Pilot tactics in combat are inadequately treated by conventional flight handbooks or by segmented mission analysis, such as that described in Section 7.2. Historical evidence demonstrates that a high price in men and aircraft will be paid by nations which pay insufficient attention to combat requirements in the design of military aircraft. Recognition of these points has lead to a growing effort to provide improved combat analysis and simulation tools. These tools include self-contained digital computer codes, such as those of References 1 through 7, and the use of dual maneuvering simulators, such as that reported in Reference 8. In this section the Combat Optimization and Analysis Program, COAP, of References 1 through 4, is described. The program was constructed under contract to the Air Force Flight Dynamics Laboratory. COAP simulates a one-on-one combative encounter between two aerospace vehicles. Repetitive sequential simulation of the resulting "dogfight" combined with perturbations of the parameters which define each pilot's course of action defines the optimal course of action for each pilot starting from given initial conditions.

7.4.1 Introduction

Single vehicle optimal flight paths presented may be obtained by the variational steepest-descent formulation program of Section 7.3. COAP is an evolutionary development from previous single vehicle trajectory analysis programs. The two-vehicle encounter is considerably more complex than the single vehicle problem. *First,* the flight paths involved are of a more complex nature. *Second,* there is often more than one optimal solution to be found. *Third,* the optimal solution(s) may be of the mini-max type. For example, consider Figure 7.4-1. From the given initial conditions at least four varied types of optimization problems arise:

1. both vehicles may attack
2. vehicle A may flee while vehicle B attacks
3. vehicle B may flee while vehicle A attacks
4. both vehicles may flee

7.4.2 Outline of the Combat Simulation

7.4.2.1 Equations of Motion

A schematic of the COAP program is presented in Figure 7.4-2. The program contains two three-degree-of-freedom equation of motion systems. The two systems are simultaneously integrated in time and may be mutually coupled through a combative guidance logic block. This block defines an appropriate role for each vehicle, and on defining the roles it specifies a suitable tactic to be followed. Vehicle angle-of-attack, bank-angle, and throttle settings for the selected tactic are automatically generated by the combat logic. Pitch, bank, and throttle rate constraints may be imposed on the simulation.

7.4.2.2 Vehicle and Planetary Characteristics

Vehicle aerodynamic and propulsion representations permit the modelling of any two current military aircraft. The aircraft data is input on punched cards and is not a fixed part of the program. Thus, opposing aircraft types may be rapidly changed. Basic planetary characteristics are represented by a rotating oblate planet having a multi-layered atmosphere. Arbitrary wind profiles and non-standard day atmospheres may be introduced at the user's option. The simulation is thus adequate for representation of aircraft, rocket, or spacecraft encounters.

7.4.2 3 Operating Modes

The combative encounter may be defined at several levels of complexity short of the differential game formulation including:

Option A: Self-contained role and tactic selection based on relative vehicle states

Option B: Parameterization of one vehicle's role and tactic selection rules followed by the application of multivariable search procedures to obtain the optimal parameter values. This option defines optimal parameters against a specified opponent employing fixed combat logic parameters.

Option C: Parameterization of both opponent's role and tactic selection rules followed by the application of a multivariable saddle point search technique. This option defines a "mini-max" optimal procedure for opponents employing variable combat logic parameters.

Option D: Open loop, continuous control optimization by the variational calculus against an opponent performing a pre-specified maneuver, the "maneuvering target" option.

Option E: Open loop, continuous control optimization by the variational calculus against a reacting opponent employing fixed parameters and self-contained combat tactics.

The formulation and program include as subcases two-vehicle cooperative problems. This leads to

Option F: Cooperative two-vehicle parametric control

Option G: Cooperative two-vehicle open loop continuous control

These last two options permit the optimization of two-vehicle rendezvous problems and are equally applicable to aircraft or spacecraft problems. Single vehicle problems may also be studied by means of the program.

In the Combat Optimization and Analysis Program schematic of Figure 7.4-2, data input and initializations are carried out in the MAIN program link. Integration of the two-vehicle equations of motion occurs in the EXE link. Program EXE controls the equations of motion directly, employing the COMBAT routines for definition of combative guidance logic. The basic coordinate system employed in the equations of motion is illustrated by the inset in Figure 7.4-2. When an encounter is complete, a switch controls the program logical flow. If an isolated combative simulation has been requested, a return to the MAIN program allows the next problem to be entered. When a parameter optimization problem is being studied, the switch passes program control to the AESOP link. AESOP defines new combative parameters and generates a succession of improving dogfights through the parameter optimization loop. When a variational optimization problem is studied, the switch passes program control to the TOP link. TOP defines new continuous control histories through a variational steepest-descent algorithm and sensitivities based on an adjoint equation solution obtained in the REV link. A succession of improving combative encounters is then generated through the variational optimization loop.

7.4.3 Vehicle Roles and Tactics

Five roles are available in the COAP program. One or more tactics are available to each vehicle for each role. Role selection is on the basis of vehicle relative states. This state space is partitioned in a manner which insures a unique role selection for all relative vehicle conditions encountered in a combat simulation. Figure 7.4-3 illustrates a combat simulation in schematic form.

In the schematic the thick solid line shows Vehicle A initially flying in a passive role using a prespecified continuous control history and Vehicle B attacking using his second tactic. As the maneuver develops (thin solid line), Vehicle A becomes aware of the attack and evades using tactic 1 while Vehicle B changes his plan of attack to tactic 3. Finally (dotted line) Vehicle A achieves an attacking situation; in response, Vehicle B evades.

The COAP program has the ability to automatically generate such a sequence of role and tactic decisions on the basis of relative state. Alternately, the analyst may override the program logic to force given roles and tactics during the course of an encounter.

7.4.3.1 Role Selection

Role selection is based on a global partitioning of the state space using the physically oriented coordinates:

1. separation distance, ΔR
2. cone angle to target, θ_T
3. target's angle-off, ϕ_{off}

These coordinates are illustrated in Figure 7.4-4. The basic role selection logic tree employed in the combat simulation is illustrated in Figure 7.4-5. This role selection logic tree can readily be modified to incorporate additional logic. These additions may result from a general improvement in role selection logic or be tailored to a specified combat situation; for example, provision of overshoot prevention or breakaway logic.

7.4.3.2 Tactic Selection

Following role selection each vehicle selects a tactic appropriate to the chosen role. *Defensive* tactics may be stylized maneuvers such as the split-S or Immelamn. Alternately, they may involve instantaneous maximization of a specified vector's rotation rate, such as line-of-sight or lead-pursuit vectors. *Evasive* tactics are limit control maneuvers. *Offensive* tactics operate on tracking vectors such as lag pursuit, line-of-sight, or lead-pursuit. With these tactics the instantaneous sum of the aerodynamic and propulsive force components projected onto the specified vector is maximized at all times. *Attacking* is limited to a single tactic which simultaneously tracks the firing point and eliminates any pointing error. *Passive* maneuvers involve flight along a specified preset path. A list of the available tactics is presented in Figure 7.4-6.

Within a given role the program will select the available tactics in ordered fashion as specified by the analyst or, alternately, the tactics for a given role may be selected in a random sequence by internal program logic. Additional tactics can readily be added to the basic set described in Figure 7.4-6.

All available tactics may be overridden when prespecified limits are exceeded. Thus, if the control required in a given tactic exceeds the vehicle's acceleration or lift coefficient limits, control values may automatically be modified to maintain the constraints. Again, if lower limits on altitude or Mach number are violated, an appropriate pull-out or nose-down maneuver may be automatically initiated by the COAP simulation.

7.4.3.3 Finite Control Rates

Control motions required by the various tactics may be instantaneously applied to each vehicle, or, at the user's option, the maximum control rates to be employed in the simulation may be subject to practical maneuver constraints. With the imposition of such finite control rates, the instantaneous control specified by a given tactic becomes the desired control. The difference between desired and current control values defines the instantaneous control error. Actual control rates to be employed are then defined as functions of the control error as follows:

a. Use maximum rate for large control errors

b. Use a parabolic rate variation with control error for intermediate control error values

c. Use a linear rate variation with control error for small errors to insure elimination of the error in a given time

This is illustrated in Figure 7.4-7. When the finite control rate options are employed, actual control values are obtained by timewise integration of the computed control rates.

Weapon systems are introduced into the combat simulation in the form of upper and lower inequality constraints. When all applicable boundary constraints for a given weapon are satisfied, the weapon can be activated. The total time in which a weapon system may be activated is integrated and used as a measure of a vehicle's combat ability. In the basic COAP program each vehicle may carry up to three weapon systems employing a total of nine fire control boundaries in arbitrarily defined groupings.

7.4.4 Combat Simulation

A typical combat simulation is presented in Figure 7.4-8. Initial condition is a head-on pass with lateral offset. The delta winged vehicle is a representative twin engine fighter; the unswept wing is a representative single engine fighter. Total flight duration is eighty seconds. Finite control rates are employed.

Initially, both vehicles bank towards each other in a reattack maneuver. After approximately 270° of turning flight, a near head-on pass occurs at approximately 35 seconds into the encounter. Neither vehicle achieves a firing opportunity in the near head-on pass; the single engine aircraft's steering error is approximately half that of the twin engine aircraft at this time. The encounter continues with the twin engine aircraft turning almost horizontally, and the single engine aircraft entering a near split S. At approximately 50 seconds into the encounter, the twin engine aircraft banks over, and the vehicles again approach each other in a near head-on situation at 60 seconds. Again, no firing opportunity occurs, and the single engine vehicle has the smaller steering error. The encounter continues with the single engine aircraft turning almost vertically and maintaining a superior steering error at the cost of both altitude and Mach number penalties. Both vehicles' Mach numbers have been reduced severely by the encounter as a result of the high "g" maneuvers. This is particularly true of the single engine fighter.

In combat simulations to-date the Mach number loss associated with high "g" maneuvers is a distinct characteristic. Another characteristic of combat simulations starting from near equality is that provided both vehicle concentrate on pulling hard at each other, the firing opportunities are very limited.

In Figure 7.4-8 both vehicles fly offensively throughout the encounter using line-of-sight force component maximization and perfect knowledge of the opponent's position. The simulation typifies the combat simulation flight paths. It is immediately clear that there is little in common between such paths. It follows that the determination of combat maneuver capability for the military flight vehicles should include realistic combat maneuver simulations.

7.4.5 Combat Performance Rating

Evaluation of a vehicle's combat maneuver capability requires establishing a performance measurement criteria or rating over a spectrum of encounters. This performance rating will clearly be dependent upon

a. The opposing aircraft
b. The initial condition or conditions utilized in the encounter spectrum
c. The manner in which both vehicles are flown
d. The weapon system rating function employed

One possible combat maneuver performance criteria is the percent flight time that a vehicle satisfies its weapon system firing constraints. Figure 7.4-9 presents the results of a study involving the representative twin engine (Vehicle A) and single engine (Vehicle B) fighters. The weapon system firing measure has been replaced by a simple vision rating in this study. Three boundaries are shown: the per cent time in which each vehicle keeps his opponent in 10°, 20°, and 30° cones, respectively. In Figure 7 4-9 Vehicle B flies neutrally. That is, he maximizes his aerodynamic and propulsive force component along the line-of-sight vector at all times. When Vehicle A also flies in this neutral manner, he is clearly at a disadvantage (oversteer factor = 0); for Vehicle B keeps Vehicle A in a 30° cone for 25 per cent of the time. Conversely, Vehicle A is only capable of keeping Vehicle B in a 30° cone 3 per cent of the time. Total encounter time is 400 seconds commencing from the initial states discussed in the section on combat simulation, Section 7.4.4.

The effect of varying Vehicle A's flight tactics can be assessed readily using this steering factor approach. For example, a single combat tactic parameter can be established as follows:

$$\bar{V}_s = \bar{LOS} + k \cdot (\bar{LP} - \bar{LOS} \qquad (7.4.1)$$

where

V_S = steering vector along which the force component is maximized

$\bar{LOS}$ = line of sight vector

$\bar{LP}$ = lead-pursuit vector

k = scalar combat tactic parameter

When k = 0, a vehicle pulls to the line-of-sight vector (neutral). When k > 0, the vehicle leads the line-of-sight vector. When k = 1, a lead-pursuit course is attempted. When k < 0, a lag-pursuit (understeer) course is attempted. It can be seen from Figure 7.4-9 that over the combat spectrum considered, Vehicle A should oversteer by approximately 50 per cent against a nettrally flown Vehicle B. In Figure 7.4-10 Vehicle A adopts the 50 per cent oversteer tactic, and the effect of understeer and oversteer on Vehicle B's combat maneuver performance is considered.

The difference in combat maneuver capability with flight tactic is clearly apparent from Figures 7.4-9, and 7.4-10 In Figure 7.4-9 the twin engine aircraft is outclassed unless he adopts 50 per cent oversteer. In Figure 7.4-10 Vehicle B is slightly inferior in combat maneuver capability over most of his understeer/oversteer range when Vehicle A adopts 50 per cent oversteer.

Results such as that illustrated are dependent upon the four factors listed at the beginning of this section. The COAP program provides a tool for rapidly evaluating the effect of these factors in a given aircraft design situation.

7.4.6 Optimization by Combat Guidance Parameters

Any specific combative tactic logic can be viewed as a transfer function which transforms the instantaneous relative state into vehicle control commands. Tactics logic can usually be phrased in terms of a set of combat guidance parameters; the resulting vehicle control, and hence the flight path followed, is then dependent upon these parameter values. The scalar parameter k is then dependent upon these parameter values. The scalar parameter k of the previous section typifies such a parameter. As k varies from negative to positive values, the resulting flight paths vary from lag pursuit to line-of-sight pursuit to lead-pursuit paths.

When a tactic has been suitably parameterized, multivariable search techniques, References 4 and 14, may be employed to determine the parameter values which produce the best combat outcome for one vehicle or the other. (The simultaneous optimization of both vehicles' parameters, which leads to a mini-max problem, is discussed later).

To demonstrate this technique the reattack from a head-on pass with lateral offset is considered. Two combat guidance parameters, α_1 and α_2, are employed. These parameters define the Vehicle B scalar parameter of k of Equation (7.4.1) as follows:

a. $k = \alpha_1$, for Vehicle B when Vehicle A's angle-off exceeds 45°

b. $k = \alpha_2$, for Vehicle B when Vehicle A's angle-off is less than 45°

The problem considered is

c. Minimize separation distance when Vehicle B *first* achieves a 10° steering error using the combat guidance parameters α_1 and α_2 to generate a twofold family of Vehicle B/Vehicle A combative encounters.

The resulting optimization problem can readily be solved by multivariable search procedures in the AESOP link of COAP, Figure 7.4-2. This link contains a variety of multivariable search techniques including one-parameter-at-a-time techniques, organized techniques such as steepest-descent or quadratic (Newton-Raphson), and randomized techniques. The techniques available are presented in Figure 7.4-11; the searches may be used either separately or in combination.

To solve the optimization problem by multivariable search, the combative encounter is repetitively simulated while the guidance parameters are systematically perturbed by the selected algorithms. This is illustrated for the present problem in Figure 7.4-12. In Figure 7 4-12(a) the terminal separation distance, which is to be minimized, is presented for each of thirteen sequential combat simulations. The terminal separation distance is reduced from the initial value of 6580 feet on the first trajectory to 5680 feet on the thirteenth trajectory. Figure 7.4-12(b) displays the corresponding terminal steering errors. Vehicle B retains a 10° error for all simulations for this is the termination criteria. Vehicle A's terminal steering error is reduced from 78° to 57° (this function was not directly controlled) Associated with the incidental reduction in Vehicle A's steering error is an increase in terminal closing velocity, Figure 7.4-12(c). The behavior of the guidance parameters, $_1$ and $_2$, is presented in Figure 7.4-12(d). It can be seen that $_1$ has approached the upper bound permitted, and $_2$ is oscillating about unity. This corresponds to a hard turn in which Vehicle B leads the line-of-sight vector until Vehicle A is placed in a 45° angle-off condition followed by a terminal maneuver in which Vehicle B pulls to the line-of-sight vector.

In a second simple illustration of the parameter optimization mode, Figure 7.4-13, Vehicle A attempts to escape from Vehicle B using the best constant flight path angle escape. The flight path angle becomes a combat guidance problem, and a sequence of escapes are made typified by those of Figure 7.4-13. To each escape by Vehicle A, Vehicle B performs an appropriate reattacking turn. The object is for Vehicle A to maximize the terminal separation distance in given time (50 seconds). In the example shown, horizontal flight maximizes the terminal separation. Vehicle B closes the gap for both climbing and descending escapes.

If we now limit Vehicle A to a constant altitude flight, we can create a sequence of guidance parameters for Vehicle B and seek to minimize terminal separation distance through these parameters. In Figure 7.4-14 three such angle-of-attack parameters are introduced to improve the Vehicle B reattack against a horizontal escape by Vehicle A. The result is a descending turn followed by a slow climb. Terminal separation distance is now reduced to 27,000 feet as compared to 40,000 feet in Figure 7.4-13.

In the COAP parameter optimization mode up to 100 combat guidance parameters may be employed to minimize or maximize any given combative function while simultaneously constraining other combat functions to prescribed values.

7.4.7 Variational Optimization Modes

COAP contains an optimization capability based on the variational steepest-descent formulation. Variational problems arise when we attempt to define the optimal control histories directly without introduction of combat tactic logic. Since we seek to define optimal control *histories* rather than optimal guidance *parameters*, the problem involves an infinite number of free variables--hence, the variational aspect of the problem. The method employed is that of Reference 1 which is similar to the Bryson formulation of Reference 15. Control history sensitivities are determined from the adjoint equations as shown schematically in Figure 7.4-2. The variational steepest-descent technique was applied to the two-vehicle flight path optimization problem in Reference 16. There, the second vehicle must fly a predetermined path, the *maneuvering target option*. This option is retained in COAP. A second variational option available in COAP involves optimization of a vehicle flight path with respect to an opponent employing fixed guidance parameters; this will be referred to as the *reacting opponent option*. Examples of each are presented below.

7.4.7.1 The Maneuvering Target Option

With the variational maneuvering target option, the second vehicle flies in the passive role along a prespecified flight path. The first vehicle's flight path is then optimized with respect to the second vehicle's path. Optimization is accomplished by repeated application of the variational steepest-descent algorithm. Typically, some fifteen to twenty applications of this algorithm are required to achieve an optimal solution using the second order convergence logic of Reference 9.

As an example of this approach, consider the Vehicle B reattack of Vehicle A in the previous section. Vehicle A is following a prespecified flight path in the form of an accelerating horizontal escape. Applying the variational steepest-descent maneuvering target option to the Vehicle B turn and chase, we obtain after 15 iterations the optimal pursuit path of Figure 7.4-15.

This variational solution defines the absolute minimum terminal separation distance possible against the Vehicle A horizontal accelerating escape. The minimum separation distance is 17,000 feet; this compares with a terminal separation of 27,000 feet using parameter optimization techniques in Figure 7.4-14. However, in this variational solution it is tacitly assumed that Vehicle A is rigidly committed to the horizontal escape. In consequence, Vehicle B is free to perform a split S at Vehicle A followed by a smooth pull-up. In the optimum three-parameter solution of Figure 7.4-14, Vehicle B more conservatively performs a less steeply descending turn and at all times is prepared to instantaneously react to a change of tactic by his opponent.

7.4.7.2 The Reacting Opponent Option

The most complex COAP operating mode involves the use of variational optimization for a vehicle flying against a non-passive reacting opponent which employs combat guidance logic. Solution of this class of problem involves at least 14 formal state variables in the variational formulation. These state variables are the two-vehicle masses, position components (three each), and velocity components (three each).

An example of this option is presented in Figure 7.4-16. The action commences from the previously employed head-on pass with lateral offset. In the solution Vehicle A attempts to find the absolute maximum terminal separation distance in fixed time from Vehicle B's reattack using combat guidance logic.

The solution is obtained from two nominal starting points: a climb and a descent. End points achieved by each vehicle in the two nominal paths are indicated by the isolated arrowheads. The two final solutions obtained are in close agreement as indicated by the matched pairs of solid and dotted lines. This close agreement serves to confirm optimality of the solution obtained.

The variational optimal escape for Vehicle A is a shallow dive and turn *away* from Vehicle B. The shallow turn lengthens the separation distance for a negligible performance loss over the planar escape. Vehicle A now achieves a final separation distance of 45,000 feet, some 5,000 feet better than the best constant flight path angle escape obtained by parameter optimization, Figure 7.4-13. Vehicle B almost achieves a firing opportunity at 20 seconds coming out of the turn. Vehicle B's weapon system constraints on steering error and target angle-off are satisfied (a heat seeker), however, the separation distance at this point is too great. Hence, Vehicle A can make good his escape; for at 50 seconds the gap continues to widen.

It should be noted that if Vehicle B carried a long range weapon, Vehicle A's escape would not be possible. In the converse situation where Vehicle B attempts to escape while Vehicle A reattacks, the escape is impossible for Vehicle A does carry a long range weapon. It follows that given initial conditions of this problem, Vehicle B must stay to fight while Vehicle A may select to flee or to fight at his own discretion. In the latter case, results such as those of Figure 7.4-10 assume significance.

7.4.8 Mini-Max Solutions with Finite Number of Parameters

COAP contains a technique for the solution of mini-max problems when a finite number of parameters are used in optimization studies. The technique is based on gradient vector magnitude minimization, Reference 4. The basic gradient vector magnitude minimization method finds ordinary extremals and mini-max points since at both classes of points the gradient vector magnitude is zero. Ordinary minima can be excluded by a sign-of-curvature correction, Reference 4, which transforms ordinary minima into singularities. The complete technique can, therefore, be used to locate selectively mini-max and/or ordinary extremals.

A simple mini-max problem is illustrated in Figure 7.4-17. Vehicle A pursues and seeks to minimize distance; Vehicle B escapes and seeks to maximize distance. Both vehicles are identical twin engine fighters. In the parameterized optimization study illustrated both fly constant angle-of-attack flight paths resulting in a two-parameter mini-max problem. Terminal separation contours are illustrated in Figure 7.4-17(b). An ordinary minimum problem exists at low angles of attack. The point obtained by multivariable search using the extended gradient vector magnitude minimization algorithm is at $\alpha_1 = 4.34°$, $\alpha_2 = 4.16°$ and is confirmed by the contour plot On this basis it appears that the gradient vector magnitude minimization procedure with sign-of-curvature correction appears to be capable of solving low dimensioned mini-max problems by multivariable search. It has yet to be applied to problems involving many parameters, however.

A final point regarding this solution should be made. The solution can be obtained more simply by fitting a set of polynomials to both vehicles' terminal altitude and ranges. In this case, the separation distance is a straightforward root square of the terminal altitude and range distance components

$$R = \sqrt{(H_2 - H_1) + (R_2 - R_1)} \qquad (7.4.2)$$

where H_1, H_2, R_1, and R_2 are polynomials in time. Locating the mini-max point of Equation (7.4.2) is a simple computation since no flight paths other than those required by the initial curve fits are to be integrated.

To verify this approach fourth-order polynomials were fitted to H_1, H_2, R_1, and R_2. The mini-max value of the resulting polynomial function was found to agree to three significant figures with that previously obtained using the full equations of motion throughout the computation. It is quite possible to extend this polynomial technique to more than two independent variables. However, to-date this approach has not been pursued further.

7.4.9 Conclusion

The basic structure and capabilities of the Combat Optimization and Analysis Program, COAP, have been outlined. The use of alternative and complementary optimization options has been demonstrated by several examples. COAP is a practical tool for assessing the combat maneuvering capability of existing and proposed aircraft. It can be used on any large scale digital computer. It requires no special hardware. In its more complex optimization modes, the user does need some familiarity with modern nonlinear optimization techniques. The program complements other approaches to the assessment of combat maneuvering capability. Promising directions of exploration for more time consuming and expensive techniques, such as flight checkout or the dual maneuvering simulator, may be defined. Clearly, the final assessment of an aircraft's effectiveness in combat and the associated tactics employed rest with the military pilot. It is hoped that COAP will be of some assistance to such pilots and will alleviate, to some extent, the high cost associated with assessing a vehicle's combat capability in action.

REFERENCES:

1. Hague, D. S., Jones, R. T., and Glatt, C. R., Combat Optimization and and Analysis Program, COAP, Volume I, Problem Formulation, AFFDL-TR-71-52, May 1971.

2. Hague, D. S., Jones, R. T., and Glatt, C. R., Combat Optimization and Analysis Program, COAP, Volume II,Progam User's Manual, AFFDL-TR-71-52, May 1971.

3. Hague, D. S., Jones, R. T., and Glatt, C. R., Combat Optimization and Analysis Program, COAP, Volume III, Programmers Manual, AFFDL-TR-71-52, May 1971.

4. Hague, D. S., Jones, R. T., and Glatt, C. R., Combat Optimization and Analysis Program, COAP, Volume IV, Parameter Optimization Subprogram, AFFDL-TR-71-52, May 1971.

5. Hutcheson, J. H. and Segerbolm, R. L., TACTICS: A Three-Body Three-Dimensional Intercept Simulation, Rand Memo RM-5759-PR, October 1969.

6. Harris, Terrell J., Jacobs, Barton J. and Strauss, Walter J., Fighter Versus Fighter Combat ATAC-2 Model, United States Air Force Contract AF33(615)-3707.

7. Campbell, T. K., Gold, T. T., Andrews, R. H., Moore, W., Application of Performance Optimization Techniques to Tactical Aircraft Operations (U), Volume II, AFFDL-TR-70-41, August 1970.

8. Brown, David A., "Simulator Aids for Aircraft Design," Aviation Week, February 7, 1972.

9. Hague, D. S. and Glatt, C. R., Optimal Design Integration of Military Flight Vehicles, ODIN/MFV, Section 7.3, "Atmospheric Trajectory Optimization Program," AFFDL-72-132, 1972.

10. Rozendaal, Harvey L., A General Branched Trajectory Optimization Algorithm with Applications to Space Shuttle Vehicle Mission Design, AAS/AAIA Astrodynamics Specialist Conference, August 17-19, 1971.

11. Landgraf, S. K., "Some Applications of Performance Optimization Techniques to Aircraft," Journal of Aircraft, March-April 1965.

12. Hague, D. S., Application of the Variational Steepest-Descent Method to High Performance Aircraft Trajectory Optimization, NASA CR-73366, 1969.

13. Hague, D. S. and Glatt, C. R., Study of Navigation and Guidance of Launch Vehicles Having Cruise Capability--Trajectory Parametric and Optimization Studies, Boeing Document D2-113016-5, 1967. (Title Unclassified, Document classified secret)

14. Hague, D. S. and Glatt, C. R., An Introduction to Multivariable Search Techniques for Parameter Optimization, NASA CR-73200, April 1968.

15. Bryson, A. E. and Denham, W. F., A Steepest-Ascent Method for Solving Optimum Programming Problems, Raytheon Report B31303.

16. Hague, D. S., et al., "Two-Vehicle Optimization--Theoretical Outline and Computer Program User's Manual. Report B983, McDonnell-Douglas Corporation, August 1965.

17. Hague, D. S., Jones, R. T., and Glatt, C. R., Combat Optimization and Analysis Program, COAP, Presented at Air-to-Air Combat Analysis and Simulation Symposium, Kirtland Air Force Base, New Mexico, February 29 to March 2, 1972. (Unclassified paper)

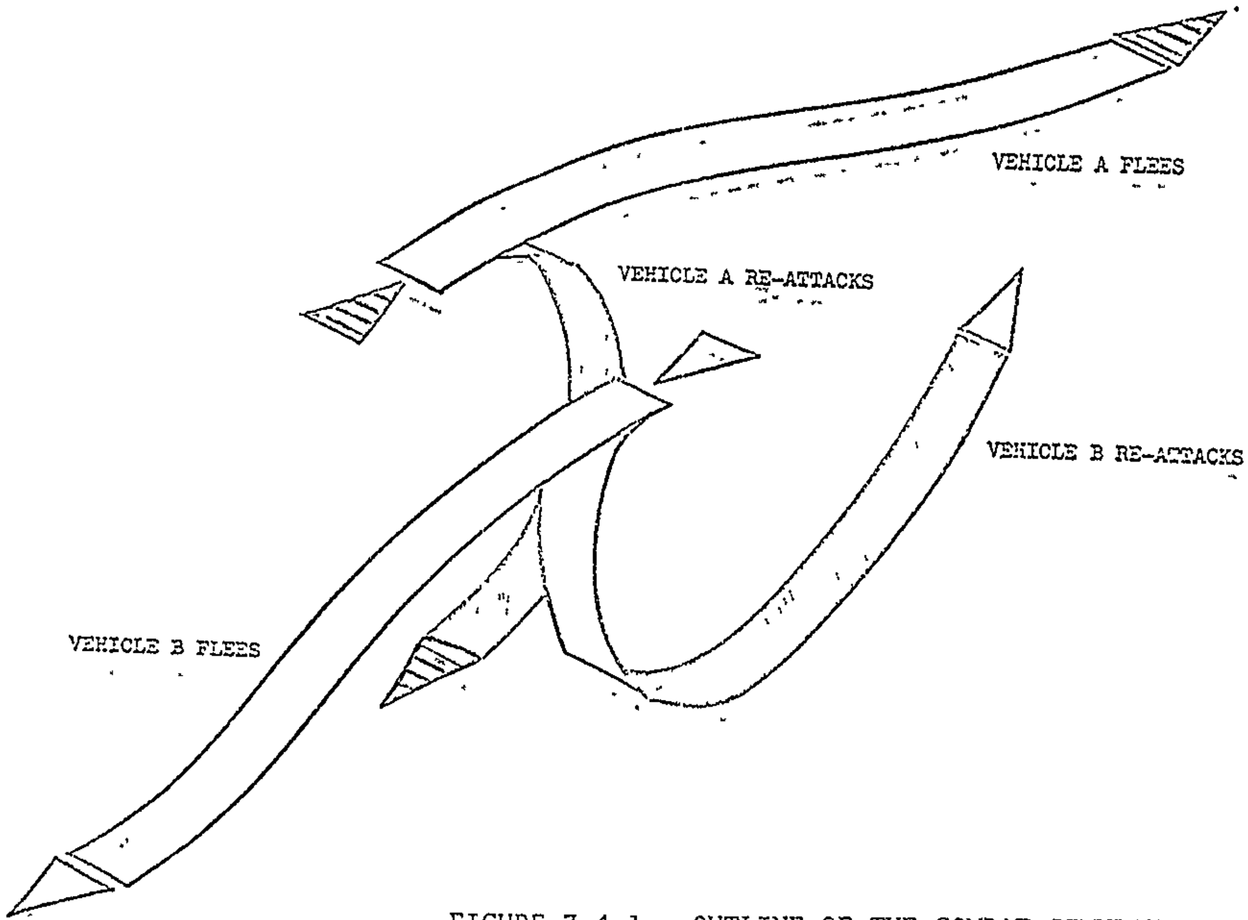


FIGURE 7.4-1. OUTLINE OF THE COMBAT SIMULATION

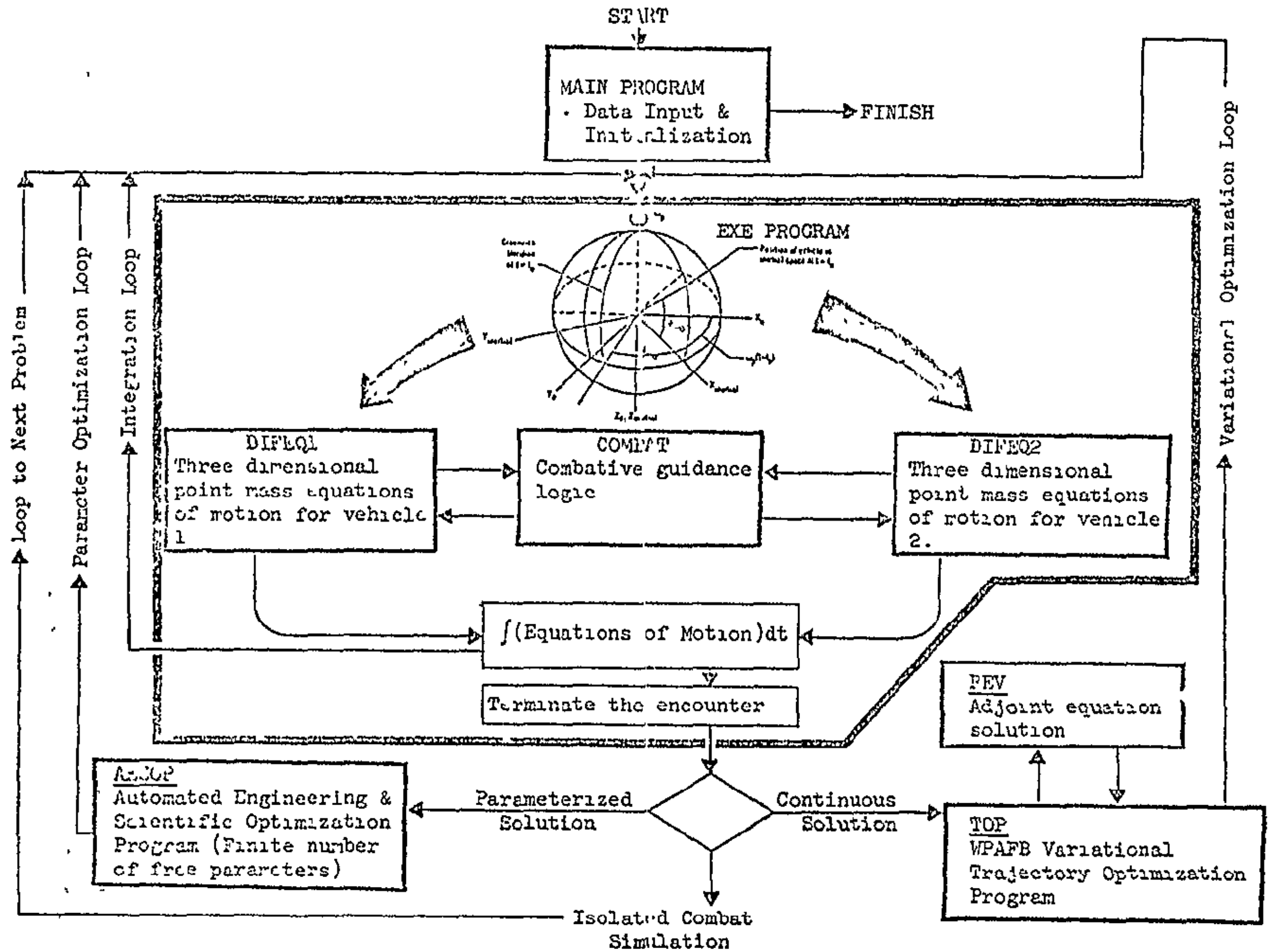


FIGURE 7.4-2. OVERALL SCHEMATIC OF COAP PROGRAM

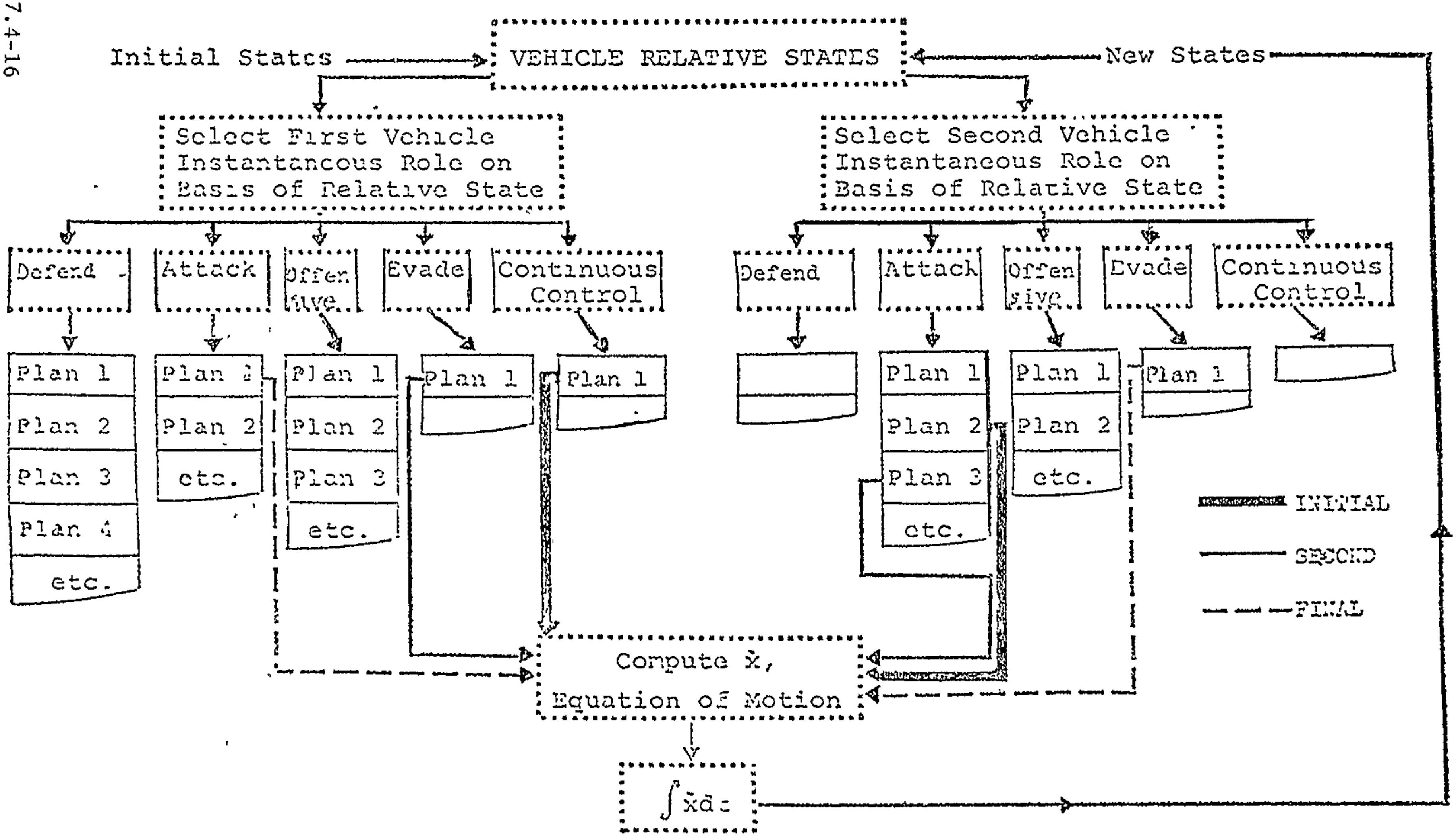


FIGURE 7.4-3. SCHEMATIC OF STATE-DEPENDENT COMBAT SIMULATION

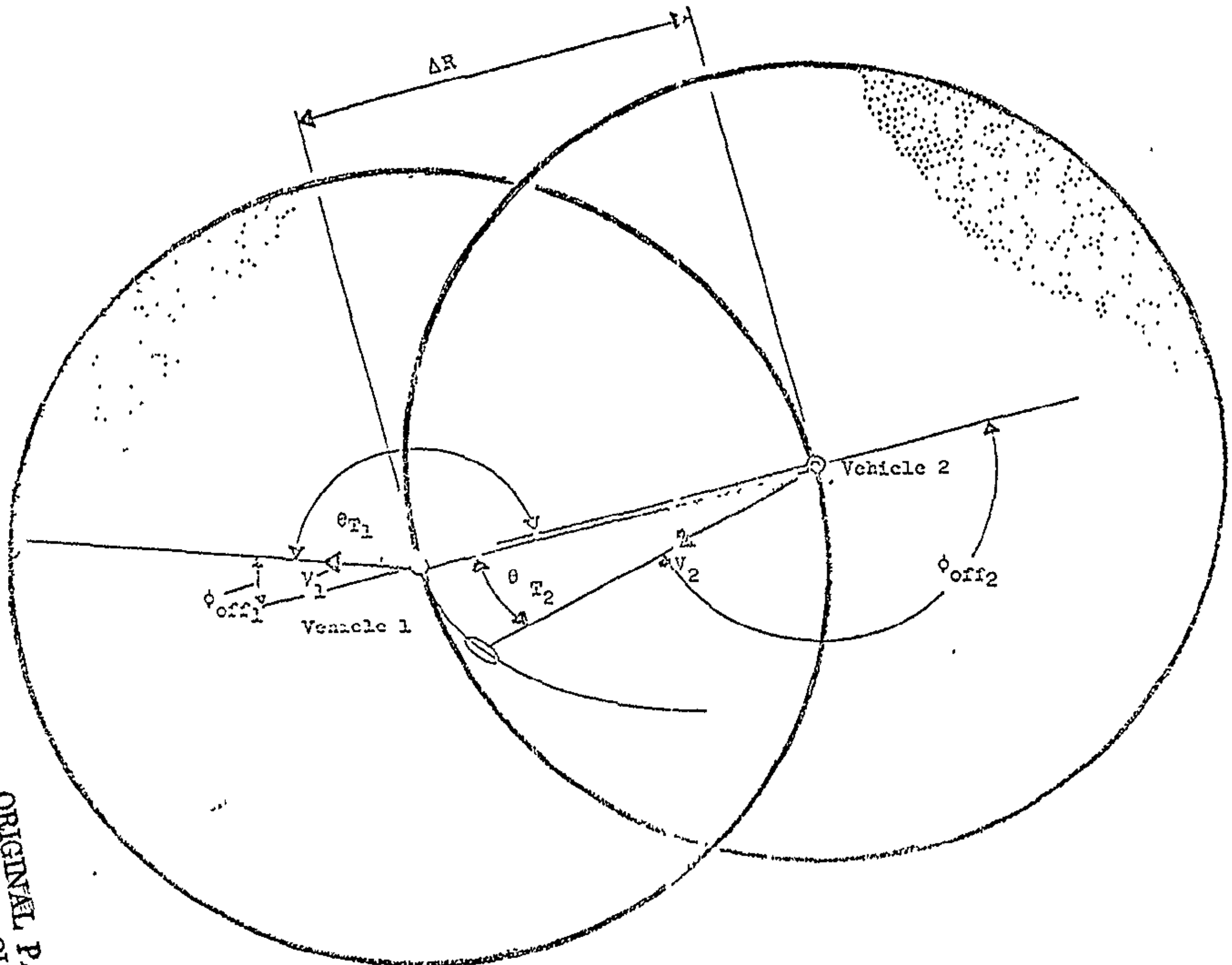


FIGURE 7.4-4. COMBAT ROLL SELECTION COORDINATES

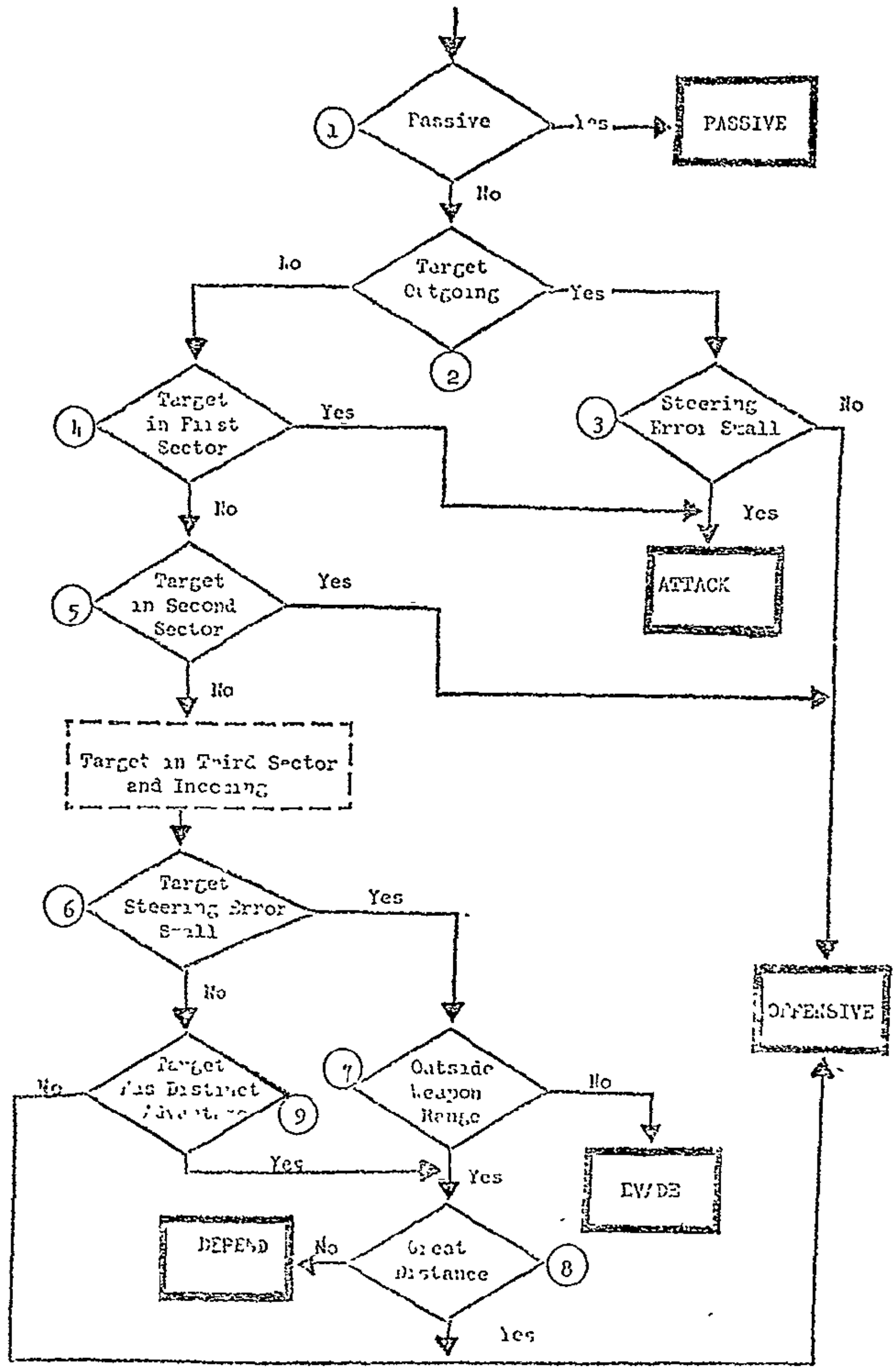


FIGURE 7.4-5. ROLE SELECTION LOGIC

DEFENSIVE ROLE.

1. Maximum Rate of Turn
2. Hard Turn
3. Maximum Opponent's Line-of-Sight Vector Rotation Rate
4. Maximum Opponent's Lead-Pursuit Vector Rotation Rate
5. Maximum Opponent's Proportional Pursuit Vector Rotation Rate
6. Split S under Opponent
7. The Maximum $\dot{E}$ Maneuver

EVASIVE ROLE:

1. Hard Pull-Up and Immelmann or Pancake
2. Random Weave
3. Split S
4. Hard Turn
5. Vertical Dive
6. Rolling

OFFENSIVE ROLE:

1. Lag Pursuit
2. Line-of-Sight Pursuit
3. Reference Vector Pursuit

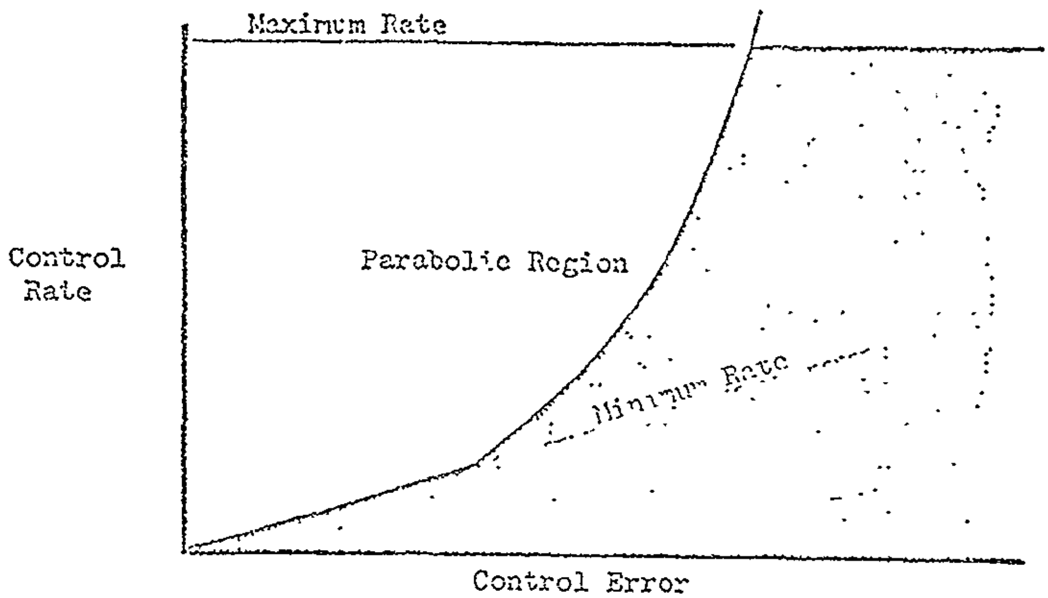
ATTACKING ROLE: Simultaneously Match

1. Rotation Induced by Target's Relative Velocity
2. Pointing Direction to Firing Point

PASSIVE ROLE:

1. Any Prespecified Maneuver

FIGURE 7.4-6. TACTIC SUMMARY



Control values at any instant of time, t, are given by

$$\alpha(t) = \alpha(t_0) + \int_{t_0}^{t} \dot{\alpha}(t)dt$$

$$B_A(t) = B_A(t_0) + \int_{t_0}^{t} \dot{B}_A(t)dt$$

$$N(t) = N(t_0) + \int_{t_0}^{t} \dot{N}(t)dt$$

FIGURE 7.4-7. FINITE CONTROL RATE OPTION

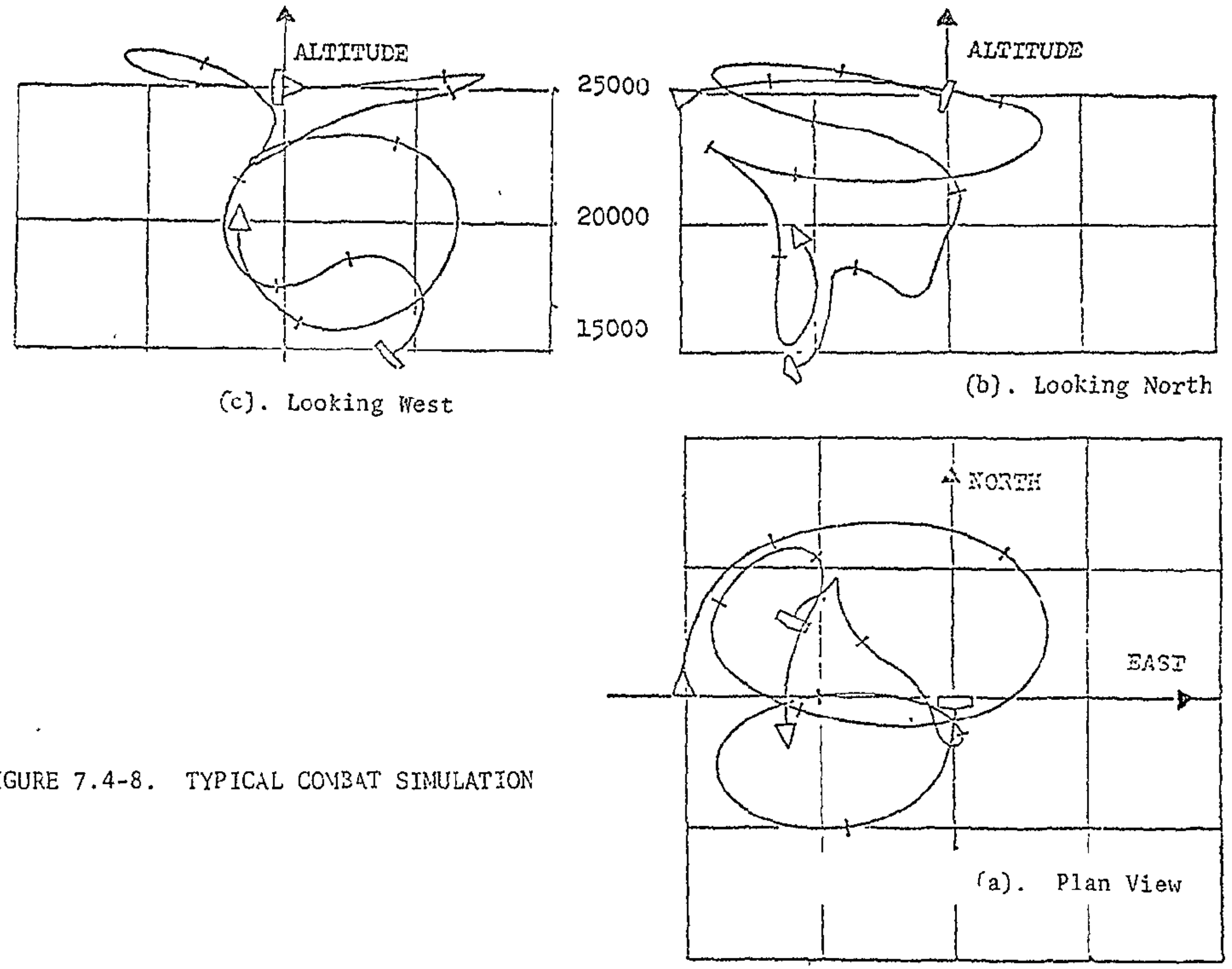


FIGURE 7.4-8. TYPICAL COMBAT SIMULATION

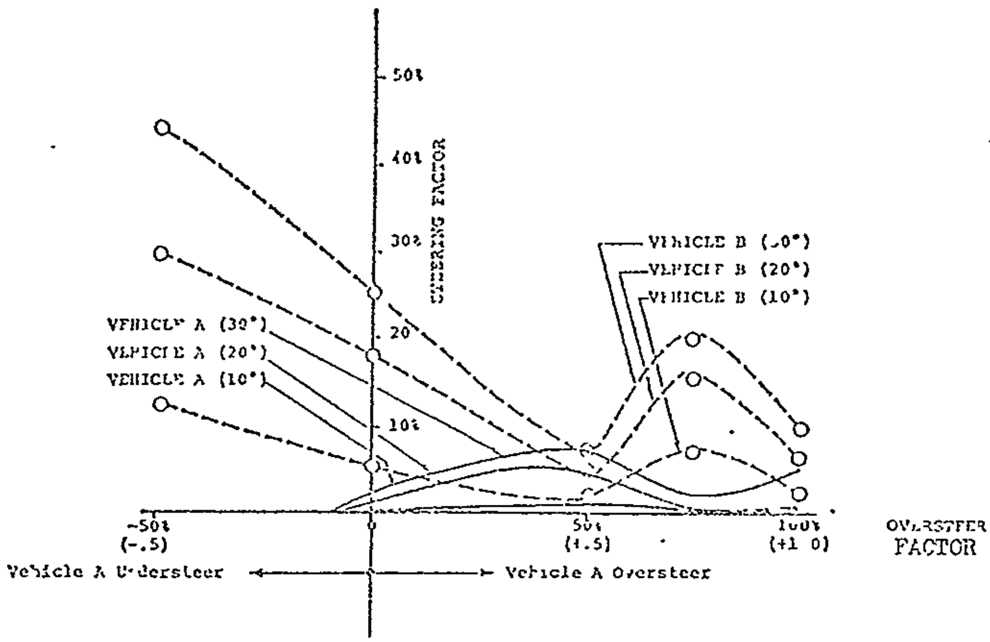


FIGURE 7.4-9. STEERING CAPABILITIES

Vehicle B Uses Neutral Steering
(Flight Time 400 Seconds)

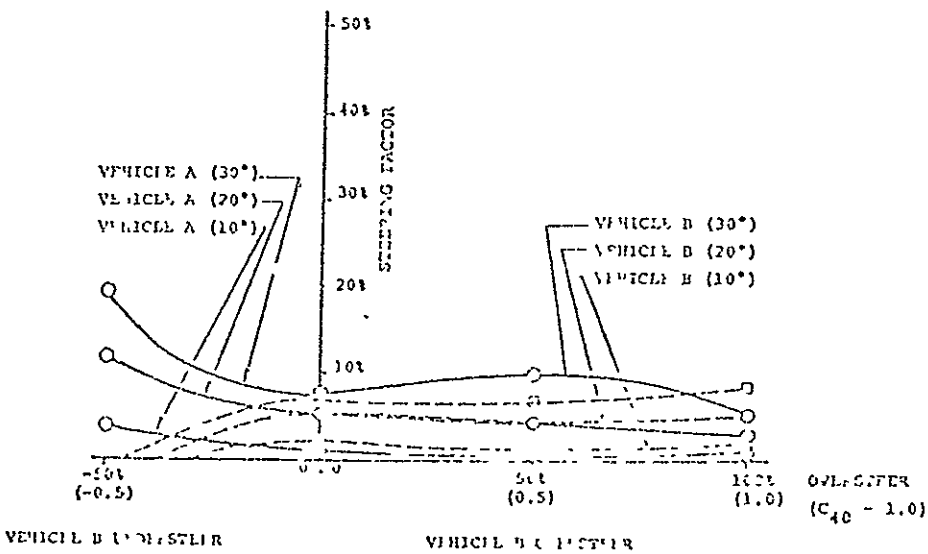


FIGURE 7.4-10. STEERING CAPABILITIES

Vehicle A Uses 50% Oversteer
(Flight Time 400 Seconds)

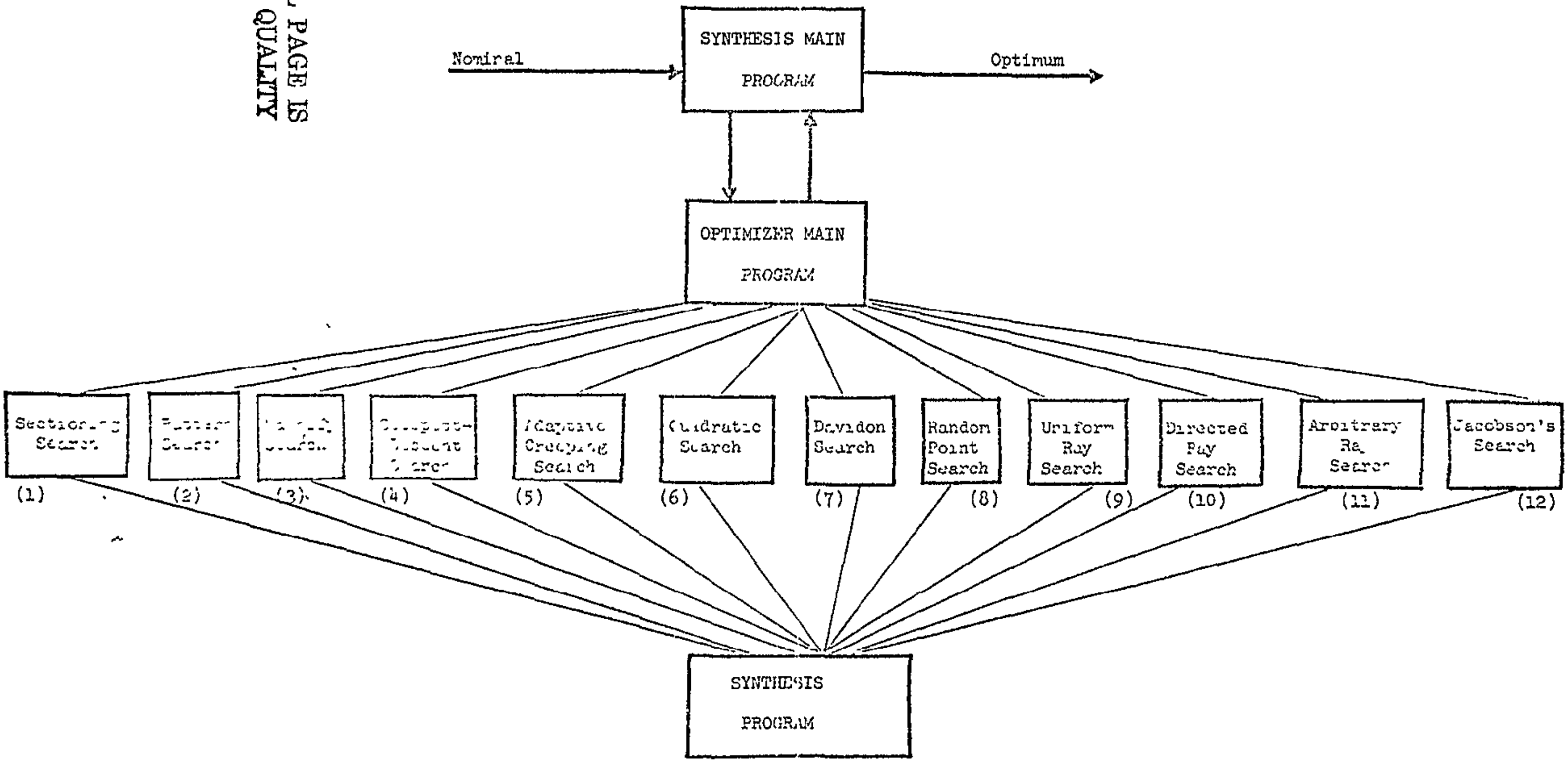


FIGURE 7.4-11. SCHEMATIC OF OPTIMIZATION PROGRAM - AESOP

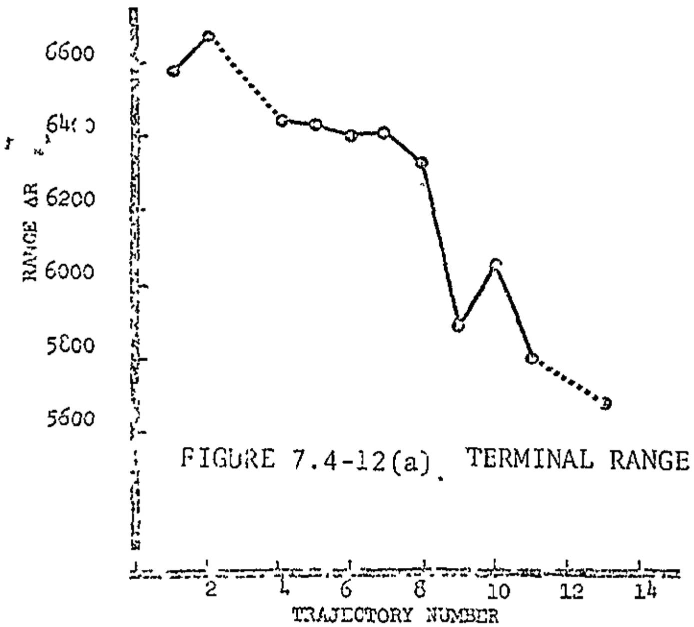


FIGURE 7.4-12(a). TERMINAL RANGE

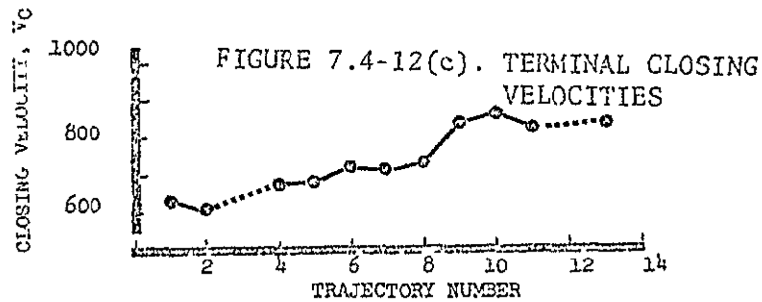


FIGURE 7.4-12(c). TERMINAL CLOSING VELOCITIES

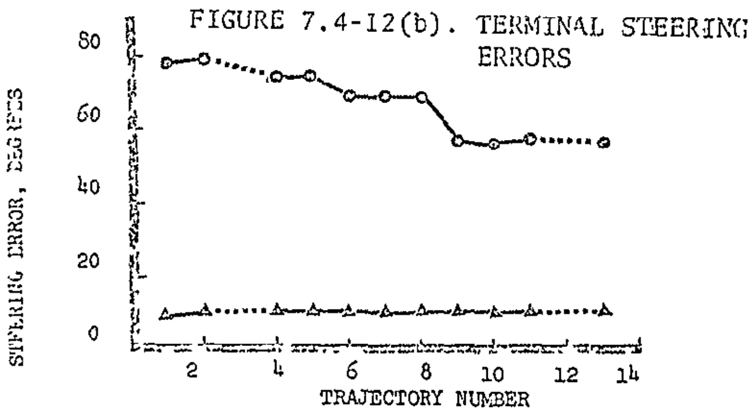


FIGURE 7.4-12(b). TERMINAL STEERING ERRORS

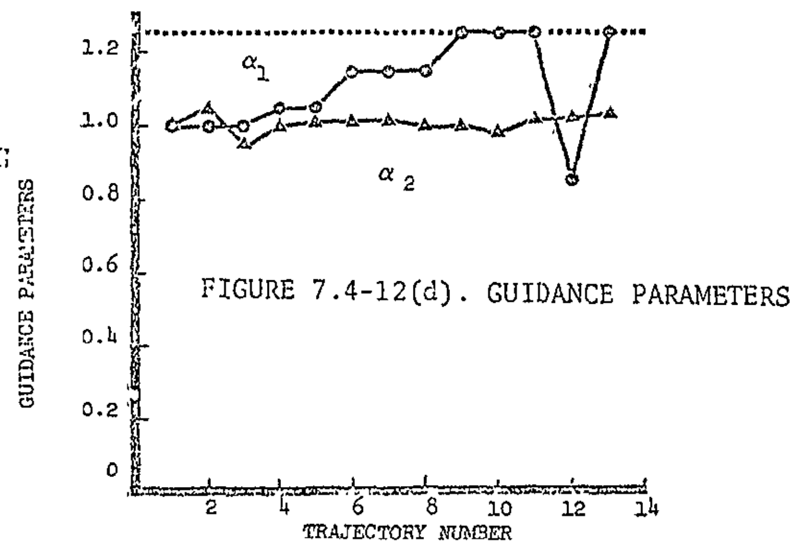


FIGURE 7.4-12(d). GUIDANCE PARAMETERS

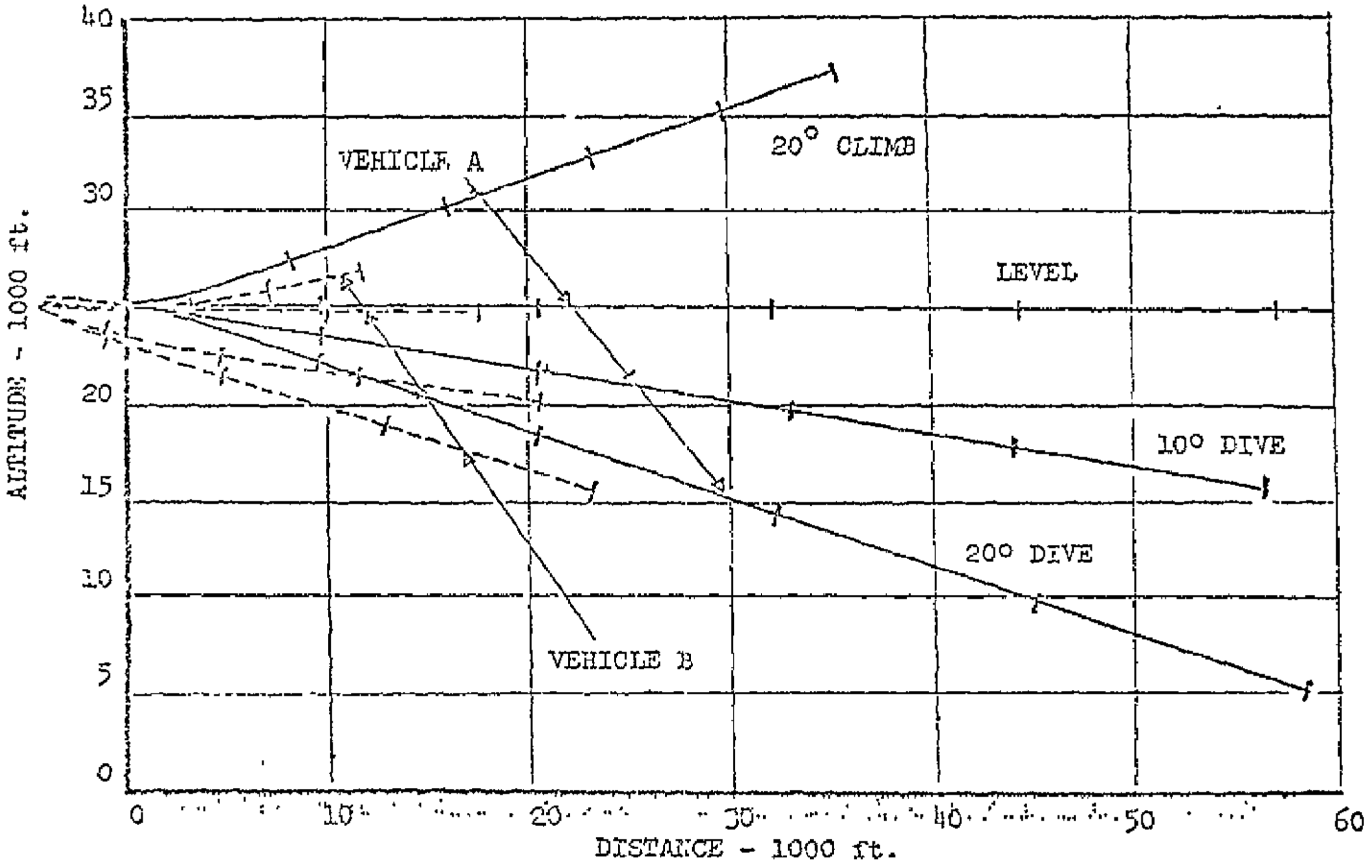


FIGURE 7.4-13. CONSTANT FLIGHT PATH ANGLE ESCAPES

ORIGINAL PAGE IS
OF POOR QUALITY

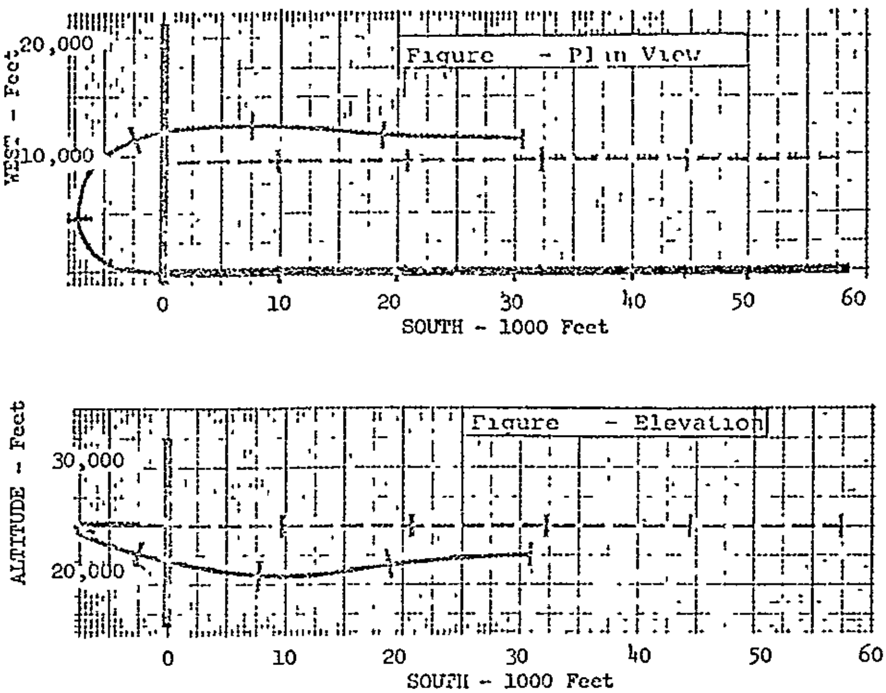


FIGURE 7.4-14. THREE PARAMETER REATTACK

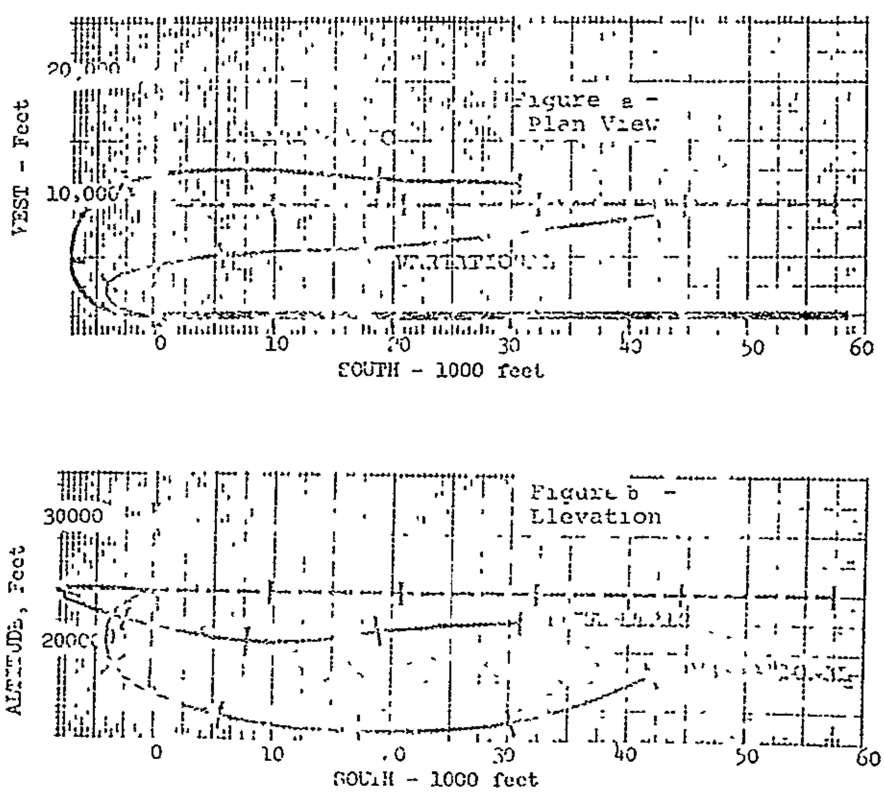


FIGURE 7.4-15. VARIATIONAL MANEUVERING TARGET SOLUTION AND COMPARISON WITH PARAMETERIZED SOLUTION

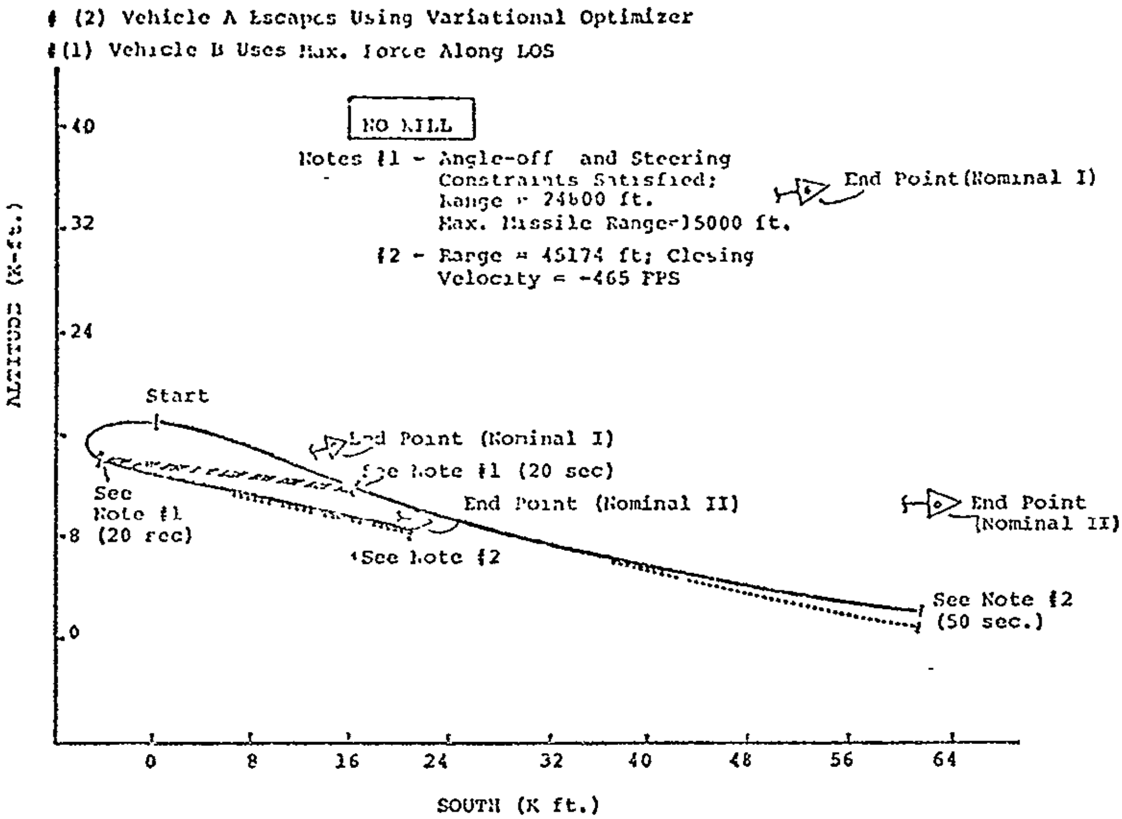
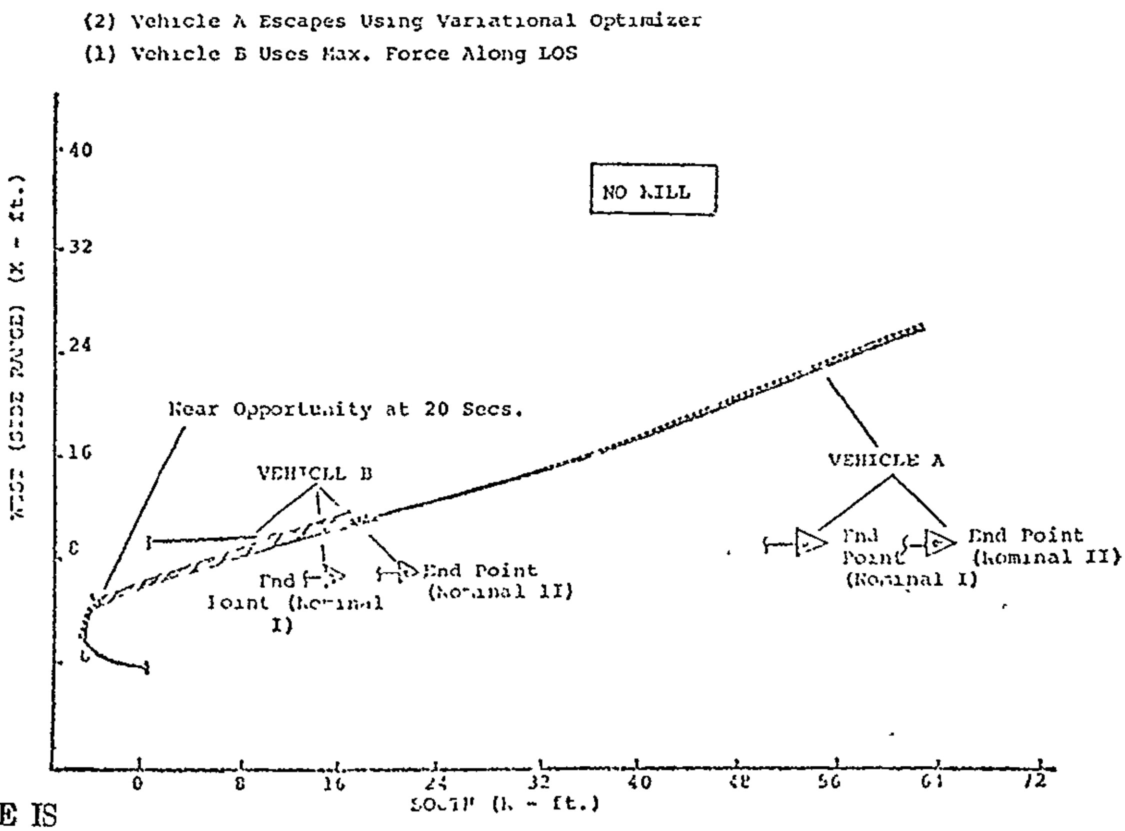


FIGURE 7.4-16(a). VARIATION REACTING OPPONENT SOLUTION--ELEVATION



ORIGINAL PAGE IS
OF POOR QUALITY

FIGURE 7.4-16(b). VARIATIONAL REACTING OPPONENT SOLUTION - PLAN VIEW

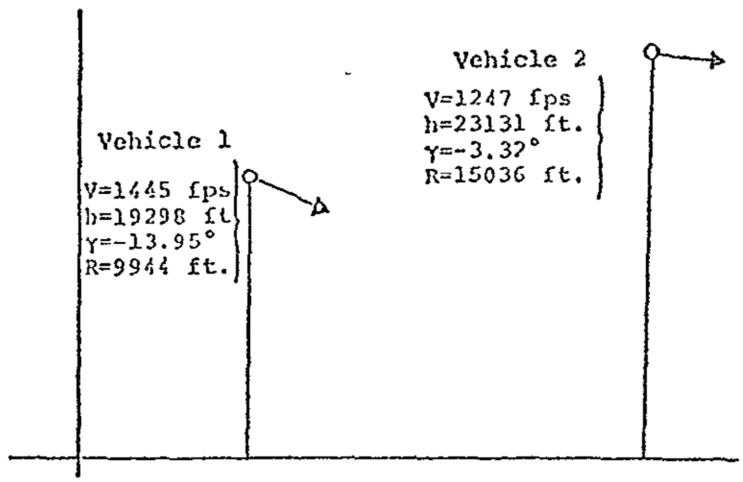


FIGURE 7.4-17(a) INITIAL CONDITIONS

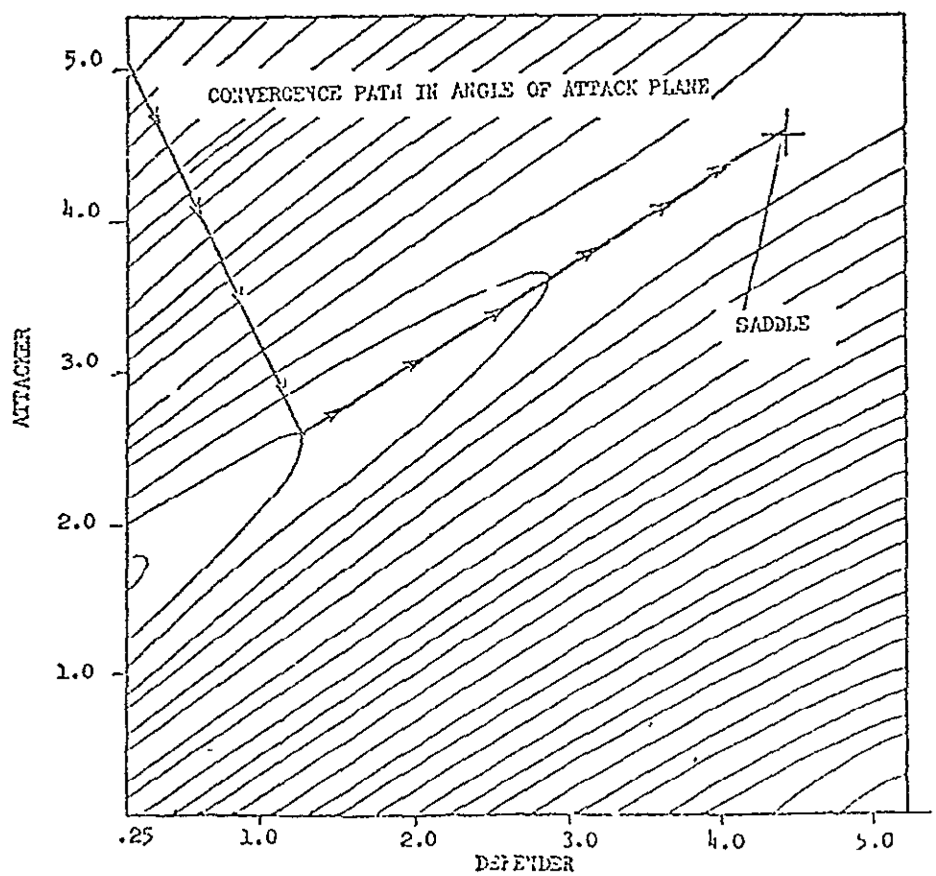


FIGURE 7.4-17(b). CONTOURS OF TERMINAL SEPARATION DISTANCE

SECTION 8

STRUCTURES

The ODIN/RLV program as installed at Langley Research Center restricts the structural analysis to a steady-state swept wing aeroelastic analysis. Engineers bending and torsion analysis about a swept elastic axis is combined with the subsonic lifting line aerodynamic analysis. Fuselage lift and moment is accounted for as is the need for a balancing tail load. Finally, the required stiffness distributions are converted to a wing box structural weight assuming conventional wing structures.

TABLE OF CONTENTS FOR SECTION 8.1, PROGRAM SSAM

8.1 PROGRAM SSAM: SWEPT STRIP AEROELASTIC MODEL

8.1.1 General Method of Analysis

Program SSAM performs an aeroelastic evaluation of the wing spanwise flight loads including the complete aircraft balance for a specified set of steady state maneuvers and/or design gust conditions. Here, proper inclusion of the wing, body, and nacelles' aerodynamic and weight effects are included in order to compute the required balancing tail load which is reflected in the wing load calculation. Figure 8-1 shows a schematic of the system of equations used which include as unknowns ten span loads along with the airplane root angle of attack and balancing tail load.

These flight loads including the aerodynamic and wing dead weight loads are then converted into the structural wing box bending and torsion loads to evaluate the resulting bending and torsional stresses. If the calculated wing stresses exceed the allowable wing stresses, a new set of values of wing section stiffness values are selected to match the allowable stress distribution specified within the program data. The wing aeroelastic load solution is then repeated until the calculated and allowable wing stresses are matched. This type of analysis is necessary for a swept elastic wing as the airfoil section angle of attack depends upon the wing bending and torsion deflections. The cycling process is fast and usually requires three to five cycles to converge depending upon the error margin set within the program. The program then computes the wing box weight based on the final set of stiffness values obtained. The resulting wing will not exceed the allowable stress distributions for the specified set of load conditions. At the present time this analysis is limited to subsonic flight conditions.

8.1.2 Aerodynamic Representation of Flight Loads and Aeroelastic Analysis

The wing flight loads are evaluated considering the wing as a finite number of panels of width 2h. These panel strips are taken parallel to the air stream. Using Weissinger's aerodynamic theory each panel contains a horseshoe vortex representation as shown in Figure 8.1-2. The circulation strength Γ_n of each vortex is related to the unknown span loading ℓ which is assumed constant over each element. For each wing panel the sum of all vortex downwash velocities must be summed such that

$$-\sum_{0}^{n} \left(\frac{w}{v}\right)_n = \alpha f_n \tag{8.1.1}$$

where w/v is the induced downwash angle of the three-quarter chord and αf is the section airfoil free air angle of attack. In matrix notation

$$\left\{\frac{w}{v}\right\}_{3c/4} = \frac{1}{4\pi v} [S_1]\{\Gamma\} \tag{8.1.2}$$

leading to the basic equation

$$\left[\frac{\Gamma^\circ}{4 q m_o}\right] [S_1] \{\ell\} = \{\alpha_f\} \tag{8.1.3}$$

where

$\left[\frac{\Gamma^\circ}{4 q m_o}\right]$ = square matrix containing only the diagonal terms shown

$[S_1]$ = square matrix representing the vortex wing geometry

$\{\ell\}$ = column matrix of the unknown span loading

$\{\alpha_f\}$ = column matrix of the free airfoil section angles of attack

The free airstream angle of attack α_f is composed of several components which must be introduced into (8.1.3). Here

$$\alpha_f = \alpha_s + \alpha_r + \alpha_g \tag{8.1.4}$$

where

α_s = change due to aeroelastic wing loads

α_r = A/p wing root, α defined by load factor and balancing A/p tail load

α_g = geometric wing twist includes flight control deflection and wing dead weight effects

When a swept wing deflects, wing bending along with the wing twist produced by the air loads causes the streamwise airfoil section angle of attack to change. A general expression for the section angle of attack change is given by the integral relation,

$$\alpha_{s_\sigma} = \int_0^\sigma \frac{m\,M ds}{EI} + \int_0^\sigma \frac{t\,T ds}{GJ} \tag{8.1.5}$$

where

m = beam bending moment per unit pitching moment

t = beam torsion moment per unit pitching moment

M = applied bending moment

T = applied torsion moment

EI, GJ = beam section stiffness characteristics

Equation (8.1.5) may be integrated from the wing tip to each of the wing stations to produce the following equation for α_S:

$$\{\alpha_s\} = [S_2]\ \{\ell\} \tag{8.1.6}$$

where
[S2] = wing aeroelastic deflection matrix containing wing geometry and section stiffness properties

The airplane wing root angle of attack α_r is calculated by balancing the external loads on the aircraft in terms of the aircraft flight condition. For the case of maneuvering flight, the wing lift is the sum of $nW + P_T$ where

n = flight maneuver load factor
W = aircraft gross weight
P_T = aircraft balancing tail load

The wing lift is expressed as a number of section lift values ℓ_n which so far are unknown quantities. To balance, the aircraft body loads must be included. In the case of the aircraft body, these effects are assumed to be known and are expressed as

$$\begin{aligned} L_{FUSELAGE} &= qS[C_{LF_O} + (C_{LF})_\alpha\ \alpha_r] \\ M_{FUSELAGE} &= qS\bar{c}[C_{mF_O} + (C_{mF})_\alpha \alpha_r] \end{aligned} \tag{8.1.7}$$

where

C_{LF_O}, C_{mF_O} = aerodynamic C_L and C_m for $\alpha_r = 0$

(C_{LF}) , (C_{mF}) = aerodynamic C_L and C_m variation with α_r

The last wing angle of attack component is the wing geometric twist α_g. This includes effects of

1. change in the airfoil zero lift angle of attack due to using different airfoil section

2. change in the airfoil zero lift angle of attack due to flight control deflection

3. built in wing twist

4. twist due to aircraft wing dead weights

From Equations (8.1.3), (8.1.4), (8.1.6) and (8.1.7), a system of N+2 linear equations may be written which express as unknowns N values of the span lift along with the airplane root angle of attack α_r and balancing tail load P_T.

It should be noted that program SSAM allows the following airfoil section characteristics to be specified for each wing panel:

m_o = slope of lift coefficient

ac = airfoil aerodynamic center position

α_o = section zero lift angle

$C_{m_{ac}}$ = section pitching moment coefficient

The wing section combined effective allowable stress σ_A is also specified within the SSAM program. Knowing this value the desired wing box section properties may be calculated based upon the allowable bending stress for the front spar, rear spar, and maximum spar depth. These areas are then averaged with weighting factors, if desired, to produce the desired wing section properties. The program may be employed to define elastic aerodynamic rolling derivatives by appropriate data input.

8.1.3 Wing Dead Weights

The wing dead weights are represented as a set of concentrated loads for each wing panel. These weights and their position coordinates must be specified with reference to the airframe. The aircraft c.g. must also be specified with respect to the wing MAC. External stores such as the nacelles must have their weight and c.g. locations defined. Wing fuel weights may be calculated within the program.

Given a set of j dead weights associated with each panel i, the user may select which of the j dead weight sets are to be summed for each specified load condition by simple inputs. That is, for any load condition, ℓ

$$\{\Delta W_i\}_\ell = k_1\{\Delta W_i^1\} + k_2\{\Delta W_i^2\} + \ldots k_n\{\Delta W_i^n\} \qquad (8.1.8)$$

where

k_j = 0 or 1.

This feature permits the rapid assembly of load conditions at partial fuel loads, for example, by assigning various partial fuel load conditions as particular $\{\Delta W_i^j\}$.

8.1.4 Wing Load Calculation

The aerodynamic air loads are calculated based upon streamwise wing panel strips as shown in Figure 8.1-3. The program converts these airloads along with the corresponding dead weight loads into the required wing bending and torsion loads along the swept elastic axis.

The wing box torsion beam is defined by a front and rear spar location and an elastic axis position. This elastic axis may be arbitrarily selected and its position defined by a wing sweep angle and chord location at the center of each wing panel. Usually, the elastic axis is selected as a fraction of the distance between the front and rear wing spars. For a swept wing a fairing of this axis is possible near the wing root as shown in Figure 8.1-3.

8.1.5 Calculation of Wing Stresses

In the program SSAM the wing bending torsion box beam cross section is defined as shown in Figure 8.1-4. Here, a front and rear spar location is defined. The wing depths are then specified in the program input at each station. These depths are input as a front spar depth, rear spar depth, and the maximum wing thickness depth. In addition, the distance between the front and rear spar is specified normal to the elastic axis.

The wing cross section is treated symetrically. The upper and lower beam bending material are treated as equal areas. A factor K_o is estimated, based upon the type of wing structure, to define the portion of the structural material area used as stringer area and the portion to be treated as skin material area. Other small correction dimensions are input into the program to allow for the stringer centroid location and average box depth.

Figure 8.1-5 shows a typical wing box cross section with the represented section dimensions, where

w = distance between front and rear wing spar
d = maximum box depth
d_a = average box depth
d_e = effective beam depth for bending
t_s = average skin thickness
A_o = total material area of one segment of bending material
K_o = skin segment area/total segment area = $t_s w/A_o$.

The general equation for representing the combined maximum wing stress in terms of the material area A_o will be developed. In terms of the above dimensions, the wing section bending moment of inertia I and the torsion box area A_B may be written as

$$I = A_o \, d_e^2/2.0$$

$$A_B = d_a w$$

The average skin thickness in terms of A_o is

$$t_s = k_o A_o / w \qquad (8.1.9)$$

The maximum wing bending and torsion stress in terms of the applied bending moment M and torsion moment T becomes

$$\sigma_b = \frac{My}{I} = \frac{Md}{A_o d_e^2} \qquad (8.1.10)$$

$$\sigma_t = \frac{T}{2A_B t_s} = \frac{T}{2A_o K_o da} \qquad (8.1.11)$$

The shear loads V also produce a shear stress, σ_{sw}, in the spar webs, Here,

$$\sigma_{sw} = \frac{V}{d_e t_s} = \frac{Vw}{d_e K_o A_o} \qquad (8.1.12)$$

The maximum combined principal stress from basic structure considerations is expressed as

$$\sigma = \frac{\sigma_b}{2} + \sqrt{\left(\frac{\sigma_b}{2}\right)^2 + (\sigma_{sw} \pm \sigma_t)^2} \qquad (8.1.13)$$

substituting Equations (8.1.10) (8.1.11), and (8.1.12) into Equation (8.1.13) gives the following expression for the maximum wing stress in terms of the material segment area A_o,

$$\sigma = \frac{1}{2A_o}\left[\frac{Md}{d_e^2} + \sqrt{\left(\frac{Md}{d_e^2}\right)^2 + \frac{w^2}{K_o^2}\left(\frac{V}{d} \pm \frac{T}{wd_a}\right)^2}\,\right] \qquad (8.1.14)$$

The wing section combined effective allowable stress σ_A is specified within the program and may be entered for each station element if desired. Knowing this value the desired wing box section properties may be calculated in terms of A_o where

$$A_o = \frac{1}{2\sigma_A}\left[\frac{Md}{d_e^2} + \quad \left(\frac{Md}{d_e^2}\right)^2 + \frac{w^2}{K_o^2}\left(\frac{V}{d} \pm \frac{T}{wd_a}\right)^2\right] \qquad (8.1.15)$$

The above general equation is used with the proper sign to calculate A_o based upon the allowable bending stress for the front spar, rear spar, and maximum spar depth. These areas are then averaged with weighting factors, if desired, to produce the desired wing section properties.

Because the upper and lower skin and stiffener areas are considered equal, the allowable stress σ_A represents an average between the allowable compression stress and the allowable tension stress. The allowable tension stress is generally constant while the allowable compression stress depends upon the type of wing construction which may vary spanwise along the wing. For this reason the program allows for specifying this stress at each wing panel. As the wing is designed for the ultimate load, the values of M, T, and V used in Equation (8.1.14) are all increased by a factor of 1.50 within the program.

8.1.6 Program Cycling

An aeroelastic solution implies that the wing twist affects the wing airloads which, in turn, determines the wing section properties. The wing structural angle of attack change is defined by Equation (8.1.6). Here, the S_2 matrix depends upon the wing geometry and wing flexibility as defined in Appendix B of Reference 2 giving

$$[S_2] = f_1\left(\begin{array}{c}\text{wing planform}\\ \text{geometry}\end{array}\right)\left[\overset{o}{\frac{1}{EI}}\right] + f_2\left(\begin{array}{c}\text{wing planform}\\ \text{geometry}\end{array}\right)\left[\overset{o}{\frac{1}{GJ}}\right]$$

The program is first cycled by assuming a distribution of 1/EI and 1/GJ values, if available. If approximate stiffness values are not available, zero values are used which correspond to having a rigid wing.

The program then computes airloads for the first flight condition. From these loads, the minimum wing section areas A_o given by Equation (8.1.15) are calculated for each wing panel based upon the allowable stress σ_A. This process is repeated for each of the other specified flight conditions. The program internally saves the maximum required value of A_o for each wing station. This is a direct cycling following the normal procedure for the analysis of a wing.

After all the input flight conditions are cycled, the minimum bending material area A_o to meet the design requirements for a zero margin of safety wing at each wing station will be known. The airloads, however, were based upon different values of 1/EI and 1/GJ and, hence, the cycling process must be repeated.

The wing iteration process now begins by using the calculated A_o values to compute a new S_2 matrix. Here

$$\left[\frac{1}{EI}\right] = \frac{A_o d_c^{\,2}}{2E}$$

$$\left[\frac{1}{GJ}\right] = \frac{1.0}{\left[\frac{d_{FS}+d_{RS}}{2t_w}\right]\left(\frac{d_a}{w}\right)^2 G}$$

where

$t_s = K_o A_o/w$

$t_w = K_w t_s$

The wing aeroelastic properties affect perhaps twenty per cent of the air load values, hence, generally three to five wing iterations of 1/EI and 1/GJ are necessary. After each iteration, the new required stiffness values are compared with the previous stiffness values at each station until they all agree within a specified margin; usually this margin is taken to be two per cent.

After the error margin is reached, the final wing box weight is calculated, and loads for each of the flight conditions based upon the final wing stiffness are determined. A diagram of this cycling process is shown in Figure 8.1-6.

8.1.7 Conclusion

An outline of Program SSAM has been presented. More complete details of the Weissinger aerodynamic analysis are given in Reference 1. The aeroelastic analysis is described in detail in Reference 2. A typical application of this program is provided by Reference 3. Typical program output is presented in Tables 8.1-1(a) through 8.1-1(e). It should be noted that a companion flutter program employing unsteady aerodynamics, Reference 4, is available for inclusion in ODIN simulations.

REFERENCES:

1. Weissinger, J., The Lift Distribution of Swept Back Wings, NACA TM-1120, 1947.

2. Gray, W. L., Schenk, K. M., Method for Calculating the Subsonic Steady State Loading on an Airplane with a Wing of Arbitrary Planform and Stiffness, NACA TN-3030, 1953.

3. White, Roland J., "Improving the Airplane Efficiency by Use of Wing Maneuver Load Alleviation," *Journal of Aircraft*, October 1971.

4. Phoa, Y.T., A Computerized Flutter Solution Procedure, National Symposium on Computerized Structural Analysis and Design, George Washington University, March 28, 1972. (available from Aerophysics Research Corporation).

TABLE 8.1-1(a). INITIAL GUESS AT WING FLEXIBILITIES

```
                    S S A M - SWEPT STRIP AEROFLASTIC MODEL              PAGE  1
                       AEROPHYSICS RESEARCH CORPORATION


     THE ESTIMATED WING FLEXIBILITY FOR THE FIRST
ITERATION IS AS FOLLOWS.

                    ETA         BFLEX         TFLEX

                   .9500      .363460       .523520
                   .8500      .143060       .183250
                   .7500     6.171000E-02 9.539000E-02
                   .6500     3.152000E-02 4.923000E-02
                   .5500     1.991000E-02 3.226000E-02
                   .4500     1.187000E-02 1.661000E-02
                   .3500     7.440000E-03 1.051000E-02
                   .2500     4.600000E-03 6.990000E-03
                   .1500     2.560000E-03 3.270000E-03
                5.0000E-02 1.990000E-03 0.

WHERE -
ETA      WING STATION (SPAN FRACTION)
BFLEX    WING FLEXIBILITY IN BENDING (10E9/IN-LB)
TFLEX    WING FLEXIBILITY IN TORSION (10E9/IN-LB)


     THE FOLLOWING DEFINITIONS ARE PROVIDED TO AID
IN INTERPRETING THE OUTPUT FROM INDIVIDUAL LOAD CASES.

DIST     LIFT DISTRIBUTION (1000 LB/IN)
CL       SECTION LIFT COEFFICIENT
CM       SECTION PITCHING MOMENT COEFFICIENT
ALP-F    ANGLE OF ATTACK OF ZERO LIFT LINE (DEG)
ALPHA    ANGLE OF ATTACK OF CHORD LINE (DEG)
SHEAR    SHEAR ALONG THE ELASTIC AXIS (1000 LB)
MOM      BENDING MOMENT ALONG THE ELASTIC AXIS(10E6 IN-LB)
TOR      TORSION ABOUT THE ELASTIC AXIS (10E6 IN-LB)
DEFL     DEFLECTION OF THE ELASTIC AXIS (IN)
```

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE 8-1-1(b). FINAL DEFLECTED WING, LOAD CASE 1

FLIGHT LOAD CONDITION 1.

ALTITUDE (FT)	0.
EQUIVALENT AIRSPEED (KT)	230.00
MACH NUMBER	.42400
COMPRESSIBILITY CORRECTION FACTOR	1.0920
HORIZONTAL TAIL ARM (IN)	751.00
SPOILER/FLAP DEFLECTION (DEG)	-0.
FUEL LOADING INDICATOR (PERCENT)	80.000
GROSS WEIGHT OF AIRCRAFT (1000 LB)	310.00
LOAD FACTOR	2.5000
CENTER OF GRAVITY (MAC FRACTION)	.16900
AIR DENSITY RATIO	1.0000
CENTERLINE AIRLOADS (ONLY)	
SHEAR (1000 LB)	403.62
MOMENT (10E6 IN-LB)	135.95
TORSION (10E6 IN-LB)	-8.3814
CENTERLINE SECTION DATA	
AIR PRESSURE (PSI)	9.4200
FUEL DENSITY (LB/GAL)	6.9000
CARGO PRESSURE (PSI)	.42000

ETA	DIST	CL	CM	ALP-F	ALPHA	SHEAR	MOM	TOR	DEFL
.950	.183	.8[illegible]2	-.0044	8.620	7.675	7.08	.17	.01	96.22
.850	.267	.941	-.0238	9.473	8.017	22.98	1.54	.13	74.03
.750	.345	1.055	-.0266	10.392	8.673	43.58	4.69	.33	54.15
.650	.406	1.132	-.0283	1[illegible].5[illegible]8	9.607	51.44	9.69	-.94	37.60
.550	.447	1.1[illegible]8	-.0293	12.232	10.639	75.26	15.71	-.39	24.63
.450	.500	1.1[illegible]9	-.0333	13.[illegible]26	11.610	83.50	25.17	-1.60	14.63
.350	.56[illegible]	1.1[illegible]3	-.0461	13.77[illegible]	12.4[illegible]1	107.58	33.81	-1.20	7.70
.250	.621	1.[illegible]71	-.0320	14.417	13.134	127.54	44.83	-2.32	3.38
.150	.665	1.[illegible]27	-.1303	14.857	13.573	150.21	56.23	-5.53	1.06
.050	.692	.966	-.1432	15.056	13.772	194.08	69.30	-17.55	0.00
.082	SIDE OF BODY LOADS					179.91	65.08	-13.67	
.082	STREAMWISE LOADS						59.84	-29.01	

AIRPLANE LIFT CURVE SLOPE (PER DEG)	7.01759E-02
BALANCING TAIL LOAD (LB/G)	-8496.2
GUST VELOCITY IF USED (FT/SEC)	-0.
THRUST PER ENGINE (10000 LB)	-0.
AIRPLANE LIFT COEFFICIENT	1.0083
MAGNIFICATION FACTOR (FOR GUST)	-0.

LOAD FACTOR	2.5000	0.
WING ANGLE OF ATTACK (DEG)	13.636	-.53278
BALANCING TAIL LOAD (LB/G)	-38397	-17157
AIRPLANE LIFT COEFF. (TAIL-OFF)	1.0583	2.2322[illegible]E-02
PITCHING MOMENT COEFF. (0.25 MAC)	-5.01116E-02	-6.1566[illegible]E-02

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE 8.1-1(c) FINAL DEFLECTED WING, LOAD CASE 12

FLIGHT LOAD CONDITION 3A

ALTITUDE (FT)	30000
EQUIVALENT AIRSPEED (KT)	342.00
MACH NUMBER	.95000
COMPRESSIBILITY CORRECTION FACTOR	1.3000
HORIZONTAL TAIL ARM (IN)	751.00
SPOILER/FLAP DEFLECTION (DEG)	-0.
FUEL LOADING INDICATOR (PERCENT)	71.900
GROSS WEIGHT OF AIRCRAFT (1000 LB)	301.90
LOAD FACTOR	2.5000
CENTER OF GRAVITY (MAC FRACTION)	.16200
AIR DENSITY RATIO	.37410

CENTERLINE AIRLOADS (ONLY)

SHEAR (1000 LB)	413.83
MOMENT (10E6 IN-LB)	133.87
TORSION (10E6 IN-LB)	-22.495

CENTERLINE SECTION DATA

AIR PRESSURE (PSI)	9.4200
FUEL DENSITY (LB/GAL)	6.9000
CARGO PRESSURE (PSI)	.42000

ETA	DIST	CL	CM	ALP-F	ALPHA	SHEAR	MOM	TOR	DEFL
.950	.154	.453	.0191	3.402	2.457	5.84	.12	.04	69.90
.850	.233	.573	-.0050	4.054	2.549	19.05	1.21	.20	70.46
.750	.31[illegible]	.658	-.0207	4.693	2.974	36.90	3.83	.36	52.58
.650	.373	.697	-.0619	4.606	3.695	44.29	8.18	-1.03	37.19
.550	.438	.729	-.1373	6.295	4.642	68.33	13.91	-1.24	24.40
.450	.514	.764	-.1519	7.383	5.757	75.62	23.27	-3.66	14.50
.350	.599	.775	-.1750	8.137	6.812	103.69	32.40	-5.16	7.57
.250	.679	.785	-.20[illegible]	8.942	7.659	129.68	44.24	-8.71	3.32
.150	.742	.767	-.2362	9.545	8.253	149.81	56.71	-15.50	1.05
.050	.783	.737	-.2543	10.031	8.747	173.88	67.79	-32.14	0.00
.082	SIDE OF BODY LOADS					166.11	64.21	-26.76	
.072	STREAMWISE LOADS						55.83	-41.50	

AIRPLANE LIFT CURVE SLOPE (PER DEG)	7.11354E-02
BALANCING TAIL LOAD (LB/G)	-15315
GUST VELOCITY,IF USED (FT/SEC)	-0.
THRUST PER ENGINE (10000 LB)	.61100
AIRPLANE LIFT COEFFICIENT	.65823
MAGNIFICATION FACTOR (FOR GUST)	-0.

LOAD FACTOR	2.5000	0.
WING ANGLE OF ATTACK (DEG)	9.0035	-.24969
BALANCING TAIL LOAD (LB/G)	-64515	-31228
AIRPLANE LIFT COEFF. (TAIL-OFF)	.71449	2.72341E-02
PITCHING MOMENT COEFF. (.25 MAC)	-9.72546E-02	-7.51130E-02

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE 8.1-1(d). FINAL WING, WEIGHT SUMMARY

S S A M - SWEPT STRIP AEROELASTIC MODEL
AEROPHYSICS RESEARCH CORPORATION

ETA	SKINL	ID	ASEG	ASTR	TSKIN	TWEB	TSTIF	AMAT	VMAT	WMAT
.950	0.000[illegible]	TS	6.816	18.022	.0640	.0704	4.42	14.92	772.26	78.00
.850	0.000[illegible]	PS	10.3[illegible]5	26.89[illegible]	.091[illegible]	.0732	6.03	22.26	1924.49	194.37
.750	0.000[illegible]	PS	15.3[illegible]0	40.221	.1350	.0995	7.80	33.29	2875.71	290.45
.650	0.000[illegible]	PS	26.6[illegible]	66.345	.178[illegible]	.1204	13.21	56.52	4566.36	461.20
.550	0.000[illegible]	RS	32.171	76.918	.1738	.0999	15.92	67.23	6224.51	628.68
.450	0.000[illegible]	MAX	39.69[illegible]	98.032	.1801	.1290	19.57	83.52	7582.30	765.81
.350	0.000[illegible]	PS	45.20[illegible]	169.952	.1757	.1142	21.92	94.67	8912.13	900.13
.250	0.000[illegible]	RS	53.[illegible]	126.955	.1504	.1249	25.44	113.22	10151.48	1025.30
.150	0.000[illegible]	PS	64.366	146.796	.1933	.1717	28.71	138.63	11815.93	1193.41
		SB		163.583					9061.93	915.26
.050		CS	46.607	100.14[illegible]	.1076	.0871	20.79	100.14	7062.99	713.36

TOTAL (BOTH SIDES) WINGBOX WEIGHT = 14331.92

DEFINITIONS

ETA WING STATION (SPAN FRACTION)
SKINL CRITICAL SKIN LOAD (10E6 LB/IN)
ID ID OF LOCATION OF THE CRITICAL STRESS (IE. AREA)
- CS CENTER SECTION
- FS FRONT SPAR
- PS PEAR SPAR
- SB SIDE BODY
- TS MINIMUM SKIN THICKNESS
- MAX MAXIMUM DEPTH

ASEG SEGMENT AREA (SQ IN)
ASTR STREAMWISE MATERIAL AREA (SQ IN)
TSKIN SKIN THICKNESS (IN)
TWEB SPAR WEB THICKNESS (IN)
TSTIF STIFFENER AREA (SQ IN)
AMAT MATERIAL AREA (SQ IN)
VMAT MATERIAL VOLUME (CU IN)
WMAT MATERIAL WEIGHT (LB/SIDE)

TABLE 8.1-1(e) FINAL WING FLEXIBILITIES

F I N A L   S T I F F N E S S

ETA STA.	EI(10E-9)	GJ(10E-9)	EI(10E-9) BOX	GJ(10E-9) BOX
.95	2.749152E+00	1.533305E+00	2.749152E+00	1.533305E+00
.85	8.332625E+00	5.145732E+00	7.055567E+00	4.116586E+00
.75	2.043946E+01	1.191465E+01	1.406709E+01	9.546919E+00
.65	3.283965E+01	2.150318E+01	2.720675E+01	1.974560E+01
.55	4.869693E+01	3.068477E+01	4.581095E+01	3.209517E+01
.45	8.125465E+01	6.435236E+01	7.643399E+01	5.571633E+01
.35	1.297903E+02	1.055615E+02	1.260100E+02	9.006952E+01
.25	2.250996E+02	1.602510E+02	2.149951E+02	1.569982E+02
.15	4.386154E+02	3.276241E+02	4.276954E+02	3.276241E+02
.05	5.396119E+02	3.884961E+02	5.279960E+02	3.884961E+02

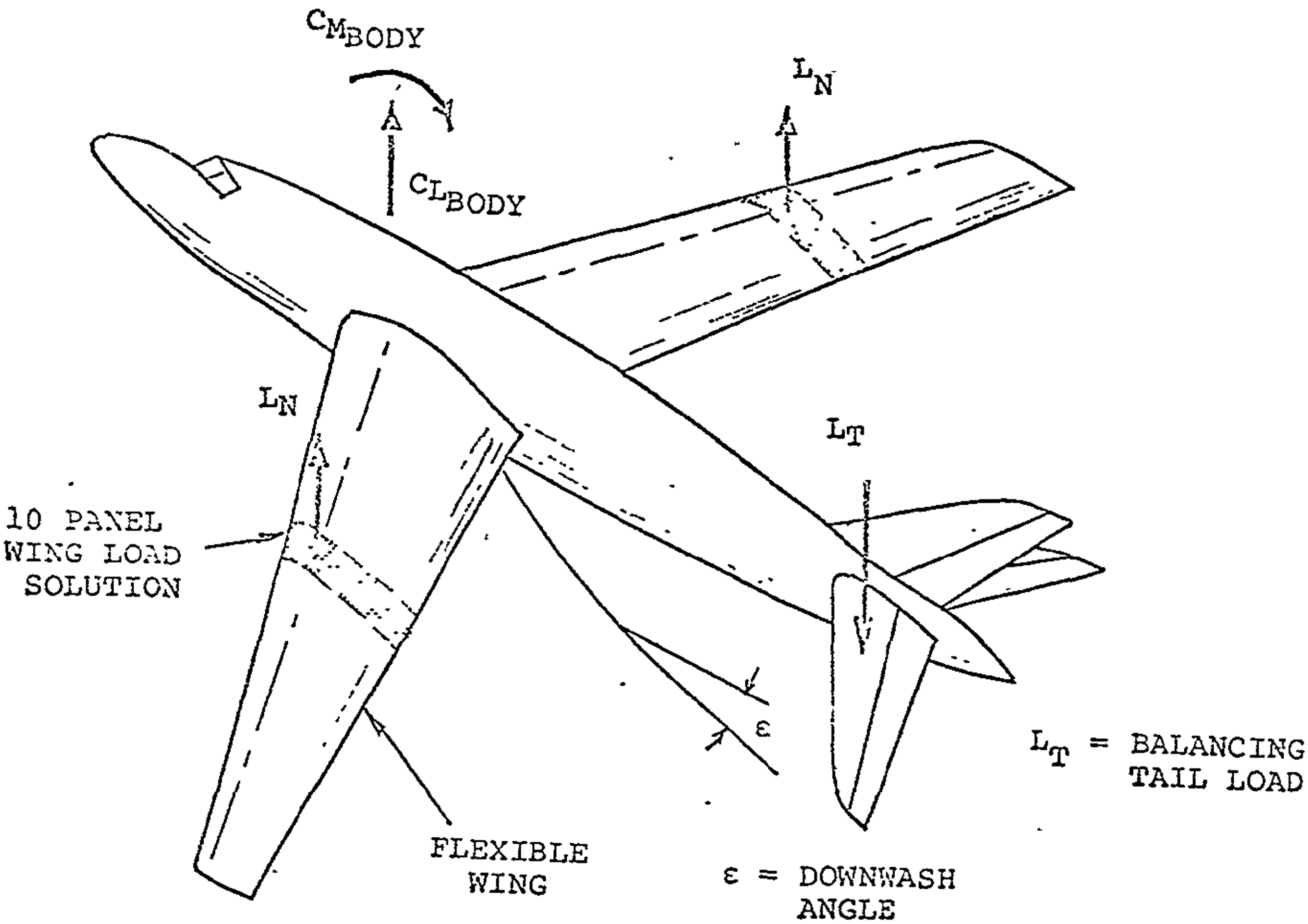
F L E X I B I L I T Y

	FINAL ITERATION		NEXT TO LAST ITERATION	
ETA STA.	10E9/EI	10E9/GJ	10E9/EI	10E9/GJ
.95	3.637485E-01	6.521458E-01	3.637485E-01	6.521858E-01
.85	1.200125E-01	1.943358E-01	1.261474E-01	1.945579E-01
.75	4.892493E-02	8.393096E-02	4.897950E-02	8.402457E-02
.65	3.045212E-02	4.651473E-02	3.048672E-02	4.655775E-02
.55	2.053517E-02	2.725927E-02	2.055683E-02	2.728801E-02
.45	1.230699E-02	1.553945E-02	1.231079E-02	1.555561E-02
.35	7.734737E-03	9.473153E-03	7.714739E-03	9.485450E-03
.25	4.442472E-03	6.014638E-03	4.447178E-03	6.021009E-03
.15	2.278962E-03	3.052278E-03	2.269929E-03	3.055046E-03
.05	1.853184E-03	2.574028E-03	1.854269E-03	2.575536E-03

THIS SOLUTION REQUIRED 3 ITERATIONS.

THE ALLOWABLE STRESS COULD NOT BE DETERMINED 0 TIMES.
THE PREVIOUS ALLOWABLE STRESS WAS USED

ORIGINAL PAGE IS
OF POOR QUALITY



PROGRAM SOLVES FOR WING PANEL LOADS AND
BALANCES AIRPLANE FOR EACH SPECIFIED
FLIGHT CONDITION

(12 EQUATIONS)

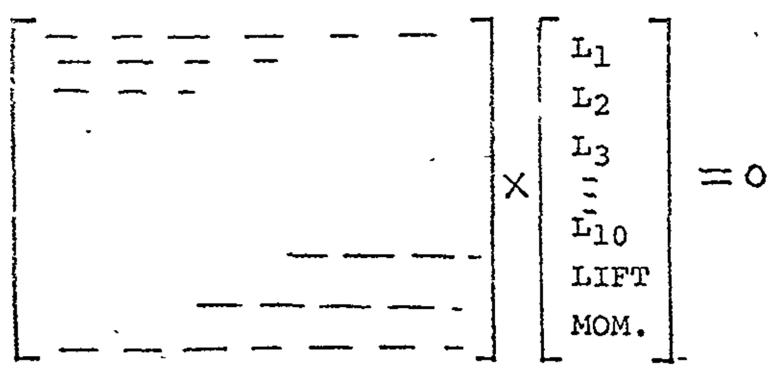


FIGURE 8.1-1. PROGRAM MODEL AND BASIC EQUATIONS

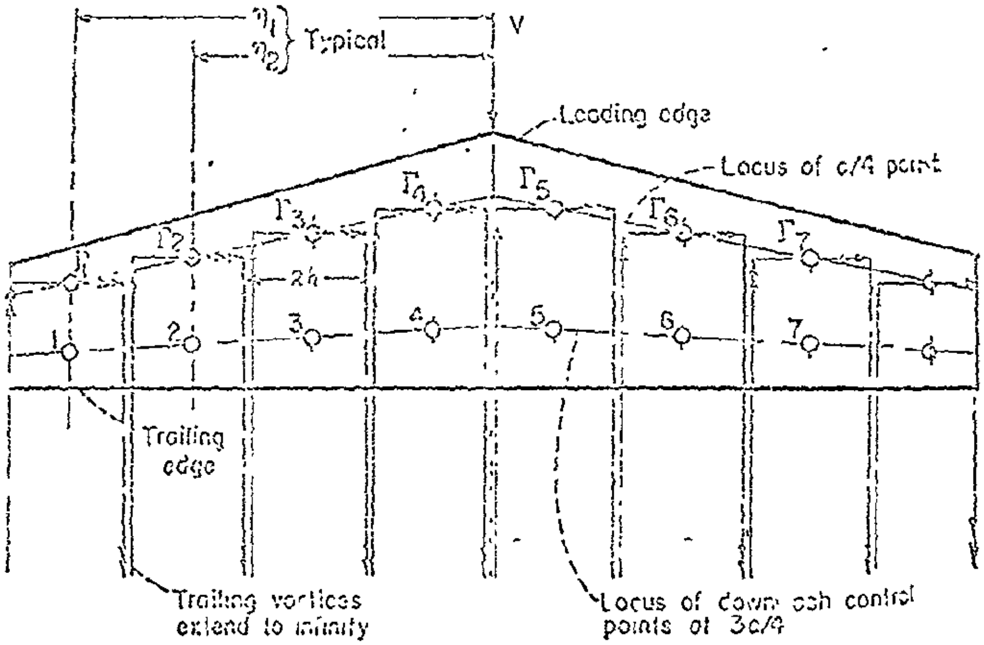


FIGURE 8.1-2. WEISSINGER SUBSONIC STEADY STATE AERODYNAMIC MODEL

ORIGINAL PAGE IS OF POOR QUALITY

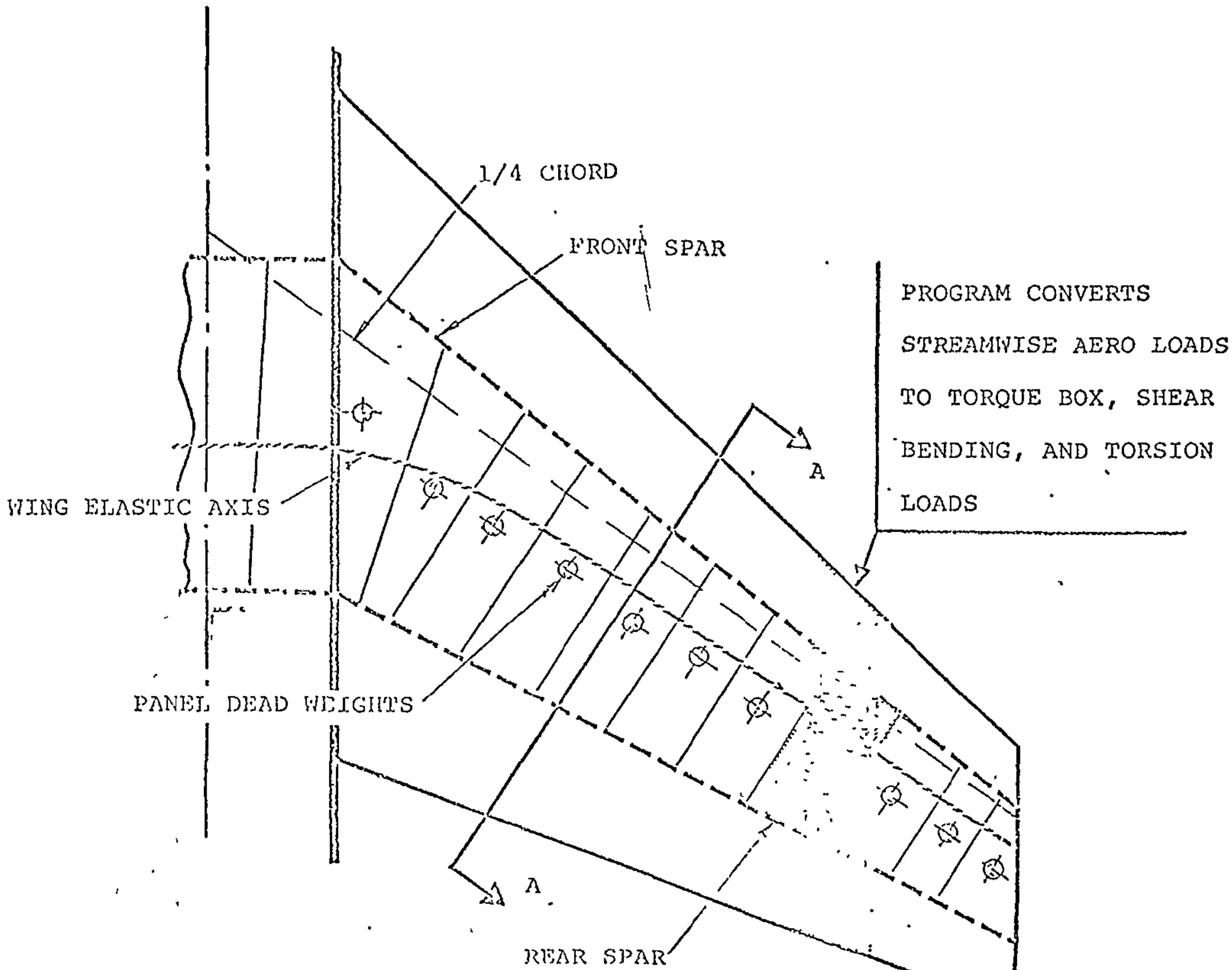


FIGURE 8.1-3. SWEPT STRIP LOAD CALCULATION

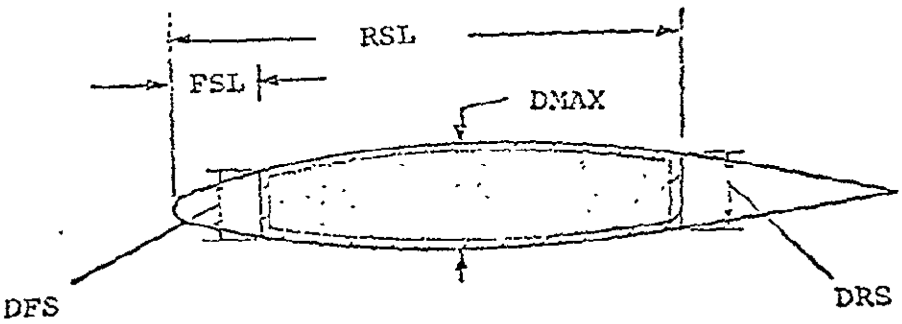


FIGURE 8.1-4. WING BOX SIZE DIMENSIONS

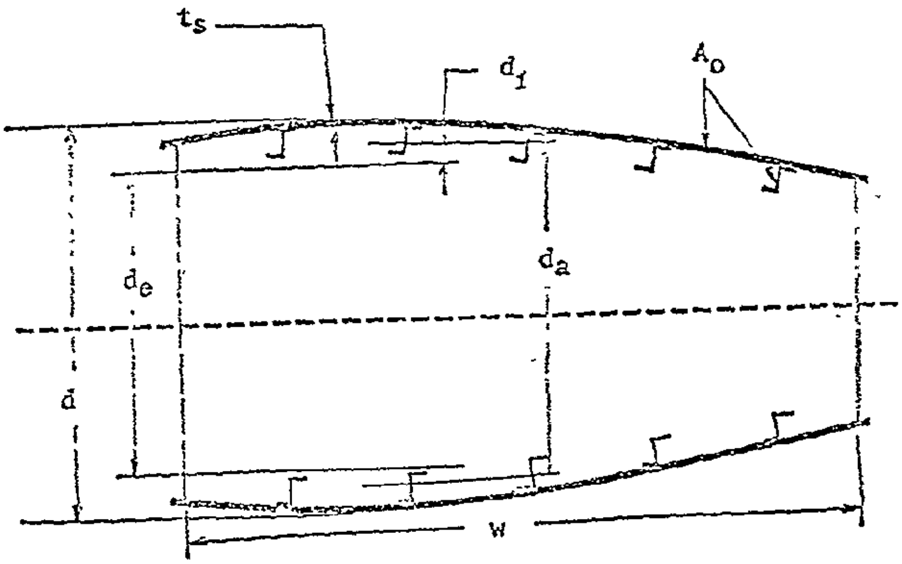


FIGURE 8.1-5. WING BOX SECTION GEOMETRY

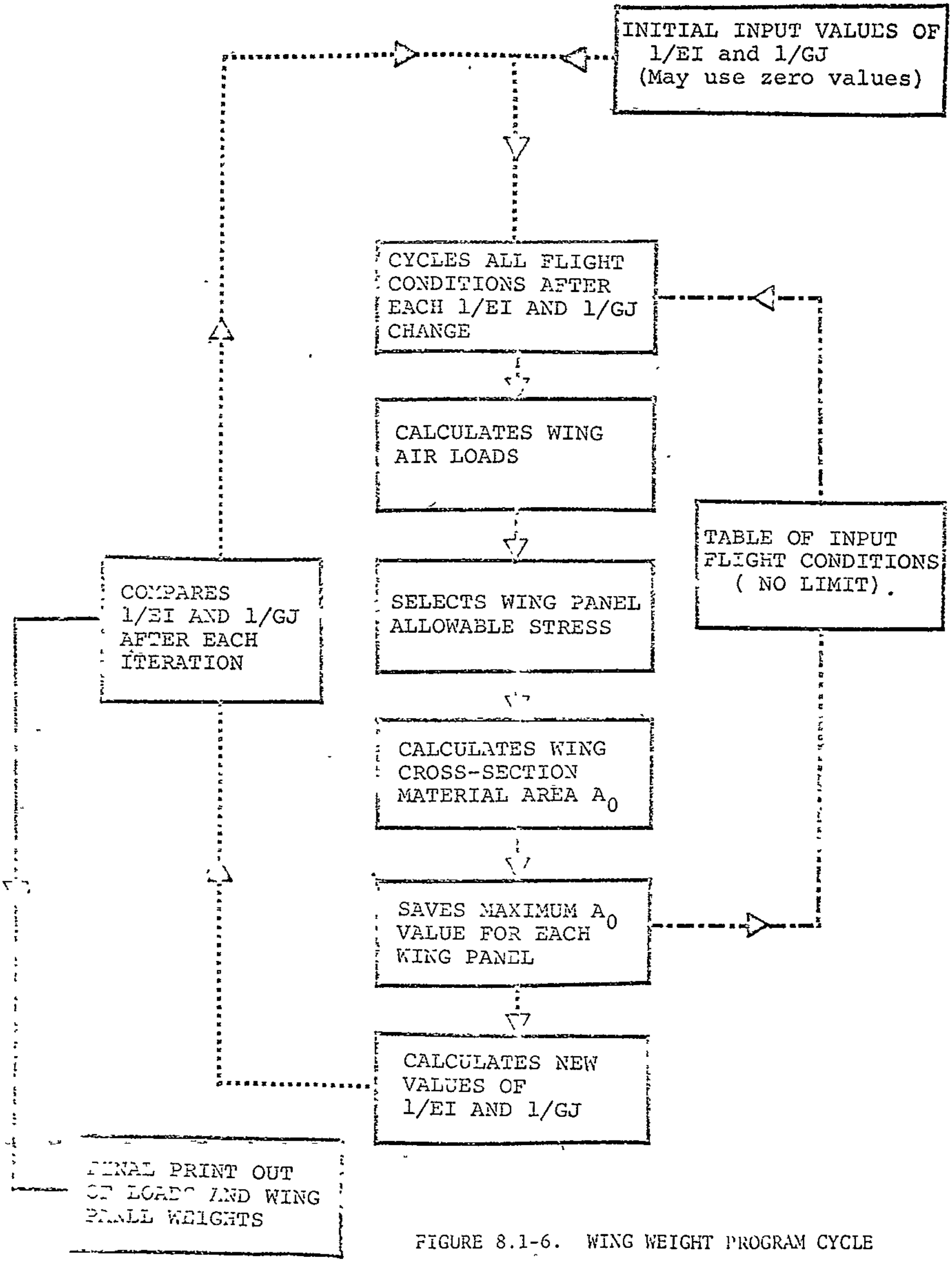


FIGURE 8.1-6. WING WEIGHT PROGRAM CYCLE

SECTION 9

ECONOMICS

Two cost estimation models are available in the ODIN/RLV System. Both programs were originally written for IBM computers but were converted to the CDC 6600 by Aerophysics Research Corporation during the ODIN studies. The two economics models used in the ODIN/RLV are

1. DAPCA: A computer program for determining aircraft development and production costs, Reference 1. This program was originally constructed at the Rand Corporation. The CDC 6600 version available in the ODIN/MFV study (for Wright-Patterson Air Force Base) was constructed by Aerophysics Research Corporation.

2. PRICE: A program for improved cost estimation for total program cost of aircraft, spacecraft, and reusable launch vehicles. This program was constructed by Mr. Darrell E. Wilcox of NASA's Advanced Concepts and Mission Division, Ames Research Center.

The more recently developed program, PRICE, is the cost model most frequently employed. Description of DAPCA is limited to a presentation of the equations employed.

REFERENCES:

1. Boren, H. E., Jr., DAPCA: A Computer Program for Determining Aircraft Development and Production Costs, The Rand Corporation, RM-5221-PR, February 1967.

2. Hague, D. S. and Glatt, C. R., Optimal Design Integration for Military Flight Vehicles, ODIN/MFV, AFFDL-TR-72-132, 1972.

TABLE OF CONTENTS FOR SECTION 9.1, PROGRAM PRICE

9.1 PROGRAM PRICE: A PROGRAM FOR IMPROVED COST ESTIMATION OF TOTAL PROGRAM COSTS FOR AIRCRAFT AND REUSABLE LAUNCH SYSTEMS

9.1.1 Introduction

Program PRICE was constructed by NASA's Advanced Concepts and Mission Division and is reported in full in Reference 2 of Section 9. The discussion below is a synopsis from that report which is originally due to D. E. Wilcox.

Cost estimation has received much attention in recent years due to the growing size and complexity of aircraft and space vehicles and the increasing cost-awareness of those involved with planning future programs. An important part of any mission analysis is the estimate of the total program cost and its variation with changes in design concept or study guidelines. A constant problem to the mission analyst is the lack of valid cost data in sufficient detail to allow the derivation of meaningful cost sensitivities as a function of design characteristics. This is particularly true of high speed/high performance vehicles where extrapolations beyond the existing data base usually are required to estimate costs.

There are a number of excellent cost models in existence; although none are entirely suitable to the present purpose. For example, Rand Corporation (Reference 1) and Planning Research Corporation (R-547a) have published cost models for conventional aircraft. Both are based on statistical correlations of historical cost data for military aircraft. Neither includes data for aircraft capable of speeds above Mach 3, nor are the cost models intended for such use. Both models are applicable mainly to large production programs and cannot be used to estimate the costs of an experimental aircraft program or a space shuttle vehicle. Moreover, both models aggregate costs at a very gross level; the Rand model has nine equations, while the Planning Research Corporation model uses only three equations to describe the total development and procurement cost. This aggregation provides very little sensitivity to design detail and is, therefore, of limited use in vehicle trade studies.

Other cost models, References 2 to 4, yield somewhat greater cost visibility by providing more detailed breakdowns of the estimates. This is accomplished by estimating at the subsystem level and by more emphasis on the functional distribution of costs. All of these models are primarily applicable to spacecraft, however, and the first three were developed specifically to study space shuttle costs. There are many other cost models not referenced here, but most are limited to a specific class of vehicle.

The cost model used in PRICE was developed in an effort to eliminate some of the shortcomings of other models. It is applicable to aircraft of all speeds, launch vehicles (airbreathing or VTO rocket), and spacecraft. It may be used for either large production programs or experimental vehicle programs. Moderate sensitivity to design characteristics is provided by estimating hardware costs at the subsystem level and all other costs at a functional level. Where possible, historical data for aircraft, spacecraft, and launch vehicles are correlated together.

The cost model is divided into three life cycle phases: RDT&E, Acquisition, and Operations. The cost of each of these phases is determined by summing numerous cost elements which conform to specific program tasks or hardware elements. The cost element structure approximates level 5 of the NASA Work Breakdown Structure, Reference 5, although conformance is not exact because the cost model accommodates aircraft data which have not been reported to this WBS. Hardware costs are computed using cost elements roughly corresponding to level 6 of the WBS. The subsystem groupings are actually based on U.S. Air Force Specification MIL-M-38310A, Reference 6, because the vehicle weight statement generated by most synthesis programs is based on this specification.

The estimating relationships used in all phases of the cost model are based on correlations of historical cost data with gross physical characteristics. This method is typical of conceptual design costing, and has the advantage of providing fairly good estimates from a minimum of design information. The method has two disadvantages. The first is the limited sensitivity to detailed vehicle design characteristics, which is a result of the failure to report costs to the detail level in past programs and an over-reliance on weight in the cost model. The second disadvantage is the difficulty associated with estimating the cost of vehicles which advance the state of the art, since by definition there is usually no historical data upon which to base the estimates. This is a problem with nearly all estimating techniques. Partial solutions can be achieved through the use of "complexity factors," but only when data exist to establish the value of such factors.

The computer program is described in detail in Reference 8 which identifies the input and output parameters, and gives a program listing. It also includes sample input and output for a lifting-body-reusable, space transportation system. It should be noted that the cost data base associated with PRICE includes proprietary data. Its contents can only be made available to qualified Government sources.

9.1.2 Cost Model

The cost model was divided into the cost elements shown in Figure 9.1-1. The cost elements fall into one of three major phases: RDT&E, Acquisition, and Operations. RDT&E as defined here includes both concept formulation and contract definition studies, plus vehicle design, development, and test, initial tooling, flight test, and all other costs up to the establish ment of an initial operating capability except facilities. The acquisition phase includes all capital expenditures required to support the operational phase, such as operational vehicles, facilities, training equipment, ground support equipment (AGE), spares, plus handbooks and other miscellaneous equipment. Operations includes all annually recurring labor and material costs required to support flight operations to program completion.

The cost of procuring flight test and operational vehicles is determined by computing the first unit manufacturing cost of the vehicle and applying a learning curve over the total number purchased. The first unit costs is the sum of the first unit costs of 31 major subsystems, each of which is described by one or more CER's based upon component weight or other input from the synthesis programs. Since the first unit cost is used in several of the cost elements of Figure 9.1-1, it is computed first by the computer program and will be discussed first in this section.

9.1.3 First Unit Manufacturing Cost

The first unit cost is defined as the manufacturing cost of the first flight test article, and it includes all labor, material, and overhead costs associated with the production of that component. Sustaining engineering and tooling are not included but are computed as separate items.

The major factors influencing manufacturing cost are the weight, size, and complexity of parts, the total number of parts, and the number of dissimilar parts. Also important are certain performance parameters such as power output of electronic equipment or thrust and specific impulse of propulsion systems. For structural components the material and type of construction is critical. In the present cost model, however, most component costs were related to weight, with a complexity factor used to account for cost variations due to material and type of construction. Although complexity factors vary from one source to another, the values of Reference 2, modified slightly, are used for all structural components in this cost model. The first unit cost is broken down according to the 31 subsystems of Figure 9.1-2.

The equations for first unit cost components are nearly all of the type

$$C = a\, W^b\, C_f$$

where a and b are correlation constants, W is the subsystem weight, and C_f is a complexity factor. For structural components the value of C_f can be taken from Table 9.1-1. For non-structural subsystems, C_f is nominally 1.0 but the user may supply a different value if the component complexity is expected to differ from that represented by the data included in the correlation.

1. <u>Body Structure</u>. The first unit manufacturing cost of the basic body is related to the structural weight by CER's which vary with the type of vehicle as shown in Figure 9.1-3. The division of costs is based on the availability of data to derive a CER for each of the components shown.

a. For VTO launch vehicles the following equations are used where all symbols are defined in Table 9.1-2.

$CADAPT = 1730\ (WADAPT)^{.678}\ (CFADAP)$, Adapters

$CFWD = 1730\ (WFWD)^{.678}\ (CFFWD)$, Forward skirts

$CAFT = 1730\ (WAFT)^{.678}\ (CFAFT)$, Aft skirts

$CINTK = 1730\ (WINTK)^{.678}\ (CFINTK)$, Intertank structure

$CTHRST = 3400\ (WTHRST)^{.678}\ (CFTHRS)$, Thrust structure

$COTANK = 7400\ (WOTANK)^{.565}\ (CFOXTK)$, Oxidizer tank

$CFTANK = 7400\ (WFTANK)^{.565}\ (CFFUTK)$, Hydrogen fuel tank

$CFTANK = 5770\ (WFTANK)^{.565}\ (CFFUTK)$, Storable fuel tank

$CNOSE = 1730\ (WNOSE)^{.678}\ (CFNOSE)$, Nose structure

b. For spacecraft

$CCOMPT = 20130\ (WCOMPT)^{.631}\ (CFCOMPT)$, Crew compartment

and

$CSERV = 8800\ (WSERV)^{.631}\ (CFSERV)$, Cargo compartment

$CADAPT = 2100\ (WADAPT)^{.631}\ (CFADAP)$, Adapter

c. For aircraft

$CBODY = 56100\ (WBODY)^{.451}\ (CFBODY)$, Body structure

2. Aerodynamic Surfaces.

$CWING = 36000\ (WWING)^{.451}\ (CFWING)$, Aircraft wing

$CEMP = 10230\ (WWEMP)^{.451}\ (CFEMP)$, Aircraft empennage

$CFAIR = 1730\ (WFAIR)^{.678}\ (CFFAIR)$, VTO launch vehicle fins and engine fairings

3. Thermal protection system.

$$CTPS = (UCABL + UCCOV + UCINS)_i (NPANEL)_i^{.926} (SPANEL)_i$$, Thermal protection systems

$$(NPANEL)_i = (STPS)_i / (SPANEL)_i$$, Number of panels in TPS's

4. Subsystems. First unit cost data for the vehicle subsystems were more difficult to obtain than that for the main structure. One reason for this is that subsystems on many military aircraft programs are Government furnished equipment (GFE), and the aircraft manufacturer is not responsible for their costs. Due to this lack of data, the confidence level attached to the CER's for these subsystems is lower than that for the structure or propulsion system CER's.

5. Landing gear.

$$CLG = 10430\ (WLG)^{.541}\ (CFLG)$$, Landing gear

6. Launch, docking, and recovery gear.

$$CLANCH = 500\ (WLANCH)\ (CFLNCH)$$, Landing

$$CDOCK = 500\ (WDOCK)\ (CFDOCK)$$, Docking

$$CDPLOY = 1340\ (WDPLOY)^{.766}\ (CFDPLY)$$, Deployable recovery gear

$$CRECOV = 42100\ (WRECOV)^{.7064}\ (CFRECV)$$, Recovery aids

7. Engines.

$$CENGS = [350000. + 475\ (T)^{.7}]\ (EN)^{zetap}$$, pump-fed cryogenic fueled engines

$$CENGS = [270000 + 24\ (T)^{.8}]\ (EN)^{zetap}$$, pump-fed, storable fueled engines

$$CENGS = [22000 + 240\ (T)^{.8}]\ (EN)^{zetap}$$, pressure-fed, radiative, storable fueled engines

$$CENGS = [35000 + 450\ (T)^{.8}]\ (EN)^{zetap}$$, pressure-fed, ablative storable fueled engines

For airbreathing engines, CER's were provided for turbojets based on thrust and ramjets or scramjets based on weight:

$$CTJ = [3270\ (T)^{.60}]\ (EN)^{zetap}\ (CFENG)$$

$$CRJ = 27000\ (WRJ)^{.523}$$

8. Inlets and nacelles.

$CINLET = 56100\ (WINLET)^{.451}\ (CFINLT)$	, inlets
$CNACEL = 56100\ (WNACEL)^{.451}\ (CFNAC)$	, nacelles

9. Tanks.

$CFUTK = 7400\ (WFUTK)^{.565}\ (CFTK)$	, cryogenic fuel tank
$COXTK = 7400\ (WOXTK)^{.565}\ (CFTK)$	, cryogenic oxidizer tank
$CFUTK = 6660\ (WFUTK)^{.565}\ (CFTK)$	, storables
$COXTK = 6660\ (WOXTK)^{.565}\ (CFTK)$	, storables
$CINSTK = (DPLBIN)\ (WINSTK)$	, tank insulation
$CFUSYS = 59000\ (WFUSYS)^{.43}\ (CFFUEL)$	, fuel systems
$COXSYS = 59000\ (WOXSYS)^{.43}\ (CFOX)$	, oxidizer systems
$CFUSYS = 300\ (WFUSYS)(CFFUEL)$	, aircraft fuel systems
$CPRSYS = 59000\ (WPRSYS)^{.43}\ (CFPRES)$	, space vehicle pressurization systems
$CPUSYS = 59000\ (WPUSYS)^{.43}\ (CFPRES)$	, space vehicle propellant utilization system
$CLUBE = 59000\ (WLUBE)^{.43}\ CFLUBE)$	, space vehicle lubricating system
$CPRSYS = 300\ (WPRSYS)\ (CFPRES)$	, aircraft pressurization system
$CPUSYS = 300\ (WPUSYS)\ (CFPUSY)$	, aircraft propellant utilization system
$CLUBE = 300\ (WLUBE)\ (CFLUBE)$	, aircraft lubricating system

10. Orientation, Separation, and Ullage Control.

$CAERO = 400\ (WAERO)\ (CFAERO)$, conventional vehicles

a. For a spacecraft

$CAERO = 63000\ (WAERO)^{.504}\ (CFAERO)$, aerodynamic stabilization system

$CACS = 61700\ (WACS)^{.529}\ (CFACS)$, reaction control system

$CACSTK = 6660\ (WACSTK)^{.565}\ (CFACTK)$, RCS tanks

$CAUXT = 61700\ (WAUXT)^{.529}\ (CFAUXT)$, auxiliary thrust (liquids)

$CAUXT = 395\ (WAUXT)^{.66}\ (CFAUXT)$, auxiliary thrust (solids)

11. Electrical power.

$CPOWER = 20950\ (WPOWER)^{.536}\ (CFPOW)$, spacecraft power supplies (battery)

$CPOWER = 36096\ (WPOWER)^{.536}\ (CFPOW)$, spacecraft power supplies (fuel cell)

$CELCAD = 16170\ (WELCAD)^{.766}\ (CFELCD)$, spacecraft electrical distribution

$CELCAD = 1970\ (WELCAD)^{.536}\ (CFELCD)$, launch vehicle/aircraft electrical distribution

12. Hydraulic system.

$CHYCAD = 1970\ (WHYCAD)^{.766}\ (CFHYCD)$, hydraulics

13. Guidance and navigation.

$CGNAV = 243{,}000\ (WGNAV)^{.485}\ (CFGNAV)$, spacecraft

$CGNAV = 22200\ (WGNAV)^{.786}\ (CFGNAV)$, launch vehicle/aircraft

14. Communications.

$CCOMM = 7220\ (WCOMM)^{.5743}\ (CFCOMM)$, cruise and launch vehicles

$CCOMM = 82500\ (WCOMM)^{.5743}\ (CFCOMM)$, spacecraft

15. Instrumentation.

$CINSTR = 12750 \; (WINSTR)^{.596} \; (CFINST)$, all vehicles

16. Environmental control.

$CEQECS = 10800 \; (WEQECS)^{.5065} \; (CFEQEC)$, unmanned vehicles

$CPECS = 202500 \; (WPECS)^{.373} \; (CFPECS)$, manned spacecraft

$CPECS = 6430 \; (WPECS)^{.5065} \; (CFPECS)$, other manned vehicles

17. Insulation.

$CINCOM = (DPLBIN) \; (WINCOM)$, crew compartments

18. Crew provisions.

$CPPROV = 1400 \; (WPPROV)^{.7625} \; (CFPROV)$, aircraft

$CPPROV = 6540 \; (WPPROV)^{.7625} \; (CFPROV)$, spacecraft

19. Crew controls and display panels

$CCANDP = 26800 \; (WCANDP)^{.4926} \; (CFREW)$

20. Abort.

$CABORT = 16960 \; (WABORT)^{.556}$

21. Final assembly and checkout.

$CFASSY = (XFASSY) \; (CSTRUC)$

22 Propellants and gases.

$CFUEL = (DPLBFU) \; (WFUEL)$, fuel/launch

$COXID = (DPLBOX) \; (WOXID)$, oxidizer/launch

$CAUXP = (DPLBAU) \; (WAUXP)$, auxiliary/launch

$CGASPR = (DPLBGS) \; (WGASPR)$, gases/launch

9.1.4 Research, Development, Test, and Evaluation (RDT&E)

RDT&E consists of all costs required to design and develop the vehicle and subsystems. Included are engineering, initial tooling, flight test, test hardware and spares, ground support equipment, and documentation. Although it would be instructive to detail RDT&E costs by component as was done with first unit manufacturing costs, there was not sufficient data in literature to do so. Attempts have been made for spacecraft, References 2 to 4. but for aircraft this type of data is very scarce. The breakdown employed for computing RDT&E costs follows.

1. Concept formulation. This preliminary study activity includes system description, cost and schedule estimates, and mission analyses. The cost is estimated by the product of user inputs for the number of contractors involved, NOCON, the number of engineers per contractor assigned to the program, NOENG, and the time span of the activity, NOYRS:

$$CF = 35000\ (NOCON)\ (NOENG)\ (NOYRS)$$

The indicated cost of $35000 per man-year reflects an engineering and support labor rate, including overhead, of about $17/hour.

2. Contract definition. The contract definition phase establishes preliminary design of the vehicle and detailed mission analyses, leading to selection of one contractor for development of the vehicle. The cost of this activity is estimated with user inputs as above:

$$CD = 35000\ (NOCON1)\ (NOENG1)\ (NOYRS1)$$

3. Airframe design and development.

$$DDEL = 3145\ (WA)^{.5825}\ (RE)\ (CONFIG)$$, subsonic prototypes

$$DDEL = 23.85\ (WA)^{.5825}\ (VMAX)^{.771}\ (RE)\ (CONFIG)$$, advanced aircraft

$$DDEL = 207\ (WA)^{.931}\ (RE)\ (CONFIG)$$, subsonic production aircraft

$$DDEL = 348\ (WA)^{.931}\ (RE)\ (CONFIG)$$, Mach 2 aircraft

$$DDEL = 115000\ (WA)^{.509}\ (RE)\ (CONFIG)$$, spacecraft

, expendable VTO launch vehicles

$$DDEL = [1930000\ (WA)^{.484} + 16.65\ (NENG)^{.26}\ (TPEREN)^{.14}]\ (CONFIG)$$

4. Subsystem development.

$$SUBSYS = 35000\ (WACS + WECS + WPOWER + WPOWCD + WDPLOY + WCANDP + WAERO + WRECOV + WPPROV)\ (XNEW)$$

5. Avionics development.

$$AD = [5.3 \times 10^6\ (WGNAV)^{.439} + 2.19 \times 10^6\ (WCOMM)^{.439} + 0.55 \times 10^6\ (WINST)^{.439}](XAVD)$$

6. Propulsion development.

$$PDTJ = 29.5 \times 10^6 \left(\frac{T}{1000}\right)^{.55} (MACH)^{.66} [(NV + NFV)\ (EN)\ (1. + ENSPAR)]^{.1}$$

, turbine engines

$$PDCSJ = 204 \times 10^6\ (ASJMOD)^{.41}$$

, ramjet and scramjet

a. For liquid rocket engines the following equations are used.

For regenerative cooled, pump fed, oxygen/hydrogen engines,

$$PDROCK = 50 \times 10^6 + 1.405 \times 10^6\ (T)^{.422};$$

for regenerative cooled, pump fed, storable propellant engines,

$$PDROCK = 50 \times 10^6 + 8.65 \times 10^5\ (T)^{.422};$$

for ablative cooled, pressure fed, storable propellant engines,

$$PDROCK = 10 \times 10^6 + 8.4 \times 10^4\ (T)^{.678};\ \text{and}$$

for radiation cooled, pressure fed, storable propellant engines,

$$PDROCK = 5 \times 10^6 + 4.86 \times 10^4\ (T)^{.678}.$$

7. Flight test hardware.

$$FV = (AV + AMFG)\ (NFV)^{ZETA} + (PROPU)\ (NFV)^{ZETAP}$$

$$ZETA = 1 + \frac{\ln [.01\ (LEARN)]}{\ln 2}$$

and

$$ZETAP = 1 + \frac{\ln [.01\ (LEARNP)]}{\ln 2}$$

$$FS = .20\ (FV)$$

8. Ground test hardware.

$$GTS = .10\ (GTV)$$

9. Tooling and special test equipment.

$TST = 6.19\ (RT)\ (WA)^{1.062}\ (TOOLC)$, conventional aluminum

$TST = 45.0\ (RT)\ (WA)^{1.062}\ (TOOLC)$, VTO launch vehicles

$TST = 0.0267\ (RT)\ (WA)^{.99}\ (VMAX)^{1.21}\ (RATE)^{.4}\ (TOOLC)\ (NFV)^{.14}$, advanced aircraft

10. Flight test operations.

$FTO = 0.75\ (NFV)^{1.1}\ (WG)^{.08}\ (VMAX)^{0.9}$, typical aircraft certification flight test

$FTO = (ROPF)\ (NFTEST)^{ZETA}$, other flight test programs

11. Ground support equipment.

$$AGEP = .05\ (ADDE) + .15\ (FV)$$

12. Technical data.

$$TDP = .02\ (FV)$$

13. Basic RDT&E cost.

$$RDTE = CF + CD + DDEL + AD + PD + SUBSYS + FV + FS + GTV + GTS + TST + FTO + AGEP + TDTP$$

14. Fee.

$$RDFEE = (RDTE)(FEE)$$

15. Project management.

$$RDMGMT = (RDTE)(PMGMT)$$

16. Total RDT&E cost.

$$TRDTE = RDTE + RDFEE + RDMGMT$$

9.1.5 Acquisition

The initial acquisition cost includes operational vehicles, ground support equipment, facilities, and all capital investment required before the operational phase can begin, such as spares, training equipment and training initial stocks and miscellaneous equipment. The cost of all sustaining engineering and sustaining tooling costs associated with the operational vehicles must be included also. The following acquisition cost breakdown is used.

1. Operational vehicles.

$$CAF = (AMFG)\ [(NVHF)^{ZETA} - (NFV)^{ZETA}] \qquad \text{, airframe}$$

$$AVO = (AV)\ [(NVHF)^{ZETA} - (NFV)^{ZETA}] \qquad \text{, avionics}$$

$$PO = (PROPU)\ [(NVHF)^{ZETAP} - (NFV)^{ZETAP}] \qquad \text{, engines}$$

$$CVO = CAF + AVO + PO \qquad \text{, total}$$

The assumed value of the learning curve can be critical. For example, a five per cent error in the assumed rate of learning yields errors in total fleet cost of more than 16% for 10 vehicles, and more than 46% for 1000 vehicles. The effect of learning curve on average cost versus quantity is shown in Figure 9.1-4.

2. Ground support equipment.

$$AGEO = 0.15\ (OV)$$

3. Spares.

$$OS = 0.13\ (OV)$$

4. New facilities. The cost of new facilities depends on the individual requirements of each program, the size and nature of the vehicle, the number of operational sites, and the type of facilities already in existence. The user must supply his own facility cost, FAC, by input. For most conventional aircraft existing facilities can be used; in that case, FAC = 0.

5. Sustaining engineering.

$$SE = (DDEL)\ [(NV)^{.20} - 1]$$

6. Sustaining tooling.

$$ST = (TST)\ [(\frac{NV}{NFV})^{.14} - 1]$$

7. Miscellaneous equipment.

$$MEQ = 500\ (NPER)$$

8. Training equipment.

$$OT = .1442\ (OV)\ (NV)^{-.4525} \quad \text{, aircraft}$$

$$OT = 0.2088\ (CSTRUC)^{1.3822} \quad \text{, spacecraft}$$

9. Initial training.

$$IT = 50000\ (NPL)$$

10. Initial transportation.

$$TRI = 0.005\ (OV + OS + MEQ + OT + AGEO)$$

11. Basic acquisition.

$$IV = OV + AGEO + OS + FAC + SE + ST + MEQ + OT + IT + TRI$$

12. Fee.

$$AQFEE = (IV)\ (FEE)$$

13. Project management.

$$AQMGMT = 0.01\ (IV)$$

14. Total acquisition cost.

$$AQ = IV + AQFEE + AQMGMT$$

9.1.6 Recurring Operations

Recurring operations is the 10 year operating cost of the fleet of vehicles, including the following: salaries of launch personnel, support personnel, and pilots; maintenance of vehicles, facility, and ground support equipment; propellants; replacement training and transportation; vehicle retrieval from oceans (if applicable); miscellaneous expendables (including small rocket motors, adapters, etc.). Cost breakdown employed for recurring operations follows.

1. Wages, salaries, and allowances.

$NPL = 2\ (NCREW)\ (NV)$, flight crew number

$NMTPS = (SURF)(HRPFT2)(XREPL)(LPM)/160$, TPS personnel

$NMPR = NMAINT + NMTPS$, total support personnel

$WSA = 500{,}000\ (NPL) + 270{,}000\ (NSPR)$, total 10 year support and crew cost

$CLO = 58700(WDRY)^{.524}\ [12(LPM)]^{-.891}\ [120(LPM)]$, launch vehicle 10 year cost

$CLSUP = 0.598\ (CLO)^{1.204}$, launch support 10 year cost

$WSA = CLO + CLSUP$, total 10 year launch vehicle cost

2. Vehicle maintenance.

$VMTPS = (XMTPS)(CTPS)\ [120(LPM)-NV]^{ZETA}$, TPS maintenance

$VM = VMTPS + 270{,}000\ (NMPR)$, total maintenance

or

$VM = 120\ (LPM)\ (XMRA)\ (CSTRUC)$

3. Vehicle retrieval.

$WTR = 1.98 \times 10^7\ (WE/1000)^{.585}\ (LPM)$, water retrieval

4. Propellants.

$PF = 120\ (LPM)\ (CFUEL + COXID + CAUXP + CGASPR)$

5. Miscellaneous expendables.

$MFL = 120\ (LPM)\ (EXPEND)$

6. Facilities maintenance.

$FM = 0.4\ (FAC)$

7. GSE maintenance.

$AGEM = 0.3\ (AGEO)$

8. Miscellaneous equipment maintenance.

$MEM = 550\ (NPER)$

9. Training.

TO = 400,000 (NPL)

10. Transportation.

TRO = 0.15 (PF + MFL + FM + AGEM + MEM)

11. Total operations.

RO = WSA + VM + WTR + PF + MFL + FM + AGEM + MEM + TO + TRO

12. Fee.

ROFEE = 0.5 (RO)(FEE)

13. Project management.

ROMGMT = 0.057 (RO)

14. Total operations.

ROTOT = RO + ROFEE + ROMGMT

9.1.7 Total Program Cost

The total program cost is the sum of RDT&E, acquisition, and operations:

TOTAL = TRDTE + AQ + ROTOT , total cost of program

9.1.8 Conclusion

It is again emphasized that the above description merely summarizes program PRICE; for complete details refer to Reference 8. For user convenience Tables 9.1-2 to 9.1-5 present the prime input and output parameters for the PRICE program. A typical sample case is presented in Table 9.1-6.

REFERENCES

1. Levenson, G. S. and Barro, J. M., "Cost-Estimation Relationships for Aircraft Airframes," Rand Corporation Memo RM-4845-PR, 1966.

2. Anon., "Optimized Cost Performance Design Methodology," McDonnell-Douglas Corporation, Report G975, April 1969.

3. Anon., "Research Study on Low Cost Earth Orbital Transportation System Synthesis by Economic Analysis," Final Report, Contract NAS 8-30522, The Boeing Company, December 1969.

4. Campbell, H. G. and Dreyfuss, D. J., "Manned Spacecraft Cost Estimating Relationships," Rand Corporation Memo RM-5317-NASA, June 1967.

5. NASA, Office of Advanced Manned Missions, "Specification for Contractor Presentation of Cost and Schedule Plans for New Space Projects," February 17, 1969.

6. Anon., "Mass Properties Control Requirements for Missile and Space Vehicles," USAF Military Specification MIL-M-38310, December 15, 1964.

7. Wilcox, D. E. and Gregory, T. J., "A Fundamental Approach to Aircraft Manufacturing Cost Estimating in the Conceptual Design Phase," Presented at the 29th Annual Conference of the Society of Aeronautical Weight Engineers, Washington, D. C. May 1970.

8. Hague, D. S. and Glatt, C. R., Optimal Design Integration for Military Flight Vehicles, ODIN/MFV, AFFDL-TR-72-132, 1972.

9. Anon., "Influence of Structure and Material Research on Advanced Launch Systems," NASA Contractor Report CR-1116, July 1968.

10. Morris, R. E. and Williams, N. B., "Study of Advanced Air Breathing Launch Vehicles with Cruise Capability," Volume 5, NASA CR-73198, Lockheed California Company, February 1968.

11. Anon., "Integral Launch and Re-entry Vehicle," General Dynamics, (Convair) First Monthly Progress Report, Contract NAS 9-9207, 1969.

12. Large, J. P., "Estimating Aircraft Turbine Engine Costs," Rand Corporation Memo RM-6384/1-PR, September 1970.

13. Anon., "Study of Navigation and Guidance of Launch Vehicles Having Cruise Capability," Final Report Contract NAS 2-23691, The Boeing Company, April 10, 1967.

14. Anon., "Gas Turbine Engine Market 1970-1974," DMS Inc., April 1970.

15. Anon., "Launch Vehicle Companent Costs Study," Lockheed Missiles and Space Co., Report 895429, Contract NAS 8-11368, June 1965.

16. Johnson, D. R., "Design Considerations of Reusable Launch Vehicles," Volume VI, Final Report, Contract NAS 2-3191, Douglas Aircraft Co., October 1966.

17. Schlosser, K. P., "Cargo Aircraft Cost Model," Directorate of Operations Research, Wright-Patterson Air Force Base, November 2, 1965.

18. Anon., "Aircraft Systems Cost Factors," Comptroller Office, Aeronautical Systems Division, Wright-Patterson Air Force Base, December 1969.

19. Anon., "Supersonic Transport Development and Production Cost Analysis Program," Booz-Allen Applied Research Inc., Final Report, Contract FA-SS-66-13, December 1966.

20. Anon., "U. S. Air Force Cost and Planning Factors," Air Force Manual 172-3 (Confidential)

21. Woolsey, J. P., "American Seeks Flight Training Reductions," Aviation Week and Space Technology, July 13, 1970, p. 41.

22. Anon., "Optimized Cost Performance Design Methodology," Final Oral Briefing, Contract NAS 2-5022, February 2, 1970.

23. Anon., "Supersonic Transport Development Program, Phase III Proposal," The Boeing Company, Report V6-B2707-2, September 6, 1966.

24. Anon., "Standard Method of Estimating Comparative Direct Operating Costs of Turbine Powered Transport Airplanes," Air Transport Association of America, December 1967.

Table 9.1-1.
Type of Material and Construction Complexity Factors

Type Construction / Type Material	Single Skin with Frames	Sheet Stringer with Frames	Single-Skin Corrugations with Frames	Honey-comb Sandwich
Aluminum	.9	1.0	1.2	1.6
Stainless Steel	1.4	1.5	1.9	2.7
Magnesium	1.5	1.7	2.1	2.7
Titanium	2.0	2.2	2.8	4.2
Inconel-718	2.2	2.4	3.0	4.3
L-605 (Columbium base superalloy)	2.2	2.4	3.0	---
Rene' 41	2.6	2.9	3.6	4.3
TD-NiC	3.2	3.5	4.5	---
Coated columbium (TPS)	---	---	10.0	---

TABLE 9.1-2.
INPUT PARAMETERS - CONTROL PROGRAM

Parameter	Description	Units
ICLASS	Vehicle type. ICLASS = 1 Prototype aircraft 2 Horizontal takeoff launch vehicle (airbreathing) 3 Horizontal takeoff launch vehicle (rocket) 4 Vertical takeoff launch vehicle 5 Spacecraft	
MACH	Maximum Mach number for which airbreathing engines are designed	
NENG	Number of main engines	
NCREW	Number of crew members	
STPS	Thermal protection system surface area for each of 10 areas on the vehicle. Ten values must be specified, any of which may be zero.	sq. ft.
TPEREN	Thrust per main engine	lb.
TOVERW	Thrust-to-weight ratio; $\frac{\text{Total main thrust}}{\text{Takeoff weight}}$	
VMAX	Maximum vehicle velocity	knots
WABORT	Weight of range safety and abort systems	lb.
WACCOM	Weight of passenger accommodations	lb.
WACS	Dry weight of attitude control system	lb.
WACSTK	Weight of attitude control system tankage	lb.
WAERO	Weight of aerodynamic control system	lb.
WAFT	Aft skirt weight (for ICLASS = 4)	lb.
WAUXP	Weight of auxiliary propellants (if separate from main propellants)	lb.
WAUXT	Weight of auxiliary propulsion or separation system	lb.
WBODY	Weight of body structure (for ICLASS = 1, 2, or 3)	lb.
WCANDP	Weight of crew controls and panels	lb.
WCOMM	Weight of communications system	lb.
WDOCK	Weight of docking structure	lb.
WDPLOY	Weight of deployable aerodynamic devices	lb.
WDRY	Total dry weight of vehicle	lb.

TABLE 9.1-2. (Continued)

Parameter	Description	Units
WELCAD	Weight of electrical power conversion and distribution systems	lb.
WEMP	Empennage weight	lb.
WENGS	Weight of all main engines and accessories, except scramjets	lb.
WENGS2	Weight of all secondary and tertiary engines and accessories	lb.
WEQECS	Weight of environmental control system for equipment	lb.
WFAIR	Weight of fairings	lb.
WFTANK	Weight of main fuel tank (structural)(for ICLASS = 4)	lb.
WFUSYS	Weight of fuel system	lb.
WFUTK	Weight of main fuel tank (non-structural)	lb.
WFUTK2	Weight of secondary fuel tank (non-structural)	lb.
WFUTOT	Weight of main fuel	lb.
WFWD	Forward skirt weight (for ICLASS = 4)	lb.
WGASPR	Weight of pressurization gases	lb.
WGNAV	Weight of guidance and navigation system	lb.
WGROSS	Vehicle gross takeoff weight	lb.
WHYCAD	Weight of hydraulic power conversion and distribution system	lb.
WINCOM	Weight of crew compartment insulation	lb.
WINLET	Weight of air inlets and ramps	lb.
WINST	Weight of instrumentation, telemetry, etc.	lb.
WINSTK	Weight of propellant tank insulation	lb.
WINTK	Weight of intertank structure (for ICLASS = 4)	lb.
WLANCH	Weight of launch gear and holddown devices	lb.
WLG	Weight of landing gear	lb.
WLUBE	Weight of lubrication system (turbojets)	lb.
WNACEL	Weight of engine nacelles, pods, pylons, etc.	lb.
WNOSE	Nose structure weight (for ICLASS = 4)	lb.
WOTANK	Weight of main oxidizer tank (structural) (for ICLASS = 4)	lb.
WOXSYS	Weight of oxidizer system	lb.
WOXTK	Weight of main oxidizer tank (non-structural)	lb.
WOXTK2	Weight of secondary oxidizer tank (non-structural)	lb.

TABLE 9.1-2. (Continued)

Parameter	Description	Units
WOXTOT	Weight of main oxidizer	lb.
WPAYL	Payload weight	lb.
WPECS	Weight of environmental control system for personnel compartment	lb.
WPOWER	Weight of electrical power system	lb.
WPPROV	Weight of crew provisions	lb.
WPRESS	Pressurized crew compartment weight (for ICLASS = 4 or 5)	lb.
WPRSYS	Weight of pressurization and purge systems	lb.
WPUSYS	Weight of propellant utilization system	lb.
WRECOV	Weight of vehicle recovery systems	lb.
WSCRAM	Weight of scramjets	lb.
WSERV	Weight of spacecraft service module structure (for ICLASS = 5)	lb.
WSPAD	Adapter structure weight (for ICLASS = 4 or 5)	lb.
WTHRST	Weight of thrust structure (for ICLASS = 4 or 5)	lb.
WWING	Wing weight	

TABLE 9.1-3. INPUT PARAMETERS - COST PROGRAM

Parameter	Description	Units
ADI*	Input value of avionics development cost	M$
AGEOI*	Input value of aerospace ground equipment cost-operational program	M$
AGEPI*	Input value of aerospace ground equipment cost-RDT&E program	M$
ASJMOD	Scramjet capture area (total)	sq. ft.
CONFIG	Engineering complexity factor (nominally 1.0)	
DPLBAU	Auxiliary propellant cost	$/lb
DPLBFU	Main fuel cost (including boiloff factor)	$/lb
DPLBGS	Pressurization gas cost	$/lb
DPLBIN	Cost of insulation for personnel compartment	$/lb
DPLBOX	Main oxidizer cost (including boiloff factor)	$/lb
ENSPAR	Engine spares fraction	
EXPEND	Cost per flight for other expendables (adapters, solid rocket motors, etc.)	M$
FAC	Facilities cost	M$
FEE	Contractor fee, expressed as a fraction (of program cost)	
FTOIN*	Input value of flight test operations cost	M$
HRPFT2	Unit labor cost per flight for thermal protection system maintenance	hr/sq.ft.
IAERO	Indicator for type flight control system: IAERO = 1 for automatic flight control system; IAERO = 0 otherwise	
ICONFIG	Indicator for vehicle type (normally same value as ICLASS)	
IDATA	Indicator to select printout format	
IENG	Indicator for type propulsion system (main): IENG = 1 LOX/LH_2 pump fed rocket 2 Storable pump fed rocket 3 Storable pressure fed rocket 4 Airbreathing propulsion 5 Ablative pressure fed rocket	

TABLE 9.1-3. (Continued)

Parameter	Description	Units
IENG2	Indicator for type secondary propulsion system: IENG2 = 1, 2, 3, 4, or 5 (same as IENG)	
IENG3	Indicator for type tertiary propulsion system: IENG3 = 1, 2, 3, 4, or 5 (same as IENG)	
IOPS	Indicator for type operational program: IOPS = 1 for commercial airline operation; IOPS = 0 otherwise	
IPOWER	Indicator for type electrical power system: IPOWER = 0 for fuel cell; IPOWER = 1 for battery; IPOWER = 2 for aircraft APU	
IPROD	Indicator to select prototype or production tooling IPROD = 1 for production tooling; IPROD = 0 for prototype tooling	
ISRM	Indicator for auxiliary thrust system: ISRM = 1 for solid rocket motors; ISRM = 0 otherwise	
ITANK	Indicator for type main fuel tank: ITANK = 1 for LH_2; ITANK = 2 for storable fuel	
ITANK2	Indicator for type secondary fuel tank: ITANK2 = 1 N_2O_4/N_2H_4; 2 Cryogenic; 3 JP	
IWTR	Indicator for water retrieval IWTR = 1 if water retrieval desired	
LEARN	Learning rate for airframe manufacturing, expressed as a percent (LEARN = 90. for 90% learning curve)	
LEARNP	Learning rate for engine manufacturing, expressed as a percent	
LPM	Launch rate per month	
NDATA	Number of positions on learning curve for which manufacturing costs are desired (normally 1; maximum of 5)	
NENG2	Number of secondary engines	

TABLE 9.1-3. (Continued)

Parameter	Description	Units
NENG3	Number of tertiary engines	
NFTEST	Number of flights in flight test program	
NFV	Number of flight test vehicles	
NFVCO	Number of flight test vehicles converted to operational vehicles	
NG	Number of ground test vehicles	
NMAINT	Number of vehicle maintenance personnel	
NOCON	Number of contractors in concept formulation phase	
NOCON1	Number of contractors in contract definition phase	
NOENG	Number of engineers per contractor in concept formulation phase	
NOENG1	Number of engineers per contractor in contract definition phase	
NOYRS	Duration of concept formulation phase	years
NOYRS1	Duration of contract definition phase	years
NSUPT	Number of base support personnel	
NTRAIN	Number of trainer sets required	
NV	Number of operational vehicles	
NVEH	Number of vehicles at each point on learning curve for which manufacturing costs are to be estimated. Up to 5 values of NVEH may be specified, 1 for each value of NDATA. NVEH = 1 for first unit cost	
PDEN2I*	Input value of secondary propulsion system development cost	M$
PDRJI*	Input value of ramjet or scramjet development cost	M$
PDROCI*	Input value of main rocket engine development cost	M$
PDTJI*	Input value of turbojet development cost	M$
PMGMT	NASA project office cost, expressed as a fraction of program cost	
RATE	Vehicle production rate	Veh./mo.
RE	Engineering labor rate, including overhead and G&A	$/hr
RT	Tooling labor rate, including overhead and G&A	$/hr
SPANEL	Size of thermal protection panels corresponding to the ten vehicle areas (STPS)	sq.ft.

TABLE 9.1-3. (Continued)

Parameter	Description	Units
TOOLC	Tooling complexity factor	
TPREN2	Thrust per engine for secondary propulsion system	lb.
TPREN3	Thrust per engine for tertiary propulsion system	lb.
TURNWK	Vehicle turnaround time in operational phase	weeks
UCABL	Unit cost of ablative thermal protection system	$/sq.ft.
UCCOV	Unit cost of radiative cover panels	$/sq.ft.
UCINS	Unit cost of insulation in thermal protection system	$/sq.ft.
XAVD	Complexity factor for avionics development cost (used to adjust up or down from the nominal 100,000 $/lb)	
XFASSY	Final assembly and checkout cost, expressed as a fraction of first unit manufacturing cost.	
XMRA	Maintenance cost per flight, expressed as a fraction of vehicle first unit cost. XMRA = 0 except to override the computed value	
XMTPS	Material cost per flight for thermal protection system maintenance, expressed as a fraction of first unit thermal protection system cost	
XNEW	Fraction of miscellaneous spacecraft subsystems (including attitude control, environmental control, electrical power, power conversion and distribution, deployable aerodynamic devices, crew controls, aerodynamic controls, crew provisions; and recovery system) which require new development	
XREPL	Fraction of thermal protection system replaced each flight	

* Used to override the value computed with CER's

TABLE 9.1-4. INPUT PARAMETERS - COST PROGRAM

Parameter	Description	Units
CFABRT	Abort system complexity factor (C.F.)	
CFACOM	Passenger accommodations C.F.	
CFACS	Reaction control system C.F.	
CFACTK*	Reaction control tankage C.F.	
CFADAP*	Adapter structure C.F.	
CFAERO	Aerodynamic control system C.F.	
CFAFT*	Aft skirt C.F.	
CFAUXT	Auxiliary thrust system C.F.	
CFBODY*	Fuselage structure C.F.	
CFCOMM	Communication system C.F.	
CFCOMP*	Crew compartment C.F.	
CFCREW	Crew controls and panels C.F.	
CFDOCK*	Docking structure C.F.	
CFDPLY	Deployable aerodynamic devices C.F.	
CFELCD	Electrical distribution system C.F.	
CFEMP*	Empennage structure C.F.	
CFENG	Airbreathing engine C.F. (main)	
CFENG2	Airbreathing engine C.F. (secondary)	
CFENG3	Airbreathing engine (tertiary)	
CFEQEC	Equipment ECS C.F.	
CFFAIR*	Aerodynamic fairing structure C.F.	
CFFUEL	Fuel system C.F.	
CFFUTK*	Structural fuel tank C.F.	
CFFWD*	Forward skirt C.F.	
CFGNAV	Guidance and navigation system C.F.	
CFHYCD	Hydraulic system C.F.	
CFINLT*	Inlet structure C.F.	
CFINST	Instrumentation system C.F.	
CFINTK*	Intertank structure C.F.	
CFLG	Landing gear C.F.	
CFLNCH*	Launch structure C.F.	
CFLUBE	Engine lubrication system C.F.	

TABLE 9.1-4. (Continued)

Parameter	Description	Units
CFNAC*	Engine nacelle C.F.	
CFNOSE*	Nose structure C.F.	
CFOX	Oxidizer system C.F.	
CFOXTK*	Structural oxidizer tank C.F.	
CFPECS	Personnel ECS C.F.	
CFPOW	Electrical power system C.F.	
CFPRES	Pressurization system C.F.	
CFPROV	Crew provisions C.F.	
CFPUSY	Propellant utilization system C.F.	
CFRECV	Recovery system C.F.	
CFSERV*	Spacecraft service module structure C.F.	
CFTHRS*	Thrust structure C.F.	
CFTK*	Non-structural propellant tank C.F.	
CFTK2*	Non-structural propellant tank C.F. (secondary propulsion system)	
CFWING*	Wing C.F.	

ORIGINAL PAGE IS OF POOR QUALITY

* Value of complexity figure obtained from Table A.

TABLE A. TYPE OF MATERIAL AND CONSTRUCTION COMPLEXITY FACTORS

Type Construction / Type Material	Single Skin With Frames	Sheet Stringer With Frames	Single Skin Corrugations With Frames	Honeycomb Sandwich
Aluminum	.9	1.0	1.2	1.6
Stainless Steel	1.4	1.5	1.9	2.7
Magnesium	1.5	1.7	2.1	2.7
Titanium	2.0	2.2	2.8	4.2
Inconel-718	2.2	2.4	3.0	4.3
L-605	2.2	2.4	3.0	-
Rene' 41	2.6	2.9	3.6	4.3
TD-NiC	3.2	3.5	4.5	-
Coated Columbium	-	-	10.0	-

TABLE 9.1-5. OUTPUT PARAMETERS

The following output will appear, in the order shown, for each case unless IDATA is used to suppress printout or unless the particular cost element is not applicable to the vehicle under consideration. The abbreviation f.u.m.c. designates first unit manufacturing cost. All cost output is in millions of dollars.

Parameter	Description
CSURF	Aerodynamic surface f.u.m.c.
CBODY	Body structure f.u.m.c.
CTPS	Thermal protection system f.u.m.c.
CLG	Landing gear f.u.m.c.
CLRD	Launch, recovery, and docking gear f.u.m.c.
CENGS	Main engines f.u.m.c. (total per vehicle)
C(6,N)	Sum of secondary and tertiary engines f.u.m.c. (total per vehicle)
CINLET	Air induction system f.u.m.c.
CNACEL	Nacelles f.u.m.c.
CFUTK	Main fuel tank f.u.m.c.
COXTK	Main oxidizer tank f.u.m.c.
CFUTK2	Secondary fuel tank f.u.m.c.
COXTK2	Secondary oxidizer tank f.u.m.c.
CINSTK	Tank insulation f.u.m.c.
CFUSYS	Fuel system f.u.m.c.
COXSYS	Oxidizer system f.u.m.c.
CPRSYS	Pressurization system f.u.m.c.
CPUSYS	Propellant utilization system f.u.m.c.
CLUBE	Engine lubrication system f.u.m.c.
CAERO	Aerodynamics control system f.u.m.c.
CORSUL	Orientation, separation, and ullage control system f.u.m.c.
CPOWER	Electrical power system f.u.m.c.
CPOWCD	Power conversion and distribution system f.u.m.c.
CGNAV	Guidance and navigation system f.u.m.c.
CINST	Instrumentation f.u.m.c.

TABLE 9.1-5. (Continued)

Parameter	Description
CCOMM	Communication system f.u.m.c.
CEQECS	Equipment environmental control system f.u.m.c.
CPECS	Personnel environmental control system f.u.m.c.
CINCOM	Personnel compartment insulation f.u.m.c.
CPROV	Personnel provisions f.u.m.c.
CCANDP	Crew controls and panels f.u.m.c.
CABORT	Abort system f.u.m.c.
CFASSY	Final assembly and checkout cost
CV	Total vehicle f.u.m.c.
CFUEL	Main fuel cost per launch
COXID	Main oxidizer cost per launch
CAUXP	Auxiliary propellants cost per launch
CGASPR	Pressurization gases cost per launch
TRDTE	Total RDT&E cost
ADDE	Total airframe design and development engineering cost, including concept formulation and contract definition
CF	Concept formulation phase cost
CD	Contract definition phase cost
DDEL	Airframe design and development engineering cost
SUBSYS	Miscellaneous subsystem development cost
AD	Avionics development cost
PD	Propulsion development cost
CV	Total vehicle f.u.m.c.
AMFG	Total airframe f.u.m.c.
AVP	Total avionics f.u.m.c.
PROPU	Total engines f.u.m.c.
NFV	Number of flight vehicles
FV	Flight vehicle cost
NG	Number of ground test vehicles
GTV	Ground test vehicles cost
GTS	Ground test spares cost
FTS	Flight test spares cost
TST	Tooling and special test equipment cost

TABLE 9.1-5. (Continued

Parameter	Description
FTO	Flight test operations cost
AGEP	Development program GSE Cost
TDP	Technical data cost - RDT&E
RDFEE	Contractor fee - RDT&E
RDMGMT	Government project management cost - RDT&E
AQ	Total acquisition cost
NV	Number of operational vehicles
OV	Operational vehicles cost
OS	Operational vehicle spares cost
FAC	Facility investment cost
SE	Sustaining engineering cost
ST	Sustaining tooling cost
AGEO	Operational GSE cost
MEQ	Miscellaneous equipment cost
IT	Initial training cost
OT	Training equipment cost
TRI	Initial transportation cost
AQFEE	Contractor fee - acquisition phase
AQMGMT	Government project management - acquisition phase
ROTOT	Total 10-year cost of recurring operations
NPL	Total number of flight crew personnel
NMPR	Total number of maintenance personnel
NSPR	Total number of support personnel
WSA	Total 10-year cost of flight crew and support personnel pay and allowances
VM	Total 10-year cost of vehicle maintenance, including maintenance personnel pay
WTR	Total 10-year cost of water retrieval
PF	Total 10-year cost of propellants and gases
MFL	Total 10-year cost of miscellaneous expendables
FM	Total 10-year cost of facility maintenance
AGEM	Total 10-year cost of GSE maintenance
MEM	Total 10-year cost of miscellaneous equipment maintenance

TABLE 9.1-5. (Continued)

Parameter	Description
TO	Training - recurring operations
TRO	Transportation - recurring operations
ROFEE	Contractor fee - recurring operations
ROMGMT	Government project office cost - recurring operations
TOTAL	Total program cost, including RDT&E, acquisition, and recurring operations

TABLE 9.1-6. SAMPLE CASE

```
    MINI SHUTTLE ORBITER 002 (3-27-70)
ICLASS=5,  NENG=2., TPEREN=297000.,  WWING=13260., WPRESS=15260.,
WLG=4800.,  WDPLOY=0.,  WDOCK=550.,  WLANCH=0.,  WRECOV=0., WENGS=6875.,
WENGS2=3692., WINLET=0.,  WNACEL=1122.,  WFUTK=12890.,  WOXTK=0.,
WFUTK2=1727.,  WOXTK2=586.,  WINSTK=602.,  WFUSYS=1237.,  WOXSYS=1383.,
WPRSYS=4405.,  WPUSYS=323.,  WLUBE=0.,  WAERO=1095.,  WAUXT=0.,
WACS=2573.,  WACSTK=0.,  WPOWER=649.,  WELCAD=1465.,  WHYCAD=1644.,
              WGNAV=790.,  WINST=885.,  WCOMM=165.,  WEOECS=0.,
WPECS=1123.,                WINCOM=0.,  WPPROV=200.,  WCANDP=460.,
WABORT=0.,  WFUTOT=47200.,  WOXTOT=282600.,  WAUXP=7012., WGASPR=1170.,
WGROSS=439971.,  WDRY=92662.,  WPAYL=15400.,            NCREW=2.,
            STPS=1690., 1510.,6533.,610., 900., 2180., 4X0.,
*
NVEH=1., NFV=1., NG=1., NV=6., DPLBFU=.45, DPLBOX=.03, DPLBAU=.03,
DPLBGS=3.00, IENG=1, ITANK=1, DPLBIN=200., PDROCI=0.,    XMTPS=.0541,
                FEE=.10, PMGMT=.10, IENG2=4, RE=18.50, RT=14.50,
         FAC=0., IAERO=1, LPM=2.5, ICONFG=5, TOOLC=7.2,
        IPOWER=0, XFASSY=.20, ITANK2=2, PDEN2I=121., HRPFT2=16.,
XREPL=.054,NFTEST=150., IENG3=1, NENG2=6., NENG3=2., TPREN2=6000.,
TPREN3=15000.,
UCABL=0., 0.,  0.,   2000., 6X0.,
UCCOV=4300., 1000., 2000., 7X0.,
UCINS=0., 0., 0., 0., 200., 100., 4X0.,
SPANEL=100., 100., 4., 7X100.,
*
CFWING=2.2, CFCOMP=2.8, CFTK=1.5,    *
```

TABLE 9.1-6. (Continued)

MANUFACTURING COST BREAKDOWN MILLION DOLLARS	
NUMBER OF VEHICLES	1.
AERODYNAMIC SURFACES	5.7278
BODY STRUCTURE	24.5937
INDUCED ENVIRONMENTAL PROTECTION	16.0736
LAUNCH,RECOVERY AND DOCKING	1.2979
MAIN PROPULSION	
ENGINES AND ACCESSORIES	6.4943
SECONDARY ENGINES AND ACCESSORIES	4.2041
NACELLES,PODS,PYLONS,SUPPORTS	1.3320
FUEL CONTAINERS AND SUPPORTS	2.3314
SECONDARY FUEL TANKAGE AND SYSTEMS	0.4453
SECONDARY OXIDIZER TANKAGE AND SYSTEMS	0.2418
PROPELLANT INSULATION	0.1204
FUEL SYSTEM - MAIN	1.2606
OXIDIZER SYSTEM - MAIN	1.3225
PRESSURIZATION AND PURGE SYSTEMS	2.1764
PROPELLANT UTILIZATION SYSTEM	0.7076
AERODYNAMIC CONTROLS	2.2784
ORIENTATION,SEPARATION AND ULLAGE CONTROL	3.9301
PRIME POWER SOURCES	1.1610
POWER CONVERSION AND DISTRIBUTION	4.8753
GUIDANCE AND NAVIGATION	6.1795
INSTRUMENTATION	0.7276
COMMUNICATION	1.5487
ENVIRONMENTAL CONTROL	
EQUIPMENT ECS	0.0
PERSONNEL ECS	2.7811
COMPARTMENT INSULATION	0.0
PERSONNEL PROVISIONS	0.1300

TABLE 9.1-6. (Continued)

CREW STATION CONTROLS AND PANELS	0.5493
FINAL ASSEMBLY AND CHECKOUT	18.2257
DRY STRUCTURE TOTAL	109.3545
EXPENDABLES COST PER LAUNCH	
FUEL	0.0212
OXIDIZER	0.0085
AUXILIARY PROPELLANTS	0.0002
PRESSURIZATION GASES	0.0035

TABLE 9.1-6. (Continued)

				COST(MILLIONS OF DOLLAR
RESEARCH,DEVELOPMENT,TEST,AND EVALUATION				0.24003E 0
AIRFRAME DESIGN AND DEVELOPMENT ENGINEERING			0.68896E 03	
CONCEPT FORMULATION		0.11812E 02		
CONTRACT DEFINITION		0.35000E 02		
AIRFRAME ENGINEERING		0.64215E 03		
MISCELLANEOUS SUBSYSTEM DEVELOPMENT			0.32231E 03	
AVIONICS DEVELOPMENT			0.13058E 03	
PROPULSION DEVELOPMENT			0.12100E 03	
DEVELOPMENT SUPPORT			0.73738E 03	
(MANUFACTURING--FIRST UNIT)		(0.10935E 03)		
AIRFRAME	(0.90200E 02)			
AVIONICS PROCUREMENT	(0.84558E 01)			
PROPULSION PROCUREMENT	(0.10698E 02)			
FLIGHT VEHICLES(1.0)		0.10935E 03		
GROUND TEST VEHICLES(1.0)		0.90200E 02		
GROUND TEST SPARES		0.90200E 01		
FLIGHT TEST SPARES		0.21871E 02		
TOOLING AND SPECIAL TEST EQUIPMENT		0.96444E 02		
FLIGHT TEST OPERATIONS		0.35745E 03		
AGE		0.50851E 02		
TECHNICAL DATA		0.21871E 01		
FEE			0.20002E 03	
PROGRAM MANAGEMENT			0.20002E 03	
INITIAL INVESTMENT				0.12237E 04
OPERATIONAL VEHICLES(6.0)			0.47806E 03	
SPARES			0.75455E 02	
FACILITIES			0.0	
SUSTAINING ENGINEERING			0.29692E 03	
SUSTAINING TOOLING			0.27497E 02	
AGE			0.71710E 02	
TECHNICAL DATA			0.95613E 01	
MISCELLANEOUS EQUIPMENT			0.88900E 00	
TRAINING EQUIPMENT			0.13734E 03	
INITIAL TRAINING			0.12000E 01	
INITIAL TRANSPORTATION			0.38173E 01	
FEE			0.11025E 03	
PROGRAM MANAGEMENT			0.11025E 02	
RECURRING OPERATING(10 YEAR)				0.72017E 03
WAGES,SALARIES,ALLOWANCES(24 PILOTS 488 MAINT. 1266 SUPPORT PERS)			0.35382E 03	
VEHICLE MAINTENANCE			0.24974E 03	
PROPELLANTS			0.10031E 02	
MISCELLANEOUS EXPENDABLES			0.0	

TABLE 9.1-6. (Continued)

FACILITIES MAINTENANCE	0.0
AGE MAINTENANCE	0.21513E 02
MISCELLANEOUS EQUIPMENT	0.97790E 00
TRAINING	0.96000E 01
TRANSPORTATION	0.48783E 01
FEE	0.32528E 02
PROGRAM MANAGEMENT	0.37082E 02
TOTAL SYSTEMS COST	0.43442E 04

Figure 9 1-1. Total Program Cost Element Structure

Total Program Cost

RDT&E	Acquisition	Operations
Concept Formulation	Operational Vehicles	Pilot & Support Personnel
Contract Definition	Spares	Vehicle Maintenance
Airframe Design & Dev. Engineering	Sustaining Engineering	Propellants
Avionics Development	Sustaining Tooling	Miscellaneous Expendables
Miscellaneous Subsystem Dev.	AGE	Facility Maintenance
Propulsion Development	Data	AGE Maintenance
Flight Test Vehicles	Miscellaneous Equipment	Miscellaneous Equipment Maint.
Flight Test Spares	Training Equipment	Training
Ground Test Hardware	Initial Training	Transportation
Ground Test Spares	Initial Transportation	Contractor Fee
Tooling & Special Test Equipment	Contractor Fee	Program Management (Government)
Flight Test Operations	Program Management (Government)	
AGE		
Data		
Contractor Fee		
Program Management (Government)		

Figure 9.1-2. First Unit Manufacturing Cost Element Structure

<u>First Unit Manufacturing Cost</u>

- Aerodynamic Surfaces
  - Wing
  - Empennage
  - Fairings
- Body Structure
- Thermal Protection System
- Launch, Recovery, & Docking
  - Landing Gear
  - Deployable Aerodynamic Devices
  - Docking Structure
  - Launch Gear
  - Recovery Gear
- Main Engines
- Secondary Engines
- Air Induction System
- Nacelles
- Fuel Tank (Non-Structural)
- Oxidizer Tank (Non-Structural)
- Secondary Fuel Tank (Non-Structural)
- Secondary Oxidizer Tank (Non-Structural)
- Tank Insulation
- Fuel System
- Oxidizer System
- Pressurization System
- Propellant Utilization System
- Engine Lubrication System
- Aerodynamic Control System
- Orientation, Separation, & Ullage Control
  - Auxiliary Thrust System
  - Reaction Control System
  - Reaction Control System Tankage
- Electrical Power Source
- Power Distribution System
  - Electrical Distribution
  - Hydraulics & Pneumatics
- Navigation & Guidance
- Instrumentation
- Communications
- Environmental Control (Equipment)
- Environmental Control (Personnel)
- Crew Compartment Insulation
- Crew Provisions
- Controls & Panels
- Abc_: System
- Final Assembly & Checkout

Figure 9.1-3. Body Structure Cost Elements

VTO Launch Vehicle	Spacecraft	Aircraft
Crew Compartment	Crew Compartment	Fuselage
Thrust Structure	Thrust Structure	
Forward Adapter	Service Module	
Forward Skirt	Adapter	
Intertank Structure		
Aft Skirt		
Fuel Tank		
Oxidizer Tank		
Nose Structure		

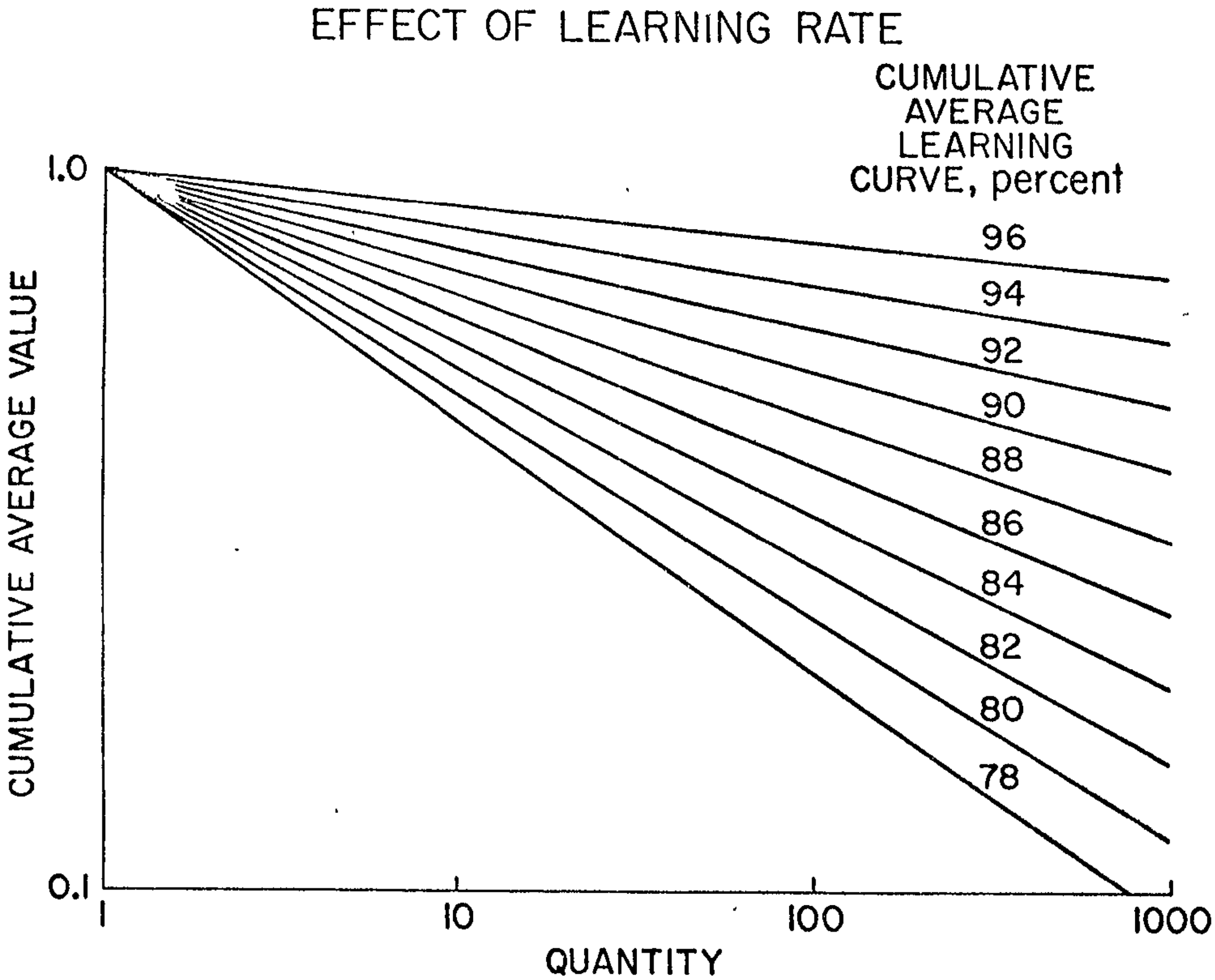


Figure 9.1-4. Effect of Learning Rate

TABLE OF CONTENTS FOR SECTION 9.2, PROGRAM DAPCA

9.2 PROGRAM DAPCA: DETERMINING AIRCRAFT DEVELOPMENT AND PRODUCTION COSTS

Program DAPCA was originally developed by the Rand Corporation; complete program details are given in Reference. Analytic models are discussed in References 1 through 3. The presentation below is limited to a listing of the equations employed, the program input, and the program output. The ODIN/RLV version of DAPCA is an Aerophysics Research Corporation conversion of the original IBM 7094 Rand program modified for the CDC 6600. The converted program is identical in function to the original RAND program except for the use of NAMELIST input in place of a rigid format input.

9.2.1 Basic Equations

This section presents the basic equations used in the DAPCA computer program. These equations are listed for each major category of cost (airframe components, engines, and avionics) and for the totals. All costs are calculated in millions of dollars. A listing of the constants and symbols is presented in Sections 9.2.1.3 and 9.2.2, respectively.

9.2.1.1 RDT&E Costs

(a) Airframes

Initial Engineering Cost

$$EI = 10^{0.90462} \cdot S^{0.54716} \cdot TT^{0.88} \cdot 12.25 \cdot DM \cdot CR1$$

Development Support Cost

$$DS = EI \cdot \frac{15.6}{12.25}$$

Initial Tooling Cost

$$TI = 10^{-0.91248} \cdot S^{1.07437} \cdot W^{0.83913} \cdot 10.55 \cdot DM \cdot CR2$$

Flight Test Cost

$$FTC = 0.58 \cdot TA^{1.1} \cdot W^{0.80} \cdot S^{0.90} \cdot (DM)$$

Total Airframe Cost of Test Aircraft

$$CT = TA \cdot AFA_1$$ (AFA_1 is calculated in the section on production costs)

Total Airframe Cost of Test Aircraft Including Engines and Avionics for Test Aircraft

$$CTE = CT + TCT + AVT$$

Total Airframe RDT&E Cost

$$RDA = EI + TI + DS + FTC + CTE$$

(b) Engines

Initial Development Cost (If Not Entered as Input)

$$DEI_1 = \begin{cases} 0.13937 \cdot T^{0.74356} \cdot CR5, & \text{(turbojet or turbofan)} \\ 2.82917 \cdot ESH^{0.35497} \cdot CR5, & \text{(turboprop)} \end{cases}$$

Total Engine Development Cost

$$RDE = DEI$$

$$DEI = DEI_1 + EDAC$$

(c) Avionics

$$RDAV = \text{Input}$$

(d) Totals

Total RDT&E Cost (Airframes, Engines, and Avionics)

$$RDT = RDA + RDE + RDAV$$

Grand Total Cost of Aircraft for Airframe Quantity QT (Airframes, Engines and Avionics)

$$TCOST_I = TC_I + RDT, \qquad (I > 1)$$

9.2.1.2 Production Costs

(a) Airframes

Unit Labor for Airframe Number 1

$$PAL = 5.762 \cdot 10^{0.16314} \cdot W^{0.73672} \cdot S^{0.43113} \cdot 9.50 \cdot 1.11 \cdot 1.14 \cdot DM \cdot CR3$$

Unit Material for Airframe Number 1

$$PAM = 2.169 \cdot 10^{-0.76558} \cdot W^{0.77933} \cdot S^{0.856650} \cdot DM \cdot CR4$$

Cumulative Average to Unit Ratio for Airframe Labor

$$BLA_I = \frac{(QT + 0.5)^{0.585} - 0.5^{0.585}}{QT^{0.585} \cdot 0.585}$$

CumulativeAverage to Unit Ratio for Airframe Material

$$BMA_I = \frac{(QT + 0.5)^{0.832} - 0.5^{0.832}}{QT^{0.832} \cdot 0.832}$$

Unit Labor Cost at Quantity QT (Total Aircraft Including Test Aircraft)

$$ULA_I = PAL \cdot QT^{-0.415}$$

Total Labor Cost for Aircraft Quantity QT

$$CALAC = ULA_I \cdot BLA_I \cdot QT$$

Cumulative Average Labor Cost for Test Aircraft

$$CALA_I = \frac{CALAC_{TA}}{QT}, \qquad (I = 1,\ QT = TA)$$

Cumulative Average Labor Cost for Aircraft Quantity QT Less Test Aircraft

$$CALA_I = \frac{CALAC_{QT} - CALAC_{TA}}{QT - TA}, \qquad (I > 1,\ QT > TA)$$

Unit Material Cost at Aircraft Quantity QT

$$UMA_I = PAM \cdot QT^{-0.168}$$

Total Material Cost for Aircraft Quantity QT

$$CAMAC = UMA_I \cdot BMA_I \cdot QT$$

Cumulative Average Material Cost for Test Aircraft

$$CAMA_I = \frac{CAMAC_{TA}}{QT}, \qquad (I = 1,\ QT = TA)$$

Cumulative Average Material Cost for Aircraft Quantity QT Less Test Aircraft

$$CAMA_I = \frac{CAMAC_{QT} - CAMAC_{TA}}{QT - TA}, \qquad (I > 1,\ QT > TA$$

Airframe Production Rate Calculation

$Q2 = QT$

If $QT > QM$, Q2 is set equal to QM

$$QTEMP = \frac{\pi}{2 \cdot QM} \cdot Q2$$

$PR_I = PRM \cdot \sin^2$ (QTEMP, (PR_I is rounded to nearest tenth)

If $PR_I < 1$, PR_I is set equal to 1

Total Sustaining Tooling Cost for Aircraft Quantity QT

$$STAC = (PR_I^{0.4} \cdot QT^{0\ 138} - 1) \cdot TI$$

Cumulative Average Sustaining Tooling Cost for Test Aircraft

$$STA_I = \frac{STAC_{TA}}{QT}, \qquad (I = 1,\ QT = TA)$$

Cumulative Average Sustaining Tooling Cost for Aircraft Quantity QT Less Test Aircraft

$$STA_I = \frac{STAC_{QT} - STAC_{TA}}{QT - TA}, \qquad (I > 1,\ QT > TA)$$

Total Sustaining Tooling Cost for Aircraft Quantity QT - 1 (Q1)

$$Q1 = QT - 1$$

$$STA1 = (PR_I{}^{0.4} \cdot Q1^{0.138} - 1) \cdot TI, \qquad (PR_I \text{ is recalculated for Q1})$$

Unit Sustaining Tooling Cost at Aircraft Quantity QT

$$STU_I = STAC - STA1$$

Total Sustaining Engineering Cost for Aircraft Quantity QT

$$SEAC = EI \cdot (QT^{0.2} - 1)$$

Cumulative Average Sustaining Engineering Cost for Test Aircraft

$$SEA_I = \frac{SEAC_{TA}}{QT}, \qquad (I = 1,\ QT = TA)$$

Cumulative Average Sustaining Engineering Cost for Aircraft Quantity QT Less Test Aircraft

$$SEA_I = \frac{SEAC_{QT} - SEAC_{TA}}{QT - TA}, \qquad (I > 1,\ QT > TA)$$

Total Sustaining Engineering Cost for Aircraft Quantity QT - 1

$$SEA1 = EI \cdot (Q1^{0.2} - 1)$$

Unit Sustaining Engineering Cost at Aircraft Quantity QT

$$SEU_I = SEAC - SEA1$$

Total Cumulative Average Cost for Test Aircraft (I=1) or for Aircraft Quantity QT less Test Aircraft (I > 1)

$$AFA_I = SEA_I + STA_I + CALA_I + CAMA_I$$

Total Unit Cost at Aircraft Quantity QT

$$AFU_I = SEU_I + STU_I + ULA_I + UMA_I$$

(b) Engines

Quantity of Engines Required for Aircraft Quantity QT

$$S_I = V \cdot QT$$

Total Engine Recurring Development Cost for Aircraft Quantity QT

$$ENDAC = \begin{cases} DEI \cdot (X_I^{0.07751} - 1), & \text{(Turbojet or turbofan)} \\ DEI \cdot (X_I^{0.09334} - 1), & \text{(Turboprop)} \end{cases}$$

Production Cost of Engine Number 1 (If Not Entered as Input)

$$CAPE = \begin{cases} 0.18700 \cdot T^{0.84845} \cdot CR6, & \text{(turbojet with afterburner)} \\ 0.31979 \cdot T^{0.81626} \cdot CR6 & \text{(turbojet with no afterburner)} \\ (TFW \cdot 0.18700 \cdot T^{0.84845} + TFN \cdot 0.31979 \cdot T^{0.81626}) \cdot CR6 & \text{(turbofan)} \\ (4.86224 \cdot ESH^{0.45873} \cdot CR6, & \text{(turboprop)} \end{cases}$$

Total Engine Production Cost for Aircraft Quantity QT

$$PAC = \begin{cases} CAPE \cdot X_I^{0.86745}, & \text{(turbojet with afterburner)} \\ CAPE \cdot X_I^{0.87088}, & \text{(turbojet with no afterburner)} \\ CAPE \cdot (TFW \cdot X_I^{0.86745} + TFN \cdot X_I^{0.87088}), & \text{(turbofan)} \\ CAPE \cdot X_I^{0.89055}, & \text{(turboprop)} \end{cases}$$

Engine Cumulative Average Production Cost for Test Aircraft

$$PA_I = \frac{PAC_{TA}}{X_I}, \qquad (I = 1,\ X_I = V \cdot TA$$

Engine Cumulative Average Production Cost for Aircraft Quantity QT Less Test Aircraft

$$PA_I = \frac{PAC_{QT} - PAC_{TA}}{V \cdot (QT-TA)} , \qquad (I > 1,\ QT > TA$$

Total of Engine Recurring Development and Production Cost for Test Aircraft

$$TCT = EDAC$$

Total Engine Production Cost for Engine Quantity $X_I - 1$ (X1)

$$X1 = X_I - 1$$

$$PA1 = \begin{cases} CAPE \cdot X1^{0.86745}, & \text{(turbojet with afterburner)} \\ CAPE \cdot X1^{0.87088}, & \text{(turbojet with no afterburner)} \\ CAPE \cdot (TFW \cdot X1^{0.86745} + TFN \cdot X1^{0.87088}), & \text{(turbofan)} \\ CAPE \cdot X1^{0.89055}, & \text{(turboprop)} \end{cases}$$

Engine Unit Production Cost at Engine Quantity X_I

$$PU_I = PAC - PA1$$

Total Engine Unit Production Cost per Aircraft at Aircraft Quantity QT

$$UEA_I = (PAC - PA1_{X_I-1}) + PA1_{X_I-1} - PA1_{X_I-2})$$
$$+ (PA1_{X_I-2} - PA1_{X_I-3}) + \ldots (PA1_{X_I-V-1)} - PA1_{X_I-V})$$

$$UEA_I = PAC - PA1_{X_I-V}$$

(c) Avionics

Total Avionics Production Cost for Aircraft Quantity QT

$$AVAC = AVI \cdot QT^{0.81558}$$

Avionics Cumulative Average Production Cost for Test Aircraft

$$AVC_I = \frac{AVAC_{TA}}{QT} \qquad (I = 1,\ QT = TA)$$

Avionics Cumulative Average Production Cost for Aircraft Quantity QT Less Test Aircraft

$$AVC_I = \frac{AVAC_{QT} - AVAC_{TA}}{QT - TA}, \qquad (I > 1,\ QT > TA)$$

Total Avionics Production Cost for Aircraft Quantity QT - 1 (Q1)

$$AVA1 = AVI \cdot Q1^{0.81558}, \qquad (QT > 1)$$

Avionics Unit Production Cost at Aircraft Quantity QT

$$AVU_I = AVAC - AVA1$$

(d) Totals

Total Unit Cost at Aircraft Quantity QT (Airframes, Engines, and Avionics)

$$TUN_I = AFU_I + UEA_I + AVU_I$$

Total Cumulative Average Cost for Test Aircraft (Airframes, Engines, and Avionics)

$$TCA_I = \frac{(CALAC + CAMAC + STAC + SEAC + PAC + AVAC)_{TA}}{QT}$$

$$(I = 1,\ QT = TA)$$

Total Cumulative Average Production Cost for Aircraft Quantity QT Less Test Aircraft (Airframes, Engines, and Avionics)

$$C_{QT} = (CALAC + CAMAC + STAC + SEAC + PAC + AVAC)_{QT}$$

$$C_{TA} = (CALAC + CAMAC + STAC + SEAC + PAC + AVAC)_{TA}$$

$$TCA_I = \frac{C_{QT} - C_{TA}}{QT - TA}, \qquad (I > 1, QT > TA)$$

Total Aircraft Cumulative Average Cost Including RDT&E for Quantity QT (Airframes, Engines, and Avionics)

$$TCQ = \frac{TCOST_I}{QT}$$

Total Production Cost for Test Aircraft

$$TC_I = TCA_I \cdot QT, \qquad (I = 1, QT = TA)$$

Total Production Cost for Aircraft Quantity QT Less Test Aircraft

$$TC_I = TCA_I \cdot (QT - TA), \qquad (I > 1, QT > TA)$$

9.2.1.3 Constants

12.25	Engineering direct labor--overhead, G&A, and miscellaneous direct charges in 1965 constant dollars
15.6	Dollars per engineering hour required for development support
10 55	Tooling direct labor -- overhead, G&A, miscellaneous direct charges, and material in 1965 constant dollars
9.50	Production direct labor -- overhead, G&A, and miscellaneous direct charges
1 11	Engineering changes as a percentage of production costs
1.14	Quality control as a percentage of production costs
6.762	Conversion factor from unit 100 to unit 1
2.169	Conversion factor from unit 100 to unit 1

9.2.2 List of Symbols Used in Equations

9.2.2.1 Inputs

Symbol	Meaning
AVI	Avionics production cost of unit number one
CR1	Adjustment factor for airframe initial engineering cost
CR2	Adjustment factor for airframe initial tooling cost
CR3	Adjustment factor for airframe labor cost
CR4	Adjustment factor for airframe material cost
CR5	Adjustment factor for engine initial development cost
CR6	Adjustment factor for production cost of engine number one
DM	$10^{-6} \cdot (L+P)$. (For P, see below)
ESH	Equivalent shaft horsepower per engine
P	Airframe profit factor
PRM	Maximum airframe production rate (airframes per month)
QM	Quantity at which maximum airframe production rate is first achieved
RDAV	Avionics RDT&E cost (in millions of dollars)
S	Maximum speed of aircraft at best altitude (knots)
T	Maximum thrust (pounds) per engine (cruise engine, sea-level static)
TA	Number of test aircraft
TFW	Turbofan weighting factor, applied to cost of turbojet with afterburner
V	Number of engines per aircraft
W	Gross take-off weight of aircraft (pounds)

9.2.2.2 Outputs

Symbol	Meaning
AFA_I	Total airframe cumulative average cost
AFU_I	Total unit cost at aircraft quantity QT
AVAC	Total avionics production cost for aircraft quantity QT
AVC_I	Avionics cumulative average production cost
AVA1	Total avionics production cost for aircraft quantity QT-1
AVT	Total avionics production cost for test aircraft
AVU_I	Avionics unit production cost at aircraft quantity QT
BLA_I	Ratio of cumulative average cost to unit cost for airframe labor
BMA_I	Ratio of cumulative average cost to unit cost for airframe material
$CALA_I$	Airframe cumulative average labor cost
CALAC	Total airframe labor cost for quantity QT
$CAMA_I$	Airframe cumulative average material cost
CAMAC	Total airframe material cost for quantity QT
CAPE	Production cost of engine number one
CT	Total airframe cost of test aircraft
CTE	Total airframe cost of test aircraft including engines and avionics for test aircraft
DEI_1	Engine initial development cost (may also be entered as an input)
DEI	Total engine development cost
DS	Airframe development support cost
EDAC	Total engine recurring development cost
EI	Airframe initial engineering cost

Symbol	Meaning
FTC	Airframe flight test cost
I	Used as subscript to some of the variables. (For test aircraft I = 1, for any aircraft quantity QT less test aircraft, I > 1.)
PA_I	Engine cumulative average production cost
PAC	Total engine production cost for aircraft quantity QT
PA1	Total engine production cost for aircraft quantity QT - 1.
PAL	Unit labor cost for airframe number one
PAM	Unit material cost for airframe number one
PR_I	Airframe production rate (airframes per month)
PU_I	Engine unit production cost at engine quantity X_I
QT	Total aircraft quantity (for I = 1, QT = TA; for I > 1, QT > TA).
Q1	QT - 1
Q2	Set equal to QT. However, if QT > QM, Q2 is set equal to AM
QTEMP	$\frac{\pi}{2 \cdot QM} \cdot Q2$
RDA	Total airframe RDT&E cost
RDE	Total engine development cost (set equal to DEI)
RDT	Total aircraft RDT&E cost (airframes, engines, and avionics)
SEA_I	Airframe cumulative average sustaining engineering cost
SEAC	Total airframe sustaining engineering cost for quantity QT
SEA1	Total airframe sustaining engineering cost for quantity QT - 1
SEU_I	Airframe unit sustaining engineering cost at quantity QT
SIN	Sine function
STA_I	Airframe cumulative average sustaining tooling cost

Symbol	Meaning
STAC	Total airframe sustaining tooling cost for quantity QT
STA1	Total airframe sustaining tooling cost for quantity QT - 1
STU_I	Airframe unit sustaining tooling cost at quantity QT
TC_I	Total aircraft production cost (airframes, engines, and avionics)
TCA_I	Total aircraft cumulative average cost (airframes, engines, and avionics)
$TCOST_I$	Grand total cost of aircraft for quantity QT (RDT&E and production costs for airframes, engines, and avionics)
TCQ	Total aircraft cumulative average cost including RDT&E for quantity QT (airframes, engines, and avionics)
TCT	Total of engine recurring development cost and production cost for test aircraft
TFN	Turbofan weighting factor, applied to cost of turbojet with no afterburner (TFN = 1.0 - TFW).
TI	Airframe initial tooling cost
TT	Total maximum thrust (pounds) per aircraft (cruise engines, sea level static) $(V \cdot T)$
TUN_I	Total aircraft unit cost at quantity QT (airframes, engines, and avionics)
ULA_I	Airframe unit labor cost at quantity QT
UMA_I	Airframe unit material cost at quantity QT
X_I	Quantity of engines required for aircraft quantity $QT(V \cdot QT)$
ZPT	Total engine unit production cost per aircraft at aircraft quantity QT

SYSTEM A
RUN 1

PAGE 1

INPUT DATA

PRINCIPAL INPUTS

* GROSS TAKEOFF WEIGHT OF AIRCRAFT (IN POUNDS)	200000.
* SPEED OF AIRCRAFT (IN KNOTS)	500.
MAXIMUM PRODUCTION RATE (PRM)	10.
QUANTITY AT WHICH PRM IS FIRST ACHIEVED	50.
EXPONENT FOR PRODUCTION RATE EQUATION	2.00
NUMBER OF TEST AIRCRAFT	3.
NUMBER OF ENGINES PER AIRCRAFT	4.
* THRUST PER ENGINE (IN POUNDS) -- TURBOFAN	18000.
WEIGHTING FACTOR FOR TURBOJET WITH AFTERBURNER	0.50
AVIONICS RDTE COST (IN MILLIONS OF DOLLARS)	5.000
AVIONICS COST FOR UNIT 1 (IN MILLIONS OF DOLLARS)	0.300
AIRFRAME PROFIT FACTOR	0.10

AIRFRAME ADJUSTMENT FACTORS

ENGINEERING	1.00
TOOLING	1.00
LABOR	1.00
MATERIAL	1.00

ENGINE ADJUSTMENT FACTORS

DEVELOPMENT	1.00
PRODUCTION	1.00

DESIGNATORS

ENGINE TYPE (ADDRESS 007)	3.
ENGINE THRUPUT (ADDRESS 012)	0.
AIRFRAME DELETE (ADDRESS 016)	0.
ENGINE DELETE (ADDRESS 017)	0.
AVIONICS DELETE (ADDRESS 018)	0.
CLEAR (ADDRESS 027)	0.

First page of output (input data listing)

SYSTEM A
RUN 1

AIRFRAME RESEARCH AND DEVELOPMENT COSTS (INCLUDING PROFIT)
(MILLIONS OF DOLLARS)

INITIAL ENGINEERING	INITIAL TOOLING	DEVELOPMENT SUPPORT	TEST AIRCRAFT	FLIGHT TEST	ENGINES FOR TEST A/C
61.006	31.631	77.689	60.165	9.990	7.428

AVIONICS FOR TEST A/C	TOTAL R AND D
0.735	248.643

TEST AIRCRAFT PRODUCTION COSTS (INCLUDING PROFIT)
(MILLIONS OF DOLLARS)

NUMBER OF TEST A/C	SUSTAINING ENGINEERING	SUSTAINING TOOLING	LABOR	MATERIAL	TOTAL PER AIRFRAME
3	4.397	1.726	12.298	1.034	20.055

Second page of output (airframe RDT&E costs)

ORIGINAL PAGE IS
OF POOR QUALITY

AIRFRAME PRODUCTION COSTS (INCLUDING PROFIT)
(MILLIONS OF DOLLARS)

-- COST OF TEST AIRCRAFT EXCLUDED --

	UNIT COSTS					CUMULATIVE AVERAGE COSTS				
QUANTITY	SUSTAINING ENGINEERING	SUSTAINING TOOLING	LABOR	MATERIAL	TOTAL	SUSTAINING ENGINEERING	SUSTAINING TOOLING	LABOR	MATERIAL	TOTAL
1	4.501	1.491	8.584	0.899	15.475	4.501	1.491	8.584	0.899	15.475
2	3.674	1.198	7.825	0.866	13.562	4.087	1.344	8.215	0.883	14.529
3	3.126	1.006	7.255	0.840	12.227	3.767	1.232	7.896	0.869	13.764
4	2.733	0.871	6.805	0.818	11.228	3.508	1.141	7.624	0.856	13.130
5	2.437	0.769	6.438	0.800	10.445	3.294	1.067	7.388	0.845	12.594
6	2.204	0.691	6.131	0.785	9.810	3.112	1.004	7.178	0.835	12.130
7	2.016	0.627	5.869	0.771	9.283	2.956	0.950	6.992	0.826	11.724
8	1.861	2.287	5.641	0.759	10.547	2.819	1.117	6.823	0.817	11.577
9	1.730	5.242	5.441	0.748	13.160	2.698	1.576	6.669	0.810	11.753
10	1.618	3.395	5.263	0.738	11.014	2.590	1.758	6.529	0.802	11.679
20	1.011	2.980	4.154	0.670	8.815	1.911	2.569	5.554	0.750	10.784
30	0.753	2.349	3.576	0.631	7.309	1.559	2.577	4.973	0.716	9.825
40	0.608	0.975	3.204	0.603	5.390	1.336	2.350	4.570	0.690	8.946
50	0.513	0.361	2.937	0.583	4.394	1.179	2.012	4.266	0.671	8.128
60	0.446	0.310	2.734	0.566	4.057	1.062	1.732	4.025	0.654	7.474
70	0.396	0.273	2.572	0.552	3.794	0.970	1.526	3.828	0.641	6.964
80	0.357	0.244	2.439	0.540	3.581	0.895	1.367	3.661	0.629	6.553
90	0.326	0.221	2.326	0.530	3.404	0.834	1.241	3.518	0.618	6.212
100	0.300	0.203	2.230	0.521	3.254	0.782	1.138	3.394	0.609	5.922
200	0.174	0.113	1.682	0.465	2.434	0.503	0.643	2.652	0.549	4.347
300	0.126	0.080	1.425	0.435	2.066	0.384	0.460	2.281	0.515	3.641
400	0.101	0.062	1.266	0.414	1.843	0.316	0.363	2.046	0.493	3.217
500	0.084	0.051	1.154	0.399	1.689	0.271	0.301	1.878	0.475	2.926
600	0.073	0.044	1.071	0.387	1.575	0.239	0.259	1.750	0.462	2.710
700	0.064	0.039	1.005	0.377	1.485	0.215	0.228	1.648	0.450	2.541
800	0.058	0.034	0.951	0.369	1.412	0.196	0.204	1.564	0.441	2.404
900	0.053	0.031	0.906	0.362	1.351	0.180	0.185	1.493	0.432	2.290
1000	0.048	0.028	0.867	0.355	1.299	0.167	0.169	1.433	0.425	2.194
2000	0.028	0.016	0.651	0.316	1.011	0.102	0.095	1.087	0.379	1.663
3000	0.020	0.011	0.550	0.296	0.877	0.076	0.068	0.923	0.354	1.421
4000	0.016	0.009	0.488	0.282	0.794	0.061	0.053	0.822	0.338	1.274
5000	0.013	0.007	0.445	0.271	0.737	0.052	0.044	0.750	0.326	1.172
6000	0.012	0.006	0.413	0.263	0.693	0.045	0.038	0.697	0.316	1.096
7000	0.010	0.005	0.387	0.256	0.659	0.040	0.033	0.654	0.308	1.036
8000	0.009	0.005	0.366	0.251	0.631	0.037	0.030	0.619	0.301	0.987
9000	0.008	0.004	0.349	0.246	0.607	0.033	0.027	0.590	0.295	0.946
10000	0.008	0.004	0.334	0.242	0.587	0.031	0.025	0.565	0.290	0.911
20000	0.004	0.002	0.250	0.215	0.472	0.018	0.014	0.425	0.258	0.716

Third page of output (airframe production costs)

ORIGINAL PAGE IS
OF POOR QUALITY

ENGINE RESEARCH AND DEVELOPMENT COST (INCLUDING PROFIT)
(MILLIONS OF DOLLARS)

307.988

ENGINE PRODUCTION COSTS (INCLUDING PROFIT)
(MILLIONS OF DOLLARS)

-- COST OF ENGINES FOR TEST AIRCRAFT EXCLUDED --

ENGINE QUANTITY	UNIT COSTS (ENGINES)	CUM. AVG. COSTS (ENGINES)	AIRCRAFT QUANTITY	TOTAL UNIT COSTS (AIRCRAFT)	CUM. AVG. COSTS (AIRCRAFT)
4	0.520	0.528	1	2.110	2.110
8	0.505	0.519	2	2.047	2.076
12	0.493	0.512	3	1.989	2.047
16	0.483	0.505	4	1.945	2.021
20	0.474	0.500	5	1.909	1.997
24	0.467	0.495	6	1.878	1.977
28	0.460	0.490	7	1.851	1.961
32	0.455	0.486	8	1.827	1.944
36	0.449	0.482	9	1.805	1.929
40	0.445	0.479	10	1.785	1.914
80	0.412	0.453	20	1.653	1.810
120	0.393	0.436	30	1.576	1.743
160	0.380	0.423	40	1.521	1.691
200	0.370	0.414	50	1.480	1.654
240	0.361	0.405	60	1.447	1.622
280	0.354	0.399	70	1.417	1.595
320	0.349	0.393	80	1.395	1.571
360	0.343	0.388	90	1.374	1.550
400	0.339	0.383	100	1.356	1.532
800	0.310	0.353	200	1.240	1.411
1200	0.294	0.336	300	1.177	1.341
1600	0.283	0.324	400	1.134	1.299
2000	0.275	0.315	500	1.101	1.260
2400	0.269	0.308	600	1.076	1.231
2800	0.264	0.302	700	1.054	1.207
3200	0.259	0.297	800	1.036	1.187
3600	0.255	0.292	900	1.020	1.169
4000	0.252	0.288	1000	1.004	1.154
8000	0.230	[illegible]	2000	0.919	1.044
12000	0.218	0.250	3000	0.872	1.001
16000	0.210	0.241	4000	0.840	0.969
20000	0.204	0.234	5000	0.815	0.937
24000	0.199	0.229	6000	0.796	0.915
28000	0.195	0.224	7000	0.780	0.897
32000	0.192	0.220	8000	0.767	0.882
36000	0.189	0.217	9000	0.755	0.869
40000	0.187	0.214	10000	0.745	0.856
80000	0.171	0.196	20000	0.679	0.782

Fourth Page of Output (engine RDT&E and Production Costs)

AVIONICS RESEARCH AND DEVELOPMENT COST (INCLUDING PROFIT)
(MILLIONS OF DOLLARS)

5.000

AVIONICS PRODUCTION COSTS (INCLUDING PROFIT)
(MILLIONS OF DOLLARS)

-- COST OF AVIONICS FOR TEST AIRCRAFT EXCLUDED --

QUANTITY	UNIT	CUMULATIVE AVERAGE
1	0.194	0.194
2	0.185	0.190
3	0.179	0.186
4	0.173	0.183
5	0.169	0.180
6	0.165	0.178
7	0.162	0.175
8	0.159	0.173
9	0.156	0.171
10	0.154	0.170
20	0.138	0.157
30	0.129	0.149
40	0.123	0.143
50	0.118	0.138
60	0.114	0.134
70	0.111	0.131
80	0.108	0.129
90	0.106	0.126
100	0.104	0.124
200	0.092	0.111
300	0.085	0.103
400	0.081	0.098
500	0.078	0.094
600	0.075	0.091
700	0.073	0.089
800	0.071	0.087
900	0.070	0.085
1000	0.068	0.083
2000	0.060	0.074
3000	0.054	0.068
4000	0.053	0.063
5000	0.051	[illegible]
6000	0.049	[illegible]
7000	0.048	0.059
8000	0.047	0.057
9000	0.046	0.056
10000	0.045	0.055
20000	0.039	0.048

Fifth page of output (avionics RDT&E and production costs)

COMBINED COSTS (INCLUDING PROFIT) FOR AIRFRAMES, ENGINES, AND AVIONICS
(MILLIONS OF DOLLARS)

AIRFRAME RDTE INCL. TEST A/C WITH ENGINES AND AVIONICS	ENGINE RDTE	AVIONICS RDTE	TOTAL RDTE COST
248.643	307.988	5.000	561.631

AIRCRAFT QUANTITY	PROD. RATE (AIRFRAMES PER MONTH)	. . . . PRODUCTION COSTS WITHOUT RDTE			. PRODUCTION COSTS WITH RDTE .	
		UNIT	CUM. AVG.	TOTAL	CUM. AVG.	GRAND TOTAL
1	1.0	17.779	17.779	17.779	579.411	579.411
2	1.0	15.790	16.795	33.590	297.611	595.222
3	1.0	14.394	15.997	47.990	203.207	609.621
4	1.0	13.346	15.335	61.339	155.743	622.971
5	1.0	12.523	14.773	73.865	127.099	635.496
6	1.0	11.853	14.287	85.720	107.892	647.351
7	1.0	11.296	13.860	97.017	94.093	658.648
8	1.1	12.533	13.694	109.551	83.898	671.182
9	1.4	15.122	13.853	124.674	76.256	686.305
10	1.6	12.953	13.763	137.628	69.926	699.259
20	4.4	10.607	12.751	255.013	40.832	816.644
30	7.4	9.013	11.716	351.489	30.437	913.120
40	9.5	7.034	10.782	431.291	24.823	992.923
50	10.0	5.992	9.921	496.039	21.153	1057.670
60	10.0	5.618	9.230	553.818	18.591	1115.449
70	10.0	5.323	8.690	608.322	16.714	1169.953
80	10.0	5.084	8.253	660.201	15.273	1221.832
90	10.0	4.884	7.888	709.914	14.128	1271.546
100	10.0	4.714	7.578	757.798	13.194	1319.429
200	10.0	3.767	5.868	1173.609	8.676	1735.240
300	10.0	3.328	5.087	1526.022	6.959	2087.653
400	10.0	3.058	4.611	1844.258	6.015	2405.889
500	10.0	2.868	4.280	2139.980	5.403	2701.612
600	10.0	2.726	4.032	2419.306	4.968	2980.938
700	10.0	2.612	3.837	2685.938	4.639	3247.569
800	10.0	2.519	3.678	2942.327	4.380	3503.958
900	10.0	2.441	3.545	3190.202	4.169	3751.833
1000	10.0	2.374	3.431	3430.838	3.992	3992.469
2000	10.0	1.990	2.791	5582.867	3.072	6144.479
3000	10.0	1.805	2.491	7471.641	2.678	8033.272
4000	10.0	1.687	2.303	9213.595	2.444	9775.226
5000	10.0	1.603	2.171	10856.586	2.284	11418.217
6000	10.0	1.538	2.071	12426.241	2.165	12987.873
7000	10.0	1.487	1.991	13938.331	2.071	14479.962
8000	10.0	1.444	1.925	15403.435	1.996	15965.067
9000	10.0	1.408	1.870	16829.097	1.932	17390.728
10000	10.0	1.377	1.822	18220.935	1.878	18782.566
20000	10.0	1.191	1.546	30924.960	1.574	31486.591

Sixth Page of Output (Total Aircraft RDT&E and Production Costs)

References:

1. Boren, H. E., Jr., DAPCA: A Computer Program for Determining Aircraft Development and Production Costs, Rand Corporation Memorandum RM-5221-PR, February 1967.

2. Watts, A. Frank, Aircraft Turbine Engines--Development and Procurement Cost, The Rand Corporation, RM-4670-PR (abridged), November 1965.

3. Teng, C., An Estimating Relationship for Fighter/Interceptor Avionic System Procurement Cost, The Rand Corporation, RM-5841-PR (abridged) May 1966.

SECTION 10

OPTIMIZATION TECHNIQUES

Two program modules are available for optimization studies:

1. The variational optimization program option ATOP II, References 1 and 2. This option can be applied to any system of nonlinear ordinary differential equations by a slight program modification.

2. The multivariable search method contained in program AESOP, References 3, 4, and 5.

This program contains thirteen algorithms for solving nonlinear finite dimensioned optimization problems. AESOP is available as a separate program module in the ODIN/RLV and also as an integral part of the ATOP II program of Section 7.3.

REFERENCES:

1. Hague, D. S. and Glatt, C. R., Optimal Design Integration for Military Flight Vehicles, ODIN/MFV, Chapter 7.3, AFFDL-TR-72-132, December 1972.

2. Mobley, R. L. and Vorwald, R. R., Three Degree of Freedom Problem Optimization Formulation, Part 2, Volume III-User's Manual, FDL-TDR-64-1, McDonnell-Douglas Corporation, October 1964.

3. Hague, D. S. and Glatt, C. R., An Introduction to Multivariable Search Techniques for Parameter Optimization (and Program AESOP), NASA CR-73200, April 1968.

. Hague, D. S. and Glatt, C.R., Application of Multivariable Search Techniques to the Optimal Design of Hypersonic Cruise Vehicles, NASA CR-73202, April 1968.

5. Hague, D. S. and Glatt, C. R., A Guide to the Automated Engineering and Scientific Optimization Program, AESOP, NASA CR-73201, April 1968.

TABLE OF CONTENTS FOR SECTION 10.1,

THE VARIATIONAL STEEPEST DESCENT METHOD

10.1 THE VARIATIONAL STEEPEST DESCENT METHOD

10.1.1 The Problem Statement

Point mass motion is governed by three second order differential equations of position together with a first order differential equation governing the mass. By suitably defining additional state variables, it is possible to reduce these equations to a set of first order differential equations. Point mass motion is, therefore, governed by a set of first order differential equations. The form of these equations is

$$\left\{\dot{x}_n(t)\right\} = \left\{f\left(x_n(t),\ \alpha_m(t),\ t\right)\right\}$$

$$n = 1,2 \ldots\ldots\ldots N$$
$$m = 1,2 \ldots\ldots\ldots M \qquad (1)$$

That is, there are N state variables whose derivatives $\dot{x}_n(t)$ are defined by N first order differential equations involving the state variables, together with M control variables, $\alpha_m(t)$, and t, the independent variable itself.

Constraints may be imposed on a set of functions of the state variables and time at the end of the trajectory. In this case, a set of constraint functions of the form

$$\left\{\psi_p\right\} = \left\{\psi_p\left(x_n(T),\ T\right)\right\} = 0$$
$$p = 1,2 \ldots\ldots\ldots P \qquad (2)$$

can be constructed which the final trajectory must satisfy. Any one of the constraints may be used as a cut-off function which, when satisfied, will terminate a particular trajectory. The cut-off function can, therefore, be written in the form

$$\Omega = \Omega\left(x_n(T),\ T\right) = 0 \qquad (3)$$

and determines the trajectory termination time T. In all, then, when the cut-off function is included, there are (P + 1) end constraints.

Finally, it may be that some other function of the state variables and time at the end of the trajectory is to be optimized. Hence, a pay-off function

$$\phi = \phi\left(x_n(T),\ T\right) \qquad (4)$$

which is to be maximized or minimized, can be constructed.

Now, suppose that a nominal trajectory is available. The requirements of this trajectory are modest; it must satisfy the cut-off condition, Equation (3), but it need not optimize the pay-off function or satisfy the constraint equations. To generate this nominal trajectory by integrating Equations (1), the vehicle characteristics, the initial state variable values, and a nominal control variable history must be known. Once this nominal trajectory is available, the steepest descent process can be applied. To do this, the trajectory showing the greatest improvement in the pay-off function, while at the same time eliminating a given amount of the end point errors as measured by Equations (2) for a given size of control variable perturbation, is obtained by application of the Variational Calculus.

Equations (2) provide an end point error measure, for they will only be satisfied if the end points have been achieved. Therefore, any non-zero ψ_p represents an end point error which must be corrected. A convenient measure of the control variable perturbation can be defined by the scalar quantity,

$$DP^2 = \int_{t_o}^{T} \left\lfloor \delta\alpha(t) \right\rfloor \left[W(t) \right] \left\{ \delta\alpha(t) \right\} dt \qquad (5)$$

where W is any arbitrary symmetric matrix. In the case where all control variables have a similar ability to affect the trajectory, W is taken equal to the unit matrix, and DP^2 becomes the integrated square of the control variable perturbations $\delta\alpha(t)$. It might be noted that if Equation 5 is to have meaning, it is essential that all control variables have the same dimensions. To meet this condition, the control variables can be expressed in non-dimensional form.

The constraint on control variable perturbation size represented by Equation (5) is an essential element of the steepest descent process; for the optimum perturbation will be found by local linearization of the non-linear trajectory equations about the nominal path. To insure validity of the linearized approximation, the analysis must be limited to small control variable perturbations by means of Equation (5) which provides an integral measure of the local perturbation magnitudes.

10.1.2 Single Stage Analysis

The steepest descent process has been outlined above. To implement this method, an analysis of all perturbations about the nominal trajectory must be undertaken. In the present report, all perturbations will be linearized; only first order perturbations in the control and state variables will be considered. The objective of the linearized analysis is determination of the optimum control variable perturbation in the sense discussed in the previous section.

Denoting variables on the nominal trajectory by a bar

$$\left\{\alpha_m(t)\right\}\text{nominal} = \left\{\bar{\alpha}_m(t)\right\} \tag{6}$$

and

$$\left\{x_n(t)\right\}\text{nominal} = \left\{\bar{x}_n(t)\right\} \tag{7}$$

where there are M control variables and N state variables.

Now consider a small perturbation to the control variable history, $\delta\alpha(t)$; this in turn will cause a small perturbation in the state variable history, $\delta x(t)$. The new values of the variables become

$$\left\{\alpha(t)\right\} = \left\{\bar{\alpha}(t)\right\} + \left\{\delta\alpha(t)\right\} \tag{8}$$

and

$$\left\{x(t)\right\} = \left\{\bar{x}(t)\right\} + \left\{\delta x(t)\right\} \tag{9}$$

The nominal state variable and perturbed state variable histories can also be written as

$$\left\{\bar{x}(t)\right\} = \left\{x(t_o)\right\} + \int_{t_o}^{t} \left\{f\left(\bar{x}\,(t),\ \bar{\alpha}\,(t),\ t\right)\right\} dt \tag{10}$$

$$\left\{x(t)\right\} = \left\{x(t_o)\right\} + \int_{t_o}^{t} \left\{f\left(\bar{x} + \delta x,\ \bar{\alpha} + \delta\alpha,\ t\right)\right\} dt \tag{11}$$

Subtracting Equation (10) from Equation (11) and using Taylor's expansion to first order,

$$\left\{x(t)\right\} - \left\{\bar{x}(t)\right\} = \int_{t_o}^{t} \left\{\frac{\partial \bar{f}}{\partial x_n} \cdot \delta x^n + \frac{\partial \bar{f}}{\partial \alpha_m} \cdot \delta\alpha^m\right\} dt = \left\{\delta x(t)\right\} \tag{12}$$

where

$$\bar{f} = f\left(\bar{x}(t),\ \bar{\alpha}\,(t),\ t\right) \tag{13}$$

and where the repeated index indicates a summation over all possible values. Differentiation leads to

$$\frac{d}{dt}\left\{\delta x(t)\right\} = \left\{\frac{\partial \bar{f}}{\partial x_n}\ \delta x^n + \frac{\partial \bar{f}}{\partial \alpha_m}\ \delta\alpha^m\right\} \tag{14a}$$

or in matrix form

$$\frac{d}{dt}\left\{\delta x(t)\right\} = \left[F\right]\left\{\delta x\right\} + \left[G\right]\left\{\delta\alpha\right\} \tag{14b}$$

where

$$F_{ij} = \frac{\partial \bar{f}_i}{\partial x_j} \text{ and } G_{ij} = \frac{\partial \bar{f}_i}{\partial \alpha_j} \tag{15}$$

Here the $(i,j)^{th}$ element lies in the i^{th} row and j^{th} column of the matrices; F is an N x N matrix and G is an N x M matrix.

The effect of these perturbations on pay-off, cut-off, and constraint functions must now be determined. A general method for obtaining these effects, known as the 'adjoint method,' Reference 13, is to define a new set of variables by the equations

$$\left[\dot{\lambda}(t)\right] = -\left[F(t)\right]'\left[\lambda(t)\right] \tag{16}$$

By specifying various boundary conditions on the λ, the changes in all functions of interest can be found in turn. To show this pre-multiply Equation (14) by λ' and Equation (16) by $\delta x'$, transpose the second of these equations and sum with the first giving

$$\left[\lambda\right]' \cdot \left\{\frac{d}{dt}(\delta x)\right\} + \left[\dot{\lambda}\right]'\left\{\delta x\right\} = \left[\lambda\right]'\left[F\right]\left\{\delta x\right\} + \left[\lambda\right]'\left[G\right]\left\{\delta\alpha\right\} - \left[\lambda\right]'\left[F\right]\left\{\delta x\right\} \tag{17}$$

which may be written as

$$\left\{\frac{d}{dt}(\lambda'\delta x)\right\} = \left[\lambda\right]'\left[G\right]\left\{\delta\alpha\right\} \tag{18}$$

Integrating Equation (18) over the trajectory

$$\left\{\lambda'\delta x\right\}_T - \left\{\lambda'\delta x\right\}_{t_o} = \int_{t_o}^{T}\left[\lambda\right]'\left[G\right]\left\{\delta\alpha\right\} dt \tag{19}$$

Now define three distinct sets of λ' functions by applying the following boundary conditions at t = T:

$$\left\{\lambda(T)\right\} = \left\{\frac{\partial\phi}{\partial x_i}\right\}_T = \left\{\lambda_\phi(T)\right\} \tag{20a}$$

$$\left\{\lambda(T)\right\} = \left\{\frac{\partial\Omega}{\partial x_i}\right\}_T = \left\{\lambda_\Omega(T)\right\} \tag{20b}$$

$$\left[\lambda(T)\right] = \left[\frac{\partial\psi_j}{\partial x_i}\right]_T = \left[\lambda_\psi(T)\right] \tag{20c}$$

Equation (16) may now be integrated in the reverse direction (i.e., from T to t_o) to obtain the functions, $\{\lambda_\phi(t)\}$, $\{\lambda_\Omega(t)\}$, and $\{\lambda_\psi(t)\}$.

Substituting each of these functions into Equation (19) in turn and noting that

$$\lfloor \lambda_\phi(T) \rfloor \{\delta x\} = \left\lfloor \frac{\partial \phi}{\partial x} \right\rfloor \{\delta x\} = \delta\phi_{t=T} \tag{21a}$$

$$\lfloor \lambda_\Omega(T) \rfloor \{\delta x\} = \left\lfloor \frac{\partial \Omega}{\partial x} \right\rfloor \{\delta x\} = \delta\Omega_{t=T} \tag{21b}$$

$$\left[\lambda_\psi(T) \right]' \{\delta x\} = \left[\frac{\partial \psi_i}{\partial x_j} \right] \{\delta x\} = \{\delta\psi_{t=T}\} \tag{21c}$$

It follows that

$$\delta\phi_{t=T} = \int_{t_o}^{T} \lfloor \lambda_\phi \rfloor \left[G \right] \{\delta\alpha\} dt + \lfloor \lambda_\phi(t_o) \rfloor \{\delta x(t_o)\} \tag{22a}$$

$$\delta\Omega_{t=T} = \int_{t_o}^{T} \lfloor \lambda_\Omega \rfloor \left[G \right] \{\delta\alpha\} dt + \lfloor \lambda_\Omega(t_o) \rfloor \{\delta x(t_o)\} \tag{22b}$$

$$\{\delta\psi\}_{t=T} = \int_{t_o}^{T} \left[\lambda_\psi \right]' \left[G \right] \{\delta\alpha\} dt + \left[\lambda_\psi(t_o) \right]' \{\delta x(t_o)\} \tag{22c}$$

Now, Equations (22) give the changes in pay-off function, cut-off function and constraint functions at the terminal time of the nominal trajectory; however, on the perturbed trajectory, the cut-off will usually occur at some perturbed time, T + ΔT. In this case, the total change in the above quantities becomes

$$d\phi = \int_{t_o}^{T} \lfloor \lambda_\phi \rfloor \left[G \right] \{\delta\alpha\} dt + \lfloor \lambda_\phi(t_o) \rfloor \{\delta x(t_o)\} + \dot{\phi}(T)\,\Delta T \tag{23a}$$

$$d\Omega = \int_{t_o}^{T} \lfloor \lambda_\Omega \rfloor \left[G \right] \{\delta\alpha\} dt + \lfloor \lambda_\Omega(t_o) \rfloor \{\delta x(t_o)\} + \dot{\Omega}(T)\,\Delta T \tag{23b}$$

$$\{d\psi\} = \int_{t_o}^{T} \left[\lambda_\psi \right]' \left[G \right] \{\delta\alpha\} dt + \left[\lambda_\psi(t_o) \right]' \{\delta x(t_o)\} + \{\dot{\psi}(T)\}\Delta T \tag{23c}$$

Equations (23) supply the change in pay-off, cut-off, and constraint functions on the perturbed trajectory.

The time perturbation in Equations (23a) and (23c) may be eliminated by noting that, by definition of the cut-off function, Equation (23b) must be zero.

$$\therefore \Delta T = -\frac{1}{\dot{\Omega}(T)}\left(\int_{t_o}^{T}\left\lfloor\lambda_\Omega\right\rfloor\left[G\right]\left\{\delta\alpha\right\}dt + \left\lfloor\lambda_\Omega(t_o)\right\rfloor\left\{\delta x(t_o)\right\}\right) \tag{24}$$

Substituting Equation (24) into Equations (23a) and (23c)

$$d\phi = \int_{t_o}^{T}\left\lfloor\lambda_{\phi\Omega}\right\rfloor\left[G\right]\left\{\delta\alpha\right\}dt + \left\lfloor\lambda_{\phi\Omega}(t_o)\right\rfloor\left\{\delta x(t_o)\right\} \tag{25a}$$

$$\left\{d\psi\right\} = \int_{t_o}^{T}\left[\lambda_{\psi\Omega}\right]'\left[G\right]\left\{\delta\alpha\right\}dt + \left[\lambda_{\psi\Omega}(t_o)\right]'\left\{\delta x(t_o)\right\} \tag{25b}$$

where

$$\left\{\lambda_{\phi\Omega}\right\} = \left\{\lambda_\phi\right\} - \frac{\dot{\phi}(T)}{\dot{\Omega}(T)}\left\{\lambda_\Omega\right\} \tag{26a}$$

$$\left[\lambda_{\psi\Omega}\right]' = \left[\lambda_\psi\right]' - \frac{\left\{\dot{\psi}(T)\right\}\left\lfloor\lambda_\Omega\right\rfloor}{\dot{\Omega}(T)} \tag{26b}$$

Equations (25) reveal the significance of the λ functions, originally defined by Equations (16) and (20). At time t_o, $\lambda_{\phi\Omega}$ gives the sensitivity of $\phi(T)$ to small perturbations in the state variables at t_o. Similarly, $\lambda_{\phi\Omega}(t)$ measures the sensitivity of $\phi(T)$ to small perturbations in the state variables at any time t. The sensitivity of the constraints $d\psi$ to small state variable perturbations at any time is likewise defined by each row of the function $\lambda_{\psi\Omega}(t)$.

A measure of the sensitivity of a trajectory to control variable perturbations can be obtained from the quantities $\lambda_\phi{}'_\Omega G$ and $\lambda_\psi{}'_\Omega G$. Consider a pulse control variable perturbation at time t', that is, $\delta(t-t')$, where δ is the Dirac delta function. With this type of control variable perturbation, it can be seen from Equations (25) that the changes in pay-off and constraint functions will be $\lambda_{\phi\Omega}(t')'G(t')$ and $\lambda_{\psi\Omega}(t')'G(t')$, respectively, for fixed initial conditions.

In order to apply the steepest-descent process, the performance function change, Equation (20a), must be maximized; subject to specified changes in the constraints, Equation (25b); and a given size perturbation to the control variables, Equation (5). This can be achieved by constructing an augmented function in the manner of Lagrange which is to be maximized instead of $d\phi$. For the present problem, the augmented function is

$$U = \int_{t_o}^{T} \lfloor \lambda\phi\Omega \rfloor [G] \{\delta\alpha\} dt + \lfloor \lambda\phi\Omega(t_o) \rfloor \{\delta x(t_o)\}$$

$$+ \lfloor \nu \rfloor \left\{ \int_{t_o}^{T} [\lambda\psi\Omega]' [G] \{\delta\alpha\} dt + [\lambda\psi\Omega(t_o)]' \{\delta x(t_o)\} \right\}$$

$$+\mu \int_{t_o}^{T} \lfloor \delta\alpha \rfloor [W] \{\delta\alpha\} dt \tag{27}$$

where the ν are P undetermined Lagrangian multipliers, and μ is a single undetermined Lagrangian multiplier. The objective now is to find that variation of the control variable history which will maximize U.

Consider a variation of $\delta\alpha$, that is a $\delta(\delta\alpha)$. Then, it is always possible to write any $\delta\alpha$ distribution in the form

$$\{\delta\alpha\} = \{A(t)\} k, \text{ or } \lfloor \delta\alpha \rfloor = \lfloor A(t) \rfloor k \tag{28}$$

where A(t) prescribes the perturbation shape;and k, its magnitude. Now that part of Equation (27) which depends on $\delta\alpha$, the perturbation in the control variable, can be written in the form

$$\overline{U} = k \int_{t_o}^{T} \lfloor \lambda\phi\Omega \rfloor [G] \{A(t)\} dt + k \lfloor \nu \rfloor \int_{t_o}^{T} [\lambda\psi\Omega]' [G] \{A(t)\} dt$$

$$+ k^2 \mu \int_{t_o}^{T} \lfloor A(t) \rfloor [W] \{A(t)\} dt \tag{29}$$

So that

$$\frac{\partial\overline{U}}{\partial K} = \int_{t_o}^{T} \lfloor \lambda\phi\Omega \rfloor [G] \{A(t)\} dt + \lfloor \nu \rfloor \int_{t_o}^{T} [\lambda\psi\Omega]' [G] \{A(t)\} dt$$

$$+ 2k\mu \int_{t_o}^{T} \lfloor A(t) \rfloor [W] \{A(t)\} dt \tag{30}$$

or

$$\delta\bar{U} = \int_{t_o}^{T}\left(\lfloor\lambda_{\phi\Omega}\rfloor\left[G\right]\left\{\delta k \cdot A(t)\right\} + \lfloor\nu\rfloor\left[\lambda_{\psi\Omega}\right]'\left[G\right]\left\{\delta k \cdot A(t)\right\} + 2\mu\lfloor k \cdot A(t)\rfloor\left[W\right]\left\{\delta k \cdot A(t)\right\}\right) dt$$

$$= \int_{t_o}^{T}\left[\lfloor\lambda_{\phi\Omega}\rfloor\left[G\right] + \lfloor\nu\rfloor\left[\lambda_{\psi\Omega}\right]'\left[G\right] + 2\mu\lfloor\delta\alpha\rfloor\left[W\right]\right]\left\{\delta(\delta\alpha)\right\} dt \tag{31}$$

where it has been noted from Equation (28) that

$$\delta(\delta\alpha) = A(t)\ \delta k \tag{32}$$

Now, since Equation (31) holds for any $A(t)$, it follows that it is a general relationship. Further, for $\bar{U}$ to be an extremal, $\delta\bar{U}$ must be zero.

If $\bar{U}$ has been maximized by means of a control variable perturbation $\delta\alpha$, $\delta\bar{U}$ must be stationary for all small perturbations to the $\delta\alpha$, that is, for all $\delta(\delta\alpha)$. The only way in which Equation (31) can be zero for all $\delta(\delta\alpha)$ is for the coefficient of $\delta(\delta\alpha)$ to be identically zero. That this last statement is true follows from considering the case where, over some finite time interval between t_o and T, the coefficient of $\delta(\delta\alpha)$ is, say, positive. If this were the case, we could choose a $\delta(\delta\alpha)$ distribution that was also positive in this same interval and zero elsewhere between t_o and T. It would follow that $\bar{U}$ was also positive, and, hence, $\bar{U}$ could not be maximum. A similar argument holds when $\delta(\delta\alpha)$ is negative over any interval in t_o to T. Hence, the coefficient of $\delta(\delta\alpha)$ must be identically zero in the whole interval $t_o \leq t \leq T$. This argument is essentially based on that presented by Goldstein, Reference 14. It follows that

$$\left[\lfloor\lambda_{\phi\Omega}\rfloor + \lfloor\nu\rfloor\left[\lambda_{\psi\Omega}\right]'\right]\left[G\right] = -2\mu\lfloor\delta\alpha\rfloor\left[W\right] \tag{33}$$

Transposing, noting that W is symmetric, and solving for $\delta\alpha$,

$$\left\{\delta\alpha\right\} = -\frac{1}{2\mu}\left[W\right]^{-1}\left[G\right]'\left\{\left\{\lambda_{\phi\Omega}\right\}+\left[\lambda_{\psi\Omega}\right]\left\{\nu\right\}\right\} \tag{34}$$

Substituting Equation (34) into Equation (25b)

$$\left\{d\beta\right\} = -\frac{1}{2\mu}\left\{\left\{I_{\psi\phi}\right\}+\left[I_{\psi\psi}\right]\left\{\nu\right\}\right\} \tag{35a}$$

where

$$\left\{d\beta\right\} = \left\{d\psi\right\} - \left[\lambda_{\psi\Omega}(t_o)\right]'\left\{\delta x(t_o)\right\} \tag{35b}$$

and

$$\left[I_{\psi\psi}\right] = \int_{t_o}^{T}\left[\lambda_{\psi\Omega}\right]'\left[G\right]\left[W\right]^{-1}\left[G\right]'\left[\lambda_{\psi\Omega}\right]dt \tag{36a}$$

$$\left\{I_{\psi\phi}\right\} = \int_{t_o}^{T}\left[\lambda_{\psi\Omega}\right]'\left[G\right]\left[W\right]^{-1}\left[G\right]'\left\{\lambda_{\phi\Omega}\right\}dt \tag{36b}$$

For subsequent use define the integral

$$I_{\phi\phi} = \int_{t_o}^{T}\left\lfloor\lambda_{\phi\Omega}\right\rfloor\left[G\right]\left[W\right]^{-1}\left[G\right]'\left\{\lambda_{\phi\Omega}\right\}dt \tag{36c}$$

The multipliers ν can be expressed in terms of the multipliers μ by Equation (35a)

$$\left\{\nu\right\} = -\left[I_{\psi\psi}\right]^{-1}\left\{2\mu\left\{d\beta\right\}+\left\{I_{\psi\phi}\right\}\right\} \tag{37}$$

Substituting Equation (34) into Equation (5)

$$DP^2 = \frac{1}{4\mu^2}\left(I_{\phi\phi} + \left\lfloor I_{\psi\phi}\right\rfloor\left\{\nu\right\} + \left\lfloor\nu\right\rfloor\left\{I_{\psi\phi}\right\} + \left\lfloor\nu\right\rfloor\left[I_{\psi\psi}\right]\left\{\nu\right\}\right) \tag{38}$$

Transposing the second term in the right hand side bracket

$$DP^2 = \frac{1}{4\mu^2}\left(I_{\phi\phi} + 2\left\lfloor\nu\right\rfloor\left\{I_{\psi\phi}\right\} + \left\lfloor\nu\right\rfloor\left[I_{\psi\psi}\right]\left\{\nu\right\}\right) \tag{39}$$

Substituting Equation (37) in Equation (39) and noting that $\left[I_{\psi\psi}\right]^{-1}$ is symmetrical gives

$$4\mu^2 DP^2 = I_{\phi\phi} - \lfloor I_{\psi\phi} \rfloor \left[I_{\psi\psi}\right]^{-1} \left\{I_{\psi\phi}\right\} + 4\mu^2 \lfloor d\beta \rfloor \left[I_{\psi\psi}\right]^{-1} \left\{d\beta\right\} \tag{40}$$

So that

$$2\mu = \pm\sqrt{\frac{I_{\phi\phi} - \lfloor I_{\psi\phi} \rfloor \left[I_{\psi\psi}\right]^{-1} \left\{I_{\psi\phi}\right\}}{DP^2 - \lfloor d\beta \rfloor \left[I_{\psi\psi}\right]^{-1} \left\{d\beta\right\}}} \tag{41}$$

Substituting Equation (41) into Equation (37), the remaining Lagrangian multipliers are obtained in the form

$$\left\{\nu\right\} = -\left[I_{\psi\psi}\right]^{-1} \left\{\left\{I_{\psi\phi}\right\} \pm \sqrt{\frac{I_{\phi\phi} - \lfloor I_{\psi\phi} \rfloor \left[I_{\psi\psi}\right]^{-1} \left\{I_{\psi\phi}\right\}}{DP^2 - \lfloor d\beta \rfloor \left[I_{\psi\psi}\right]^{-1} \left\{d\beta\right\}}} \left\{d\beta\right\}\right\} \tag{42}$$

The optimum control perturbation is found by substituting Equations (41) and (42) back into Equation (34) and is

$$\begin{aligned} \left\{\delta\alpha\right\} = \mp \left[W\right]^{-1} \left[G\right]' \left\{\left\{\lambda_{\phi\Omega}\right\} - \left[\lambda_{\psi\Omega}\right] \left[I_{\psi\psi}\right]^{-1} \left\{I_{\psi\phi}\right\}\right\} \\ \times \sqrt{\frac{DP^2 - \lfloor d\beta \rfloor \left[I_{\psi\psi}\right]^{-1} \left\{d\beta\right\}}{I_{\phi\phi} - \lfloor I_{\psi\phi} \rfloor \left[I_{\psi\psi}\right]^{-1} \left\{I_{\psi\phi}\right\}}} \\ + \left[W\right]^{-1} \left[G\right]' \left[\lambda_{\psi\Omega}\right] \left[I_{\psi\psi}\right]^{-1} \left\{d\beta\right\} \end{aligned} \tag{43}$$

With this equation the steepest-descent control perturbation has been determined. Perturbing the control variables according to Equation (43) gives the optimum change in the trajectory as discussed in the section entitled, "Problem Statement," with the added effect of changes in the initial value of the state variables included through the term in $d\beta$. The appropriate sign to use on the first term of equation (43) can be determined by evaluating $d\phi$. Substituting the optimum control perturbation into Equation (25a) results in the equation shown on the following page.

$$d\phi = \mp\sqrt{\left(I_{\phi\phi} - \left\lfloor I_{\psi\phi}\right\rfloor \left[I_{\psi\psi}\right]^{-1}\left\{I_{\psi\phi}\right\}\right)\left(DP^2 - \left\lfloor d\beta\right\rfloor\left[I_{\psi\psi}\right]^{-1}\left\{d\beta\right\}\right)}$$

$$+ \left\lfloor I_{\psi\phi}\right\rfloor\left[I_{\psi\psi}\right]^{-1}\left\{d\beta\right\} + \left\lfloor \lambda_{\phi\Omega}(t_o)\right\rfloor \left\{\delta x(t_o)\right\} \tag{44}$$

As the quantity in the radical must be positive to assure the change in ϕ is real, it follows that the negative sign must be taken when minimizing the payoff function and the positive sign when maximizing the payoff function.

10.1.3 Combining Continuous Control and Finite Parameter Optimization

Many vehicle flight path optimization problems involve continuous control and finite parameter optimization. For example, with a multi-stage system the optimal control and stage points T_S may be required. These problems may be solved in an analogous manner to that employed for continuous control alone optimization. A combined perturbation stepsize parameter, DC^2, is defined by

$$DC^2 = \int_{t_o}^{T} \left\lfloor \delta\alpha(t)\right\rfloor \left[W(t)\right]\left\{\delta\alpha(t)\right\} dt + \sum_{\bar{S}=1}^{\bar{S}} V_{\bar{s}}\,\Delta T_{\bar{s}}^{2} \tag{45}$$

Equations which are analogous to those of (39) and (35a) are obtained

$$4\mu^2 . DC^2 - (J_{\phi\phi} + I_{\phi\phi}) - 2\left[\left\lfloor J_{\psi\phi}\right\rfloor + \left\lfloor L_{\psi\phi}\right\rfloor\right]\left\{\vartheta\right\}$$

$$-\left\lfloor\vartheta\right\rfloor\left[\left[J_{\psi\psi}\right] + \left[L_{\psi\psi}\right]\right]\left\{\vartheta\right\} = 0 \tag{46}$$

$$2\mu\left\{\delta\Gamma\right\} + \left\{\left\{J_{\psi\phi}\right\} + \left\{L_{\psi\phi}\right\}\right\} + \left[\left[J_{\psi\psi}\right] + \left[L_{\psi\psi}\right]\right]\left\{\vartheta\right\} = 0 \tag{47}$$

where the functions $J_{\phi\phi}$, $J_{\psi\phi}$, $J_{\psi\psi}$, $L_{\phi\phi}$, $L_{\psi\phi}$, $L_{\psi\psi}$, and $\delta\Gamma$ are defined in Reference 1. It can be shown, Reference 1, that the optimal stage point perturbations T_S are given by

$$\left\{\Delta T_{\bar{s}}\right\} = \mp\left[V_s\right]^{-1}\left[B_s^N\right]'\left\{\left\{\bar{\lambda}_{\phi\Omega N}\right\} - \left[\bar{\lambda}_{\psi\Omega N}\right]\left[J_{\psi\psi} + L_{\psi\psi}\right]^{-1}\left\{J_{\psi\phi} + L_{\psi\phi}\right\}\right\}$$

$$\times\sqrt{\frac{DC^2 - \left\lfloor\delta\Gamma\right\rfloor\left[J_{\psi\psi} + L_{\psi\psi}\right]^{-1}\left\{\delta\Gamma\right\}}{(J_{\phi\phi} + L_{\phi\phi}) - \left\lfloor J_{\psi\phi} + L_{\psi\phi}\right\rfloor\left[J_{\psi\psi} + L_{\psi\psi}\right]^{-1}\left\{J_{\psi\phi} + L_{\psi\phi}\right\}}} \tag{48}$$

$$+\left[V_s\right]^{-1}\left[B_s^N\right]'\left[\bar{\lambda}_{\psi\Omega N}\right]\left[J_{\psi\psi} + L_{\psi\psi}\right]^{-1}\left\{\delta\Gamma\right\}$$

with a similar expression for the optimal control perturbations. A more general formulation yet is provided by Petersen, Reference 12, where not only stage points but any finite set of parameters in a whole class can be incorporated into the variational steepest-descent formulation.

It is emphasized that Section 10.1 provides only an outline of the ATOP II variational formulation. Complete details including past applications can be obtained from References 1 to 15.

10.1.4 Some Past ATOP II Applications

The success of the variational steepest-descent method in solution of aircraft performance optimization problems is evident from the strong support given to this technique by a series of contracts let by leading Government research centers concerned with this area. The reason for this support is clear when performance gains obtained are examined. Figure 10.1-1 presents the 1962 time to climb record flights of the McDonnell F-4B aircraft. Figure 10.1-2 illustrates how closely these paths follow the minimum time ascent paths predicted by the References 1 and 2 program. Figure 10.1-3 provides a comparison between flight handbook performance estimates, a minimum time climb obtained by the References 1 and 2 program, and an attempt by Marine Col. Yunck to fly the predicted optimum.

The predicted optimal path and the path flown by Col. Yunck both produce a 23 per cent improvement in aircraft performance over the flight handbook. During the Cuban crisis of the early sixties results of this type were produced routinely from the References 1 and 2 program to aid in an Air Force readiness studies. It should be noted that unlike optimization studies in other technology fields, these performance gains are obtained without vehicle modification. To obtain these performance gains while retaining flight handbook methods would have required a 23 per cent increase in aircraft design capability, several years' effort and several billion dollars to replace an existing fleet of aircraft which could achieve this capability simply by being flown in the optimum manner. This one example serves as a lasting case of

1. the high cost associated with an over-simplified approach to performance optimization, and

2. the insignificant computational cost of adequate performance optimization studies for production aircraft when compared to the resulting payoff.

Further details of these F-4B performance optimization studies may be obtained from Reference 5.

In general, the variational steepest-descent method will usually converge quickly and reliably for short duration airbreathing trajectories, for booster ascent problems, and for orbital maneuver problems. Figures 10.1-4 to 10.1-6 illustrate the behavior of the method on several short duration airbreathing trajectory optimization problems. Each problem is solved from two nominal paths. In each case the two final optimal paths are in essential agreement. Figure 10.1-4(a) presents a maximum terminal velocity descent from 35,000 feet at 800 feet per second to 500 feet level flight in a fixed time of 75 seconds. Figure 10.1-4(b) is a maximum altitude in fixed time (75 seconds) path to a level flight condition. Figure 10.1-5 is a minimum time intercept from the same initial conditions. A target is coming in from 80 nautical miles at constant altitude and Mach number. In this example, the intercept range and time are not known prior to solution. Once a solution is obtained, the result can readily be verified; for it is the minimum time path to the then known interception point. In all examples the optimal paths obtained from each nominal are indistinguishable from each other. The corresponding control histories are also quite well defined, Figure 10.1-6. These examples are taken from those contained in Reference 15. The interception of Figure 10.1-5 is the simplest type of two-vehicle problem. For example, the target may accelerate as in the problem of Figure 10.1-7 which is taken from Reference 15. The figure again reveals no apparent difference between the optimal flight profiles obtained from each of the two nominal paths employed.

References:

1. Hague, D. S. and Glatt, C. R., Optimal Design Integration of Military Flight Vehicles, ODIN/MFV, Section 7.3, AFFDL-TR-72-132, 1972.

2. Mobley, R. L. and Vorwald, R. R., Three-Degree-of-Freedom Optimization Formulation, Part 2, Volume III, FDL-TDR-64-1, October 1964.

3. Brown, Robert C., Brulle, R. V., Combs, A. E., And Griffin, G. D., Six-Degree-of-Freedom Flight Path Study Generalized Computer Program, Part 1 Volume I, FDL-TDR-64-1, October 1964.

4. Seubert, F. W. and Usher, Newell E., Six-Degree-of-Freedom Flight Path Study Generalized Computer Program, Part 2, WADD Technical Report 6 -781, McDonnell-Douglas Corporation, May 1961.

5. Landgraf, S. K., Some Practical Applications of Performance Optimization Techniques to High Performance Aircraft, AIAA Paper 64-288, July 1964.

6. Hague, D. S., An Outline and Operating Instructions for the Steepest-Descent Trajectory Optimization Program--STOP, Aerodynamics Methods Note No. 1, McDonnell-Douglas Corporation, March 1963.

7. Hague, D. S., Geib, Ken, Ballew, L., and Witherspoon, J., Two-Vehicle Optimization--Theoretical Outline and Program Users Manual, Report B983, McDonnell-Douglas, 1965.

8. Hague, D. S., "The Optimization of Multiple-Arc Trajectories by the Steepest-Descent Method," Recent Advances in Optimization Techniques, edited by Lavi and Vogl, John Wiley and Sons, Inc., 1966, PP 489-517.

9. Hague, D. S. and Glatt, C. R., Study of Navigation and Guidance of Launch Vehicles Having Cruise Capability, Volume II, Boeing Document D2-113016-5, The Boeing Company, April 1967.

10. Retka, J., et al., Study of Navigation and Guidance of Launch Vehicles Having Cruise Capability, Volume IV, Boeing Document D2-113016-7, 1967.

11. Stein, L. H., Mathews, M. L., and Frenck, J. W., STOP: A Computer Program for Supersonic Transport Trajectory Optimization, NASA CR-792, May 1967.

12. Rozendaal, H. L., A General Branched Trajectory Optimization Algorithm with Applications to Space Shuttle Vehicle Mission Design, AAS/AAIA Astrodynamics Specialist Conference, August 17-19, 1971.

13. Hague, D. S., Atmospheric and Near Planet Trajectory Optimization by the Variational Steepest-Descent Method, NASA CR-73365, 1969.

14. Hague, D. S., Application of the Variational Steepest-Descent Method to High Performance Aircraft Trajectory Optimization, NASA CR-73366, 1969.

15. Hague, D. S., et al., Integration of Aerospace Vehicle Performance and Design Optimization, AIAA Paper No. 72-948, September 1972.

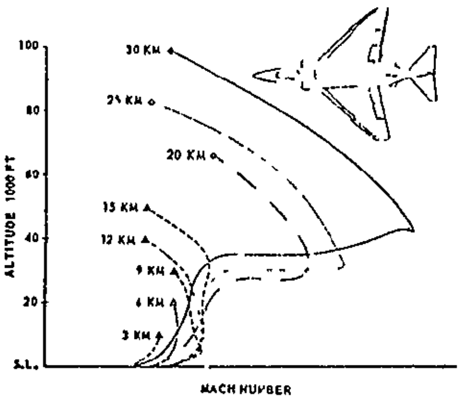


FIGURE 1. F-4B TIME-TO-CLIMB RECORD FLIGHTS

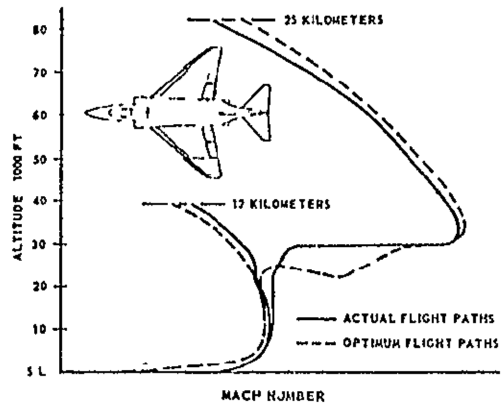


FIGURE 2. COMPARISON OF ACTUAL AND CALCULATED OPTIMUM FLIGHT PATHS FOR TWO F-4B TIME-TO-CLIMB RECORD FLIGHTS

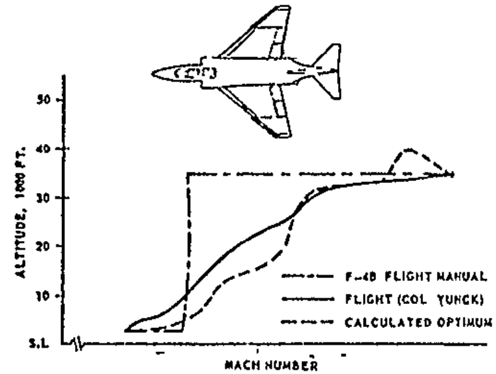


FIGURE 3. FLIGHT PATH COMPARISON

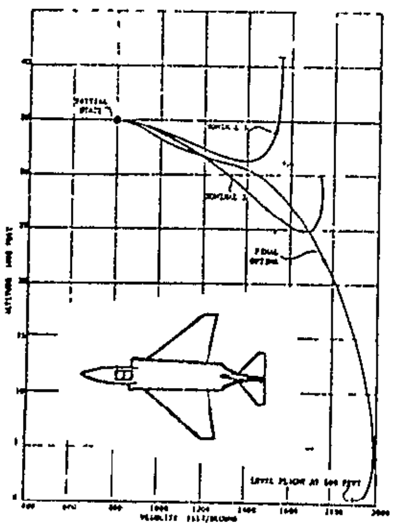


FIGURE 4(a). MAXIMUM VELOCITY DESCENT TO LEVEL FLIGHT

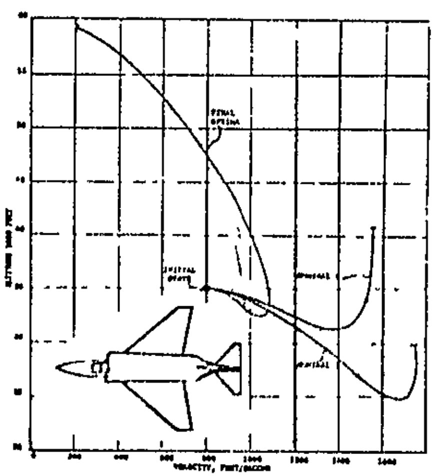


FIGURE 4(b). MAXIMUM LEVEL FLIGHT ALTITUDE IN FIXED TIME

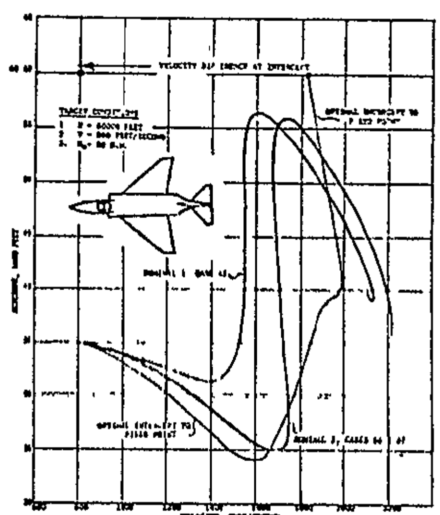


FIGURE 5. MINIMUM TIME (MAXIMUM RANGE) INTERCEPT

ORIGINAL PAGE IS OF POOR QUALITY

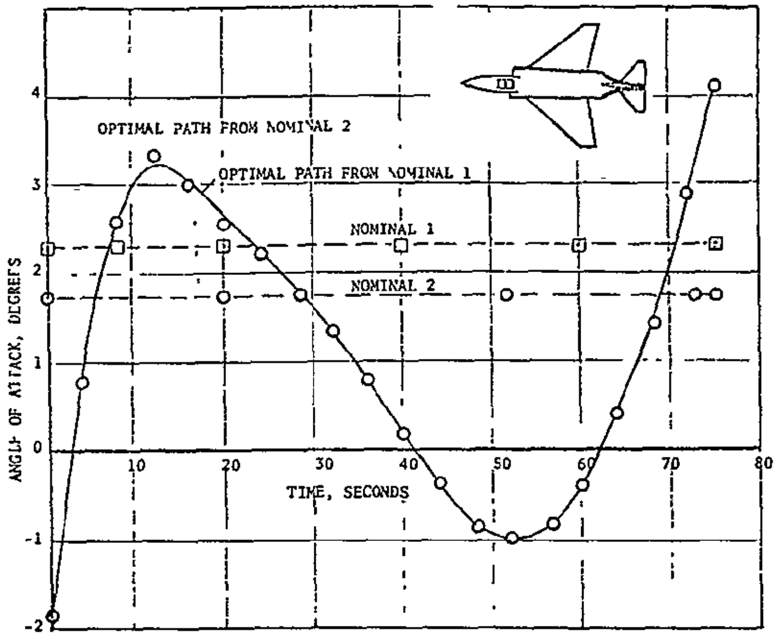


FIGURE 6 MAXIMUM VELOCITY DESCENT CONTROL HISTORY

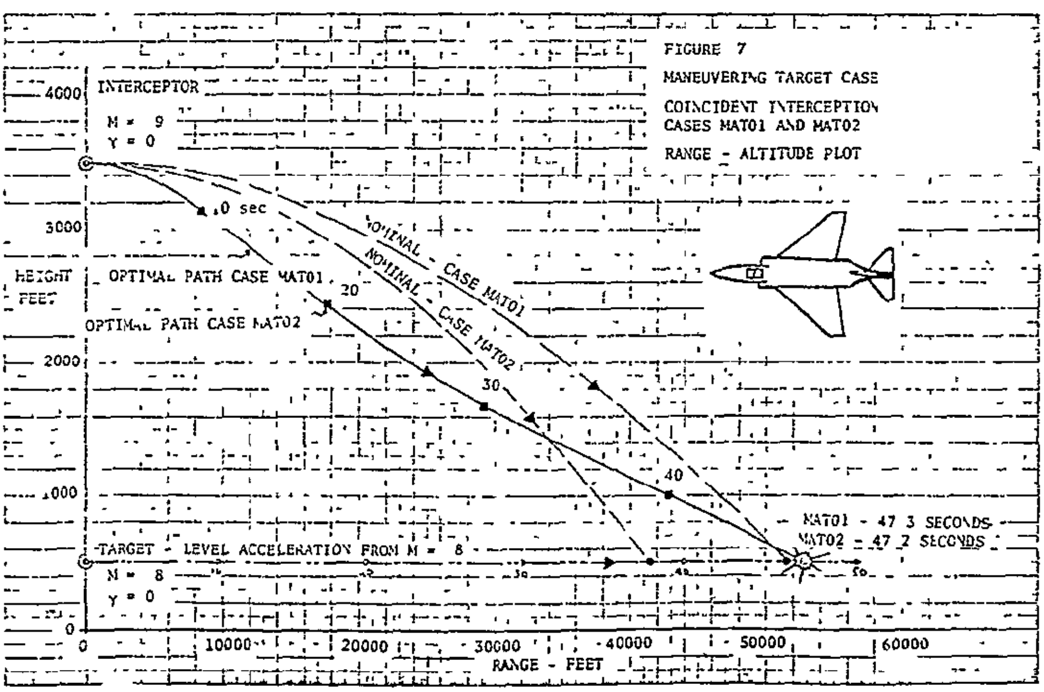


TABLE OF CONTENTS FOR SECTION 10.2, PROGRAM AESOP

ORIGINAL PAGE IS
OF POOR QUALITY

10.2 PROGRAM AESOP: COMPUTER PROGRAM FOR MULTIVARIABLE SEARCH

10.2.1 The Design Cycle

During the early fifties the moderately-sized digital computer began to appear in quantity at aerospace industrial establishments. The impact of these machines on the aerospace-vehicle design process has grown steadily since that time, and it is now commonplace to encounter a variety of large-scale computers in the CDC 6600, UNIVAC 1108, and IBM 360 series at large governmental and industrial aerospace concerns. Initially, the use of the digital computer was limited to a few relatively complex elements of the vehicle design process. Problems typified by the flutter speed calculation which requires computation of unsteady flow about an oscillating three-dimensional airfoil surface, the calculation of vibration modes of a three-dimensional elastic structure, and the solution of large-order matrix eigenvalue problems, taxed early machines to their full capacity. The structural dynamicist was rapidly joined by engineers in other disciplines with problems of comparable complexity, the aerodynamics specialist using more precise definition of the vehicle three-dimensional surface, the structures specialist employing extremely large matrices, the performance specialist employing the variational calculus, and so on. Business applications soon appeared in quantity, and in some establishments functions such as accounting, payroll, and inventory control began to utilize larger amounts of computer capacity than the design process itself. Finally, with the introduction of numerically controlled machine-tools, the manufacturing field began to establish a requirement for the digital computer.

The need for today's large-scale digital computer is, therefore, clearly established; many specialists in engineering disciplines find the capacity of today's large-scale computer the factor limiting further developments in their field. Three-dimensional real gas calculations, for example, are still by and large impractical for design purposes with today's computer speed-storage combination. Certain classes of atmospheric flight path optimization calculation require many hours of large-scale digital computer time for solution, and other examples are easily forthcoming.

Throughout increasing use of the digital computer, the essence of the design process in the aerospace industry, that of design selection and development, has remained relatively untouched. Typically, a nominal design is selected on the basis of experience, judgment, and gross level preliminary studies. The design is examined by various specialist groups. In the case of an aircraft design, these will include:

- AERODYANAMICS
- STRUCTURES
- PROPULSION
- STRUCTURAL DYNAMICS AND AEROELASTICITY
- PERFORMANCE

Each discipline will engage in a critical assessment of the design from its particular specialist aspect. Trade studies in which the specialist perturbs prime airplane design parameters, weight, wing area, wing sweep, fuselage size, etc., will be undertaken. A considerable degree of overlap exists in these trade studies. Thus, the structural engineer requires the air-load distribution on the vehicle, essentially an aerodynamic problem. The aeroelastician requires vehicle deflections under specified types of loading, an aerodynamic and structural problem, etc. Traditionally, the disciplines have tended to work independently; when the structural engineer requires air-loads, he tends to compute them himself. In this he typifies practically all the specialist disciplines. Primarily, the structural engineer is performing a vehicle structures trade; he does not wish to complicate the problem beyond that point.

Each discipline, therefore, performs its own trade studies, and it is left to the vehicle designer to perform the overall system analysis leading to an improved design. This is not a straightforward problem. The aerodynamicist sees a better design resulting from a thinner wing; the drag is less. The structural engineer sees a better design resulting from a thicker wing; the vehicle structural weight is less for given loads. On the first iteration, the structural dynamicist may not have finished his calculations, hence structural dynamics feed-back may not be available.

On the basis of the trades, the designer selects a new design, and the process is repeated. This is the traditional airplane design cycle. The weaknesses of the traditional design cycle have recently created considerable interest in a new approach to vehicle design based on simulation of the entire interdisciplinary design process within the computer. Initially, these attempts concentrated on achievement of a consistent interdisciplinary point design evaluation. References 1 and 2 typify NASA approaches to this problem. While these applications deal primarily with transports for the 1980's, other programs are being developed for application to today's aircraft designs both by NASA and in industry. Typical of this work is the General Dynamics program SYNAC of Reference 3.

Computerized airplane design simulations are illustrated schematically in Figure 1. Vehicle parameters which determine gross characteristics of a particular design, wing-area, thickness-chord ratio, fuselage length, engine-size, etc., are supplied to a geometry program. Detailed geometrical characteristics are computed and used to compute aerodynamic, propulsive, structural, and mission characteristics. From this data vehicle performance characteristics, such as payload, range, landing and take-off speed, time-to-climb, etc., can be computed. With computerized tools of this type, the designer can specify a selected set of values for the vehicle design parameters, and the corresponding performance characteristics are computed automatically within the computer. Internal computations are consistent, repeatable, and in-step with each other. Vehicle trade studies can be carried out by the designer directly without the necessity of calling upon the engineering specialist. The specialist has constructed a "black box" program within the design simulation expressing his requirements. Preliminary design and vehicle definition can proceed until the designer has evolved a satisfactory design. At this point, the specialist must re-enter the picture, critically examining the final design using a depth of analysis currently prohibitive in the repetitive design simulation itself.

Preliminary design and vehicle definition can be expedited by use of computer aided design techniques such as multivariable optimization. The remainder of this paper discusses these techniques and there application to performance optimization at the complete system level.

10.2.2 Multivariable Optimization

The airplane design problem is essentially a large scale, non-linear multivariable optimization problem. Independent variables are the gross geometric and physical parameters defining the configuration in detail -- the vehicle design parameters. Dependent variables are the system performance characteristics -- range, payload, gross weight, landing and take-off speed, direct operating cost, etc. Corresponding to a given set of design parameters, $\overline{\alpha}$, a unique set of performance characteristics $\overline{F}$, are obtained.

$$\overline{F} = \overline{F}\ (\overline{\alpha}) \tag{10.2.1}$$

In the design process one of these characteristics will be selected for minimization or maximization. This is the payoff function

$$\phi = \phi\ (\overline{\alpha}) \tag{10.2.2}$$

In the case of cost, ϕ will be minimized; in the case of range, ϕ will be maximized. In some designs, the payoff criteria to employ will not be self-evident to the designer. In this case he may seek to define value function, V, which involves some combination of the performance characteristics

$$V = V(\overline{F}) \tag{10.2.3}$$

The value function is then employed as the payoff function. An alternative approach and one more readily interpreted is that of seeking constrained extremes. Constraint functions, $\overline{\Psi}$, are selected from the performance characteristics.

These constraints can always be defined such that

$$\overline{\psi} = \overline{\psi}(\overline{\alpha}) = \overline{0} \tag{10.2.4}$$

With this approach, for example, the designer seeks the maximum range vehicle having a given take-off and landing speed. A general technique for incorporating constraints into the optimization formulation is the well-known "penalty function" approach. Here, an augmented payoff function, $\overline{\phi}$, is constructed, in the minimization case

$$\overline{\phi} = \phi + \sum_{i=1}^{M} W_i \quad \psi_i^2 \tag{10.2.5}$$

ORIGINAL PAGE IS
OF POOR QUALITY

The W_i are a set of positive constraint weighting factors. Provided the W_i are sufficiently large in magnitude, minimization of Equation (5) corresponds to minimization of Equation (2) in the presence of the constraints Equation (4). In practice, the W_i may be determined in adaptive fashion on the basis of individual constraint behavior. Alternative approaches to the penalty function approach are available. For example, the steepest-descent method of Bryson, reference 21, which permits explicit elimination of constraint errors, and the methods proposed by Morrison, reference 26, and Kowalik, et al, reference 27, both of which convert the constrained extremal problem into a sequence of unconstrained extremal problems.

The designer may wish to impose inequality constraints on the design. For example, he may seek the maximum range vehicle whose take-off speed is fixed but whose landing speed does not exceed some specified value. Inequality constraints of this type can readily be transformed into the equality constraint form, Equation (4). Suppose the inequality is to be placed on the i^{th} performance function; then define a constraint, ψ_j, such that

$$\psi_j = F_i^2; \; F_i > 0$$

$$= 0 \; ; \; F_i \leq 0 \qquad (10.2.6)$$

Constraining ψ_j to zero is now equivalent to the constraint $F_i \leq 0$.

Frequently the designer imposes inequality constraints directly on the design parameters. Thus, he may require the best fuselage length in the range 200 to 300 feet. These limits will be dictated by a priori knowledge of the vehicle and its operating environment. Generally, then, the design parameters are subject to lower and upper limiting values, α^L and α^H, such that

$$\bar{\alpha}^L \leq \bar{\alpha} \leq \bar{\alpha}^H \qquad (10.2.7)$$

The constraints limit the region of feasible designs to a hyper-rectangle lying in the multi-dimensional design parameter space. Equations (1) through (7) define in symbolic fashion the aerospace vehicle design problem. Conceptually, they define most industrial design problems; for, in practice, the designer must always seek to express his problem in terms of a finite number of parameters.

Methods for solution of non-linear multivariable optimization problem have received considerable attention during the sixties. In general, solutions are obtained by the iterative search procedures which are collectively becoming known as "optimal seeking methods."[4] The increasing interest in these techniques stems both from their ready application through the digital computer and the ease with which the designer can grasp their theoretical basis. Generally, the non-linear optimal seeking method has its basis in logic rather than the higher branches of analytic mathematics. In essense, the technique corresponds quite closely to the designers traditional design cycle. Parameters are perturbed; the system is evaluated, and, on the basis of resulting performance characteristics, a new design is evolved. Figure 2 presents a schematic diagram of the optimal seeking approach. A nominal design, $\overline{\alpha}_0$, is supplied to the optimization algorithm. The optimizer, in turn, supplies the design parameter values to a digital model of the system being designed. This system functions in "black box" fashion and returns the corresponding performance characteristics, $\overline{F}$, to the optimizer. Based on inspection of these characteristics, a new design, $\overline{\alpha}$, is supplied to the system, and the process repeats in iterative fashion until the optimal performance $\phi = \phi^*$ for the constraint levels $\overline{\psi} = \overline{0}$ is attained.

It can be seen that the optimization process is largely divorced from the system model. This fact permits construction of generalized optimization programs which can readily be coupled to digital system models. These models may be expressly constructed with this object in mind, or, equally, they may be existing digital system models constructed for conventional designer control and perturbation. An example of this type of generalized optimization program is AESOP[5,6] (Automated Engineering and Scientific Optimization Program) recently constructed under contract to the National Aeronautics and Space Administration's Office of Advanced Research and Technology. This optimization program has been successfully applied to a variety of engineering design optimization problems[7,8,9,10] some of which are listed in Figure 3.

The success of AESOP is largely due to the provision of several alternate search algorithms within the program. These searches may be employed either separately or in conjunction with each other. Techniques for search acceleration are incorporated as is a general method for location of more than one extremal. A schematic diagram of the optimization program is presented in Figure 4. The search algorithms include the following.

<u>Sectioning search</u> exhaustively searches the range of each parameter in turn for the one-dimensional optimum. The values of the parameters are fixed at the optima as they are achieved. The procedure is repeated until no further gain is possible. The parameter order can be chosen by the user or selected at random. This search can be used for evaluating non-optimum sensitivities about any point in the parameter space since each search essentially describes a one-dimensional cut through the multi-dimensional design parameter space.

<u>Creeping search</u> is similar to sectioning in that the parameters are perturbed in turn one at a time. In the creeping algorithm, however, the parameters initially undergo only small incremental changes in the favorable performance direction. On repetitive cycles the step size is increased independently in each parameter until further gain is impossible in either increasing or decreasing directions. An order of magnitude reduction in stepsize is then effected,

and the process is repeated. At any given moment some parameter stepsizes may be increasing while others are decreasing. Ultimately, all stepsizes are reduced to prespecified minimum values and the search is discontinued.

Random point search is essentially a Monte Carlo technique which distributes points uniformly in the control parameter space. After a prespecified number of evaluations of the objective function, the control vector providing the best performance characteristics is retained.

Magnify search scales all the control parameters uniformly in the favorable direction until the local optimum is achieved.

Steepest descent search relies on the numerical partial derivatives of the objective function with respect to the control parameters to predict a favorable direction. In effect, a tangent plane is fitted to the objective function surface at the starting point. Numerical derivatives are computed by two-sided perturbation of each design parameter and are thus correct to second order. In its simplest form the search proceeds in the gradient direction. Experience has shown that gradient direction search is often very inefficient. Ridge lines are rapidly located; from that point gradient search becomes a sequence of oscillatory perturbations along the ridge. Algorithm extensions have been incorporated in AESOP which allow the search to proceed in a weighted gradient direction. The weighting matrix employed as a perturbation measure is adaptially determined within the program by nondimensionalization of the search hyper-rectangle, local partial circularization of the payoff function contours, and, most important, by an adaptive learning mechanism based on previous search behavior.

Quadratic search fits a second order surface to the payoff function at a nominal design point. The extremal of the approximating quadratic surface is predicted, and the search proceeds along the ray defined by (a) the initial point and (b) the predicted extremal point. This technique, although developed as a search procedure is also useful for predicting optimal second-order sensitivities about the optimal design point.

Davidon search or deflected gradient method essentially combines features of steepest descent and quadratic searches. The procedure initially searches in the gradient direction. Recursive relationships permit development of approximate second order information from successive ray searches. This information is used to develop a weighting matrix which provides quadratic convergence. The method can become somewhat ill-conditioned if the payoff response surface does not exhibit almost quadratic form in the search region.

Pattern search can be applied after successive applications of any combination of other searches. It uses the starting point from the first search and the final point from the last search to define a new search ray. This ray is searched in the favorable direction for the local optimum. It is essentially an acceleration technique exploring gross directions revealed by other organized search algorithms.

Random ray search proceeds on the basis of small randomly selected design parameter perturbations. Perturbation magnitude is adaptively determined on basis of past performance characteristic behavior. This can be very efficient when used in conjunction with pattern search when there are many interacting design parameters.

In addition to the nine searches which assume unimodality of the performance response surface, AESOP contains a method of locating more than one extremal. The program multiple extremal technique consists of design parameter space warping. A transformation is applied to the parameter space such that all the extremals of the performance response function are retained in the transformed space but the relative locations are altered in an inverse exponential manner about an arbitrary point in the original space. In practice the transformation is performed about some previously discovered extremal point. Subsequent searches in the transformed space then have a reduced probability of finding the same extremal. This probability depends on the exponential order of the transformation selected by the user.

10.2.3 Sub-System Optimization

Sub-system optimization, as defined in this paper, refers to the optimization of the aerospace vehicle from the aspect of a single discipline. To-date applications of optimization theory to vehicle design in a single discipline have been abundantly reported.

In the field of supersonic aerodynamics, the results of Jones[11], Lomax[12], and Heaslett[12] typify analytic approaches to this problem through the variational calculus. Woodward[13,14] has demonstrated the power of numerical approaches to optimal aerodynamic shaping problems when payoff and constraints are related to vehicle surface slopes in a linear fashion. An excellent survey of recent developments in the general aerodynamic optimization problem is that of Miele[15].

In the structural design area considerable progress has been made through the combination of specialized optimal seeking methods and large scale structural matrix analysis. This work is typified by that of Gellatly[16], Venkayya[17], and others. A summary of much of this activity is provided by the recent Air Force sponsored Conference on Matrix Methods[18].

Performance optimization studies for spacecraft have been reported extensively. State-of-the-art in this area can be assessed from the work of

Jezewski and Rozendaal[19]. In the past few years this area has produced a prolific number of papers in the AIAA Journal, the Journal of Optimization Theory and Application, and elsewhere. Atmospheric flight path optimization has received considerably less attention. The state of optimization theory application in this area is summarized by the work of Rutowski[20], Bryson[21], Hague[22], and Landgraf[23]. The dominant approach in all performance optimization work to-date has been the variational calculus.

The vehicle designer confronted with the outpouring of special techniques for optimization in each area and the myriad of assumptions and approximations made to produce a tractable problem is understandably confused. Specialists in optimization theory itself experience difficulty keeping abreast of developments in more than one area. The major objective of this paper is to demonstrate that, at the expense of some elegance in technique and resultant form of the solution, optimization problems involving combined aerospace vehicle design disciplines can be solved by the straightforward non-linear optimal seeking method.

10.2.4 Aerospace Vehicle System Performance Design Optimization

It is apparent from Figures 1 and 2 that the total system performance design optimization problem can be considered as a large scale multivariable optimization problem. When the system designer examines the results of single discipline trade studies in an attempt to arrive at an improved vehicle design, he is applying the techniques of optimal seeking methods. The designer's approach to vehicle optimization does, however, depend on intimate knowledge of the vehicle performance trade study characteristics and past design experience for successful application. Hence, there has been a tendency to assume that vehicle system performance optimization would not be amenable to routine automation within the computer. The major intent of the present paper is to demonstrate that this is not so; rather the vehicle performance optimization problem differs only in the degree of complexity not in kind from the typical sub-system optimization problems discussed above.

10.2.4.1 Vehicle and Mission Characteristics.

The results presented below are obtained from a hypersonic airplane design optimization program[7], constructed under contract* to the National Aeronautics and Space Administration's Mission Analysis Division at Ames Research Center. The program has two major elements. A hypersonic aircraft synthesis program constructed by NASA personnel[1,2] and the generalized multivariable optimization program AESOP[5,6] discussed in Section 2 of this paper. Typical designs studied to-date include a hypersonic transport (HST), Figure 5 and a hypersonic research aircraft (HRA), Figure 6. Both aircraft presented illustrate configurations arrived at by application of the multivariable search techniques presented previously. The HRA configuration of Figure 6 was determined by in-house studies at NASA. The applications described below are based on the HST concept of Figure 5.

* Contract NAS2-4507 NASA Headquarters OART Mission Analysis Division, Ames Research Center

The vehicle under study is a 500,000 pound liquid hydrogen fueled, subsonic burning turboramjet powered, delta winged hypersonic transport aircraft with a range of 5500 nautical miles. The objective of the study* was selection of the vehicle geometry, propulsion, and mission characteristics which maximize the number of passengers carried over the design flight range subject to certain constraints such as vehicle takeoff distance, landing speed, and sonic boom ground overpressure.

The design synthesis developed by NASA is similar to that shown schematically in Figure 1. The synthesis commences with basic geometry, propulsion and mission characteristics. This information is supplied to a detailed geometry package. Aerodynamic coefficients are determined from the geometry description and stored as a function of Mach number and angle of attack. Engine data is determined from an engine design module based on data supplied by engine manufacturers. Mission performance is computed from preselected climb-cruise-descent profile. Structural and equipment weight is determined from historical data developed under a separate contract to NASA. The remaining mass, considered to be payload, is apportioned to passengers and passenger equipment. This design synthesis is the culmination of several years effort on the part of both NASA personnel and several NASA contractors.

10.2.4.2 Vehicle Characteristics Optimization.

Initially, five primary design parameters are chosen for optimal selection on the basis of unconstrained maximum passenger capability. The parameters are wing loading, aspect ratio, fuselage fineness ratio, an engine sizing parameter, and engine compressor pressure limit. A nominal design, Figure 5, produced 220 passengers. After approximately 50 point design evaluations by AESOP using the adaptive creeping search, the optimal design achieved 253 passengers over the specified range. Performance convergence of the optimization process is also shown (solid line) in Figure 7 in terms of number of passengers attained versus number of design evaluations. Confidence in the solution was obtained by an independent optimization calculation using a different nominal design. This design was arbitrarily chosen as that resulting from selection of the maximum allowable value for all five design parameters being perturbed. Again, the payoff function converges to about 253 passengers, Figure 7. Nominal and final values of the design parameters are given in the table accompanying Figure 7.

Convergence of the design parameters themselves is illustrated in Figures 8 and 9. Initially small perturbations are produced in the direction of favorable performance. Perturbations are increased on successive cycles until further gain in either direction is impossible. The perturbation stepsize in that parameter reduces, and the process is repeated until convergence to the optimal design point is achieved. It may be noted from Figures 8 and 9 that control parameters do not converge to identical values from the two nominal designs; although payoff function values are practically identical. Sensitivity of the objective function to changes in control parameters is low near the optimum. The "hill" is smooth. It may also be noted in Figure 9 that large perturbations occur in the engine pressure parameter even after convergence of the payoff function, an indication that engine pressure is an insensitive parameter. This fact is also evidenced in Figure 10 which illustrates a one-dimensional cut in the engine pressure-payoff function plane produced by the sectioning search discussed in Section 2 of this paper. Similar cuts are presented for fuselage fineness ratio and aspect ratio in Figure 11 and 12. Fuselage fineness ratio is a sensitive but apparently uncoupled design parameter. Cuts in the fuselage fineness ratio/payoff function plane possess the same shape about nominal and optimal design points. This is not true of wing aspect ratio. If the designer had the task of determining optimal aspect ratio from sensitivities about the nominal point design, he would choose the lower acceptable limit. When the design is optimized, the aspect ratio lies in the middle of the acceptable range.

Solution of this five-parameter problem using several different nominals and search techniques provided confidence of the ability of multivariable search techniques in solution of vehicle performance optimization problems. A more complex example is presented in Figure 13. Here the original five parameters are combined with five additional parameters: thrust deflection angle, wing and stabilizer thickness ratios, and aspect ratios of the horizontal and vertical stabilizer. The configuration which carried 220 passengers over 5500 nautical miles was used as a nominal point design. After approximately 75 perturbations of these ten design control parameters, the HST passenger carrying capability was 260. The five additional design degrees of freedom resulted in seven additional passengers, a logical result of the expansion of the parameter space. The computational requirements to achieve this result are significant. Although the problem required selection of ten design parameters, the number of evaluations to define the optimal design was only 50 percent more than that required in the five design parameter problem.

10.2.4.3 Vehicle Mission Optimization.

Vehicle mission or trajectory optimization problems traditionally have been solved by variational calculus; for optimal vehicle control must be established at all instants of time. Variational optimization techniques involve large computer requirements for time integration of the equations of motion in the atmosphere. While optimization in which continuous control and a finite number of design parameters are simultaneously considered is feasible, reference 25, this approach is both unwieldy and not necessarily representative of the actual design optimization problem.

The mission analyst may, however, treat his problem by a more elemental means with little loss in accuracy. Using a reduced set of motion equations, involving elimination of flight path angle rate terms, the analyst can uniquely describe the motion of the vehicle in the Mach-altitude plane and compute the mission time history[6]. This technique is representative of the actual performance and is well suited to parameter optimization. The array of altitude parameters at arbitrary Mach number points are taken as the problem parameters. Performance criteria selected in the present study is the same as that employed previously, payload at the mission end. Figure 14 presents a typical unconstrained optimal flight path obtained in the study. The resultant Mach-altitude profile has all the expected characteristics for this type of vehicle. A subsonic climb is followed by a dive and zoom through the transonic region. The aircraft then climbs steadily before leveling out at Mach 3 where the turbojet engine performance efficiency begins to decrease. Climb performance improves again at about Mach 4 where the ramjet engine begins to operate efficiently. A 3 psf ground sonic boom overpressure constraint is displayed in Figure 14. Flight path optimization in the presence of this constraint results in a path lying along the constraint boundary in the region $1 \leq M \leq 3$.

10.2.4.4 Combined Vehicle Design Parameter and Mission Optimization.

Combined design and trajectory optimization studies in this paper involve selection of the ten design parameters and the trajectory parameters previously employed. The object remains that of maximizing passenger carrying capability over the 5500 nautical mile mission. Design constraints of sonic boom overpressure, take-off distance, and landing speed were sequentially applied. The results of these calculations are compared to earlier results in Figure 15 and the nominal design which achieves 220 passengers. The five-variable solutions of Section 4.1 produce a payload of approximately 253 passengers independently of the search technique employed. Introduction of five additional parameters, Section 4.2, permitted seven more or a 260 passenger payload capability. Trajectory optimization alone permitted no significant gain in performance over the nominal design when the sonic boom constraint was applied. By permitting penetration of the sonic boom boundary, a gain of nine passengers is possible. Based on results obtained to that point, the designer might assume that combining both trajectory and design parameter optimization, approximately 269 passengers could be expected in the unconstrained case. When such a calculation is performed, however, a payload of 286 passengers is achieved. This result indicates a strong coupling between design and trajectory parameters. This is quite significant to the vehicle designer; for current aircraft design practice usually separates the selection of optimal design parameters and optimal mission profile.

The effect of adding vehicle operating constraints sequentially is tabulated in Figure 15. It can be seen that addition of the sonic boom constraint reduces payload to 265 passengers, a loss of approximately 20 passengers. Satisfaction of a take-off constraint (clearance of a 50 feet high obstacle within 10,000 feet of ground roll commencement) reduces payload capability to 246 passengers. a loss of twenty additional passengers from the sonic boom constrained solution. Finally, simultaneous satisfaction of sonic boom, take-off, and a landing approach speed constraint of 140 knots reduces the optimal payload to 222 passengers.

10.2.5 Conclusion

The aerospace vehicle performance optimization problem has been discussed in some detail. It has been pointed out that multivariable parameter optimization, or optimal seeking methods, are well-suited to solution of system optimization on a performance basis.

Multivariable parameter optimization techniques are discussed in some detail as is a generalized parameter optimization digital computer program, AESOP. This program has seen extensive application both to aerospace subsystem and total system design from an engineering aspect. The program is capable of rapid coupling to either existing system models or to system models specifically created for this purpose. In the examples of system optimization presented designs appear reasonable from the engineering aspect.

It should be noted that true system performance must include the impact of economic factors in addition to the engineering design parameters. The combination of large scale aerospace vehicle design and cost synthesis when coupled to the optimal seeking methods hold the prospect of a true, quantitative systems analysis approach, free of the often unrealistic limitations imposed by linear and quadratic programming approaches.

REFERENCES

1. Petersen, R. H., Gregory, T. J., and Smith, C. L., "Some Comparisons of Turboramjet-Powered Hypersonic Aircraft for Cruise and Boost Missions," Journal of Aircraft, Vol. 3, No. 5, September-October 1966.

2. Gregory, T. J., Petersen, R. H., and Wyss, J. A., "Performance Trade-Offs and Research Problems for Hypersonic Transports," Journal of Aircraft, July-August, 1965.

3. Lee, Vernon A., et al., "Computerized Aircraft Synthesis," Journal of Aircraft, Vol. 4, No. 5, September-October 1967.

4. Wilde, D. J., Optimal Seeking Methods, Prentice-Hall, Inc., 1964.

5. Hague, D. S. and Glatt, C. R., An Introduction to Multivariable Search Techniques for Parameter Optimization (and Program AESOP), NASA CR-73200, April 1968.

6. Hague, D. S. and Glatt, C. R., A Guide to the Automated Engineering and Scientific Optimization Program (AESOP), NASA CR-73201, April 1968.

7. Hague, D. S. and Glatt, C. R., Application of Multivariable Search Techniques to the Optimal Design of Hypersonic Cruise Vehicle, NASA CR-73202, April 1968.

8. Hague, D. S., Application of Multivariable Search Techniques to the Shaping of Minimum Total Heat Re-entry Bodies at Hyperbolic Velocity, NASA CR-73203, April 1968.

9. Hague, D. S., Rozendall, H. L., and Woodward, F. A., "Application of Multivariable Search Techniques to Optimal Aerodynamic Shaping Problems," The Journal of the Astronautical Sciences, Vol. 15, No. 6, November-December 1968, pp. 283-296.

10. Sandrin, W. A., Glatt, C. R., and Hague, D. S., "Design of Arrays with Unequal Spacing and Partially Uniform Amplitude Taper," IEEE Transaction on Antennas and Propagation, September, 1969.

11. Jones, R. T., Theory of Wing-Body Wave Drag at Supersonic Speeds, NACA Report 1284, 1956.

12. Lomax, H. and Heaslett, M. A., "Recent Developments in the Theory of Wing-Body Wave Drag," Journal of Aeronautical Sciences, December 1956.

13. Woodward, F. A., "Analysis and Design of Wing-Body Combinations at Subsonic and Supersonic Speeds," Journal of Aircraft, Vol. 5, No. 6, November-December 1968.

14. LaRowe, E. and Love, J. E., Analysis and Design of Supersonic Wing-Body Combinations, Including Flow Properties in the Near Field, Part II - Digital Computer Program Description, NASA CR 73107, August 1967.

15. Miele, A., Theory of Optimum Aerodynamic Shapes, Academic Press, 1965.

16. Gellatly, Ronald A., Development of Procedures for Large-Scale Automated Minimum Weight Structural Design, AFFDL-TR-66-180, December 1966.

17. Venkayya, V. B., An Iterative Method for the Analysis of Large Structural Systems, AFFDL-TR-67-194, April 1968.

18. United States Air Force, Conference on Matrix Methods, Dayton, Ohio.

19. Jezewski, D. J. and Rozendaal, H. L., An Efficient Method for Calculating Optimal Free Space N-Impulse Trajectories, NASA Manned Spacecraft Center Note 67-GM-170 (to be published in a forthcoming issue of the AIAA Journal).

20. Rutowski, E. R., "Energy Approach to the General Aircraft Performance Problem," Journal of the Aeronautical Sciences, March 1954.

21. Bryson, A. E. and Denham, W. F., A Steepest-Descent Method for Solving Optimal Programming Problems, Raytheon Report BR-1303.

22. Hague, D. S. and Glatt, C. R., Optimal Design Integration of Military Flight Vehicles, ODIN/MFV, Section 7.3, AFFDL-TR-72-132, 1972.

23. Landgraf, S. K., Some Practical Applications of Performance Optimization Techniques to High Performance Aircraft, AIAA Paper No. 64-288, Presented at the First AIAA Annual Meeting, June 1964.

24. Dolph, C. L., A Current Distribution for Broadside Arrays Which Optimizes the Relationship Between Beam Width and Sidelobe Level, Proc. IRE, Vol. 34, June 1946, pg. 335.

25. Hague, D. S., "The Optimization of Multiple-Arc Trajectories by the Steepest-Descent Method," Recent Advances in Optimization Techniques, edited by Lavi and Vogl, John Wiley, 1966, pp. 489-517.

26. Morrison, D. D., Optimization by Least Squares, SIAM Journal Numerical Analysis, (1968), p. 83.

27. Kowalik, J., Osborne, M. R., and Ryan, D. M., A New Method for Constrained Optimization Problems, (To Be Published in a Forthcoming Volume of the Journal of the Operations Research Society of America).

HYPERSONIC TRANSPORT SYNTHESIS

BASIC GEOMETRY - Wing Area, Thickness-Chord Ratio, Fuselage, Length, Engine-Size, etc.

↓

DETAILED CONFIGURATION GEOMETRY

↓

VEHICLE AERODYNAMICS

↓

ENGINE DESIGN

↓

MISSION

↓

STRUCTURAL DESIGN AND WEIGHT SUMMARY

↓

PAYLOAD, RANGE, LANDING & TAKE-OFF, SPEED, TIME-TO-CLIMB, etc.

FIGURE 10.2-1 HYPERSONIC TRANSPORT SYNTHESIS

OPTIMIZER SCHEMATIC

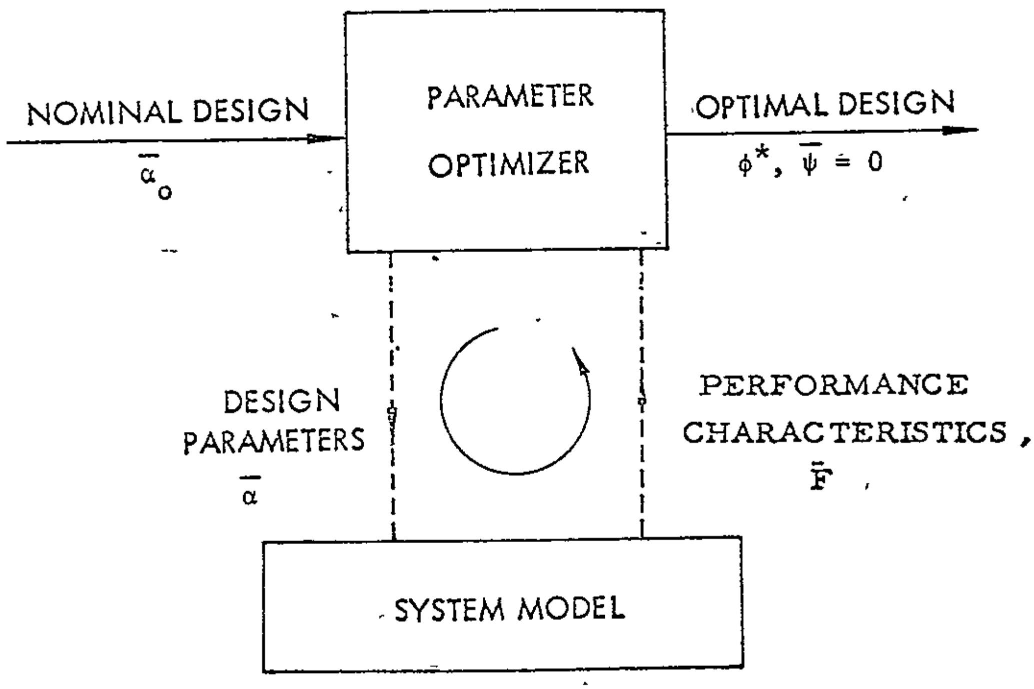


FIGURE 10.2-2 OPTIMIZER SCHEMATIC

ENGINEERING PROBLEMS SOLVED BY AESOP

- AERODYNAMIC AND AEROTHERMODYNAMIC SHAPING
  - AIRFOILS
  - BODIES
  - WINGS
  - ROTOR BLADES
- GUIDANCE AND CONTROL
  - GRAVITY GRADIENT ATTITUDE CONTROL
  - HYDRODYNAMIC TRIM CONDITIONS
  - AERODYNAMIC TRIM CONDITIONS
- MISSION PERFORMANCE
  - LOW THRUST INTERPLANETARY SHAPING
  - INTERPLANETARY MISSION PERFORMANCE
  - AIR CRAFT PERFORMANCE
- STRUCTURES
  - BORON FILAMENT STRUCTURES
  - STRUCTURES WITH NATURAL FREQUENCY CONSTRAINTS
- COMMUNICATION AND ELECTRONICS
  - ANTENNA DESIGN
- TOTAL SYSTEMS
  - HYPERSONIC TRANSPORT DESIGN AND PERFORMANCE
  - INTERPLANETARY PROPULSION MODULE DESIGN
  - STOL VEHICLE DESIGN

FIGURE 10.2-3 ENGINEERING PROBLEMS SOLVED BY AESOP

ALGORITHMS FOR OPTIMIZATION

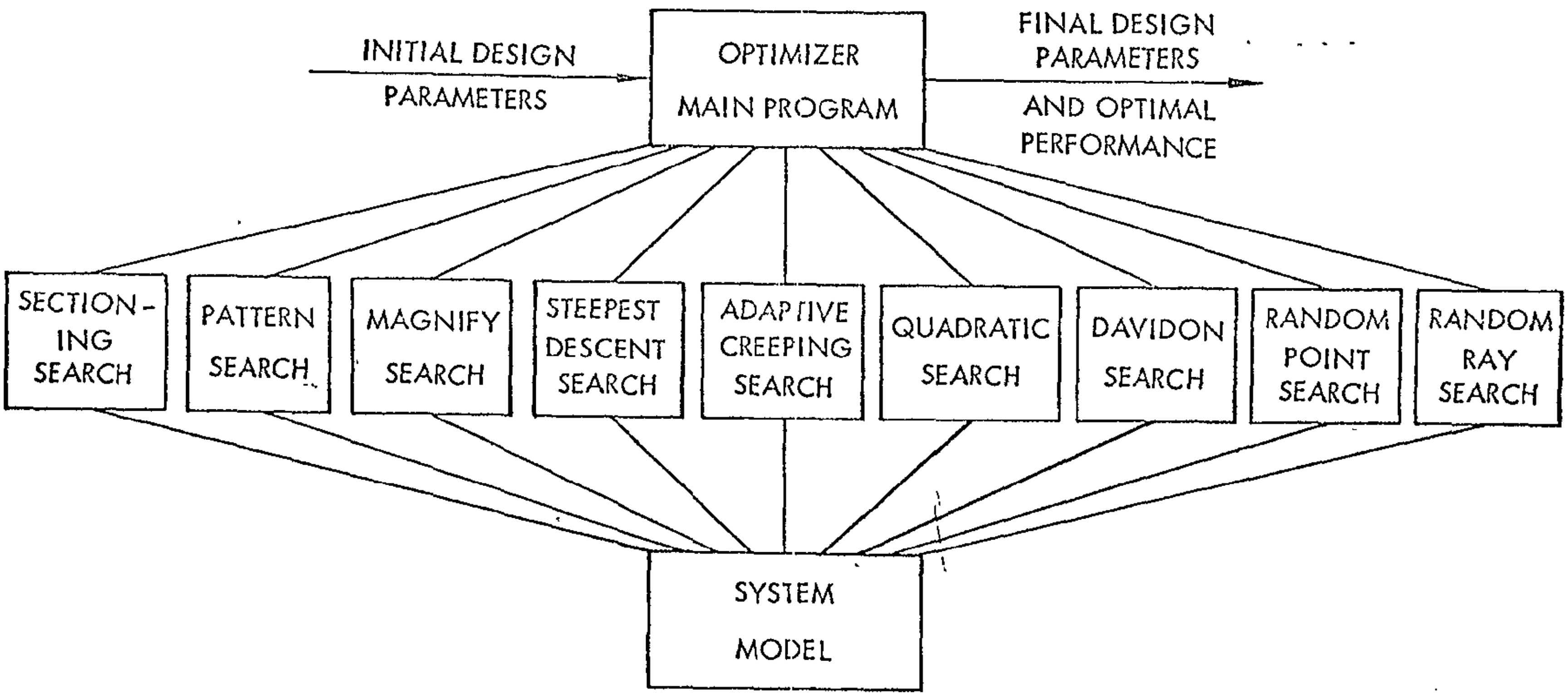


FIGURE 10.2-4 ALGORITHMS FOR OPTIMIZATION

NOMINAL VEHICLE CONFIGURATION

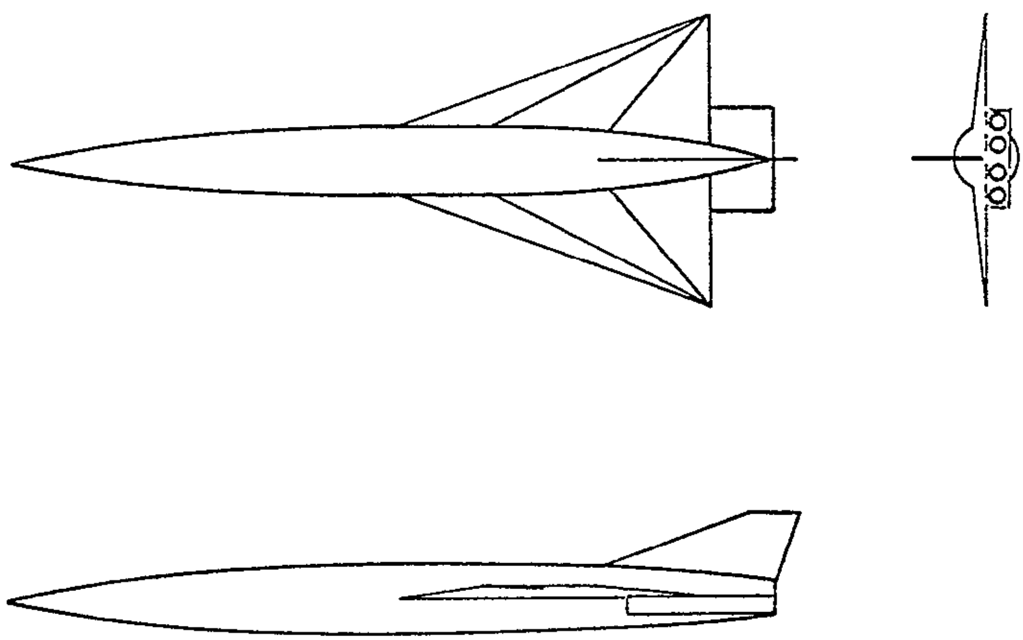


FIGURE 10.2-5. NOMINAL VEHICLE CONFIGURATION

OPTIMIZED HRA GENERAL ARRANGEMENT

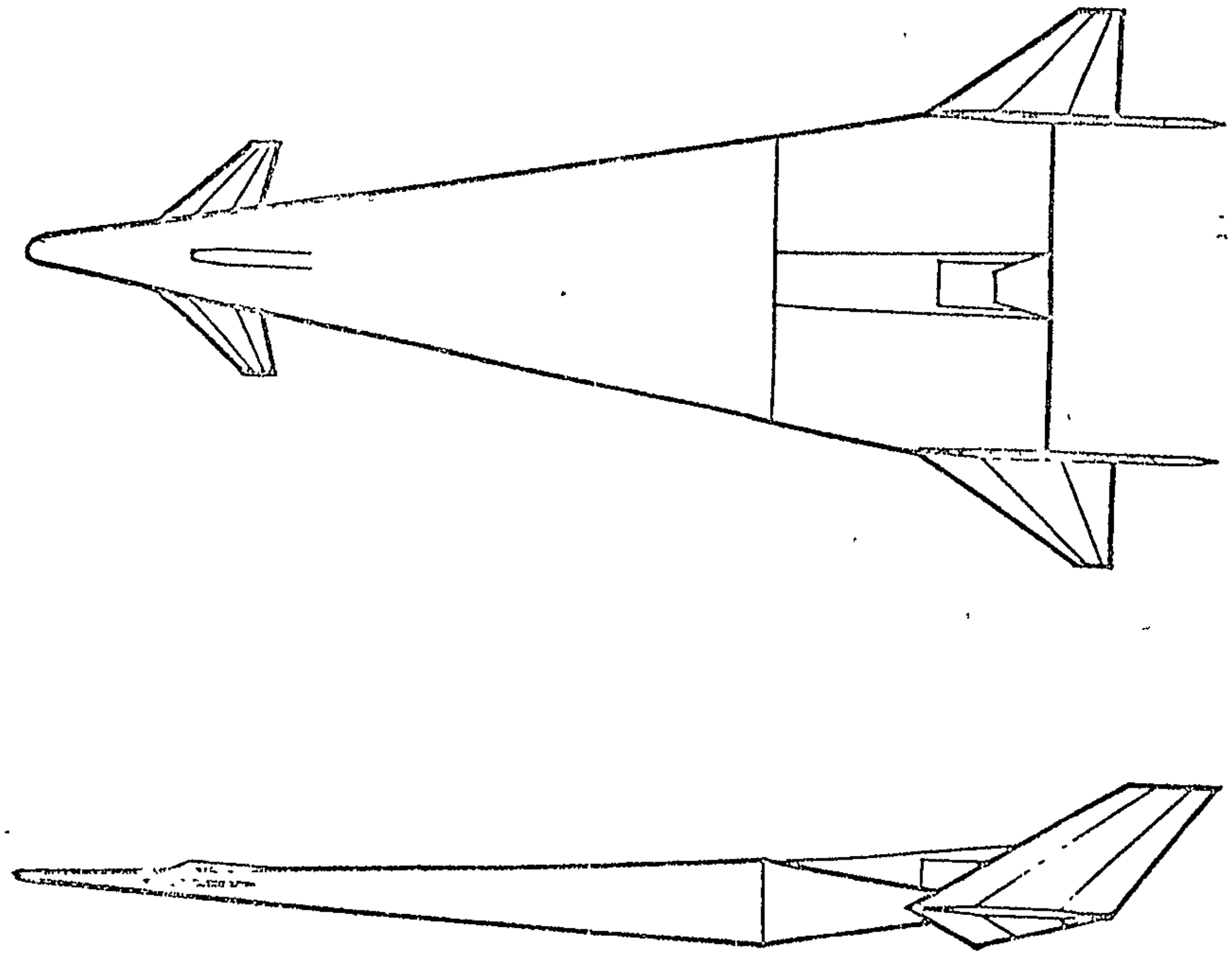
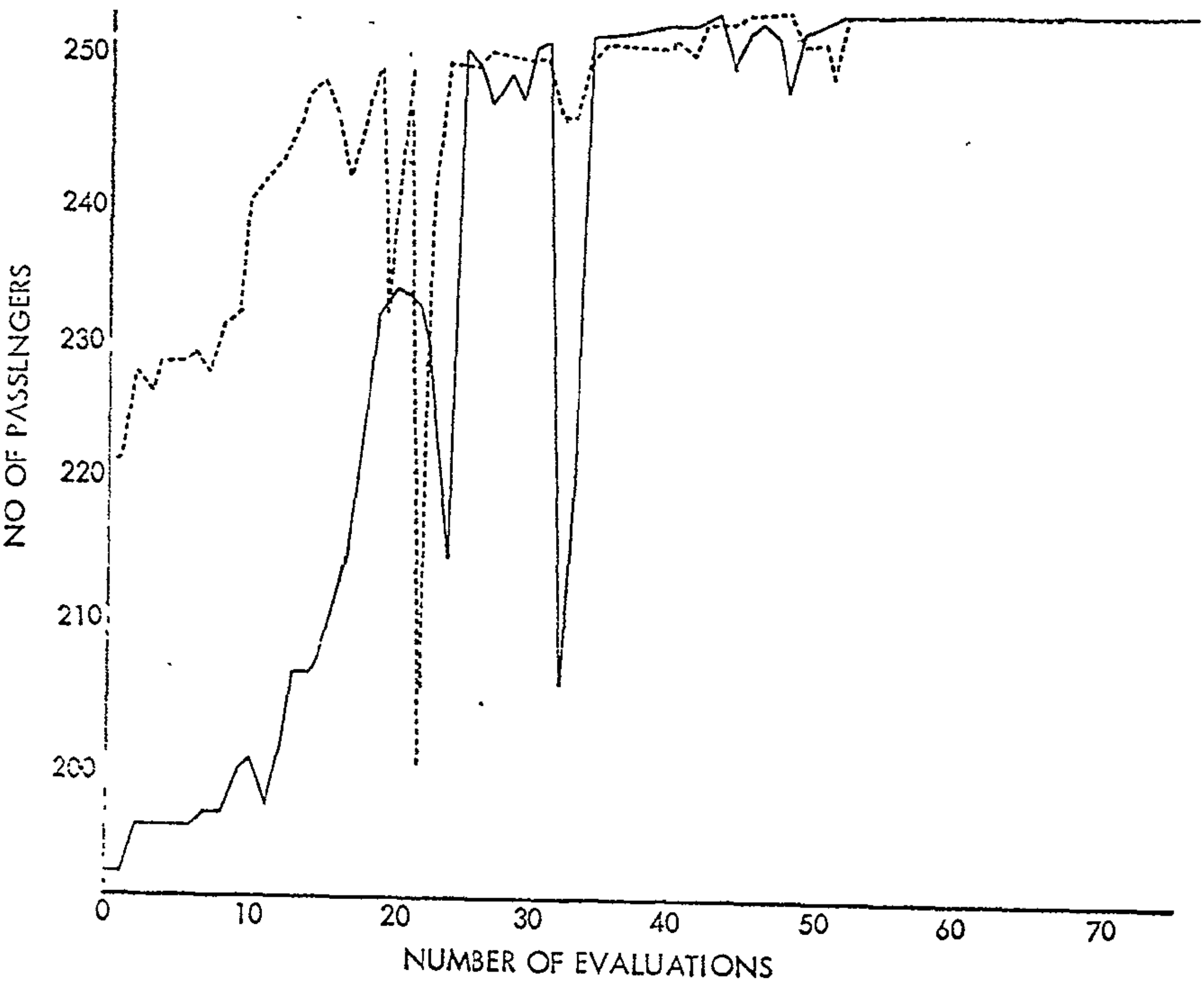


FIGURE 10.2-6 OPTIMIZED HRA GENERAL ARRANGEMENT

OBJECTIVE FUNCTION CONVERGENCE
(NO. OF PASSENGERS)



PARAMETER	NOMINAL	FINAL 1	FINAL 2
WING LOADING	80	108.5	115.2
ASPECT RATIO	1.455	1.499	1.563
FUSELAGE FINENESS	14.0	15.80	15.46
ENGINE SIZING PARAMETER	4.0	3.30	3.36
COMPRESSOR PRESSURE LIMIT	200.0	149.96	150.4
NUMBER OF PASSENGERS	220.3	253.0	253.4

FIGURE 10.2-7 OBJECTIVE FUNCTION CONVERGENCE

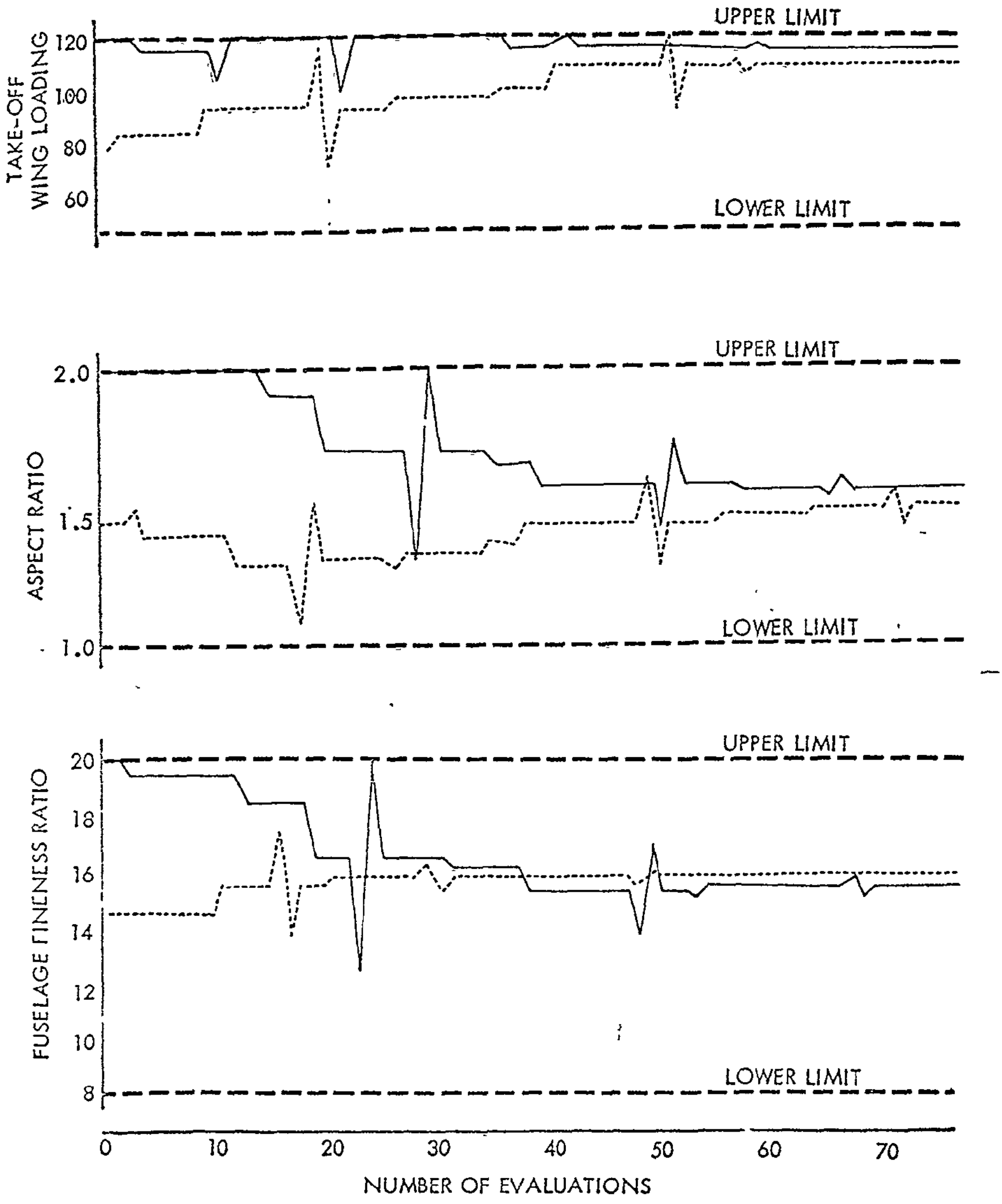


FIGURE 10.2-8 AERODYNAMIC PARAMETER CONVERGENCE - FIVE VARIABLE PROBLEM

ENGINE PARAMETER CONVERGENCE - FIVE VARIABLE PROBLEM

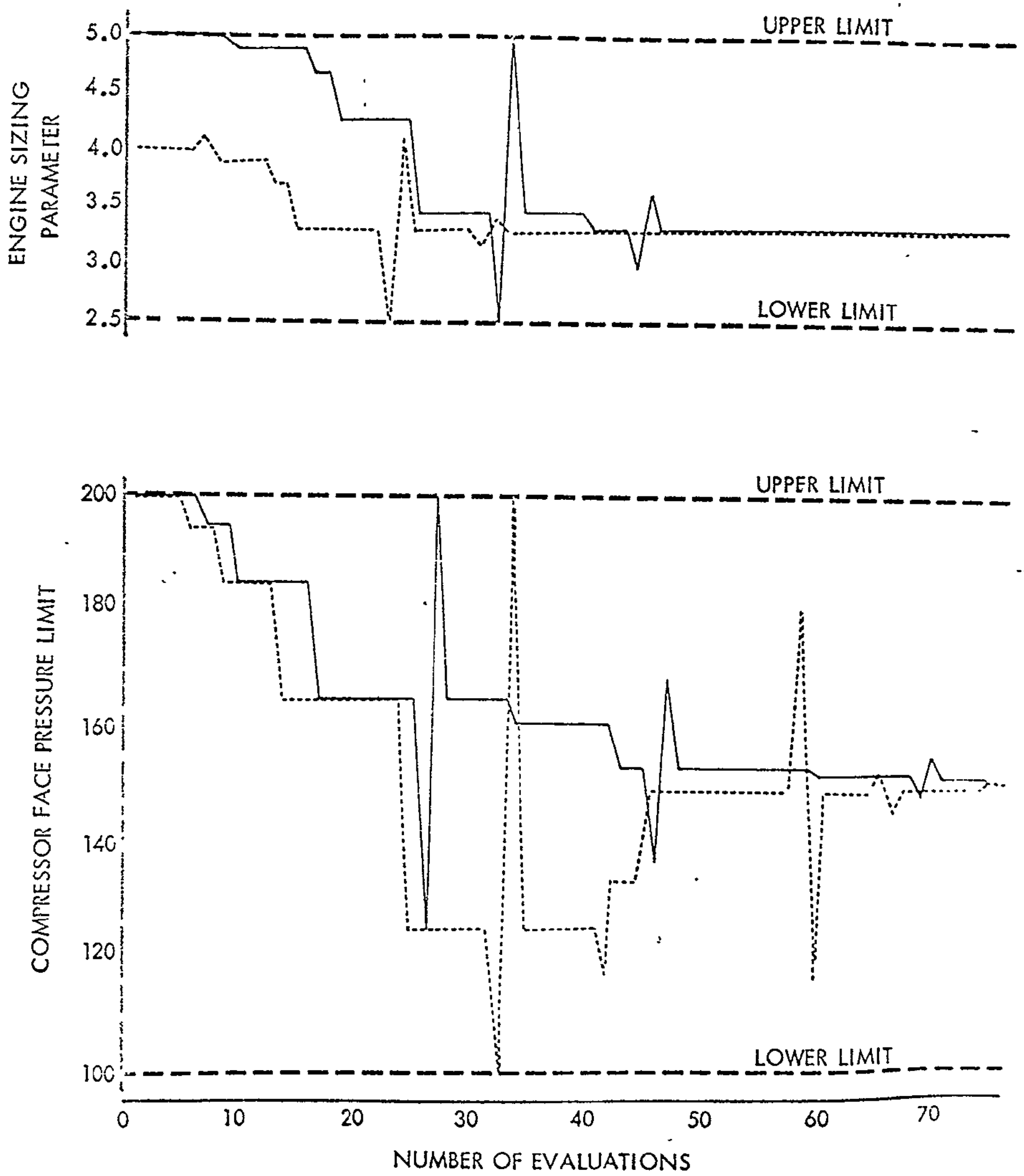


FIGURE 10.2-9 ENGINE PARAMETER CONVERGENCE - FIVE VARIABLE PROBLEM

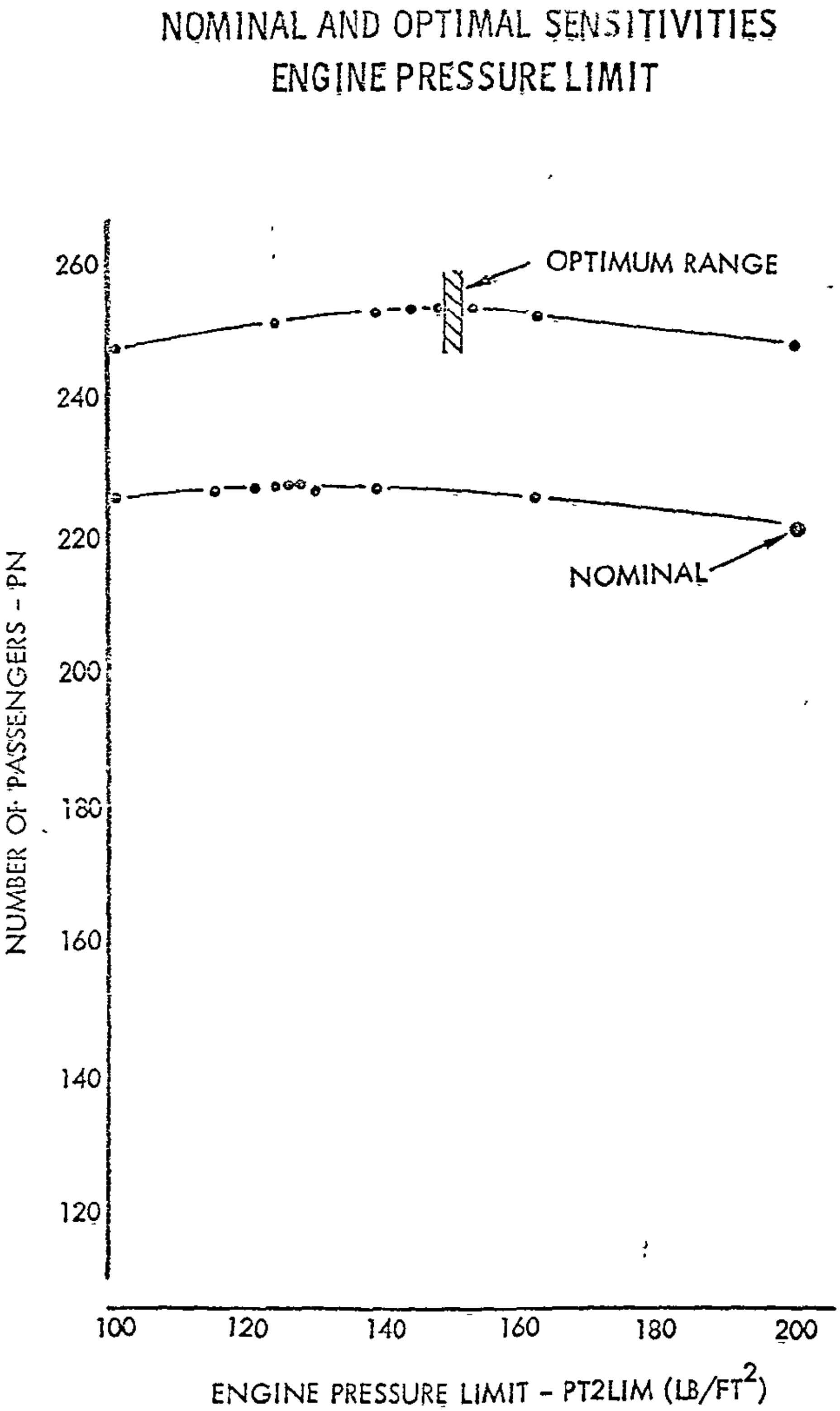


FIGURE 10.2-10 NOMINAL AND OPTIMAL SENSITIVITIES ENGINE PRESSURE LIMIT

NOMINAL AND OPTIMAL SENSITIVITIES
FUSELAGE FINENESS RATIO

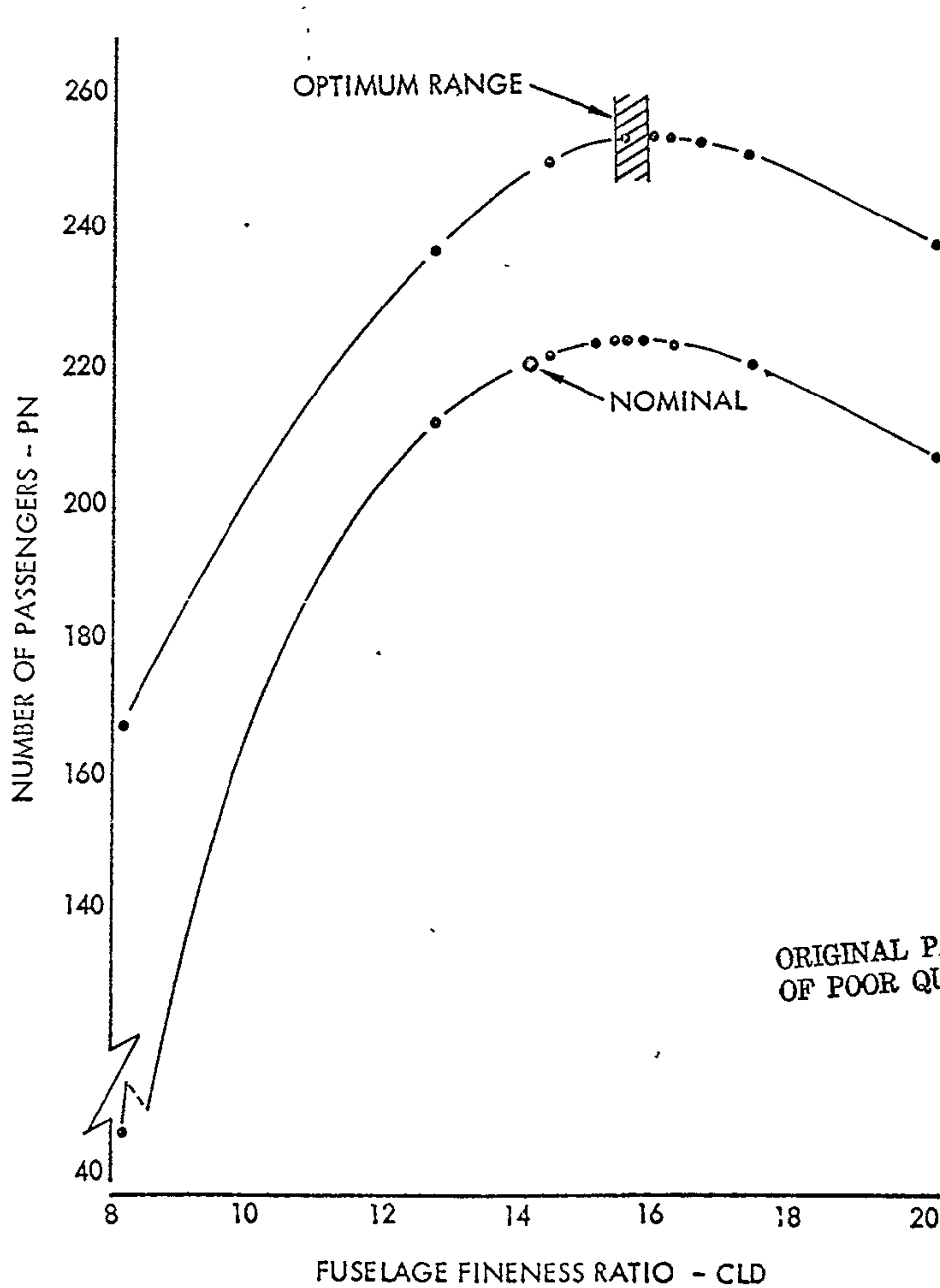


FIGURE 10.2-11 NOMINAL AND OPTIMAL SENSITIVITIES FUSELAGE FINENESS RATIO

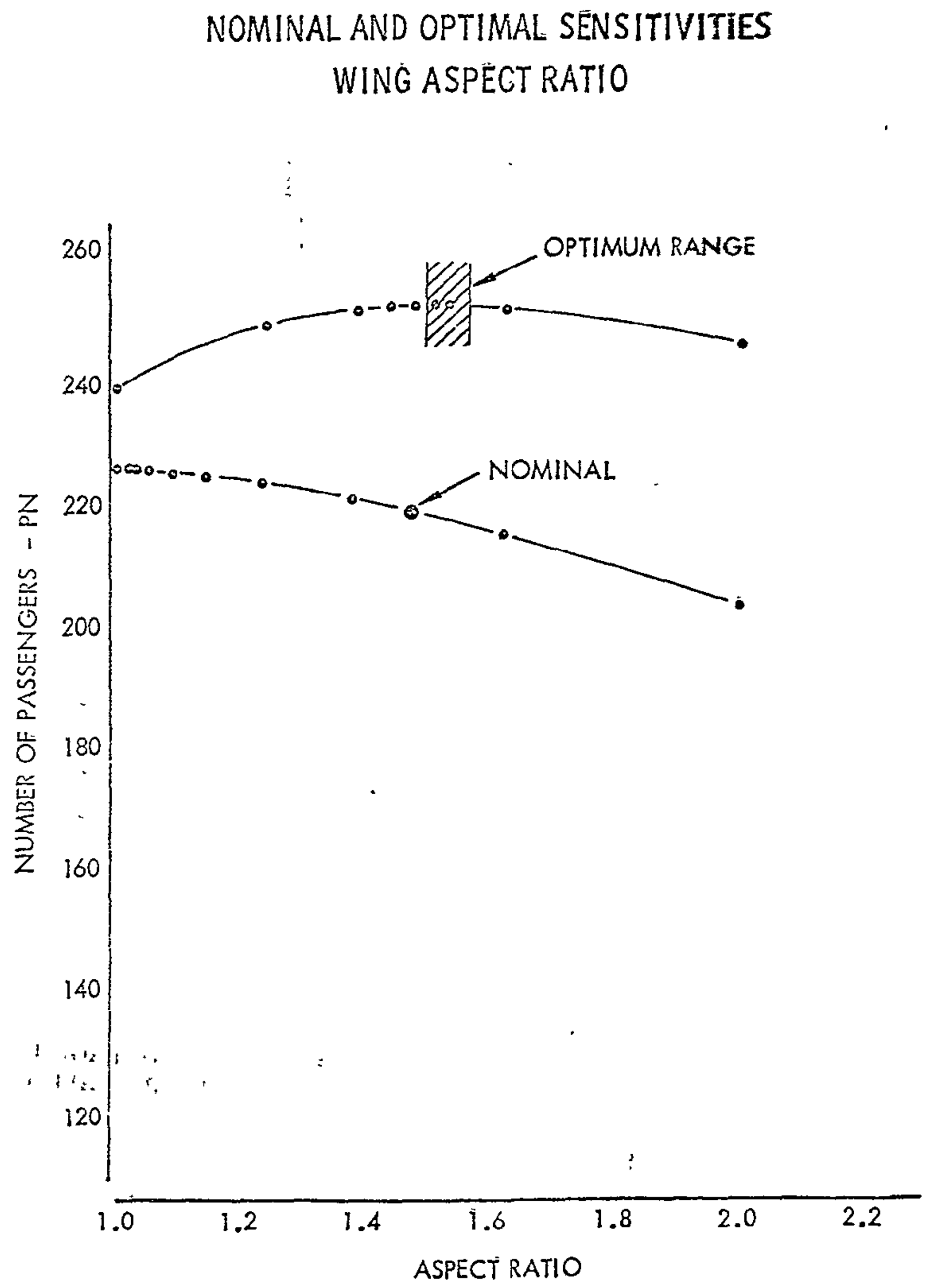
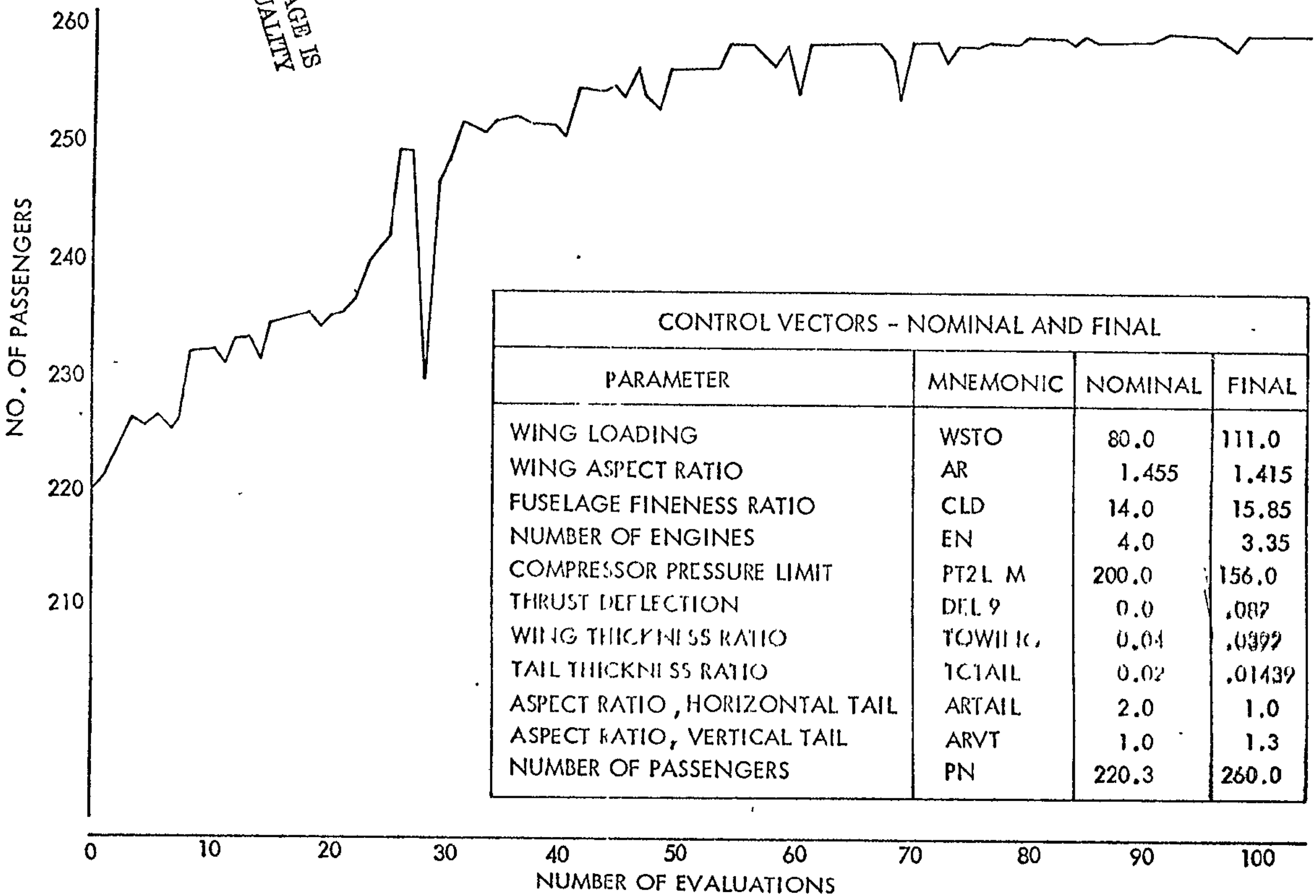


FIGURE 10 2-12 NOMINAL AND OPTIMAL SENSITIVITIES
WING ASPECT RATIO

FIGURE 10.2-13.

PAYOFF FUNCTION CONVERGENCE

TEN VARIABLES - ADAPTIVE SEARCH



CONTROL VECTORS - NOMINAL AND FINAL			
PARAMETER	MNEMONIC	NOMINAL	FINAL
WING LOADING	WSTO	80.0	111.0
WING ASPECT RATIO	AR	1.455	1.415
FUSELAGE FINENESS RATIO	CLD	14.0	15.85
NUMBER OF ENGINES	EN	4.0	3.35
COMPRESSOR PRESSURE LIMIT	PT2LIM	200.0	156.0
THRUST DEFLECTION	DEL 9	0.0	.087
WING THICKNESS RATIO	TCWING	0.04	.0399
TAIL THICKNESS RATIO	TCTAIL	0.02	.01439
ASPECT RATIO, HORIZONTAL TAIL	ARTAIL	2.0	1.0
ASPECT RATIO, VERTICAL TAIL	ARVT	1.0	1.3
NUMBER OF PASSENGERS	PN	220.3	260.0

FIGURE 10.2-14.

UNCONSTRAINED OPTIMAL HYPERSONIC CRUISE VEHICLE FLIGHT PATH $0.4 \leq M \leq 6.0$

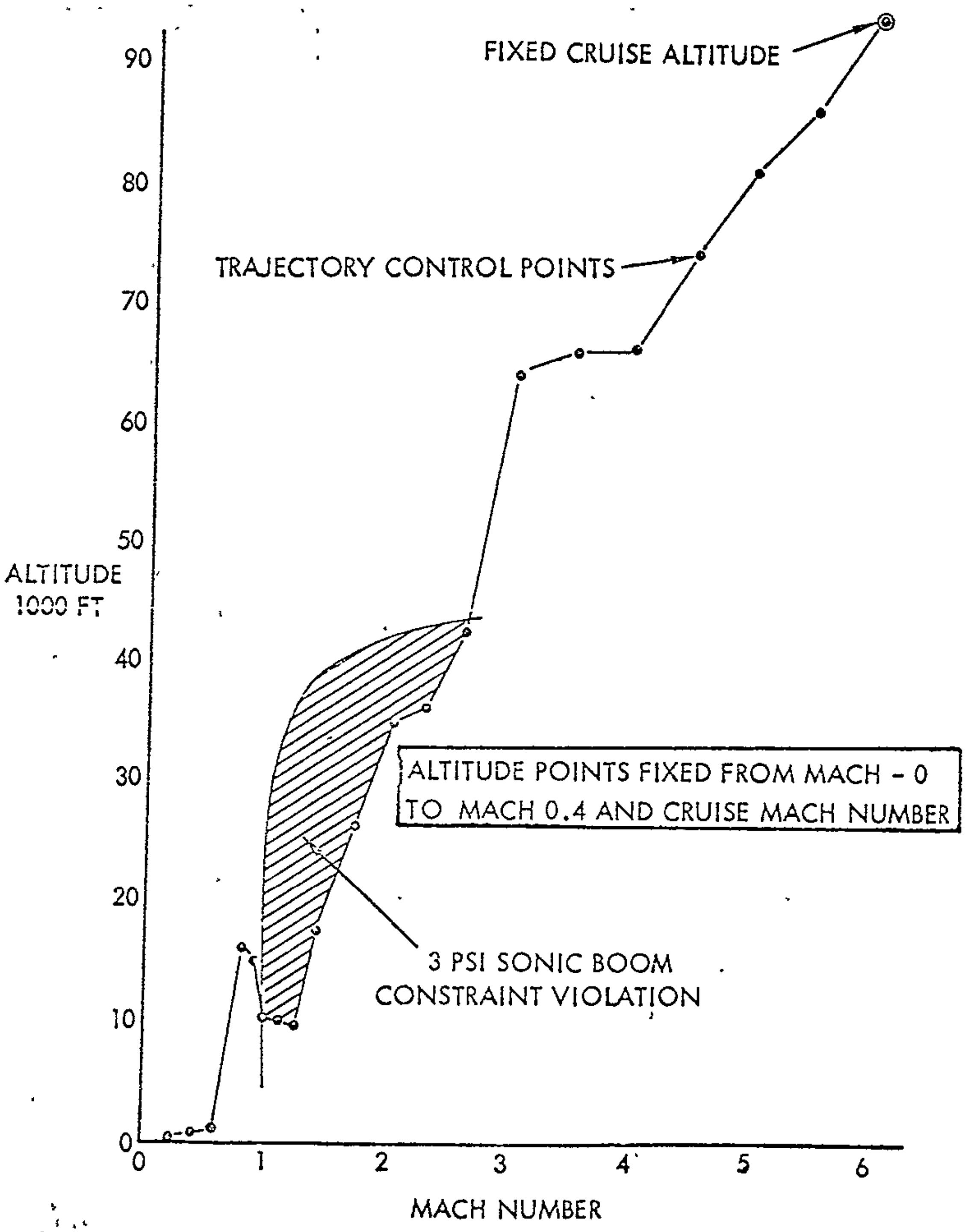


FIGURE 10.2-14 UNCONSTRAINED OPTIMAL HYPERSONIC CRUISE VEHICLE FLIGHT PATH $0.4 \leq M \leq 6.0$

RESULTS OF MAJOR OPTIMIZATION CALCULATIONS

DESCRIPTION	NUMBER OF PASSENGERS
NOMINAL SUPPLIED BY AMES RESEARCH CENTER	220.3
FIVE DESIGN-VARIABLE SOLUTIONS	
• ADAPTIVE CREEPING SEARCH	
FROM AMES NOMINAL	253.3
FROM UPPER BOUNDARY	253.4
• STEEPEST DESCENT SEARCH (EMPIRICAL METRIC)	252.0
• SECTIONING SEARCH	253.1
• QUADRATIC SEARCH	253.3
TEN DESIGN-VARIABLE SOLUTIONS	
• ADAPTIVE CREEPING SEARCH	
FROM AMES NOMINAL	260.0
FROM UPPER BOUNDARY	260.8
• STEEPEST-DESCENT SEARCH	257.0
TRAJECTORY OPTIMIZATION	
• ELEVEN VARIABLE WITH SONIC BOOM AND MONOTONIC HEIGHT CONSTRAINT	220.4
• SEVENTEEN VARIABLE UNCONSTRAINED	229.1
COMBINED DESIGN AND TRAJECTORY OPTIMIZATION	
• UNCONSTRAINED	286.1
• SONIC BOOM CONSTRAINT IMPOSED	
FROM AMES NOMINAL	263.8
FROM UPPER BOUNDARY	265.2
• SONIC BOOM PLUS TAKE-OFF CONSTRAINT	246.8
• SONIC BOOM, TAKE-OFF AND LANDING CONSTRAINT	222.5

FIGURE 10.2-15 RESULTS OF MAJOR OPTIMIZATION CALCULATIONS

SECTION 11

PRECOMPILER TECHNIQUES

The ODIN system contains a generalized precompiler program, MACRO FORTRAN. This string processor allows the user to construct his own programming language, for example, extended FORTRAN. The MACRO FORTRAN program was obtained under Aerophysics Research Corporation funds from The Boeing Company.

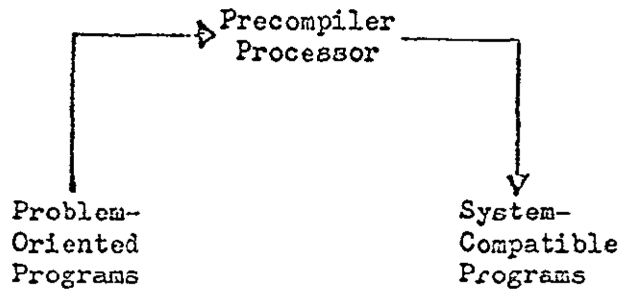
TABLE OF CONTENTS FOR SECTION 11.1, MACRO FORTRAN PRECOMPILER

11.1 MACRO FORTRAN PRECOMPILER

11.1.1 General Information

11.1.1.1 Precompilation.

Precompilation is a process by which a source program is examined and transformed, by means of prescribed algorithms, into a resultant source program. Normally, a precompiler accepts input programs in which the problem solver is able to state procedures in a concise, problem-oriented manner. The resultant program is then a language acceptable to an operating system compiler.



Several broad areas of computing to which precompilation may be applied are 1) Creation of special-purpose languages for programmers and engineers, e.g., character manipulation or plotting languages; 2) Enrichment of existing languages, e.g., FORTRAN or COBOL; 3) Simulation of the languages of other computers; 4) Creation of control languages which provide a convenient means of linking existing software routines together to perform some computing task.

11.1.1.2 MAC Software.

In order to assist programmers in developing special-purpose precompilers so that they may be used as additional computing tools whenever applicable, the following software has been implemented:

1. The basic framework for any precompiler, e.g., I/O provisions, operating system interfaces, diagnostic facilities, character manipulation routines, etc.

   All that remains to be done to make this a complete precompiler oriented toward a specific task is to supply the transformation algorithms for the problem-oriented statements desired and attach these to the framework. The result will be a complete application-oriented precompiler.

2. A language processor which makes the task of coding transformation algorithms a relatively simple procedure.

   The linkage of the coded algorithms to the precompiler framework is automatic when this language is used. This language is called the MAC language and it includes all FORTRAN IV statements plus a set of statements for identifying problem-oriented statements, manipulating character strings, and communicating with the precompiler framework.

11.1.1.3 <u>MAC Precompilers</u>.

To develop a special-purpose precompiler once the necessary transformations have been formulated, the programmer need only code these transformations as individual MAC language subprograms and supply these to the computer with appropriate control information. The precompiler produced may be used to preprocess programs immediately or it may be saved for use on subsequent computer runs.

A complete MAC language program defining a precompiler will consist of several subprograms, called macro block subprograms, and one main program, called a control block. Each subprogram will normally be devoted to identifying one specific kind of problem-oriented statement, determining which variation of that statement is currently being processed, and constructing new statements for inclusion in the transformed program. In a MAC language main program the programmer simply specifies, in a prescribed format, the names of all macro block subprograms which are to be a part of the user precompiler being constructed. After being processed by MAC, a control block main program becomes the interface between the operating system, the precompiler framework, and the processed macro block subprograms.

It may be noted that all precompilers created under this system, regardless of their intended application, are built on the same basic precompiler framework. Any subprogram defining a problem-oriented statement may be attached to any precompiler. In this sense precompiler designers have, under the MAC system, the facility for exchanging worthwhile ideas with little or no re-programming.

11.1.1.4 Coding Conventions.

Since the MAC language is an extension of FORTRAN IV, the conventions are the same as for FORTRAN IV. The programmer can manipulate variables, type names, assign common blocks, etc., in almost all cases. The exceptions are noted in the manual.

Input is 80-column card images that are put into complete statement form before analysis, i.e., columns 1-72 of first card of statement plus columns 7-72 of all continuation cards. Columns 73-80 of all cards are lost during the precompiler generation.

All FORTRAN comment cards are sent directly to the transformed program file without analysis.

11.1.1.5 Restrictions.

1) A MAC statement should not be used to end a DO loop and should not appear at the right side of a logical IF.

2) Variable names J0000 through J9999, R0000 through R9999, and any name beginning with the combination QX are reserved for use by the MAC system.

3) Statement labels should not exceed 89999. Labels above this are reserved for use by the MAC system.

11.1.1.6 Internal Data Format.

A program must have some way of identifying stored data so that it can be manipulated. In FORTRAN, of course, we use variables and arrays. All data which is to be identified and manipulated by "pure" (non-FORTRAN) MAC statements must be in MAC strings.

Each string is identified by an integer length variable and a name, the value of the length variable being the number of usable characters in the string starting with the character referred to by the name. Normally, MAC does not type string names as they are manipulated via subroutines.

Because the strings are stored in arrays of fixed words and the length variables are FORTRAN variables, it is possible for the programmer to manipulate these words via FORTRAN statements.

ORIGINAL PAGE IS
OF POOR QUALITY

It is the programmer's responsibility to not violate the MAC format. One of the big advantages of strings is their machine independence. Since word size is machine dependent, the programmer should carefully label all FORTRAN manipulations of string words to allow easy conversion to other machines. It is recommended that the equivalent MAC statements be included as comments.

11.1.1.7 String References.

To allow MAC to identify string names in MAC statements, string names are delimited by periods (.) unless otherwise declared by the programmer. If S is a string

.S.	references the entire string.
.S(I,J).	references the partial string of characters I through J of S.
.S((K)).	references the Kth character string of S.
.S=I=.	references the Ith substring of S if S is in partitioned form. (Section 3.7).

11.1.1.8 Explanation of Symbols.

The following symbol conventions are used in the following general statement forms (Section 1.9):

XXXXX — A MAC language word that must be written exactly as given.

XXXXX — A programmer-defined or FORTRAN language word.

name — any legal name in the FORTRAN sense excepting J0000 through J9999, R0000 through R9999, and any name beginning with QX.

nameD — any name followed by optional dimension information, e.g., XYZ or ST ((34)).

β — an optional 'blank-forcing' character

In all of the MAC statements actual blanks are ignored. If a coder desires to specify meaningful blanks as part of some literal text in a MAC statement, he may do so by placing some character in the position in that statement and then using it to represent the character blank in a literal within that statement. The coder should be careful to select a character for this function that is not being used for anything else within that MAC statement.

.EPCS.	any one of the four forms (Entire, Character, Partial, or Substring)
ν	any positive or negative statement number A positive number points to the statement which is to receive control if the desired test is successful. If the test is unsuccessful control passes to the next sequential statement. In the case of a negative number, the alternatives are reversed.
\$d\$	A character position designator (card column designator if the character string is a statement image) A designator of this type may be any integer constant, variable, or expression surrounded by delimiters. The delimiter will be the dollar sign (\$) unless some other character is declared for this purpose by the coder.
prototype	A MAC language description of a character string or statement that is to be identified. A prototype may contain any combination of identification text, entire string names, and position designators provided no two string names appear adjacent to one another. Identification is made on the basis of position designators and identification text alone. String names are the names of character strings to be appropriately filled if the identification match is successful.
TPF	Transformed program file - the file of transformed statements.
string expression	any combination of literal text, string references (any type), and position designators.
.E.	an entire string name
.EPCSL.	any one of the four string reference forms or a literal

A description of the individual MAC statements will now be given along with examples of their use.

11.1.1.9 General Statement Forms.

In general, the MAC statements are as follows:

CONTROL BLOCK name

CONTROL BLOCK OVERLAY name

USE name

MACRO BLOCK name

IMAGE (int var , name)

STRING (int var_1 , $nameD_1$) ,..., (int var_n , name D_n)

IDENTIFYβ.EPCS.($\mathcal{V}$)\$d\$ prototype

1β \$d\$ string expression

*β \$d\$ string expression

\$$\beta$ \$d\$ string expression

BUILDβ.E. \$d\$ string expression

SUBSTRINGβ.EPCS. INTO .E. ON .EPCSL.

COMPAREβ.EPCS. ($\mathcal{V}$) .EPCSL.

MOVE int expression FROM .E_1. \$d\$ INTO .E_2. \$$d_2$\$

MOVE .P. $\left\{\begin{matrix}\text{TO}\\\text{INTO}\end{matrix}\right\}$ .P.

COMPRESS .E.

COMPRESS .E. ALL BUT integer

CONVERT int expression to .E.

CONVERT .E. TO integer variable

DEFINE $\left\{\begin{matrix}\text{INTEGER}\\\text{REAL}\\\text{LABEL}\end{matrix}\right\}$ .EP.

ORIGIN $\left\{\begin{matrix}\text{INTEGER}\\\text{REAL}\\\text{LABEL}\end{matrix}\right\}$ integer expression

WARNINGS\$ any appropriate diagnostic note

ERRORS\$ any appropriate diagnostic note

ABORTS\$ any appropriate diagnostic note

WARNINGSβ	\$d\$ string expression	
ERRORSβ	\$d\$ string expression	
ABORTSβ	\$d\$ string expression	
LEVEL	CANCEL RESTORE	
LINK TO	name	
DELIMIT	STRING COLUMN	any one character (excluding blank)
SWITCH	FIND TYPE	integer variable
REINIT	MAC	
END		
FORMATβ	(format statement)	
(PRINT)β	\$d\$ string expression	
(READ)	\$d1,d2\$ .E1. \$d3,d4\$ .E2. etc.	
SET (PRINT) UNIT		integer expression
SET (READ) UNIT		integer expression

Symbols such as β, $\not b$, \$d\$, etc., are explained in Section 1.8.

11.1.2 Main Program Statements

11.1.2.1 Introduction.

The main program initializes various switches, calls the I/O section to input a statement, and then turns control over to the various analysis subprograms. Upon return from each such subprogram, a check is made to determine if the statement was accepted and, if not, it is sent to the next analysis subprogram. If no more subprograms exist, the statement is sent to the transformed program file (TPF).

Only the statements in this chapter and comment cards may appear in the main program. A programmer can circumvent this by writing his own FORTRAN main program; however, this is not recommended because the main program interfaces with the system I/O.

11.1.2.2 Control Block
Control Block Overlay

The first statement of the main program, or control block, is the above plus a precompiler name and file names.

CONTROL BLOCK OVERLAY allows the programmer to specify the main program be made into a 6600 overlay. MAC will build a (0, 0) level and a (1, 0) level overlay having the name specified by the programmer. For precompilers not using the reinitialize feature (section 3.17), the only function of the (0, 0) level will be to call the (1, 0) level overlay.

11.1.2.3 Use.

Tells MAC to generate coding to transfer control to the named analysis subprogram.

11.1.2.4 End.

Signals the end of the main program.

11.1.2.5 <u>Example</u>.

```
CONTROL BLOCK OVERLAY NAME(F1,F2,...,FN)
USE           ALPHA1
USE           ALPHA2(I)
USE           ALPHA3
END
```

The files F1, F2,...,FN are best described by looking at the program card generated from the <u>CONTROL BLOCK</u> card.

```
PROGRAM  NAME(F1, NFIRST, L,F2, F3, F4, F5,..., FN,
    TAPE8, TAPE13, TAPE18, TAPE5=F2, TAPE6=F3,
    TAPE77=F1)
```

The program name becomes NAME and as such is the name of the pre-compiler being built by the user. The files declared on the program card are described in the table on the next page.

<u>Note</u>: The following file names should not be declared by a user:

INPUT
OUTPUT
QXCON
TAPE 8
TAPE 13
TAPE 77
L
NFIRST
REINIT
TAPE 18

11.1.3 Analysis Block Statements

11.1.3.1 Introduction.

Analysis or macro blocks are written to analyze input statements and, if necessary, convert them to some output language. It is in these blocks that the precompiler language is defined since only in those blocks will the programmer be able to "see" input and generate output. All FORTRAN IV statements are permissible in these blocks.

It is recommended that each block be written to accept only one type of input statement. These modules can then be used by any other MAC precompiler and the precompiler and language are easily modified.

11.1.3.2 Macro Block.

The first statement in a macro block subprogram should be of the form

MACRO BLOCK name

This is similar in function to the SUBROUTINE statement in a FORTRAN subprogram but a MAC language subprogram does not receive arguments through a calling sequence. (A calling sequence can be included if desired.)

"MACRO BLOCK name" causes MAC to generate at least the following sequence of FORTRAN statements

```
SUBROUTINE name
COMMON/QXBNAM/QXNAME
DATA QXNAM/n H name/
QXNAME=QXNAM
```

These last three statements can be used as a debugging aid since, if the programmer asks for a dump of that one common block whenever his program has an abnormal halt, he can quickly determine which macro block was being executed. After checkout of macro blocks, the statement "MACRO BLOCK name" can be changed to "SUBROUTINE name" to eliminate compiler diagnostics due to data appearing after "QXNAME = QXNAM".

11.1.3.3 String Image.

All MAC variables (strings) must be defined with either of these two statements. The string declarations consist of ordered pairs (I1, I2) with I1 the name of the length variable associated with the string name I2. I1 must be a FORTRAN integer variable or else typed integer. I2 does not have a type unless the programmer manipulates it via FORTRAN statements.

The maximum string length can be declared or implied. The statement

STRING (LA, A((N)))

will reserve N character positions for the string A. N cannot exceed 1326 and for N = 1326 the programmer need not specify N, i.e.,

STRING (LA, A((1326))) $\equiv$ STRING (LA, A) .

All input statements appear in a special string of maximum size. The programmer designates the name of this string by the IMAGE statement. The particular name used is local to each block, but the actual area reserved in common to all blocks having an IMAGE statement. Only one image string is declared in each macro block. The form of the IMAGE statement is

IMAGE (LA, A)

All strings and their length variables except image strings, can be put into common blocks by the programmer. As strings are manipulated in MAC statements, the length of the strings are automatically updated.

11.1.3.4 Identify.

This is one of the more powerful statements in MAC. The initial string is checked for the specified literal pattern and if the pattern is found, the remaining characters are placed in the named strings. The form of the statement is

IDENTIFY $\not b$ .S. ($\mathcal{V}$) prototype

where .S. is an EPCS string, $\mathcal{V}$ is a statement label, and prototype is any combination of literals, column delimiters, and strings that satisfy Rule 3 below. ($-m$) means transfer to m if unsuccessful, ($+m$) or (m) means transfer to m if successful. In general, the prototype will begin with a column delimiter and have string names separated by literals or column delimiters.

Three rules must be observed when coding this statement:

1) leading or interspersed blanks are ignored, but trailing ones are not;

2) any successful IDENTIFY statement sets the FIND switch (Section 3.18) on;

3) two string names cannot be immediately adjacent.

Rule 1 does not cause problems if the IDENTIFY statement always terminates with a string since then all trailing characters are put in this string.

Rule 2 means that the programmer does not have to eliminate the image string to prevent it from appearing in the TPF. It also means the programmer may lose an image string if an IDENTIFY is successful on any string.

Rule 3 is fairly obvious since if two strings are adjacent, it is impossible to determine how many characters are to go into each string. If column delimiters or literals are used between the strings, no diagnostic will appear.

As the following examples show, this statement is particularly useful in breaking input statements into component strings.

Example 1 - Suppose the problem-oriented statement being considered is of the form

PLOT ON opt_1/ opt_2/ opt_3

where opt_1 is the name of the plotting device to be used, opt_2 is the type of plot desired, and opt_3 is a list of the arrays containing the data points to be plotted. Then a macro block subprogram to identify such a statement and isolate the optional information for further analysis could be written

```
    MACRO BLOCK XPLOT
    IMAGE (LS, S)
    STRING (LL,L((6))),(LA,A((24))),(LB,B((24))),(LC,C)
    IDENTIFY .S. (-100) $1$ .L. $7$ PLOT ON.A./.B./.C.
       .
       .
       .      (further analysis)
       .
       .
100 RETURN
    END
```

If the image of the statement being processed contained

bbb42bbbPLOTbbONbDEVICEbA/bLINEAR/bX,bY,bXX,bYY

then identification would be positive and strings L,A,B, and C would be adjusted to the following:

L would be filled with LL would be set to six	bbb42b
A would be filled with LA would be set to nine	bDEVICEbA
B would be filled with LB would be set to seven	bLINEAR
C would be filled with LC would be set to thirteen	bX,bY,bXX,bYY

Control would then pass to the next sequential statement.

Because of the way the IDENTIFY statement is written in this example, blanks in the image statement do not affect the test for identification. If the originator of this problem-oriented statement had desired, he could have specified that the words PLOT ON had to be separated by at least one blank. The IDENTIFY line for this would be

IDENTIFY' .S. (-100) 1.L.7 PLOT * ON.A./.B./.C.

In this case the character * is treated as a meaningful blank within the identification text.

Example 2 - IDENTIFY .ABC. (+80) 7 CLOSE PLOT

This line says to look into string ABC beginning at character 7 for the pattern CLOSEPLOT. Leading or intervening blanks should be ignored. ANY CHARACTERS following the T will cause the identification test to fail, e.g., the patterns

CLOSE PLOTTER
or CLOSE PLOT X
or CLOSEbPLOTb

would be rejected by the identify test. Acceptable patterns might be

CLOSEPLOT
or CLOSE PLOT
or C L O S E   P L O T

Example 3 - IDENTIFY 9 .T=K= . (16) 1 RETRIEVE 9 DATA .CODE.

This line says to look into the Kth substring of T for the words RETRIEVE and DATA. These words must be separated by at least one blank, as signified by the character 9. If found, properly update string CODE with the remaining characters of the substring and transfer control to statement 16.

Example 4 - IDENTIFY* .R(J,J+6).(-60) 1*******

This line says to transfer control to statement 60 if the referenced seven characters are not all blanks.

Example 5 - IDENTIFY .S. (-10) 1 .A. 7 .B. 30 .C.

10 CONTINUE

This IDENTIFY will always be successful. Upon completion, .A. = .S(1,6) . ; .B. = .S(7,29) . , .C = .S(30,LS) . where LS is the length variable for .S. .

NOTE: Immediately following the IDENTIFY line in a macro block, it is generally a good practice to remove the blanks from those individual character strings in which blanks are not significant. This technique will simplify the subsequent analysis to be performed on such strings. Removal of blanks may be accomplished conveniently by using the COMPRESS statement which will be described in Section 3.10.

11.1.3.5 *, 1, $

The *, 1, and $ statements assist the MAC language coder in constructing statements and cause these to be included in the transformed program. The meaning of the three symbols is as follows.

* Include this line in the transformed program in place.

1 Include this line in the transformed program at the top.

The transformed program will be rearranged so that all lines of this type appear first. This is useful for including type statements, common statements, etc., in the program.

C-1

NOTE: Immediately before rearranging the transformed program, the first line in that program is checked to determine whether it is a SUBROUTINE, FUNCTION, or BLOCK DATA statement. If it is, that line will remain first in the rearranged program.

$ Include this line in the transformed program in place, but flag it as a statement requiring further analysis.

If any flagged statements are present in the transformed program after the initial precompilation pass, another pass will be made so that those flagged lines can be sent through the macro blocks for analysis as possible problem-oriented statements. If this analysis produces new flagged lines, another precompilation pass will be made, and so forth. It should be noted that extra precompilation passes require extra computer time; however, the additional time is not much greater than needed to process the same number of input statements as there are $ statements.

Example 1 - *7 CALL .A. (.S(I,I+5).,1H.T((J)).)

The included line would contain six blanks in columns one through six followed by a concatenation of the following:

The characters CALL
The characters in string A
A left parenthesis
The six characters, I through I+5, of string S
The characters 1H
The Jth character of string T
A right parenthesis

Example 2 - */ 1 .L. 7 GO / TO / .LOC.

The included line would contain the characters from string L in columns one through six. Then, beginning in column seven would be

GObTOb

followed by the contents of string LOC.

Example 3 - * 7 10 .S=K= . 77 .R=L=.

The included line would contain blanks in columns one through nine. This would be followed by the characters from the Kth substring of S. This would finally be followed by three blanks and the characters from the Lth substring of R.

11.1.3.6 Build

The BUILD statement is similar to the *, 1, and $ statements already discussed except that the string which is constructed in this case is not included in the transformed program. Instead, it is stored in a specified string within the macro block subprogram. As an example, consider the following:

BUILD..X. J CALL .A. (.B(13,18).)

This line says to build into string X the following:

Blanks in columns one through J-1,
the characters CALL,
the characters from string A, then a (,
the characters 13-18 from string B,
and finally a).

The former contents of string X are erased.

11.1.3.7 Substring.

The purpose of the SUBSTRING statement is to separate a given pattern of characters into a set of substrings. These substrings may then be referenced individually by using the bracketed subscript form. As an example, consider the following:

Assume that string A contains the following character pattern:

XYAXYBXYC

Then the statements

SUBSTRING .A. INTO .B. ON Y
SUBSTRING .A. INTO .C. ON XY

would produce the 'partitioned' strings B and C containing these substrings

B (4 substrings)	C (4 substrings)
X	empty
AX	A
BX	B
C	C

In this example the associated length cells of strings B and C would each be set to four. The length cells of partitioned strings will contain the number of substrings present rather than the number of characters present.

As another example, assume string X contains

A,B(1,J),C

then the statements

```
LEVEL CANCEL
SUBSTRING .X. INTO .W. ON ,
LEVEL RESTORE
SUBSTRING .X. INTO .R. ON ,
```

would produce

W (4 substrings)	R (3 substrings)
A	A
B(1	B(1,J)
J)	C
C	

This illustrates the effect of the LEVEL statement. This statement is meaningful only in conjunction with the SUBSTRING statement. When the level is in a 'restored' condition, the only separators valid are those which are not enclosed within parentheses (i.e., those which are at zero parenthesis level). When the level is in a 'cancelled' status, all separators are valid.

The level setting for each macro block is independent of the settings for the other blocks. The 'restored' or 'on' status is assumed for each block at the beginning of precompilation. Once altered, however, level settings are not reset automatically. Thus, if control is given up by a macro block at a time when the level is in a cancelled status, this status will be retained at the next entry to that block.

11.1.3.8 Compare.

The COMPARE statement is for testing two character patterns for equality. In order for two patterns to be equal they must be identical in all respects including number of characters. For example, suppose

```
    string .A.          contains ANGLEbOFbATTACK
and substring .B=3=.    contains ANGLEOFATTACK
```

then the line

```
COMPARE .A. (-20) .B=3=.
```

would cause control to go to statement 20 due to the inequality of the two patterns involved. However, the line

```
COMPARE* .A. (-20) ANGLE*OF*ATTACK
```

would produce a valid equality and the next sequential statement would receive control.

11.1.3.9 Move.

The MOVE statement provides for movement of characters within strings or from one string to another. For example, the line

```
MOVE J-1 FROM .A. $7$ INTO .B. $K$
```

is interpreted as "move J-1 characters from string A beginning with character seven into string B beginning at character position "K". If strings A and B contained the respective patterns

```
    1234567890
and ABCDEFGHIJKLMNOP
```

and J-1 and K were two and six respectively, then the final pattern in B would be

```
ABCDE78HIJKLMNOP
```

and the associated length of string B would be unchanged.

As an example of joining two strings with the MOVE statement is the following:

```
STRING (LR,R), (NX,X)
    .
    .
    .
MOVE LR FROM .R. $1$ INTO .X. $NX+1$
    .
    .
    .
END
```

In this example, all of the characters in string R are moved into string X following the characters which were previously there. After execution of this statement, NX is properly adjusted to the new length of string X.

An alternate form of the MOVE statement is the following:

MOVE .C. $\frac{\text{TO}}{\text{INTO}}$ .C.

This form is best explained by the following example:

MOVE .A(I,J). TO .B(K,L).

case a) (J-I + 1) ≥ (L-K + 1)

moves L-K + 1 characters of string A beginning at position I into string B starting at position K.

case b) (J-I + 1) < (L-K + 1)

moves J-I + 1 characters of string A beginning at position I into string B starting at position K. In addition, (L-K) - (J-I) blanks are moved into string B starting at position K + (J-I + 1)

11.1.3.10 Compress.

The COMPRESS statement removes blanks from an entire string and appropriately decreases its associated length. If, for example, string A had length seven and contained LINbLOG

COMPRESS .A.

would reduce the length to six and string A would then contain LINLOG.

A second form of the COMPRESS statement removes all but a specified number of consecutive blanks from a string. Suppose string X had a length of 10 and contained PLOTbbbbON.

COMPRESS .X. ALL BUT 1

would reduce the length to seven and X would then contain PLOTbON. Similarly,

COMPRESS .X. ALL BUT 3

would alter string X to the form PLOTbbbON with a length of nine. If the integer following BUT is greater than the number of consecutive blanks no compression is done. In the case where the integer is zero, this second form reduces to the first form of the compress statement.

11.1.3.11 Convert.

The CONVERT statement provides a convenient neams for converting FORTRAN integers into equivalent character strings and character strings of decimal integers into FORTRAN integers. For example, if J had the value 57 and K had the value 72,

```
CONVERT J+K-1 TO .V.
```

would produce the three characters 128 in string V. The statement

```
CONVERT .V. TO I
```

would assign the value 128 to the variable I.

11.1.3.12. Define Origin.

The DEFINE statement enables the MAC language coder to 'make-up' unique statement labels, integer variable names, and real variable names so that they may be used as component parts of statements to be included in the transformed program.

For example, the lines

```
   STRING (LA,A),(LB,B),(LC,C),(LD,D)
       .
       .
   DEFINE REAL .A.
   DEFINE INTEGER .B.
   DO I = 1,3
   DEFINE LABEL .C(I*8-7,I*8).
10 CONTINUE
   DEFINE REAL .D(15,19).
       .
       .
```

would result in strings A,B,C, and D being filled as follows:

A	R0001
LA	would have the value five
B	J0001
LB	would have the value five

C 90001bbb90002bbb90003bbb
LC would have the value 24 or its previous value, whichever is greater

D ppppppppppppppR0002 (p represents previous character)
LD would have the value 19 or its previous value, whichever is greater

Each successive request, regardless of which macro block the request is from, results in the next available item of its type being defined and stored appropriately.

Storing of the requested item is done differently depending on whether the receiving string is in entire or partial form. If the entire form is used, the former string contents are erased and the length is adjusted to five. If the partial form is used, the requested item is moved into the beginning of the partial string and blanks are used to fill out the rest of the partial string field. The length is adjusted only if the string is lengthened in this case.

The ORIGIN statement enables the MAC language coder to preset the initial value of the defined labels or variables to values other than x0001. For example, if I has the value 50 and J the value 17, the statements

ORIGIN LABEL	I-2*J
ORIGIN REAL	32
ORIGIN INTEGER	J+62

will cause the first label generated to be 90016, the first variable generated to be R0032, and the first integer generated to be J0079.

11.1.3.13 WARNING$
ERROR$
ABORT$

WARNINGS$
ERRORS$
ABORT$

These statements permit the macro block coder to provide appropriate diagnostics for users of his precompiler. The "plural" statements are included to allow the programmer to output dynamic messages, i.e., to be written as a *, 1, or $ line. The proper statement to be used when a mistake or ambiguity is detected in a problem-oriented statement depends upon the degree of seriousness as follows:

WARNING$ WARNINGS$	note...	This should be used if the mistake found is not too serious and precompilation can continue provided certain assumptions are made about what the user meant.
ERROR$ ERRORS$	note...	This is for more serious errors in which the macro block coder does not feel it is safe to 'guess' what is meant. Precompilation will continue so that anyother errors in the user program may be detected. No transformed program will be produced.
ABORT$ ABORTS$	note...	After this type of diagnostic note is printed for the user, his precompilation is aborted. No transformed program is produced.

Normally, the most desirable statement to use is ERROR$ or ERRORS$ since it will not allow an erroneous transformed program to be built but it will allow precompilation to continue so that any additional errors may be detected.

An example use of the diagnostic facility is as follows:

```
    IDENTIFY .S. (-100) $7$ ANALYZE .X. STRUCTURE
    COMPRESS .X.
    COMPARE .X. (30) WING
    COMPARE .X. (40) TAIL
    COMPARE .X. (50) FUSELAGE
    ERROR$ WING, TAIL, OR FUSELAGE OPTION MISSING
   xOR MISSPELLED
    RETURN
 30 CONTINUE
          (include wing analysis lines in the transformed program)
 40 CONTINUE
          (include tail analysis lines in the transformed program)
 50 CONTINUE
          (include fuselage analysis lines in the transformed program)
100 RETURN
```

11.1.3.14 Level.

The two forms of this statement, LEVEL CANCEL and LEVEL RESTORE have already been discussed in conjunction with the SUBSTRING statement (Section 3.7).

ORIGINAL PAGE IS
OF POOR QUALITY

11.1.3.15 LINK TO.

Just as one FORTRAN subroutine may use another via CALL, one macro block may use another via LINK TO. An example of the use of this statement might be

```
   MACRO BLOCK COMP
   IMAGE (LS,S)
   STRING (LW,W),(LX,X)
      .
      .
      .
C  SAVE THE CURRENT IMAGE IN STRING X
   BUILD .X. $1$ .S.
C  SET UP IMAGE WITH NEW STATEMENT
   BUILD .S. $7$ ANALYZE .W.
C  TRANSFER TO THE BLOCK WHICH CAN
C  TRANSFORM A STATEMENT OF THIS TYPE
   LINK TO XBLK
C  RESTORE THE IMAGE
   BUILD .S. $1$ .X.
      .
      .
      .
   END
```

The above task could also have been accomplished requiring an extra precompilation pass by using

```
   $ $7$ ANALYZE .W.
```

instead of the link technique. In this case the ANALYZE statement would have been included in the transformed program during the first precompilation pass. Then a second pass would occur, this time using the transformed program as data, so that any lines of this type ($) could be properly processed. On this pass the ANALYZE line would be identified and transformed by macro block XBLK.

11.1.3.16 DELIMIT.

This statement allows programmers to declare any non-blank character to be a string or column delimiter. By using different delimiters, the period (.) or dollar sign ($) may be freed for use as normal text characters. For example,

```
DELIMIT STRING +
COMPARE +A+ (20) 12E9.4
DELIMIT COLUMN /
° /7/ FORMAT (E12.4 ,5S3A B C$)
DELIMIT STRING .
DELIMIT COLUMN $
        .
        .
        .
```

(back to standard delimiters again)

NOTE: DELIMIT is a 'pseudo-statement', not an executable statement. DELIMIT affects all statements following it (within one subprogram) until another DELIMIT is encountered.

11.1.3.17 REINIT MAC.

Occasionally, a programmer is unable to build his precompiler in one run because the control card buffer cannot hold all the required cards. By using the REINIT MAC statement to signal the end of the separate blocks, the programmer can batch the precompiler through MAC with one MAC control card. MAC will treat a REINIT MAC card as though it had read a 7-8-9 card and then continue on to process the next block.

MAC-built precompilers can have the above feature by using the CONTROL BLOCK OVERLAY (Section 2.2). The programmer must identify his own "reinit" statement. The macro block where this is done must also, upon successful identification, set a cell to allow the auxilliary routines to reinitialize correctly. This is done by including a labeled common

```
COMMON/QXMACH/I1, I2
```

in the macro block, and setting I2 to non-zero when the "reinit" card is found. It is recommended that the "reinit" statement be a non-FORTRAN statement.

11.1.3.18 FORMAT.

MAC will accept FORMAT statements with implied Hollerith counts and convert them to FORTRAN FORMAT statements. The Hollerith data is delimited by a programmer-designated character that directly follows FORMAT. The form is

```
FORMATβ ( Hollerith data   conversion specs.   Hollerith
data  conv. specs. etc.)
```

11.1.3.19 (PRINT)
(READ)
SET (PRINT) UNIT
SET (READ) UNIT

The programmer can read card images or write from the units specified. If SET (PRINT) UNIT is not used 6 is assumed, and if SET (READ) UNIT is not used 5 is assumed. Note that the unit can be changed during execution.

The (PRINT) outputs 120 characters per line with every line except the first starting with a blank. The (READ)inputs 80 charactor card images and puts the specified characters into the specified strings starting in character position 1 of all the strings. The character designators and string names are matched up in order of occurrence. The forms are

(PRINT) β	\$d\$ string expression
(READ)	\$d1\$,d2\$ .E1. \$d3,d4\$ .E2. etc.
SET (PRINT) UNIT	integer variable
SET (READ) UNIT	integer variable

11.1.3.20 SWITCH.

This statement gives the MAC language coder access to several internal switches or flag cells which are normally used only by the basic precompiler framework. Two forms of this statement are currently available.

SWITCH FIND integer variable$_1$

SWITCH TYPE integer variable$_2$

If either or both of these statements appear in a macro block subprogram, the declared integer variables will be properly equivalenced to the appropriate flag cells in the framework routines. For example, to equate the integer variable IJK to the internal FIND switch the following MAC language statement would be used.

SWITCH FIND IJK

The functions of these two switch cells will now be discussed.

The FIND SWITCH indicates whether or not a successful IDENTIFY has occurred within a macro block. Each time a source statement is placed in the image string so that it may be examined by the various macro blocks, the find switch is set to zero. The control block program will now transfer to each of the macro blocks in turn. Upon return from each macro block, the control block tests this cell to determine if it is still zero. If so, the process continues. If not, this indicates that the source line has been properly identified and does not need to be passed on to the remaining macro blocks. Any successful IDENTIFY test in a macro block will cause the find switch to be set non-zero. An unsuccessful test will not change the setting. This switch modification is done automatically by the routines which do the actual character testing. Thus, if a programmer obtains a successful IDENTIFY within a macro block, then decides, on the basis of certain analysis and testing, that the current image contents should be passed on to the remaining blocks, the programmer should set the declared integer variable back to zero before returning. (IJK = 0 for the above example)

Note: If a source statement is passed through all of the macro blocks and the find switch is still zero, that statement will be included in the transformed program unchanged.

The TYPE switch will contain the integer value 1, 2, 3, 4, or 5 depending on whether the conventions to be assumed (continuations, comments, etc., in the source program and the transformed program) are NON-STANDARD, FORTRAN, COBOL, SLEUTH, or ASCENT. The basic precompiler framework is set up to expect FORTRAN conventions (type 2) in the source and transformed programs as being the normal case. In view of this, most programmers will not be concerned with the type switch.

Suppose, however, that the ABC precompiler is to transform programs which use conventions other than those of FORTRAN.

Program X → ABC precompiler → Program Y

In this case, the first statement of program X must be one of the following:

NON STANDARD
COBOL
SLEUTH
IBMAP

A statement of this kind is a 'pseudo-statement' which causes the basic precompiler framework to change the type switch setting from its assumed value of 2 (for FORTRAN) to a new value of 1, 3, 4, or 5, respectively. This line is then discarded and is not passed through ABC's macro blocks. Each remaining line of program X will then be sent through the macro blocks as an 80 column card image rather than a statement image as is done if FORTRAN continuations, etc., can be assumed.

Although the above declarations properly set the type switch value, COBOL, SLEUTH and IBMAP conventions are currently treated as though they were NON STANDARD. Proper considerations for these conventions may be included in the basic framework at a later date. Thus, if a precompiler is being designed to preprocess programs using conventions other than those of FORTRAN, the MAC language programmer must provide for the treatment of continuations, comments, etc., himself.

Similarly, lines to be included in program Y are sent out as card images. If an output line is loss than 80 columns, the remainder is filled out appropriately with blanks.

Since the type switch is available for testing by the programmer, a macro block could be set up so that it identifies a given problem-oriented line, tests the type switch to determine which of several languages is being pre-processed, then outputs the appropriate transformed lines in the language, (e.g., FORTRAN, COBOL, etc.)

The programmer can change the input or output at any time during execution. This can be used to allow non-standard input with FORTRAN output, etc. The programmer should not change the switch indiscriminately. Intermediate results are in a MAC format and only TPF output appears in the specified format at the end of the precompilation.

11.1.4 Examples

11.1.4.1 Example 1.

Several example macro blocks will now be illustrated. These are, admittedly, quite simple and are intended only to show the complete coding of some elementary transforms.

Identify and transform a statement of the type

```
   42 REWIND TAPES 1, 3, KTAPE
```

into the following

```
   42 CONTINUE
      REWIND 1
      REWIND 3
      REWIND KTAPE
```

```
      MACRO BLOCK QTAPE
      IMAGE (LS,S)
      STRING (LL,L((6))),(LX,X),(LY,Y)
      IDENTIFY .S. (-10) $1$ .L. $7$ REWIND TAPES .X.
      * $1$ .L. CONTINUE
      SUBSTRING .X. INTO Y. ON  ,
      DO 5 I=1,LY
      * $7$ REWIND .Y=I=.
    5 CONTINUE
   10 RETURN
      END
```

11.1.4.2 Example 2.

Identify and transform a statement of the type

```
   16 READ INPUT TAPE JTAPE, FMT, A, B, C
```

into the following

```
   16 READ (JTAPE, FMT) A, B, C

      MACRO BLOCK RD
      IMAGE (LS,S)
      STRING, (LL,L((6))),(LT,T((12)))
      STRING  (LF,F((12))),(LLIST,LIST)
      IDENTIFY .S. (-20) $1$ .L. $7$ READ INPUT TAPE
     x .T., .F. , .LIST.
      * $1$ .L. $7$ READ ( .T. , .F. )  .LIST.
   20 RETURN
      END
```

11.1.4.3 Example 3.

Identify and transform statements of the form

```
   10 WRITE OUTPUT TAPE 6, 12
or 20 WRITE OUTPUT TAPE K, FMT, X, Y
```

into the following

```
   10 WRITE ( 6, 12)
or 20 WRITE ( K, FMT) S, Y respectively
```

```
      MACRO BLOCK WRT
      IMAGE (LS,S)
      STRING (LT,T((12))),(LX,S),(LF,F((12)))
      IDENTIFY .S. (-30)$7$ WRITE OUTPUT TAPE .T. , .X.
      IDENTIFY .X. (20) $1$ .F. , .X.
C     NO LIST, FORMAT ONLY
      * $1$ .S(1,6). $7$ WRITE (.T.,.X.)
      RETURN
20    CONTINUE
C     LIST PRESENT
      * $1$ .S(1,6). $7$ WRITE (.T.,.F.).X.
30    RETURN
      END
```

11.1.4.4 Example 4.

Consider a statement of the form

```
OUTPUT list
```

which will cause the specified list of variables to be printed in a standard format and also to be 'titled' so that it may be properly identified. The resultant output is similar to the NAMELIST output in FORTRAN IV. A statement of this type could be useful as a debugging tool for FORTRAN users or it could be used in lieu of the formatted WRITE statement by inexperienced programmers or by students.

Macro block XOUT would transform a program such as

```
      SUBROUTINE GAMMA
           .
           .
      OUTPUT A, B, CALC
           .
           .
   12 OUTPUT (C(1), 1=1, 10)
```

into the following:

```
          SUBROUTINE GAMMA
          DIMENSION J0001(2)
          DATA J0001/8HA,B,CALC/
          DIMENSION J0002(3)
          DATA J0002/13H(C(1),1=1,10)/
              .
              .
90001     FORMAT (1HO, 1P8E14.5)
90002     FORMAT (1HO, 11A10)
          WRITE (6,90002) J0001
          WRITE (6,90001) A,B,CALC
              .
              .
   12     WRITE (6,90002) J0002
          WRITE (6,90001) (C(1),1=1,10)
              .
              .
```

This transformed program is set up to output and appropriately title the desired items.

In addition to showing the complete transformation algorithm, macro block XOUT illustrates the use of ten of the MAC statements.

```
          MACRO BLOCK XOUT
          IMAGE (LS,S)
          STRING (LLIST,LIST),(LLA,LA((6))),(LLB,LB((6)))
          STRING (LD,D((12))),(LINT,INT((6)))
          DATA INIT/23/
          IDENTIFY .S. (-100) $7$ OUTPUT .LIST.
          COMPRESS .LIST.
          IF(INIT.EQ.0) GO TO 10
          INIT = 0
          DEFINE LABEL .LA.
          DEFINE LABEL .LB.
C         WATCH OUT FOR THE DECIMAL POINT IN THE FORMAT
          DELIMIT STRING +
          * $1$ +LB+ $7$ FORMAT(1HO,Ip8E14.5)
          * $1$ +LA+ $7$ FORMAT(1HO,19A6)
          DELIMIT STRING .
10        CONVERT (LLIST+5)/6 TO .D.
          DEFINE INTEGER .INT.
          1 + $7$ DIMENSION + .INT. (.D.)
          CONVERT LLIST TO .D.
          1 + $7$ DATA + .INT./.D.H.LIST./
          * $1$ .S(1,6) $7$ WRITE(6,.LA.) .INT.
          * $7$ WRITE (6,.LB.) .LIST.
100       RETURN
          END
```

11.1.4.5 Example 5.

This example provides a tool which makes the writing of complicated FORMAT statements less susceptible to coder and keypunch errors because of incorrect Hollerith counts. This is accomplished by Macro Block FMT which extends the flexibility of the existing FORMAT statement in that it allows a FORTRAN programmer to define a Hollerith delimiter for any FORMAT in which one is desired. The delimiter may be any available character, and it changes dynamically from FORMAT to FORMAT.

For example, the statements

```
10 FORMAT * (*1*, E12.4, *EXAMPLE FORMAT*, A6, *TEST*)
20 FORMAT $ ($ THIS ONE USES THE DOLLAR SIGN $)
30 FORMAT ALPHA (ALPHA COMPLEX PATTERN ALPHA)
```

would be transformed by the FMT algorithm into

```
10 FORMAT(1H1, E12.4, 14HEXAMPLE FORMAT, A6, 4HTEST)
20 FORMAT(31H THIS ONE USES THE DOLLAR SIGN )
30 FORMAT(17H COMPLEX PATTERN )
```

respectively.

Any formats which do not have a delimiter pattern before the initial left parenthesis will remain unchanged by block FMT.

Formats which have a delimiter pattern but do not use it within the body of the FORMAT will simply have the pattern removed by FMT.

If a delimiter pattern appears an odd number of times within the body of the FORMAT, an error message will be given stating

```
ALPHA DELIMITERS NOT PAIRED CORRECTLY
```

and no erroneous transformed program will be produced.

```
MACRO BLOCK FMT
IMAGE (LS,S)
STRING (LR,R),(LQ,Q),(LP,P((18))),(LL,L((12)))
IDENTIFY .S. (-100 $1$ .L. $7$ FORMAT .R. ( .Q.
COMPRESS .R.
IF (LR.GT.0) GO TO 10
* $1$ .S.
RETURN
```

```
 10 LEVEL CANCEL
    SUBSTRING .Q. INTO .S. ON .R.
    IF (LS.GT.1) GO TO 20
    * $1$ .L. $73 FORMAT(.Q.
    RETURN
 20 IF (LS/2 + LS/2 .NE. LS) GO TO 30
    ERRORS ALPHA DELIMITERS NOT PAIRED CORRECTLY
    RETURN
 30 BUILD .Q. $1$ .L. $73 FORMAT( .S=1=.
    DO 40 I=2,LS,2
    BUILD .R. $1$ .S=I=.
    CONVERT LR TO .P.
    BUILD .Q. $1$ .Q. .P. H .R. .S=I+1=.
 40 CONTINUE
    * $1$ .Q.
100 RETURN
    END
```

11.1.5 Typical Control Cards

CHARGE CARD

JOB CARD

```
RFL,10000.
LABEL,MACMAC,R,VSN=XXXX.
RFL,30000.
REWIND(MACMAC)
COPYBF(MACMAC,MACFOR,1)
COPYBF(MACMAC,MACLIB,1)
COPYBF(MACMAC,AUTOLAY,1)
RFL,120000.
 MACFOR.
RFL,[illegible]0000.
REWIND(OXCOM)
RUN(S,,,OXCOM)
AUTOLAY(NEWPGM,LGO,MACLIB)
REWIND(NEWPGM)
RFL,7777      (AS REQUIRED BY NEWPGM)
NEWPGM.
 [illegible]
       SOURCE DECK CONTAINING MAC STATEMENTS)
 [illegible]
       AUTOLAY TEXT CARDS
          (SEE AEROPHYSICS RESEARCH CORPORATION TN-122)
 [illegible]
       DATA DECK TO BE PROCESSED BY NEWPGM)
 [illegible]
 [illegible]
```

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE I. TABLE OF FILE USAGE

FILE NAME	POSITION ON CONTROL BLOCK CARD	POSITION ON PROGRAM CARD	USE ON PROGRAM CARD
F1	1	1	The name of the file containing the transformed program from precompiler NAME.
NFIRST	not present	2	If this parameter is FIRST, all information on file F1 is ignored.
L	not present	3	If this parameter is present and is not L the MAC listing will be suppressed.
F2	2	4	Name of the input file, normally INPUT
F3	3	5	Name of the output file, normally OUTPUT
F4	4	6	User declared files used by NAME
F5	5	7	User declared files used by NAME
.	.	.	
.	.	.	
.	.	.	
FN	N	N+2	User declared files used by NAME
TAPE 8	not present	N+3	Scratch file used by NAME
TAPE 13	not present	N+4	Scratch file used by NAME
TAPE 18	not present	N+5	Scratch file used by NAME
TAPE 5	not present	N+6	Equated to F2
TAPE 6	not present	N+7	Equated to F3
TAPE 77	not present	N+8	Equated to F1

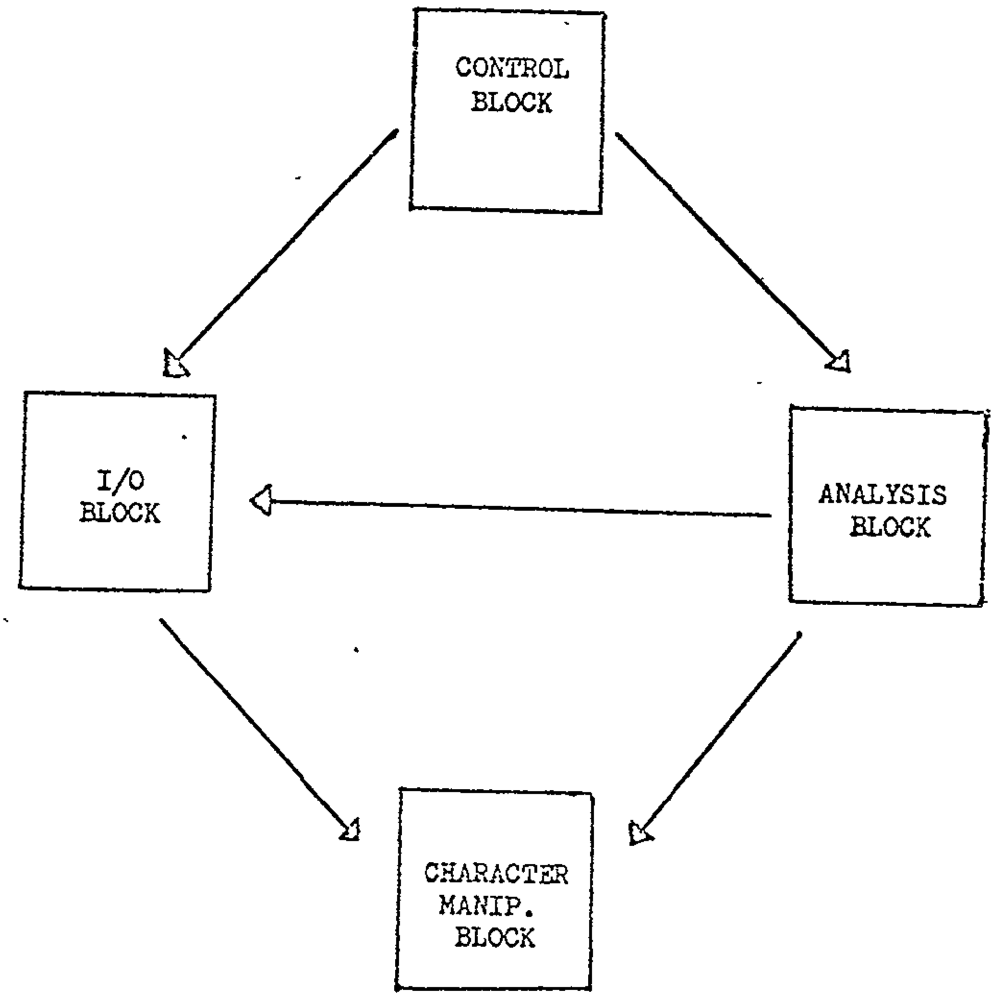


FIGURE 11.1-1 LOGICAL MAP OF A MAC PRECOMPILER.

TABLE OF CONTENTS FOR SECTION 12.1, PROGRAM PLOTTER

SECTION 12

GRAPHICS

12.1 PROGRAM PLOTTER: INDEPENDENT PLOT PROGRAM

Program PLOTTER provides a generalized x-y plotting and contour drawing capability in the ODIN system. Plot data may be stored in files created by other elements in the ODIN system and plotted output can be obtained on CALCOMP or COMPLOT printer devices by subsequent execution of PLOTTER. The PLOTTER program may also be used as a stand-alone plot program by input of all data including plot arrays. The x-y plotting program option was written by Watson and Glatt as part of the ODIN/RLV contract effort. The contour plotting option was written by Hague of Aerophysics Research Corporation as part of the related Air Force-sponsored ODIN/MFV contract. Both programs are now combined into the single PLOTTER program.

Section 7.2 contains several illustrations of the program's contour drawing ability. A typical plot obtained from the x-y plot option is presented in Figure 12.1-1.

Data input is through NAMELIST PLOTIN. A description of all input variables follows in Section 12.2.2. The analytic basis of the contour plotting option is described in Section 12.1.1. This analysis was originally described in a limited distribution Aerophysics Research Corporation technical note TN-140, "CONPLOT: A Rapid Code for Production of Three-Dimensional Contour Plots," by D. S. Hague.

12.1.1 Program CONPLOT

CONPLOT is a simple and rapid digital computer code for producing contour plots of three-dimensional functions. The function must be available in the form

$$Z_{ij} = F(X_i, Y_j); \quad i = 1, 2, \ldots, M \quad j = 1, 2, \ldots, N \qquad (12.1\text{-}1)$$

That is, the function must be defined over a regular grid in the X-Y plane. Contour plots are produced on CALCOMP or Houston plotters or any other device employing CALCOMP-compatible graphics calls. The program may readily be extended to other machines by virtue of its simplicity. Program CONPLOT in its stand-alone configuration consists of approximately 200 source cards plus CALCOMP Fortran-callable graphics subroutines. The analytic basis for program CONPLOT is presented below.

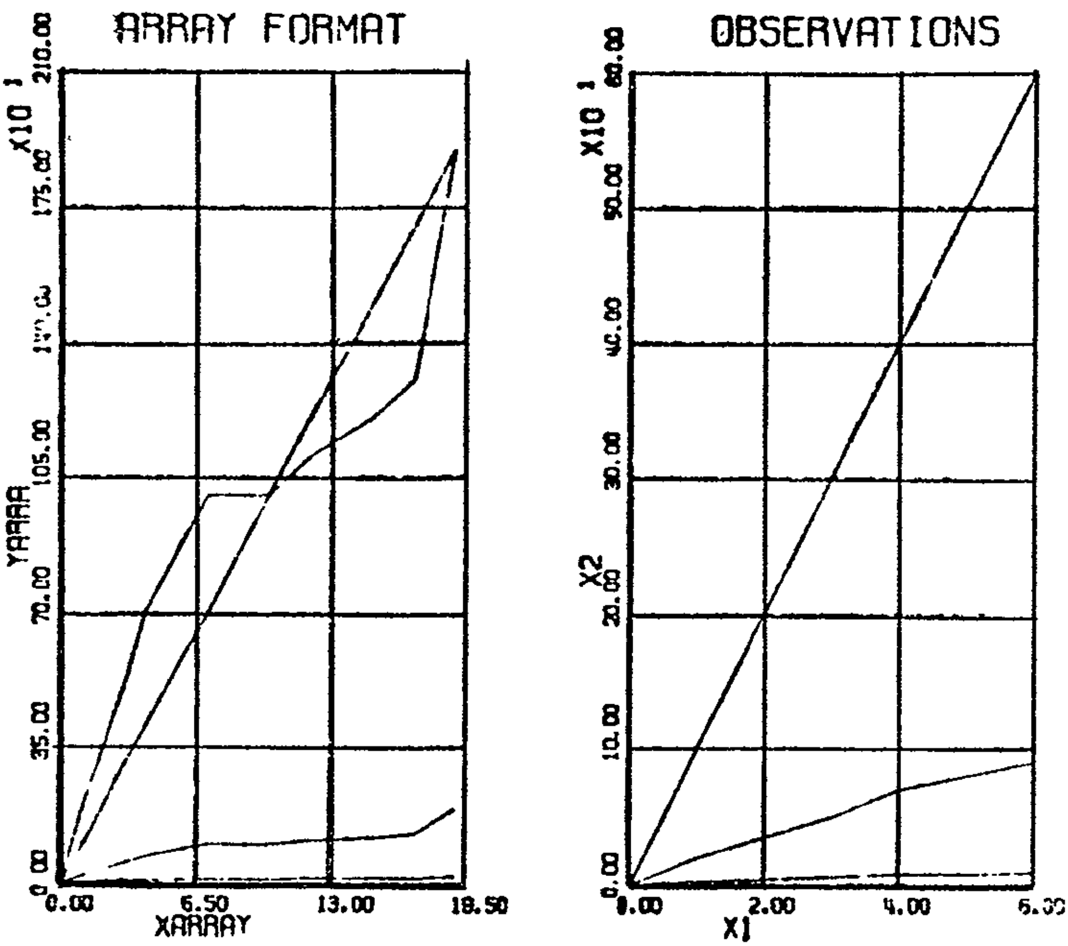


FIGURE 12.1-1. ILLUSTRATION OF X-Y PLOT

12.1.1.1 Analytic Basis of CONPLOT

Suppose a function of two independent variables is defined over the regular mesh X , Y_j. Then (M-1)(N-1) rectangular boxes can be selected from the adjacent points

$$P = (X_i,\ Y_j),\ (X_i,\ Y_{j+1}),\ (X_{i+1},\ Y_{j+1}),\ (X_{i+1},\ Y_j) \qquad \begin{array}{l} i = 1,\ M-1 \\ j = 1,\ N-1 \end{array} \tag{12.1-2}$$

Let the corners of any such box be identified in a clockwise manner as in Figure 12.1-2

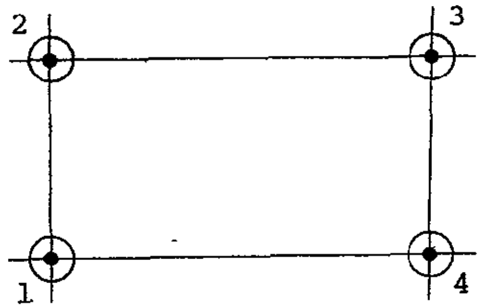


Figure 12.1-2. Rectangular Element Corner Identification

Given such a rectangle, the corner points and the function values at these corner points may also be uniquely defined by the notation of Figure 12.1-3.

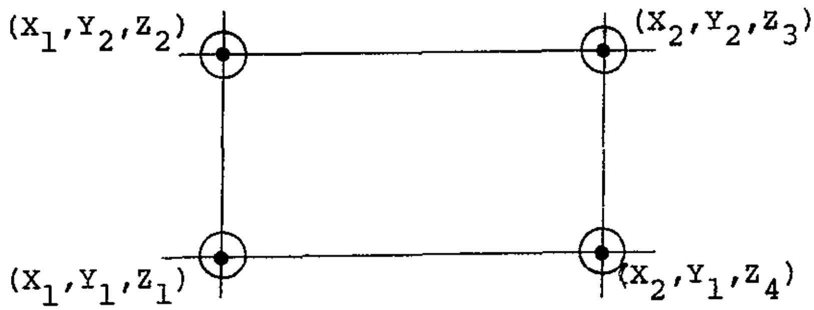


Figure 12.1-3. Local Coordinate and Function Values

Now consider the possibility that a contour of value z = z' passes through a given elemental rectangle. First, transform the corner values of Z by subtracting Z' to give the modified corner values

$$\begin{aligned} \bar{Z}_1 &= Z_1 - Z' \\ \bar{Z}_2 &= Z_2 - Z' \\ \bar{Z}_3 &= Z_3 - Z' \\ \bar{Z}_4 &= Z_4 - Z' \end{aligned} \tag{12.1.3}$$

Examining the topology of the contour line trace across the elemental rectangle, it follows that sixteen (16) possible types of contour trace exist; for each modified corner value given by (3) is either greater than or equal to zero, or less than zero, giving 2^4 topological types of trace. These trace types may be uniquely identified by a four digit binary number whose elements sequentially correspond to the corner points in the clockwise sequence of Figure 12.1-2 where 1 signifies that $\tilde{Z}_k > 0$ and 0 signifies that $\tilde{Z}_k \leq 0$. These sixteen types of trace are illustrated in Figure 12.1-4 with their corresponding four-digit binary number and contour trace type.

It can be seen that only two of the sixteen possible element topologies result in no contour trace. Two of the element topologies result in two alternative possible contour traces across the element. (These contour traces may be of either form displayed in Figure 12.1-4 for I = 1010 or I = 101).

Assuming linear interpolation for $\tilde{Z}$ along the elemental rectangle sides and local straight line contour traces within the element, the end points of the contour trace are readily found to be given by the expressions such as

$$\underline{I = 1011}$$

$$X_{e_1} = X_1$$
$$Y_{e_1} = Y_1 + F(Y_1, Y_2, Z_1, Z_2)$$
$$X_{e_2} = X_1 + F(X_1, X_2, Z_3, Z_4)$$
$$Y_{e_2} = Y_2 \qquad (12.1\text{-}4)$$

and

$$\underline{I = 1110}$$

$$X_{e_1} = X_1 + F(X_1, X_2, Z_1, Z_4)$$
$$Y_{e_1} = Y_1$$
$$X_{e_2} = X_2$$
$$Y_{e_2} = Y_1 + F(Y_1, Y_2, Z_4, Z_3) \qquad (12.1\text{-}5)$$

where (X_{e1}, Y_{e1}) and (X_{e2}, Y_{e2}) are the contour trace end points and the function F is defined by

$$F(A, B, C, D) = (B - A) \cdot \left|\frac{C}{D - C}\right| \qquad (12.1\text{-}6)$$

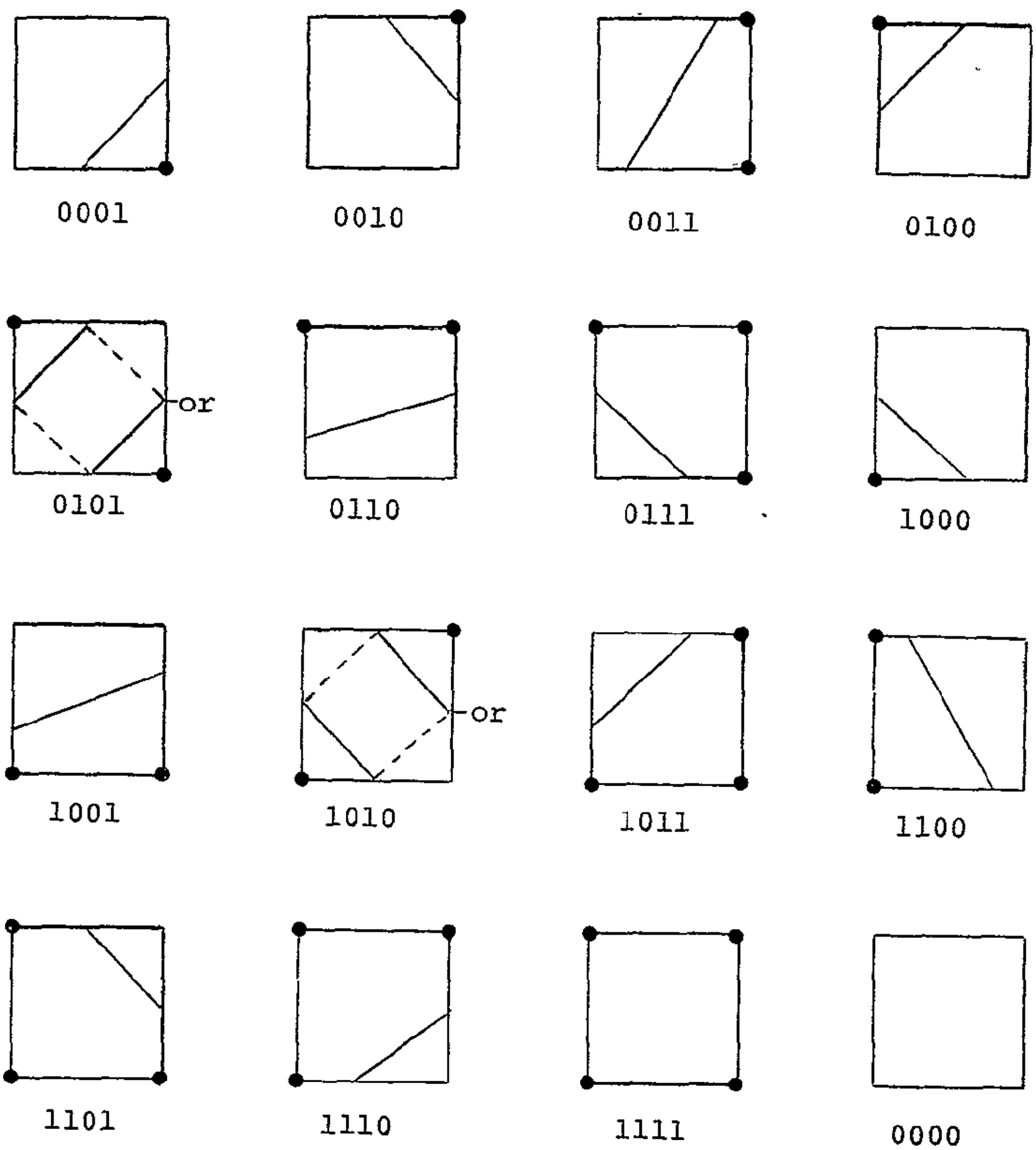
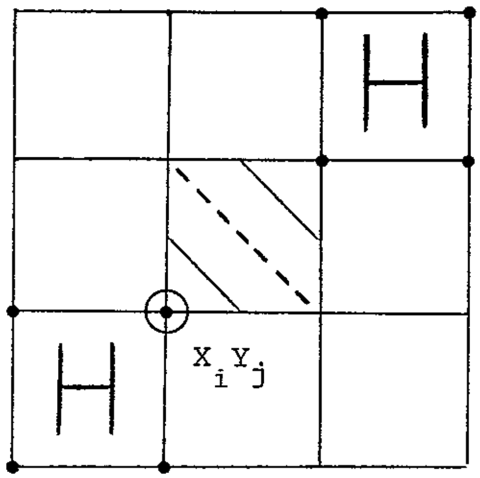


FIGURE 12.1-4. RECTANGULAR ELEMENT BINARY CODE AND CONTOUR TRACE TOPOLOGIES

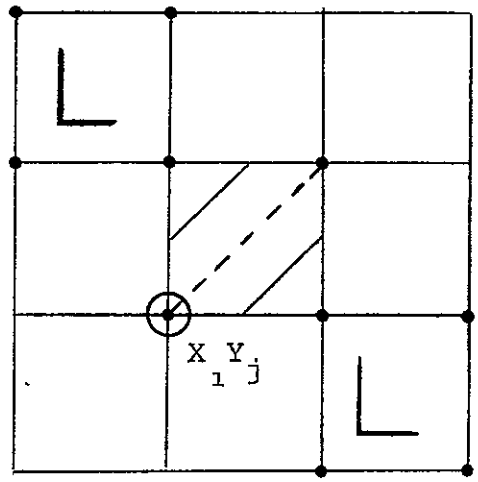
12.1.1.2 The Two Ambiguous Contour Cases

Two ambiguous cases arise for I = 1010 and I = 101. The contours may then be of either the solid or dotted line type in Figure 12.1-4. The ambiguity is resolved when the "fold diagonal" is defined. Thus, when I = 0101 if the principle diagonal is the fold diagonal, the dotted lines supply the correct contour types; if the other diagonal is the fold line, then the solid lines are the correct contour types. The converse is true when I = 1010.

Definition of the fold line requires more information than is available within a single elemental rectangle. Consider the case I = 1010 in Figure 12.1-5. In Figure 12.1-5(a) the upper right and lower left high elemental rectangle regions indicate a principle diagonal fold. In Figure 12.1-5(b) the low upper left and lower right elemental rectangle regions illustrate a fold about the other diagonal.



12.1-5(a). Principle Diagonal Fold



12.1-5(b) Other Diagonal Fold

CONPLOT contains logic for resolving the ambiguous cases using the above technique. A weighted assessment of the probability of each diagonal by the fold line is incorporated within the code. This logic covers the possibility of any adjacent elemental rectangle being absent due to edge or corner conditions in the z_{ij} array.

12.1.1.3. Stand-Alone CONPLOT Program Input

The CONPLOT program accepts the following data:

AXLEN	Length of plot on x-axis
AYLEN	Length of plot on y-axis
NX	Number of X_i values
XLOW	Smallest value of X_i

XHIGH	Greatest value of X_i
NY	Number of Y_j values
YLOW	Smallest value of Y_j
YHIGH	Greatest value of Y_j
NZ	Number of $Z = Z'$ contours desired
PLOTPC	Plot percentage, eliminate the lower and upper PLOTPC of the region for the contour plotting purposes

From this data the mesh X_i, Y_j is created and the function $Z(X_i, Y_j)$ is evaluated by a user-supplied subroutine. The NZ contours are then computed by means of 16 equations typified by (12.1-4) and (12.1-5) and the resulting contours are output on CALCOMP or Houston plotters. Typical plots obtained from CONPLOT are illustrated in Figures 12.1-6 through 12.1-11. Figure 12.1-6 illustrates a simple parabolic function with contours parallel to the x axis. Figure 12.1-7 illustrates the contours of a parabolic function in the coordinate $x^2 + y^2$. A more complex set of contours is presented in Figure 12.1-8; the contours are for a fourth-order function in x and y. Results are presented at two resolutions using scales from 0 to 5 and 0 to 10 in the x and y. Figure 12.1-9 presents contours of the well known Rozenbrock Valley function at three resolutions. Figures 12.1-10 and 12.1-11 illustrate the modulus contours of a fourth-order function of x and y having roots at the point

$$z_1 = (x, y)_1 = (-1, 1)$$

$$z_2 = (x, y)_2 = (0, .75)$$

$$z_3 = (x, y)_3 = (0, 0)$$

$$z_4 = (x, y)_4 = (.5, .5)$$

At the most coarse resolution a single minima appears. At the next resolution irregularities appear in the minimal contour profile. At the third and fourth resolutions two distinct minima appear. Finally, in Figure 12.1-11 all four minima appear in the contour plot. It may be noted that the contours of some functions of more interest have been presented in the discussion of Section 7.3.

12.1.1.4. Plotting Efficiency

Two alternative plotting procedures have been tested in constructing CONPLOT. These procedures are

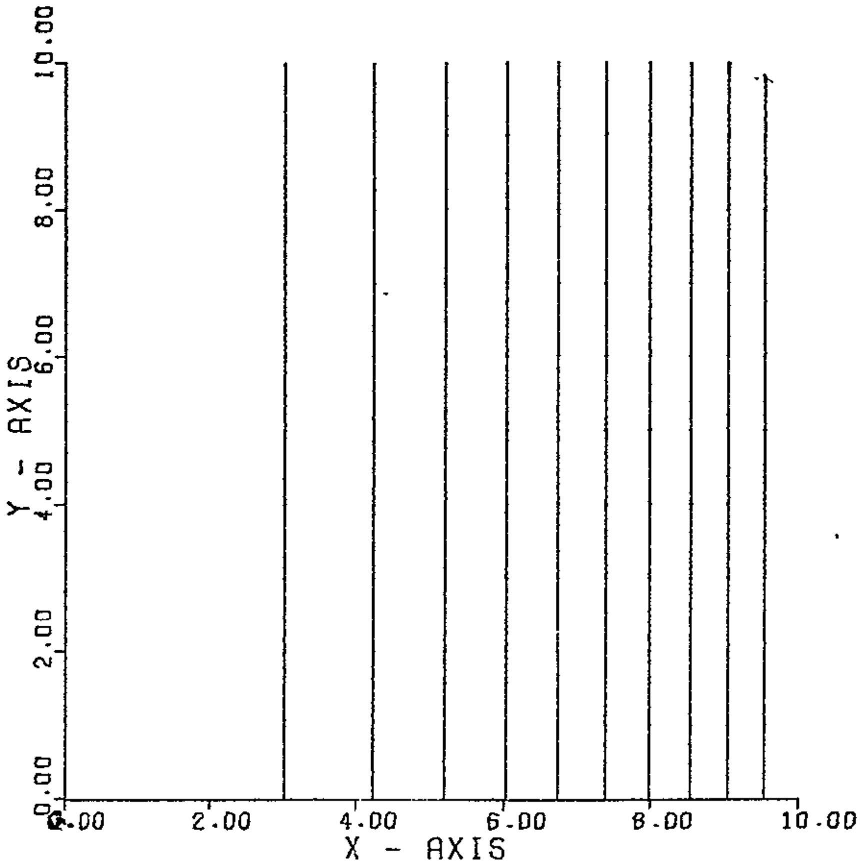


FIGURE 12.1-6. A PARABOLIC VALLEY IN X

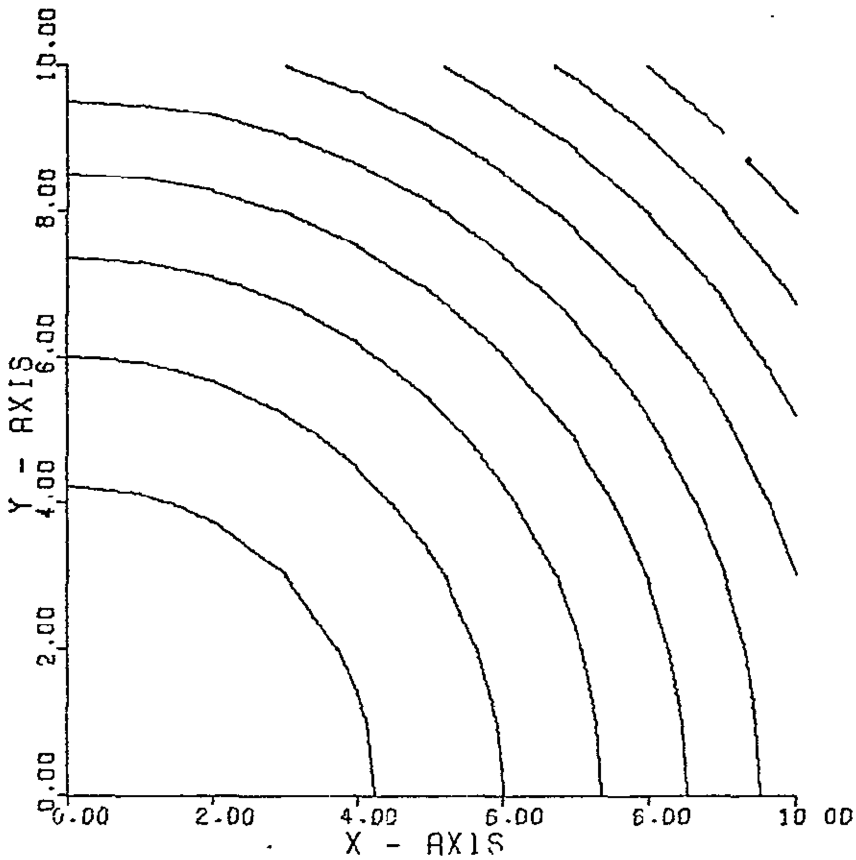


FIGURE 12.1-7. A PARABOLIC VALLEY IN $r = x^2 + y^2$

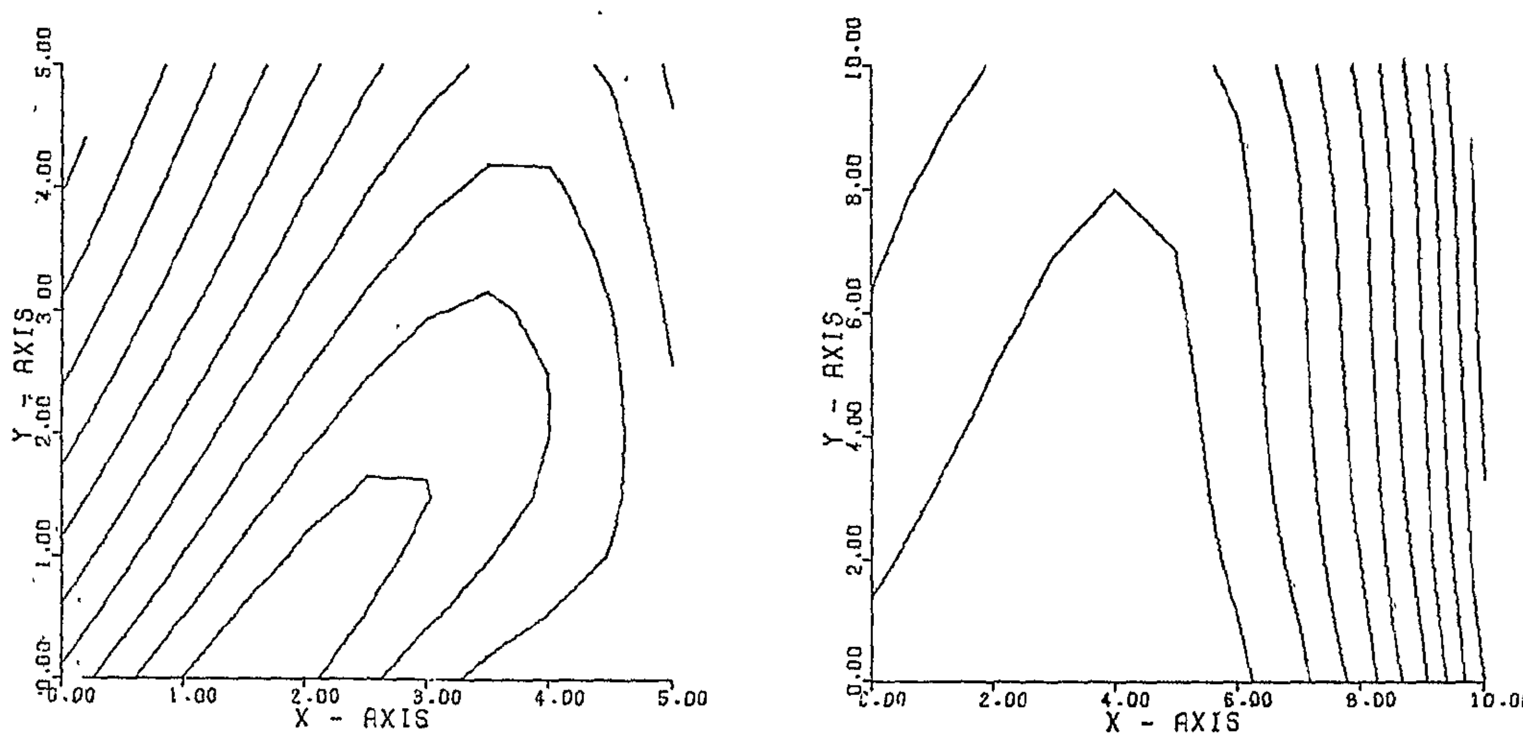


FIGURE 12.1-8. A FOURTH-ORDER VALLEY AT TWO RESOLUTIONS

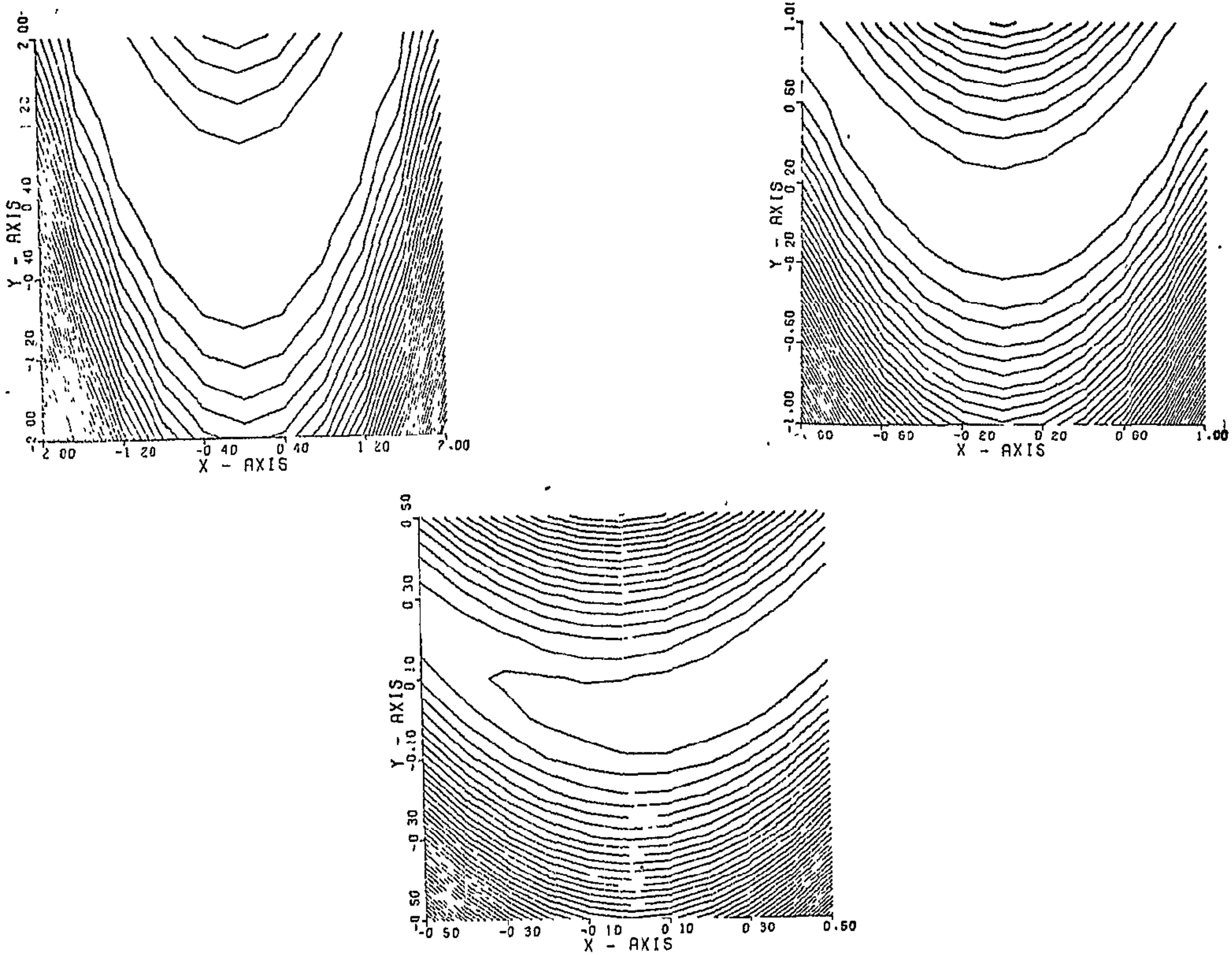


FIGURE 12.1-9. ROZENBROCKS'S VALLEY AT THREE RESOLUTIONS

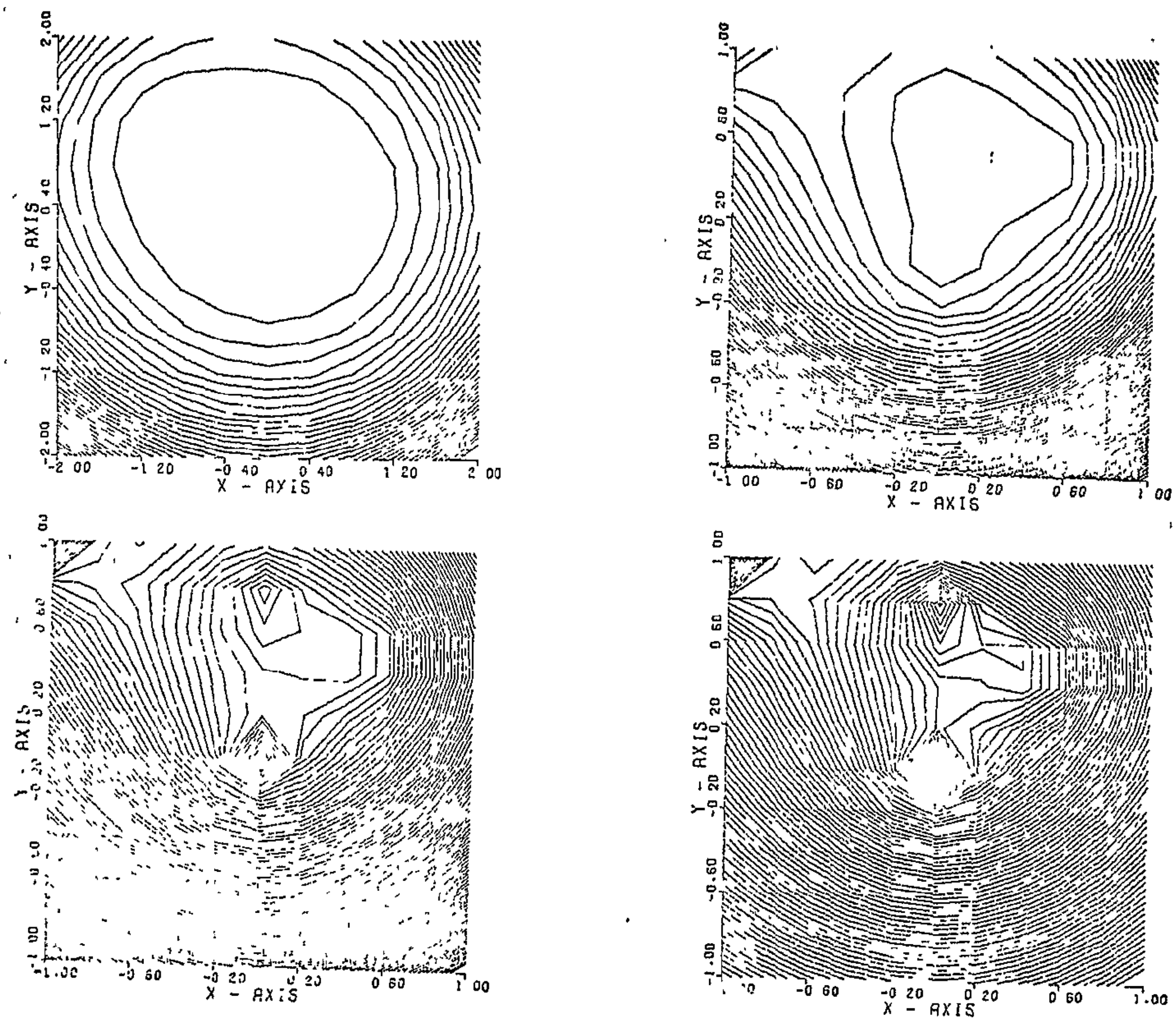


FIGURE 12.1-10. A FUNCTION WITH FOUR MINIMA AT VARYING RESOLUTIONS AND NUMBER OF CONTOURS

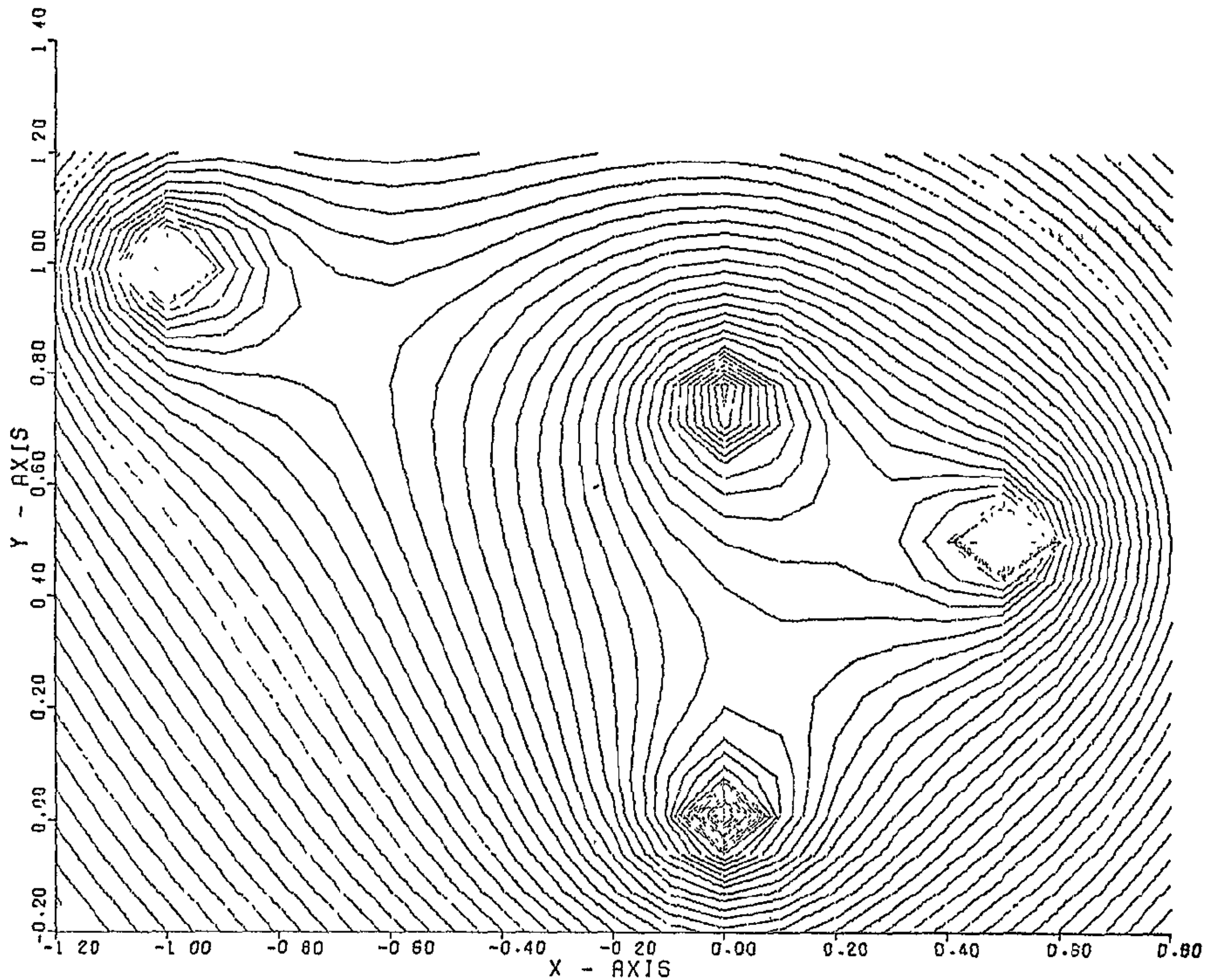


FIGURE 12.1-11. DETAILED CONTOURS FOR THE FUNCTION HAVING FOUR MINIMA

a. plot by contour

b. plot by mesh

When plotting by contour each mesh box is systematically searched to locate and sequentially plot the contour level being generated. Since the systematic mesh search is not formally related to the topology of the contour line, significant amounts of wasted plotting device pen movements are created as each contour line is developed. For example, a given contour line may exist in only the lowest (y = 1) and highest (y = N) set of mesh boxes. In this case since the systematic search employed fixes the x-mesh index i while varying the y-mesh index j, the plotter must hunt from bottom to top of the plot drawing each segment of the contour line. Of course, in this case were the order of variables in the systematic search be interchanged a fairly efficient plotter-pen motion would result. However, in general, the form of each contour line is not known a priori and wasted pen motion will result. Generally, on a typical contour plot the wasted pen motion in the plot by contour method creates very significant wasted plotter pen motions for the pen proceeds at the same rate in the down (line drawing) mode or the up (line skip) mode.

When plotting by mesh the procedure is to search for all contour levels drawn in one mesh box at a time. Thus, if K contours are required the trace of all contours and the corresponding lines in a given box are found before proceeding to the next box. This approach almost minimizes pen movements;for all lines in one box are drawn sequentially. In addition, since boxes are then treated sequentially, very few up pen movements are wasted in proceeding from box to box. The plot by mesh method is now used exclusively in CONPLOT. The result can be witnessed by an observer at the plotting device. The CONPLOT routine proceeds from mesh box to mesh box shading in all contours within a given box as it does so. When all mesh boxes have been processed, the complete set of contours emerges.

12.1.2. X-Y Plotter

The independent plot program provides for the generation of x-y plots and contour plots on a number of hardware display devices. Auxiliary plot text and tabulations of plot data can also be obtained from the program.

The plot instructions are read from the input data file and the data to be plotted can be read from either the input file or from auxiliary data files. The input instructions are read using a single NAMELIST input list.

$PLOTIN.....$

Default values are preset for all input variables. The default values tend to minimize the amount of user input required to generate a plot. Generally the input specifications are patterned after the procedure one might follow in preparing plot by hand. The user can select such options as grid, axis generation, annotation, titles, auxiliary text, line type, symbol specifications, etc. Those options not specifically selected by the user are generally bypassed in the program.

12.1.2.1 Plot Data Input Option

The data to be plotted is read into the program in either of two formats, array format or observation format. The data may come from the normal input or from a binary file in accordance with the following specifications:

INPUT = 0	Data will be loaded directly into the OBSTH array from the normal input unit.
INPUT = n	n Specifies the logical unit number from which the OBSTH array will be loaded.

Array Format. - Array fromat specifies the numerical values to be plotted are arranged in groups of similar observations such as time, attitude or velocity. Mathematically, array format is the sequence of numerical values:

$$OBSTH_{ij};\ i = 1, NUMP;\ j = 1, NOBSER$$

where NUMP is the number of observations and NOBSER is the number of observation functions. The plot data is actually a single dimensional array $OBSTH_k$ where k is the array element defined as:

$$k = (j - 1) \text{ NUMP} + 1$$

The independent plot program normally reads data from the input file in array format but other options are also available.

Observation Format. - Observation format specifies the numerical values are arranged in groups called observations. Each observation represents a sequence of values defining one element in each plot array. Observation format may be expressed mathematically as:

$$OBSTH_{k(i,j)};\ i = 1, \text{NOBSER};\ j = 1, \text{NUMP}$$

where NOBSER is the number of observation functions and NUMP is the number of observations. k Is the array element defined as:

$$k(i,j) = (j - 1) \text{ NOBSER} + 1$$

The independent plot program has the capability of reading data as observation format when the alternate value variable:

ALTVAL = .TRUE.

option is specified. If specified, a transposition of the observation format to array format is performed and the array format ultimately overwrites the input values of OBSTH.

Alternate Data File. - In addition to the two formats which can be selected for reading from input, an alternate data file can be specified:

INPUT = n

When this option is specified, the plot data is read from the logical unit n in observation format, transposed and placed into the OBSTH array in accordance with the specified values of NUMP on NOBSER. No file positioning is done by the program but the user can manipulate the file with the following input variables:

REWIND = .TRUE.	Rewinds n.
NSKIPF = m	Skips forward m Fortran files.
NSKIPR = m	Skips forward m Fortran records.

The alternate file format may include as the first record one word specifying the number of records on the file. If the one word record exists, the plot program can read it and store the value as NUMP. This option is activated by the specification:

NUMP15 = .TRUE.

The CDC 6600 has a Fortran callable binary blocking feature. If the alternate file was generated as a "binary blocked" file, the variable:

BLOCK = .TRUE.

must be set to read the file properly.

12.1.2.2 Plot Output Control

The output type specification can be x-y plots, contour plots, plot text or a printed tabulation in accordance with the options illustrated in Figure 12.1-12. Some combination of options are permissible but others are not. Contour plots and x-y plots are generated in different sections of the program and therefore, are mutually exclusive options. Multiple cases may be executed for as many plot cases as desired. Therefore, different options may be specified on successive cases. On the last case the parameter:

STOP = .TRUE.

is set which causes program termination. No other plot instructions are executed in the last case.

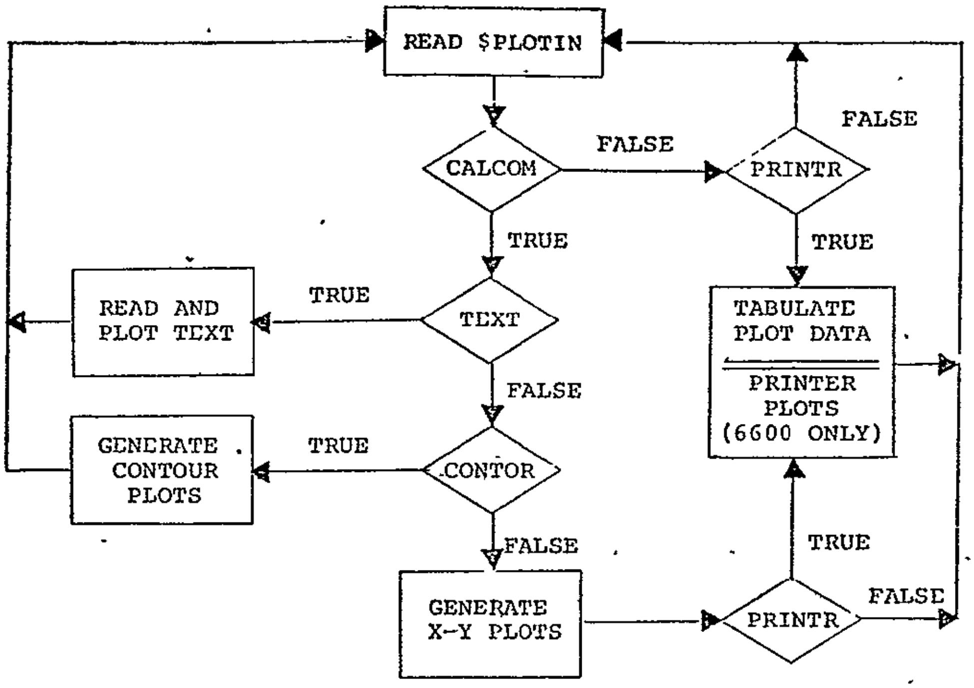
12.1.2.3 X-Y Plots

The PLOTTR program provides for the display of numerical information of the form:

$Y_i = f_j(X_i)$; i = 1, NUMP; j = 1, NPLOT

where X and Y are arrays of observations of known length, NUMP. Any array may be plotted with respect to any other array and any number of pairs may be presented on a single plot. NPLOT is the number of PLOTS desired.

Plotting is specified in accordance with the input variable array:



(A) PROGRAM SCHEMATIC.

DESIRED OUTPUT	INPUT VARIABLE OPTION			
	CALCOM	CONTOR	TEXT	PRINTR
X-Y PLOTS	TRUE	FALSE	FALSE	T or F
CONTOUR PLOTS	T or F	TRUE	FALSE	T or F
PLOT TEXT	TRUE	FALSE	TRUE	NA
TABULATION	T or F	T or F	FALSE	TRUE

(B) OPTION SPECIFICATION.

FIGURE 12.1-12. PLOT OPTIONS

$OBSPLT = K_1, K_2\ K_3 --- K_n$

Each K is a packed integer which defines the array pair in the OBSTH array to be plotted:

$K_i = \underline{XXYY}$

where XX represents a two-digit sequence defining the array to be plotted as the x-coordinate. YY represents a two-digit integer sequence defining the y-coordinate of the plotted line. Any number of K values may be specified, each representing a line on the plot. A plot sequence is terminated by a value:

$K_i = 0$

A second plot sequence may be specified by simply adding values of K:

$K_{i+1} = XXYY$

The last value of K should be:

$K_n = 7777$

which terminates the plotting and returns program control for new input data.

Plot Positioning. - The PLOTTR program has input parameters which control the defining of a new frame and the position of the plot within the frame. The parameters:

DXG and DYG

define the X- and Y- coordinates of the lower left corner of the plot in inches with respect to the lower left corner of the current frame. The parameters:

XMOVE and YMOVE

define the coordinates of the lower left corner of the next frame with respect to the lower left corner of the current frame. The frame includes all physical plotting which takes place as a result of a single set of plot instructions (defined by $PLOTIN). Frames of data may be superimposed to accomplish certain analysis objectives.

Line Type and Symbols. - The parameters, LINTYP control the type of lines that are to be drawn between the data points. The magnitude determines the frequency of symbols and the sign determines the combination of lines and symbols.

LINTYP = n	Means a symbol is to be drawn every nth point.
If n > 0,	Lines and symbols are drawn.
If n < 0,	Only symbols are drawn.

The parameter INTEQ determines which symbol is to be used by the specification of an integer from 1 to 22. A value of 0 specifies no symbols.

Data Scaling. - Data arrays can be scaled absolutely in terms of the axis length on which they are plotted or the arrays can be scaled relative to one another. The parameters:

XSIZE and YSIZE

specify the x- and y- axis lengths in inches. The first curve specified (see OBSPLT) determines the scale factor and the relative starting position for all curves on the plot. The array:

$SCALEF_i$

specifies individual scale factors for each plot array. This allows meaningful comparisons of multiple curve plots where the range of the data array differs significantly.

Elimination of automatic scaling may be specified in either or both directions by the specification:

MYX = .TRUE.
and/or
MYY = .TRUE.

If the MYX option is specified, the user must specify the scale factor and starting value the X-axis:

SCALEX = units/inch

STARTX = inches

If the MYY option is specified, the user must specify the scale factor and starting value of the Y-axis.

SCALEY = units/inch

STARTY = inches

The scale factors and start values are set by the program each time automatic scaling (MYX and MYY = .FALSE.) is specified so the scaling from previous cases may be used by properly setting the option flags MYX and MYY.

Scale Annotation. - Under the automatic scaling option the program generates axes of the specified length with tick marks at one inch intervals. The tick marks are identified by numerical values below or to the left of the axis centered on the tick. The axes may be notated additionally by a 6-character input name centered on the axis. One name may be input for each observation function as follows:

$OBSERV_i = N_i$, i = 1, NOBSER

where N_i is a hollerith name of the form, nH name, and NOBSER is the number of observation functions.

Grid Generation. - A grid may be generated which represents vertical and horizontal lines at the tick marks by the input parameter:

GRID = .TRUE.

Title Generation. - A title from the plot may be generated by setting the parameter:

NREM = n

where n is the number of words of the title. The program will use the first n words of the REM array.

REM = N_i; i = 1, NREM

and place this title at the top of the plot. The height of the characters is in the title as specified by the parameter REMSIZ in inches.

Auxiliary Plot Text

The plotter program has the capability of generating auxiliary plot text by a separate case setup as follows:

$PLOTIN TEXT = .TRUE.,$

(text cards)

The text cards immediately follow the case data. The character strings given on the text cards are reproduced exactly starting the first card at a location specified by:

DXG and DYG

in inches. The character height specification is given by HTEXT. Cards will be read and the characters plotted with line spacing of:

1.5 * HTEXT

until a card is encountered with a numeric 2 in column 1. Control is then returned for a new case.

Virtual and Display Window

Virtual and display graphics deals with the translation of the user's data to a physical location on the display device. The virtual space may be of any dimension while the display space is limited by the physical size of the display device. The function of the virtual graphics package is to map the virtual space into the specified display area. With an understanding of the relationship between the virtual space and the display area, the user can freely manipulate the display to reflect his need. For example, he can plot three different sets of data in the same display area or he can display the same data to different scales to meet special needs.

12.1.3. Data Summary

Name	Default(s)	Description of Input
ALTVAL	.FALSE.	Logical variable. If .True. data will be read as observation functions. Otherwise data will be read as plot arrays.
BLOCK	.FALSE.	Logical variable. If .True., binary blocking of the observation unit will be expected. (CDC 6600 only)
CALCOM	.TRUE.	Logical variable. If .True., x-y plots will be generated. This variable is used to activate any device to which the program is linked such as Tektronics, SD4060 or Varian.
CONTOR	.FALSE.	Logical variable. If .True., a contour plot will be generated from OBSTH data CALCOM must be set .True.
COPY	.FALSE.	Generate hard copy (online devices).
DTALOG	.FALSE.	Logical variable. If .True., data base output routine will be called.
DXG	1.0	X-axis origin for the current chart relative to the current device origin specified by XMOVE.
DYG	1.0	Y-axis origin for the current chart relative to the current device origin specified by YMOVE.

Name	Default(s)	Description of Input
GRID	.FALSE	Logical variable. If .TRUE., one inch grid will be generated on x-y plots.
HTEXT	.21	Height of title in inches.
INPUT	0	Zero for reading plot data from cards .GT.0 for reading plot data from another unit.
INTEQ	1	Integer from 0 to 22 indicating the plot symbol.
LINTYP	0	Control parameter which describes the type of line to be drawn through the data points. The magnitude determines the frequency of plotted symbols. I.E. LINTYP=-4 Means every fourth point. LINTYP=0 Straight lines with specified symbol at the end of the line (see INTEQ). LINTYP=+ Lines and symbols. LINTYP=- Symbols only.
MYX	.FALSE.	Logical variable. If .TRUE., user may input STARTX and SCALEX.
MYY	.FALSE.	Logical variable. If .TRUE., user may input STARTY and SCALEY.
NAMES (100)	.FALSE.	Logical variable. If .TRUE., NOBSER six-character names will be read. These are the plot array titles.
NAXIS	1	Number of times the beam or pen will trace the x- and y- axes.
NDECP	2	Number of decimal places used in scale annotation.
NOBSER	NONE	Number of observation functions (or plot arrays).
NPAGE	0	Page number of the plot (printer only).
NREM	0	Number of words of title to be read. If not zero, NREM 10-character words will be read immediately after the $PLOTIN namelist.

Name	Default(s)	Description of Input
NSKIPF	0	Number of observation function files that will be skipped on observation unit before starting to read data from it.
NSKIPR	0	Number of observation functions to skip on observation unit before starting to read data.
NUMP	NONE	Number of plot points per plot array or the number of observations. There is an internal limit of 213 points.
NUMP15	.FALSE.	Logical variable. If .TRUE., NUMP will be read from observation unit as the first record.
NZCUTS	NONE	Number of contours requested.
OBSERV (100)	BLANK	Observation function names to be used for scale annotation. They are read in namelist format as: OBSERV=nJIname1,nIIname2,----.
OBSPLT (120)	7777	Integer definition of the plot functions. Each pair of plot arrays or observation functions is defined by a 4-digit number, the first two represent the independent variable, the second two represent the dependent variable according to the input order of the plot arrays or OBSTH functions. A zero indicates the end of one chart. More charts can be generated up to a limit of 120 entries in the OBSPLT array. User should enter a value of 7777 after the last chart.
OBSTH (2000)	NONE	Plot data in one of the following formats. For ALTVAL=.FALSE., plot arrays are loaded, A(1),--,A(N),B(1),--B(N),C(1)--C(N), For ALTVAL=.TRUE., plot arrays are loaded, A(1),B(1),C(1),------,A(N),B(N),C(N), NOTE: The maximum number of plot points is 2000 but can be changed to the alteration of 3 Fortran statements in main program as follows:

Name	Default(s)	Description of Input
		COMMON/TEMOBS/ODATA(n+1)
		REAL OBSTH(n)
		DATA NADATA/n/
		The value of n may be set to any desired value. Program load size is altered in direct relation to the number n.
PAGE	.FALSE.	Start a new frame at XMOVE, YMOVE from previous origin.
PRINTR	.FALSE.	Logical variable. If .TRUE., tabulation will be generated.
PRINTO	.FALSE.	Logical variable. If .TRUE., the namelist input data will be printed.
REM (30)	BLANK	Title to be placed at the top of the chart. It is read in namelist format as:
		REM=nH title of plot,
REMSIZ	0.21	Height of title in inches.
REREAD	.FALSE.	Logical variable. If .TRUE., observation [illegible] will be [illegible] before reading it.
SCAL	7.5	Size of the plot device window in inches. Virtual plot specified by the maximum of XSIZE and YSIZE will be scaled to this dimension.
SCALEF (100)	1.0	Scale factor array. One for each plot array or observation function. It is used to scale one plot array relative to the others for plot purposes but does not affect the original data in OBSTH.
SCALEX	1.0	Units per inch for X.

Name	Default(s)	Description of Input
SCALEY	1.0	Units per inch for Y.
STARTX	0.0	Starting value for X-axis.
STARTY	0.0	Starting value for Y-axis.
STOP	.FALSE.	Logical variable. If .True., program will stop without generating additional plot information.
TEXT	.FALSE.	Logical variable. If .True., card images will read and plotted scaling will be in accordance with the following formula: 6*NREM*HTEXT/SCAL
X[illegible] (100)	None	Array of points defining the X-axis of a contour plot.
XMOVE	8.5	X-distance between plot origins. For on-line devices, a screen erasure is affected.
XORIGN	1.0	X-axis origin for the current chart relative to the current device origin specified by XMOVE.
XSIZE	6.0	X-size length in inches for virtual plot. It also specifies number of scale (and grid) divisions which will be employed. Origin is moved before plotting if the PAGE option is involked.
YMESH (100)	None	Array of points defining the Y-axis of a contour plot.
YMOVE	0.0	Y-distance between plot origins. For on-line devices, a screen erasure is affected.
YORIGN	1.0	Y-axis origin for the current chart relative to the current device origin specified by YMOVE.

Name	Default(s)	Description of Input
YSIZE	8.0	Y-axis length in inches for virtual plot. It also specifies number of scale (and grid) divisions which will be employed. Origin is moved before plotting if the PAGE option is invoked.
ZCUTS (25)	None	Values of the contours.

12.1.4. Subroutine Descriptions

This section contains an alphabetically arranged list of descriptions for the Independent Plot Program Subroutines.

12.1.4.1 Program PLOTTR

PLOTTR is the main program for the Independent Plot Program. Written in Fortran, PLOTTR establishes nominal values for input variables, reads data, sets up program options, initializes the plot devices, establishes certain scaling parameters and controls the plotting of text and data. Figure 12.1-13 shows a functional flow chart for PLOTTR. The flow logic begins with the reading of namelist data called $PLOTIN. An input parameter PRINTR provides the user with the option of printing all the input data except the actual data to be plotted. If the target plot device requires initialization, the logic for initializing the device is called on the first case but not thereafter. A logical input variable, STOP, provides for a normal stop after the last case. Logic is included to obtain the actual data in three different formats.

First the data may be read in BCD format through the normal namelist input channels via an input variable called OBSTH. The data may be read in array format (the format used internally by the program) or it may be read in observation format. The latter requires that the data be reordered by the program before use. The reordering is done by a routine called WRITE15. WRITE15 simply writes the data out in binary format on a user specified unit to be read back in array format.

A third option allows the data to be read in binary observation format from a file generated outside the independent plot program. In the latter case the data is converted to array format as it is read into the program. After all data is read in the plot size is established through a call to SCRENE and the main plot generation subroutine GENPLT is called. GENPLT controls the generation of contour and x-y plot options. After return from GENPLT, the flow logic loops back to the beginning of PLOTTR to read more data.

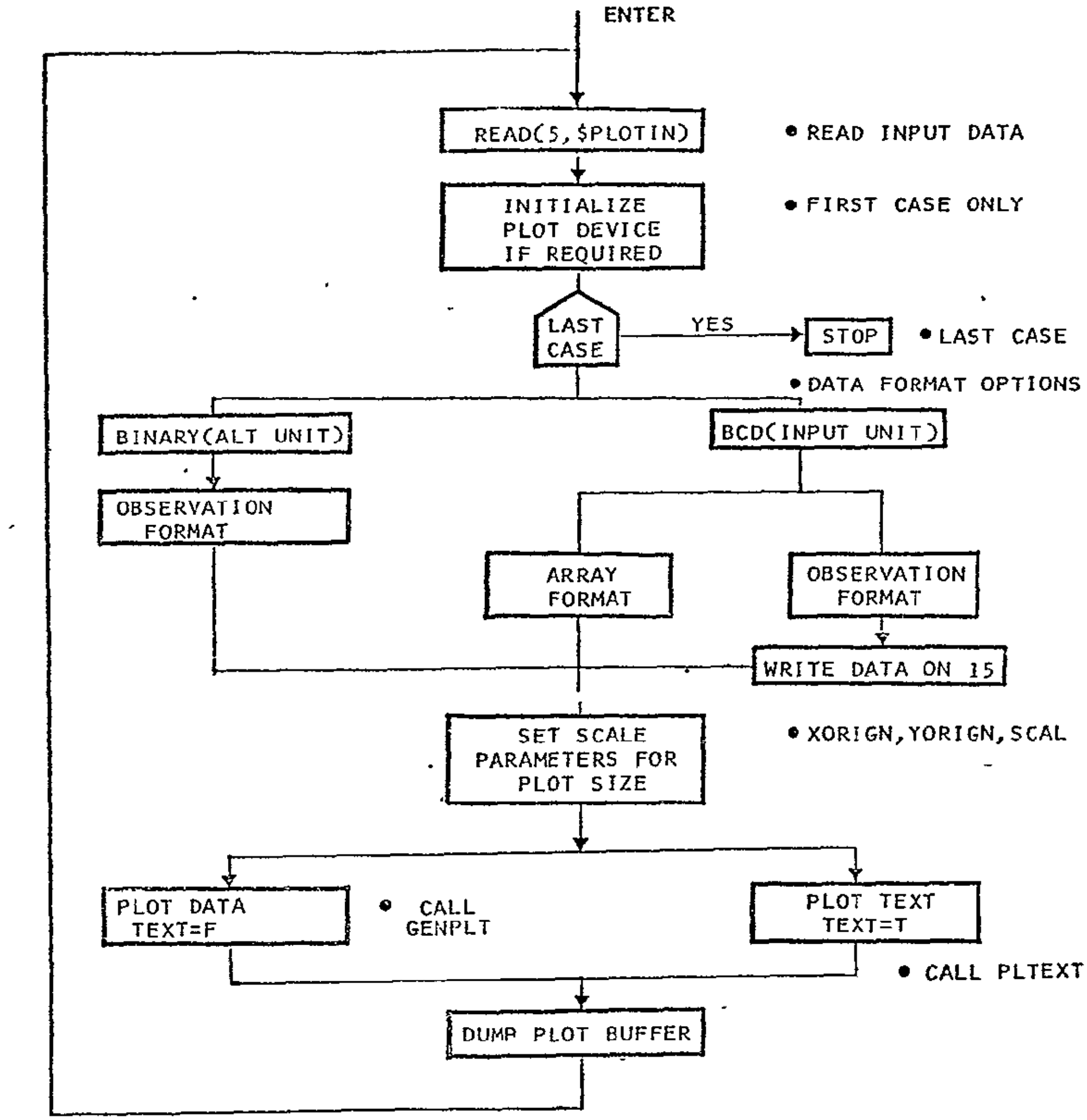


FIGURE 12.1-13 FUNCTIONAL FLOW CHART - PLOTTR

12.1.4.2 Subroutine AXIS

AXIS is a modified CALCOMP subroutine written in Fortran for generating annotated axes with tick marks at one inch intervals. The use of the subroutine is as follows:

```
Call AXIS(X,Y,BCD,NC,SIZE,THETA,YMIN,DY)
```

- **X, Y** — Coordinates of the starting point of the axis with respect to the plotting area origin in inches.
- **BCD** — Character label for the axis.
- **NC** — Number of characters in the label.
- **SIZE** — Length of the axis in inches.
- **THETA** — Angle of rotation measured clockwise from the x-axis in degrees.
- **YMIN** — Functional value to be assigned to the first position on the axis.
- **DY** — Change in the functional value for inch.

The AXIS routine is normally called after scale which determines the YMIN and DY values. AXIS calls PLOT and SYMBOL to generate the lines, tick marks and annotation on the axes. NUMBER and ROUND are used to convert binary numbers to characters which can be plotted as scale and annotation.

12.1.4.3 Subroutine CONPLT

CONPLT is the main subroutine for generating contour plots from an input mesh of data defined in the input array, OBSTH. The data is defined in equal increment mesh points in x and equal increments of mesh points in y. The boundary of each incremental window is searched for intersections with specified contours. Fig. 12.1-14 is a functional flowchart of the subroutine CONPLT. Upon entering the subroutine, a title and scale are generated for the data. The x-limits, y-limits and z-level are set in a triple DO-LOOP. The type of intersection is determined from the values of z at the four corners of the incremental window. Fig. ~~12.1-4~~ illustrated the types of intersections which can be accounted for. Each has a separate algorithm which is coded at the statement label identified in the figure. Once the boundary points are computed, the subroutine LINE is used to generate the contour vector. Contour vectors are determined for each z-level and for each incremental window. The result of all evaluations is the appearance of the continuous contour plot for the entire region specified by the x-mesh and y-mesh.

12.1.4.4 Subroutine DEF

DEF is a Fortran subroutine for ejecting a page and placing a title on the new page.

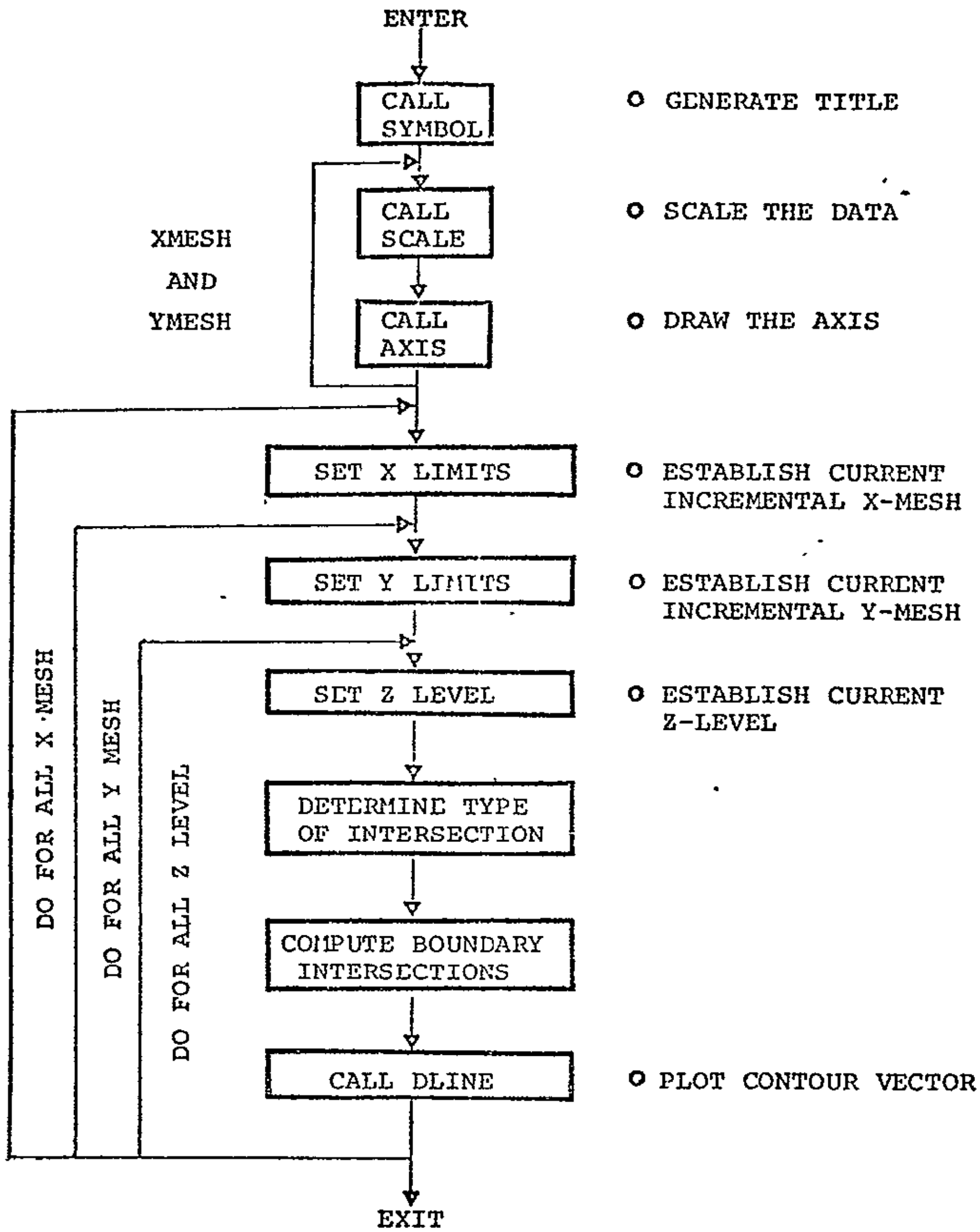


FIGURE 12.1-14. FUNCTIONAL FLOW CHART, CONPLT

12.1.4.5 Subroutine GENPLT

GENPLT is a Fortran subroutine which controls the data acquisition from alternate files but also controls X-Y plot and contour plot options for the program. All data in this subroutine is passed through the common as follows:

```
Call GENPLT(NADATA,OBSTH,NUMP,INPUT,CALCOM,OBSERV,
            NOBSER,YOBSTH,CONTOR)
```

NADATA	Maximum size of the internal storage array OBSTH.
OBSTH	Internal storage array for plot data.
NUMP	Number of points for plot array.
INPUT	Logical unit for binary input data.
CALCOM	Vector plot option flag.
OBSERV	Names of the plot arrays.
NOBSER	Number of observations for plot arrays.
XOBSTH	An array containing a single observation.
CONTOR	Contour plot option flag.

A detailed flow chart for subroutine GENPLT is shown in Fig. 12.1-15. Upon entry, GENPLT tests the parameter INPUT to determine the source of the data. If 0, the data is assumed to be in core in the OBSTH array. Otherwise, the data will be read from the observation unit defined by the value of INPUT. The routine READ15 loads the data. Two tests are performed to determine the limits for the number of observation functions. If NOBSER is less than one, an error message is printed and the control is returned to the main program. If the number of observation functions, NOBSER is greater than NADATA/2, an error message is printed and the control is returned to the main program. The subroutine THROBS is then called to actually generate the X,Y plot. THROBS contains logic which controls the generation of printer plots, tabulations and/or vector plots. The contour plot option flag CONTOR is then tested to determine whether a contour plot is being generated. If the flag is true, the subroutine CONPLT is used to generate the contour plot. Control is then returned to the main program, PLOTTR.

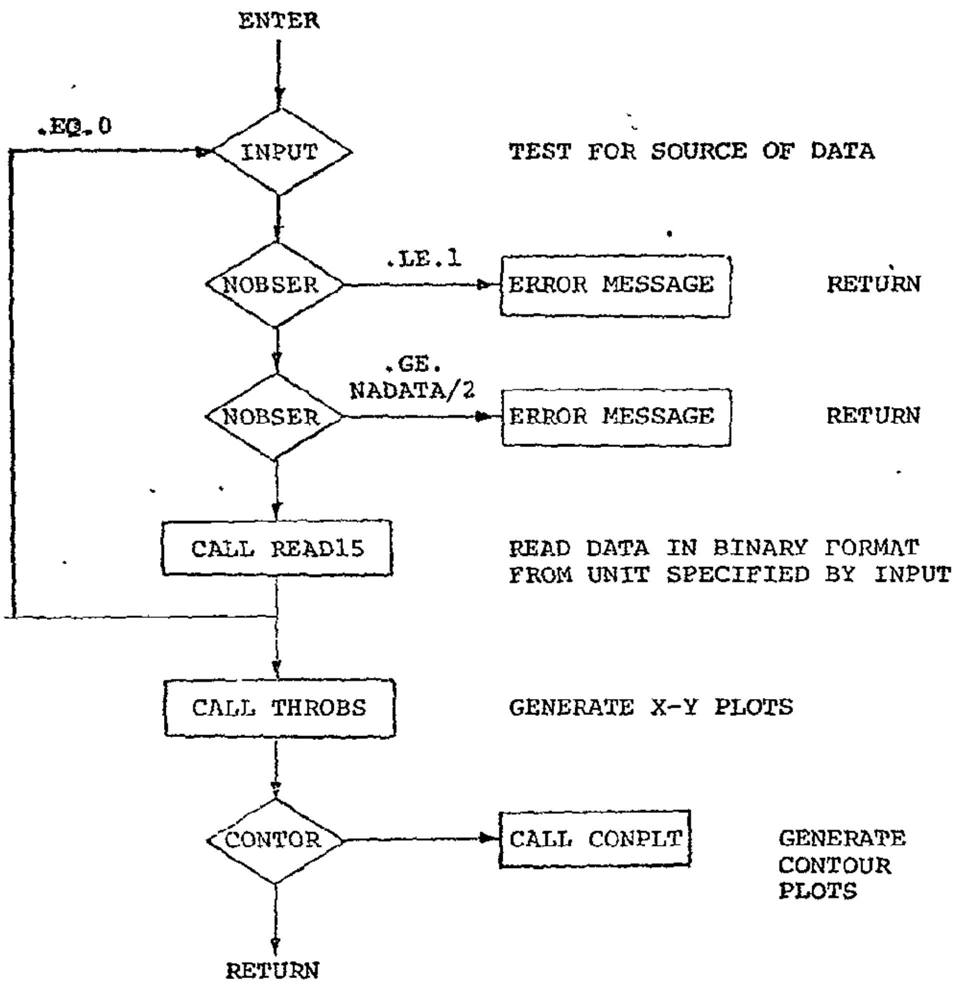


FIGURE 12.1-15. DETAILED FLOW CHART, GENPLT

12.1.4.6 Subroutine ESCALE

The subroutine ESCALE is used in the generation of printer plots for the CDC 6000 version of the PLOTTR program. It scales the data to the size of the standard 11 x 16 printer page size.

12.1.4.7 Subroutine FTNBIN

FTNBIN is an assembly language routine which establishes binary blocking for sequential files on CDC 6600 computer. The usage is as follows:

Call FTNBIN(I,J,K)

I — Flag which determines whether blocking is to be on or off.

- I = 1 Binary blocking is to be turned on.
- I = 0 Binary blocking is to be turned off.

J — Number of files to be blocked or unblocked in accordance with the flag I.

K — Name of the array containing the integer numbers of the file to be blocked or unblocked.

FTNBIN can be called to block a specified set of files or can be called to block or unblock all files by setting the parameters J and K equal to zero. FTNBIN is a dummy subroutine in this library but can be replaced with the system FTNBIN routine when the program is compiled on the CDC 6600. Binary blocking is automatically provided for on Univac Exec 8 Fortran.

12.1.4.8 Subroutine GETMAX

This subroutine is used in the generation of paper plots for the 6600 version of the PLOTTR program. It scans a data array to determine the maximum value which can be placed on the paper plot.

12.1.4.9 Subroutine GRIDHP

GRIDHP is a Fortran subroutine used for generating for grids on a display device. The usage is as follows:

Call GRIDHP(XSIZE,ZSIZE)

XSIZE The length of the x-axis in inches.

YSIZE The length of the y-axis in inches.

The routine generates a rectangular grid pattern in one inch increments. The subroutine PLTT is used to draw the grid lines.

12.1.4.10 Subroutine GRIDXY

GRIDXY is a Fortran subroutine for generating the grid representation on printer plots. The usage of GRIDXY is as follows:

Call GRIDXY(XMIN,XMAX,YMIN,YMAX,BX,BY)

XMIN The minimum X value.

XMAX The maximum X value.

YMIN The minimum Y value.

YMAX The maximum Y value.

BX The X distance between grid lines.

BY The Y distance between grid lines.

The routine generates the representation of grid lines by placing periods (.) appropriately into the plot buffer. It also simulates the coordinate axes of the plot by placing the character I appropriately to represent segments of the Y-axis and the character (-) to represent segments of the X-axis. The character (+) is used at the grid intersection point.

12.1.4.11 Subroutine HPLNLN

This subroutine is the main driver for generating X,Y plots. It performs the data scaling, access generating, title generation and the drawing of the lines associated with the specified data arrays.

12.1.4.12 Subroutine LINE

This is a Fortran subroutine which makes the appropriate calls to generate a line on the output device. The subroutine has six calling arguments as follows:

```
CALL LINE(X,Y,N,K,J,L)
```

X,Y — X and Y are arrays containing the X and Y coordinates respectively for the n points to be plotted.

K — The significance of K is two fold.

(1) The magnitude of K specifies that the data is stored in every K cell.

(2) If the sign of K is positive, the pen will be moved to the first point. If the sign of K is negative, the pen will be moved to the first point in a lowered position.

J — The magnitude of J is the number of points for every point to be plotted. Where J equals 0, a line plot only. Where J equals 1, a point for every data point. Where J equals 5, a point will be plotted at every 5th. point. A minus sign of J specifies a point plot without connecting points. A positive J specifies a line connecting every data point.

L — The value of L specifies the type of symbol to be used. Integer numbers 1 through 22 may be specified. These symbols are different for different plot devices.

12.144.13 Subroutine NUMBER

This subroutine is used to convert a floating point number to be BCD and to draw the resulting alpha numeric characters on the plot. The calling sequence is:

```
Call NUMBER(X,Y,HGHT,FPN,THETA,N)
```

X,Y X and Y specify the position of the first character in inches.

HGHT The height of the characters.

FPN The value of the floating point number.

THETA The angle at which the characters will be drawn, measured from the x-axis, positive in a counter-clockwise direction.

N This is the number of decimal places to be retained in the conversion. N may be specified at any value from -1 to 11. If N is -1 or 0, no decimal places will be drawn. If N is -1, a decimal point will be suppressed. Any other value of N specifies the number of decimal places to be drawn.

The integer portion of the number is restricted to a maximum of 6 characters. The decimal portion is restricted to 11 digits.

12.1.4.14 Subroutine PAPERP

This subroutine serves as a calling program for controlling the X,Y plot options. The two options are as follows:

If PRINTR is .TRUE., then a paper plot will be generated on the normal output (6600 only).

If CALCOM is .TRUE., then a vector plot will be generated on whatever device is interfaced to the PLOTTR program.

12.1.4.15 **Subroutine PLCPTS**

This subroutine is used in the generation of paper plots on the CDC 6600. It contains the logic for placing plot characters within a buffer region in the appropriate position to simulate a plot when the buffer is printed on the normal 11 x 16 printer output. It also flags and identifies the points which did not fall within the plot region.

12.1.4.16 **Subroutine PLTEXT**

This subroutine is used for generating auxiliary plot text associated with X,Y plots or contour plots. The subroutine is called from the main program when the option TEXT is specified as .TRUE. Upon entering this subroutine, the plot window is established by input values of DXG, DYG and SCAL. Then cards are read from the input and the characters on the card are scaled and placed within the established window. Spacing between card images is automatically provided at 1.5 times the character height. The buffer is dumped at the completion of the plotting.

12.1.4.17 Subroutine PLOT

This subroutine performs all line drawing functions.

12.1.4.18 Subroutine PPLNLN

PPLNLN is a Fortran subroutine which generates X,Y plots as part of the normal printed output. The subroutine is used as a utility routine as follows:

```
Call PPLNLN(X,Y,NPTS,XMAX,XMIN,YMAX,YMIN,NPLTS,TT,TB)
```

X	Array of X points to be plotted.
Y	Array of Y points to be plotted.
NPTS	Number of points to be plotted.
XMAX	Maximum X-value.
XMIN	Minimum X-value.
YMAX	Maximum Y-value.
YMIN	Minimum Y-value.
NPLTS	Number of points.
TT	Title array at the top of the plot.
TB	Annotation at the bottom of the plot.

Upon entry into the PPLNLN routine, a plot vector area is set up which will contain the characters representing the grid, scales and plotted information. Scale factors are determined based on the size of the paper (11 x 16). The plot area is hard coded to be seven inches on the Y-axis and ten inches on the X-axis. Therefore the resolution is 100 units in the X direction and 42 units in the Y direction. The subroutine GRIDXY is called to set up a grid in the plot buffer. Vertical grid lines are made up of the character minus. The PLCPTS routine is called to place the plot points into the plot buffer. The plot data points are identified by characters in the following sequence:

X,O,+,$,/,2,3,4,5,A,B,C,D,E,F,G,H,K,M,N,P,R,S,U,V,W,X,Y,Z,

Thirty characters are supplied for up to thirty plots which may be specified for single chart. Plot points which fall within the same resolution area use the character asterisk to replace the normal plot character. The subroutine POPLOT is called to print out the plot buffer and place the title and annotation on the printed page. The maximum, minimum and scale factors for all of the data are printed at the bottom of the plot. The functional flow chart for PPLNLN is given in Fig. 12.1-16.

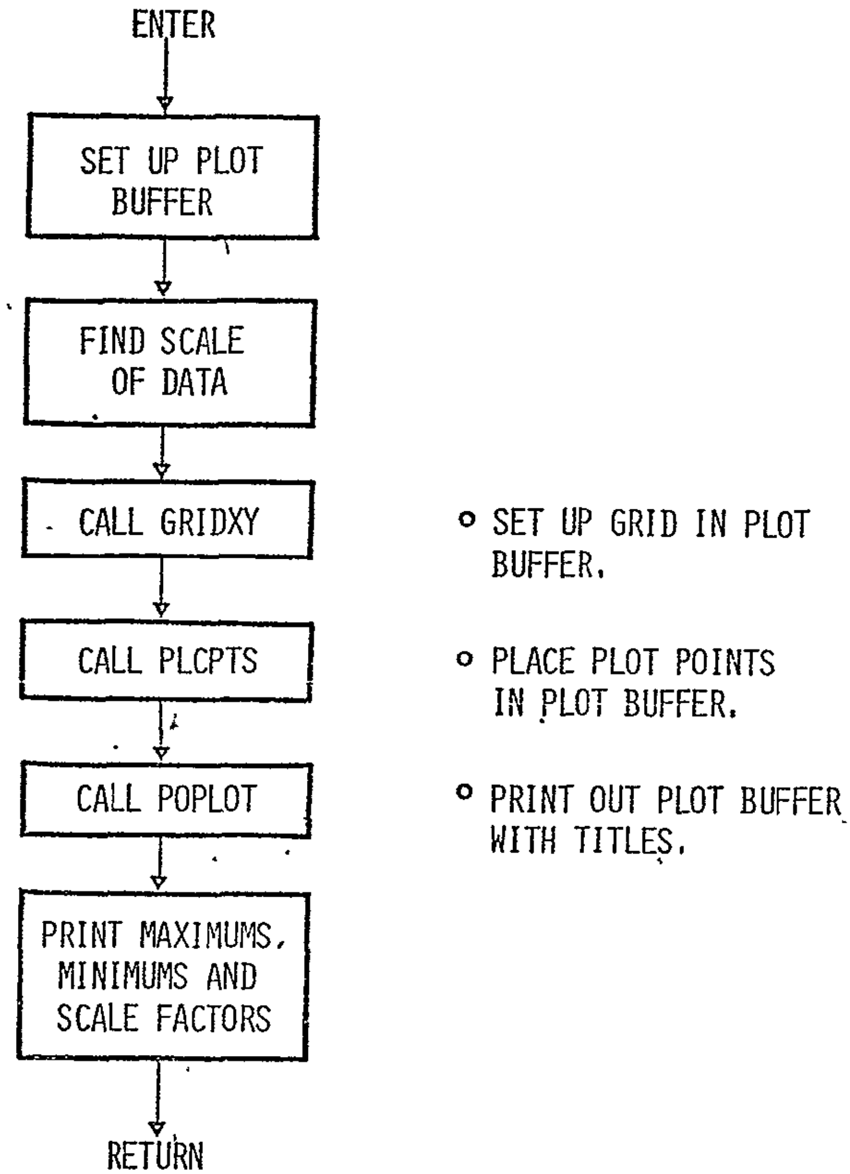


FIGURE 12.1-16. FUNCTIONAL FLOW CHART - PPLNLN

ORIGINAL PAGE IS
OF POOR QUALITY

12.1.4.18. **Subroutine SCALE**

The subroutine is used to find the minimum and maximum values for a given set of data and determines the reasonable scale values to be placed within the plot dimension. It also establishes scale factors for actually scaling the data to fit on the plot. The calling sequence is:

Call SCALE(X,S,N,K)

X — An array containing N data values which are to be scaled. The data being stored in every Kth. cell. That is, X(1) X(1+k) X(1+2k), etc.

S — The length of the plot over which the data is to be plotted in inches.

N — The number of points in the X-array.

K — The repeat cycle for data elements in the X-array.

The routine scans all of the elements in the array to find the maximum and minimum. It adjusts these values to give reasonable scale values for the start value and the maximum value. The two values are saved in the last elements of the X-array for use by the line subroutine in scaling the data for plotting. The X-array must be dimensioned two extra elements for every variable set in the array.

12.1.4.19 Subroutine READ15

This subroutine reads observation functions from the specified input unit and reorders the data into plot arrays. The subroutine contains an option to obtain the number of points as the first record on the observation function file.

12.1.4.21 Subroutine SYMBOL

Subroutine SYMBOL is the modified CALCOMP assembly language routine for converting a string (array) of alpha-numeric information into plot vector format. It works from a directory of stored plot vector sequences which represent the standard character set. The characters in the string are identified and their plot vector representation is converted to the proper format through calls to the subroutine PLTT. The user has the option of selecting the start position in terms of X and Y, the size of the characters to be used, the angle to which the data is to be placed and the number of characters to be placed. SYMBOL is called as follows:

Call SYMBOL (X,Y,SIZE,FDATA,THETA,N)

X — The X-coordinate of the left most raster unit of the first character to be plotted, inches.

Y — The Y-coordinate of the lower most raster unit of the first character to be plotted, inches.

SIZE — The height of the character, inches.

DATA — The starting location of the array of characters to be plotted.

THETA — The angle of inclination of the character string, degrees.

N — The number of characters to be plotted by the SYMBOL routine.

The character height is a variable in the subroutine but the width-to-height ratio is fixed at 4/7. Since the characters are stored in a series of bi-octal offset pairs for a 4 x 7 matrix, the reference origin for the offset pairs, which define each character, is the lower left corner of the matrix. The X and Y values define the location of the lower left hand corner of the first character to be plotted for this entry. Subsequent characters to be plotted are spaced from the previous character origin by 6/7 of the specified character heights.

12.1.4.22 Subroutine THROBS

This is the main control program for the generation of tabulated output and also serves some scaling functions for both printer plots and vector plots. The subroutine sorts the data which is stored in array format and prints it eight columns and 54 rows per page. The array title is printed at the top of each page.

12.1.5. Internal Variables Description

12.1.5.1 Common /COMON/ Description

Location	Local Name	Description
1	NUMP	Number of points per plot array.
2	ONDAT	The logical unit number for the file containing observation functions in binary format. INDAT is set in accordance with the input variable INPUT. If INPUT is 0, the observation file is not used. Otherwise, data is expected on the unit number specified in INPUT.
3	CALCOM	A logical variable. If true, vector plot file will be generated.
4	OBSERV(100)	Hollerith array containing the annotation names (left justified and blank filled) for the plot arrays.
104		Not used.
105	NOBSER	Number of observation functions for plot arrays.
106	OBSPLT(120)	An integer array containing the plotting instructions. See program input section for definition.
226		Not used.
227	REM(30)	An integer array containing the title of the plot in hollerith characters.
257		Not used.
258	XOBSTH(100)	A real array used for reading observations from the observation function file.
358		Not used.
359	NPAGE	Integer page number used for printed plots.

Location	Local Name	Description
360	SCALEF(100)	An array of scale factors used for scaling plot arrays relative to one another.
460		Not used.
461	XSIZE	The length of the X-axis in the virtual window in inches.
462	YSIZE	The length of the Y-axis in the virtual window in inches.
463	LINTYP	An integer variable defining the type of line used for the current plot. See program input for description.
464	INTEQ	An integer variable defining the plot symbol to be used.
465	STARTX	Starting location for the X-axis in the virtual window in inches.
466	STARTY	Starting location for the Y-axis in the virtual window in inches.
467	SCALEX	Scale factor of the X-axis data in the virtual window in data units per inch.
468	SCALEY	Scale factor of the Y-axis data in the virtual window in data units per inch.
469	MYX	A logical variable set by input. If true, no X-axis or scaling will be generated.
470	MYY	A logical variable set by input. If true, no Y-axis or scaling will be generated.
471		Not used.
472	NREM	Number of words used for the title.
473	REMSIZ HTEXT	Character height of the title in inches.
474	PRINTR	A logical variable set by input. If true, printer plot will be generated.

Location	Local Name	Description
475	XMESH(100)	An array of mesh points in the X-direction for contour plots.
575	YMESH(100)	An array of mesh points in the Y-direction for contour plots.
676	YCUTS(25)	An array containing desired contour values.
701	NZCUTS	The number of contour lines requested.
702	CONTOR	A logical variable set by input. If true, contour plots will be generated.
703	XMOVE	A value describing the movement of the plot origin in the X-direction after plotting.
704	YMOVE	A movement of the plot origin in the Y-direction after plotting.
705	EVRCAL	A logical variable set internally. If true, the plot device has been initialized.
706	NSKIPR	A number of records on the observation function file to be skipped before reading the data.
707	NSKIPF	A number of logical files to be skipped on the observation file before reading the data.
708	NAXIS	The number of lines incrementally displaced to be used to represent the plot axes.
709	GRID	A logical variable set by input. If true, a plot grid will be generated at one inch intervals in the virtual window.
710		Not used.
711	REWIND	Logical variable set by input. If true, the observation file will be rewound before reading the data.

Location	Local Name	Description
712	SCAL	The size of the plot device window in inches. The maximum axis length (defined by XSIZE and YSIZE) in the virtual space will be scaled to this dimension.
713	XORIGN DXG	The X-coordinate of the plot device window with respect to the plot device origin.
714	YORIGN DYG	The Y-coordinate of the plot device window with respect to the plot device origin.

12.1.5.2 **Common /ADATA/ Description**

Location	Local Name	Description
1	NADATA	The number of locations used for storing the data to be plotted. NADATA is set by a data statement in the main program PLOTTR.
2	OBSTH(2000)	The real array containing the data being plotted 2000 locations are provided for internal storage of data. This number may be altered at compile time by the adjustment of dimension statements and data statements in the main program, PLOTTR, as follows: COMMON/AESOPD/ADATA(N+1) REAL OBSTH(N) DATA NADATA/N The value of N is now 2000 but can be set to any value by the programmer. The size of the program is increased or decreased by the dimensions of this array.

TABLE OF CONTENTS FOR SECTION 13.1, PROGRAM MYPROGRAM

SECTION 13

IN LINE COMPILATION

13.1 PROGRAM MYPROGRAM: COMPILATION AT EXECUTION TIME CAPABILITY

The ODIN/MFV system has the ability to compile, store, and execute a user-designated program at execution time. The program to be compiled and its associated data form part of the normal ODIN input stream. In actuality, two ODIN programs perform the compile and execute sequence. These are "COMPILER" and "MYPROGRAM." The input stream associated with the in-line compile and execute process is as follows:

'EXECUTE COMPILER'

(Insert program to be compiled here)

$7_8{}_9$ End of File Card

'EXECUTE MYPROGRAM'

(Insert data for compiled program at this point)

$7_8{}_9$ End of File Card

The compiled program is saved in the ODIN system as MYPROGRAM and the appropriate Job Control Language (JCL) cards to execute MYPROGRAM form part of the ODIN control card data base CCDATA. There are no limitations on the program to be compiled as MYPROGRAM other than those limitations imposed by the FORTRAN compiler itself.

13.1.1 Use of Data Base Names in Source Code

The ability to compile at execution time allows programs to be generated which use data base value in the code. This is a result of the source code being examined by the DIALOG executive program prior to compilation. At this time any data base names contained in the source code are replaced by their numeric values in the normal ODIN manner as explained in Section 2.

13.1.2 Use of Overlays

The program to be compiled may contain as many overlays as permitted by the CDC 6600 loading system.

13.1.3 Separation of Compile and Execute Functions

The compile and execute functions are separated to permit multiple execution of MYPROGRAM without the necessity of a recompilation, for example,

```
EXECUTE COMPILER
EXECUTE MYPROGRAM
EXECUTE MYPROGRAM
EXECUTE MYPROGRAM
```

13.1.4 Redefining MYPROGRAM

At any point in the ODIN input stream MYPROGRAM can be redefined by new source cards, that is, by inserting the additional ODIN job control language cards

```
EXECUTE COMPILER
EXECUTE MYPROGRAM
```

with the associated source and data input cards as discussed previously. This redefinition of MYPROGRAM can occur at any point in the ODIN input stream and be repeated as often as desired by the user; for example

```
EXECUTE COMPILER
EXECUTE MYPROGRAM
EXECUTE COMPILER
EXECUTE MYPROGRAM
```

13.1.5 Multiple MYPROGRAMs

In the preliminary ODIN/MFV as installed at Wright-Patterson Air Force Base three MYPROGRAMs can be employed simultaneously in the ODIN simulation. These programs are designated

```
MYPROGRAM
MYPROGRAM2
MYPROGRAM3
```

Each of the three programs may be redefined independently or executed as discussed above.

13.1.6 Other Languages

The basic set of MYPROGRAMs are defined as FORTRAN source language statements. However, by a simple update of the ODIN control card data base, CCDATA, any or all programs could be written in any language available to the CDC 6600 computer being employed. For example, COBOL or COMPASS source codes can be readily employed as the basis for any of the MYPROGRAM's. However, the type of source code used for a given MYPROGRAM cannot be altered during an ODIN simulation at the present time.

13.1.7 MAIN PROGRAM Card

The MYPROGRAM MAIN PROGRAM cards employed *must be* in the form:

```
PROGRAM NAME(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE78)
```

Tape 78 is reserved for information which must be transferred into the ODIN data base from the compiled program by the usual NAMELIST write procedure of Section 2.

TABLE OF CONTENTS FOR SECTION 14.1, PROGRAM AUTOLAY

SECTION 14
OVERLAY CONSTRUCTION

14.1 PROGRAM AUTOLAY: AUTOMATIC OVERLAY CONSTRUCTION FOR CDC 6600 COMPUTER

In constructing a program overlay file, an Aerophysics Research Corporation developed utility program AUTOLAY is used. AUTOLAY is a user library simulation copy routine developed by Aerophysics Research Corporation to take most of the work out of building overlay or normal load files.

AUTOLAY is called by a control card and reads text cards.

14.1.1 AUTOLAY Control Card

The AUTOLAY control cards is

AUTOLAY (*OUTFILE*, *LIB1*, *LIB2*, . . ., *LIBi*)

where *OUTFILE* is the name of the disk or tape file upon which the output (the program overlay file) is to be written, and *LIB1*, *LIB2*, . . ., *LIBi* $(1 < i < 6)$ are the names of the user supplied library files containing subprograms output from a CDC 6600 compiler or assembler in relocatable object form (odd parity).

14.1.2 AUTOLAY Text Card

The order and content of the text cards define the output file. They are free form in columns 1 through 72, blanks ignored. The text cards are listed below:

ident

where *ident* is a subprogram name. The purpose of the *ident* card is to name the main program. Once the main program name is known, it and all the routines it calls or references, and all they call or reference that were available in the library files are copied onto the output file specified.

Usually only the one card naming the main program need be given except for certain cases such as Block Data routines that are necessary but not specifically called or referenced by any program. (In the case of an otherwise unnamed Block Data routine, the additional *ident* card would contain only BLKDATA). Another instance might be one in which the order of loading was important to guarantee that the longest named COMMON reference would come first. The order would be forced by the insertion of additional cards containing the names of the routines in the order required.

OVERLAY (fn, I_1, I_2)

Overlay text cards are necessary to properly define the structure of the file to be built for overlay loading. These cards cause an overlay loader directive record containing all the information on the text card to be written on the OUTFILE. The order and form of this text card must be exactly as defined in the Scope Reference Manual or the FORTRAN Reference Manual, with the exception of the starting column.

An *ident* text card containing the name of the main program in the overlay must follow each overlay text card.

Given correct *overlay* and *ident* text cards, AUTOLAY will correctly build an overlay structure file; no routine needed or defined in a more fundamental overlay will be placed in a less fundamental level. If an *ident* card incorrectly attempts to call a routine that somehow has been placed in the more fundamental level either through a previous *ident* text card of through a call by a subroutine at that level, the *ident* card is ignored, and an informative diagnostic is printed. It is possible to have several 0,0 level overlay cards in the text stream if the purpose is to build different overlay structured programs.

WEOF

This text card causes an end of file to be written on the OUTFILE after all the preceding text cards are processed. (A file mark might be between two separate overlay structured programs being output in a single run).

14.1.3 AUTOLAY Details and Limitations

For perhaps 98 per cent of all the times AUTOLAY is used, 50000_8 will be sufficient field length. AUTOLAY will abort if the following internal tables overflow:

Name of Table	Size
Library Subprogram Name	768
Subprogram Entry Points	1280
Subprogram External References	3840
Current Overlay Need Stack	383
Working Storage Buffer	variable (see below)

The working storage buffer size can be determined by subtracting 40500_8 from the field length. It is difficult to determine what the minimum size required will be unless the lengths of the relocatable binary records present on the user library files are known. The length of the longest record determines the minimum working storage buffer size. (Note: this length is not the amount of core required to load the subprogram for execution but the number of words output by the compiler or assembler). In other words, it is proportional to the number of binary cards that would be punched out, were the subprogram punched out, not *necessarily* related to the size of any arrays dimensioned inside the subprogram. This length can be obtained exactly, if necessary, from the information output by a "LIBLIST" of the library files, and should be rounded upward to the nearest 1000_8 when figuring the minimum field length necessary for AUTOLAY.

AUTOLAY rewinds each user library file starting with the first mentioned then transfers every routine contained in it to a random access file, rewinds the library file, and then repeats this process with the next user library file mentioned for every file given.

If during the transfer process a subprogram is found that has a name duplicating one found previously, the latter subprogram is skipped, an informative diagnostic printed, and the process continues. This is handily put to use when one wishes to use a newer version of a routine instead of the version contained in one of the user library files, e.g., by placing the name of the newer library file to the left of the older version, the user causes the duplicate routines on the later file to be ignored.

Entry points must be unique to one subprogram. If two or more have the same entry point names, AUTOLAY output may be scrambled. The responsibility for proper overlay text card sequence is entirely the user's. Incorrect sequencing as defined in the Scope and FORTRAN Reference manuals will not be flagged until an attempt is made to load the OUTFILE.

The OUTFILE is rewound at the beginning and end of AUTOLAY. It will be ended with one end of file mark unless more are forced through *WEOF* cards at the end of the text cards.

The random access file mentioned earlier is called RANSCR and *must* be a disk file; however, at the conclusion of AUTOLAY it can be rewound and copied by the normal control cards (REWIND and COPYBF) if the user wishes to save a new version of the user library. This file contains all of the routines found in the library files input to AUTOLAY minus any duplicate routines, overlay cards, and compiler or assembly error records.

The present version does not allow the use of INPUT (the card reader) as a library file.

14.1.4 AUTOLAY Examples

Example 1. The initial installation of AUTOLAY as a permanent file:

```
RFL,60000.
FTN.
LOAD(LGO)
NOGO.
CATALG(AUTOLAY,AUTOLAY,ID=ARCLIB01,EX=ARC1,
        CN=ARC1,MD=ARC1,RP=999)
```

end of record

AUTOLAY source deck

end of record

Example 2. The initial installation of the program overlay file

The following deck set up is used for the initial generation of the program overlay file NEWPGM on tape ARC01.

```
REQUEST NEWPGM,HI.     (ARC01/RING)
RFL,60000.
RUN(S,,,,,,77000)
ATTACH(AUTOLAY,AUTOLAY)
AUTOLAY(NEWPGM,LGO)

    FORTRAN source decks

 OVERLAY(PROGRAM,0,0)
 MAIN
 OVERLAY(PROGRAM,1,0)
 MAIN1
 OVERLAY(PROGRAM,2,0)
 MAIN2             +
 OVERLAY(PROGRAM,3,0)
 MAIN3
 OVERLAY(PROGRAM,4,0)
 MAIN4
 OVERLAY(PROGRAM,5,0)
 MAIN5
 OVERLAY(PROGRAM,6,0)
 MAIN6
 OVERLAY(PROGRAM,7,0)
 MAIN7

 end of record
 end of file
```

Example 3. Modification of the program overlay file.

The following deck set up is used when making modifications to the program:

```
REQUEST OLDPGM,HI.    (ARC01/NORING)
REQUEST NEWPGM,HI.    (ARC02/RING)
RFL,60000.
RUN(S,,,,,,77000)
ATTACH(AUTOLAY,AUTOLAY)
AUTOLAY(NEWPGM,LGO,OLDPGM)
```

end of record

Modified source decks

end of record

```
OVERLAY(PROGRAM,0,0)
MAIN
OVERLAY(PROGRAM,1,0)
MAIN1
OVERLAY(PROGRAM,2,0)
MAIN2
OVERLAY(PROGRAM,3,0)
MAIN3
OVERLAY(PROGRAM,4,0)
MAIN4
OVERLAY(PROGRAM,5,0)
MAIN5
OVERLAY(PROGRAM,6,0)
MAIN6
OVERLAY(PROGRAM,7,0)
MAIN7
```

end of record

end of file

TABLE OF CONTENTS FOR SECTION 15.1,

PROGRAM AFSP

SECTION 15

AEROELASTICITY

ODIN aeroelasticity computations are currently limited to use of the AFSP program. This program was constructed by Y. T. Phoa during the study effort. The AFSP program contains the following computational capability:

1. Normal modes using swept strips
2. Unsteady compressible subsonic aerodynamic strip theory
3. Flutter equation solution using either conventional V-g analysis or an automated Nyquist-like search for flutter speed.

An outline of the AFSP procedure is presented below. Additional details may be found in Reference 1. It should be noted that more sophisticated flutter analysis technology modules are readily available for inclusion in the ODIN system, for example, Reference 2.

REFERENCES:

1. Phoa, Y. T., A Computerized Flutter Solution Procedure, Paper presented at the National Symposium on Computerized Structural Analysis and Design, Washington, D.C., 1972.

2. Albano, E., Perkinson, F., and Rodden, W. P., Subsonic Lifting Surface Theory Aerodynamics and Flutter Analysis of Interfering Wing/Horizontal Tail Configurations, Part 2, Wing-Tail Flutter Correlation Study, AFFDL-TR-70-59, 1970.

15.1 PROGRAM AFSP: A PROGRAM FOR AUTOMATED FLUTTER SOLUTION

15.1.1 Normal Modes

Normal modes for a multi-segment swept wing are computed within the AFSP code. Uncoupled bending and torsion modes are considered with the fuselage mass represented by concentrated mass and inertia terms. Modes may be obtained for any of the following boundary condition assumptions:

1. Cantilever wing

2. Free-free antisymmetric motion

3. Free-free symmetric motion

The wing idealization is presented in Figure 1. A series of streamwise oriented panels are attached to a swept elastic axis. The root point of the elastic axis lies on the center line of the airplane and coincides with the origin of the reference axes of the whole airplane.

Panel inertia is accounted for by means of the following data:

m = panel mass
A = streamwise distance between the elastic axis attachment point and the center of gravity
I_1 = moment of inertia about axis 1
I_2 = moment of inertia about axis 2

Degrees of freedom associated with the inertia data are

k_3 = vertical translation of the panel's center of gravity
$\bar{k}_1$ = panel torsion about axis 1
$\bar{k}_2$ = panel rotation about axis 2

Flexibility of the elastic axis is specified by EI (bending) and GJ (torsion) stiffness values at the successive reference points on the elastic axis. A value at reference point (i) is used as the constant stiffness distribution between (i) and (i-1). The length of the corresponding beam element is denoted by L(i); the root point of the elastic axis can be considered the zero point.

Variables that specify the elastic axis deformation at these reference points are

h = vertical displacement due to bending
r = rotation (slope) due to bending
t = rotation due to torsion

For the free-free airplane, mass matrices are required which include the inertia data of the whole airplane as well as the inertia coupling terms which relate the motions of the airplane reference axes to the motions of the respective panels relative to those axes. For anti-symmetric airplane motions only the roll degree of freedom needs to be considered for a meaningful wing flutter analysis. The symmetric airplane motions are accounted for by the vertical translation A and the pitch degree of freedom θ. The airplane data are specified in terms of

$M0_r$ = mass of airplane

S_r = static moment of the airplane about the Y axis

$I1_r$ = moment of inertia of the airplane with respect to the X axis

$I2_r$ = moment of inertia of the airplane with respect to the Y axis

These quantities do not include the wing inertias. The wing contributions are computed from the panel inertias and the additional input data

x = distance from the panel's center of gravity to the airplane's Y axis

y = distance from the panel's center of gravity to the airplane's X axis

Transformations convert the input data into mass and stiffness matrices relevant to "uncoupled" bending and torsion vibration equations for the elastic axis. The general format of the vibration equations is

$$(-\omega^2 M + K)q = 0 \qquad (15.1.1)$$

Because of the simplicity of the mass matrices concerned, Equation (15.1.1) is rewritten

$$Kq = \lambda Mq \qquad \text{with} \quad \lambda = \omega^2$$

or

$$K(M^{-1/2})(M^{1/2})q = \lambda(M^{1/2})\,(M^{1/2})q$$

or

$$(M^{-1/2})K\,(M^{-1/2})(M^{1/2})q = \lambda(M^{1/2})q$$

or

$$\bar{A}\xi = \lambda\xi \qquad (15\ 1.2)$$

with

$$A = (M^{-1/2})K\,(M^{-1/2}) = \text{symmetric axis}$$

and

$$\xi = (M^{1/2})q \qquad \text{or} \qquad Q = (M^{-1/2})\xi$$

The eigenvalue problem then is solved by means of Jacobi's iterative algorithm which fully exploits the symmetry of the A matrix.

The flutter equations are formulated as follows:

$$\{-\omega^2 M + S - \frac{1}{2} \rho V^2 C\}X = 0 \qquad (15.1.3)$$

with
M = mass matrix
S = stiffness matrix
C = matrix of aerodynamic coefficients, evaluated for given Mach number and $k = \omega b/V$
ω = vibration frequency
V = speed of flight
ρ = air density
b = reference length, required to render k dimensionless
X = system degrees of freedom

The solution of these equations consists of a set of (ω, ρ, V) values which satisfies the following conditions:

a. consistent with the k-value and Mach number for which the C-matrix has been evaluated

b. render the matrix of coefficients of (11.1.3) equal to zero.

15.1.2 The V-g Technique

The flutter equations of (11.1.3) present an eigenvalue problem. When compressibility of air can be ignored or when the Mach number is fixed, the ρ-value is constant and the C matrix depends only on k. Equation (15.1.1) can then be rewritten as follows:

$$\{A-\lambda K\}X = 0 \qquad (15.1.4)$$

with

$$A = M + \frac{1}{2} \rho (b/k)^2 C$$

$$\lambda = 1/\omega^2$$

For selected k values, the eigenvalues of Equation (15.1.4) are found. The corresponding flutter speed follows directly from

$$V_F = \omega b/k \qquad (15.1.5)$$

Due to the fact that the C matrix is complex and non-Hermitian, however, arbitrarily assumed k values, in general, lead to complex eigenvalues.

In this general case, the flutter equation is rearranged so that

$$\lambda = \left(\frac{1.0}{\omega}\right)^2 (1+jg) \tag{15.1.6}$$

Here λ is the system frequency and g is the artificial structural damping which maintains a simple harmonic motion.

$$\lambda = R + jI \tag{15.1.7}$$

The system frequency corresponding to this eigenvalue is

$$\omega = \frac{1.0}{\sqrt{R}} \tag{15.1.8}$$

and the corresponding velocity is

$$V = \frac{1.0}{\sqrt{R}} \ b/k \tag{15.1.9}$$

The artificial structural damping is given by

$$g = \omega^2 I \tag{15.1.10}$$

Plotting V versus g for varying k points at which sustained harmonic motion is possible without artificial (negative) structural damping can be found. Figure 15.1-2 illustrates a typical result for a four degree of freedom system. At the flutter point a real eigenvalue is found. This eigenvalue satisfies Equation (15.1.4). In actuality, structural damping is a small positive quantity, $g \approx .02$.

15.1.3 Automated Flutter Solution Procedure

Returning to the flutter equations of (15.1.4) a direct and fully automated search can be carried out for those values of ω, ρ and V which satisfy all the flutter conditions. Such a search has been organized for the AFSP program. The objective is to determine the flutter parameter values at a given Mach number. A ρ-value can be computed when a V-value is specified. When, in addition, an ω-value is chosen, the k-value, the C matrix and the complete flutter matrix can be evaluated. The search domain scanned by the AFSP can be depicted in a V-ω diagram as shown in Figure 3. The boundaries of this domain are established on the basis of practical considerations as follows.

Because the Mach number is fixed, the lower speed limit is set by the highest altitude for which the airplane is designed. The upper limit has to correspond with an appropriate speed at sea level.

It is known from experience that flutter speeds of practical interest occur within a limited frequency interval and that bandwidth is usually determined during the early stages of the design effort.

The lines of constant k value are straight lines in the V-ω diagram. The highest k-value is determined by the procedure that is used to compute oscillatory airforces. Those procedures are numerical and the obtainable accuracy decreases with increasing k-value; a ball park figure is

$$k_{max} = 3$$

It should be emphasized that the extent of the search can be reduced considerably when parameter studies are carried out and regions of particular interest have been determined from a few initial computations.

The objective of the search is to find those points within the search domain at which the flutter matrix has a zero determinant value. This leads to a direct link between the flutter analysis and the theory of linear control systems. To establish that link, the flutter equations are first rederived along the following lines. The mass and stiffness properties of the airplane structure can be represented by means of a transfer function relationship $X = G\varepsilon$ with

$G^{-1} = (j\omega)^2 M+K$
X = column-vector of airplane degrees of freedom
ε = column-vector comprising the forces acting in the respective airplane degrees of freedom

Assuming a given flight speed, V, the C matrix evaluated for a specified Mach number provides a similar relationship: $F = HX$ with

$H = \frac{1}{2} \rho V^2 C$
F = column-vector comprising aerodynamic forces due to motions and acting in the respective airplane degrees of freedom

Referring to Figure 15.1-4 block diagram algebra yields the transfer function relationship for the closed loop feedback system

$$(G^{-1} - H)X = ZX = Y \qquad (15.1.11)$$

By definition, flutter vibrations occur without the presence of external excitations Y; i.e., $Y \equiv 0$.

Reference now is made to Nyquist's technique for determining relative stability of a feedback system. To apply that technique, it is required to derive the expression for the open loop frequency response. Open loop frequency responses for multivariable systems are derived from the general relation:

$$X = Z^{-1}Y \tag{15.1.12}$$

in which the X and Y column vectors are related in such a manner that the (k,j) element of the Z^{-1} matrix represents the closed loop frequency response in degree of freedom (i) to excitation with unit amplitude in the j^{th} degree of freedom. The elements of the Z^{-1} matrix are further related to those of the Z matrix as follows

$$t_{(i,j)} = Z^{-1}_{(i,j)} = (-1)^{i+j}\{\Delta_{ji}/\Delta\} \tag{15.1.13}$$

with

Δ = determinant of the Z matrix

Δ_{ji} = minor of the (j,i) element of the Z matrix

Expansion of Δ as below:

$$\Delta = (-1)^{i+j} Z_{ij}\Delta_{ij} + \zeta_{ij} \tag{15.1.14}$$

then defines ξ_{ij}, and t(i,j) can now be expressed as the closed loop transfer function of a single loop system as shown in Figure 11.1-5.

$$t(i,j) = \frac{(-1)^{i+j}\Delta_{ji}/\zeta_{ij}}{1+(-1)^{i+j}Z_{ij}\Delta_{ij}/\zeta_{ij}} \tag{15.1.15}$$

Referring to textbooks on control systems theory, the second term in the denominator is identified as an open loop frequency response or Nyquist function

$$N(i,j) = (-1)^{i+j}Z_{ij}\Delta_{ij}/\zeta_{ij} \tag{15.1.16}$$

The Nyquist function computed in the AFSP is the direct, open loop frequency response in the n^{th} degree of freedom such that

$$N = Z_{nn}\Delta_{nn}/\xi_{nn} \tag{15.1.17}$$

with n equal to the total number of degrees of freedom.

It is evident that the condition for N to be equal to (-1) applies to both the concept of neutral stability in control theory as well as to flutter considering that in the case

$$\Delta = Z_{nn} \Delta_{nn} + \xi_{nn} = 0 \tag{15.1.18}$$

The standard procedure in control theory for determining relative stability of a feedback system consists of a mapping of the so-called s plane onto another image plane. When the Nyquist function is used as the mapping function, the image of the (J) axis of the s plane occurs as a Nyquist locus and all characteristic roots of the system are mapped simultaneously onto the (-1) point of the image plane. The AFSP search consists of frequency sweeps at constant V-value. Each sweep yields a Nyquist locus as shown in Figure 6. A logic has been built into the program to change V values automatically in such a manner that the locus is forced to pass through (-1).

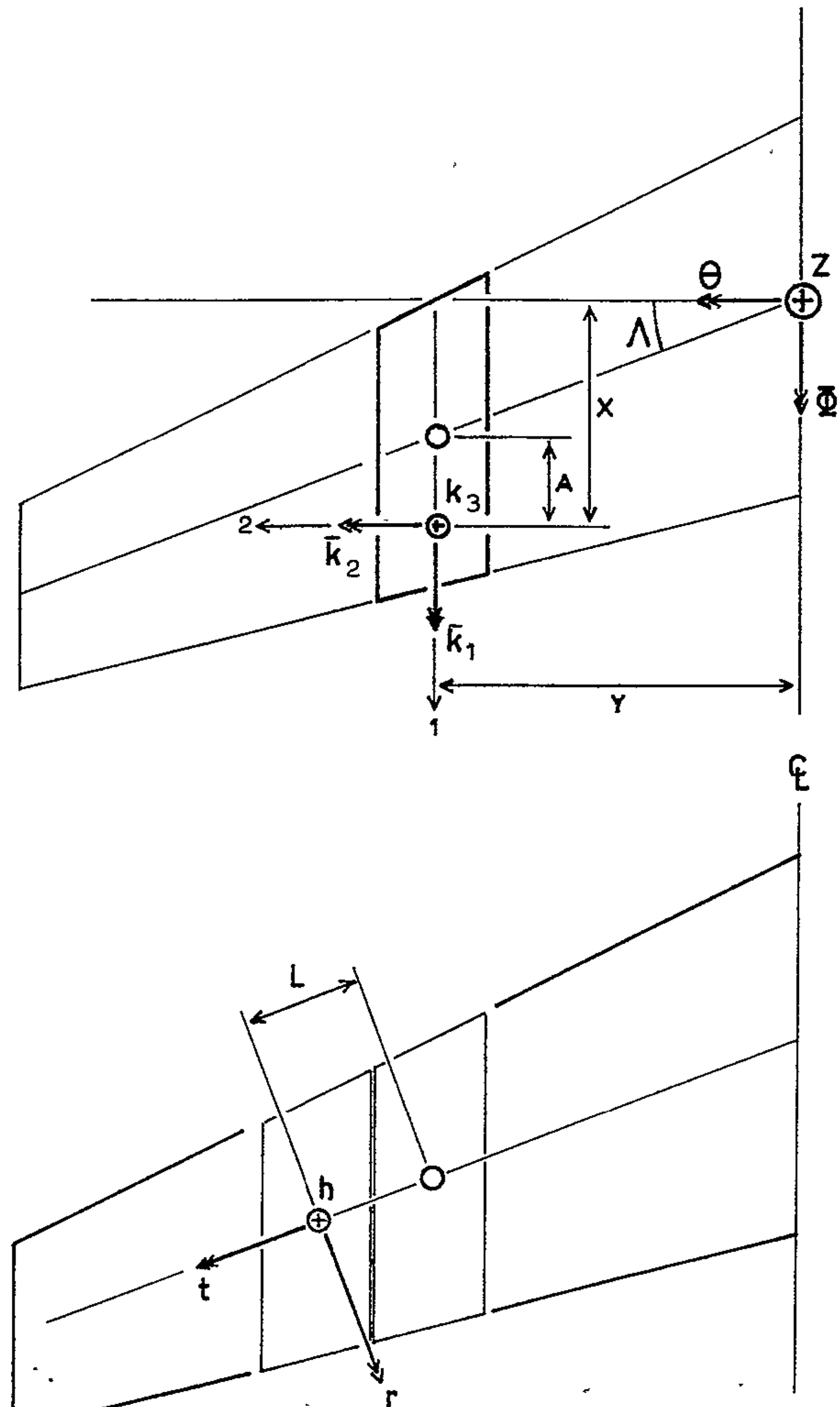


FIGURE 15.1-1

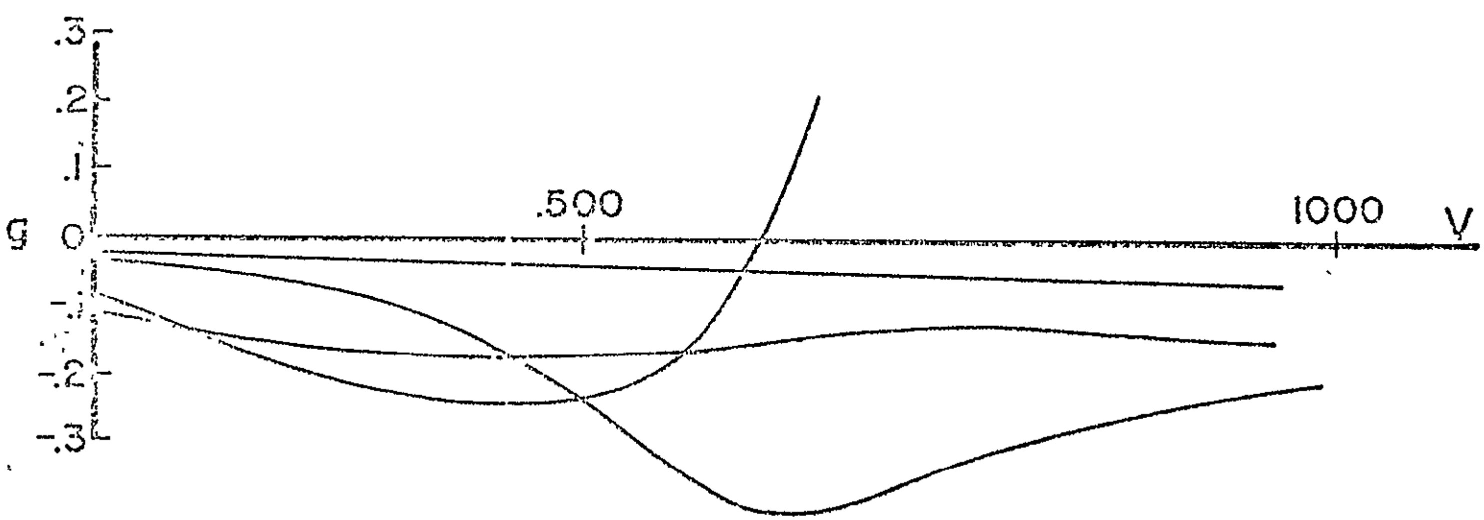


FIGURE 15.1-2 TYPICAL V-g PLOT

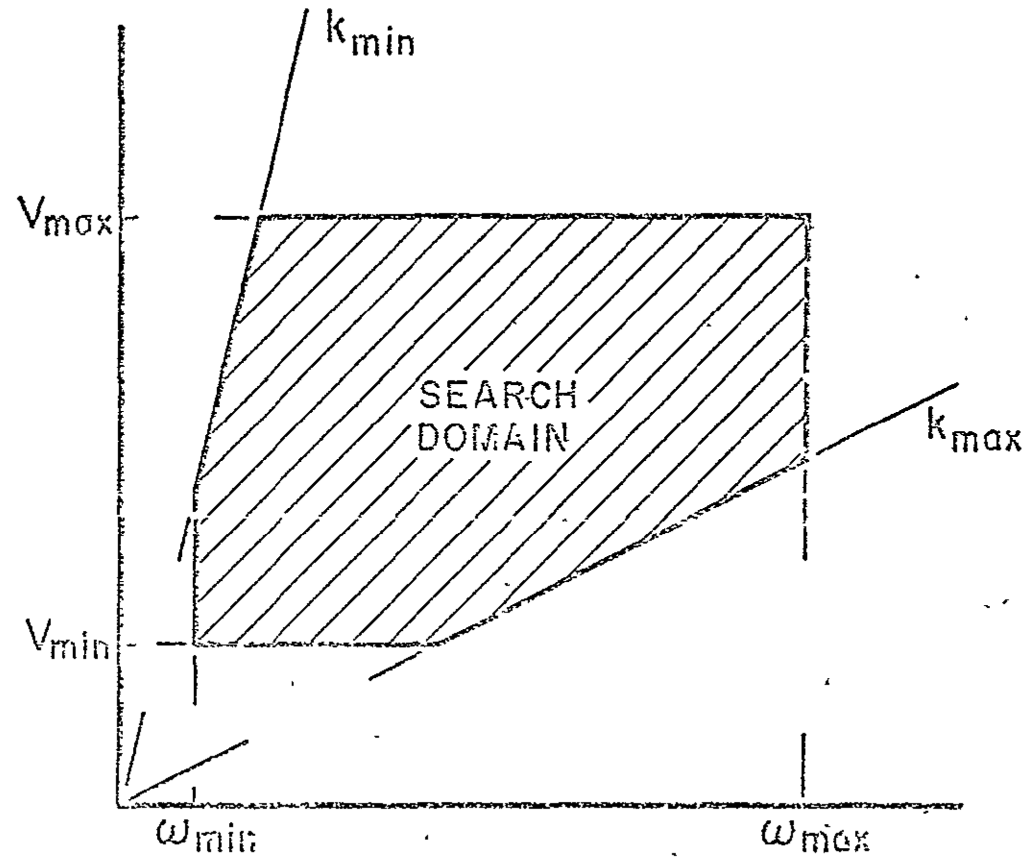
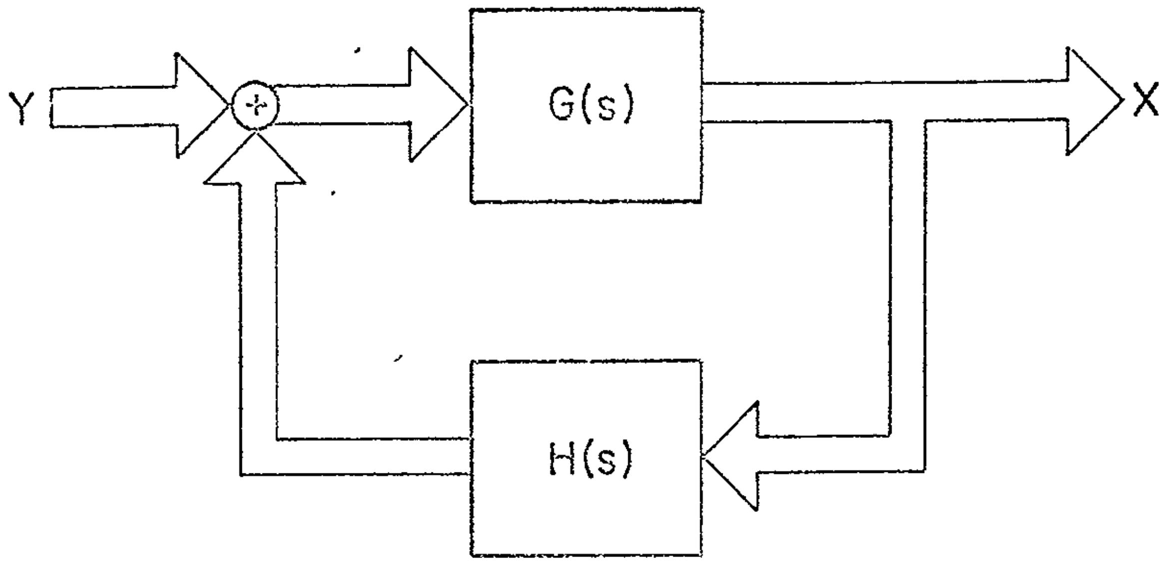


FIGURE 15.1-3 AFSP SEARCH DOMAIN

$$G^{-1} = (j\omega)^2 M + K$$
$$H = \tfrac{1}{2}\rho V^2 C$$



G(s) = matrix of structural transfer functions

H(s) = matrix of aerodynamic transfer fcns

X = set of generalized coordinates

Y = set of generalized, external excitation forces

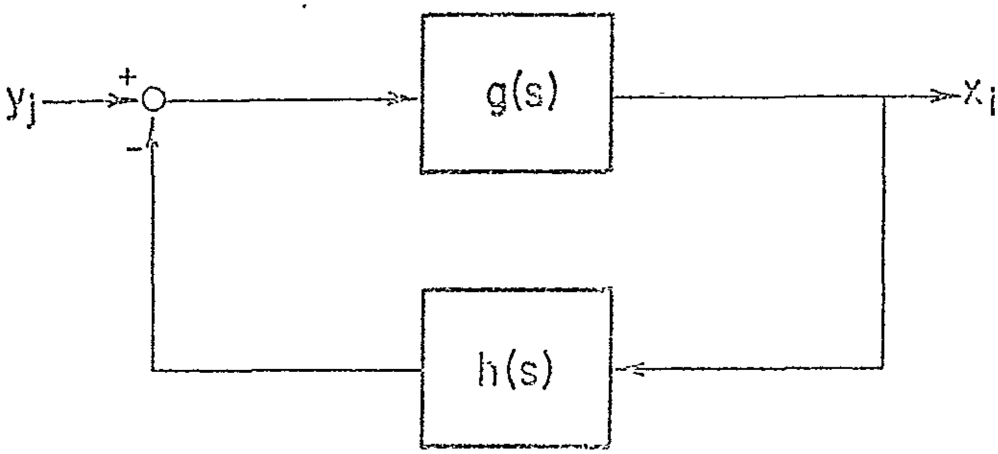
$$(G^{-1} - H)X = ZX = Y$$

FIGURE 15.1-4 THE AEROELASTIC SYSTEM

$$x_i/y_j = t(i,j) = \frac{g(s)}{1 + g(s).h(s)}$$

$$g = (-1)^{i+j}\Delta ji/\xi ij$$

$$h = Z ij \Delta ij/\Delta ji$$



$$\left.\begin{array}{lll} x_i & = & \text{i-th output variable} \\ y_j & = & \text{j-th input variable} \\ t(i,j) & = & \text{closed loop transfer fcn} \end{array}\right\} \begin{array}{c}\text{ORIGINAL} \\ \text{MULTI-VARIABLE} \\ \text{SYSTEM}\end{array}$$

FIGURE 15.1-5 SINGLE LOOP SUBSTITUTE SYSTEM

C-8

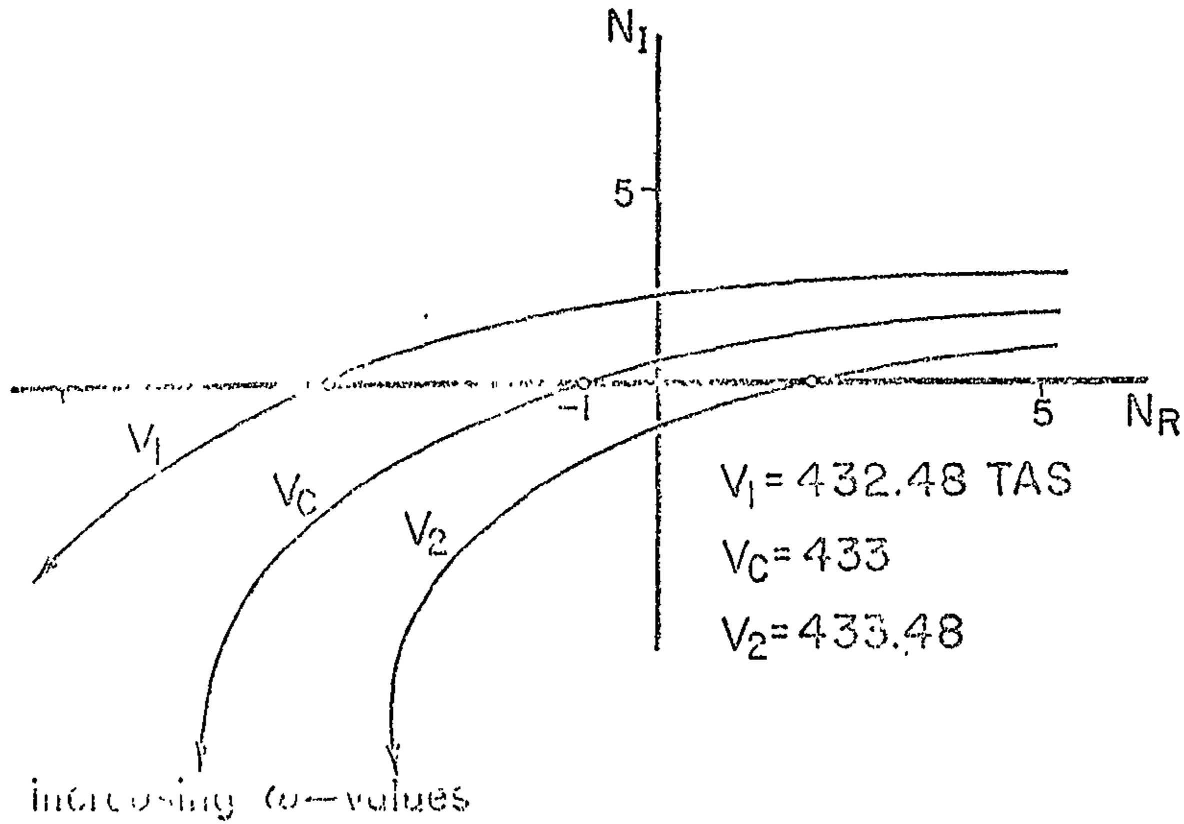


FIGURE 15.1-6 N-LOCI USED IN AFSP

TABLE OF CONTENTS FOR SECTION 16.1, PROGRAM ACMOTAN

SECTION 16

STABILITY AND CONTROL

The ODIN system contains a closed loop stability and control analysis technology module, ACMOTAN, Reference 1. ACMOTAN is an acronym for linear aircraft motion analysis, and the program was constructed by William B. Kemp and Charles H. Fox, Jr., of Langley Research Center.

It should be noted that if nonlinear analyses are required, the Air Force Flight Dynamics Laboratory's six degree of freedom program, References 2 and 3, can be readily introduced as a technology module. The six degree of freedom program has compatible input with the Section 7.3 ATOP II program.

REFERENCES:

1. Kemp, William B. and Fox, Charles H. Jr., Users Guide to Program ACMOTAN for Linear Aircraft Motion Analysis, Program No. A2541

2. Brown, R. C., Brulle, R. V., Combs, A. E. and Griffin, G. D., Six Degree of Freedom Flight Path Study Generalized Computer Program, Part 1, Volume I, AFFDL-TDR-64-1, October 1964.

3. Vorwald, R. L., Six Degree of Freedom Flight Path Study Program Generalized Computer Program, Part 2, Volume I, AFFDL-TDR-64-1, October 1964.

16.1 PROGRAM ACMOTAN: LINEAR AIRCRAFT MOTION ANALYSIS

Program ACMOTAN was developed Kemp and Fox of Langley Research Center. The description below is taken directly from their internal Langley note. Since the original program documentation is not generally available, the description below includes program input description from the program authors.

16.1.1 General Description

Program ACMOTAN is a versatile program for linear aircraft motion analysis which allows the user to supplement standard airplane equations of motion with auxiliary equations written by the user to represent control laws or additional variables. The program prepares the system of linear differential equations using several optional forms of input data and then carries the solution to an extent determined by the output options selected. Minimum output includes the *characteristic polynomial and its roots.* Additional outputs in the form of *transfer functions, frequency responses and time histories* can be selected.

The basic equations operated on by the program represent a set of Laplace transformed linear differential equations and can be expressed in matrix form as

$$[A]\{X\} = [\gamma]\{d\} \qquad (16.1.1)$$

where

[A] is the coefficient matrix of size n x n having polynomial elements of degree k. Program limitations require $n \leqslant 12$ and $k \leqslant 2$; therefore each element of [A] is represented by up to three polynomial coefficients

{X} is a vector of n variables which can be supplemented by n' additional variables, each representing a linear combination of any of the n basic variables; n + n' must not exceed 14.

{d} is a disturbance vector of m elements implied by the user. Each element of {d} defines an implied disturbance such as a set of initial conditions, a control deflection, an external gust, etc. $m \leqslant 10$.

[Y] is the disturbance coefficient matrix having n x m polynomial elements of degree ℓ. $n \leqslant 14$, $m \leqslant 10$, $\ell \leqslant 2$.

The solution of Equation (16.1.1) can be expressed as follows:

$$\{X\} = [A]^{-1}[\gamma]\{d\} = \frac{[\tilde{A}][\gamma]\{d\}}{P} = \frac{[G]\{d\}}{P} \qquad (16.1.2)$$

[G] is the (n + n') x m matrix of transfer function numerators having polynomial elements of maximum degree $k(n - 1) + \ell$.

Inputs to the program consist of control codes and data from which the elements of [A] and [Y] are generated. Either of two standard [A] submatrices can be generated by the program. These represent the longitudinal (variables u, α,q, θ, δ) or lateral-directional (variables β, p, r, ϕ, δ) equations of motion for unaugmented airplanes, linearized about a reference flight condition with arbitrary V_0, α_0, θ_0, and q_0. The use of either submatrix requires the following minima, $n \geqslant 5$, $k \geqslant 1$, $m \geqslant 2$, $\ell \geqslant 0$. The standard submatrices are formulated on body axes as shown in Figure 16.1-1 and 16.1-2. The program will accept several optional forms of aerodynamic data for computing the standard submatrix elements. Auxiliary equations are input by reading in the K+1 polynomial coefficients at an arbitrary number of [A] element locations. This process can also be used to modify the standard submatrix equations since the coefficients read in at this stage are added to those already existing in that element location. If the complete [A] matrix has a zero determinant, the program will terminate after printing zeros for the characteristic polynomial.

For each standard submatrix of [A], a standard submatrix of [γ] is provided. The first column (corresponding to the first element of the implied {d} vector) represents a unit impulsive δ input. The second column represents initial conditions of the pertinent perturbation variables in the standard [A] submatrix. The [γ] matrix can also be established or expanded by reading in the $\ell + 1$ polynomial coefficients in any specified [Y] element location. The [Y] matrix is filled only if transfer function or time history outputs are selected.

Any list of polynomial coefficients (either input or output) is considered in order of increasing powers.

The basic output consists of the coefficients and roots of the characteristic polynomial. These computations are performed in double precision. Extremely large roots are ignored and extremely small real or imaginary parts of roots are set to zero. Roots are printed in both *complex form* and *Bode form* (τ, ξ, ω_n).

If *transfer function* or *time history* outputs are selected, the variable list can be expanded to include any new variable that can be expressed as a linear combination of any previously established variables. The number of rows in [G] is thereby expanded to a maximum of 14. The polynomial coefficients of each element of [G] (each transfer function numerator) are then printed, and the numerator roots are extracted and printed in complex and Bode form for all variables identified by an index list read in.

If time history output is selected, time histories are printed for all variables in the above index list. Values of total time duration and time increment between points must be supplied along with a list of factors by which the output variables are multiplied before testing or printing. The total duration can be broken into an arbitrary number of segments. The first segment gives the response to any linear combination of disturbances implied in the {d} vector imposed at time zero. This segment terminates when the value of the variable identified by a test variable index (must be contained in the above index list) crosses a supplied test value. At this time, a new combination of disturbances is superimposed on those previously considered, and responses are calculated until the new test variable crosses the new test value. If for any segment, the test variable index is given as zero, that segment will continue for the remainder of the specified duration. Terminal times for each segment are determined by linear interpolation within the specified time interval. There is no assurance that all segments will be entered before the total time duration is reached. A program option provides the opportunity to check the [A] matrix for poor conditioning. Since

$$[A]^{-1} = \frac{[\tilde{A}]}{P},$$

then

$$[A]^{-1}[A] = [I] = \frac{[\tilde{A}][A]}{P}$$

Thus, $[\tilde{A}][A]$ should equal P times the identity matrix. If the check option is selected, the product $[\tilde{A}][A]$ is formed, and the diagonal elements are printed just prior to the P coefficients. The accuracy of the matrix inversion may then be judged by comparison.

16.1.2 Equations of Motion

The usefulness of program ACMOTAN depends to a large extent on the ability of the user to build up an effective system of equations of motion using the standard equations contained in the program as a basis. For this reason, a clear understanding of the standard equations is necessary.

The standard force and moment equations (see Figures 16.1-1 and 16.1-2) are resolved on body axes so that control laws involving feedbacks from body-mounted accelerometers or gyros can be formulated more easily. The angular attitudes, θ and ϕ, however, are Euler angles. In linearized form the Euler pitch attitude, θ, is identical to the integral of the pitching velocity, q, and therefore should cause no confusion. The Euler bank angle, ϕ, linearized about a level flight condition (i.e., $\alpha_o = \theta_o$) is less than the integral of the stability axis rolling velocity by the factor $\cos \theta_o$. Since θ_o is a constant, the poles and zeros of the transfer functions of ϕ are identical to those of the integral of stability axis rolling velocity and therefore are directly applicable in handling qualities analysis. The generalized control deflection is introduced by including the control derivatives as coupling terms in the force and moment equations and introducing the equation, $\delta(s) = 1$, to represent a unit impulsive control deflection input. Other input forms require modification of this equation; for example, a unit step input is represented by $\delta(s) = 1/s$. To be compatible with the ACMOTAN requirement for polynomial matrix elements, this equation must be multiplied by s yielding $s\delta(s) = 1$. Thus, the standard [A] submatrix can be altered to provide a step input by adding s - 1 to A(5,5) where the -1 cancels the 1 already existing in A(5,5). Similarly, a ramp input is provided by adding s^2-1 to A(5,5).

The standard longitudinal and lateral directional submatrices each represent the Laplace transformation of the three degree of freedom perturbation equations obtained by subtracting the reference equations from the set of equations of motion linearized about a reference flight condition having arbitrary V_o, α_o, θ_o, and q_o. The reference equations obtained by setting all perturbation variables in the complete linearized equations to zero are given below.

$$X_o = \frac{\left(T_o \cos \xi - \frac{1}{2} \rho V_o^2 S C_{A_o}\right)}{mV_o} = q_o \sin \alpha_o + \frac{g}{V_o} \sin \theta_o$$

$$Z_o = \frac{\left(-T_o \sin \xi - \frac{1}{2} \rho V_o^2 S C_{N_o}\right)}{mV_o} = -q_o \cos \alpha_o - \frac{g}{V_o} \cos \theta_o$$

$$M_o = \frac{\left(T_o Z_T \cos \xi + \frac{1}{2} \rho V_o^2 S\bar{c} C_{m_o}\right)}{I_y} = 0$$

$$Y_o = \frac{\frac{1}{2} \rho V_o^2 S C_{y_o}}{mV_o} = 0$$

$$L_o = \frac{\frac{1}{2} \rho V_o^2 S b C_{l_o}}{I_x} = 0$$

$$N_o = \frac{\frac{1}{2} \rho V_o^2 S b C_{n_o}}{I_z} = 0$$

To insure a valid set of perturbation equations the input data to the program must be compatible with the reference equations.

The standard equations may be supplemented by auxiliary equations defining control laws or additional variables. The following equations express several commonly used perturbation variables in a linearized form compatible with the standard equations.

16.1.2.1 Normal Load Factor at Specified Point

Normal load factor at a point x feet ahead of CG.

$$n_z + \frac{V_o}{g}(s \sin \alpha_o - q_o \cos \alpha_o)u + \frac{V_o}{g}(s \cos \alpha_o + q_o \sin \alpha_o)\alpha$$

$$- \left(\frac{sx}{g} + \frac{V_o}{g} \cos \alpha_o\right)q + (\sin \theta_o)\theta = u_{t_o} \frac{V_o}{g} \sin \alpha_o + \alpha_{t_o} \frac{V_o}{g} \cos \alpha_o - q_{t_o}\frac{x}{g}$$

16.1.2.2 Lateral Acceleration at Specified Point

Lateral acceleration at a point x feet ahead of and z feet below CG, g units

$$a_y - \left(s \frac{V_o}{g}\right)\beta + \left(\frac{sz}{g} + \frac{V_o}{g} \sin \alpha_o\right)p - \left(\frac{sx}{g} + \frac{V_o}{g} \cos \alpha_o\right)r + (\cos \theta_o)\phi$$

$$= - \beta_{t_o} \frac{V_o}{g} + p_{t_o}z - r_{t_o}x$$

16.1.2.3 Altitude Perturbation

$$sh + V_o(\sin \alpha_o \cos \theta_o - \cos \alpha_o \sin \theta_o)u + V_o(\cos \alpha_o \cos \theta_o + \sin \alpha_o \sin \theta_o)\alpha$$

$$- V_o(\cos \alpha_o \cos \theta_o + \sin \alpha_o \sin \theta_o)\theta = h_{t_o}$$

16.1.2.4 Heading Perturbation Variable

Euler heading angle

$$(s \cos \theta_o)\psi - r - q_o\phi = \psi_{t_o} \cos \theta_o$$

16.1.3 Filter Networks and Actuator Dynamics

Control laws involving filter networks or actuator dynamics might require some manipulation to be compatible with the requirements of ACMOTAN. For example, a signal flow through a fourth order filter is represented by

$$X_1 \longrightarrow \boxed{\frac{a_o + a_1 s + a_2 s^2 + a_3 s^3 + a_4 s^4}{b_o + b_1 s + b_2 s^2 + b_3 s^3 + b_4 s^4}} \longrightarrow X_3$$

An equivalent representation is

$$X_1 \longrightarrow \boxed{\frac{c_o + c_1 s + c_2 s^2}{d_o + d_1 s + d_2 s^2}} \xrightarrow{X_2} \boxed{\frac{e_o + e_1 s + e_2 s^2}{f_o + f_1 s + f_2 s^2}} \longrightarrow X_3$$

The following equations describe this signal flow in a form suitable for use with ACMOTAN:

$$(c_o + c_1 s + c_2 s^2)X_1 - (d_o + d_1 s + d_2 s^2)X_2 = 0$$

$$(e_o + e_1 s + e_2 s^2)X_2 - (f_o + f_1 s + f_2 s^2)X_3 = 0$$

16.1.4 Program Definitions

MOVE	A seven digit control code
MOVE (1)	1 for longitudinal submatrices 2 for lateral directional submatrices 3 no standard submatrices
MOVE(2)[a]	0 for body axis data 1 for stability axis data
MOVE(3)[a]	0 for dimensional derivatives 1 for nondimensional derivatives 2 for primed dimensional derivatives (used only with MOVE(1) = 2)
MOVE(4)[a]	0 for α, β, and δ derivatives per radian 1 for α, β, and δ derivatives per degree
MOVE(5)	0 no check matrix 1 computes check matrix

[a] Irrelevant if MOVE(1) = 3

MOVE(6)	0 characteristic roots only 1 characteristic roots and transfer functions 2 characteristic roots, transfer functions and time histories
MOVE(7)[b]	0 no standard disturbance submatrix 1 for standard disturbance submatrix
N	number of rows and columns in A and number of rows in Υ. (number of equations) $N \leqslant 12$; if MOVE(1) = 1 or 2, $N \ngtr 5$.
K	maximum degree of polynomial elements in A. $K \leqslant 2$; if MOVE(1) is equal to 1 or 2, K is equal to or greater than 1.
NADD	number of A elements into which added terms are to be read
M	number of columns in Υ (number of disturbances. $M \leqslant 10$; if MOVE(7) = 1, $M \geqslant 2$
L	Maximum degree of polynomial elements in Υ. $L \leqslant 2$; if MOVE(7) = 1, $L \geqslant 0$
NGADD	number of Υ elements into which added terms are to be read
NVCOMB	number of new variables formed from linear combinations of variables represented in A. (N + NVCOMB) $\leqslant 14$
b	reference span
$\bar{c}$	reference chord
C_A	axial force coefficient, positive aft
C_N	normal force coefficient, positive upward
g	acceleration due to gravity. The value read in does not affect conversion from weight to mass
I_x, I_y, I_z, I_{xz}	moments and product of inertia about reference body axes
p,q,r	angular velocities, body axes unless used as a subscripts to form stability axis derivatives
S	reference wing area
T	thrust

[b] Irrelevant if MOVE(6) = 0

u nondimensional perturbation in total velocity, $\frac{V-V_o}{V_o}$

V total velocity

W weight

z_T intercept of thrust line with Z body axis, positive if below CG

α angle of attack of reference body axes

β angle of sideslip

δ generalized control deflection

θ pitch attitude of body axes

ρ mass density of air

ϕ Euler bank angle

ξ thrust inclination above X body axis

Subscripts

M partial derivative with respect to Mach number

o total value in reference flight condition

to initial value of perturbation variable

16.1.5 Input Cards

Card Group 1. Required for all cases
Card 1. Format 8A10. 80 arbitrary characters to identify case
Card 2. Format 7I1, 7I3. MOVE(1) through MOVE(7), N, K, NADD, M, L, NGADD, NVCOMB

Card Group 2A. Required if MOVE(1) = 1 and MOVE(3) $\neq$ 0. Four cards in format 6F12.0. Data must conform to MOVE number indicated by superscript.

X_u 2	Z_u 2	M_u	$X_{\dot{\alpha}}$ 2	$Z_{\dot{\alpha}}$ 2	$M_{\dot{\alpha}}$
X_α 2,4	Z_α 2,4	M_α 4	X_q 2	Z_q 2	M_q
X_δ 2,4	Z_δ 2,4	M_δ 4	V_o ft/sec	α_o deg	q_o rad/sec
θ_o deg	g ft/sec^2	u_{t_o}	α_{t_o} rad	q_{t_o} rad/sec	θ_{t_o} rad

Card Group 2B. Required if MOVE(1) = 1 and MOVE(3) = 1. Seven cards in format 6F12.0. Data must conform to MOVE number indicated by superscript.

C_{D_M} or C_{A_M} 2	C_{L_M} or C_{N_M} 2	C_{m_M}	$C_{D_{\dot{\alpha}}}$ or $C_{A_{\dot{\alpha}}}$ 2	$C_{L_{\dot{\alpha}}}$ or $C_{N_{\dot{\alpha}}}$ 2	$C_{m_{\dot{\alpha}}}$
C_{D_α} or C_{A_α} 2,4	C_{L_α} or C_{N_α} 2,4	C_{m_α} 4	C_{D_q} or C_{A_q} 2	C_{L_q} or C_{N_q} 2	C_{m_q}
C_{D_δ} or C_{A_δ} 2,4	C_{L_δ} or C_{N_δ} 2,4	C_{m_δ} 4	V_o ft/sec	α_o deg	q_o rad/sec
θ_o deg	g ft/sec^2	u_{t_o}	α_{t_o} rad	q_{t_o} rad/sec	θ_{t_o} rad
C_{D_o} or C_{A_o} 2	C_{L_o} or C_{N_o} 2	C_{m_o}	ρ slug/ft^3	S ft^2	$\bar{c}$ ft
W lb	I_y slug ft^2	Mach	z_T ft	ξ deg	T_δ lb/rad
T_M lb					

Card Group 2C. Required if MOVE(1) = 2 and MOVE(3) = 0 or 2. Five cards in format 6F12.0. Data must conform to MOVE number indicated by superscript.

Y_β 4	L_β 2,3,4	N_β 2,3,4	$Y_{\dot\beta}$	$L_{\dot\beta}$ 2,3	$N_{\dot\beta}$ 2,3
Y_p 2	L_p 2,3	N_p 2,3	Y_r 2	L_r 2,3	N_r 2,3
Y_δ 4	L_δ 2,3,4	N_δ 2,3,4	V_o ft/sec	α_o deg	q_o rad/sec
θ_o deg	g ft/sec^2	I_x slug ft^2	I_y slug ft^2	I_z slug ft^2	I_{xz} slug/ft^2
β_{t_o} rad	p_{t_o} rad/sec	V_{t_o} rad/sec	ϕ_{t_o} rad		

Card Group 2D. Required if MOVE(1) = 2 and MOVE(3) = 1. Six cards in format 6F12.0. Data must conform to MOVE number indicated by superscript.

C_{y_β} 4	C_{l_β} 2,4	C_{n_β} 2,4	$C_{y_{\dot\beta}}$	$C_{l_{\dot\beta}}$ 2	$C_{n_{\dot\beta}}$ 2
C_{y_p} 2	C_{l_p} 2	C_{n_p} 2	C_{y_r} 2	C_{l_r} 2	C_{n_r} 2
C_{y_δ} 4	C_{l_δ} 2,4	C_{n_δ} 2,4	V_o ft/sec	α_o deg	q_o rad/sec
θ_o deg	g ft/sec^2	I_x slug ft^2	I_y slug ft^2	I_z slug ft^2	I_{xz} slug ft^2
β_{t_o} rad	p_{t_o} rad/sec	V_{t_o} rad/sec	ϕ_{t_o} rad		
ρ slug/ft^3	S ft^2	b ft	W lb		

Card Group 3. Number of cards required equal to NADD. Format 2I2, 3F12.0.

Columns 1-2: a matrix row number, right adjusted
3-4: a matrix column number, right adjusted
5-16: constant — polynomial coefficients added in this A matrix element. (Only K+1 coefficients are read.)
17-28. first power
29-40: second power

Card Group 4. Not required if MOVE(6) = 0. N-mber of cards required equal to zero. Number of cards required equal to NGADD. Format 2I2, 3F12.0.

Columns 1-2: γ matrix row number, right adjusted
3-4: γ matrix column number, right adjusted
5-16: constant — polynomial coefficients added in this γ matrix element. (only L+1 coefficients are read.)
17-28: first power
29-40: second power

Card Group 5. Not required if MOVE(6) = 0. Number of basic cards equal to NVCOMB. Format I2, 9(I2, F6.0).

Columns 1-2: number of old variables combined in new variable
3-4, 11-12, etc.: index of old variables
5-10, 13-18, etc.: multiplying factor for old variable identified by index in previous field.

Any card containing an entry greater than nine in columns 1-2 must be followed by a continuation card giving remaining indices and factors in format 9 (I2, F6.0).

Card Group 6. Not required if MOVE(6) = 0. One card in format 36I2.

Columns 1-2: number of variables for which factored transfer functions or time histories are desired.
3-4, 5-6, etc.: list of indices identifying above variables listed in ascending order

See addendum for frequency response input

Card Group 7. Required if MOVE(6) = 2. One or more cards in format 6F12.0.

Columns 1-12: total duration of time history in seconds
13-24: time increment between time history points in seconds.

Card Group 7. (Continued)

Columns 25-36, 37-48, etc. } list of factors by which variables are multiplied before time history output. Number and order of factors must conform to variable index list in card group 6.

Card Group 8. Required if MOVE(6) = 2. This group is repeated for each segment of time history.

Card 1. Format 2I2, F12.0.

Columns 1-2: number of disturbances (Y matrix columns) to be combined for this segment. Does not include disturbances carried over from previous segments.

To specify that this segment will end when one of the computed variables reaches a specified value, enter the test variable index in columns 3-4 and the specified value of the test variable in columns 5-16.

To specify a segment of given duration, enter 0 in column 4 and the duration of the *segment* in columns 5-16.

To specify that this segment will continue for the remainder of the time history duration, enter 0 in column 4 and a number greater than the remaining duration in columns 5-16.

To specify that this segment will continue for the remainder of the time history duration and that a new time history starting at time zero will follow, enter -1 in columns 3-4; the entry in columns 5-16 is irrelevant. Then follow this card group 8 with one card giving the new duration and time increment as in columns 1-24 of card group 7 and follow with card groups 8 as needed for the new time history.

Card 2. Format 5(I2, F12.0)

Columns 1-2, 15-16, etc. } Disturbance index (matrix column number).

3-14, 17-28, etc. } Multiplying factor for disturbance identified by index in previous field. Disturbances in excess of 5 are given in succeeding cards having same format.

Card Group 9. Required for all cases. One card in format A10.

ANOTHER in columns 1-7 with columns 8-10 blank indicates another case to follow

THATS ALL in columns 1-9, with column 10 blank indicates the end of the job.

APPENDIX A TO SECTION 16.1

FREQUENCY RESPONSE AND TIME HISTORIES

16.1.A.1 Frequency Response

Frequency response can be calculated for up to ten transfer functions whose numerators are contained in the [G] matrix. This feature is called by entering the [G] matrix row and column indices (variable and disturbance indices) in format 10(2I2) starting at column 31 in the input card of group 6. For example, the following entries will produce frequency response output for G(2,1) and G(13,2):

Column 32	2
Column 34	1
Columns 35-36	13
Column 38	2

All variables called in the frequency response inputs must also be named in the variable list in columns 3-30 of card group 6.

The frequency response output for each transfer function will be listed immediately following the root and Bode listing for that transfer function. Frequency response amplitude (db) and phase (deg) are given at the intervals of $\log_{10}\omega$ of .1 over a range from at least one decade below the minimum pole or zero frequency to at least one decade above the maximum pole or zero frequency. This list is in ten columns headed by the mantissa of $\log_{10}\omega$ with the amplitude and phase in alternate rows identified by the characteristic of $\log_{10}\omega$.

Note that frequency responses are calculated even though poles and/or zeros may lie in the right half complex plane. The results represent only the steady state portion of the oscillatory response to a sinusoidal input; the transient response is ignored. To avoid computer problems in the vicinity of unstable resonances, amplitudes having magnitudes greater than 200 db are truncated to + or - 200 db. Phase angles in these regions, however, should be correct.

Note also that to obtain frequency response in the usual, or general form, the disturbance should be in the form of a unit impulsive input.

16.1.A.2 Time Functions

If time history outputs are called for (MOVE(6) = 2), the time history output listing will be preceded by a listing of time functions. These are algebraic expressions of the time variation of each variable named in the variable index list (columns 3-30 in card group 6) in response to each disturbance in the {d} vector. The combined disturbances and delayed disturbances used in the time history calculations do not apply to the time functions. The variables, are, however, multiplied by the factors given in card group 7.

For each combination of variable and disturbance, the time function contains one term of the form $Ke^{\lambda t}$ for each real characteristic root, one term of the form $Ke^{\sigma t}\cos(\omega t+\phi)$ for each conjugate pair of complex characteristic roots, and n power series terms from K to $Kt^{(n-1)}$ where n is the number of characteristic roots at the origin. The time function terms are listed in the same order as the characteristic roots.

$$
\begin{array}{c}
\\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ - \\ n
\end{array}
\begin{array}{c}
\begin{array}{ccccccc}
1 & 2 & 3 & 4 & 5 & - & n
\end{array}
\\
\left[
\begin{array}{ccccc|cc}
s\cos\alpha_o - X_u + q_o\sin\alpha_o & -s(\sin\alpha_o + X_{\dot{\alpha}}) - X_\alpha + q_o\cos\alpha_o & -X_q + \sin\alpha_o & (g/V_o)\cos\theta_o & -X_\delta & & \\
s\sin\alpha_o - Z_u - q_o\cos\alpha_o & s(\cos\alpha_o - Z_{\dot{\alpha}}) - Z_\alpha + q_o\sin\alpha_o & -Z_q - \cos\alpha_o & (g/V_o)\sin\theta_o & -Z_\delta & & \\
-M_u & -sM_{\dot{\alpha}} - M_\alpha & s - M_q & 0 & -M_\delta & & \\
0 & 0 & -1 & s & 0 & & \\
0 & 0 & 0 & 0 & 1 & & \\
\hline
 & & & & & & \\
 & & & & & &
\end{array}
\right]
\end{array}
\left\{
\begin{array}{c}
u \\ \alpha \\ q \\ \theta \\ \delta \\ \hline - \\ -
\end{array}
\right\}
$$

(a).- [A] and {X}

FIGURE 16.1-1. THE LONGITUDINAL STANDARD SUBMATRICES

$$
\begin{array}{c}
 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ - \\ n
\end{array}
\begin{array}{c}
\begin{array}{cccc} 1 & 2 & - & m \end{array} \\
\left[\begin{array}{c|c|c|c}
0 & u_{to}\cos\alpha_o - \alpha_{to}(\sin\alpha_o + X_{\dot{\alpha}}) & & \\
\hline
0 & u_{to}\sin\alpha_o + \alpha_{to}(\cos\alpha_o - Z_{\dot{\alpha}}) & & \\
\hline
0 & q_{to} - \alpha_{to}M_{\dot{\alpha}} & & \\
\hline
0 & \theta_{to} & & \\
\hline
1 & 0 & & \\
\hline
 & & & \\
\hline
 & & &
\end{array}\right]
\end{array}
\quad
\left\{\begin{array}{c}
\delta' \\ IC \\ \hline - \\ -
\end{array}\right\}
\begin{array}{c}
1 \\ 2 \\ - \\ m
\end{array}
$$

(b.) [γ] and {d}

FIGURE 16.1-1. CONCLUDED

	1	2	3	4	5	-	n	{X}
1	$s(1-Y_{\dot{\beta}})$ $-Y_\beta$	$-\sin\alpha_o - Y_p$	$\cos\alpha_o - Y_r$	$-(g/V_o)\cos\theta_o$	$-Y_\delta$			β
2	$-sL_{\dot{\beta}}$ $-L_\beta$	s $-L_p - q_o I_{xz}/I_x$	$-sI_{xz}/I_x$ $-L_r + q_o(I_z - I_y)/I_x$	0	$-L_\delta$			p
3	$-sN_{\dot{\beta}}$ $-N_\beta$	$-sI_{xz}/I_z$ $-N_p + q_o(I_y - I_x)/I_z$	s $-N_r + q_o I_{xz}/I_z$	0	$-N_\delta$			r
4	0	1	$\tan\theta_o$	$-s$ $q_o\tan\theta_o$	0			ϕ
5	0	0	0	0	1			δ
-								-
n								-

(a). [A] and {X}

FIGURE 16.1-2. THE LATERAL-DIRECTIONAL STANDARD SUBMATRICES

ORIGINAL PAGE IS OF POOR QUALITY

$$
\begin{array}{cc}
 & \begin{array}{cccc} 1 & \quad\quad 2 \quad\quad & - & m \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ - \\ n \end{array} &
\left[\begin{array}{c|c|c|c}
0 & \beta_{to}(1-Y_{\dot{\beta}}) & & \\ \hline
0 & -\beta_{to}L_{\dot{\beta}}+p_{to}-r_{to}I_{xz}/I_x & & \\ \hline
0 & -\beta_{to}N_{\dot{\beta}}-p_{to}I_{xz}/I_z+r_{to} & & \\ \hline
0 & -\phi_{to} & & \\ \hline
1 & 0 & & \\ \hline
 & & & \\ \hline
 & & &
\end{array}\right]
\end{array}
\quad
\begin{array}{cc}
\begin{Bmatrix} \delta \\ IC \\ - \\ - \end{Bmatrix} & \begin{array}{c} 1 \\ 2 \\ - \\ m \end{array}
\end{array}
$$

(b). [Y] and {d}

FIGURE 16.1-2. CONCLUDED

TABLE OF CONTENTS FOR SECTION 17.1,

THERMODYNAMIC ANALYSIS OPTIONS OF PROGRAM ATOP

TABLE OF CONTENTS FOR SECTION 17.2, PROGRAM ABLATOR

SECTION 17

THERMODYNAMICS

Two thermodynamic analysis technology modules are included in the ODIN technology module library:

1. The thermodynamic analysis options of program ATOP, Section 7.3. These options can only be employed in conjunction with an ATOP trajectory analysis.

2. A one-dimensional analysis of a charring ablator; this module was constructed by Swann, Pittman, and Smith of Langley Research Center.

17.1 THERMODYNAMIC ANALYSIS OPTIONS OF PROGRAM ATOP

The Six-Degree-of-Freedom Trajectory program and the earlier version of the trajectory optimization program (Reference 1) included a subprogram to calculate the structural temperature of a hemispherical stagnation point or an unswept wedge. The air properties used were those of calorically imperfect (vibration equilibrium) air. The structural temperature was determined by assuming a surface temperature, calculating the corrective and radiative heating rate, and iterating to find the equilibrium surface temperature at which the convective and radiative heating rates balanced. Experience with these programs has shown that the surface temperature iteration significantly increases the computing time and sometimes fails to converge properly. In addition, the calorically imperfect gas properties were good approximations to real air only at lower temperatures than those which occur at near-satellite speeds on hypersonic lifting vehicles which are currently under study.

Steve Rinn of the Air Force Flight Dynamics Laboratory has developed an improved aerodynamic heating subroutine which is included in the present trajectory analysis program. The formulation outlined in this section is made up of two parts; one of which computes the transient skin temperature of a flat swept wing at angle of attack assuming an attached shock wave, and the second which computes the transient surface temperature at the stagnation point of a hemispherical nose. The transient temperature is obtained by integration of temperature rate, considering convective and radiative heating rates as well as the heat absorbed by the skin. This differential equation is then added to the trajectory equations defining the skin temperature as a state variable. The gas properties are those of air in chemical equilibrium.

An option has been added by which ideal gas properties may be used instead of equilibrium air. A second option replaces transient temperature integration by calculation of the radiation equilibrium temperature, using an improved iteration technique. These two options permit a reduction in the amount of calculation at the cost of a loss of accuracy which may be acceptable for some applications.

The following discussion consists of the formulation provided by Steve Rinn, plus a description of the two options mentioned in the previous paragraph.

17.1.1 General Heating Analysis

The heat transfer at a surface element is a function of many energy sources. Many of these sources, however, are extremely small and are generally not even considered in more exact analyses. The predominant energy sources are aerodynamic heat transfer, surface radiation, surface heat absorption and conduction, shock layer radiation, and internal radiation. Conduction and internal radiation require a detailed knowledge of both the internal structure and composition of the structural materials and as such are beyond the scope of this program. In addition, these heating terms are small, generally resulting in a heat loss at the two surfaces under consideration. Shock layer radiation represents the electromagnetic radiation from the high temperature gases in the shock layer and is of little significance in the flight regime of the presently envisioned reentry vehicles. Since lifting vehicles will largely be confined to the flight regime bounded by the equilibrium glide paths corresponding to W/C_LA's of 10 and 1000, only vehicles of extremely large nose radii will be adversely affected by shock layer radiation.

Ignoring the effects of conduction, internal and shock layer radiation, the general energy balance equation for a radiatively cooled surface element can be written as

$$q_c - q_r = q_s \tag{17.1.1}$$

which states that the energy stored in the surface material is the difference between the convective aerodynamic heat input and the heat radiated to space. The basic definition of these quantities may be expressed as follows:

$$q_s = G dT_w/dt \tag{17.1.2}$$

$$q_r = 4.758 \times 10^{-13} \, \varepsilon (T_w^{\,4} - T_r^{\,4}) \tag{17.1.3}$$

$$q_c = h(H_{aw} - H_w) \tag{17.1.4}$$

q_s represents the net rate that heat is transferred into or out of the surface element. The heat absorption capacity of the surface material is defined as

$$G = \rho_w \, Cp_w \, \delta_w \tag{17.1.5}$$

where ρ_w and Cp_w are properties of the material and δ_w is the skin thickness. The properties of some of the representative materials which are presently in use or have been proposed for reentry vehicles are presented in Table I and were obtained from Reference (2). These properties, although a function of the skin temperature, are input to the program as constants, in contrast to the tables which were required by the previous heating subprogram, for several reasons. First of all, over much of the reentry trajectory the skin temperatures are relatively constant in which case there is relatively little change in the material properties. Secondly, over much of the trajectory the temperatures are approaching equilibrium temperature values in which case the convective heat transfer is balanced by the radiative heat transfer and hence any drastic changes in the material properties, if they were to occur, would have only a very minor effect on the surface temperature. Finally most of the common and refractory materials suffer drastically from unsatisfactory oxidation resistance at much lower temperatures than those noted in Table I and hence are confined to temperatures at which these large property changes do not occur.

q_r represents the heat radiated from the surface element to space, or in the case of atmospheric flight, to the freestream. The surface

TABLE I

SKIN MATERIAL PROPERTIES

	Aluminum	Stainless Steel	Titanium	Molybdenum	Silicon Carbide	Graphite
TYPE		AISI 301				3474D
T_{max}	1680	3100	3510	5210	5350	9000+
ρ	169	494	287	639	185	105

	Aluminum			Stainless Steel			Titanium			Molybdenum			Silicon Carbide			Graphite		
T	Cp	ε	ρCp	Cp	ε	ρCp	Cp	ε	ρCp	Cp	ε	ρCp	Cp	ε	ρCp	Cp	ε	ρCp
250	.156	.05	26.36	.085	.21	42.16	.092	.31	26.40	.057	.04	36.42	.058	.83	10.73			
500	.213	.05	36.00	.108	.21	53.11	.123	.31	35.30	.062	.05	39.42	.150	.84	27.75	.160	.80	16.67
750	.235	.05	39.72	.123	.22	60.39	.135	.31	38.75				.220	.85	40.70			
1000	.251	.05	42.42	.133	.23	64.53	.140	.31	40.18	.065		41.54	.245	.85	45.33	.300	.82	31.26
1250	.267	.05	45.12	.138	.25	66.65	.142	.31	40.75									
1500	.284	.05	48.00	.142	.29	68.02	.145	.31	41.62	.067	.08	42.81	.270	.86	49.95	.390	.85	40.64
1750				.145	.34	68.88	.147	.32	42.19									
2000				.152	.39	71.59	.149	.34	42.76	.070	.13	44.73	.290	.87	53.65	.440	.87	45.85
2250				.165	.54	77.06	.150	.38	43.05									
2500				.175	.60	80.85	.151	.39	43.34	.075	.20	47.93	.310	.86	56.98	.480	.89	50.02
3000							.176	.40	50.51	.083		53.04	.330	.84	60.50	.500	.90	52.10
3500							.222	.41	63.71	.091	.25	58.15	.347	.82	64.20			
4000										.100		63.90				.520	.90	54.18
4500										.109	.30	69.75	.373	.79	68.13			
5000										.121		77.32				.530	.88	55.23
5500													.400	.75	74.00			
6000																.550	.87	57.31
7000																.760	.86	79.19

emissivity is also input to the program as a constant. As noted in Table I the emissivities for the common and refractory metals are quite low and thus in order to obtain high radiation rates special coatings are required. Intermetallic silicon and camouflage paint coatings have been developed which possess emissivities between 0.6 and 0.75. These coatings also serve as protection against severe oxidation damage possessing capabilities of 3000°R for long time durations and 3500°R for short periods.

q_c, the aerodynamic heat transfer, represents the heat energy transferred to the surface element through the boundary layer. The heat transfer coefficient, h, is a function of both the vehicle geometry and the local air properties and is thus dependent upon the location of the surface element on the vehicle.

Solving the general heating equation for the temperature derivative yields

$$\dot{T}_w = \frac{h}{G}(H_{aw} - H_w) - \frac{4.758 \times 10^{-13}\varepsilon}{G}(T_w{}^4 - T_r{}^4) \tag{17.1.6}$$

Wall temperatures are obtained from this equation by means of the numerical integration subroutine within the SDF and TOP programs. Let the subscript e refer to a hemispherical nose stagnation point, and subscript s refer to a point on the centerline of a swept wing with a hemispherical tip. The following differential equations are then obtained for these special cases.

$$\dot{T}_s = \frac{h}{G_s}(H_{aw} - H_s) - \frac{4.758 \times 10^{-13}\varepsilon_s}{G_s}(T_s{}^4 - T_r{}^4) \tag{17.1.7}$$

$$G_s = \rho_s \, C_{p_s} \, \delta_s \tag{17.1.8}$$

$$\dot{T}_e = \frac{h}{G_e}(H_{aw} - H_e) - \frac{4.758 \times 10^{-13}\varepsilon_e}{G_e}(T_e{}^4 - T_r{}^4) \tag{17.1.9}$$

$$G_e = \rho_e \, C_{p_e} \, \delta_e \tag{17.1.10}$$

The use of this heating subprogram in these computer programs increases the computation or run time by a factor of from 1 to 2 depending on the sensitivity of this temperature derivative. This sensitivity is largely controlled by the magnitude of G or more aptly the skin thickness since the variation of the ρC_p product is relatively insensitive to both temperature and material composition as indicated in Table I. In an approximate program of this type it is not overly important that the wall thickness be realistic as long as it is neither excessively large nor excessively small. Experience with this program has indicated that the wall temperatures obtained will consistently approximate equilibrium temperatures if the nose thickness is between .01 and .1 feet and the swept wing thickness is between .001 and .01 feet.

17.1.2 Swept Wing Stagnation Line Formulation

(a) Heat Transfer - The heat transfer coefficient presented in the previous heating subprogram is only applicable to an unswept flat plate. Consequently various modifications are necessary in order to include high sweep effects.

At present there is no one method available which adequately describes the heat transfer to the stagnation line of a highly swept delta wing. As a consequence three flow regimes are frequently distinguished in order to provide adequate correlation throughout the angle of attack range of interest.

The first of these regimes occurs at low angles of attack and corresponds to the planar flow of an unswept flat plate in which the flow streamlines are essentially uniform and parallel to the wing centerline. The second regime is characterized by the divergence of the flow streamlines from the centerline towards the wing leading edges and, as the flow approximately parallels the ray lines emanating from the wing virtual apex, the streamlines are considered conical in nature. This regime is applicable until the flow stagnates. The third regime is characterized by subsonic, stagnation flow which occurs after shock detachment. This regime is confined to angles of attack greater than the theoretical cone shock detachment angle and, since these angles do not normally occur in a lifting reentry, the heating formulation for this regime has been excluded.

In the first flow regime, the heat transfer coefficient is determined by the Reference Enthalpy Strip Theory for an unswept flat plate (Reference 3) as was used in the previous heating subprogram. For laminar flow this coefficient can be written as

$$h_{FP} = \left(\frac{0.332}{778.26}\right)(P_r{}^*)^{-2/3}\left(\frac{\rho^*\mu^*V_2}{l_H}\right)^{0.5} \qquad (17.1.11)$$

In the second flow regime the heat transfer coefficient is determined by applying a correction factor to nondivergent Strip Theory, a procedure frequently referred to as Outflow or Streamline Divergence Theory. This correction factor, for laminar flow, is given in Reference (4) as

$$\frac{h}{h_{FP}} = (2n + 1)^{0.5} \qquad (17.1.12)$$

where

$$n = .17 \tan \alpha \tan \Lambda \qquad (17.1.13)$$

If it is desirable to include the third flow regime then reference is made to References (5), (6), and (7).

Since it has long been noted that there is a marked increase in the heat transfer rate in turbulent flow as contrasted to laminar flow, information on boundary layer transition is of particular importance. Unfortunately the state-of-the-art of hypersonic transition theory is relatively primitive and at present there are no reasonably accurate methods available which predict transition while taking into account all of the pertinent parameters. However, Reference (8) has presented an empirical equation which considers all of these parameters with the exception of angle of attack. In this procedure the transition Reynolds number at zero angle of attack was approximated by

$$R_{N_T} = \left(\frac{R_{N_{1H}}}{l_H}\right)^{0.4} \left[\frac{1 \times 10^6 + 0.36 \times 10^6 \sqrt{|M_1 - 3|}}{1.552 \times 10^2}\right] (\cos \Lambda)^{0.5} \tag{17.1.14}$$

which is applicable for sweep angles greater than 25 degrees. In order to include the effects of angle of attack it is assumed that the transition Reynolds number is based on the local rather than the freestream properties noted previously, a fact which has some experimental justification. The form of the transition criterion used in the present program is then

$$R_{N_T} = \left(\frac{R_{N_{2H}}}{l_H}\right)^{0.4} \left[\frac{1 \times 10^6 + 0.36 \times 10^6 \sqrt{|M_2 - 3|}}{1.552 \times 10^2}\right] (\cos \Lambda)^{0.5} \tag{17.1.15}$$

Because of the uncertainties involved in the transition state it is often assumed that transition between laminar and turbulent flow is instantaneous at the point where the local Reynolds number exceeds this transition or critical Reynolds number. However, the step discontinuity is not compatible with certain optimization procedures, since the partials give no indication of the jump in heating and wall temperature that will result from crossing a transition boundary. An exponential function is therefore used to give a continuous fairing from the laminar heat transfer coefficient, h_l at the transition point to the turbulent value, h_t at a slightly higher Reynolds number (or boundary layer length).

$$h = h_l + (h_t - h_l)\left(1 - e^{\left[-\frac{(R_{N_2} - R_{N_T})}{R_{N_T}} \tau_{tr}\right]}\right) \qquad (R_{N_2} > R_{N_T}) \tag{17.1.16}$$

$$h = h_l \qquad (R_{N_2} \leq R_{N_T}) \tag{17.1.17}$$

The nominal value of 100. for τ_{tr} gives effectively a step change, a value of about 3. gives a gradual transition which may help the optimization process, and a value of 0 gives completely laminar heating.

In the first flow regime turbulent Reference Enthalpy Strip Theory is also applicable. However, rather than using the more familiar Colburn relation applied in the previous heating subprogram, this program makes use of the heat transfer coefficient given in Reference (9) because of its increased accuracy over the entire flight regime. This coefficient is

$$h_{FP} = \frac{0.181}{778.26} (P_r^*)^{-2/3} \frac{\rho^* V_2}{(\text{Log}_{10} R_{N_{1_H}})^{2.58}} \qquad (17.1.18)$$

Whenever a flow discontinuity, such as a geometry change or transition from laminar to turbulent flow, occurs this heat transfer coefficient is no longer applicable. In order to use this equation in a region downstream of the discontinuity it is first necessary to relate the characteristics of the actual boundary layer to the characteristics of an effective boundary layer which has no discontinuity. This is accomplished through the use of an effective boundary layer length which is given in Reference (4) as

$$l_{He} = l_2 + l_{X_2} \qquad (17.1.19)$$

where l_{X_2} is the geometric distance from the discontinuity to the point of interest and l_2 is the effective starting length. For transition from laminar to turbulent flow the effective starting length is given by

$$l_2 = 65.3 \left(\frac{\mu^*}{\rho^* V_2} \right)^{\frac{3}{8}} l_t^{\frac{5}{8}} \qquad (17.1.20)$$

where l_t is the distance from the stagnation point of the nose to the point at which transition occurs and, by definition,

$$l_{X_2} = l_H - l_t \qquad (17.1.21)$$

Thus the effective boundary layer length is

$$l_{H_e} = \left[1 + 65.3 \left(\frac{\mu^*}{\rho^* V_2 l_H} \right)^{\frac{3}{8}} \left(\frac{l_t}{l_H} \right)^{\frac{5}{8}} - \frac{l_t}{l_H} \right] l_H \qquad (17.1.22)$$

in which case the effective Reynolds number becomes

$$R^*_{N_{1_{He}}} = R^*_{N_{1_H}} + 65.3 \left(\frac{R^*_{N_{1_H}} \; R_{N_T}}{R_{N_2}} \right)^{\frac{5}{8}} - \left(\frac{R_{N_{1_H}}^* \; R_{N_T}}{R_{N_2}} \right) \qquad (17.1.23)$$

It is this term which should be used in the turbulent heat transfer coefficient.

In the second flow regime the correction factor for including turbulent outflow effects is given in Reference (4) as

$$\frac{h}{h_{FP}} = (1 + 1.25n)^{0.2} \qquad (17.1.24)$$

where n is as was given previously for laminar Outflow Theory.

The preceding equations are only applicable for a continuum, equilibrium flow and, thus, at high altitudes and Mach numbers various "low Reynolds number" phenomena, such as viscous interaction and slip flow, are not accounted for. From Reference (10), the combined effects of these nonclassical phenomena are approximated by

$$\frac{h}{h_c} = \frac{h_v/h_c}{\left[1 + \frac{1}{M_1}\sqrt{\frac{H_w}{H_2}\frac{M_2}{\sqrt{R_{N2}}} + \frac{H_w}{H_2}\left(\frac{M_2}{\sqrt{R_{N2}}}\right)^2}\right]} \tag{17.1.25}$$

where h_c is the continuum heat transfer coefficient given previously. In addition

$$\frac{h_v}{h_c} = 1 + \frac{a'\,\bar{x}}{} \tag{17.1.26}$$

when $a'\;\bar{x}\;\leq 4$ and

$$\frac{h_v}{h_c} = \sqrt{a'\;\bar{x}} \tag{17.1.27}$$

when $a'\;\bar{x}\; > 4$. The term, a', as approximated by a least squares curve fit, is

$$a' = 0.040714 + 0.20829(H_w/H_T) + 0.86713(H_w/H_T)^2 - 0.79738(H_w/H_T)^3$$
$$+ 0.442979(H_w/H_T)^4 \tag{17.1.28}$$

and the term, $\bar{x}$, is

$$\bar{x} = \frac{M_2^3}{\sqrt{R_{N2}}}\sqrt{\frac{\rho_w\,\mu_w}{\rho_2\,\mu_2}} \tag{17.1.29}$$

This equation approaches free molecular flow values at extremely high altitudes and as such can probably be applied throughout the entire flight regime.

The equations which define the chemical properties of air are common to all of the flow fields around a vehicle and as such the auxiliary functions defining the properties in the heat transfer equations have been subdivided into two parts; the formulation of the thermodynamic and transport property equations which are contained in a separate subroutine CHEMP and presented in Subsection (4), and the formulation of the auxiliary functions which are peculiar to either the swept wing or stagnation point regions and are contained in the heating subprogram proper.

(a) Swept Wing Auxiliary Functions.- The chemical property equations in Section (4) indicate that all of the thermodynamic and transport properties required are determined when the pressure and either the enthalpy or temperature of the particular flow field are known. Accordingly, since the remaining auxiliary functions are also dependent upon these terms, these dynamic properties will be considered first.

At present there are no simple, theoretical techniques available which adequately predict the local pressure on a swept delta wing throughout the entire angle of attack regime. Oblique Shock and the Tsien Similarity Theory used in the previous heating subprogram generally overpredict the local pressure while Newtonian Theory, also frequently applied to a swept wing, generally underpredicts the pressure. Wedge-cone Theory is the most applicable of the various techniques but the complexity of the conical equations makes their use extremely prohibitive in this program. A semi-empirical equation based on the Newtonian concept which is applicable in the angle of attack range of interest, is

$$C_p = 1.95 \sin^2\alpha + \frac{0.3925 \sin\alpha \cos\alpha}{M_1^{0.3}} \tag{17.1.30}$$

where

$$\frac{P_2}{P_1} = 1 + 0.7 M_1^2 C_p \tag{17.1.31}$$

The unswept flat plate heat transfer coefficients were derived by solving the incompressible boundary layer equations and hence in order to include compressibility effects these coefficients must be computed using reference rather than local properties. Reference (3) has empirically derived an equation for the reference enthalpy, which is defined as follows:

$$H^* = 0.22 H_{aw} + 0.28 H_2 + 0.5 H_w \tag{17.1.32}$$

The adiabatic wall or recovery enthalpy, H_{aw}, is the value that the enthalpy at the wall would attain if the heat transfer was zero and if defined as

$$H_{aw} = r_H H_T + (1 - r_H) H_2 \tag{17.1.33}$$

The recovery factor, r_H, is approximated by

$$r_H = \sqrt{P_r^*} \tag{17.1.34}$$

for laminar flow and

$$r_H = \sqrt[3]{P_r^*} \tag{17.1.35}$$

for turbulent flow in the suborbital flight regime (Reference (11)) where $P_r{}^*$ is the Prandtl number based on the reference enthalpy. Since the reference enthalpy is a function of the reference Prandtl number which in turn is a function of the reference enthalpy, an iterative procedure is required in order to determine the reference enthalpy. However, the variation of the Prandtl number is small and hence can be assumed constant. Over the flight regime of greatest interest the average value of the Prandtl number is about 0.75 and hence this value was used whenever the Prandtl number was required.

The local enthalpy, H_2, is defined by means of the conservation of energy across an oblique shock wave as

$$H_2 = H_T + 0.5V_2{}^2 \tag{17.1.36}$$

The local velocity, V_2, is determined from the conservation of mass and momentum across an oblique shock wave and in terms of the pressure coefficient is given by

$$\frac{V_2}{V_1} = (1 - 0.5C_p)/\cos\alpha \tag{17.1.37}$$

The stagnation or total enthalpy, H_T, is constant across the shock wave and can be expressed in terms of the freestream properties as

$$H_T = \frac{V_1{}^2}{2}\left(\frac{M_1{}^2 + 5}{M_1{}^2}\right) \tag{17.1.38}$$

The wall enthalpy, H_w, is obtained directly from subroutine CHEMP.

With the dynamic properties so defined all of the other chemical properties are determined through subroutine CHEMP.

The other required auxiliary functions are the local Reynolds number and the local Mach number which are defined as

$$R_{N_2} = \frac{\rho_2 V_2 l_H}{\mu_2} \tag{17.1.39}$$

where

$$l_H = l_{H_1} + (1.5708 - \alpha_s)r_0 \tag{17.1.40}$$

$$\alpha_s = \alpha + D_7 \tag{17.1.41}$$

α_s is the surface slope relative to the free stream (x wind axis) at the point of interest. α is the vehicle angle of attack, and D_7 is the wedge angle relative to the x body axis at the point of interest.

l_{H1} is the geometric distance along the wing centerline measured from the shoulder of the nose to the point of interest and r_o is the nose radius. The Mach number is defined as

$$M_2 = \frac{V_2}{a_2} \tag{17.1.42}$$

where a_2 is the local speed of sound and is obtained from subroutine CHEMP.

17.1.3 Hemispherical Nose Stagnation Point Formulation

(a) Heat Transfer - Of the many methods presently available for computing stagnation point heat transfer the technique presented by Fay and Riddell in Reference (12) is probably the most highly regarded. In terms of the heat transfer coefficient the Fay and Riddell equation is (17.1.43)

$$h_{FR} = \frac{0.763}{778.26} (Pr_w)^{-0.6} \left(\frac{\rho_w \mu_w}{\rho_T \mu_T}\right)^{0.1} \left(\rho_T \mu_T \frac{dV}{dS}\right)^{0.5} \left[1 + (Le_w^{0.52} - 1)\frac{H_D}{H_T}\right]$$

The definition and formulation of each of these terms is contained in either Subsection (4) or in (6).

The previous heating subprogram employed the method of Detra, Kemp, and Riddell (Reference (13))to obtain the stagnation point heat transfer, which is an empirical equation based on the Fay and Riddell coefficient and experimental data. A comparison was made between these two methods by computing equilibrium temperature heat transfer rates which in the case of the Fay and Riddell coefficient were based on the formulation presented herein while for the Detra, Kemp, and Riddell equation the previous formulation was utilized. Based on this comparison the Fay and Riddell coefficient was employed because of the increased accuracy afforded by it.

The Fay and Riddell heat transfer coefficient is only applicable in a continuum fluid flow in chemical equilibrium and since deviations from this classical flow do occur they should be noted.

Nonequilibrium phenomena result from the incomplete development of the chemical reactions in the flow and, like noncontinuum effects, are a low density phenomena. These effects are, at present, not clearly defined but they appear to be rather insignificant from a standpoint of heat transfer and as such will be given no further consideration.

The deviations from the classical continuum stagnation point equations, termed "low Reynolds number" effects in the flight regime of interest in this program, are categorized as vorticity interaction, viscous layer, slip flow, and merged layer. A detailed explanation of these phenomena can be obtained from References (4) through (8) . Although the first two flow regimes have been fairly well documented there is very little literature available on the combined effects of all of these phenomena and, as such, there are presently no closed form solutions for the "low Reynolds number" regime. In this subprogram the combined effects of these deviations were obtained by curve fitting the numerical solutions of Reference which, in terms of the heat transfer ratio, are approximated by

$$\frac{h}{h_{FR}} = \left(\frac{0.04}{e}\right)^n \bar{R}^m \tag{17.1.44}$$

where

$$e = \rho_1/\rho_T \tag{17.1.45}$$

$$R_{e_s} = \rho_1 V_1\ r_o/\mu_T \tag{17.1.46}$$

$$\bar{R} = 50\ e^2\ R_{e_s} + A_R \tag{17.1.47}$$

$$A_R = .285, \quad (x \leq -1) \tag{17.1.48}$$

$$A_R = 0, \quad (x \geq 4) \tag{17.1.49}$$

$$A_R = .493 + .272667\ x + 0.07\ x^2 + 0.0063\ x^3, \quad (-1 < x < 4) \tag{17.1.50}$$

$$m = 0.6(\bar{R})^{-0.51428} \tag{17.1.51}$$

$$x = 2 + \text{Log}_{10}\ (e^2\ R_{e_s}) \tag{17.1.52}$$

$$n = 0.51973 - 8.0762 \times 10^{-3}\ x - 0.21707\ x^2 - 2.4891 \times 10^{-2} x^3 + 6.2601 \times 10^{-2} x^4 - 1.2118 \times 10^{-2} x^5 \quad (0 < x \leq 2.95) \tag{17.1.53}$$

$$= 0 \quad (x > 2.95) \tag{17.1.54}$$

$$= 0.52\ (x < 0) \tag{17.1.55}$$

The term, h_{FR}, represents the Fay and Riddell heat transfer. These equations are restricted to values of e $\geq .04$ and $e^2 Re_s \geq .01$ which in terms of altitude is between 300,000 and 350,000 feet depending on the nose radius.

(a) Auxiliary Functions - As was the case with the swept wing auxiliary functions, all of the terms in the stagnation heat transfer coefficient are related to the chemical properties. Accordingly the formulation of the dynamic properties required to obtain these chemical properties will be considered first.

The local stagnation pressure behind a normal shock wave for an incompressible boundary layer is

$$\frac{P_T}{P_1} = 1 + \frac{\rho_1 V_1^2}{P_1} \left(1 - 0.5 \frac{\rho_1}{\rho_2}\right) \tag{17.1.56}$$

An exact real gas solution of this equation requires a double iterative procedure because of the dependency on the density ratio. However, the real gas solution can be closely approximated by applying the normal shock density ratio for a perfect gas using a fictitious specific heat ratio of 1.2. Thus the real gas stagnation pressure is approximated by

$$\frac{P_T}{P_1} = 1 + 1.4\, M_1^2 \left(1 - 0.5 \frac{\rho_1}{\rho_2}\right) \tag{17.1.57}$$

where

$$\frac{\rho_1}{\rho_2} = \frac{M_1^2 + 10}{11 M_1^2} \tag{17.1.58}$$

The second state variable required in computing the stagnation properties is the stagnation enthalpy which was given previously as

$$H_T = 0.5 V_1^2 \,(M_1^2 + 5)/M_1^2 \tag{17.1.59}$$

It should be noted that the atmosphere subroutines in the previous SDF and TOP programs cease to compute the free stream speed of sound for altitudes in excess of 300,000 feet in which case the Mach number becomes undefined and all of the equations given previously in terms of this parameter are no longer applicable. The 1959 ARDC atmosphere subroutine has been modified to calculate approximate values of speed of sound above 300,000 feet but the 1962 atmosphere option is limited to about 300,000 feet. This option could be used with HETS by adding an equation of the following form to the program.

$$M_1^2 = 0.71428 \;\rho_1 V_1^2 / P_1 \tag{17.1.60}$$

With the stagnation pressure and enthalpy and an initial value of the wall temperature, the remaining chemical properties required by the heat transfer coefficient can be computed.

The velocity gradient at the stagnation point of a hemispherical nose can be determined through the use of a Modified Newtonian pressure distribution (References (76) and (35)) which yields

$$\frac{dV}{dS} = \frac{1}{r_0} \sqrt{\frac{2(P_T - P_1)}{\rho_T}} \tag{17.1.61}$$

This equation is only applicable for Mach numbers in excess of 5 because of a like restriction on Modified Newtonian Theory. For Mach numbers less than this value the velocity gradient is approximated by an empirical equation in Reference (4) as

$$\frac{dV}{dS} = 1.5 \frac{V_2}{r_0} (1 - 0.252M_2^2 - 0.0175M_2^4) \qquad (17.1.62)$$

where

$$\frac{V_2}{V_1} = \frac{M_1^2 + 5}{6M_1^2} \qquad (17.1.63)$$

$$M_2^2 = \frac{M_1^2 + 5}{7M_1^2 - 1} \qquad (17.1.64)$$

for $M_1 > 1$ and

$$V_2 = V_1 \qquad (17.1.65)$$

$$M_2 = M_1 \qquad (17.1.66)$$

for $M_1 \leq 1$.

The value of the Lewis number used in this program is

$$Le_w = 1.4 \qquad (17.1.67)$$

which is commonly used in the Fay and Riddell equation because it is somewhat representative of its maximum value and additionally correlates well with experimental data. Although the Lewis number presented in Reference (14) varies significantly the effect on the heat transfer is small. Since the additional formulation required to incorporate the variable Lewis number is considerable, this effect will be neglected and the Lewis number parameter,

$$1 + (Le_w^{0.52} - 1)\, H_D/H_T$$

can be rewritten as

$$1 + 0.191\, H_D/H_T$$

The dissociation enthalpy, H_D, was obtained through an empirical equation in Reference (4) as

$$H_D = 1.8219 \times 10^8 (Z - 1) \qquad (17.1.68)$$

for $Z < 1.2$ and

$$H_D = 3.6438 \times 10^7 + 3.4906 \times 10^8 (Z - 1.2) \qquad (17.1.69)$$

for $Z \geq 1.2$.

The Fay and Riddell heat transfer coefficient is a function of the wall Prandtl number. Since in a typical hypervelocity reentry the wall temperature will range between approximately 3500°R and 6000°R the average value of the Prandtl number will be approximately 0.75 as was the case for the reference Prandtl number in the swept wing formulation and thus this value was used in this program.

17.1.4 Chemical Property Subroutine, CHEMP

The chemical properties associated with a gas describe its macroscopic and microscopic behavior or, in other words, the chemical state of a gas is described by its thermodynamic and transport properties. The transport properties are themselves defined in terms of the thermodynamic properties and hence the thermodynamic properties will be considered first.

The thermodynamic properties of a gas are categorized as either thermal or caloric state variables.

The thermal properties are those properties which are not explicitly involved with the energy of the system and, in this program, the significant thermal properties are pressure, temperature, and density. The relationship between these terms is expressed by the thermal equation of state,

$$P = \rho ZRT \tag{17.1.70}$$

The compressibility factor, Z, is a measure of the number of moles of dissociated, ionized gas to the number of moles of undissociated, unionized gas. Under atmospheric conditions the compressibility factor for air is one, the perfect gas assumption. However, for real air, Z can deviate from unity for two reasons: at low temperatures and high pressures the intermolecular forces between the air molecules, which account for the possibility of liquefying the gas, become important while at high temperatures and low pressures dissociation and ionization phenomena occur. Intermolecular phenomena, although important in high speed test facilities, are of little consequence under free flight conditions and hence only dissociation and ionization need be considered.

Dissociation is a two-body chemical process in which a molecule breaks up into atoms when the internal vibrational energy is sufficiently increased, through collision with the other particle, to sever its intramolecular bond. In turn recombination is a three-body process in which two atoms and a third particle collide, releasing energy to the third particle, and forming a molecule. In a gas in equilibrium a continuing process of molecular dissociation and atomic recombination occurs in such a manner that a statistical net degree of dissociation results. In a like manner ionization is much the same process with the exception that a particle colliding with a free atom releases enough energy to the atom to enable an electron to overcome the electrostatic force field of the atomic nucleus and escape from its shell.

The computational procedures required in solving for the compressibility factor are relatively complex, i.e. References (14) and (15) Consequently machine storage and computational time limitations involved in this program require that these procedures be left to more sophisticated programs. Fortunately, Reference (16) has empirically curve fitted the compressibility factor of air and the resulting equation is

$$Z = 2.5 + 0.1\,\mathrm{Tanh}(A_Z/900-7) + 0.4\,\mathrm{Tanh}(A_Z/1800-7) + \mathrm{Tanh}(A_Z/4500-5.8) \tag{17.1.71}$$

where

$$A_Z = T(1-.125\mathrm{Log}_{10}(P/P_0)) \tag{17.1.72}$$

The caloric state variables are those properties which describe the energy or energy related state of the system and, as such, are functions of the thermal properties. The important caloric properties in this program are the enthalpy and the speed of sound. The relationship between the thermal and caloric variables is given through the definition of the enthalpy,

$$H = E + P/\rho \tag{17.1.73}$$

or

$$H = E + ZRT \tag{17.1.74}$$

The energy of the system is the sum of the translational, rotational, vibrational, and electronic energies of the molecular and atomic species within the gas. When a mixture of gases is considered the equations associated with the various mol fractions and component energies are quite complex and thus machine storage and computational requirements are again prohibitive. However Reference (2) has also empirically curve fitted the statistical net energy of the system for air. When combined with the equation above, the enthalpy of air can be given as

$\underline{1 < Z < 1.2}$ (17.1.75)

$$H/RT = Z + (2-Z)(2.5 + (5400/T)/(\exp(5400/T)-1)) + (Z-1)(3 + 106200/T)$$

$\underline{1.2 < Z < 2}$

$$H/RT = Z + (2-Z)(2.5 + (5400/T)/(\exp(5400/T)-1) + 0.2(3 + 106200/T) + (Z-1.2)(3 + 203400/T) \tag{17.1.76}$$

$\underline{2 < Z < 2.2}$

$$H/RT = Z + (4-Z)(1.5 + 91800/T) + (Z-2)(3 + 396000/T) \tag{17.1.77}$$

The speed of sound is defined as

$$a^2 = \gamma\,(\partial P / \partial\rho)_T \tag{17.1.78}$$

which in terms of previously defined variables can be expressed as

$$a^2 = \frac{\gamma ZRT}{1-\frac{P}{Z}\left(\frac{\partial Z}{\partial P}\right)_T} \tag{17.1.79}$$

The specific heat ratio, γ, is defined as

$$\gamma = C_P/C_V \tag{17.1.80}$$

where

$$C_P = (\partial H/\partial T)_P \tag{17.1.81}$$

and

$$C_V = (\partial E/\partial T)_V \tag{17.1.82}$$

and thus can be obtained through differentiation of the enthalpy equations. Since this requires double differentiation for both a constant pressure and a constant volume process, the specific heat ratio can be rewritten in terms of previously defined parameters and just one of the specific heats, in this case Cp which will be required by another section of subroutine CHEMP, as

$$1/\gamma = 1 - \frac{(Z + T(\partial Z/\partial T)_P)^2}{Z - P(\partial Z/\partial P)_T}\left(\frac{R}{C_P}\right) \tag{17.1.83}$$

The specific heat at constant pressure, from the above enthalpy definition, can be expressed as

$$C_P = H/T + R(T\partial\,(E/RT)/\,\partial T + T\partial\,Z/\partial T)_P \tag{17.1.84}$$

where, from the enthalpy equations,

$\underline{1 < Z < 1.2}$

$$\begin{aligned} T\partial(E/RT)/\partial T = {} & (2-Z)\left[\frac{5400/T}{(\exp(5400/T)-1)}\right]\left[\frac{5400/T}{(\exp(5400/T)-1)} \quad \exp(5400/T)-1\right] \\ & - (Z-1)(106200/T) + \left[(3+106200/T)-(2.5 + \frac{5400/T}{\exp(5400/T)-1})\right] \\ & T(\partial\,Z/\partial T)_P \end{aligned} \tag{17.1.85}$$

1.2 < Z < 2

$$T\partial(E/RT)/\partial T = (2-Z)\left[\frac{5400/T}{\exp(5400/T)-1}\right]\left[\frac{5400/T}{\exp(5400/T)-1}\exp(5400/T)-1\right]$$

$$- 0.2(106200/T) - (Z-1.2)(203400/T) + \left[(3+203400/T)\right.$$

$$\left.-(2.5 + \frac{5400/T}{\exp(5400/T)-1})\right] T(\partial Z/\partial T)_P \qquad (17.1.86)$$

2 < Z < 2.2

$$T\partial(E/RT)/\partial T = -(4-Z)(91800/T) - (Z-2)(396000/T) + \left[(3+396000/T)-\right.$$

$$\left.(1.5+91800/T)\right] T(\partial Z/\partial T)_P \qquad (17.1.87)$$

Finally the compressibility derivatives are obtained through differentiation of the compressibility equation.

$$T(\partial Z/\partial T)_P = \frac{A_Z}{9000}\left[5-\mathrm{Tanh}^2\left(\frac{A_Z}{900} - 7\right)-2\left(\mathrm{Tanh}^2\left(\frac{A_Z}{1800} - 7\right) + \mathrm{Tanh}^2\right.\right.$$

$$\left.\left.\left(\frac{A_Z}{4500} - 5.8\right)\right)\right] \qquad (17.1.88)$$

$$P(\partial Z/\partial P)_T = -.0542868(T/A_Z)\; T(\partial Z/\partial T)_P \qquad (17.1.89)$$

where A_Z is the term given previously for the compressibility factor.

These equations for the speed of sound appear, perhaps, unnecessarily complicated in that the local speed of sound, required in the swept wing computations, could be approximated by

$$a^2 = 1.3P_2 / \rho_2 \qquad (17.1.90)$$

without introducing a significantly large error into the program. However, as will be seen in a following section of CHEMP, the only additional formulation required for the speed of sound which is not required by the rest of the subroutine is the equations for γ and $(\partial Z/\partial P)_T$. Accordingly the complexity of the speed of sound was retained simply because the equations are a requirement for another section of CHEMP.

The transport properties of a gas are those properties which determine the change in the internal dynamic flux due to collisions and reactions or, in other words, they define the transfer or transport of molecular mass, momentum, and energy. Mass transport is defined in terms of diffusion, momentum transport in terms of viscosity, and energy transport in terms of thermal conductivity. In terms of the heat transfer equations used in this program, diffusion and thermal conductivity are only applied implicitly in that they define two important transport parameters, the Prandtl and Lewis numbers. Although these parameters were noted previously they, in conjunction with the viscosity, will be treated more thoroughly in this section.

The transport properties of low temperature air have been relatively well defined for a number of years but, in contrast to the fairly satisfactory state of development in regard to the thermodynamic properties, knowledge of high temperature transport properties is in a relatively elementary state. Of the many techniques presently available for computing these properties, those of Reference (15) are probably the most reliable. Because of the complexity of the equations given in Reference (15), however, this program has relied heavily upon the procedures of References (14) and (16) which do not differ greatly from those of Reference (15).

The viscosity of low temperature, undissociated air is given by Sutherland's equation as

$$\mu = 2.27 \times 10^{-8} \frac{T^{1.5}}{T + 198.6} \tag{17.1.91}$$

which is used to determine the viscosity throughout this program. The viscosity of dissociated, ionized air was obtained from Reference (58) which approximated it by

$$\frac{\mu}{\mu_o} = \left\{ 1 + .023 \frac{T}{1800} \left[1 + \tanh \left\{ \frac{\frac{T}{1800} \left[(1 - .125 \mathrm{Log}_{10}(P/P_0) \right] - 6.5}{1.5 + .125 \mathrm{Log}_{10}(P/P_0)} \right\} \right] \right\} \Big/ \left[1 + \exp \left(\frac{\frac{T}{1800} - 14.5 - 1.5 \mathrm{Log}_{10}(P/P_0)}{0.9 + 0.1 \mathrm{Log}_{10}(P/P_0)} \right) \right] \tag{17.1.92}$$

where μ_o is Sutherland's equation above. This equation has not been programmed in this heating subprogram because of the other approximations made with the transport properties but it was used when making the comparison between the constant and variable Lewis numbers in the Fay and Riddell equation.

The Prandtl number, as used in the heat transfer equations, is defined as

$$\bar{P}_r = \frac{\bar{C}_p \mu}{\bar{K}} \tag{17.1.93}$$

where $\bar{C}_p$ and $\bar{K}$ symbolize the frozen specific heat and thermal conductivity. The frozen values result from the fact that in considering the definition of the heat transfer in its most basic form,

$$q = \frac{K \, \partial T}{\partial y} \tag{17.1.94}$$

the thermal conductivity, K, can be rewritten as

$$K = \overline{K} + K_r \tag{17.1.95}$$

where $\overline{K}$ is the frozen thermal conductivity due to molecular collisions and K_r is the reaction thermal conductivity due to mass and chemical diffusion. In solving the energy flux equations, the frozen and reactions terms are considered separately and the analytical equations resulting from these solutions are generally expressed in such a way that the transport properties are expressed in terms of the frozen chemical properties.

Since pressure obviously has little effect on the frozen Prandtl number, it was curve fitted as a function of enthalpy at a pressure ratio of approximately 0.01 atmospheres as given below.

$\underline{H \leq 1.5}$

$$\overline{P}_r = 0.83854-0.615H_1+0.7544H_1^2-0.31888H_1^3+0.04388H_1^4 \tag{17.1.96}$$

$\underline{1.5 < H_1 \leq 30}$

$$\overline{P}_r = 0.75858+9.2825 \times 10^{-3}H_1-1.98875 \times 10^{-3}H_1^2+9.50557 \times 10^{-5}H_1^3 - 1.40088 \times 10^{-6}H_1^4 \tag{17.1.97}$$

where

$$H_1 = H/10^7 \tag{17.1.98}$$

Because the variation of the Prandtl number is small, it was not programmed but again was used in the variable Lewis number comparison.

The Lewis number, noted in this program, is defined as

$$Le_w = \frac{D\,\rho\overline{C}_p}{K} \tag{17.1.99}$$

where D is the binary diffusion coefficient. From Reference (14) this coefficient can be approximated by

$$D\rho = 1.46775\ \frac{\mu_0}{ZS} \tag{17.1.100}$$

and thus

$$Le_w = 1.46775\ \frac{\overline{P}_r\mu_0}{ZS\mu} \tag{17.1.101}$$

where

$$S = 0.9245-5.9214\times10^{-2}T_1+9.6307\times10^{-3}T_1^{\,2}-1.1901\times10^{-3}T_1^{\,3}+8.9775\times10^{-5}T_1^{\,4}$$
$$-3.5915\times10^{-6}T_1^{\,5}+5.7939\times10^{-8}T_1^{\,6} \qquad (17.1.102)$$

and

$$T_1 = T/10^3 \qquad (17.1.103)$$

$\bar{P}_r$ indicates that the Prandtl number is frozen. Again this parameter was not programmed but was only used for the variable Lewis number comparison.

There are obviously significant differences between the real or imperfect gas properties and the calorically imperfect (those properties used in the previous heating subprogram) and perfect gas properties. Real gas effects on the heat transfer, however, are not nearly as pronounced because the discrepancies tend to have a compensating effect and the errors incurred are generally not excessive. The real gas equations were retained in this program, because of the increased accuracy afforded by them.

As long as the continuum, chemical equilibrium restrictions on the real gas equations are satisfied, they may be used to obtain the properties of the freestream, inviscid shock layer, and boundary layer, the only flow fields of significance in this program. The freestream properties, however, are computed in the atmosphere subroutines within the SDF and TOP programs and hence will not be considered further.

The boundary layer properties are considered to be those properties at the inner edge of the boundary layer or at the surface. From the real gas equations all of the required thermodynamic and transport properties are determined when the pressure and temperature are known. The wall temperature is readily determined either as an initial input to the program or, being the variable of immediate importance, through the integration subroutines within the SDF and TOP programs proper. The wall or surface pressure is assumed to be the local pressure computed in the heating subprogram proper as the pressure gradients through the boundary layer are generally extremely small in a continuum flow.

The inviscid shock layer properties are considered to be those properties at the outer edge of the boundary layer and are referred to as the local properties. Again all of the chemical properties are determined whenever the pressure and temperature are known. The local pressure is obtained from the equations presented previously but it is the local enthalpy rather than the local temperature which is accessible from the subprogram proper. Thus, as a matter of convenience it would be more desirable to express the real gas equations as a function of enthalpy and pressure, in direct conflict with the boundary layer requirements. Various methods of obtaining the real gas equations as functions of enthalpy and pressure were examined, i.e. References (10) and (17), but in general these techniques either required considerable machine storage and/or afforded neither the accuracy nor the reliability available with the equations presented in this program. In addition the use of two separate procedures was somewhat impractical considering the limitations already imposed on this subprogram. Consequently when pressure and enthalpy, as the independent variables, are used in conjunction with the real gas equations given previously, an iterative procedure is required to compute the chemical properties.

Although the SDF and TOP programs contain an iteration subroutine CONVRG, this subroutine was not used for the iteration required by the aforeemntioned equations. The technique used in CONVRG is not particularly fast and is susceptible to occasional divergence. The iteration procedure used in CHEMP is a numerical integration technique employing the Runge-Kutta second-order formula. Although this technique possibly requires slightly more machine storage than CONVRG, it has the added advantage of a rapid solution and, in the suborbital flight regime was always found to be convergent. In terms of the symbolism used previously in this program, the Runge-Kutta formula is

$$T_{n+1} = T_n + .5(K_1 + K_2) \tag{17.1.104}$$

where

$$K_1 = \frac{H_{n+1} - H_n}{dH_n/dT} \tag{17.1.105}$$

and

$$K_2 = \frac{H_{n+1} - H_n}{DH_n/dT} \tag{17.1.106}$$

This technique involves the use of the enthalpy derivative but, since the pressure is held constant while the iteration is performed, this derivative is actually C_p which was defined previously in the speed of sound formulation. Most of the terms contained in the C_p equations have been previously defined for the enthalpy equations and thus the use of this derivative is not overly prohibitive.

The manner in which this procedure is utilized is as follows: Subprogram HETS enters subroutine CHEMP with a known value of H and P at the given flight condition and desires to find a value of T corresponding to H and P. Since HETS also enters CHEMP with a value of T corresponding to the preceding flight condition, CHEMP designates T as T_n and proceeds to compute H_n which it then compares with H. If the difference between H_n and H is within the set tolerance then CHEMP sets T_{n+1} equal to T_n and proceeds to compute the other chemical properties. If the difference is not within the required tolerance then CHEMP computes dH_n/dT and

$$K_1 = \frac{H-H_n}{dH_n/dT} \tag{17.1.107}$$

and sets

$$T_{n+K_1} = T_n + K_1 \tag{17.1.108}$$

The value of H_{n+K_1} is computed and again compared to H. If the required tolerance is met then CHEMP sets $T_{n+1} = T_{n+K_1}$ and proceeds as above. If not then dH_{n+K_1}/dT and

$$K_2 = \frac{H-H_n}{dH_{n+K_1}/dT} \qquad (17.1.109)$$

are computed and CHEMP sets

$$T_{n+1} = T_{n+K_1} + .5(K_2-K_1) \qquad (17.1.110)$$

If the required tolerance is still not met then CHEMP sets

$$T_n = T_{n+1} \qquad (17.1.111)$$

and the entire process is repeated.

17.1.5 Ideal Gas Properties

The calculation of real gas properties, especially the iteration for T as a function of H, uses a significant part of the computing time required for heating calculations because it is repeated so often. It may sometimes be desirable to reduce the computing time by changing to the simpler but less exact ideal gas properties. This option has been added to the program, and will be used in heating calculations unless real gas properties are specified by input. The equations are

If temperature is given:

$$H = 6008.\ T \qquad (17.1.112)$$

If enthalpy is given

$$T = H/6008. \qquad (17.1.113)$$

$$\rho = 1.232819 \cdot P/T \qquad (17.1.114)$$

$$\mu = 2.27 \times 10^{-8}\ T^{1.5}/(T + 198.6) \qquad (17.1.115)$$

$$a = 49.022 \cdot T^{1/2} \qquad (17.1.116)$$

17.1.6 Radiation Equilibrium Temperature

A vehicle designed for radiation cooling is likely to have a very thin wing skin, with small heat capacity. If the actual skin thickness is used in the transient skin temperature calculation, an integration step size smaller than that required by the trajectory integration may be required by the transient temperature integration, with a corresponding increase in the amount of calculation. This difficulty can be reduced by assuming a larger skin thickness, with greater heat capacity. Another possibility is to assume zero heat capacity, and solve for the equilibrium temperature at which the convective and radiative heating rates balance. The integration of the transient temperature differential equation is replaced by an iterative solution of a nonlinear algebraic equation for net heat flux. This should save computing if too many iterations are not needed, and may be closer to the right answer than the transient temperature of a thicker skin would be.

The heating routine has been modified to calculate equilibrium temperature instead of transient temperature for the wing skin when the skin thickness is zero. As previously noted, the iteration for equilibrium temperature in the previous optimization program sometimes did not work very well. An improved iteration method is used in the present program. The method of false position is used with the Aitken δ^2 process to improve convergence. The net heating rate equation

$$q_{net}(T_s) = q_c - q_r \qquad (17.1.117)$$

is solved with trial values of T_s until q_{net} is zero within a tolerance ε_q. The tolerance is the smaller of

$$\varepsilon_q = .001\left(\frac{q_c + q_r}{2}\right) \qquad (17.1.118)$$

and

$$\varepsilon_q = .01\ (4\varepsilon\sigma\ T_s^3) \qquad (17.1.119)$$

The sequence of trial values is generated in the following way. An initial value of T_{s_1} and a slightly perturbed value (T_{s_2}) are used to calculate the corresponding values of q_{net_1} and q_{net_2}. The method of false position given a third trial value

$$T_{s_3} = F\ (T_{s_1}, T_{s_2}) \qquad (17.1.120)$$

where

$$F(r,s) = \frac{r\ q_{net}(s) - s\ q_{net}(r)}{q_{net}(s) - q_{net}(r)} \qquad (17.1.121)$$

T_{s_3} and one of the pair (T_{s_1}, T_{s_2}) are then used to find a new trial value by the same method. Let the value of T_{s_1} or T_{s_2} which was used be called T_{s_4}, let the one not used be T_{s_z}, and call the new value T_{s_5}. Then

$$T_{s_5} = F\ (T_{s_3}, T_{s_4}) \qquad (17.1.122)$$

T_{s_z} and the two values T_{s_3} and T_{s_5} which were generated by successive application of the method of false position then form a sequence from which an improved estimate T_{s_6} is generated by Aitkens δ^2 process.

$$T_{s6} = D(T_{s_2}, T_{s3}, T_{s_5}) \qquad (17.1.123)$$

where

$$D(r, s, t) = t - \frac{(t-s)^2}{t-2s+r} \qquad (17.1.124)$$

One of the set $(T_{s_2}, T_{s3}, T_{s_5})$ and the last trial T_{s6} are then used to make a new pair (T_{s1}', T_{s_2}'), and the sequence begins again at equation (120). This procedure is repeated until q_{net} is zero within the tolerance ε_q. The method has proven reliable and uses less computing time than the transient temperature calculation.

REFERENCES:

1. Hague, D. S., Three Degree of Freedom Problem Optimization Formulation, FDL-TDR-64-1, Part I, Vol. 3, 1964.

2. Goldsmith, Alexander, Herschorn, Harry J., and Waterman, Thomas E., Thermophysical Properties of Solid Materials, WADC-TR 58-476.

3. Eckert, Ernst R. O., Survey on Heat Transfer at High Speeds, ALD-189.

4. ———., Aerodynamic Heat Transfer Handbook, Volume I, Boeing Report D2-9514.

5. Beckwith, I. W. and Gallagher, J. J., Local Heat Transfer and Recovery Temperatures on a Yawed Cylinder at Mach Number 4.15 and High Reynolds Numbers, NASA TR-104, 1961.

6. Bertram, M. H. and Everhard, P. E., An Experimental Study of the Pressure and Heat Transfer Distribution on a 70° Sweep Slab Delta Wing in Hypersonic Flow, NASA TR-153, 1963.

7. Wittliff, C. E. and Curtiss, J. T., Normal Shock Wave Parameters in Equilibrium Air, CALL-111, November 1961.

8. Deem, R. E., Erickson, C. R. and Murphy, J. S., Flat-Plate Boundary-Layer Transition at Hypersonic Speeds, FDL-TDR-64-129, October 1964.

9. Sieron, Thomas R., Methods for Predicting Skin Friction Drag from Subsonic to Hypervelocity Speeds on a Flat Plate and Delta Wings, ASR-MDF TM 62-45, 1962.

10. Sieron, Thomas R. and Martinex, Conrad, Effects and Analysis of Mach Number and Reynolds Number on Laminar Skin Friction at Hypersonic Speeds, AFFDL-TR-65-5, April 1965.

11. Cohen, Nathaniel B., Boundary-Layer Similar Solutions and Correlation Equations for Laminar Heat Transfer Distribution in Equilibrium Air at Velocities up to 41,100 Feet Per Second, NASA TR R-118, 1964

12. Fay, J A. and Riddell, F. R.,"Theory of Stagnation Point Heat Transfer in Dissociated Air," *Journal of the Aeronautical Sciences*, February 1958.

13. Detra, R. W., Kemp, N. H., and Riddell, F. R., Addendum to Heat Transfer to Satellite Vehicles Re-entering the Atmosphere, Jet Propulsion Laboratory, December 1957.

14. Hansen, C. Frederick, "Approximations for the Thermodynamic and Transport Properties of High Temperature Air," NASA TRR-50, 1959.

15. Peng, T. C. and Pindroh, A. I., An Improved Calculation of Gas Properties at High Temperatures: Air," Boeing Document D2-11722, February 1962.

16. Hansen, C. F. and Heims, S. F., A Review of the Thermodynamic Transport and Chemical Reaction Rate Properties of High Temperature Air, NACA TN 4359, July 1958.

17. Moretti, G., San Lorenzo, E. E., Magnus, E. E., and Weilerstein, G., Supersonic Flow About General Three-Dimensional Blunt Bodies, Volume III, Flow Field Analysis of Re-Entry Configurations by a General Three-Dimensional Method of Characteristics, ASD TR 61-727, October 1962.

17.2 PROGRAM ABLATOR: ONE-DIMENSIONAL ANALYSIS OF THE TRANSIENT RESPONSE OF THERMAL PROTECTION SYSTEMS

ABLATOR is a Langley Research Center program developed by Robert T. Swann, Claud M. Pittman, and James C. Smith. The description provided below is taken directly from their original report, NASA TN D-2976.

The original documentation provides additional information on program ABLATOR including

1. Finite difference equation development

2. Results

3. Comparison with more simple models

4. Methods for reducing computer time required

This section presents the general method of analysis employed. For further details, reference should be made to the original source document referenced above.

The thermal protection system that is to be analyzed is shown schematically in figure 1. Although this discussion is confined to a charring ablator system, all the concepts and equations apply equally as well to any other thermal protection system composed of not more than three primary layers. For a charring ablator system, the outer (heated) layer is the char, the center layer contains the uncharred material, and the third layer consists of insulation. Heat sinks can be located at the back of the second or third layers or at both locations.

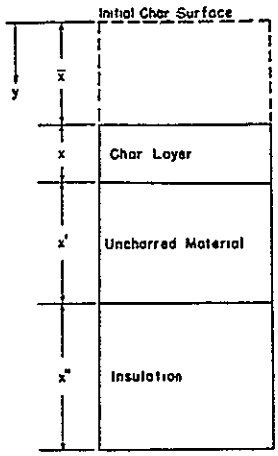


Figure 1.- Schematic diagram of system employing charring ablator.

The outer (char) surface is subjected to aerodynamic heating. The char layer provides both insulation and a high-temperature outer surface for reradiation. The heat passing through this layer is partially absorbed by pyrolysis at the interface between the char layer and the uncharred material, and the remaining heat is conducted into the uncharred material. The gases generated by pyrolysis transpire through the char layer and are injected into the boundary layer. The gases are heated as they pass through the char, and this heat removal from the char layer reduces the quantity of heat conducted to the pyrolysis interface. When these gases are injected into the boundary layer, the convective heat transfer is reduced. This reduction in convective heating is the same effect as that obtained with simple subliming ablators. In addition to the gases produced by pyrolysis, the carbonaceous residue remaining at the interface adds to the thickness of the char layer. While the processes of pyrolysis, transpiration, and injection are underway, char removal may also be taking place as a result of thermal, chemical, or mechanical processes. Thus the total char thickness may increase or decrease depending on the relative rates of formation and removal of the char. The various processes discussed are related quantitatively in the following sections.

17.2.1 Differential Equations

It is assumed that thermal properties in a given layer of material are functions only of temperature, that all heat flow is normal to the surface, and that gases transpiring through the char are at the same temperature as the char. Then the governing differential equations (from ref. 9) for the char layer $\left(\bar{x} \leqq y \leqq \bar{x} + x\right)$ are as follows:

$$\frac{\partial}{\partial y}\left(k \frac{\partial \theta}{\partial y}\right) \quad + \quad \dot{m}_p \bar{c}_p \frac{\partial \theta}{\partial y} \quad + \quad F \quad = \quad \rho c_p \frac{\partial \theta}{\partial t} \tag{17.2.1}$$

Heat conducted — Heat absorbed by transpiring gases — Heat generated — Heat stored

for the uncharred layer $\left(\bar{x} + x \leqq y \leqq x_o + x_o'\right)$,

$$\frac{\partial}{\partial y}\left(k' \frac{\partial \theta'}{\partial y}\right) = \rho' c_p' \frac{\partial \theta'}{\partial t} \tag{17.2.2}$$

and for a layer of insulation $\left(x_o + x_o' \leqq y \leqq x_o + x_o' + x_o''\right)$,

$$\frac{\partial}{\partial y}\left(k'' \frac{\partial \theta''}{\partial y}\right) = \rho'' c_p'' \frac{\partial \theta''}{\partial t} \tag{17.2.3}$$

The thicknesses of the layers to which the first two of these equations apply vary with time in a manner which is determined by the boundary conditions.

17.2.2 Initial Conditions

The initial temperature distribution is assumed to be given as a function of position:

$$\theta(y,0) = g(y) \tag{17.2.4}$$

The initial mass-transfer rates must also be specified. It should be noted that these values can be other than zero for some cases.

17.2.3 Surface Boundary Conditions

Two conditions must be specified at the heated surface. One must determine either the rate of removal of material at the surface or the temperature of the surface; the other is provided by the energy balance.

Surface ablation.- In general, the relative importance of the mechanisms involved in char removal from specific materials is not well established at this time. It has been established, however, that oxidation of the char surface is one important mechanism. Spalling of the char as a result of internal pressure is observed in some cases. Ablation at a given temperature (that is, sublimation) occurs if the heating rate is sufficiently high. Ablation of the surface may also occur as a result of aerodynamic shear stresses.

To provide maximum flexibility, provision is made for the following mechanisms of surface erosion:

(1) Ablation at a given temperature which may be a function of ablation rate (sublimation)

(2) Removal of char at a rate which is a given function of time (spalling, aerodynamic shear)

(3) Removal of char at such a rate that the char thickness is a given function of time (spalling, aerodynamic shear)

(4) Ablation as a result of a chemical process (oxidation)

For ablation at a given temperature, two cases are considered. In one case, ablation occurs at a fixed temperature. In the other case, the char mass loss rate is an exponential function of the surface temperature. For ablation at a fixed temperature, no surface erosion occurs if the calculated surface temperature is less than the specified ablation temperature. If the calculated surface temperature is higher than the ablation temperature, ablation occurs at a rate sufficient to reduce the temperature to the ablation temperature; that is, $\dot{m}_c$ is equal to zero for $T_1 < \bar{T}_1$, and $\dot{m}_c$ is calculated from an energy balance at the surface for $T_1 = \bar{T}_1$. In the second case, char mass loss rate and surface temperature are related as follows:

$$\dot{m}_c = Ae^{-B/T_1} \tag{17.2.5}$$

An equation of this form has some physical significance, because decomposition reactions proceed more rapidly at higher temperatures. By an appropriate selection of A and B, equation (5) yields results similar to those obtained by specifying an ablation temperature.

If the rate of char removal is given function of time

$$\dot{m}_c = f(t) \tag{17.2.6}$$

then the rate of char removal is obtained from the input data and the surface temperature is calculated from an energy balance. Such a relation might be used to compare calculated and experimental results when a more basic quantitative relation for the experimental rate of char removal is not available.

If the char thickness is a given function of time

$$x = f(t) \tag{17.2.7}$$

then the rate of char removal is calculated from this relation together with the rate of char formation which is calculated from the conditions at the pyrolysis interface. This condition can be used when it is desirable to perform calculations for applications in which the char thickness is known as a function of time even though mechanisms of char removal may be present which cannot be expressed quantitatively.

It has been shown experimentally that oxidation is an important mechanism of char removal. (See ref. 11.) For a half-order reaction, the rate of oxidation of carbon can be determined from the following equation (ref. 14):

$$\dot{m}_C = Ae^{-B/T_1}\sqrt{C_w p_w} \tag{17.2.8}$$

The pressure at the wall must be specified for subsonic and supersonic flow. However, in hypersonic flow, the pressure can be related to the stagnation heating rate and enthalpy. The stagnation pressure in hypersonic flow is approximately (see ref. 15):

$$p_{w,s} = \frac{11}{12}\rho_\infty V_\infty^2 \tag{17.2.9a}$$

Further (from ref. 16),

$$q \propto \sqrt{\frac{\rho_\infty}{R}}\, V_\infty^3 \tag{17.2.9b}$$

and

$$h_e \propto V_\infty^2 \tag{17.2.9c}$$

Then, $p_{w,s}$ can be expressed by the relation

$$p_{w,s} = GR\left(\frac{q_s}{h_{e_s}}\right)^2 \tag{17.2.9d}$$

where the constant of proportionality is

$$G = 410.72\ \frac{\text{ft}^3\text{-sec}^2\text{-atm}}{\text{lb}^2} = 56.2\ \frac{\text{m}^3\text{-sec}^2\text{-atm}}{\text{kg}^2}$$

The pressure at the wall is therefore given by

$$p_w = G\,\frac{p_w}{p_{w,s}}\,R\left(\frac{q_s}{h_{e_s}}\right)^2 \tag{17.2.9e}$$

where $\frac{p_w}{p_{w,s}}$ depends on vehicle attitude and body location. The rate at which char is removed by oxygen must be proportional to the net rate at which oxygen diffuses to the surface. From reference 17 this rate is

$$\dot{m}_{ox} = \frac{N_{Le}^{0.6}\, q_{C,net}}{h_e - h_w}\left(C_e - C_w\right) = \frac{\dot{m}_C}{\lambda} \tag{17.2.10a}$$

As shown in a subsequent section, $q_{C,net}$ is the hot-wall convective heating rate corrected for transpiration (see eq. (13)); that is,

$$q_{C,net} = q_C\left(1 - \frac{h_w}{h_e}\right)\left\{1 - (1 - \beta)\left[0.724 \frac{h_e}{q_C}\left(\alpha_c\dot{m}_c + \alpha_p\dot{m}_p\right)\right.\right.$$

$$\left.\left. -\,0.13\left(\frac{h_e}{q_C}\right)^2\left(\alpha_c\dot{m}_c + \alpha_p\dot{m}_p\right)^2\right] - \beta\eta\left(\alpha_c\dot{m}_c + \alpha_p\dot{m}_p\right)\frac{h_e}{q_C}\right\} \qquad (17.2.10\ b)$$

By eliminating the concentration of oxygen at the wall C_w in equations (8) and (10a), the rate of removal of char by oxidation is found to be

$$\dot{m}_c = \frac{1}{2}\left\{-\frac{(h_e - h_w)K^2p_w}{q_{C,net}\lambda N_{Le}^{0.6}} + \sqrt{\left[\frac{(h_e - h_w)K^2p_w}{q_{C,net}\lambda N_{Le}^{0.6}}\right]^2 + 4K^2p_wC_e}\right\} \qquad (17.2.10\ c)$$

where $K = Ae^{-B/T_1}$.

The equation for a first-order oxidation reaction is obtained similarly. The resulting equation is

$$\dot{m}_c = \frac{Kp_wC_e}{1 + \dfrac{Kp_w(h_e - h_w)}{q_{C,net}\lambda N_{Le}^{0.6}}} \qquad (17.2.10\ d)$$

Surface location.- When char removal occurs, the char surface moves with respect to a coordinate system fixed in the material. The distance between the surface of the char and the initial surface location is given by

$$\bar{x} = \int_0^t \frac{\dot{m}_c}{\rho}\,dt \qquad (17.2.11)$$

The thickness of the char at any time is equal to the initial char thickness, plus the thickness of char formed by pyrolysis, less the thickness of char removed; that is,

$$x = x_0 + \int_0^t \frac{\dot{m}_p}{\rho' - \rho}\,dt - \int_0^t \frac{\dot{m}_c}{\rho}\,dt \qquad (17.2.12)$$

Surface energy balance.- The heat input consists of convective and radiant heating. This energy must be accommodated at the surface by a combination of four mechanisms:

(1) **Blocking by mass transfer into the boundary layer**

(2) Reradiation or reflection from the surface

(3) Conduction into the material

(4) Sublimation of the char

The effect of mass transfer on heat transfer has been studied extensively. With low-mass-transfer rates it is found that the reduction in heat-transfer rate is directly proportional to the product of the mass-transfer rate and the enthalpy difference across the boundary layer. With high-mass-transfer rates, which may occur when a large fraction of the heat input is radiant, the linear approximation is no longer adequate and it is necessary to use a higher order approximation. A second-degree approximation is derived in appendix A.

The surface energy balance, expressed in a form in which either approximation to the blocking effectiveness can be selected, is as follows:

$$\underbrace{\underbrace{q_C}_{\substack{\text{Cold wall}\\ \text{convective}\\ \text{heating rate}}} \underbrace{\left(1 - \frac{h_w}{h_e}\right)}_{\substack{\text{Hot wall}\\ \text{correction}}} \underbrace{\left\{1 - (1 - \beta)\left[0.724 \frac{h_e}{q_C}\left(\alpha_c m_c + \alpha_p m_p\right) - 0.13\left(\frac{h_e}{q_C}\right)^2\left(\alpha_c m_c + \alpha_p \dot{m}_p\right)^2\right] - \beta\eta\left(\alpha_c \dot{m}_c + \alpha_p \dot{m}_p\right)\frac{h_e}{q_C}\right\}}_{\text{Aerodynamic blocking}}}_{\text{Net convective heating rate}}$$

$$+ \underbrace{\alpha q_R}_{\substack{\text{Radiative}\\ \text{heating rate}}} + \underbrace{\left[1 - S\left(\theta - \bar{T}_1\right)\right]\dot{m}_c \,\Delta h_c}_{\substack{\text{Combustive}\\ \text{heating rate}}} = \underbrace{\sigma\epsilon_1\theta_1{}^4}_{\text{Reradiation}} - \underbrace{k\frac{\partial\theta}{\partial y}}_{\substack{\text{Conduction}\\ \text{to interior}}} + \underbrace{S\left(\theta - \bar{T}_1\right)m_c H_c}_{\substack{\text{Heat of sublimation}\\ \text{of char}}} \qquad (17.2.13)$$

If transpiration theory (second-degree approximation, appendix A) is used, β is equal to zero. For linear ablation theory, β is equal to 1. In either case, the heat absorbed by vaporization of the char H_c and the heat of combustion of the char Δh_c are considered separately. The coefficients α_c and α_p can be used to differentiate between the blocking effectiveness of the gases produced at the surface and at the pyrolysis interface. Evaluation of these coefficients is discussed briefly in appendix A.

The heat transfer to the outer surface is assumed to be a given function of time and consists of the cold-wall convective heating rate q_C and the radiant heating rate q_R incident on the surface. These two components must be specified separately because mass transfer at the surface blocks part of the aerodynamic heating but, in general, has no effect on radiant heating. Additional terms can easily be included in equation (13) to account for other phenomena which may affect the heat balance at the char surface. For example, reference 18 discusses a gas-phase combustion in the boundary layer involving the gases of pyrolysis. This effect has not been clearly identified at the Langley Research Center and is, therefore, not included in the equation. However, phenomena such as this may be important in some cases and their existence should certainly be considered.

Equation (13) is normally used in this analysis as the boundary condition on the temperature at the outer surface. However, when θ is equal to the sublimation temperature $\overline{T}_l$, the specified sublimation temperature provides the boundary condition on the temperature and equation (13) is used to calculate the rate of ablation $\dot{m}_c$.

17.2.4 Pyrolysis-Interface Boundary Condition

Energy balance.- The heat conducted to the pyrolysis interface must be either absorbed by pyrolysis reactions or conducted into the uncharred material; that is, at $y = \bar{x} + x$,

$$-k \frac{\partial \theta}{\partial y} = \dot{m}_p \Delta h_p - k' \frac{\partial \theta'}{\partial y} \qquad (17.2.14)$$

In addition, the temperatures in the char and in the uncharred material must be equal at the interface; that is, at $y = \bar{x} + x$,

$$\theta = \theta' \qquad (17.2.15)$$

Pyrolysis rate.- Two approaches are available for calculating the rate of pyrolysis. In the first approach, it is assumed that pyrolysis occurs at a given temperature $\overline{T}_i$. If

$$\theta_{\bar{x}+x} < \overline{T}_i \qquad (17.2.16a)$$

then

$$\dot{m}_p = 0 \qquad (17.2.16b)$$

If

$$\theta_{\bar{x}+x} = \overline{T}_i$$

the temperature is known and equation (14) is used to calculate the rate of pyrolysis.

In an alternate approach, it is assumed that the rate of pyrolysis is a known function of temperature, for example

$$\dot{m}_p = A' e^{-B'/\theta_{\bar{x}+x}} \qquad (17.2.17)$$

when

$$\theta_{\bar{x}+x} < \bar{T}_i$$

In this case, equations (14) and (17) are solved for both temperature and pyrolysis rate. The value of $\bar{T}_i$ is still specified, and if this temperature is reached, the pyrolysis rate is determined only from equation (14) so that this temperature is not exceeded.

Pyrolysis-interface location.- As pyrolysis occurs, the interface between the char layer and the uncharred material moves with respect to a fixed coordinate. Its distance from the initial char surface location is

$$\bar{x} + x = x_o + \int_0^t \frac{\dot{m}_p}{\rho' - \rho} \, dt \qquad (17.2.18)$$

The instantaneous thickness of the uncharred material is

$$x' = x_o' - \int_0^t \frac{\dot{m}_p}{\rho' - \rho} \, dt \qquad (17.2.19)$$

17.2.5 Boundary Conditions at Back Surface of Ablation Material

A number of conditions can be imposed at the back surface of the ablation material, depending on whether additional insulation is provided or some provision is made for temperature control. Whether insulation is used or not, the ablation material may be attached to a thermally thin plate which functions as a concentrated heat sink.

Three-layer system.- If insulation is used, the temperature of the ablation material is equal to the temperature of the insulation at their interface.

$$\theta'_{x_o+x_o'} = \theta''_{x_o+x_o'} \qquad (17.2.20a)$$

From an energy balance at the back surface of the ablation material,

$$-k' \frac{\partial \theta'}{\partial y} = C_{i+j} \frac{\partial \theta'}{\partial t} - k'' \frac{\partial \theta''}{\partial y} \qquad (17.2.20b)$$

Two-layer system.- If no insulating layer is used, the back surface can be assumed to be perfectly insulated, cooled at a given temperature, or may exchange radiation with a sink of known temperature in the interior of the structure. An energy balance yields the following equation:

$$-k' \frac{\partial\theta'}{\partial y} = C_{i+j} \frac{\partial\theta'}{\partial y} + S\left(\theta' - \overline{T}_{i+j}\right)\Delta W_f \; \Delta h_f + \sigma\epsilon_{i+j}\left[(\theta')^4 - T_B^4\right] \quad (17.2.21)$$

The temperature at which the cooling system is activated is $\overline{T}_{i+j}$. The choice of conditions is accomplished by making the inapplicable terms equal to zero (that is, $C_{i+j} = 0$ and/or $\overline{T}_{i+j} > \theta'$ and/or $\epsilon_{i+j} = 0$).

17.2.6 Boundary Condition at Back Surface of Insulating Material

If an insulating material is used behind the ablating material, the boundary condition at the back surface $\left(y = x_o + x_o' + x_o''\right)$ is similar to equation (21); that is,

$$-k'' \frac{\partial\theta''}{\partial y} = C_{i+j+m} \frac{\partial\theta''}{\partial t} + S\left(\theta'' - \overline{T}_{i+j+m}\right)\Delta W_f \; \Delta h_f + \sigma\epsilon_{i+j+m}\left[(\theta'')^4 - T_B^4\right] \quad (17.2.22)$$

17.2.7 Transformation of Coordinates

The equations derived in the preceding discussion are similar to those presented in reference 9. In reference 9, these equations are expressed in finite-difference form and solved in a fixed coordinate system. To maintain a fixed number of stations in layers of varying thicknesses, it is necessary to change the locations of the stations and to interpolate to determine the temperatures at the new locations after each step in the calculation. This procedure not only increases the time required to perform the computations, but also introduces a small error in each step of the calculation. This difficulty can be eliminated by transforming the equations to a coordinate system in which the finite-difference stations remain fixed, and the coordinates themselves move to accommodate the changes in the locations of the surfaces of the different materials.

The y-coordinate can be transformed to ξ- and ζ-coordinates in the char and uncharred layers, respectively, by using the following equations:

$$\xi = \frac{y - \int_0^t \frac{\dot{m}_c}{\rho}\, dt}{x} \quad (17.2.23a)$$

$$\zeta = \frac{y - x_o - \int_0^t \frac{\dot{m}_p}{\rho' - \rho}\, dt}{x'} \quad (17.2.23b)$$

In this coordinate system the outer surface remains fixed at $\xi = 0$. The interface is located at $\xi = 1$ in the char and at $\zeta = 0$ in the uncharred material. The back surface of the uncharred material is located at $\zeta = 1$. A number of advantages result from the use of this double transformation. First, the char always extends from $\xi = 0$ to $\xi = 1$. Therefore, the temperatures tend to be more nearly steady state than would be the case with a coordinate system fixed at the surface only. A second advantage is the positive location of the pyrolysis interface. A similar transformation would also be very beneficial in locating the center of the reaction zone when the pyrolysis reactions are considered in detail. Because the reaction zone is typically very thin, a very fine finite-difference network is required to analyze it. With transformations similar to those here, the center of the reaction zone can be located, and the fine network can be restricted to this region rather than covering the entire range of possible reaction-zone locations.

In the transformed coordinate system, equations (1) and (2) are as follows (for the char layer and uncharred layer, respectively):

$$\frac{1}{x^2}\frac{\partial}{\partial\xi}\left(k\frac{\partial\theta}{\partial\xi}\right) + \frac{1}{x}\frac{\partial\theta}{\partial\xi}\left[\dot{m}_c c_p + \dot{m}_p\bar{c}_p + \xi\left(\frac{\dot{m}_p\rho}{\rho' - \rho} - \dot{m}_c\right)c_p\right] + F = \rho c_p\frac{\partial\theta}{\partial t} \quad (17.2.24a)$$

$$\frac{1}{(x')^2}\frac{\partial}{\partial\zeta}\left(k'\frac{\partial\theta'}{\partial\zeta}\right) + \frac{1}{x'}\frac{\dot{m}_p\rho' c_p'}{\rho' - \rho}(1 - \zeta)\frac{\partial\theta'}{\partial\zeta} = \rho' c_p'\frac{\partial\theta'}{\partial t} \quad (17.2.24b)$$

The boundary conditions are as follows:

At $\xi = 0$,

$$q_{aero} = \sigma\epsilon_1\theta^4 - \frac{k}{x}\frac{\partial\theta}{\partial\xi} + \dot{m}_c\left[S\left(\theta - \bar{T}_1\right)\left(H_c + \Delta h_c\right) - \Delta h_c\right] \quad (17.2.25a$$

where

$$q_{aero} = \alpha q_R + q_C\left(1 - \frac{h_w}{h_e}\right)\Bigg\{1 - (1 - \beta)\left[0.724\frac{h_e}{q_C}\left(\alpha_c\dot{m}_c + \alpha_p\dot{m}_p\right)\right.$$

$$\left. - 0.13\left(\frac{h_e}{q_C}\right)^2\left(\alpha_c\dot{m}_c + \alpha_p\dot{m}_p\right)^2\right] - \beta\eta\left(\alpha_c\dot{m}_c + \alpha_p\dot{m}_p\right)\frac{h_e}{q_C}\Bigg\} \quad (17.2.25b)$$

at $\xi = 1$, $\zeta = 0$,

$$\theta = \theta' \quad (17.2.26a)$$

and

$$-\frac{k}{x}\frac{\partial\theta}{\partial\xi} = \dot{m}_p\,\Delta h_p - \frac{k'}{x'}\frac{\partial\theta'}{\partial\zeta} \quad (17.2.26b)$$

and at $\zeta = 1$ for only two layers, the condition at the back surface $\left(y = x_o + x_o'\right)$ is

$$-\frac{k'}{x'}\frac{\partial\theta'}{\partial\zeta} = C_{i+j}\frac{\partial\theta'}{\partial t} + S\left(\theta' - \overline{T}_{i+j}\right)\Delta W_f\ \Delta h_f + \sigma\epsilon_{i+j}\left[(\theta')^4 - T_B^4\right] \quad (17.2.27)$$

When three layers are used, the conditions at the back of the second layer are

$$\theta' = \theta'' \quad (17.2.28a)$$

and

$$-\frac{k'}{x'}\frac{\partial\theta'}{\partial\zeta} = C_{i+j}\frac{\partial\theta'}{\partial t} - k''\frac{\partial\theta''}{\partial y} \quad (17.2.28b)$$

If three layers are used, the condition at the back surface $(y = x_o + x_o' + x_o'')$ is

$$-k''\frac{\partial\theta''}{\partial y} = C_{i+j+m}\frac{\partial\theta''}{\partial t} + S\left(\theta'' - \overline{T}_{i+j+m}\right)\Delta W_f\ \Delta h_f + \sigma\epsilon_{i+j+m}\left[(\theta'')^4 - T_B^4\right]$$

(17.2.29)

REFERENCES:

1. Swann, Robert T., Composite Thermal Protection Systems for Manned Re-entry Vehicles, *American Rocket Society Journal*, Vol. 32, No. 2, February 1962, pp. 221-226.

2. Chapman, Andrew J., Effect of Weight, Density, and Heat Load on Thermal-Shielding Performance of Phenolic Nylon, NASA TN D-2196, 1964.

3. Farmer, Rex W., Ablative Behavior of Plastics in Subsonic and Supersonic Hyperthermal Flow, WADD Technical Report 60-648, U. S. Air Force, November 1960.

4. Schwartz, H. S. (compiler), Conference on Behavior of Plastics in Advanced Flight Vehicle Environments, WADD Technical Report 60-101, U. S. Air Force, September 1960.

5. Brooks, William A., Jr., Wadlin, Kenneth L., Swann, Robert T., and Peters, Roger W., An Evaluation of Thermal Protection for Apollo, NASA TM X-613, 1961.

6. Schmidt, Donald L., Behavior of Plastic Materials in Hyperthermal Environments, WADC Technical Report 59-574, U. S. Air Force April 1960.

7. Munson, T. R., and Spindler, R. J., Transient Thermal Behavior of Decomposing Materials, Part I: General Theory and Application to Convective Heating, Paper No. 62-30, S.M.F. Publishing Fund, Institute of Aerospace Sciences, January 1962.

8. Kratsch, K. M., Hearne, L. F., and McChesney, H. R., Thermal Performance of Heat Shield Composites During Planetary Entry, Presented at AIAA-NASA National Meeting, Palo Alto, California, September 30-October 1, 1963.

9. Swann, Robert T and Pittman, Claud M., Numerical Analysis of the Transient Response of Advanced Thermal Protection Systems for Atmospheric Entry, NASA TN D-1370, 1962.

10. Beecher, Norman, and Rosenweig, Ronald E., Ablation Mechanisms in Plastics with Inorganic Reinforcement, *American Rocket Society Journal*, Volume 31, No. 4, April 1961, pp. 532-539.

11. Dow, Marvin B. and Swann, Robert T., Determination of Effects of Oxidation on Performance of Charring Ablators, NASA TR R-196, 1964.

12. Scala, Sinclaire M. and Gilbert, Leon M., Thermal Degradation of a Char-Forming Plastic During Hypersonic Flight, *American Rocket Society Journal*, Volume 32, No. 6, June 1962, pp. 917-924.

13. Mechtly, E. A., The International System of Units - Physical Constants and Conversion Factors, NASA SP-7012, 1964.

14. Nolan, Edward J. and Scala, Sinclaire M., Aerothermodynamic Behavior of Pyrolytic Graphite During Sustained Hypersonic Flight, *American Rocket Society Journal*, Volume 32, No. 1, January 1962, pp. 26-35.

15. Truitt, Robert Wesley, Hypersonic Aerodynamics, The Ronald Press Co., c. 1959.

16. Chapman, Dean R., An Approximate Analytical Method for Studying Entry Into Planetary Atmospheres, NASA TR R-11, 1959. (Supersedes NACA TN-4276)

17. Swann, Robert T., Approximate Analysis of the Performance of Char-Forming Ablators, NASA TR R-195, 1964.

18. Vojvodich, Nick S. and Pope, Ronald B., Effect of Gas Composition on the Ablation Behavior of a Charring Material, *AIAA Journal*, Volume 2, No. 3, March 1964, pp. 536-542.

19. Brooks, William A., Jr., Temperature and Thermal-Stress Distributions in Some Structural Elements Heated at a Constant Rate, NACA TN-4306, 1958.

20. Roberts, Leonard, Mass Transfer Cooling Near the Stagnation Point, NASA TR R-8, 1959, (Supersedes NACA TN 4391)

21. Roberts, Leonard, A Theoretical Study of Stagnation-Point Ablation, NASA TR R-9, 1959, (Supersedes NACA TN 4392)

22. Swann, Robert T. and South, Jerry., A Theoretical Analysis of Effects of Ablation on Heat Transfer to an Arbitrary Axisymmetric Body, NASA TN D-741, 1961.

23. Swann, Robert T., Effect of Thermal Radiation from a Hot Gas Layer on Heat of Ablation, *Journal of Aerospace Sciences*, Readers Forum, Volume 28, No. 7, July 1961, pp. 582-583.

24. Savin, Raymond C., Gloria, Hermilo R., and Dahms, Richard G., Ablative Properties of Thermoplastics under Conditions Simulating Atmosphere Entry of Ballistic Missiles, NASA TM X-397, 1960.

25. Dusinberre, G. M., Numerical Methods for Transient Heat Flow, Transactions of the American Society of Mechanical Engineers, Volume 67, No. 8, November 1945, pp. 703-712.

APPENDIX I

AN INTRODUCTION TO ODINÈX:

A DESIGN INTEGRATION AND ANALYSIS LANGUAGE

ABSTRACT

A design integration and analysis language has been implemented in a CDC 6000 series computer code called ODINEX. It controls the sequence of execution and data management function for a community of interdependent design computer programs. The language includes a FORTRAN-like loop and bypass logic on groups of independent programs. Each individual program constitutes a single member of the design network. As a result of this development of DIALOG, any existing checked out computer program is immediately available for inclusion in the community. Each program can access a dynamically maintained design data base which forms the common information link among the programs of the community.

TABLE OF CONTENTS

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES

APPENDIX I

AN INTRODUCTION TO ODINEX:
A DESIGN, INTEGRATION AND ANALYSIS LANGUAGE
WITH OPEN-ENDED GROWTH CAPABILITY

1. SUMMARY

A *program community* concept called ODIN has been implemented on the CDC 6600 computer which features

a. Multiple computer program execution
b. Mutual data communication among programs

The concept, shown schematically in Figure 1, allows interdependent design programs to be sequentially executed in a single job stream while maintaining individual program identity. Communication between programs is through a dynamically constructed design data base. Any subset of the total input or output from the individual programs may be communicated to the data base or from the data base to any of the other programs in the community.

ODINEX is the control and communication executive computer program which implements a *community of programs* concept. Figure 1 shows the relationship between ODINEX and the ODIN library of independent technology modules. ODINEX draws on the ODIN community for design elements and controls the computational sequence involved in synthesizing and optimizing a given vehicle design. All interdisciplinary data is stored in the design data base of engineering information. Any design element may access and modify the data base through the ODINEX executive.

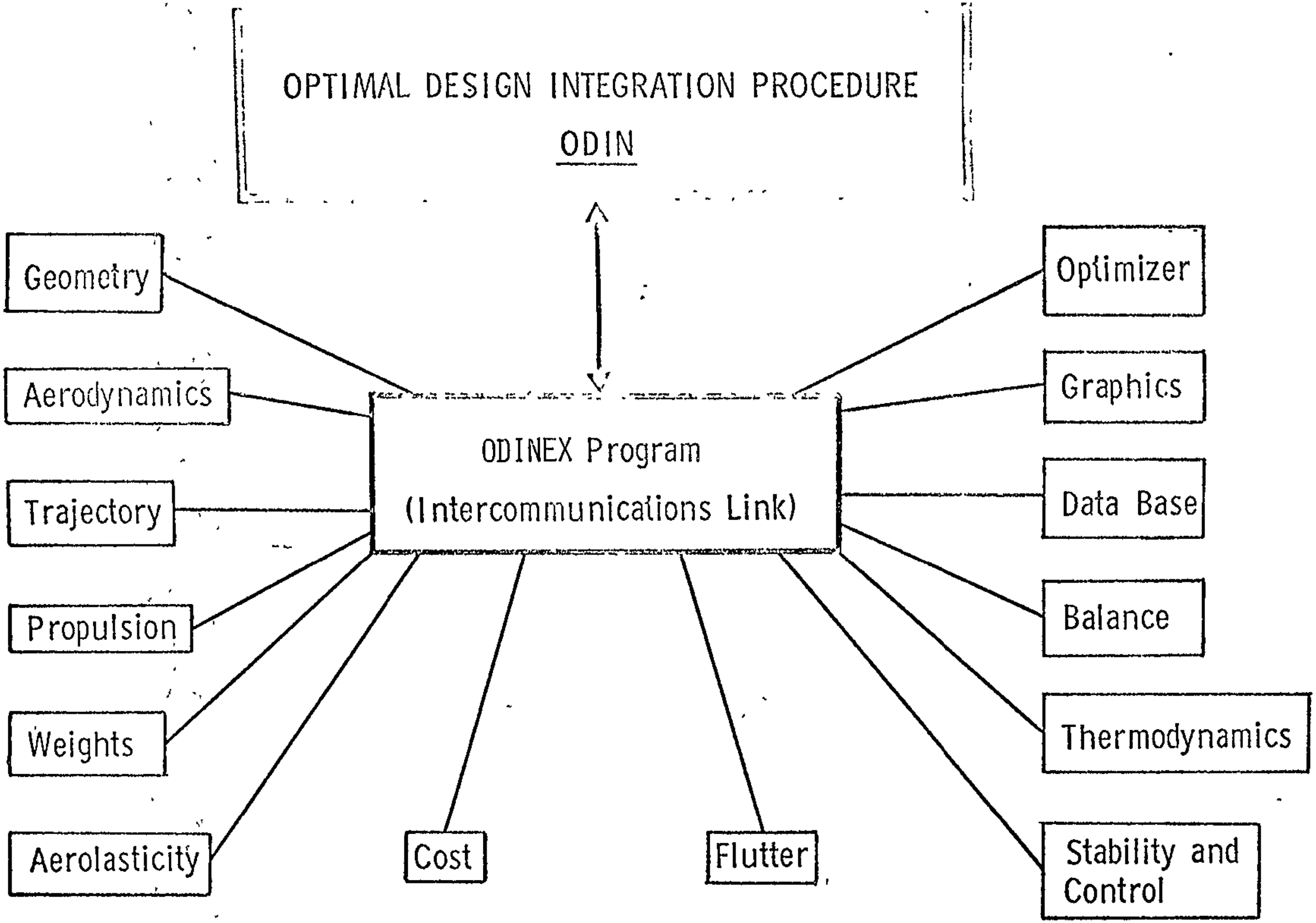


Figure 1. Schematic of the ODIN Concept

As a result of the development of ODINEX, any program can be included in the ODIN community once its interface requirements are established. The program intercommunication techniques consist of

1. *A language for controlling the execution* of an arbitrary network of independent programs by simple commands, as shown in Figure 2.

2. *A control card data base* for storing information with regard to the execution of individual programs. These data base files can be updated either by a separate run or dynamically in the simulation.

3. *A dynamically constructed data base* containing all inter-program data. These data can be saved at user-selected points in the simulation. The data base size can be adjusted by the user.

4. *A language for automatically retrieving* data base information as input to any program in the synthesis. An advanced information access and retrieval system was developed and included as an integral part of ODINEX. The language requires no modification to the ODIN program.

5. *A simple technique* for allowing any program in the synthesis to update the data base. The technique *does not influence* the normal stand-alone operation of the program.

6. *A user-oriented method* for generating reduced size/reduced scope modules from the parent programs.

7. *A capability for generating* one or more stylized reports as a part of the normal computer output.

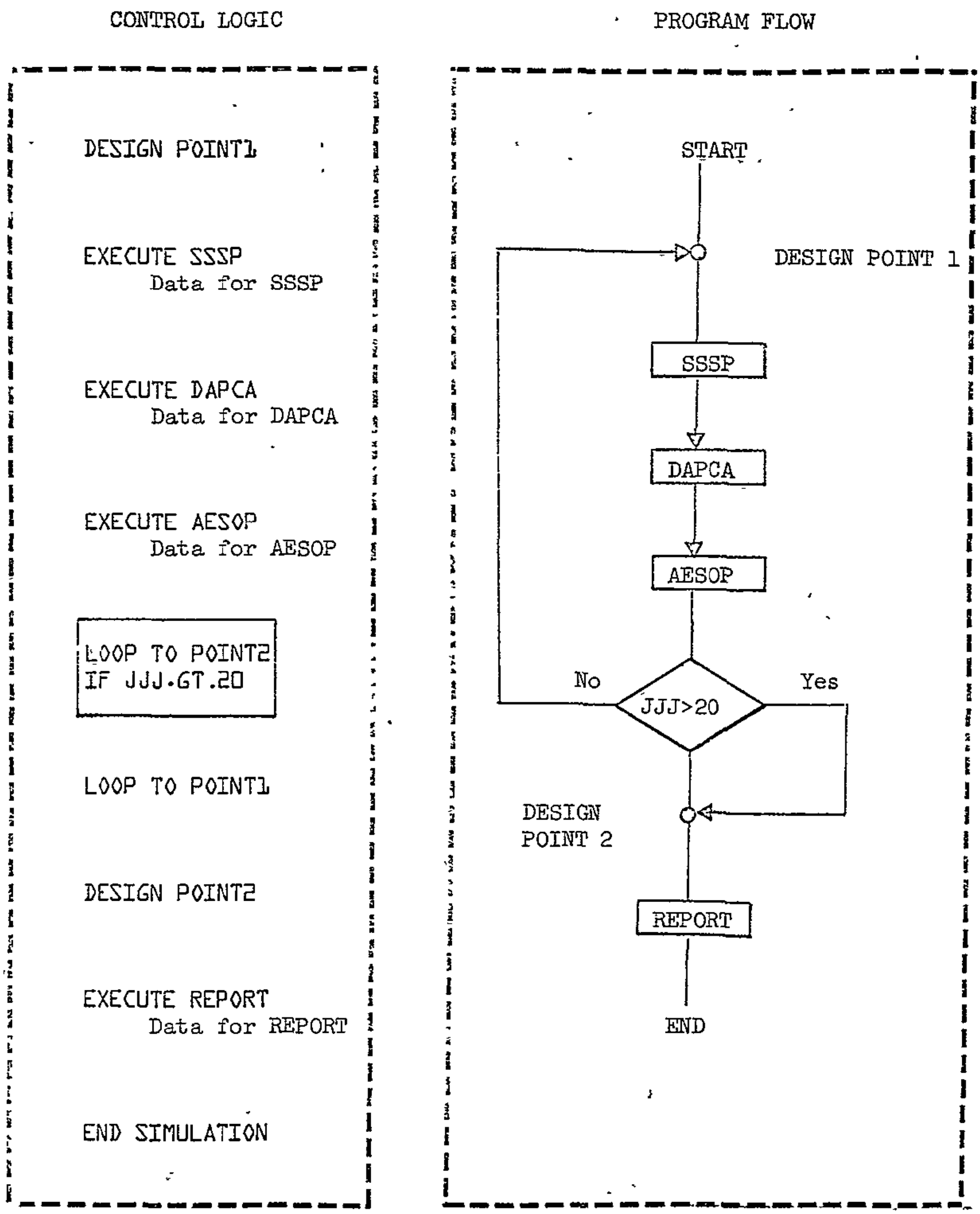


Figure 2. ODIN Control Logic

8. *The operational flexibility* of batch or interactive modes of operation.

All elements of the program intercommunication system are directly controlled by the independent executive program ODINEX. Significant advantages of this technique over a single design synthesis program are listed below.

1. Rapid response to ever changing design requirements. The user has the choice of design synthesis model complexity through replacement of independent functional modules or enrichment by inclusion of new or additional functional modules.

2. The data base reflects the status of the current design. Individual programs such as aerodynamics, structures, or graphics, can be exercised by using data base information without need for execution of other functional modules.

3. The developer of new technological modules is unconstrained by the requirements of the synthesis. New computer programs are immediately available for inclusion in the community of programs.

4. The elapsed time for design analysis is significantly reduced by performing the bulk of the information transfer in the computer. An improvement in data quality results from more accurate transfer of information.

5. The contributors to the design process place greater confidence in the results due to the use of proven independent technological modules.

2. INTRODUCTION

The design of an aerospace vehicle demands the involvement of specialists from all engineering disciplines. Many design iterations are usually required. Each discipline generally is constrained by the requirements of other disciplines, and much laborious data communication is required at each step. Automation of the individual disciplines has played a key role in the design process for more than a decade. Structural analysis and system performance have led the way in computer applications. More recent attempts at merging the technologies into a single preliminary design tool are exemplified by Reference 1. Here, a complete synthesis of the design and mission analysis is contained in a single computer program.

Concurrent with the development of integrated design computer programs were efforts at optimization of the designs themselves. A modular approach to optimization is reported in Reference 2. This approach was employed when the programs of Reference 1 and Reference 2 were coupled. The results reported in Reference 3 indicate that optimization of the design process is possible; in a mathematical sense it represents a much simpler optimization problem than many single discipline problems.

The confidence gained in early simulation attempts has led to the development of more detailed and complex modules. References 4 through 6 are examples of advanced simulation programs. However, most modern day programs tend to suffer from one or more of the following discrepancies:

1. Lack of depth in analysis
2. Insufficient data intercommunication
3. Poor response to rapidly changing requirements

The practical value of a simulation technique is measured by its useful life. That is, it should be open ended from the point of view of additions, deletions, substitutions, and improvements in engineering capability. The techniques should impose no constraints on development in new technology areas. One recent development reported in Reference 7 addressed these problems by retaining the functional identity of individual technologies but demanded a special input-output format supplied by the executive. The ODIN concept implemented by the ODINEX is an effort to overcome most of the shortcomings inherent in its predecessors.

3. GENERAL DESCRIPTION OF ODINEX

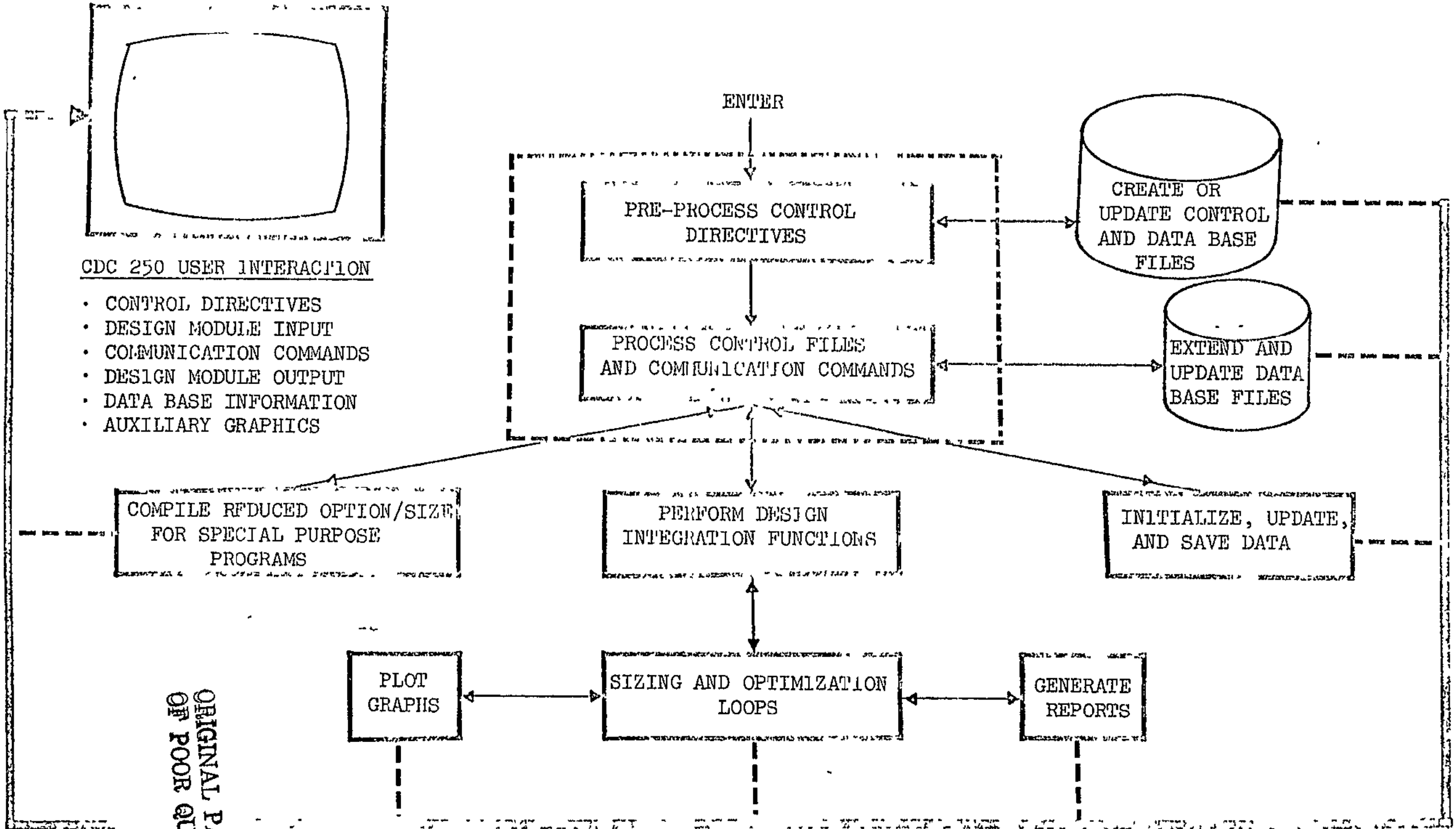
An independent executive computer program called ODINEX has been developed for linking separately developed computer programs in an arbitrary network, the objective being to study complex engineering systems whose elements are represented by other computer programs. Any subset of the incoming or outgoing data of a member program may be communicated to the other members through ODINEX. The objective of the development is to implement the Optimal Design INtegration concept, ODIN.

3.1 ODIN Concept

The ODIN concept allows a *community* of interdependent design, mission and sizing programs to be sequentially executed in an arbitrary network while maintaining full individual program identity. Communication between programs is maintained through a dynamically constructed *design data base*. Any subset of the total input or output from the individual programs may be communicated to the data base, and any subset of data may be communicated to any of the other programs in the community. The *language* for controlling the execution of computer programs *and* the flow of engineering information is contained in the ODINEX executive computer program.

3.2 ODINEX Executive

The ODINEX functions are shown schematically in Figure 3. It preprocesses the program control directives resulting in the creation or updating of a control file. This file establishes the network of computer program executions which will perform the intended design activity. As the computer processes the control file, ODINEX is repeatedly executed after each design program. At each successive execution ODINEX extracts selected information from the design program output and installs or updates the information in the data base. Additionally, ODINEX interrogates the input stream of the succeeding design program and merges data base information with it. In practice, the ODINEX program is essentially transparent



ORIGINAL PAGE IS OF POOR QUALITY

Figure 3. ODINEX Executive Functions

to the user, appearing only as an *input language* which augments the input of existing design programs.

The design activities can include the performance of design integration function, creation of reduced size modules and special purpose programs, and data handling functions. Provisions have been made for stylized report generation including graphical data. The use of interactive graphics can be tied to any or all the design activities at the user's option.

The objective in the development of ODINEX was to provide a technique which would permit the simulation of the design process in an open-ended manner. Modules of varying complexity from all disciplines can be selected to suit the level of detail involved. Free flow of information from one module to the other is provided in a hands-off manner; yet manual override capability is provided at appropriate points.

The objectives of earlier developments are not entirely dissimilar, but the efforts to achieve these objectives have been directed toward a single computer program containing all necessary modules and subroutines with a main executive program. The result is a software system which strains the core capacity of even the largest computers. They have relied heavily on the overlay feature available on most machines to solve the *core limit problem*. The overlay technique involves splitting the analysis into several separately executed *links* with a block of core reserved in the root for *resident* data required by two or more of the subordinate links. The resident data contains the engineering information referenced by location. As the complexity of the program increases, the resident data requirements limit the size of the links; the number of links increases until a point is reached where the overhead expense (peripheral processing cost) of overlay seriously detracts from the usefulness of the program. The resultant program is closed-ended; deletion and replacement of functional modules is difficult since there is an inevitable waste of core due to uncertainty of previous allocation of the space.

The ODIN concept essentially replaces the overlay structured program with a sequence of independently executed programs which perform the same functions. Instead of drawing on and replacing the resident data in the root link, each program simply reads and writes data in the normal manner; i.e., card, tape, or disk. ODINEX is designed essentially to perform the *root link function* in a separate program execution. It extracts information *and* merges data base information with the input stream of other programs. Since the data base is dynamically constructed, it need only contain that information of interest to the present simulation. The result is a concept that allows an indefinite number of program modules representing the design activity yet the core requirement is no larger than the largest module in the simulation.

The success of ODINEX is largely attributable to two ancillary developments, RANDAC and CCLINK. RANDAC is a Rapid and Accurate Name-Oriented Directory Access Code, described in Reference 8, which forms the basis for the data base construction and intercommunication capability. Appendix A provides a description of the data base construction and access techniques. CCLINK is a Control Card LINKage program which provides the multiple program execution capability on the CDC 6000 series computers. This program is described in Appendix B. ODINEX combines the capabilities of RANDAC and CCLINK to form a unified approach to vehicle design synthesis.

3.3 Control Card Data Base

ODINEX contains FORTRAN-like branching logic for controlling the flow of computer programs through the machine. It performs this function in the environment of a *control card data base*, assembling a sequence of machine instructions based on simple user commands. Each data base entry is a subset of machine instructions for performing some task such as execution of a computer program, saving some data or compiling some source code. The branching capability permits conditional transfer to alternate sequences of program executions. Sizing and optimization loops are easily constructed using the ODINEX language.

3.4 Design Data Base

ODINEX dynamically maintains a data base of engineering information from user supplied information as well as information from the community of ODIN programs. It is executed at the beginning of the sequence of design programs to initialize the data base then is executed repeatedly after each design program, as shown in Figure 4. At each successive execution ODINEX extracts selected information from the design program output and installs or updates the information in the data base. Additionally, ODINEX interrogates the input stream of the succeeding design program and merges data base information with it. In practice, the ODINEX program is essentially transparent to the user, appearing only as an *input language* which augments the input of existing design programs.

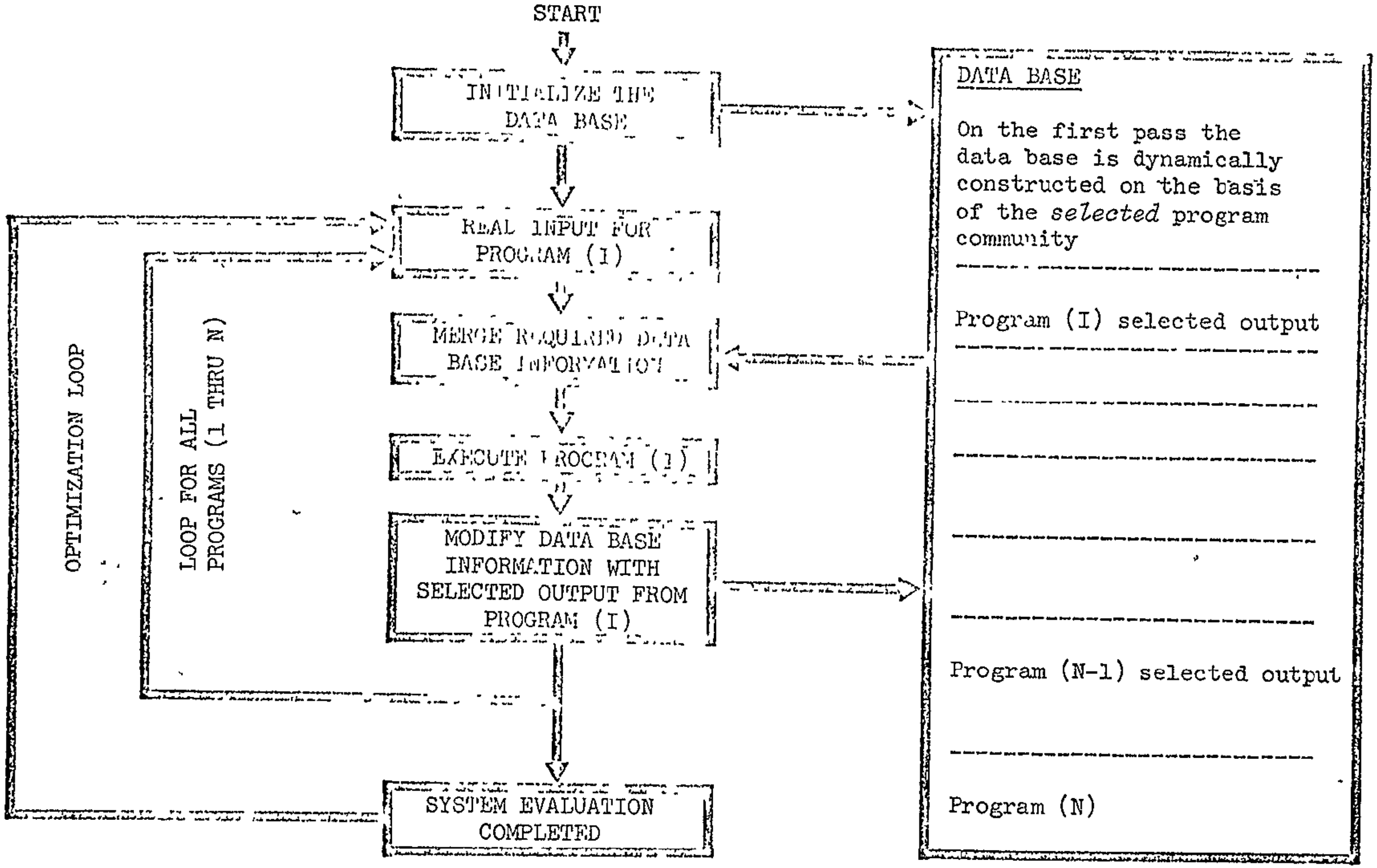
3.5 Information Storage and Retrieval

ODINEX contains an advanced computer code for storing and retrieving information *by name*. The primary objective of this method, called indirect access, is to reduce computer time required to locate information. In a directory of n items, the commonly used method of linear probing techniques requires an average of n/2 probes to locate a given item of information. The average number of probes required to locate an item in the directory is independent of directory size. This number is usually less than two and typically approaches one.

3.6 Units Conversion and Scaling

The ODINEX language contains a FORTRAN-like scaling capability for the convenience of the user. Any variable residing in the data base may be altered by any combination of arithmetic operations before being passed to the user program. Alternately, the *scaled* variable may be restored in the data base. The operations may be addition, subtraction, multiplication, division, or exponentiation. The operands may be constants or data base variables. In special cases, entire arrays may be scaled by a single statement.

Figure 4. ODIN/ODINEX FUNCTIONS



3.7 Report Generation

The ODINEX program contains a *built-in* report generation capability which permits the user to format stylized reports based on data base information. Reports may be requested at any point in the simulation and may be merged with independently generated graphical output data.

4. THE ODINEX LANGUAGE

The ODINEX language consists of *communication commands* and *control directives* as shown schematically in Figure 5. It is designed to augment the normal input stream of the ODIN programs. As such, all information intended for interpretation by ODINEX must be delimited. Communication commands are generally imbedded in the ODIN program data. They control the flow of information from the *design data base* to the ODIN program modules. As such, they form the common information link among the ODIN community of programs. It will be shown that a special output file from each ODIN program is itself a communication command to ODINEX. It passes information from the ODIN program to the design data base. The special output file *does not affect the normal operation* of the ODIN program.

The control directives define the flow of ODIN program modules through the computer. The control directives pertaining to the individual programs generally precede the data for the program. A set of eight control directives have been coded which provide user oriented control and minimize the amount of usual control information required to execute an ODIN simulation. In the ODIN community of programs, the user at Langley Research Center will need the following control cards to launch a simulation:

```
JOB,------
USER.-----
FETCH(A3682,SPR___,BOTH,ODINRLV,CCDATA)
CCLINK(ODINRLV)
789
```

The execution of these control cards results in fetching and linking to the ODIN system. All other ODIN module control will be defined by control directives,and all intercommunication will be defined by communication commands.

ORIGINAL PAGE IS
OF POOR QUALITY

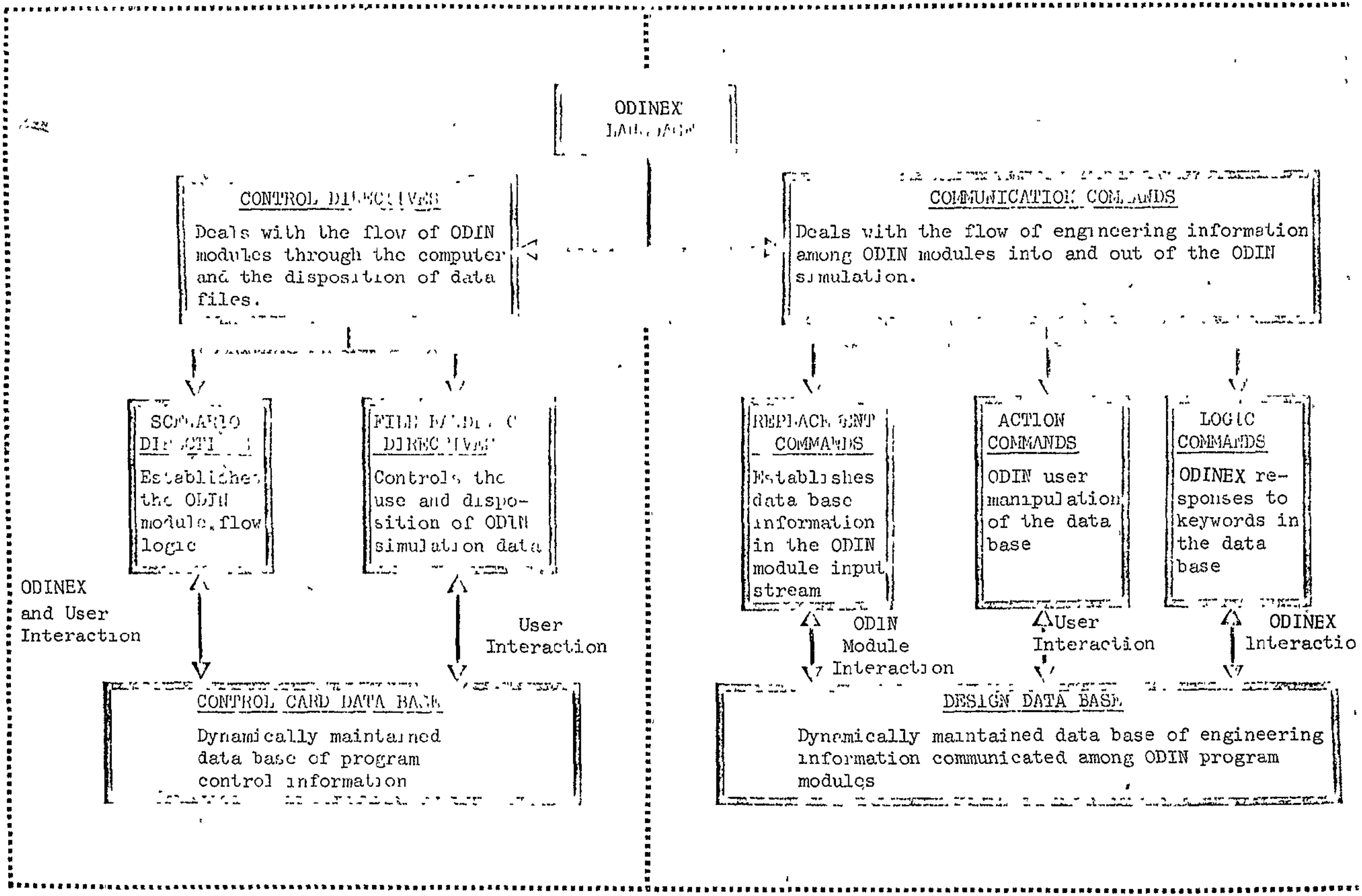


Figure 5. The ODINEX Schematic

4.1 Control Directives

The *control directives* are a set of simple commands summarized in Figure 6. They are used to establish the network of programs which will be executed in the design simulation. They are placed sequentially prior to the data (if any) associated with the directive. *Associated data is always followed by an end of record.* Each record may be augmented by other directives but must *precede* the directive which has data associated with it. Figure 7 is an example of a set of control directives representing the simulation of a space shuttle preliminary design cost optimization. In the example, a fresh copy of DBASE is specified, and control cards are temporarily updated. The input format follows the communication rules defined in Section 4.2.

The next command initializes the AESOP data base required outside the AESOP *loop*. DESIGN POINT1 identifies the beginning of the AESOP loop. The next four sets of data are straightforward EXECUTE instructions followed by the appropriate data files. The LOOP directive specifies a branch to POINT1, specified earlier in the sequence.

The IF directive is a condition on the LOOP directive; JJJ must be a data base name. EXECUTE REPORT is a *special* ODINEX command which permits the user to stylize a report in his own format. This report may include information as required from the data base. This is discussed in detail in Section 4.3. The 'END--- statement must be the last instruction in the simulation.

In summary, the user defines a sequence of EXECUTE directives for the ODIN programs by name. Following each EXECUTE directive is the data for the program to be executed. Design loops may be defined by the DESIGN and LOOP directives. The DESIGN directive defines a unique point in the sequence where control may be returned. The LOOP directive may be *conditional* upon satisfying any number of IF directives.

The INITAL directive provides a fresh copy of the named file, for example,

'INITAL DBASE'

* 'EXECUTE name'	A scenario directive for executing a program or function by name. Any name for which a prestored set of control cards has been defined is legal.
'INITAL name'	File handling directive for initializing files; the three acceptable names are DBASE⁻ - design data base CCDATA*- control card data base LUSOP - AESOP data base
* 'UPDATE name'	Same as INITAL except that LUSOP is not acceptable.
'DESIGN name'	Scenario directive defining a point in the execution sequence (like a FORTRAN statement label); name is any data base name.
'LOOP TO name'	Branching instruction referring to a design name. It can be conditional or unconditional.
'IF name.OP.name' †	Condition for branching. Any number of conditions may be specified on separate cards *after* a LOOP directive. If more than one condition is specified, .OR. is implied. That is, any of the conditions satisfied will trigger the branch instruction.
'RESTRT'	Means use the existing data base. It must have previously been defined and stored.

--- OP is a conditional operator (LT,LE,EQ,GE,GT)

- Data is expected; *end of record* (789) is required.

Figure 6. Control Directives

```
'INITAL DBASE'
     Data Base Initialization Data

789 . . . . . . . . . . . . . . . . . . . . . . End of Record

'UPDATE CCARDS'
     Updates to Control Card Data Base
789 . . . . . . . . . . . . . . . . . . . . . . End of Record

'INITAL AESOP'

'DESIGN POINT1'

'EXECUTE SSSP'
    {Data for SSSP
789 . . . . . . . . . . . . . . . . . . . . . . End of Record

'EXECUTE DAPCA'
    {Data for DAPCA
789 . . . . . . . . . . . . . . . . . . . . . . End of Record

'EXECUTE AESOP'
    {Data for AESOP
789 . . . . . . . . . . . . . . . . . . . . . . End of Record

'LOOP TO POINT1'

'IF JJJ.LT.15'

'EXECUTE REPORT'
    {Data for REPORT
789 . . . . . . . . . . . . . . . . . . . . . . End of Record

'END ODIN'

6789 . . . . . . . . . . . . . . . . . . . . . . End of Information
```

Figure 7. ODIN/RLV Control Directives

blanks all information in the design data base and prepares to accept newly defined entries with the information which follows. The UPDATE directory loads the current file and prepares to accept updates.

The END directive signifies the end of the input and the end of the simulation. This directive (*must be present*) is used to terminate reading of input and also to normally terminate execution of the simulation.

4.2 Communication Commands

The *communication commands* provide a means of dynamically maintaining a data base of engineering information pertaining to the design being simulated. They control all exchange of information within the program community. These commands can be freely interspersed within the ODIN program data. They are generally used for identifying, adding, modifying, scaling, and printing data base information, retrieving and replacing scaled information from the design data base for use by the ODIN programs in the simulation, and identifying and printing ODIN program module input data.

The communication commands consist of three types: the *replacement command* in which data base names placed on an input card are replaced with data base information, the *action commands* which cause the alteration or manipulation of data base variables, and the *logic commands* which alter the flow of information in ODINEX.

4.2.1 Replacement command. Among the ODINEX functions is a passive replacement command which augments the normal input of any ODIN program module. The input stream of the individual programs are read by ODINEX, and the communication commands are interpreted. Based on the instructions encountered, ODINEX builds a modified input stream which is acceptable to the ODIN program module and contains the desired data base information. The ODIN module is then executed in the normal batch mode with its normal input format and is totally unaware that it forms an element of a design simulation.

4.2.1.1 Simple replacement. The replacement commands consist of simple element replacement and array transfer. For element replacement, the user places the name of the desired data base variable on the input data card enclosed with ODINEX delimiters; the delimiters define the field to be used. For example, suppose the normal card input were a series of six numbers in fields of ten columns each

```
1.        2.        3.        4.        5.        6.
```

If the user desired that the fifth number be taken from the data base and if the data base name were BETA, the card would read

```
1.        2.        3.        4.        'BETA  '  6.
```

ODINEX would isolate the word BETA by identifying the delimiters, '; locate the name BETA in the data base directory; retrieve the value associated with BETA, and replace the name and delimiters by the most significant part of the data base value *within the field defined by the delimiters*. There must be *no imbedded blanks* between the first delimiter and the data base name. In the case of an integer, the number would be *right adjusted* in that field. Each time the program is executed, the user program would receive the current data base value of BETA. Logical variables, .TRUE. and .FALSE., and hollerith information are also permitted. Further, an element of an array such as A(3) may also be used in the replacement function.

4.2.1.2 Replacement with scaling. In addition to simple replacement, DIALOG has the capability of scaling data base values before replacement. Suppose the previous example required that BETA be scaled by a factor of two, then the input card would read

```
1.        2.        3.        4.        'BETA*2.' 6.
```

The scaling law in ODINEX conforms to simple arithmetic operations such as addition, subtraction, multiplication, division, and exponentiation *performed in a sequential manner. Mixed mode arithmetic is not allowed.*

A thorough knowledge of the arithmetic structure of data base information is expected of the user. The factors used for scaling can be constants or other data base variable names, i.e.,

'BETA+GAMMA'

and up to ten operations can be performed before replacement. The rules for using communication commands are set forth in Figure 8. Each operation applies to the result of all previous operations. In the expression

'A+B*C+D'

B is added to A, the result multiplied by C, and that result added to D.

4.2.1.3 Array replacement. In addition to simple replacement, ODINEX has the capability of transferring entire arrays from the data base to the user's input stream. This command is limited to namelist or other suitable free field input packages. Its usage is identical with simple replacement. The user places the data base array name in the input stream enclosed by ODINEX delimiters. ODINEX identifies the array and places all values from the data base into the input stream adding card records as required to perform this function.

The scaling rules of Figure 8 apply to array transfer as well as simple replacement. In the expression A*B where A is an array and where B is a scalar, each element of A is multiplied by the scalar B before replacement. If A and B were arrays, the arrays would be multiplied together element by element until the array with the least elements were exhausted; i.e., the resultant array is limited to the length of the smallest array in the expression.

4.2.2 <u>Action Commands</u>. Action commands consist of five types ADD, DELETE, DEFINE, INITAL, and . (comment). They permit the addition, deletion, definition, initialization of data base information, and comments pertaining to user data; they are itemized in Figure 9. In general, these commands are placed in the data stream with suitable delimiters. They are interpreted and executed by ODINEX, but *not seen* by the ODIN modules. All five commands

1. Subscripts are enclosed in parentheses.

2. Operators are applied to everything which appears before the operator; i.e., A = B+C*D means (A+B)*D.

3. First word beyond equal (=) must be a name.

4. Card must end with a comma or data base delimiter; i.e., new card must start a new dialog.

5. Operations cannot exceed 20 characters not counting blank.

6. In simple replacement, first word beyond delimiter must be a data base name; '-A' is not allowed.

7. No imbedded blanks are allowed between first delimiter and the data base name.

8. ADD command allows initialization of integer or real scalars to any value or any expression as long as modes of variables in the expression are not mixed; i.e., A=4*.5 will yield 4 values equal to .5; A = 4.*.5 will yield A equal to 2.; A = 4*.5 will yield 4 values equal to 0.

9. ADD command for array initialization accepts only two forms:

   1. A = 4 * 5.
   2. A = .5, .5, .5, .5

   yields 4 values of 5

10. Mixed mode arithmetic is not permitted in scaling operations. Results from breaking this rule are unpredictable.

Figure 8. Rules for Using Communication Commands

USER INTERACTION WITH THE DATA BASE

Command	Description
'ADD A=B,...'	Used to create a new data base entry or alter the information associated with an existing data base entry A is a new or existing data base entry, scalar or vector. B is the update information which can be real, integer or logical constants, variables, or scaled combinations of scalar or vector elements. Multiple commands can be executed.
'DELETE A,B,...'	Used to delete entries in the data base. Multiple deletions can be executed.
'DEFINE A=n, description'	Used to define new or existing entries in the data base A is the new or existing data base entry n is the desired number of data base locations; it is ignored if the entry exists; the default is 1 of omitted. description is the hollerith information associated with the variable A.
'INITAL A=B'	Used to initialize the data base. The rules are the same as for the ADD command.
'. comment'	Used to identify ODIN program data. ODINEX replaces the comment and delimiters with blanks in the ODIN program data deck.

Figure 9. Action Commands

permit the same general format; a single command opens the way to any number of actions of the same type separated by commas.

'COMMAND action,action,action'

4.2.2.1 ADD command. The ADD command has the following format:

'ADD A=B'

where A is a new or existing data base variable. B can be real, integer, logical hollerith, or another data base name. B can also be a combination of data base names and numbers such as C*D or C*12. or I*2+J. However, combinations cannot start with a number since A=12*B would imply that A is an array of 12 values all equal to B. This array initialization (A=n*X) is not considered mixed mode arithmetic and is interpreted correctly. Similar acceptable forms of the ADD command are

1. 'ADD A=B,C,D,E' — A is an array of four elements whose values are the values of the data base variables B,C,D,E, if they exist. *If they do not exist in the data base, B,C,D, or E are stored as elements of A.*

2. 'ADD A=B, C=D' — A is a scalar or the first element of an existing array whose value will be the value of the data base B. C is a scalar or the first element of an array whose value will be the value of the data base variable D. In general, more than one variable may be added or modified with a single command. Continuation cards may be used.

3. 'ADD A=B*C, D*E' — A is a two-element array whose elements are the combination, B*C and the combination D*E. In general, all the scaling rules of Figure 8 apply to each element of an array.

4. 'ADD A=12*5.' — A is an array of twelve elements which will be loaded with 5.0. In general, this is the only *combination* which permits a leading number. All other combinations must have leading names. For example, A = -B or A = -1*B would not be acceptable. The user would have to use A = B*-1.

4.2.2.2 The DELETE command. The DELETE command has the following format:

'DELETE A'

A is the name of a data base variable. This command deletes the name from the data base directory but does not delete the space allocated in the data base. Multiple variables may be deleted as follows:

'DELETE A,B,C,'

until a second delimiter is encountered.

4.2.2.3 The DEFINE command. This command is used for defining data base variables. The definitions are stored in the data base directory and are recalled when the data base information is recalled. The format of the DEFINE command is

'DEFINE A=n, FIRST LETTER'

where A is the name of a new or existing variable, and n is the number of elements if A is a new array. It is ignored if A is an existing data base entry. If omitted, n defaults to 1. The information following the comma is a string of hollerith information defining the data base variable A.

4.2.2.4 Special ADD-like commands. The INIT command is used to initialize the data base with information needed by the ODIN simulation

C-9

but not already existing in the data base. The format and rules are all the same as for the ADD command

'INIT A=B'

In reality the word INIT is optional; any word up to ten characters may be used. All commands are stored in the data base directory as the first word of the description. This scheme is used to identify the origin of every piece of information in the data base. The output from every ODIN module is a special namelist file. ODINEX treats the namelist name as an ADD-like command based on the value of BUILD (see Section 4.2.3). The information added from this file is assigned the namelist name as part of the description. The namelist name can and should be descriptive of the originating module. In this manner, the user can easily identify the origin of a particular piece of data base information. Figure 10 is a *snapshot* of the data base during a sample simulation identifying the names, locations, lengths, values, origins, and descriptions of all entries in the data base.

4.2.2.5 The . (comment) command. This command is used to identify user data. It represents information which will not be *seen* by the ODIN program nor passed to the data base. ODINEX interprets it as a meaningless data and simply replaces the information with blanks. The format for the command is

'. THIS IS A COMMENT'

The comment is enclosed with data base delimiters. The (.) signifies the information which follows (including the command and delimiters) is to be ignored, that is, replaced by blanks. The comment may be used on a data card or on a separate card. If a separate card is used, the entire card is ignored.

4.2.3 <u>Logic Commands</u>. Logic commands control the flow of information within the ODINEX program. They are in the form of data base entries or keywords which are interrogated at each pass through ODINEX. Depending on

* * * * * THERE ARE CURRENTLY 19 DATA BASE ENTRIES AS FOLLOWS * * * * * * * * * * * * *

NAME	LOCATION	DIMENSION	CURRENT VALUE(S)	ORIGIN	DESCRIPTION
AESOP	39	1		DEFINE	AESOP USED IN SIMULATION
ALPHA	7	2	0.525E+06 0.2715E+01	AESOUT	AESOP CONTROL PARAMETERS
BUILD	19	1	0	DEFINE	DYNAMIC DB BUILD OPTION
CRDSKP	1	1	32	ADD	CARD SKIP OPTION
DBDUMP	31	1		DEFINE	CAUSES DATA BASE DUMP
DISPLAY	13	1	WPAYLO	INITAL	SCOPE DISPLAY FUNCTION
FLTIME	23	1	28.0920000	DEFINE	TOTAL ELAPSED TIME
JJBEST	25	1	1	AESOUT	BEST PERFORMANCE EVALUATION
JJJ	17	1	0	AESOUT	AESOP EVALUATION COUNTER
MIXR	3	1	2.74999999999989200	ADD	PROPELLANT MIXTURE RATIO
RUNID	21	1	*ODIN JAN 22, 72	DEFINE	IDENTIFICATION FOR DIALOG RUN
TCOSTB	35	1	48673.85593780665658	ADD	TOTAL COST OF BOOSTER
TCOSTN	37	1	0.19037115923077E+05	DAPOUT	TOTAL COST OF ORBITER
THRUST	5	1	525000.000000000000	ADD	ENGINE VACUUM THRUST
VSTGB	29	1	0.91501786518072E+04	WTOUT	BOOSTER STAGING VELOCITY
VSTGO	33	1	0.25853097798483E+05	WTOUT	ORBITER STAGING VELOCITY
WGROSB	11	1	0.26662821532874E+07	WTOUT	GROSS WEIGHT OF BOOSTER
WGROSO	15	1	0.71802039902725E+06	WTOUT	GROSS WEIGHT OF ORBITER
WPAYLO	27	1	0.31847761977751E+05	WTOUT	PAYLOAD OF THE LAUNCH SYSTEM

* *

Data base entry name

Data base location and dimension of information

Current data base information

Description entered with a 'DEFINE' command

The command which last updated the information the information in the data base. ODIN output namelist names are treated as commands.

FIGURE 10. OUTPUT FROM LOGICAL COMMAND 'DBDUMP'

its presence and/or its value, ODINEX performs some special function. The following keywords currently have significance in the DIALOG program.

1. BUILD=n — Keyword providing disposition of previously undefined names in the special ODIN module output files.
   n = 0, Ignore undefined names
   n = 1, Add undefined names and values to data base

2. CRDSKP=n — Controls the number of cards or records to be skipped before looking for more data base information. This keyword facilitates a reduction in processing time for the ODIN module input data; always reset to zero.

3. DBDUMP — Keyword providing for the printing of all names, values, origin, and definitions in the data base. Figure A3 is an example of this report.

4. ELTIME — Prints the elapsed time after each ODIN module.

5. INDUMP — Keyword providing for printing the modified input for the next ODIN module, just as the ODIN module will see it.

6. OUTDMP — Keyword providing for printing the special output file from ODIN program modules. It contains the candidate information for the ODIN data base.

7. RUNID — Hollerith identification given to the simulation.

4.3 Report Generation and Graphics

One of the primary functions of a good design simulation is the communication of input and output information to the design staff. Each member is interested in a specific subset of information and generally does not want to be burdened with unneeded data. Providing just the right amount of data would seem an extremely difficult task for a simulation with the flexibility of the ODIN system. However, the unique feature of the ODINEX executive which permits the user to manipulate design programs at the input/output level also provides the basic capability required for automatic report generation.

4.3.1 Stylized reports. During the initial phase of coordinating the simulation requirements the design specialists staff selects subsets of the data base information to be communicated to each staff member for analysis. The format of the individual reports are tailored to the needs of the individuals receiving the information. Once the format is established, it is keypunched on data cards with data base information being identified by name in the manner described in Section 4.2.2. These data cards become a report file which can be fed to the ODINEX executive at any point in the simulation. The report is generated by the control directive 'EXECUTE REPORT'. ODINEX interrogates the report file for data base names and replaces the name and delimiters with the appropriate data base information. The file is then printed resulting in a summary report on the current status of the design. Later modifications to the format are as simple as changing a data card.

A mini-report exemplifying this technique is shown in Figure 11. Many of the features of the ODINEX language including scaling and adding data base information are being used in a completely free field report format. The first column of each card is reserved for printer carriage control providing a convenient means of paging and spacing for report clarity. Figure 11 also shows the printed results of the report file as augmented by data base information.

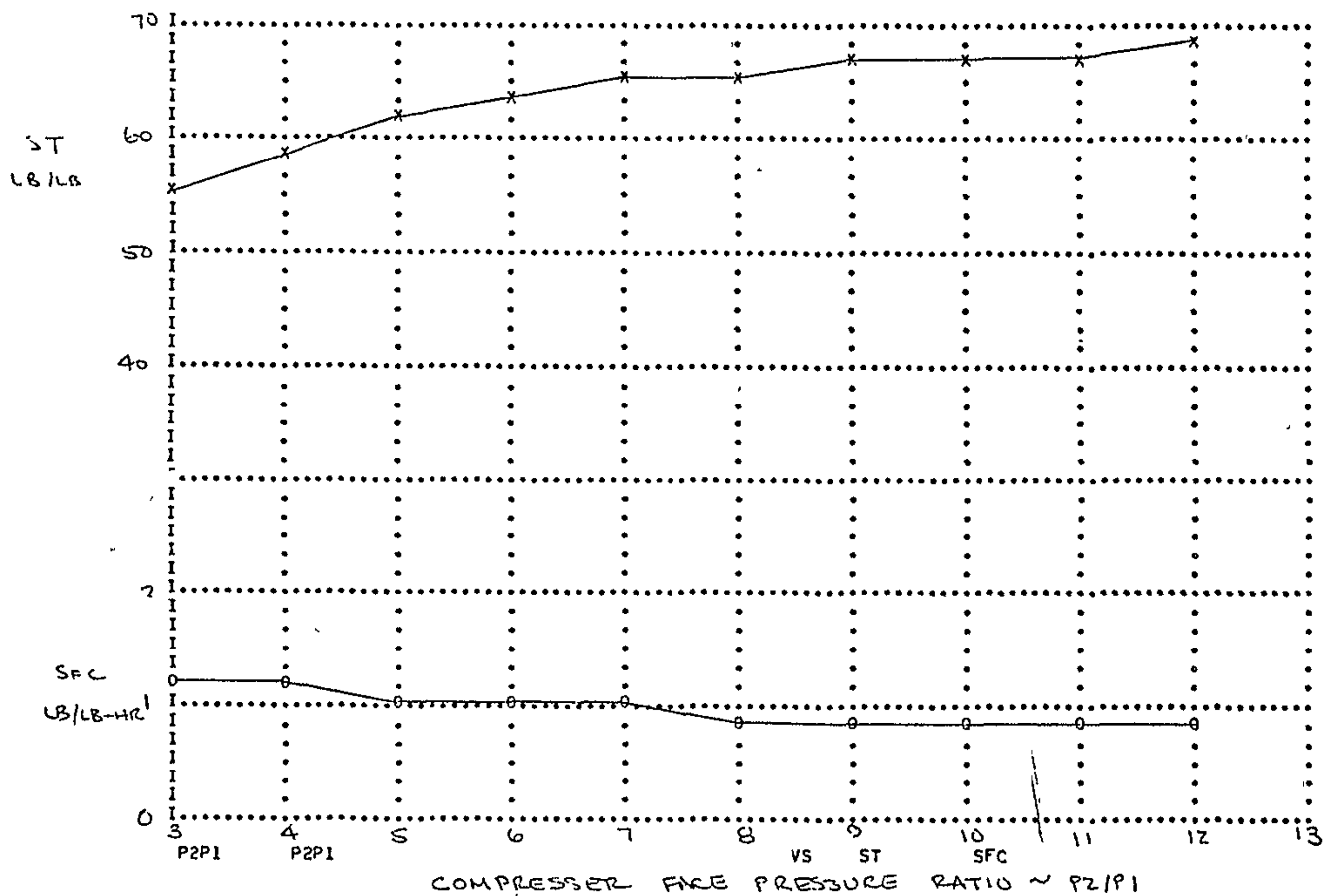
4.3.2 Graphics. ODINEX contains no graphics within the program. Instead, two independent plot programs, References 9 and 10, have been provided which can plot input information, information from the data base, or information from special binary files. These programs have several plot device options. There is a *quick look* printer plot which provides low resolution plotted information from the on-line printer. Figure 12 exemplifies the quality of information from this option.

```
                                                   ≠EXECUTE REPORT ≠
                                                   ≠. DATA FOR SUMMARY REPORT ≠
1                                                                         PAGE 1
0
                              SUMMARY REPORT FOR ODIN/RLV
(≠JJJ≠ CYCLE(S)                                         ELAPSED TIME = ≠ELTIME≠ CPU SECONDS
  ≠ADD A=TCOSTN+TCOSTB≠
  ≠ADD KN=1.689≠
  ≠ADD K=0.45359≠
0MAXIMUM PAYLOAD           ..... ≠WPAYLO  ≠ LBS        ≠WPAYLO*K≠ KILOGRAMS
        BOOSTER WEIGHT ..... ≠WGROSB  ≠ LBS        ≠WGROSB*K≠ KILOGRAMS
        ORBITER WEIGHT ..... ≠WGROSO  ≠ LBS        ≠WGROSO*K≠ KILOGRAMS
0TOTAL COST                .....$≠A       ≠ MILLIONS
        BOOSTER COST    .....$≠TCOSTB  ≠ MILLIONS
        ORBITER COST    .....$≠TCOSTN  ≠ MILLIONS
0STAGING VELOCITY          ..... ≠VSTGB   ≠ FPS        ≠VSTGB/KN≠ KNOTS
0ENGINE VACUUM THRUST      ..... ≠THRUST  ≠ LBS
0BOOSTER MASS RATIO        ..... ≠ALPHA(2)≠
 ≠ADD N=JJBEST≠
0BEST CYCLE=≠JJBEST≠, THE ≠N*2-1≠TH RECORD ON FILE OUTSAV. N=≠N≠
0
                                    END OF ODIN/RLV SUMMARY
+                                                        REPORT
1
C0000000000000000000
```

```
                                                                         PAGE 1


                              SUMMARY REPORT FOR ODIN/RLV

    1 CYCLE(S)                                        ELAPSED TIME = 58.26500 CPU SECONDS

MAXIMUM PAYLOAD           ..... 31845.6096 LBS        14444.8500 KILOGRAMS
        BOOSTER WEIGHT ..... 2671117.97 LBS        1211592.40 KILOGRAMS
        ORBITER WEIGHT ..... 713186.621 LBS        323494.319 KILOGRAMS

TOTAL COST                .....$67809.5935 MILLIONS
        BOOSTER COST    .....$48835.2395 MILLIONS
        ORBITER COST    .....$18974.3540 MILLIONS

STAGING VELOCITY          ..... 9202.29158 FPS        5448.36683 KNOTS

ENGINE VACUUM THRUST      ..... 525000.000 LBS

BOOSTER MASS RATIO        ..... 2.72500000

BEST CYCLE=       1, THE       1TH RECORD ON FILE OUTSAV. N=  1

                                    END OF ODIN/RLV SUMMARY REPORT
```

Figure 11. Example Input and Output for ~~Report~~

DRY TURBOFAN ENGINE B=1. T3=3000



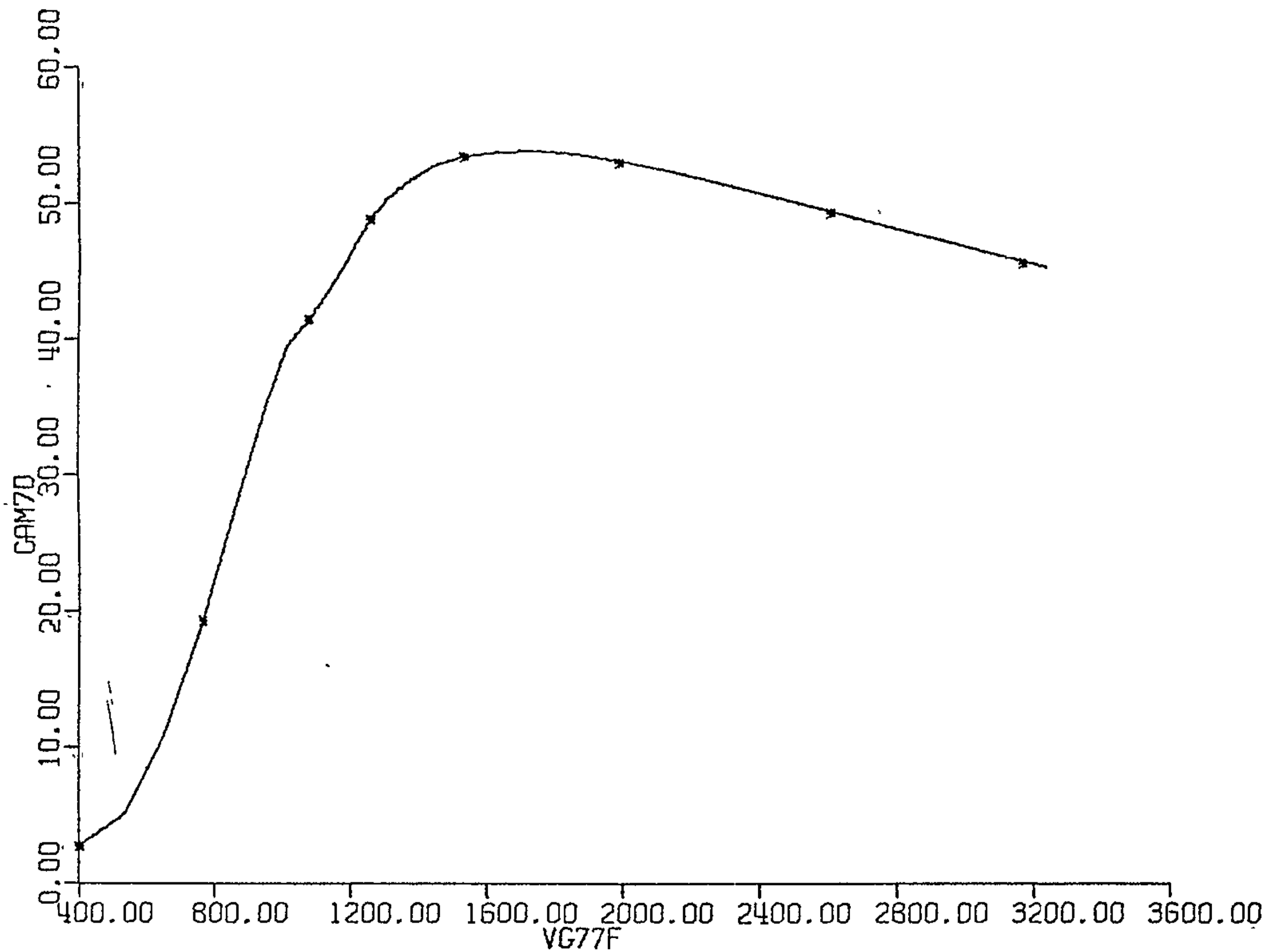
X SCALE = 1.000E+00 UNITS / INCH MIN. X = 3.000E+00 MAX. X = 1.300E+01
Y SCALE = 1.000E+01 UNITS / INCH MIN. Y = 0. MAX. Y = 7.000E+01
FRAME 4

Figure 12. Example Output from Printer Plot Option

A report quality CALCOMP plot may be generated and plotted off-line. This option permits scales and annotation on the graph. Figure 13 exemplifies this type of chart from the CALCOMP option.

The program may be used in an interactive mode on the CDC 250 display console. Plots which are generated and displayed can be scaled and regenerated to suit the user requirements. Hard copy may be obtained directly from the console. Figure 14 is an example output from the CDC 250 hard copy option.

Figure 13. Example Output from CALCOMP Plot Option



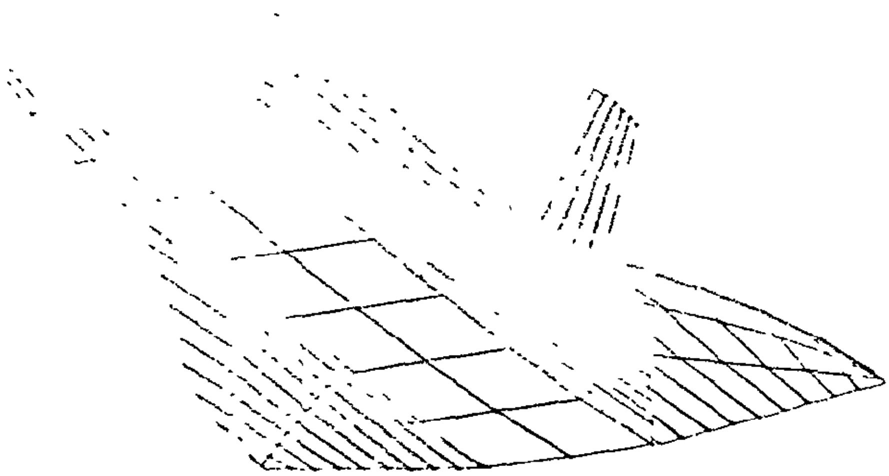


Figure 14. Example of CDC 250 Hard Copy Option

ORIGINAL PAGE IS
OF POOR QUALITY

5. ODINEX USAGE EXAMPLES

The ODIN simulation concept requires only five control cards to initiate regardless of the type of synthesis to be performed.

```
JOB-i----
USER----
FETCH(A3682,SPR---,BOTH,ODINRLV,CCDATA)
CCLINK(ODINRLV)
789                                end of record
```

All program control is handled through the ODINEX *control directives*. All data intercommunication is handled through ODINEX *communication commands*. These ODINEX functions are discussed in Section 4.

Five sample cases using the ODINEX executive are discussed below.

1. A sample optimization problem involving the use of SSSP, DAPCA, and AESOP. Two parameters, engine thrust and booster mass ratio, were selected as the performance criteria. Figure 15 shows the ODINEX control directives and flow diagram for this example. Results of this study are discussed in Appendix C.

2. An example of construction and storage of a control card data base using ODINEX is shown in Figure 16. This setup is equally applicable to construction and storage of a design data base. In the latter case initialization data would be included for 'INITAL DBASE.'

3. A coupling of the ENCYCL and PLOTTR program, Figure 17, demonstrates the ability to generate engine cycle analysis data and to plot the essential information with no special programming provisions. PLOTTR is equally useful for obtaining plotted information from any analysis program.

4. The use of FORTRAN language to augment the existing synthesis capabilities of the ODIN program community is accomplished by a user written FORTRAN program (it could be any language) which can be compiled and

```
'INITAL DBASE'
   (initialization data)
789
'DESIGN START'
'EXECUTE SSSP'
   (data for SSSP)
789

'EXECUTE DAPCA'
   (data for DAPCA/booster)
789

'EXECUTE DAPCA'
   (data for DAPCA/orbiter)
789

'LOOP TO FINISH'
'IF JJJ.GE.MAXJJJ'
'EXECUTE AESOP'
   (data for AESOP)
789

'LOOP TO START'
'DESIGN FINISH'
'EXECUTE REPORT'
   (data for REPORT)
789

'END ODINRLV'
6789
```

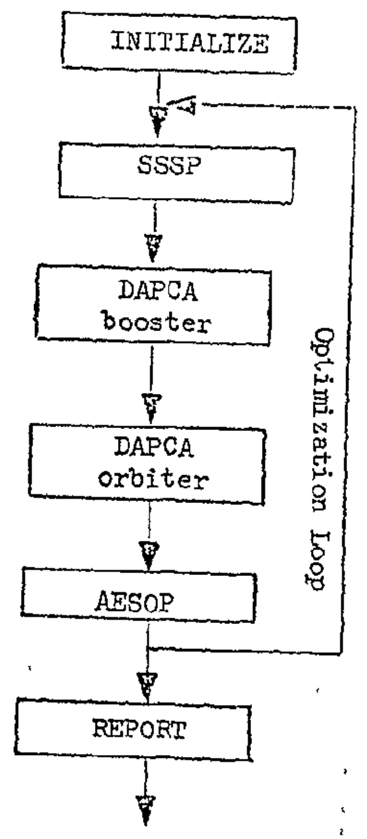


Figure 15. A Sample Optimization Using ODIN

```
'INITAL DBASE'
   (no data necessary)
789

'INITAL CCDATA'
   (control card data)
789

'EXECUTE CCSAVE'
   (data for REPLACE)
789

'END'
6789
```

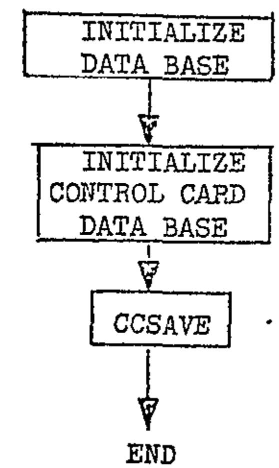


Figure 16. Example of Construction and Storage of Control Card Data Base

```
'INITAL DBASE'
   (define arrays to be plotted)
789

'EXECUTE ENCYCL'
   (data for engine cycle analysis)
789

'EXECUTE PLOTTR'
   (data for plot program)
789

'END'
6789
```

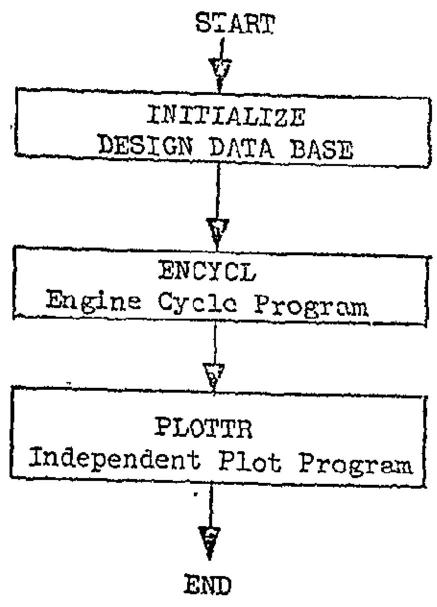


Figure 17. Graphical Presentation of Data from Any Analysis Program

executed during an ODIN simulation. Figure 18 shows the procedure in isolation. Note that the FORTRAN program itself can draw on data base information for array dimensions and data.

5. A coupling of the VAMP and HABACP computer programs to demonstrate the ability of ODINEX to handle geometry perturbations in a compatible manner for differing geometry input schemes. Figure 19 shows the geometry perturbations for VAMP. The exact same perturbations were made for HABACP although the panels are defined differently.

Both batch and on-line graphics capability are automatically available with ODIN due to the flexible coupling of independent programs through the ODINEX executive. The ODIN concept permits any analysis program which has on-line or batch graphics capability to be used in an ODIN simulation, for example, the HABACP program, the VAMP program, etc. Furthermore, two independent plot programs, IMAGE and PLOTTR, were developed specifically for ODIN batch and on-line graphics. The image program displays vehicle configuration 'pictures.' PLOTTR displays analysis-type plotted information. Each display may be manipulated with regard to location and magnification on the CRT. This is accomplished through a Langley Research Center-developed software system available with any CDC 250 graphics program.

Finally, the independent EDIT program developed at Langley permits the on-line editing of any file of information available to the current job. This is particularly useful to the ODIN concept since multiple programs are being executed. The input, output, and program control files can be edited in an in-line manner or under abnormal termination conditions.

Batch and on-line graphics application for the same problem is shown in Figure 20. Here, the basic analysis modules are augmented by the graphics capabilities of EDIT, PLOTTR, and IMAGE. The efficient use of the interactive mode depends on the use of the standard CDC utility, RFL. Typical core requirements are indicated in Figure 20. Although the maximum field length for the job is 140,000 octal, the majority of central processing

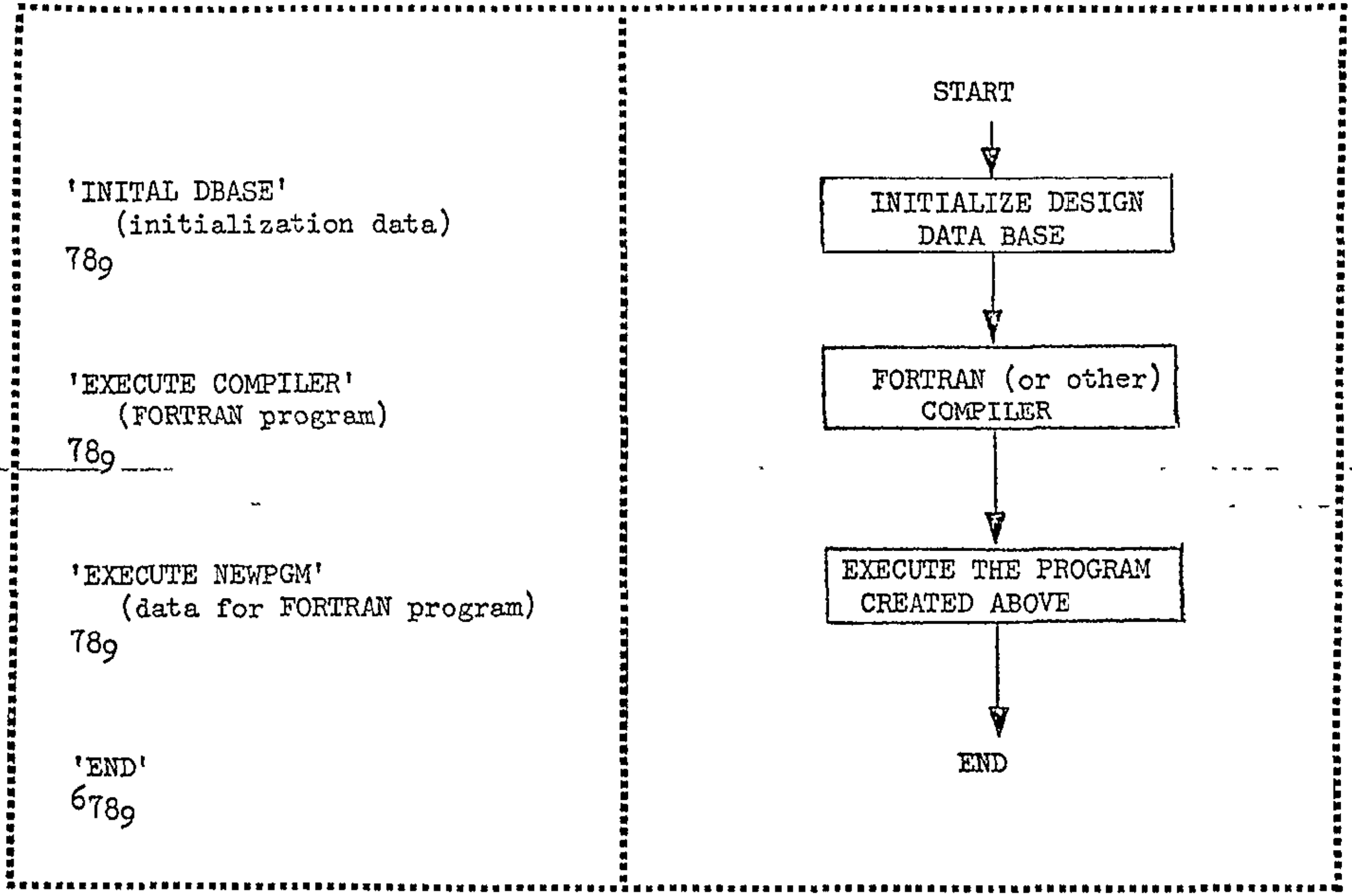


Figure 18. Use of FORTRAN Language to Augment Analysis

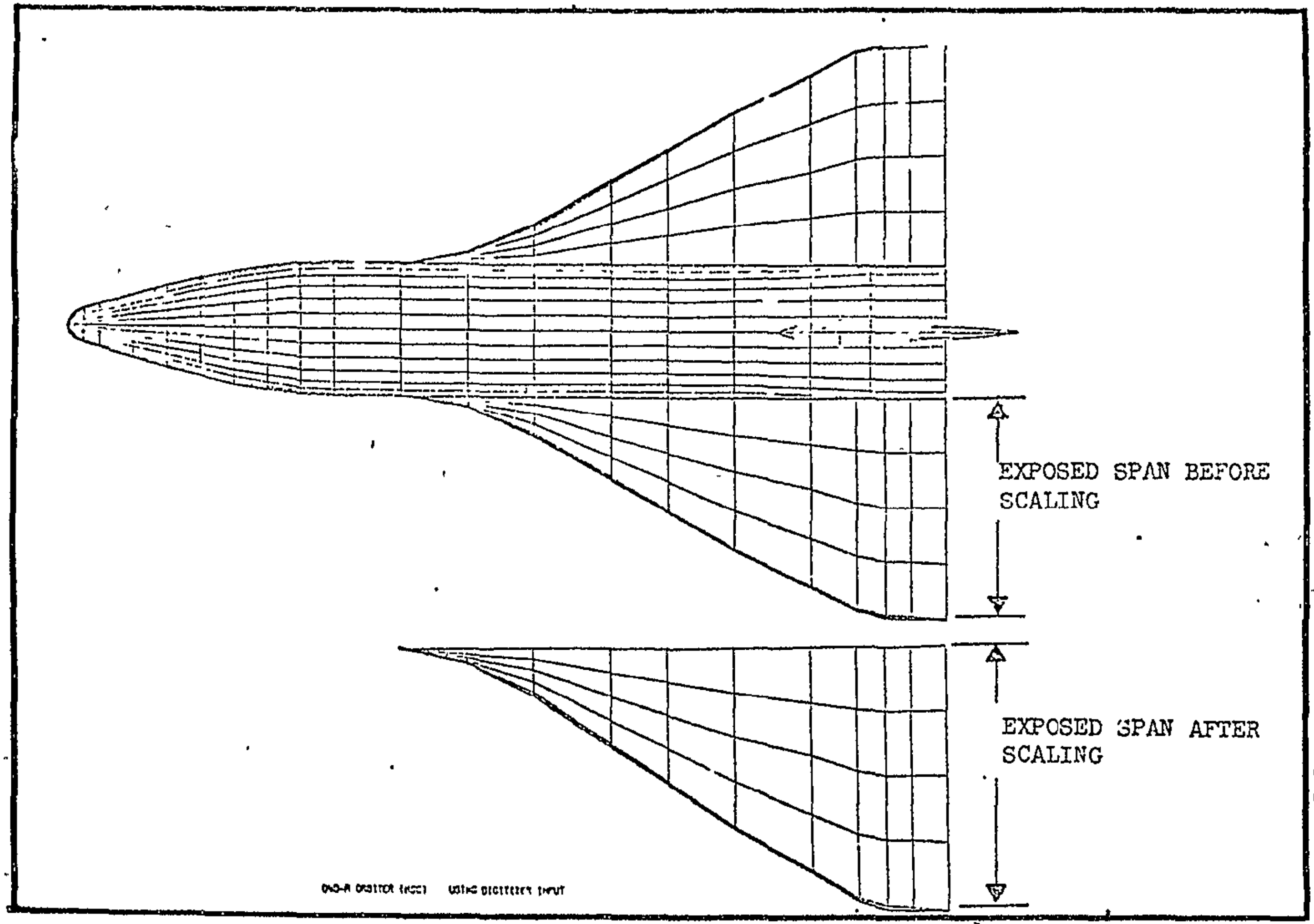


Figure 19. Geometry Perturbation for the 040A Orbiter

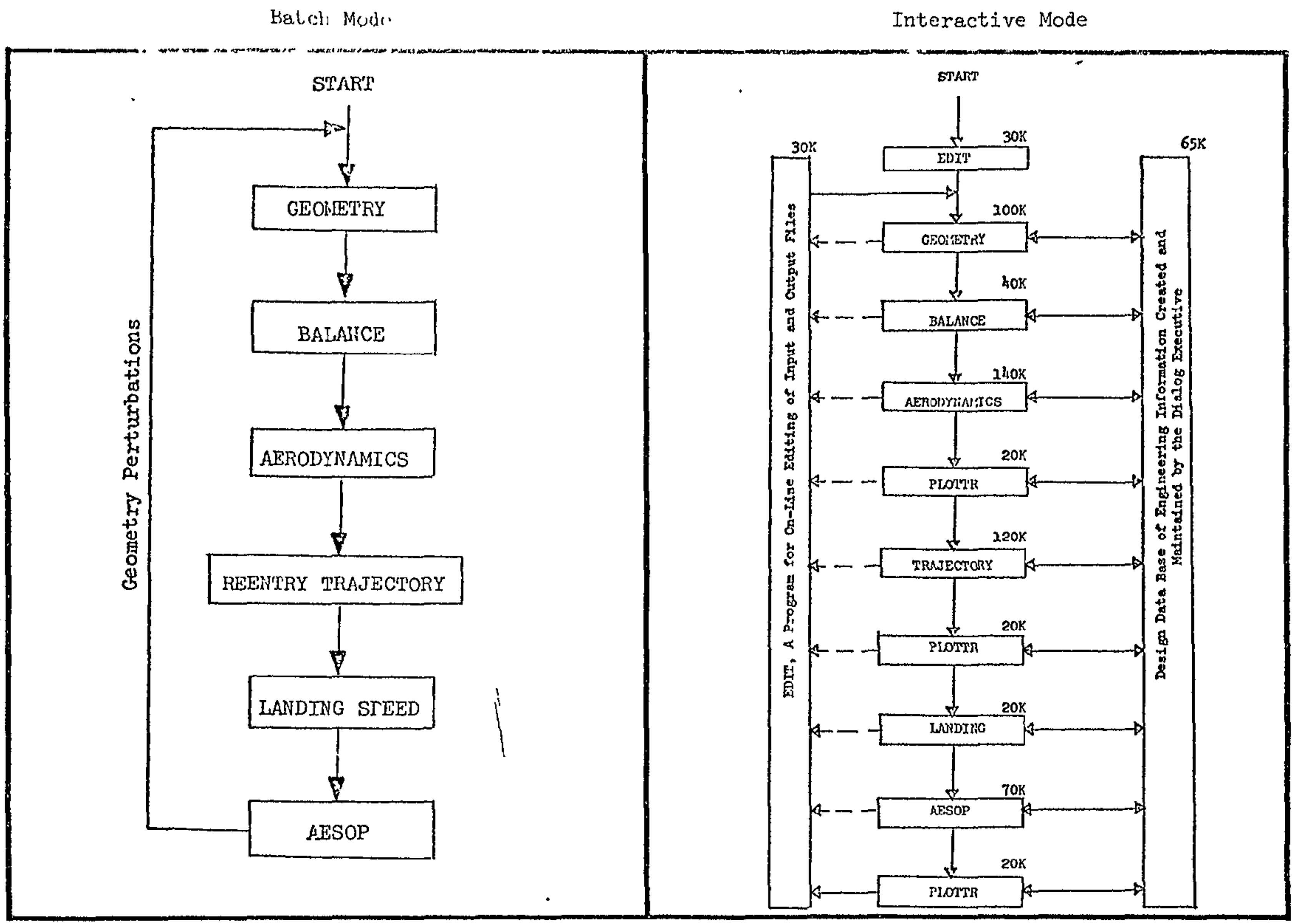


Figure 20 Comparison of Batch and Interactive Modes

time will be spent in the EDIT, PLOTTR, and IMAGE programs. These typically require less than 30,000. The RFL utility permits adjustment of field length to accomodate the program currently in core.

6. REFERENCES , APPENDIX I

1. Gregory, T.J., Petersen, R.H., and Wyss, J.A.: "Performance Trade-Offs and Research Problems for Hypersonic Transports." *Journal of Aircraft,* July-August 1965.

2. Hague, D.S. and Glatt, C.R.: An Introduction to Multivariable Search Techniques for Parameter Optimization. NASA CR-73200, April 1968.

3. Hague, D. S. and Glatt, C.R.: Application of Multivariable Search Techniques to the Optimal Design of a Hypersonic Cruise Vehicle. NASA CR-73202, April 1968.

4. Lee, V. A., Ball, H.G., Wadsworth, E.A., Moran, W.J. and McLeod, J.D.: "Computerized Aircraft Synthesis". *Journal of Aircraft,* Volume 4, Number 5, September-October 1967, Pages 402-408.

5. Herbst, W.B. and Ross, H.: Application of Computer-Aided Design Programs for the Management of Fighter Development Projects. Presented at the AIAA Fighter Aircraft Conference, St. Louis, Missouri, March 5-7,1970, AIAA Paper No. 70-364.

6. Wennagel, G.J., Mason, P.W., and Rosenbaum, J.D.: Ideas, Integrated Design and Analysis System. SAE Paper 68-0728, presented to SAE Aeronautics and Space Engineering Manufacturing Meeting, Los Angeles, California, October 1968.

7. Gold, R. and Ross, S.: Automated Mission Analysis Using a Parametric Sensitivity Executive Program. AAS Paper No. 68-146, presented at AAS/AIAA Astrodynamics Specialist Conference, Jackson, Wyoming, September 3-5, 1968.

8. Glatt, C. R. and Watson, D. A.: RANDAC: A Rapid and Accurate Name-Oriented Directory Access Code. Aerophysics Research Corporation, Technical Note TN-134, January 1972.

9. Glatt, C. R. and Jones, R. T.: A Program for Generating Plotted Information from Digital Data. Aerophysics Research Corporation Technical Note TN-142, February 1972.

10. Glatt, C. R.: A Program for Generating Vehicle Geometry Pictures in Arbitrary Image Planes. Aerophysics Research Corporation Technical Note TN-143, February 1972.

APPENDIX I-A

DATA BASE CONSTRUCTION

The data base consists of a *free storage array* where desired information is stored and a directory of unique name-oriented identifiers for the stored information. The directory acts as a table of contents and identifies the location of the data and the number of elements that reside in the *free storage array*. A brief description of the variable can also be stored in the directory. An advanced *keyword* access technique called RANDAC (Reliable and Accurate Name-Oriented Directory Access Code) has been developed for storing and accessing data in the data base through the use of the name-oriented directory. Approximately 10,000 variables per second can be located using this technique. ODINEX makes extensive use of RANDAC for communicating information to and from the data base.

1. DATA BASE-SIZE

The ODINEX program has been written such that both the design data base and the data base directory can be varied in length. Further, the data base directory can be expanded to provide more definitive information. The nominal size of the design data base is 300 elements. This can be expanded to the limits of the computer or approximately 50,000 elements (on the CDC 6600) with the alteration of a single dimension in the ODINEX main program. In addition, more than one data base may be defined for any one simulation.

The nominal size of the data base directory is 70 entries including four words of definitive information. Both the number of entries and the length of the definition can be expanded or contracted as simply as the design data base discussed above.

2. DESCRIPTION OF INDIRECT ACCESS METHOD

Indirect access techniques like RANDAC employ similar algorithms. Items are entered into a directory using a pointer, or *probe*, which is computed from the name of the item by means of some *hash coding* scheme. As long as no two inserted items have the same hash code, retrieval of the information can be performed in a single step regardless of the size of the directory.

However, when two items have the same hash code, a *collision* is said to exist. If a collision is encountered upon entering an item, an alternate directory location must be defined. This is accomplished by *chaining* the collision location to the alternate location in the directory. Upon retrieval of the colliding item, the collision must be *resolved* by following the chain until the item is found.

2.1 Hash Coding the Key

Every character used by the computer has a unique numerical representation. Combinations of characters which form words also display uniqueness characteristics. For example, the word GPAK does not have the same numerical representation as FPAK. This uniqueness characteristic is used by RANDAC in assigning a value to the directory *probe*, a candidate entry location in the directory.

If the directory were very large (say 2^k-1, where k is the computer word length) then hash coding would not be necessary. The unique numerical representation would be used as the probe. For smaller directories, the minimum requirement for hash coding is to modify the numerical representation of the key with the length of the directory. This provides a probe value which is within the limits of the directory but may not point to a unique location in the directory.

The objective of hash coding is to spread the calculated addresses uniformly over the available directory locations thereby reducing the number of collisions which may occur.

These methods are broadly subdivided into *logical* and *arithmetic*. Logical methods seek to eliminate adverse characteristics, such as imbedded blanks, which tend to group items with dominant numerical characteristics. Arithmetic methods seek to alter the numerical representation into a more unique form by performing some arithmetic operation on it.

The hash coding technique used in RANDAC *performs no logical or arithmetic operation on the key*. It has been shown in tests of many directories that operations performed on the key are a *greater penalty* in time than the resolution of the collisions hash coding seeks to avoid. The number of collisions typically runs from 10 to 20 per cent for uncoded keys. This means that more than 80 per cent of the variables can be accessed with a single operation. If hash coding is used beyond the normal MOD function, 100 per cent of the variables require more than one operation. Further, there is no guarantee that the coded key will significantly reduce the collisions.

2.2 Resolving Collisions

After the initial entry has been made into a directory, the possibility exists for the computed addresses of new keys to duplicate existing entries causing a collision between the storage locations allocated to each. Alternate locations must be established for colliding entries. In general, when the table is nearly full, many collisions may occur while probing the table for an empty slot. Hence, some procedure is needed which generates additional calculated addresses until an empty slot is found, probing the entire table if necessary. Of course, the same procedure for generating additional calculated addresses must be used when the item is looked up later.

In practice, when the RANDAC routine is initially called, it is not necessary to specify whether an item is being entered or being looked up. What is required of the routine is to determine the address at which the offered key belongs and to report whether the key was already entered. Then the calling routine can make the entry or extract the information, as appropriate. The procedure then will be to generate successive hash addresses until encountering either the location that *contains the desired key* _or_ *unused location*. In the latter case the key can be entered in the unused location.

2.3 Storage and Retrieval Method

The RANDAC method of storage and retrieval involves a directory entry of four basic elements:

KEY	A unique literal representation which identifies the information being stored. It may be one or more words which are hash coded into an address called a probe.
VALUE	One or more words of information associated with the KEY
HASH TABLE	Table of directory entry locations addressed by the probe. This table is appended to and the same length as the directory.
COLLISION TABLE	Table of alternate directory locations which *chain* directory entries with the same hash address or probe.

These four elements form the width of the directory which can be variable depending upon how many words are used for the KEY and VALUE. Figure A1 shows a schematic of the directory layout. The length of the directory is also specified by the user.

When a FIND operation is requested, the key is converted into a hash address, KPROBE.

1. If the hash table at that address is empty, the logical variable FOUND is returned as .FALSE. The user then has the option of installing a new directory entry with an INSTAL operation. In this case, the directory entry is made at the next available location, KFREE. That location is loaded

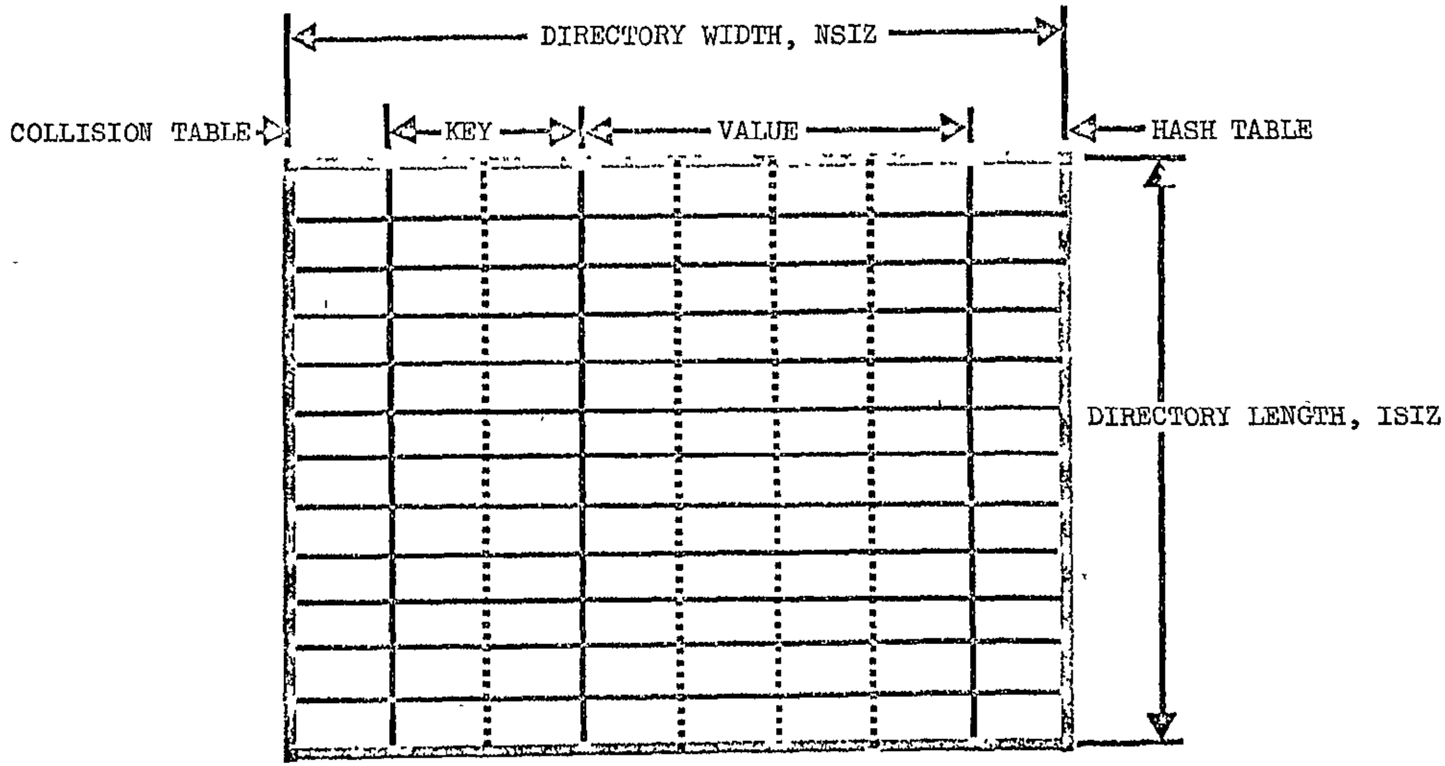


Figure A1. Composition of the Directory, FSL in the hash table at the probe address, and the next available location, KFREE, is bumped by one entry.

2. If the hash table at the probe address is occupied, the directory entry associated with the hash table entry is compared with the key.

2a. If the key compares, then the logical variable FOUND is returned as .TRUE. The user has the option of installing new information at that location with INSTAL operation or deleting the entry with a DELETE operation.

2b. If the key does not compare, then the collision location, which is part of the directory entry, is checked for an alternate directory location.

2c. If the collision location is occupied, the directory entry associated with that address is compared with the key. Items 2a through 2c are repeated until either the key is located in the directory or the collision location is empty.

3. If the collision location is empty, the variable FOUND is returned as .FALSE. The user has the option of installing the new directory entry. In this case, the next available directory location is used, and the address of that entry is loaded into the collision location of the last entry in the chain. KFREE is bumped by one entry. In the event that a directory entry is deleted, the current value of KFREE is stored in the collision locations of the deleted entry, and the deleted entry location is used for KFREE.

APPENDIX I-B
MULTIPLE PROGRAM EXECUTION

Multiple program execution is the key to success for the ODINEX concept. The objective is to provide a vehicle design synthesis made up of several individual design/mission programs. This is desirable from the designer's standpoint because it makes the synthesis highly modular and quite amenable to design concept changes. From the user's standpoint, it places little additional learning burden in excess of the knowledge required to use the individual programs. Further, the computer core requirements do not exceed the requirements of the individual programs.

The full benefit of the ODINEX program is realized when a control file is *built* and executed in the same job stream. The user can select by input the program stream he wishes to execute. Each program has a *catalogued procedure*, a file containing the necessary control cards to execute the program. This *procedure* is stored in the control card data base. Further, the user can specify matching and/or optimization loops within the program community. The use of catalogued procedures requires a system level utility program which allows the user to specify alternate files for job control (other than card input). A special utility program developed for ODINEX called CCLINK provides all the capability needed. Versions of this utility are operational on all Control Data Corporation CYBERNET computers and is in general use throughout the country. CCLINK has been designed to minimize the impact on the SCOPE system.

1. DESCRIPTION OF CCLINK

CCLINK is a program designed for the 6000 series computers which allows the user to transfer to an alternate control card file for job control. Conditional branching to selected files can be accomplished by testing an index register. The value in the index register is controlled by the ancillary program SETIDEX. CCLINK offers the user the ability to execute multiple program jobs with relatively few control cards. Further, it provides a looping capability useful in design matching and optimization problems. Other advantages of CCLINK include

- Reduces card handling errors
- Reduces errors due to bad control cards
- Provides standard procedures for heavily used programs
- Maintains minimum core requirements for all catalogued programs

In general, CCLINK simplifies the use of the computer resulting in fewer errors. This is a benefit to all users.

2. USE OF CCLINK

CCLINK is a control card-callable program which reloads the control card buffer from a given file. The execution of CCLINK is dependent on the validity of the relation of the control card index register and the comparison integer with respect to one of the conditional operators.

The index register is nominally set to zero and can be incremented or decremented by the control card SETIDEX(i); where i is positive or negative increment.

Call format:

CCLINK(lfn,xx,n)

where

lfn = the logical file name of the linkage file

xx = a conditional operator (one of the following)

LT (less than); link if CCIR LT<n

LE (less, equal)

GT (greater than)

GE (greater, equal)

EQ (equal)

NE (not equal)

omitted (unconditional linkage implied)

n = the comparison integer

Figure B1 is an example of CCLINK including use of the indexing feature SETIDEX. The SETIDEX feature which is not essential to the DIALOG system does provide useful capability for controlling program execution loops.

Assume the user had a library of programs which he could execute sequentially to perform an interdisciplinary design function. Further, assume the sequence of the program required execution ten times in order to satisfy the scaling and matching requirements. Figure B1 shows a schematic of the sequence to be performed and the sequential execution of four programs followed by the incrementing of an index. The sequence of programs is iteratively executed until the value of the index reaches ten. At this value the simulation job is terminated. The control card set up for doing this job with CCLINK and SETIDEX is shown in Figure B1.

An additional capability of the CCLINK software package is the FORTRAN callable routine LINK. LINK is a run compiled program loaded with the user library which permits the user to specify the next control card file to be used following termination of the current program. The calling sequence is

CALL LINK (lfn)

where lfn is the logical file name from which the next control card will be obtained. This program gives the user complete logical control over the sequence of programs to be executed.

3. SYSTEM INTERFACE REQUIREMENTS FOR CCLINK

The CCLINK software package consists of four programs/subprograms. They basically allow the use of alternate control card files on the CDC 6000 series computers. These programs are listed below and are followed by a brief description of the scope system interface requirements.

	JOB.	
	COPYBF,INPUT,USERFIL.	Builds a file, USERFIL, from the first file of the input stream.
	REWIND,USERFIL.	Establishes the SCOPE file pointer at the beginning of file USERFIL.
	CCLINK,USERFIL.	Transfers control to file USERFIL.
	7-8-9	End of file indicator preceeding input.
USERFIL	....	Control cards required to access, load and execute one or more user programs.
	SETIDEX,+1.	Increments the CCLINK index counter by +1.
	CCLINK,USERFIL,LT,10.	Transfers control to file USERFIL, if CCLINK index counter is less than 10, otherwise exits.
	EXIT.	Job terminates when this statement is executed.
	6-7-8-9	End of job indicator following last statement of job.

NOTE: The file pointer to file USERFIL is initially positioned at the beginning of this file, and is not altered during execution of this job.

Figure B1. Example Use of CCLINK

1. CCL is a PPU program that handles setting or clearing the index and the actual linking of control cards. It must be loaded with the system.

2. CCLINK is a CPU program that calls CCL to return the index value, tests the index against conditions on the control card, and calls CCL to link the control card, if necessary.

3. SETIDEX is a CPU program that calls CCL to *alter the index.*

4. LINK is a RUN FORTRAN-callable routine to call CCL to link the control card. It is stored with the user programs.

Almost all Scope 3 systems will accept these programs *without modification.* The theory of operation is that 1AJ uses a one-word FSP entry to define the next record of the control card source. CCL re-establishes that word with a pointer to a user-provided file. A *control feature* is provided by the use of eighteen bits in the control point area which is referred to as the index. *The cards establishing the location of these eighteen bits are marked in the source decks.* They should be installed into the SCPTEXT. Index setting and testing need not be used, *eliminating the need for control point area storage.*

APPENDIX I-C

ODIN SIMULATION EXAMPLE

1. Summary

This report presents the results of a recent optimization study using ODINEX. The objective was to determine the optimal engine size and mass distribution of the stages of a two-stage, fully reusable launch system having common liquid rocket engines. The vehicle's mass calculation and trajectory simulation were synthesized by the SSSP computer program of Reference C1. The optimal design solution was obtained by a straightforward multivariable search procedure available through the use of AESOP described in Reference C2.

A cost sensitivity analysis was performed at the end of the study using the program DAPCA described in Reference C3. These four programs are a part of the ODIN (Optimal Design INtegration) design program community constructed for Langley Research Center. They were linked as shown in Figure 1 to form the synthesis reported in this note.

A significant improvement in payload was achieved as illustrated by the results given below.

	Payload, Pounds	Vacuum Thrust Pounds	Mass Ratio $W_{STARTBURN}/W_{ENDBURN}$
NOMINAL	28500	470000	3.045 (V_{STG}=10400 FPS)
OPTIMUM	31850	527000	2.715 (V_{STG}=9160 FPS)

The results indicate a twelve per cent change in both mass ratio and vacuum thrust of the engines from the nominal. The change in mass parameter had a significant effect on staging velocity as indicated in the table.

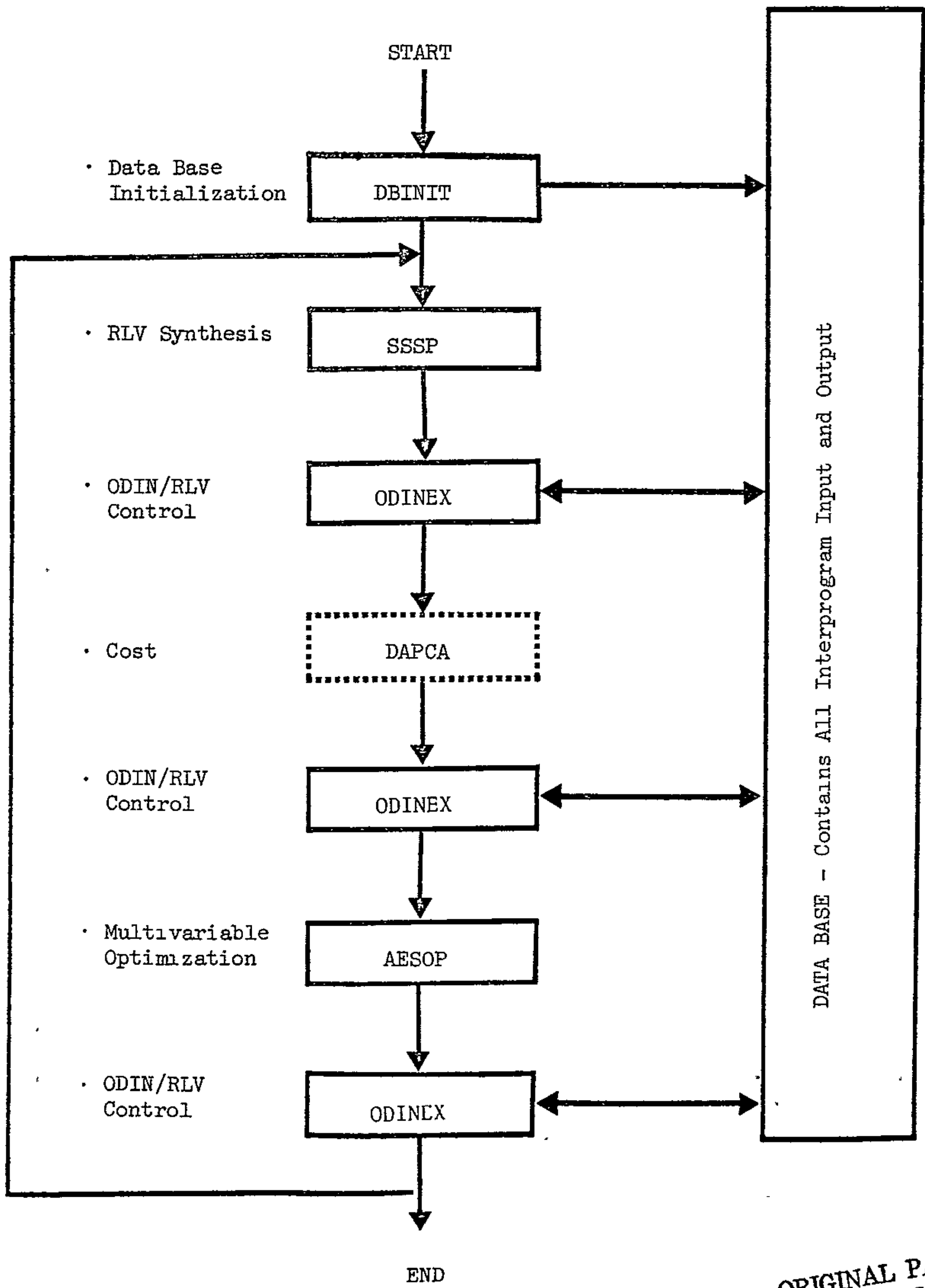


Figure 1. Schematic of Sample ODIN/RLV Model

2. Introduction

Recent study contracts with NASA Langley Research Center and the Air Force Flight Dynamics Laboratory have lead to the development of a new concept in modular programming. One objective of the study efforts was to facilitate the formation of very large design synthesis programs. This report exemplifies the use of the concept.

It basically consists of multiple program execution where each program is a *separately developed and documented computer program* which performs a particular technological function. By selective stringing of several programs together an interdisciplinary design function can be performed. The difficulty associated with stringing programs arises in communicating information from one program to another in an efficient and reliable manner. This difficulty has led to the development of the ODINEX system described in Reference C4.

ODINEX is a computer program which dynamically constructs and maintains a data base of information. It is executed before and after each technology program forming the synthesis. Its function is to merge data from the preceding program with the data base and extract information from the data base needed by the next program. It has been successfully used for a number of synthesis applications.

This report presents an example of an ODIN/RLV synthesis using the ODINEX concept. In the synthesis, the mission and sizing is simulated by the SSSP program of Reference C1, and the optimization of the selected design parameters is performed by the program, AESOP, of Reference C2. The test results are probably not significant in the overall shuttle design context but serve to exemplify the use of the ODINEX system for forming very large synthesis programs.

3. The ODIN/RLV Mission and Vehicle Model

The two-stage space shuttle concept is shown in Figure 2. The two vehicles are mated and launched vertically with the orbiter attached in a piggy back fashion on the booster. A typical mission is logistics resupply of an orbital space station.

During the boost phase only the booster engines are operating. At staging, when the booster has depleted its main propellants, the stages separate, and the booster performs a glide/decelerate maneuver to subsonic velocity where the turbojet engines are started and cruiseback is initiated for a conventional airplane type landing at an airfield in the proximity of the launch site. Subsonic cruise range to the launch site is about 400 nautical miles. After staging the orbiter engines are ignited, and the stage accelerates to orbit, docks at the space station and transfers passengers and cargo to the station. A gliding/maneuvering entry into the earth's atmosphere is made so that the vehicle arrives over the landing site. Turbojet engines are ignited, and the vehicle makes a conventional airplane type landing.

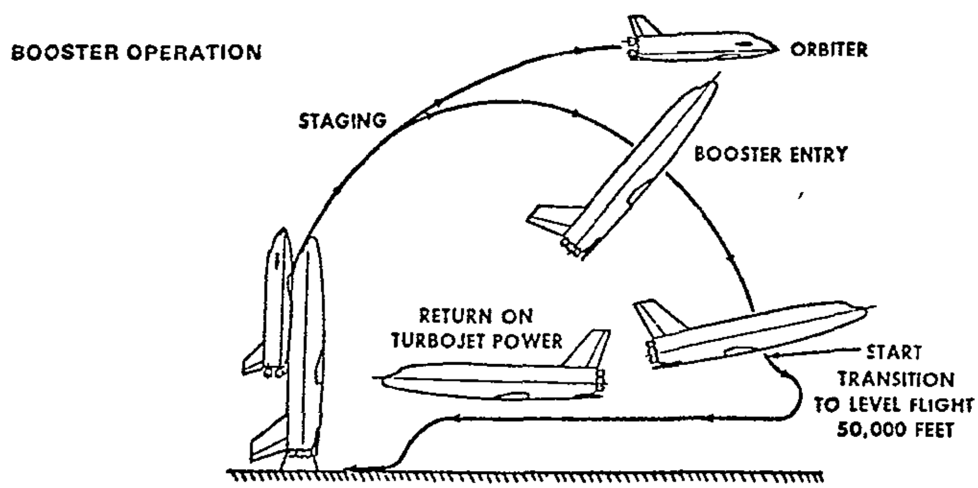


Figure 2. The Two-Stage Space Shuttle Concept

3.1 Trajectory Profile.

A baseline ascent trajectory profile was established within the SSSP program. The terminal conditions for the ascent trajectory are perigee injection into an elliptic parking orbit (50 nautical miles perigee altitude with 100 nautical miles apogee altitude). This parking orbit provides a reasonable start for space station logistics and other missions. Insertion at this low altitude provides good performance and allows an efficient entry trajectory for the booster stage. The orbiter entry trajectory is initiated by retro from orbit and is, therefore, not dependent on the ascent trajectory and is not simulated in SSSP.

The ascent trajectory sequence is as follows:

1. Vertical rise for a specified time (14 seconds)

2. Pitchover (10 seconds)

3. Gravity turn ($\alpha = 0$ between thrust and velocity vector) maneuver to booster propellant depletion, stage separation (booster entry initiated).

4. Orbiter burn with linear cotangent steering ($\cot \psi = A + Bt$) to perigee insertion

A multiplier on the pitch rate during the initial pitchover maneuver is iteratively determined to yield a specified dynamic pressure at stage separation. The separation dynamic pressure of two psf was chosen to yield a near-optimal ascent trajectory, a "cool" booster entry trajectory with short cruise range requirements and an acceptable environment for stage separation if necessary.

The orbiter flight is then simulated with the two parameters A and B (the cotangent of the pitch attitude being linear in time) being determined to yield specified injection altitude (h_f) and injection flight path angle (γ_f) at attainment of the specified injection velocity (V_f). The weight iteration necessary to make propellant extended by the orbiter to achieve V_f agree with the weight-sizing propellant computations is iteratively computed in SSSP. The simplified pitch control program yields near-optimal performance for a wide variety of vehicle parameters and yields good convergence properties for the trajectory iterations.

At stage separation the trajectory conditions (V_s, h_s, γ_s, etc.) are stored for use in determining the cruise fuel requirements of the booster stage. This determination may be accomplished with a number of program options all of which are described in detail in Reference C1. and are based on the cruise range requirement for the mission and Breguet's equation for the fuel required for a constant L/D cruise. Subsonic L/D and specific fuel consumption are input constants. The option used for determination of the example booster cruise requirement was

> Flyback range to the launch site as a function of the dynamic pressure at stage separation. Since dynamic pressure was constant as determined by the iteration above, the flyback range was also constant. Some error is involved in this assumption for the second part of the optimization since the staging velocity varied considerably during the perturbations.

3.2 Vehicle Characteristics and Constraints.

The fundamental concept of earlier space shuttle synthesis is the complete reusability of both stages with the maximum use of such common hardware items as the main rocket engines. The booster and orbiter engines are essentially the same; although a larger number of engines will be installed on the booster than on the orbiter (e.g., eleven booster engines and two orbiter engines) and an extendable skirt was added to the orbiter nozzle to improve vacuum performance. The computation sequence in the SSSP was chosen to best provide this propulsion system commonality. SSSP input specifies the ratio of the booster to orbiter vacuum thrust, T_b/T_o, the number of engines per stage, and the thrust and the specific impulses (vacuum and sea level) of each type.

Man rating the vehicle for a wide variety of possible passenger types imposes a limit on loads of three g's. This requires throttling of the rocket engines during the main burn after a specified axial load is reached.

Structural, dynamic, and thermodynamic constraints (such as maximum αq loading, balance, heating, etc.) are not considered in the SSSP. The effects of these constraints can be analyzed externally by monitoring and using SSSP trajectory, weights, and geometry data. Alternately, the simulation can be augmented by program modules that adequately represent the effects of these constraints. The SSSP provides for a number of basic options as described in Reference 1 which may be utilized to constrain the basic vehicle design or to investigate alternative approaches to the space shuttle concept. The option used in the present example was

> Fixed GLOW with an iteration for determination of the payload

Fixed size common engines were assumed for both the booster and the orbiter stages.

3.3 Weights and Geometry.

The weight/volume portion of the SSSP is a library of weight and volume equations for the components of space shuttle vehicles. The subprogram accepts inputs in the form of coefficients to various weights and volume equations written in terms of the geometry of a particular vehicle type. It uses existing weight data plus inputs describing the thermal protection system, propulsion and other subsystems, as well as performance mass ratios and other mission requirements derived from the trajectory subprogram. The second generation weight breakdown in MIL-M-38310 was used as a guide to determine the level of detail and order of weight output listings. Weight equations for each component or group of components were written by incorporating appropriate provisions for varying weights correctly as the vehicle weight and/or size changes. Volume equations for important volume components are also included. An iteration process is employed so that component weights/volumes and overall weights/volumes are mutually consistent.

The SSSP program solves the following basic problem: for a specified payload weight and mass ratio, find the stage gross weight and volume. This problem is solved separately for the orbiter and booster stages; then iterations are performed to satisfy the specified mission fixed GLOW constraint.

The weight equations used in SSSP rely heavily on a unit weight approach, with any sophistication based more on selection of proper weight coefficients for input rather than on the equations themselves. This method gives the user more latitude for judgment and permits the same equations to be used for a wide range of vehicles. To do this, however, a data library of vehicle weight coefficients obtained from detailed design studies must be available. The source for the example problem is the Weight/Volume Handbook, Volume II, of Reference 1. The Weight/Volume Handbook contains the compilation of all the weight/sizing equations utilized in the SSSP subprogram and a procedure for obtaining the proper coefficients that are input for each equation.

3.4 Other Limitations on the Simulations.

The sample case involves a LO_2/LH_2 fueled Space Shuttle configuration which existed as an initial point design developed by personnel at the NASA Manned Spacecraft Center in early 1970. This configuration was converted to the SSSP input requirements for the example problem.

Each stage has a single vertical surface and wings with a fourteen degree leading edge sweep. The theoretical (gross) wing area is fixed, and the orbiter and booster wing loading is computed internally at the initial entry and initial flyback conditions, respectively. The thermal protection system assumes coverage of the total body wetted area excluding the aerodynamic surfaces. However, the corresponding weight coefficient is an average value that is representative of a combined insulation-cover panel weight less than 0.75 lb/ft^2 on the upper surface and greater than 1.75 lb/ft^2 on the lower surface.

The orbiter has two main engines and the booster has eleven. Both stages have fixed gimble system weights. The subsonic cruise engines for both stages operate with LH_2 propellant stored in main tanks. Each stage has a ten per cent contingency factor applied to the dry weight.

The system payload volume is fixed at 10,600 ft^3. The orbiter main propellant flight performance reserves are based on 225 fps total characteristic velocity (≈0.75 per cent of mission velocity to parking orbit insertion) with a 1500 fps incremental velocity requirement (O/F mixture ratio = 5) reserved for the post-insertion or on-orbit maneuvers. The booster flight performance reserve is fixed at 1370 lbs. of propellants. For both stages the main impulse propellants (including reserves) utilize an O/F mixture ratio = 6 for the sizing basis. The booster sizing base includes fixing the main impulse mass ratio at a value of 3.045 for the nominal evaluation. *This ratio was selected as a control parameter for the optimization runs.*

The stage burn sequence selected was that of the sequential stage burns. The orbiter main engine unit thrust (vacuum) was fixed initially at 470 K lbs/engine and later varied as an AESOP parameter. The booster main engine unit thrust at vacuum conditions was set at 0.968556 of the orbiter thrust level to account for the difference in the nozzle configurations. The specific impulses values for each ascent flight phase were constant for each stage.

The booster gross weight was specified at an input value of 3,384,390 pounds and the payload was used as the performance criteria by AESOP.

The booster reference range method selected was that based on the staging dynamic pressure with the cruise fuel method being the simplified single segment mode of operation.

The ascent trajectory mission profile includes the following bases:

1. Built-in atmosphere tables used from liftoff to parking orbit insertion.

2. Table input combined axial aerodynamic characteristics for boost phase only, no normal aerodynamic coefficients and no aerodynamics for orbiter ascent phase.

3. Launch pad at KSC coordinates, launch azimuth = 37.65 degrees

4. Fourteen-second vertical rise from liftoff, ten-second pitchover, target to a two psf staging dynamic pressure, linear cotangent steering during orbiter burn.

5. Throttle booster engines at axial load of 2.5 g limit; orbiter engines at 3.0 g limit.

6. Terminate simulation sections 01 on 2.5 g limit; (02 automatic on booster propellant depletion); 03 and 04 on three seconds and four seconds (relative), respectively; 05 on eighty seconds (relative); 06 on 3.0 g limit; 07 automatic on speoified insertion velocity.

7. Perigee parking orbit insertion at 51/100 nautical miles in 55-degree inclination orbit.

4. Payload Optimization

The performance criteria for the example problem was chosen to be payload. The objective was to maximize payload while constraining the gross lift-off weight (GLOW) to 3384390 pounds. Two optimization problems were posed. A one-parameter problem varying the vacuum thrust of the common engines from its nominal value of 470000 (which produced 28500 pound payload) resulted in an improvement in payload over 1000 pounds. A two-parameter problem was then posed which included vacuum thrust but added mass ratio of the booster to the AESOP parameter list. Mass ratio (MR) is defined as the ratio of mass at the start of the burn to the mass at the end of the burn. The SSSP program was set up to solve for the size of the orbiter which yields the fixed GLOW. Therefore, the variation in the mass ratio of the booster, in effect, varies the mass distribution between the stages and has a direct influence on the staging velocity. The two-parameter solution used the best one-parameter solution as a nominal. This yielded an additional 2300 pounds of payload over the one-parameter solution.

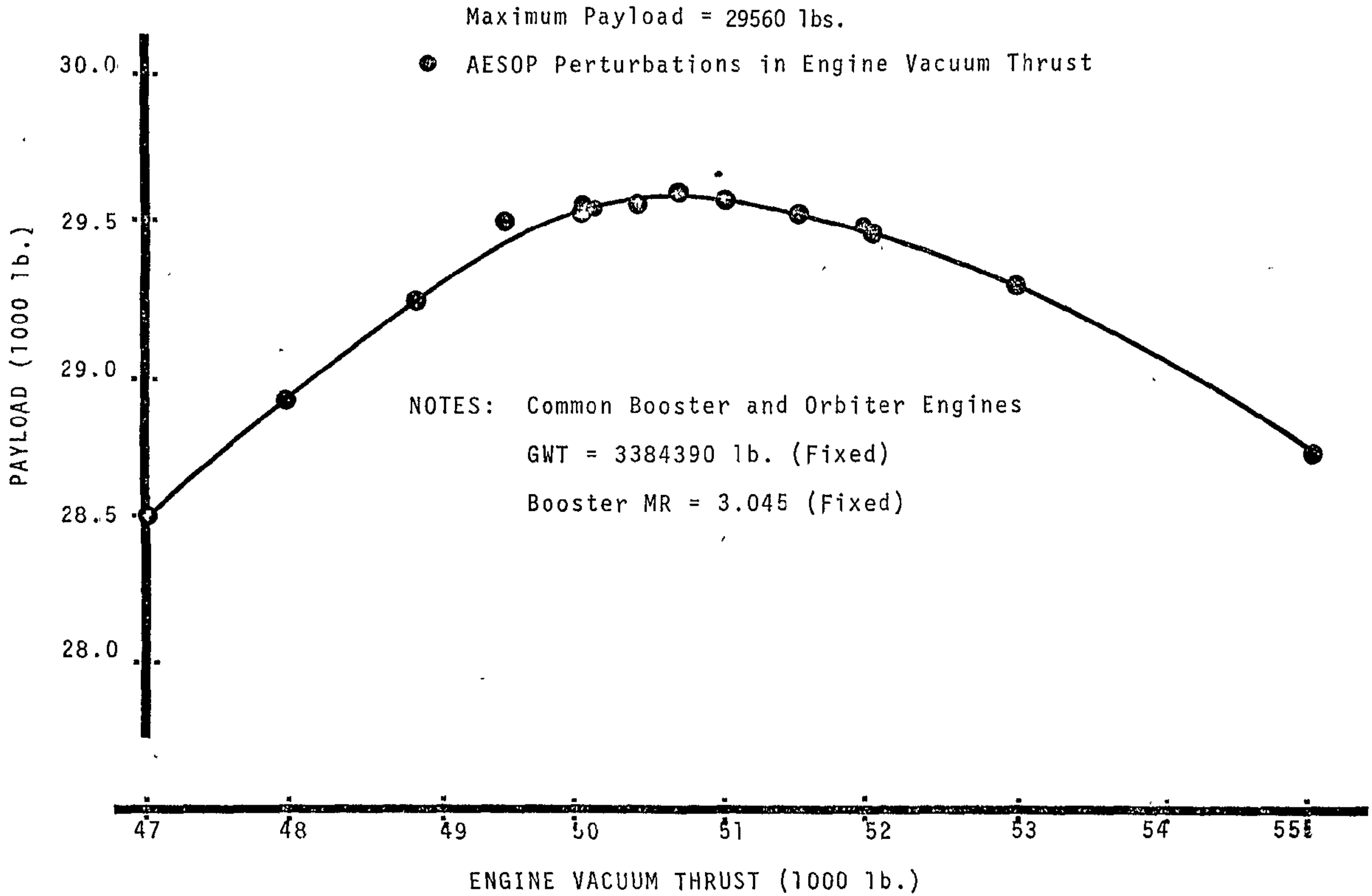
4.1 Selection of Engine Vacuum Thrust.

In the selection of vacuum thrust the function of SSSP was to evaluate the influence of vacuum thrust on payload. The function of AESOP was to perturb the value of thrust based on the value of payload generated by SSSP. Special considerations in communicating the information between programs is given in Section 5.

The sectioning search in AESOP was used to maximize the payload. The result of this search is presented in Figure 3. The nominal and best performance are compared in the table below.

PERFORMANCE	VACUUM THRUST	PAYLOAD
Nominal	470000 lbs.	28500
Best	505000 lbs.	29560

FIGURE 3.
SHUTTLE ENGINE SIZING STUDY



4.2 Selection of Booster and Engine Size.

In the second optimization study, the booster mass ratio was added to the AESOP parameter list, and payload was retained as the performance criteria. Mass ratio is the ratio of the initial mass to the mass at engine cutoff. For a fixed GLOW, this parameter is a measure of the booster size.

Figure 4 shows a chronological history of the payload as the values of thrust and mass ratio were perturbed. The first series of calculations using random ray search resulted in a spurious perturbation on the eighth evaluation (parameter driven to the boundaries) which yielded a significant increase in performance. This chance improvement was accepted as the nominal for the second series of calculations using the creeping search. Upon widening the boundaries, further gains are produced as the creeping search was continued.

A sectioning search on booster mass ratio was performed at the twenty-sixth evaluation to determine the sensitivity on the parameters. The results are shown in Figure 5. Here, the vacuum thrust was fixed at 504000 pounds, and the mass ratio was allowed to vary over a wide range. The results indicated slightly more than 500 pounds to be gained by altering the mass ratio. In reality, a great deal of payload was gained by perturbing the mass ratio and thrust simultaneously.

This can be easily seen in Figure 6, a contour map of payloads as a function of the two parameters: vacuum thrust and mass ratio. The section at evaluation twenty-six only located the ridge line shown dotted. The maximum payload is shown at a considerably higher thrust value.

FIGURE 4. MASS DISTRIBUTION AND ENGINE SIZE OPTIMIZATION

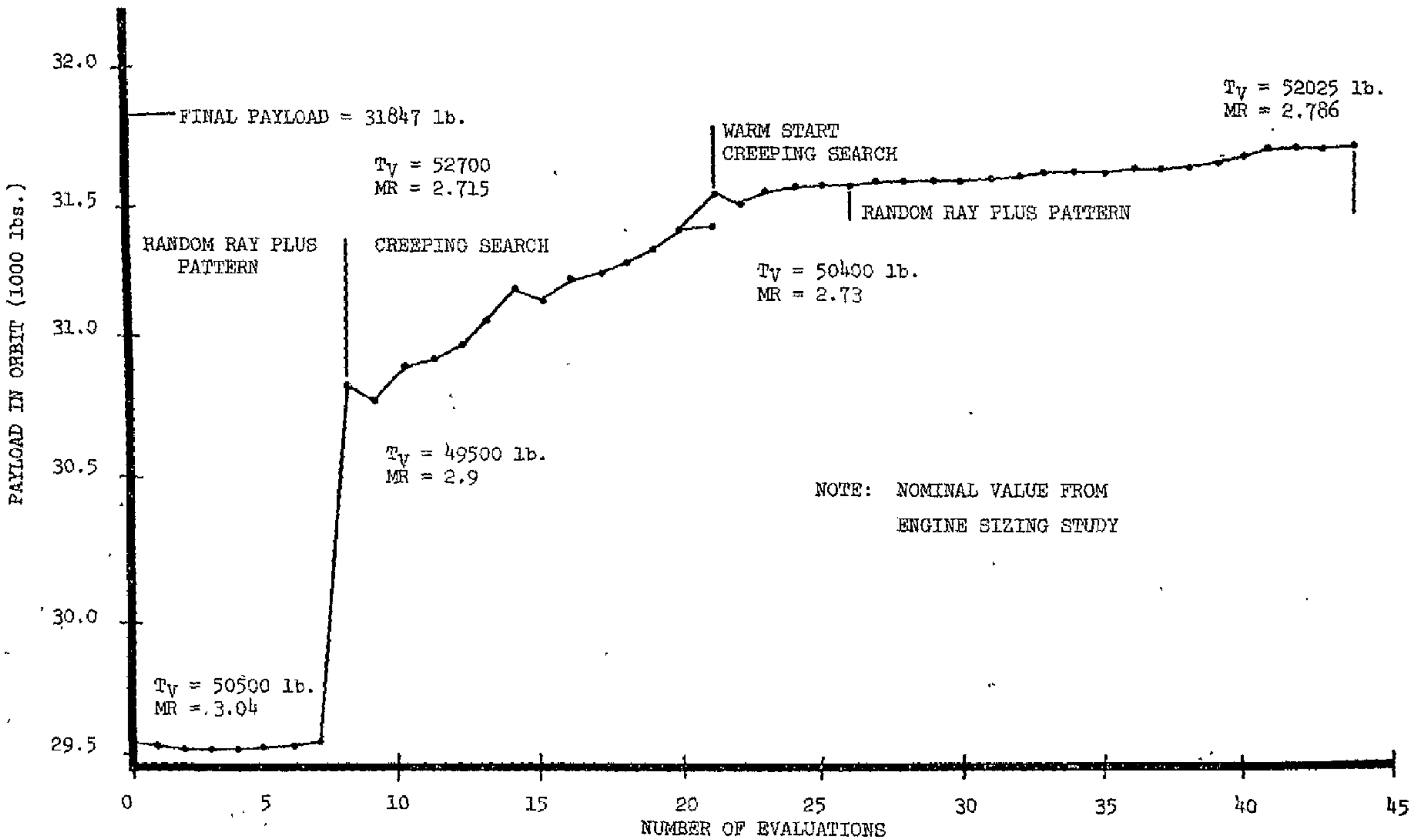


FIGURE 5

BOOSTER SIZING STUDY

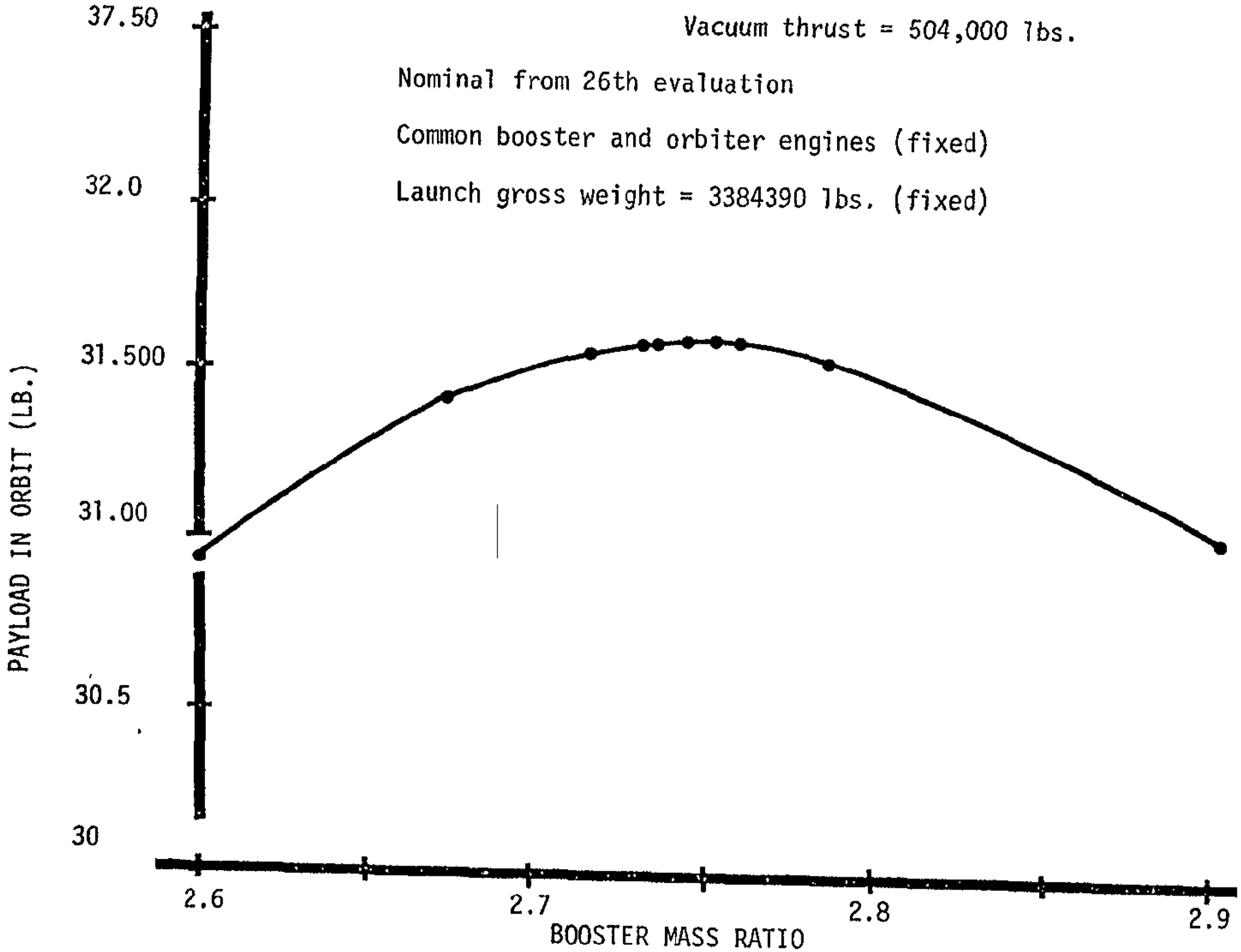
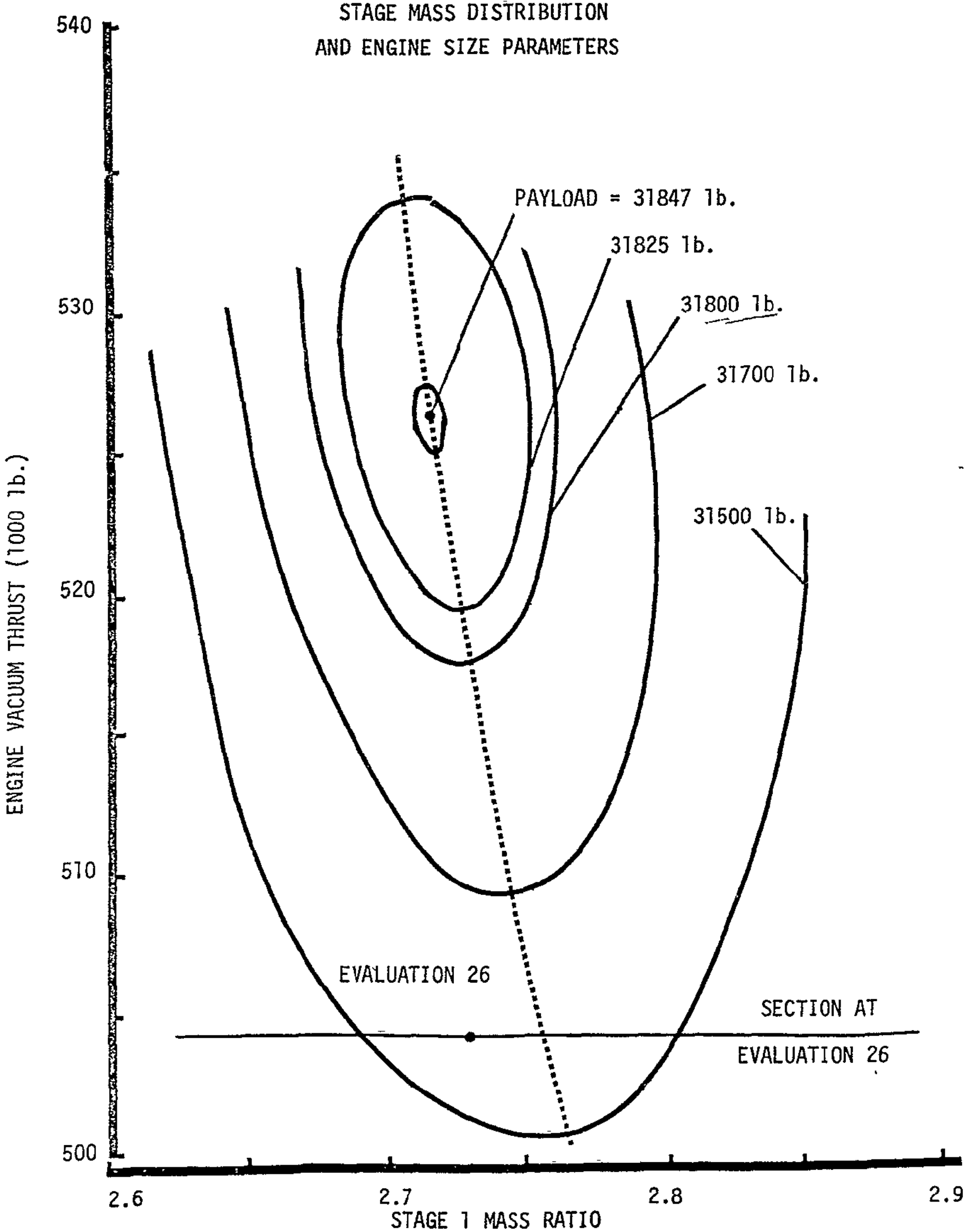


FIGURE 6.
PAYLOAD CONTOURS
STAGE MASS DISTRIBUTION
AND ENGINE SIZE PARAMETERS



4.3 Cost Sensitivity.

A cost sensitivity analysis was performed at the optimum payload point using the program DAPCA, a computer program for determining aircraft development and production costs, described in Reference C3. The DAPCA computer program computes development and production costs for the major subsystems of flyaway aircraft, engines, airframes, etc. Avionics cost is input. The cost output generated by the program is in the form of cost-quantity, unit and cumulative average, improvement curves. Most of the input relates to aircraft and engine performance characteristics, such as gross weight, speed, engine type, engine thrust, etc.

The program was developed primarily for horizontal take-off aircraft so the basic methodology underlying the program is not strictly applicable to the reusable launch vehicle. However, the costing principles are not unlike those used in more applicable programs such as Reference 5. The latter program is also available for use in ODIN.

The coupling of DAPCA for the ODIN/RLV synthesis involved the execution of the program twice, once for the booster and once for the orbiter. The cost sensitivity was based on first unit cost. For the booster, the gross weight was the launch weight; the speed was staging velocity. The thrust for both the booster and orbiter was the rocket engine vacuum thrust (common engines). For the orbiter, the gross weight was the weight at staging; the speed was the difference between insertion and staging velocity. The results are shown in the following table.

T_{VAC} 1000	MR	BOOSTER COST $1,000,000	ORBITER COST $1,000,000	TOTAL COST $1,000,000	PAYLOAD LB
527	2.715	39759	14850	54609	31847
527	2.725	39889	14804	54693	31846
527	2.705	39626	14896	54522	31844
529	2.715	39890	14878	54768	31846
525	2.715	39628	14821	54449	31846

5. Data Intercommunication Between SSSP and AESOP

The data requirements are largely identical with the individual programs involved. The exceptions are associated with the data extracted from the data base. In these specific cases, the user replaces the value ordinarily input on the data card with a data base variable name isolated by special delimiters. In the normal sequence of calculations, ODINEX *reads* the input data prior to execution of the user program and *builds* a new input stream modified by the values associated with the data base names identified.

The discussion that follows exemplifies the special considerations for the one-parameter selection of vacuum thrust. The simulation consisted basically of two functional programs, SSSP and AESOP. The input to these programs are modified by the user to extract selected data base values. SSSP generates the payload for a fixed GLOW, and ODINEX merges it in the data base. AESOP extracts the payload and generates parameter perturbations, and ODINEX merges them into the data base.

In the sequence of calculations, the data base is initialized with the control parameter

ALPHA=470000.,

This name and value are placed in the data base by ODINEX, which also reads the entire SSSP input searching for key words denoted by the delimiters @name@. The vacuum thrust in the SSSP program is defined as C(129) and the input card for the simulation is punched as follows:

C(129) = @ALPHA@,

ODINEX identifies the word ALPHA as a data base variable and replaces it and the special delimiters with the data base value

C(129) = 470000.,

For the one-parameter problem, the ODINEX function is complete. The SSSP input has been modified with the selected data base values (in this case, only vacuum thrust).

The SSSP program is then executed with the modified input stream. It *doesn't know* that a data base value is being used. The program executes as if it were the only job in the stream. All the normal output functions are available as well as a special name list output the only physical modification to the SSSP program. These data are used by the ODINEX program. One of the special namelist output variables is WPAYLO, the payload in orbit as determined by the SSSP program.

WPAYLO = 28500.,

This name and value are placed in the data base by ODINEX. This completes the SSSP function; the simulation continues to AESOP. AESOP is a separately executed program with the primary function of perturbing the control parameters (thrust, in this case) based on changes in performance (WPAYLO). This function is performed by reading the performance as input and writing the control parameter perturbations as output. Using the ODINEX system, both are data base variables. AESOP is *unaware* of the system which it is optimizing. It simply executes as an isolated program. The user of ODIN/RLV specifies the performance criteria,

FUNCTN(1)=@WPAYLO@,

as part of the card input along with the other AESOP inputs such as search procedures. ODINEX recognized the word WPAYLO as a data base variable and replaces it and the delimiters @ with the data base value

FUNCTN(1)=28500.,

In this manner AESOP obtains the current performance of the system from the data base. Upon execution AESOP perturbs the control parameter ALPHA(1), an AESOP array element which coincides with the data base name ALPHA. This array is written in the special namelist file to be interrogated by ODINEX. ODINEX alters the original value of ALPHA to the perturbed value as determined by AESOP.

This completes the SSSP/AESOP optimization *nominal evaluation.* The parameter has been perturbed, and the new value placed in the data base. The simulation is repetitively recycled in this manner until the optimization is terminated by AESOP.

6. Sample Output from ODINEX

Figures 7 through 11 show the design data base configuration during one evaluation of the synthesis.

Figure 7 - After Initialization
Figure 8 - After SSSP
Figure 9 - After DAPCA (booster)
Figure 10 - After DAPCA (orbiter)
Figure 11 - After AESOP

All variable names and definitions together with some initial values were created at initialization, Figure 7. Updates are indicated in Figures 8 through 11 by the name.

* * * * * THERE ARE CURRENTLY 19 DATA BASE ENTRIES AS FOLLOWS * * * * * * * * * * * * *

NAME	LOCATION	DIMENSION	CURRENT VALUE(S)	ORIGIN	DESCRIPTION
AESOP	39	1		DEFINE	AESOP USED IN SIMULATION
ALPHA	7	2	525000. 2.71	INITAL	AESOP CONTROL PARAMETERS
BUILD	19	1	0	DEFINE	DYNAMIC DB BUILD OPTION
CRDSKP	1	1	250	ADD	CARD SKIP OPTION
DBDUMP	31	1		DEFINE	CAUSES DATA BASE DUMP
DISPLAY	13	1	WPAYLO	INITAL	SCOPE DISPLAY FUNCTION
ELTIME	23	1	4.30200000	DEFINE	TOTAL ELAPSED TIME
JJBEST	25	1		DEFINE	BEST PERFORMANCE EVALUATION
JJJ	17	1		DEFINE	AESOP EVALUATION COUNTER
MIXR	3	1	2.714999999999989200	ADD	PROPELLANT MIXTURE RATIO
RUNID	21	1	*ODIN JAN 22, 72	DEFINE	IDENTIFICATION FOR DIALOG RUN
TCOSTB	35	1		DEFINE	TOTAL COST OF BOOSTER
TCOSTN	37	1		DEFINE	TOTAL COST OF ORBITER
THRUST	5	1	525000.000000000000	ADD	ENGINE VACUUM THRUST
VSTGB	29	1		DEFINE	BOOSTER STAGING VELOCITY
VSTGO	33	1		DEFINE	ORBITER STAGING VELOCITY
WGROSB	11	1		DEFINE	GROSS WEIGHT OF BOOSTER
WGROSO	15	1		DEFINE	GROSS WEIGHT OF ORBITER
WPAYLO	27	1		DEFINE	PAYLOAD OF THE LAUNCH SYSTEM

* *

Figure 7. Data Base Configuration After Initialization

* * * * * THERE ARE CURRENTLY 19 DATA BASE ENTRIES AS FOLLOWS * * * * * * * * * * * * *

NAME	LOCATION	DIMENSION	CURRENT VALUE(S)	ORIGIN	DESCRIPTION
AESOP	39	1		DEFINE	AESOP USED IN SIMULATION
ALPHA	7	2	525000. 2.71	INITAL	AESOP CONTROL PARAMETERS
BUILD	19	1	0	DEFINE	DYNAMIC DB BUILD OPTION
CRDSKP	1	1	250	ADD	CARD SKIP OPTION
DBDUMP	31	1		DEFINE	CAUSES DATA BASE DUMP
DISPLAY	13	1	WPAYLO	INITAL	SCOPE DISPLAY FUNCTION
ELTIME	23	1	4.30200000	DEFINE	TOTAL ELAPSED TIME
JJBEST	25	1		DEFINE	BEST PERFORMANCE EVALUATION
JJJ	17	1		DEFINE	AESOP EVALUATION COUNTER
MIXR	3	1	2.7149999999999989200	ADD	PROPELLANT MIXTURE RATIO
RUNID	21	1	*ODIN JAN 22, 72	DEFINE	IDENTIFICATION FOR DIALOG RUN
TCOSTB	35	1		DEFINE	TOTAL COST OF BOOSTER
TCOSTN	37	1		DEFINE	TOTAL COST OF ORBITER
THRUST	5	1	525000.0000000000000	ADD	ENGINE VACUUM THRUST
VSTGB	29	1	0.91501786518072E+04	WTOUT	BOOSTER STAGING VELOCITY
VSTGO	33	1	0.25853097798483E+05	WTOUT	ORBITER STAGING VELOCITY
WGROSB	11	1	0.26662821532874E+07	WTOUT	GROSS WEIGHT OF BOOSTER
WGROSO	15	1	0.71802039902725E+06	WTOUT	GROSS WEIGHT OF ORBITER
WPAYLO	27	1	0.31845761977751E+05	WTOUT	PAYLOAD OF THE LAUNCH SYSTEM

* *

Figure 8. Data Base Configuration After SSSP

* * * * * THERE ARE CURRENTLY 19 DATA BASE ENTRIES AS FOLLOWS * * * * * * * * * * * * *

NAME	LOCATION	DIMENSION	CURRENT VALUE(S)	ORIGIN	DESCRIPTION
AESOP	39	1		DEFINE	AESOP USED IN SIMULATION
ALPHA	7	2	525000. 2.71	INITAL	AESOP CONTROL PARAMETERS
BUILD	19	1	0	DEFINE	DYNAMIC DB BUILD OPTION
CROSKP	1	1	85	ADD	CARD SKIP OPTION
DBDUMP	31	1		DEFINE	CAUSES DATA BASE DUMP
DISPLAY	13	1	WPAYLO	INITAL	SCOPE DISPLAY FUNCTION
ELTIME	23	1	25.6240000	DEFINE	TOTAL ELAPSED TIME
JJBEST	25	1		DEFINE	BEST PERFORMANCE EVALUATION
JJJ	17	1		DEFINE	AESOP EVALUATION COUNTER
MIXR	3	1	2.7149999999999989200	ADD	PROPELLANT MIXTURE RATIO
RUNID	21	1	#ODIN JAN 22, 72	DEFINE	IDENTIFICATION FOR DIALOG RUN
TCOSTB	35	1		DEFINE	TOTAL COST OF BOOSTER
TCOSTO	37	1	0.48673855937800E+05	DAPOUT	TOTAL COST OF ORBITER
THRUST	5	1	525000.0000000000000	ADD	ENGINE VACUUM THRUST
VSTGB	29	1	0.91501786516072E+04	WTOUT	BOOSTER STAGING VELOCITY
VSTGO	33	1	0.25853097798483E+05	WTOUT	ORBITER STAGING VELOCITY
WGROSB	11	1	0.26607821532874E+07	WTOUT	GROSS WEIGHT OF BOOSTER
WGROSO	15	1	0.71802039902725E+06	WTOUT	GROSS WEIGHT OF ORBITER
WPAYLO	27	1	0.3194576197751E+05	WTOUT	PAYLOAD OF THE LAUNCH SYSTEM

* *

Figure 9. Data Base Configuration After DAPCA (Booster)

* * * * * THERE ARE CURRENTLY 19 DATA BASE ENTRIES AS FOLLOWS * * * * * * * * * * * * *

NAME	LOCATION	DIMENSION	CURRENT VALUE(S)	ORIGIN	DESCRIPTION
AESOP	39	1		DEFINE	AESOP USED IN SIMULATION
ALPHA	7	2	525000. 2.71	INITAL	AESOP CONTROL PARAMETERS
BUILD	19	1	0	DEFINE	DYNAMIC DB BUILD OPTION
CRDSKP	1	1	85	ADD	CARD SKIP OPTION
DBDUMP	31	1		DEFINE	CAUSES DATA BASE DUMP
DISPLAY	13	1	WPAYLO	INITAL	SCOPE DISPLAY FUNCTION
ELTIME	23	1	26.8380000	DEFINE	TOTAL ELAPSED TIME
JJBEST	25	1		DEFINE	BEST PERFORMANCE EVALUATION
JJJ	17	1		DEFINE	AESOP EVALUATION COUNTER
MIXR	3	1	2.7149999999989200	ADD	PROPELLANT MIXTURE RATIO
RUNID	21	1	*ODIN JAN 22, 72	DEFINE	IDENTIFICATION FOR DIALOG RUN
TCOSTB	35	1	48673.8559378066565B	ADD	TOTAL COST OF BOOSTER
TCOSTN	37	1	0.19037115923077E+05	DAPOUT	TOTAL COST OF ORBITER
THRUST	5	1	525000.0000000000000	ADD	ENGINE VACUUM THRUST
VSTGB	29	1	0.91501786518072E+04	WTOUT	BOOSTER STAGING VELOCITY
VSTGO	33	1	0.25853097798483E+05	WTOUT	ORBITER STAGING VELOCITY
WGROSB	11	1	0.26662821532874E+07	WTOUT	GROSS WEIGHT OF BOOSTER
WGROSO	15	1	0.71802039502725E+06	WTOUT	GROSS WEIGHT OF ORBITER
WPAYLO	27	1	0.3184576197751E+05	WTOUT	PAYLOAD OF THE LAUNCH SYSTEM

* *

Figure 10. Data Base Configuration After DAPCA (Orbiter)

* * * * * THERE ARE CURRENTLY   19 DATA BASE ENTRIES AS FOLLOWS * * * * * * * * * * * * *

NAME	LOCATION	DIMENSION	CURRENT VALUE(S)	ORIGIN	DESCRIPTION
AESOP	39	1		DEFINE	AESOP USED IN SIMULATION
ALPHA	7	2	0.525E+06 0.2715E+01	AESOUT	AESOP CONTROL PARAMETERS
BUILD	19	1	0	DEFINE	DYNAMIC DB BUILD OPTION
CRDSKP	1	1	32	ADD	CARD SKIP OPTION
DBDUMP	31	1		DEFINE	CAUSES DATA BASE DUMP
DISPLAY	13	1	WPAYLO	INITAL	SCOPE DISPLAY FUNCTION
FLTIME	23	1	28.0920000	DEFINE	TOTAL ELAPSED TIME
JJBEST	25	1	1	AESOUT	BEST PERFORMANCE EVALUATION
JJJ	17	1	0	AESOUT	AESOP EVALUATION COUNTER
MIXR	3	1	2.7499999999989200	ADD	PROPELLANT MIXTURE RATIO
RUNID	21	1	*ODIN JAN 22, 72	DEFINE	IDENTIFICATION FOR DIALOG RUN
TCOSTB	35	1	48673.855937806655658	ADD	TOTAL COST OF BOOSTER
TCOSTN	37	1	0.19037115923077E+05	DAPOUT	TOTAL COST OF ORBITER
THRUST	5	1	525000.000000000000	ADD	ENGINE VACUUM THRUST
VSTGB	29	1	0.91501785518C72E+04	UTOUT	BOOSTER STAGING VELOCITY
VSTGO	33	1	0.25853097798483E+05	WTOUT	ORBITER STAGING VELOCITY
WGROSB	11	1	0.26662821532874E+07	WTOUT	GROSS WEIGHT OF BOOSTER
WGROSO	15	1	0.71802039902725E+06	WTOUT	GROSS WEIGHT OF ORBITER
WPAYLO	27	1	0.31847761977 51E+05	WTOUT	PAYLOAD OF THE LAUNCH SYSTEM

* *

Figure 11. Data Base Configuration After AESOP

7. Conclusions

The significant conclusion from this report is not the payload obtained from the calculations; although this is another of a long history of AESOP optimization examples where little additional effort was required to make an *optimal design* program out of a *point design* program. The significance is rather that a rapid method has been developed for doing design synthesis work which is a reliable, modular, and efficient alternative to other methods, which generally involve considerable reprogramming effort.

The ODINEX system allows the user to essentially build a synthesis *through input* selecting the model complexity from a library of programs representing literally hundreds of man-years of effort in both engineering and programming talent. This can be done in a short period of time. Often, results are obtained within a few hours of conceiving the problem. The speed is usually limited by the user's familiarity with the individual programs.

8. References for Appendix I-C

C1. ———., Space Shuttle Synthesis Program (SSSP), Volume I, Part 1, Report GDC-DBB70-002, General Dynamics Corporation, December 1970.

C2. Hague, D. S. and Glatt, C. R., An Introduction to Multivariable Search Techniques for Parameter Optimization (and Program AESOP), NASA CR-73200, April 1968.

C3. Boren, H. E., Jr., DAPCA: A Computer Program for Determining Aircraft Development and Production Costs, The Rand Corporation, RM-5221-PR, February 1967.

C4. Glatt, C. R., Hague, D. S., and Watson, D. A., ODINEX: An Executive Computer Program for Linking Independent Programs, NASA CR-2296, 1973.

C5. Hague, D. S. and Glatt, C. R., Optimal Design Integration for Military Flight Vehicles, ODIN/MFV, Section 9.1, AFFDL-TR-72-132, December 1972.