

AD-760 568

OPTIMAL DESIGN INTEGRATIONS OF
MILITARY FLIGHT VEHICLES - ODIN/MFV

D. S. Hague, et al

Aerophysics Research Corporation

Prepared for:

Air Force Flight Dynamics Laboratory

December 1972

DISTRIBUTED BY:

NTIS

National Technical Information Service
U. S. DEPARTMENT OF COMMERCE
5285 Port Royal Road, Springfield Va. 22151

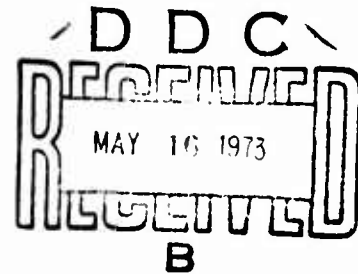
AFFDL-TR-72-132

**OPTIMAL DESIGN INTEGRATION OF
MILITARY FLIGHT VEHICLES
-ODIN/MFV-**

*D. S. HAGUE AND C. R. GLATT
AEROPHYSICS RESEARCH CORPORATION*

TECHNICAL REPORT AFFDL-TR-72-132

DECEMBER 1972



Approved for public release; distribution unlimited.

Report of the
NATIONAL TECHNICAL
INFORMATION SERVICE
5052 BRADLEY BLVD
ALEXANDRIA, VA 22304

**AIR FORCE FLIGHT DYNAMICS LABORATORY
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO**

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

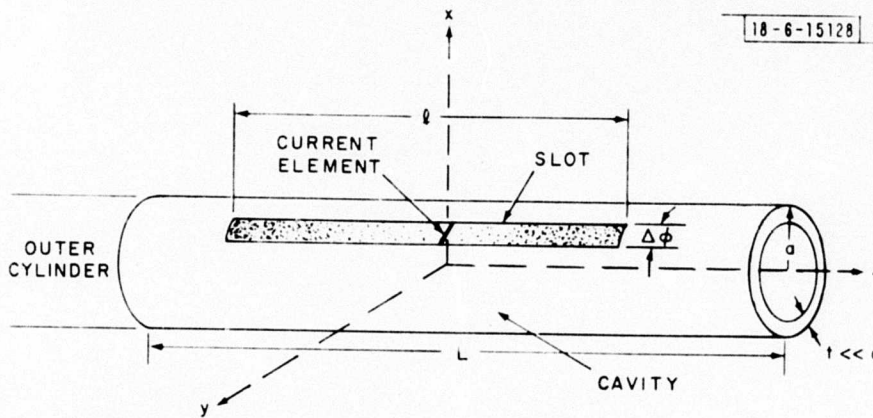
| | | |
|---------------------------------|--------------|---------|
| FIG | UNCLASSIFIED | 1 |
| ENC | UNCLASSIFIED | 1 |
| UNCLASSIFIED | | 1 |
| JUSTIFICATION | | |
| BY | | |
| DISTRIBUTION AVAILABILITY CODES | | |
| Dist. | Avail. | Special |
| A | | |

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

895656-8
AD

ERRATA SHEET
for Technical Note 1973-17

Please substitute the figure below for Figure 1 on Page 5 of Technical Note 1973-17 (H. R. Raemer, "Admittance of an Axial Slot Backed by a Cylindrical Cavity Inside an Infinite Cylinder," 3 April 1973).



8 May 1973

Publications Office
M.I.T. Lincoln Laboratory
P. O. Box 73
Lexington, Mass. 02173

FOREWORD

The research project outlined in this report was completed in the period from May 1971 to September 1972 under the sponsorship of the Air Force Flight Dynamics Laboratory, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio. The report was prepared by Aerophysics Research Corporation, Bellevue, Washington, under Project 1431 and United States Air Force Contract F33615-71-C-1480. Mr. David T. Johnson, AFFDL/FXG, was the cognizant Air Force representative for the study.

This report was authored by D. S. Hague and C. R. Glatt. J. F. MacRae organized the manuscript. Programming support was supplied by D.A. Watson and R. T. Jones. The study has led to a major advance in development of computer aided design techniques. This report and the study benefit directly from a number of previous Government-sponsored research studies including:

| | |
|---|--|
| USAF contract AF33(616)-6848, | for trajectory equation and program development |
| USAF contract AF33(657)-8829, | for development of the variational optimization procedure and program |
| USAF contract F33615-70-C-1036, | for development of a digital combat optimization and analysis program |
| NASA contract NAS 1-10692, | for additional development of the optimal design integration procedure |
| NASA contracts NAS 2-3691, NAS 2-4880, NAS 1-9936, NAS 3-13331 and NAS 2-6147 | for extension of the optimization procedure |

In addition to the development of improved computer aided design techniques, this project has resulted in considerable extension of the previously available AFFDL generalized steepest descent and segmented mission analysis computer programs.

This report was submitted by the authors in November 1972. The technical report has been reviewed and is approved.



PHILIP P. ANTONATOS
Chief, Flight Mechanics Division
Air Force Flight Dynamics Laboratory

UNCLASSIFIED

Security Classification

| DOCUMENT CONTROL DATA - R & D | | |
|--|---|--|
| <i>(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)</i> | | |
| 1. ORIGINATING ACTIVITY (Corporate author) Aerophysics Research Corporation P.O. Box 87 Bellevue, Washington 98009 | | 2a. REPORT SECURITY CLASSIFICATION Unclassified |
| | | 2b. GROUP N/A |
| 3. REPORT TITLE OPTIMAL DESIGN INTEGRATIONS OF MILITARY FLIGHT VEHICLES - ODIN/MFV | | |
| 4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Final Report | | |
| 5. AUTHOR(S) (First name, middle initial, last name) D.S. Hague and C.R. Glatt | | |
| 6. REPORT DATE Dec 72 | 7a. TOTAL NO. OF PAGES 587 590 | 7b. NO. OF REFS 130 |
| 8a. CONTRACT OR GRANT NO. F33615-71-1480 | 9a. ORIGINATOR'S REPORT NUMBER(S) None | |
| b. PROJECT NO. 1431 | 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) | |
| c. | | |
| d. | | |
| 10. DISTRIBUTION STATEMENT Approved for public release; distribution unlimited. | | |
| 11. SUPPLEMENTARY NOTES None | 12. SPONSORING MILITARY ACTIVITY Air Force Flight Dynamics Laboratory AFFDL/FXC Wright-Patterson AFB, Ohio | |
| 13. ABSTRACT ODIN/MFV is a digital computing system for the synthesis and optimization of military flight vehicle preliminary designs. The system consists of a library of technology modules in the form of independent computer programs and an executive program, DIALOG, which operates on the technology modules. The technology module library contains programs for estimating military flight vehicle characteristics, for example, aerodynamics, trajectory and geometry. In addition a generalized optimization module, plotting module, and a program precompiler are available in the technology module library. The initial program library was limited to only a few programs due to the limited scope of the study however other programs may readily be introduced into the library by a minor program modification. The DIALOG executive program controls the design synthesis and optimizations by operating on the technology module library under control of a user specified data input stream. Synthesis procedures in a given simulation are established by input data hence any number of vehicle component matching and sizing loops can be defined. | | |

DD FORM 1 NOV 65 1473

19

UNCLASSIFIED

Security Classification

UNCLASSIFIED

Security Classification

| 14 KEY WORDS | LINK A | | LINK B | | LINK C | |
|---|--------|----|--------|----|--------|----|
| | ROLE | WT | ROLE | WT | ROLE | WT |
| Design synthesis methodology, independent program library, executive program, optimization, procedures established by input data. | | | | | | |

UNCLASSIFIED

Security Classification

ABSTRACT

ODIN/MFV is a digital computing system for the synthesis and optimization of military flight vehicle preliminary designs. The system consists of a library of technology modules in the form of independent computer programs and an executive program, DIALOG, 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/MFV program technology module library.

The DIALOG 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/MFV 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/MFV program provides the designer with a "building block" approach to vehicle design. The design simulation parallels that now employed in industry; however, the ODIN/MFV 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/MFV 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

| <u>SECTION</u> | | <u>PAGE</u> |
|----------------|---|-------------|
| 1 | INTRODUCTION | 1-1 |
| 2 | STRUCTURE AND OPERATION OF THE OPTIMAL DESIGN INTEGRATION PROCEDURE FOR MILITARY FLIGHT VEHICLES, ODIN/MFV | 2-1 |
| | 2.1 Structure of the ODIN/MFV System | 2.1-1 |
| | 2.2 The Basic ODIN/MFV Program Elements | 2.2-1 |
| | 2.3 Installation of the ODIN/MFV | 2.3-1 |
| | 2.4 Sequential Independent Program Execution | 2.4-1 |
| | 2.5 Summary of the ODIN/MFV System | 2.5-1 |
| 3. | GEOMETRY | 3-1 |
| | 3.1 Program PANEL: A Computer Code for Generating a Paneled Aerospace Vehicle Surface Definition | 3.1-1 |
| | 3.2 Program IMAGE: A Computer Code for Display of the Three-Dimensional Objects | 3.2-1 |
| | 3.3 Program LRCACP: A Code for Producing Aircraft Configuration Plots | 3.3-1 |
| 4. | AERODYNAMICS | 4-1 |
| | 4.1 Program HABACP: The Gentry Hypersonic Arbitrary Body Aerodynamics Computation Program | 4.1-1 |
| | 4.2 Program ARPII: Rapid Supersonic Area Rule Aerodynamic Program | 4.2-1 |
| | 4.3 Program TREND: A Rapid Subsonic/Supersonic/ Hypersonic Aerodynamic Trade-Off Code | 4.3-1 |
| 5 | PROPULSION | 5-1 |
| | 5.1 Program GENENG II: A Program for Calculating Design and Off-Design Performance of Two- and Three-Spool Turbofans with as Many as Three Nozzles | 5.1-1 |
| 6 | MASS AND VOLUMETRIC PROPERTIES | 6-1 |
| | 6.1 Program VSAC: An Approximate Aircraft Mass Properties and Volume Analysis Code | 6.1-1 |
| 7 | PERFORMANCE | 7-1 |
| | 7.1 Program TOLAND: A Simplified Take-Off and Landing Analysis Code | 7.1-1 |
| | 7.2 Program NSEG II: A Rapid Approximate Segmented Mission Performance Analysis Code | 7.2-1 |

TABLE OF CONTENTS

| <u>SECTION</u> | | <u>PAGE</u> |
|----------------|--|-------------|
| | 7.3 Program ATOP II: Atmospheric Trajectory Optimization Program | 7.3-1 |
| 8 | ECONOMICS | 8-1 |
| | 8.1 Program PRICE: A Computer Program for Estimating Total Program Cost of Aircraft, Spacecraft and Reusable Launch Vehicles | 8.1-1 |
| 9 | STRUCTURES | 9-1 |
| | 9.1 Program SSAM: Swept Strip Aeroelastic Module | 9.1-1 |
| 10 | OPTIMIZATION | 10-1 |
| | 10.1 The Variational Steepest-Descent Method | 10.1-1 |
| | 10.2 Program AESOP: Computer Program for Multi-variable Search | 10.2-1 |
| 11 | PRECOMPILER TECHNIQUES | 11-1 |
| | 11.1 MACRO FORTRAN Precompiler | 11.1-1 |
| 12 | GRAPHICS | 12-1 |
| | 12.1 Program Plotter: Independent Plot Program | 12.1-1 |
| 13 | IN LINE COMPILATION | 13-1 |
| | 13.1 Program MYPROGRAM: Compilation at Execution Time Capability | 13.1-1 |
| 14 | OVERLAY CONSTRUCTION | 14-1 |
| | 14.1 Program AUTOLAY: Automatic Overlay Construction for CDC 6600 Computer | 14.1-1 |
| | APPENDIX I: AN INTRODUCTION TO DIALOG: A DESIGN INTEGRATION AND ANALYSIS LANGUAGE | I-i |

SECTION 1

INTRODUCTION

Efficient preliminary design of a modern military flight 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 aircraft which will operate near civilian population centers in times of peace, there exists an increasing requirement for satisfying environmental constraints, such as noise and engine exhaust pollution. Optimal design of a military flight vehicle 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

Military flight 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 military flight 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 engineers' 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 an interceptor 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 United States computational power. The ODIN procedure for military flight vehicles developed during the present study and the related study of Reference 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; and
- 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 study of Reference 5 contains all the above features. The system and its operation is described in Section 2. Description of the technology program library initially installed at Wright-Patterson Air Force Base follow in Section 3.

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 Figure 1-3.

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

- a. B1 bomber performance studies
- b. Air-to-air surface missile performance studies
- c. Advanced manned interceptor design studies

The most comprehensive problem investigated was the synthesis of an advanced manned interceptor; Figure 1-4 presents three-views of this vehicle as produced by the ODIN system. Figure 1-5 illustrates the complex system of technology modules executed to accomplish the synthesis.

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., Three-Degree-of-Freedom Problem Optimization Formulation, FDL-TDR-64-1, Part 1, Volume III, 1964.
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. _____, Study to Develop a Computer Program for Synthesis and Optimization of Reusable Launch Vehicles, Contract NAS 1-10692, Langley Research Center.

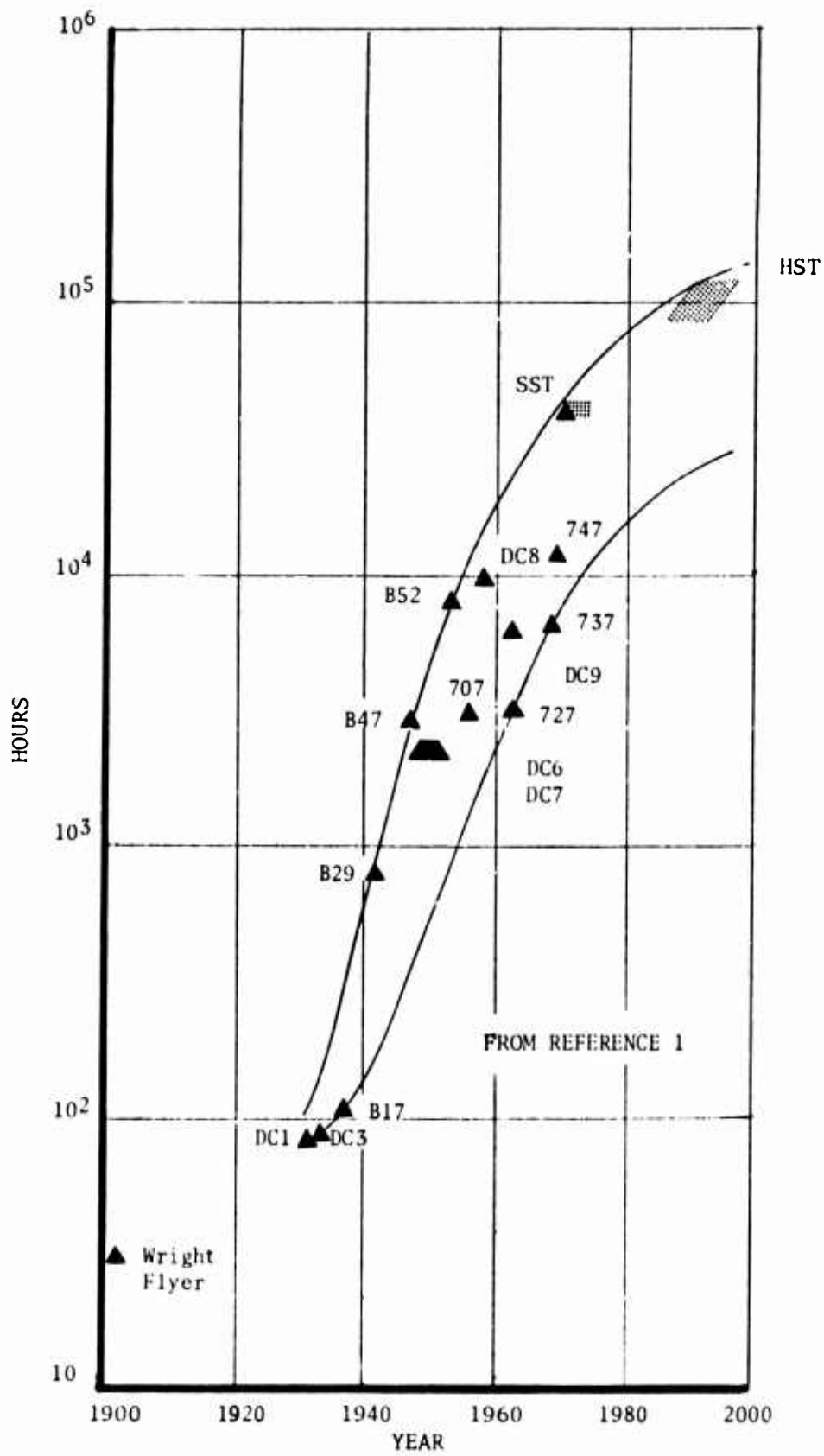


Figure 1-1. Wind Tunnel Hours for Vehicle Design

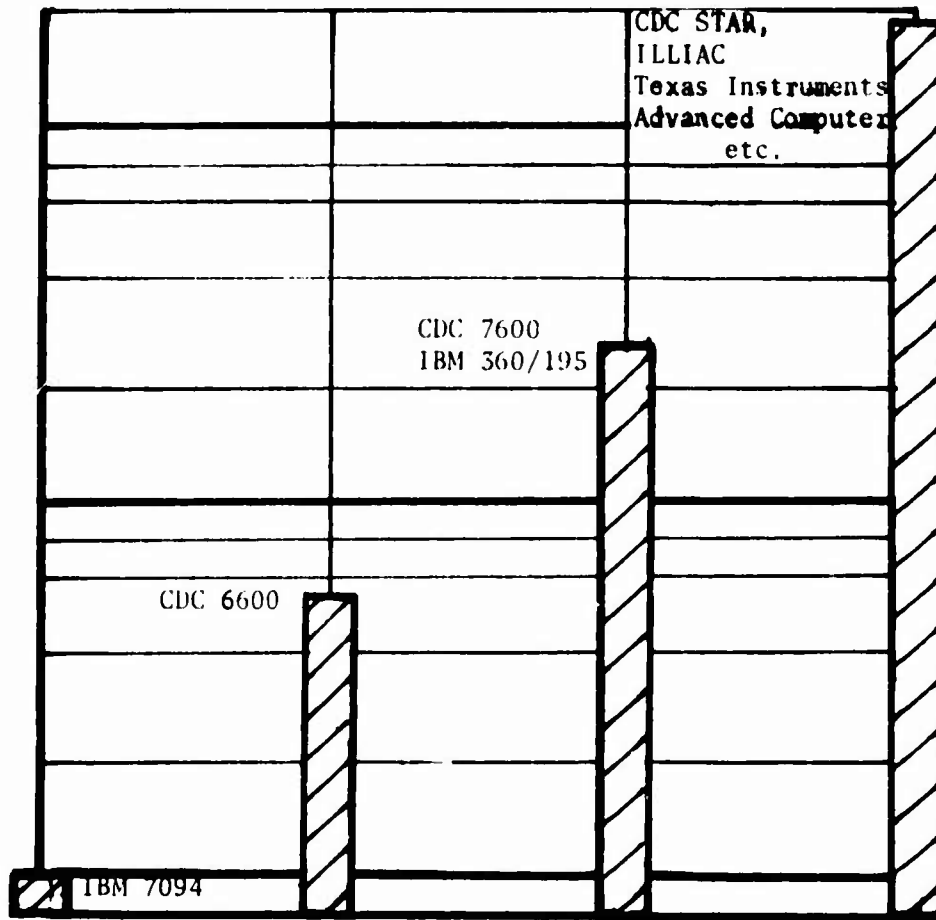


FIGURE 1-2. GROWTH OF AVAILABLE COMPUTATIONAL POWER

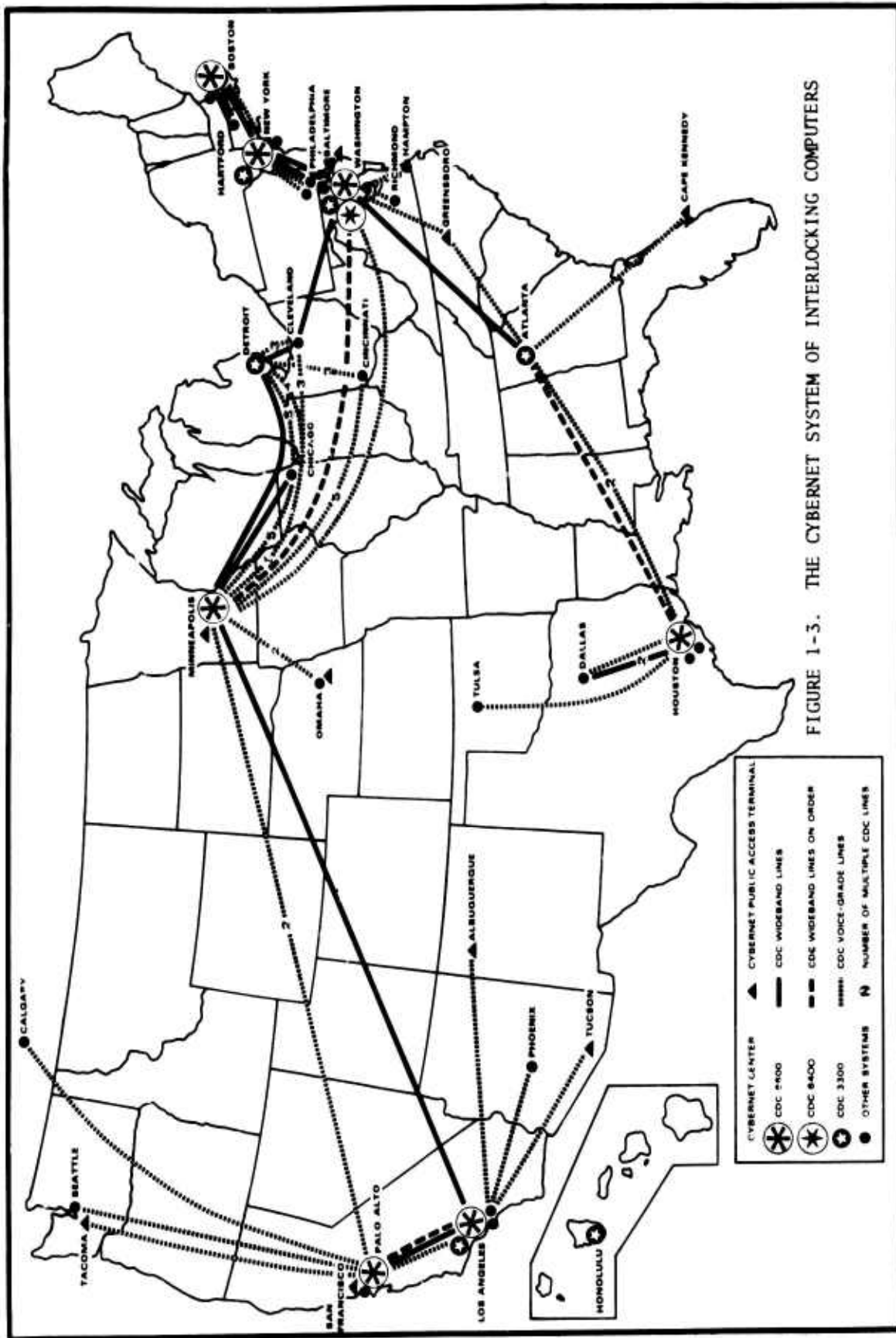
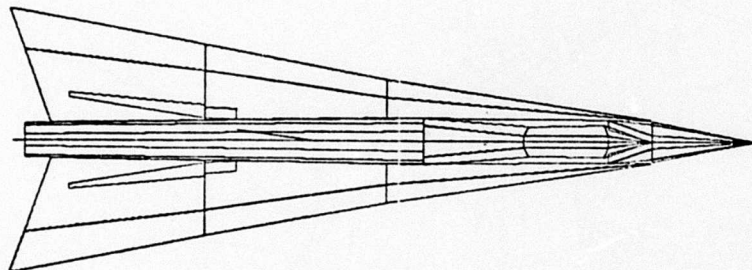


FIGURE 1-3. THE CYBERNET SYSTEM OF INTERLOCKING COMPUTERS

ODIN / IMAGE PICTURE DRAWING PROGRAM
ADVANCED MANNED INTERCEPTOR SKIN FRICTION MODEL



ODIN / IMAGE PICTURE DRAWING PROGRAM
AMI CONFIG SCL=1.000, SFX=1.000, SFY=1.000, CAMBER=1.000000

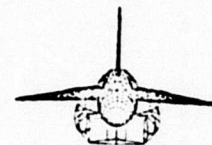
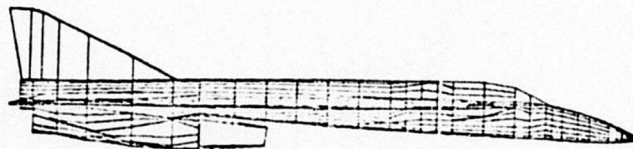
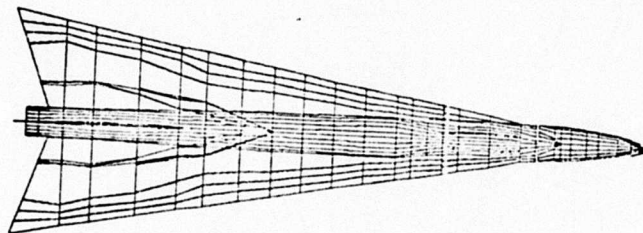
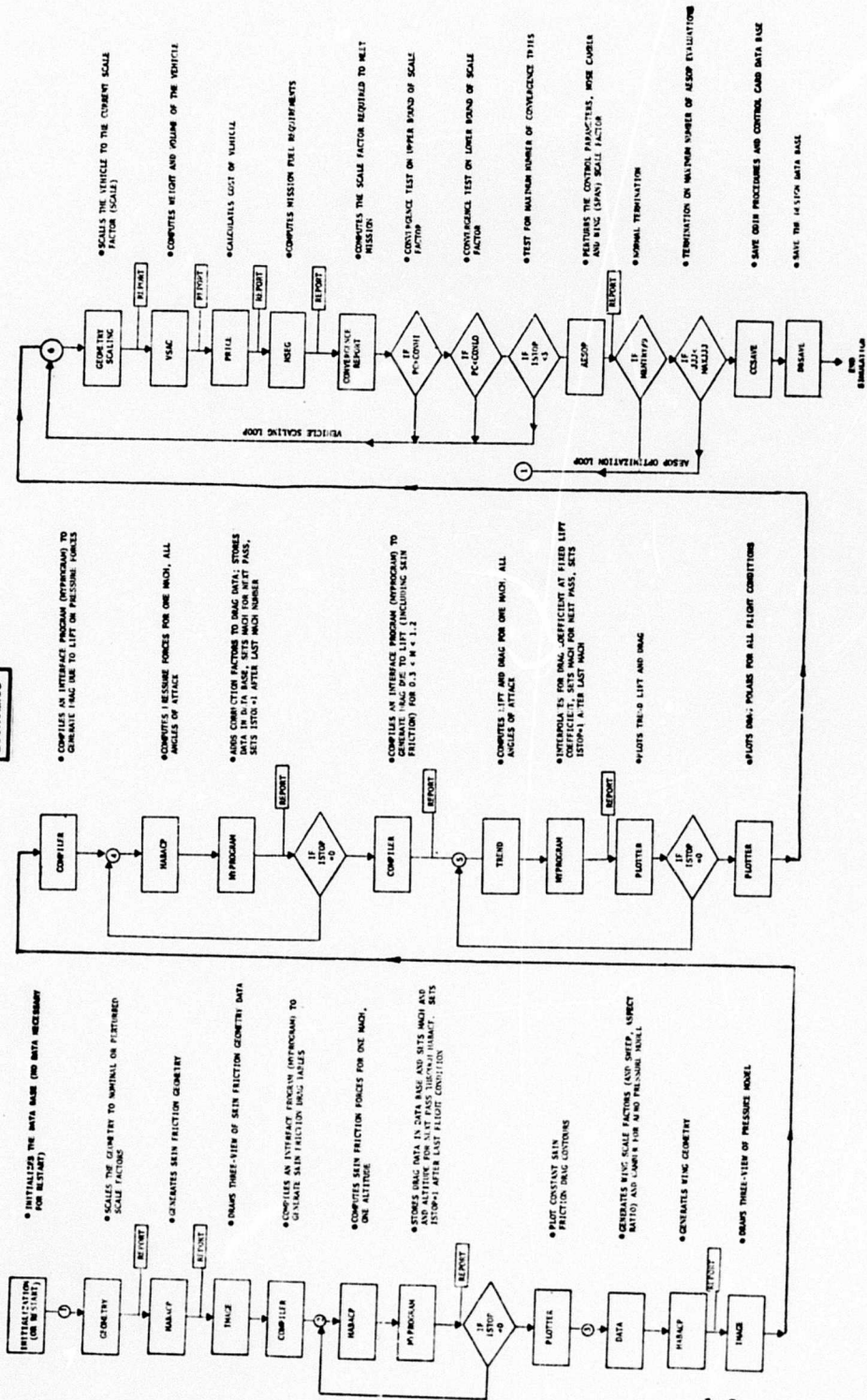
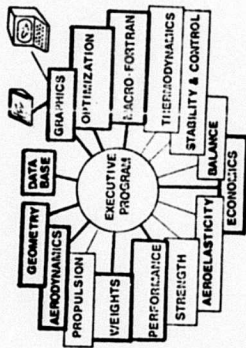


FIGURE 1-4. ADVANCED MANNED INTERCEPTOR VIEWS FROM THE ODIN GRAPHICS PROGRAM

FIGURE 1-5. ODIN SIMULATION OF
ADVANCED MANNED INTERCEPTOR



SECTION 2

STRUCTURE AND OPERATION OF THE OPTIMAL DESIGN INTEGRATION PROCEDURE FOR MILITARY FLIGHT VEHICLES, ODIN/MFV

This section describes the Optimal Design Integration Procedure for Military Flight Vehicles (ODIN/MFV) computational system, its structure, and its application. The ODIN/MFV 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/MFV library were developed independently of the present study, several under previous Government funded studies. The ODIN/MFV executive control program which allows the independent programs of the ODIN/MFV library to communicate with each other was developed entirely within the context of the present study, and the related NASA supported study of Reference 1.

In developing the ODIN/MFV a survey was conducted of existing technology programs and methodologies which are generally available to the aerospace industry. Programs surveyed are listed in References 2 through 130. The ODIN/MFV initial program library was limited to only a few of these programs due to the limited scope of the present study effort. Other programs may readily be introduced into the ODIN/MFV by a minor program modification as described later in this section.

REFERENCES:

Vehicle Synthesis References

1. ———., Study to Develop a Computer Program for Synthesis and Optimization of Reusable Launch Vehicles, contract NAS 1-10692, Langley Research Center.
2. Craidon, Charlotte B., Description of a Digital Computer Program (D2290) for Aircraft Configuration Plots, NASA TM X-2074, 1972.
3. Adams, J.D., Vehicle Synthesis of High Speed Aircraft, VSAC, Volume I, AFFDL-TR-71-40, June 1971.
4. Oman, B., Vehicle Synthesis for High Speed Aircraft, VSAC, Volume II, AFFDL-TR-71-40, June 1971.
5. Norton, Patrick J., Volume, Area, and Mass Properties of Aerospace Vehicles, Langley Research Center Program A3128.
6. ———., Space Shuttle Synthesis Program, SSSP, Volume I, Part 1, Engineering and Programming Discussion, General Dynamics GDC-DBB70-002, December 1970.
7. ———., Space Shuttle Synthesis Program, SSSP, Volume I, Part 2, Program Operating Instructions, General Dynamics, GDC-DBB70-002, December 1970.
8. ———., Space Shuttle Synthesis Program, SSSP, Volume I, Part 3, Program Output, General Dynamics, GDC-DBB-70-002, December 1970.
9. ———., Space Shuttle Synthesis Program, SSSP, Volume II, Weights/Volume Handbook, General Dynamics GDC-DBB70-002, December 1970.
10. 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.
11. Petersen, R. H., Gregory, T. J., and Smith, C. L., "Some Comparisons of Turbojet-Powered Hypersonic Aircraft for Cruise and Boost Mission," *Journal of Aircraft*, September-October 1966.
12. Lee, V. A., et al., "Computerized Aircraft Synthesis," *Journal of Aircraft*, Volume IV, Number 5, (also General Dynamics, FZA-418), 1967.
13. Klees, G. W., Description and Operating Instructions, Boeing Engine-Airplane Matching Program, BEAMS, Boeing Document D6-20383, 1968.
14. Jensen, Soren, DAMPS: Dynamic Airplane Mission and Performance Synthesis, The Boeing Company.
15. Elliot, Irwin I., et al., Automated Procedures for Evaluating Powered Hypersonic Vehicles from Subsonic through Hypersonic Speeds, Volume I, Part 1, AFFDL-TR-68-25.

16. Elliot, Irwin I., et al., Automated Procedures for Evaluating Powered Hypersonic Vehicles from Subsonic through Hypersonic Speeds, Volume I, Part 2, AFFDL-TR-68-25, 1968.
17. Elliot, Irwin, et al., Automated Procedures for Evaluating Powered Hypersonic Vehicles from Subsonic through Hypersonic Speeds, Volume II, User's Manual, AFFDL-TR-68-25, April 1968.
18. Schoen, A. H., User's Manual for VASCOMP II, The V/STOL Aircraft Sizing and Performance Computer Program, Boeing Document D8-0375, Volume VI, October 1971.
19. Herbst, W. B. and Ross, H., Application of Computer-Aided Design Programs for the Management of Fighter Development Projects, AIAA Paper 70-364, 1970.
20. Clauss, J. S., et al., Aerospace-Vehicle Preliminary Design Synthesis Formulation, AFFDL-TR-67-109, October 1967.

Steady-State Aerodynamic Analysis References

21. Fox, M.K., et al., Investigation of Techniques to Evaluate Design Trade-Offs in Lifting Reentry Vehicles, TREND, Volume I, AFFDL-TR-66-77, October 1966.
22. Fox, M.K., et al., Investigation of Techniques to Evaluate Design Trade-Offs in Lifting Reentry Vehicles, TREND, Volume II, AFFDL-TR-66-77, October 1966.
23. Woodward, F. A., "Analysis and Design of Wing-Body Combinations at Subsonic and Supersonic Speeds," *Journal of Aircraft*, Volume V, Number 6, November-December 1968.
24. Woodward, F. A., Tinoco, E. N., and Larsen, J. W., Analysis and Design of Supersonic Wing-Body Combinations Including Flow Properties in the Near Field, Part 1, NASA CR-73106, August 1967.
25. Woodward, F. A. and Hague, D. S., A Computer Program for Aerodynamic Analysis and Design of Wing-Body-Tail Combinations at Subsonic and Supersonic Speeds, General Dynamics ERR-FW-867, February 1969.
26. Bradley, R. G. and Miller, B. D., Lifting Surface Theory-Advances and Applications, AIAA Paper 70-192, Presented at AIAA Eighth Aerospace Sciences Meeting, New York, January 19-21, 1970.
27. Hague, D. S., Estimation of Wing-Body Wave Drag, Avro Aircraft (Canada) Report GEN/THEOR.AERO/4, Parts 1 and 2, July 2, 1961, (available from Aerophysics Research Corporation library).

28. Hague, D. S., Engineering Instructions for Use of the Supersonic Area Rule Program, Avro Aircraft Report P470/THEOR.AERO/7, March 1961, (report available from Aerophysics Research Corporation library).
29. Ellison, D. E., et al., U. S. Air Force Stability and Control, DATCOM, Air Force Flight Dynamics Laboratory, October 1960 (revised).
30. Gebman, Jean R., et al., An Introduction to the Capabilities and Contents of AERSYS 1B; A Computer Program for Subsonic Airplane Aerodynamics, Rand Report WN-7373-PR, April 1971.
31. Burkholz, Gail M., User's Manual for AERSYS 1B: A Computer Program for Subsonic Airplane Aerodynamics, Rand Report WN-7405-PR, April 1971.
32. Gentry, Arvel, Hypersonic Arbitrary-Body Aerodynamic Computer Program, Volume I, Douglas Report DAC 56080, June 1967.
33. Gentry, Arvel, Hypersonic Arbitrary-Body Aerodynamic Computer Program, Volume II, Douglas Report DAC 56080, June 1967.
34. Powers, S. A. and Beeman, E. R., Flow Fields over Sharp Edged Delta Wings with Attached Shocks, NASA CR-1738, April 1971.
35. Powers, S. A., et al., A Numerical Procedure for Determining the Combined Viscid-Inviscid Flow Fields Over Generalized Three-Dimensional Bodies, AFFDL-TR-67-124, Volumes I and II, December 1967.
36. Hoerner, S. F., *Fluid Dynamic Drag*, published by the author.
37. Weissinger, J., The Lift Distribution of Swept Back Wings, NACA TM-1120, 1947.
38. Pitts, W. C., Nielsen, J. N., and Kaatari, G. E., Lift and Center of Pressure of Wing-Body-Tail Combinations at Subsonic, Transonic, and Supersonic Speeds, NACA Report 1307, 1959.

Performance References

39. Hague, D. S., Three-Degree-of-Freedom Problem Optimization Formulation, FDL-TDR-64-1, Part 1, Volume III, 1964.
40. Mobley, R. L. and Vorwald, R. F., Three-Degree-of-Freedom Problem Optimization Formulation, FDL-TDR-64-1, Part 2, Volume III, 1964.
41. Peterson, L. D., Trajectory Optimization by the Method of Steepest-Descent, AFFDL-TR-67-108, Volume I, 1967.
42. Rutowski, E. R., "Energy Approach to the General Aircraft Performance Problem," *Journal of Aeronautical Sciences*, March 1954.

43. Landgraf, S. K., "Some Applications of Performance Optimization Techniques to Aircraft," *Journal of Aircraft*, March-April 1965.
44. Hague, D. S., Application of the Variational Steepest-Descent Method to High Performance Aircraft Trajectory Optimization, NASA CR-73366, 1969.
45. Stein, L. H., et al., STOP: A Computer Program for Supersonic Transport Trajectory Optimization, NASA CR-793, May 1967.
46. 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.
47. Willwerth, Robert E., et al., PRESTO: Program for Rapid Earth to Space Trajectory Optimization, NASA CR-158, 1965.
48. Hague, D. S. and Glatt, C. R., Study of Navigation and Guidance of Launch Vehicles Having Cruise Capability - Trajectory Parametric and Optimization Studies, Boeing Doc. D2-113016-5, 1967.
49. Retka, J., Harder, D., Hague, D. S., Glatt, C. R., and Minden, D., Study of Navigation and Guidance of Launch Vehicles Having a Cruise Capability, Volume IV, Boeing Document D2-113016-7, 1967.
50. 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 library).
51. Linden, L. H., An Introduction to the Capabilities, Contents, and Structure of NSEG: A Computer Program for Aircraft Performance, Rand Report WN-7114-PR, 1970.
52. Linden, L. H., NSEG Mission Profile Coding Procedure, Rand Report WN-7139-PR, November 1970.
53. Hague, D. S., Jones, R. T., and Glatt, C. R., Combat Optimization and Analysis Program--COAP, Volume I, AFFDL-TR-71-52, May 1971.
54. Hague, D. S., Jones, R. T., and Glatt, C. R., Combat Optimization and Analysis Program--COAP, Volume II, AFFDL-TR-71-52, May 1971.
55. Hague, D. S., Jones, R. T., and Glatt, C. R., Combat Optimization and Analysis Program--COAP, Volume II, AFFDL-TR-71-52, May 1971.
56. Raybould, B., "Approximate Trajectories for Maximum Range," *Journal of the Aeronautical Society*, June 1963.

Propulsion System References

(Also see Vehicle Synthesis References, Nos. 1-20)

57. Vanco, Michael, Computer Program for Design Point Performance of Turbojet and Turbofan Engine Cycles, NASA TM X-1340, February 1967.
58. Robbin, W. H. and Dugan, J. F., Prediction of Off-Design Performance of Multi-stage Compressors, Ch. X, NASA SP-36, 1965.
59. Bastress, E. K., Statistical Derivation of Design Criteria for Liquid Rocket Combustion Instability, NASA CR-72730, December 1967.
60. Hague, D. S., Reichel, R. H., Jones, R. T., and Glatt, C. R., A Demonstration of the Automated Combustor Design Optimization Code, AUTOCOM, NASA CR-120856, December 1971.
61. Klees, G. W. and Orton, H. E., Engine Selection for Bomber Aircraft, ASME Annual Aviation and Space Conference, Beverly Hills, June 1968.
62. Gerend, R. P., A Method of Estimating Weight of Bare Cruise Turbojet and Turbofan Engines, Boeing Document D6-6134-TN, May 1963.
63. Gerend, R. P., A Method of Estimating the Dimensions of Direct Lift Turbojet and Turbofan Engines, D6-6132, May 1963.

Economics References

64. Levenson, G. S. and Barrow, S. M., Cost Estimating Relationships for Aircraft Airframes, Rand Corporation Memo RM-4845-PR, 1965.
65. Boren, H. E., DAPCA: A Computer Program for Determining Aircraft Development and Production Costs, Rand Corporation Memo RM-5221-PR, 1967.
66. Wilcox, Darrell, Computer Program for Estimating Total Program Cost of Aircraft, Spacecraft, and Reusable Launch Systems, Ames Research Center Working Paper, MA-71-2, August 1971.
67. Slye, Robert E., Budget Smoothing Model for Program Planning, Joint AAS/Operations Research Society Meeting, June 17-20, 1969, Denver, CO
68. ———., Space Planner's Guide, Headquarters, Air Force Systems Command, Andrews Air Force Base, Washington, D.C.
69. ———., Direct Operating Costs for Transport Aircraft, Air Transport Association of America, June 1960.

Structural Analysis References

70. McCormick, Caleb W., NASTRAN User's Manual, NASA SP-222, October 1969.

71. Jordan, S., MAGIC II: An Automated General Purpose System for Structural Analysis, Volume I, Engineer's Manual, AFFDL-TR-71-1, 1971.
72. Jordan, S., MAGIC II: An Automated General Purpose System for Structural Analysis, Volume II, User's Manual, AFFDL-TR-71-1, 1971.
73. Gallo, A. M., MAGIC II: An Automated General Purpose System for Structural Analysis, Volume III, Programmer's Manual, AFFDL-TR-71-1, 1971.
74. Gellatley, R. A. and Berke, Laszlo, Optimal Structural Design, AFFDL-TR-70-165, 1970.
75. ———., Structural Design Guide for Advanced Composite Applications, Air Force Materials Laboratory, January 1971.
76. Utku, Senoi and Akyuz, Fevzican, ELAS: A General Purpose Computer Program for the Equilibrium Problems of Linear Structures, Volume I, User's Manual, Jet Propulsion Laboratory CALTEC, Technical Report 32-1240, 1968.
77. Utku, Senoi, ELAS: A General Purpose Computer Program for the Equilibrium Problems of Linear Structures, Volume II, Documentation for the Program, Jet Propulsion Laboratory, CALTEC, Technical Report 32-1240, 1969.
78. Boddy, J. A., et al., Influence of Structure and Material Research on Advanced Launch Systems' Weight, Performance, and Cost, User's Manual, NASA OART Computer Program 41, NA Rockwell Company Report SD 67-1204-2, January 1968.
79. Boddy, J. A. and Mitchell, J. C., Influence of Structure and Material Research on Advanced Launch Systems Cost, Phase II Interim Report, North American Aviation, Inc., Report SID 67-542-2, August 1967.
80. Boddy, J. A. and Mitchell, J. C., Influence of Structure and Material Research on Advanced Launch Systems Cost, Phase II Interim Report, Weight and Cost Data, North American Aviation, Inc. Report SID 67-542-3, August 1967.
81. Giles, Gary L., A Procedure for Automating Aircraft Wing Structural Design, 5th National ASCE Structural Division Conference on Electronic Computation, 1970.

Flutter and Aeroelasticity References

82. Phoa, Y.T. and Watson, D.A., AFSP: An Automated Flutter Solution Procedure, Aerophysics Research Corporation Technical Note TN-109, 1971.
83. Harris, G.Z., The Calculation of Generalized Forces on Oscillating Wings in Supersonic Flow by Lifting Surface Theory, R&M 3453, 1966, Aeronautical Research Council, Great Britain.

84. Richardson, J. R., A Method for Calculating the Lifting Force on Wings (unsteady Subsonic and Supersonic Lifting Surface Theory) R&M 3157, 1960, Aeronautical Research Council, Great Britain.
85. 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.
86. White, Roland J. and Watson, David A., A Swept Strip Aeroelastic Model for the Preliminary Design of an Aircraft Wing--Program SSAM, Aerophysics Research Corporation Technical Note TN-115, 1971.
87. Timman, R. and VanDeVooren, A.I., The Oscillating Wing with Aerodynamically Balanced Control Surface in Two-Dimensional Subsonic, Compressible Flow, National Luchtvaart Laboratorium, Amsterdam, Report F.54, 1949.
88. Timman, R., VanDeVooren, A.I., and Greidanus, J.H., "Aerodynamic Coefficients of an Oscillating Airfoil in Two-Dimensional Subsonic Flow," *Journal of the Aeronautical Sciences*, December 1951.
89. Garrick, I.E. and Rubinow, S. I., Theoretical Study of Air Forces on an Oscillating or Steady Thin Wing in a Supersonic Main Stream, NACA TN-1383, July 1947.
90. Ashley, H. and Zartarian, G., "Piston Theory--A New Aerodynamic Tool for the Aeroelastician," *Journal of the Aeronautical Sciences*, December 1956.
91. Phoa, Y.T., DYNLOD: A Strip Theory Dynamic Loads Analysis Program, Aerophysics Research Corporation Technical Note TN-114, 1971.
92. Landahl, Marten T. and Stark, Walter J.E., "Numerical Lifting Surface Theory--Problems and Progress," *AIAA Journal*, November 1968.
93. Phoa, Y.T., VIBCAL: A Code for Computation of Swept Wing Bending and Torsional Vibration Modes and Frequencies, Aerophysics Research Corporation Technical Note TN-105, 1971.

Graphics and Higher Order Language Development References

94. Jones, R.T., SEAL User's Manual, Boeing Doc. D2-139592-1, 1969.
95. ———, Macro Fortran Fortran: An Algebraic Fortran String Processor for the CDC 6600, User's Manual, Aerophysics Research Corporation TN-116, 1971.
96. Glatt, C. R., PANEL: A CDC 6600 Graphics Program for Panelled Aircraft Surfaces, Aerophysics Research Corporation TN-117, 1971.

97. Sally, G.C., CALCOMP Plotting System Manual, Analysis and Computation Center, Langley Research Center, 1967.
98. Jones, R. T., An Introduction to the SPEED Code: A Synthesis Program for Efficient Element Design, Aerophysics Research Corporation Technical Note TN-113, 1971.

Optimization

(Also See Performance References and Miscellaneous References)

99. Hague, D. S. and Glatt, C. R., An Introduction to Multivariable Search Techniques for Parameter Optimization, NASA CR-73200, April 1968. (to be reissued as a low number NASA Contractor Report under NAS 2-6147).
100. Hague, D. S. and Glatt, C. R., A Guide to the Automated Engineering and Scientific Optimization Program, AESOP, NASA CR-73201, April 1968.
101. Sandrin, W. A., Glatt, C. R., And Hague, D. S., Design of Arrays with Unequal Spacing and Partially Uniform Amplitude Taper, IEEE Transactions on Antennas and Propagation, September 1969.
102. Hague, D. S. and Glatt, C. R., Computer-Aided Aerospace Vehicle Performance and Design Optimization, Joint ACM/SIAM/IEEE Conference on Mathematical and Computer Aids to Design, October 27-31, 1969, Anaheim, California.
103. Jones, R. T. and Hague, D. S., Application of Multivariable Search Techniques to Structural Design Optimization, NASA CR-2038, 1972.
104. Jacobson, D.H. and Oksman, W., An Algorithm that Minimizes Homogeneous Functions of N Variables in N+2 Iterations and Rapidly Minimizes General Functions, Division of Engineering and Applied Physics Report No. 618, Harvard University, October 1971.
105. 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, 1971.
106. ———., Transportation Planning System, TRANPLAN, Control Data Corporation Data Centers Division, July 1966.
107. ———., OPHELIE, Linear Programming User's Manual, Control Data Corporation Data Services Division, July 1969.
108. Elliott, Jarrell R., and Teague, W.F., A Method for Reducing the Sensitivity of the Optimal Non-linear Systems to Parameter Uncertainty, NASA TN D-6218, 1971.

Stability and Control References

109. Kemp, W. B. and Fox, C. H., User's Guide to Program ACMOTAN for Linear Aircraft Motion Analysis Program No. A2541, Langley Research Center.
110. 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.
111. Vorwald, R.L., Six-Degree-of-Freedom Flight Path Study Program Generalized Computer Program, Part 2, Volume I, AFFDL-TDR-64-1, October 1964.
112. Sansom, F.J., et al., MIMIC: A Digital Simulator Program, SESCO Internal Memo 65-12, Wright-Patterson Air Force Base, Ohio.

Statistical Analysis References

113. Dixon, W.J., BMD: Biomedical Computer Programs, University of California Publications in Automatic Computation No. 2., 1968.
114. ———., STAT-PACK, Program Abstracts, UNIVAC Data Processing Division, Document UP4041.
115. Browne, et al., Flight Path Error and Dispersion Analysis Generalized Computer Program, ASD Technical Report 61-552, Parts 1 and 2, USAF Flight Dynamics Laboratory, 1961.

Program Module Intercommunication References

116. Glatt, C. R. and Watson, D.A., An Introduction to the DIALOG System: A Concept of Multiple Program Execution with Mutual Data Intercommunication, Aerophysics Research Corporation Technical Note TN-108, 1971.

Automatic Programmed Tooling References

117. ———., APT Reference Manual, Control Data Corporation Publication 60174500, 1966.

Management Control References

118. ———., PERT General Information Manual, Control Data Corporation Publication No. 60133300, August 1965.
119. ———., PERT/TIME Reference Manual, Control Data Corporation Publication No. 60133600, May 1966.

120. ———, PERT/COST Reference Manual, Control Data Corporation
Publication No. 60214800, October 1967.

Miscellaneous References

121. Wagner, H., Uber Die Entstehung des Dynamischen Auftriebes von
Tragflugeln, Z. angew. Math. Mech., Bd. 5, Heft 1, February 1925.
122. Gellatley, R. A., et al., Development of a Procedure for Automated
Synthesis of Minimum Weight Structures, AFFDL-TDR-64-141, October
1964.
123. ———, Military Specification: Charts; Standard Aircraft Charac-
teristics and Performance, Piloted Aircraft, Mil. Spec. MIL-C-5011A.
124. Hague, D. S. and Glatt, C. R., Application of Multivariable Search
Techniques to the Optimal Design of a Hypersonic Cruise Vehicle,
NASA CR-73202, 1968.
125. Ward, G.N., The Drag of Source Distributions in Linearized Super-
sonic Flow, College of Aeronautics Report 88, February 1955.
126. 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, 1956.
127. Einton, E., On the Minimization and Numerical Evaluation of Wave
Drag, R.A.E. Report Aero 2564, November 1955.
128. Harris, R.V. and Craidon, Charlotte B., Airplane Drag at Zero Lift,
Langley Research Center Program D1280.
129. Harris, R.V. and Craidon, Charlotte B., Airplane Turbulent Skin
Friction Drag, Langley Research Center Program D1266.
130. White, Roland J., "Improving the Airplane Efficiency by Use of Wing
Maneuver Load Alleviation," *Journal of Aircraft*, October 1971.

2.1 STRUCTURE OF THE ODIN/MFV SYSTEM

The components of the ODIN/MFV system are illustrated in Figure 2.1-1; each system component exists in the form of one or more independent computer programs. 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/MFV 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/MFV system comprises more than a 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/MFV *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 final element of the ODIN/MFV 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.

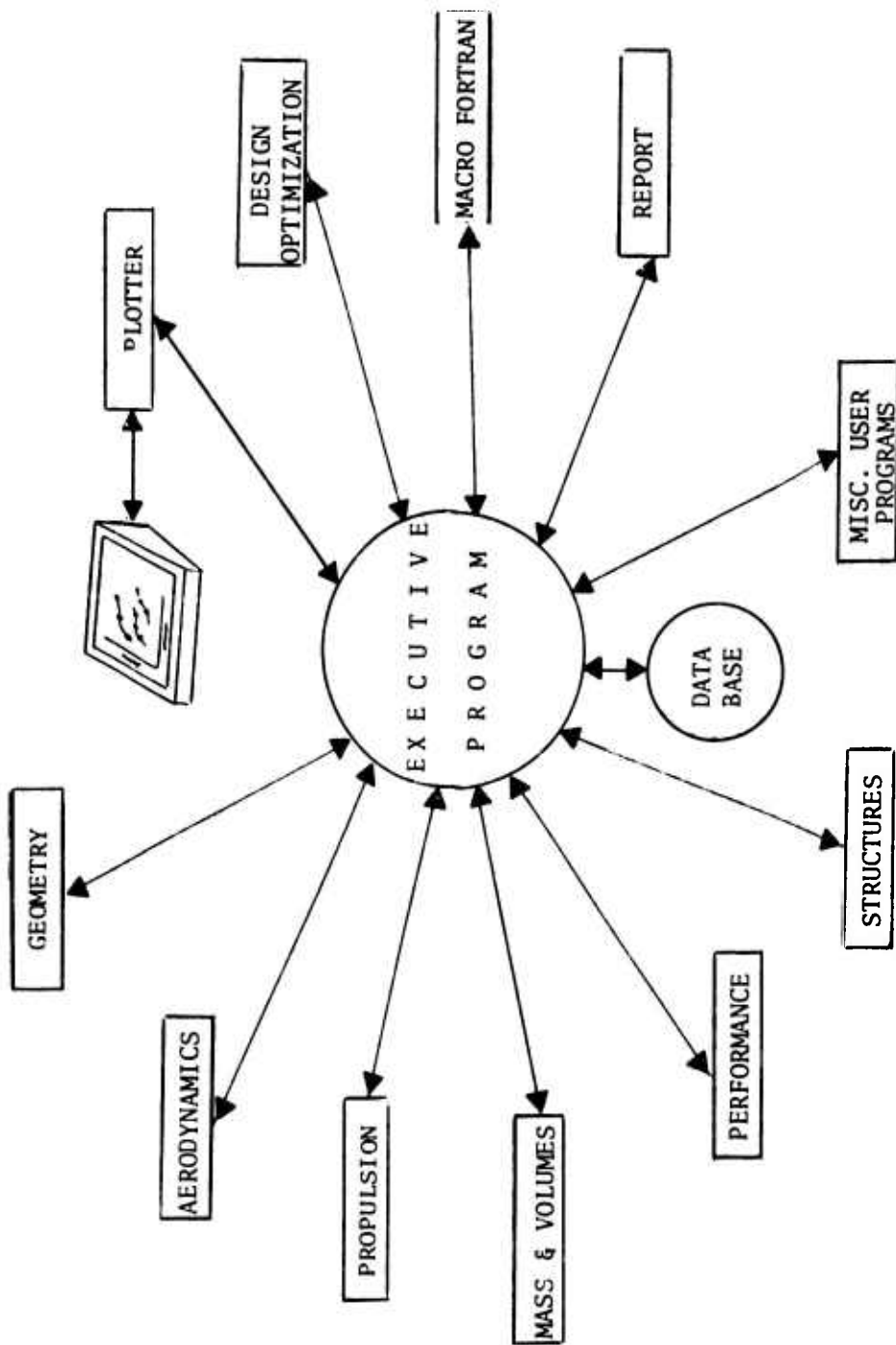


FIGURE 2.1-1. ODIN/MFV SYSTEM

2.2 THE BASIC ODIN/MFV PROGRAM ELEMENTS

The independent program elements which form a basis for the ODIN/MFV system are written in Fortran. Each program in the system has been assigned a four to six letter mnemonic for reference purposes and for operation in the ODIN/MFV system. Table 2.2-1 presents a list of the basic ODIN/MFV program library and the mnemonics assigned. When constructing the sequence of analyses which lead to the synthesis and optimization of a military flight vehicle, each program must be referred to by its mnemonic code in the ODIN/MFV system. Mnemonic control of the elements in the ODIN/MFV program library is discussed in more detail in later sections.

TABLE 2.2-1. MNEMONICS FOR THE BASIC SET OF
ODIN/MFV INDEPENDENT PROGRAM LIBRARY

| <u>PROGRAM</u> | <u>MNEMONIC</u> | <u>TECHNOLOGY AREA</u> |
|--|-----------------|------------------------|
| Executive Control Program | DIALOG | 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 | LRCFS | 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 |
| Development & Production Costs of Aircraft | DAPCA | Economics |
| Program for Improved Cost Optimization | PRICE | Economics |
| Swept Strip Aeroelastic Model | SSAM | Structures |
| Design & Off-Design Perfor- mance for Two- and Three- Spool Turbofans with as Many as Three Nozzles | GENENG II | Propulsion |
| Automated Engineering & Analysis | AESOP | Optimization |
| 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/MFV

The ODIN/MFV can be installed on any CDC 6600 computer which has an operating system containing the Appendix 1-b system utility routine CCLINK. These two versions of CCLINK are available in the basic ODIN/MFV library. Since the ODIN/MFV 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/MFV 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/MFV program library. Basic data for each program element must be set up in the usual manner for that program operating independently of the ODIN/MFV. 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 relevant 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 final 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/MFV. It insures that the analysis 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/MFV's system size, of doubtful validity.

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/MFV executive program is outlined in Section 2.4.

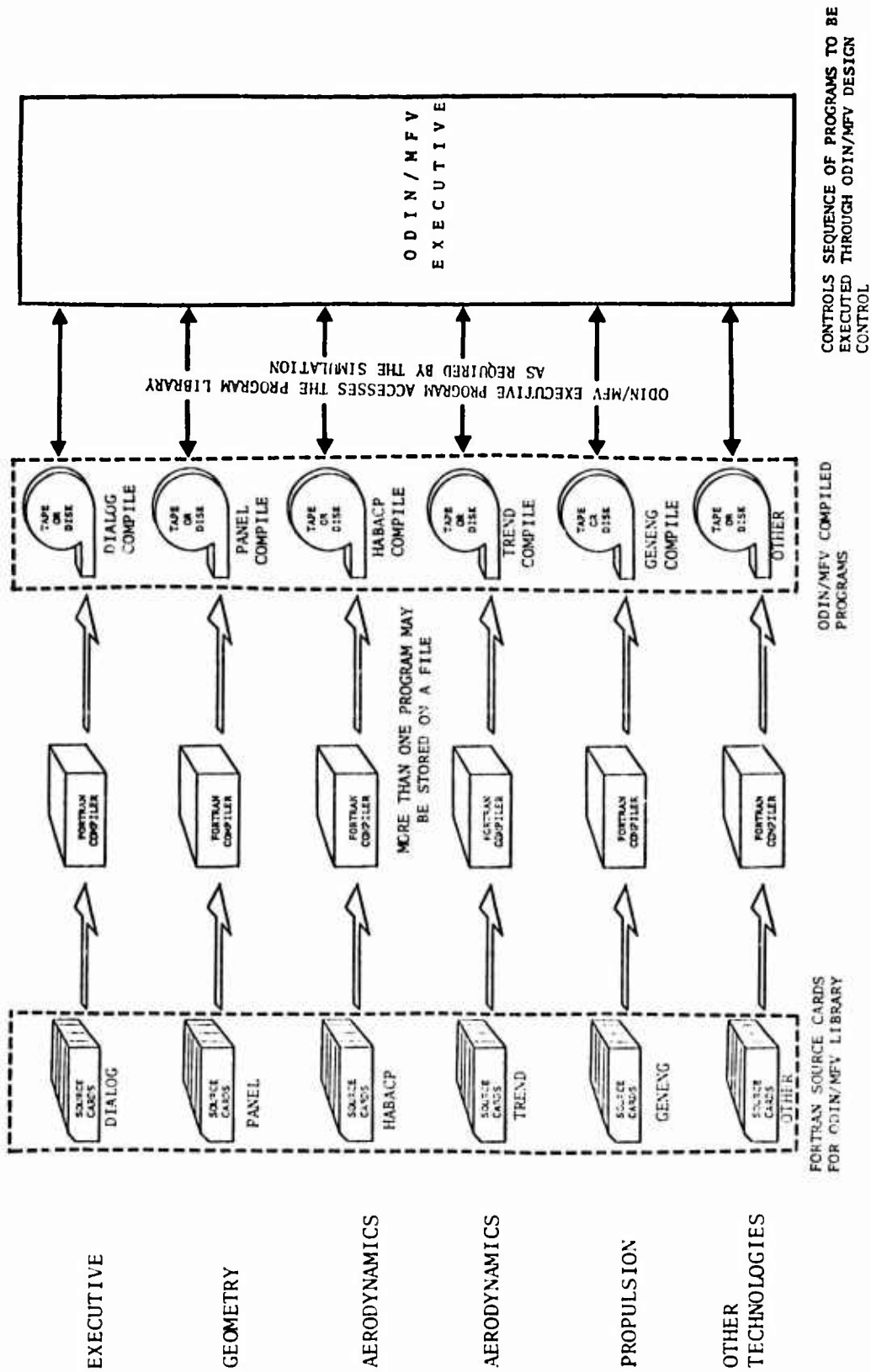


FIGURE 2.3-1 ASSEMBLY OF THE ODIN/MFV PROGRAM SYSTEM

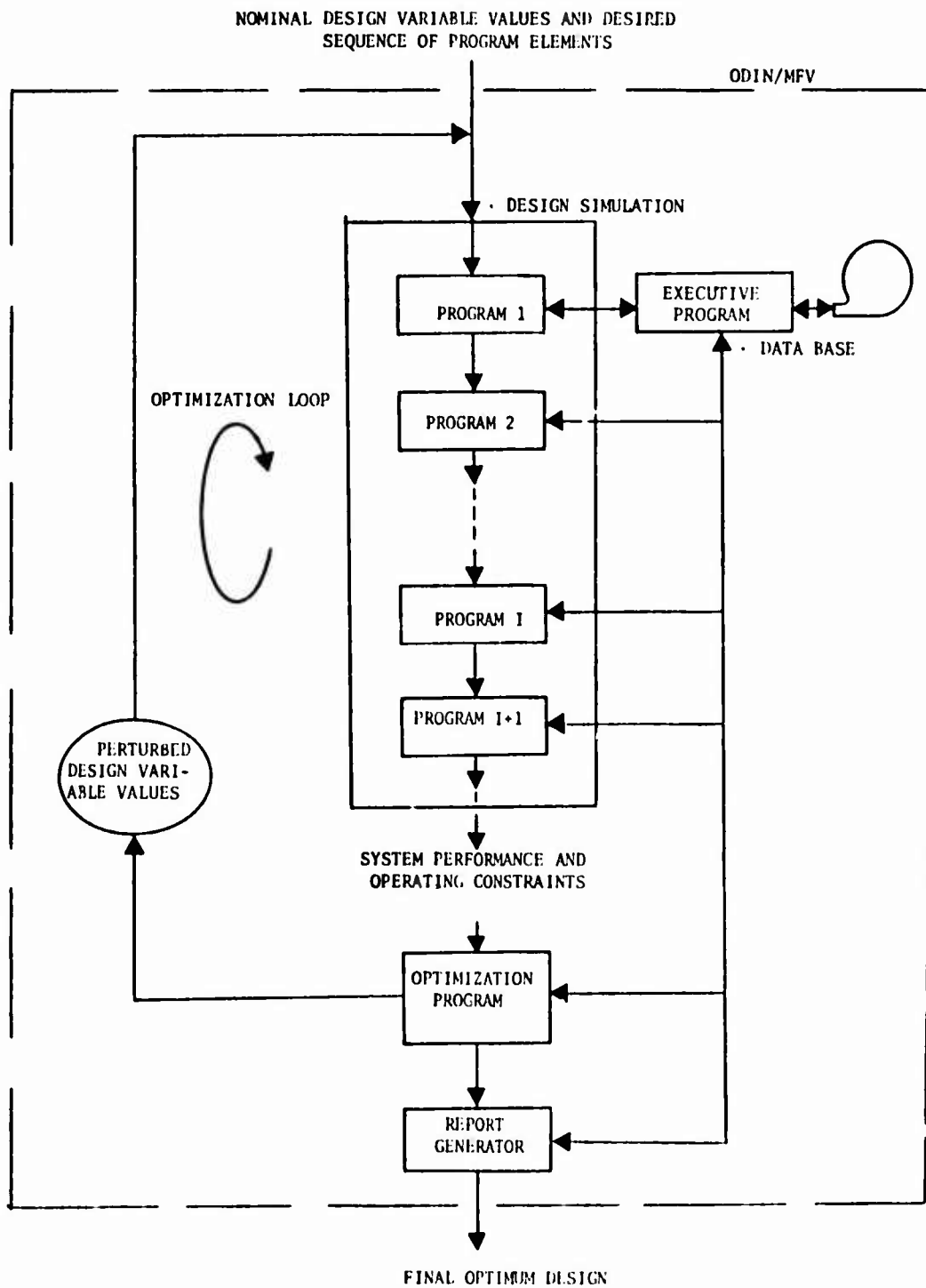


FIGURE 2.3-2 SCHEMATIC OF A VEHICLE DESIGN OPTIMIZATION SIMULATION

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/MFV 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/MFV executive program, Figure 2.4-1. This analogy forms the basis of the ODIN/MFV:

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

In this sense, the ODIN/MFV 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, $JCL_{(A+B)}^E$.

In general using this notation

$$JCL_{(A+B+\dots+N)}^E = JCL_A^E + JCL_B^E + \dots + JCL_N^E$$

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

This factor forms the basis of the ODIN/MFV; however, in the ODIN/MFV 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/MFV Job Control Language which employs comments such as

$$JCL_A^E \equiv \text{EXECUTE A}$$

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 equation must be performed:

$$\begin{aligned} \sum_{i=1, M} (JCL_{(A+B+\dots+N)}^E)_i &= JCL_A^E + JCL_B^E + \dots + JCL_N^E \\ &+ JCL_A^E + JCL_B^E + \dots + JCL_N^E \\ &\dots \dots \dots \\ &+ JCL_A^E + JCL_B^E + \dots + JCL_N^E \end{aligned}$$

where M rows of JCL are to be represented on the left hand side. This capability has been achieved by creating the ability to loop through the ODIN/MFV 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

```

DESIGN POINT I
-----
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.

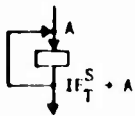
2.4.2 Topology of the General Design Synthesis Calculation

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/MFV. Instead, the analysis sequence to be performed is defined by the ODIN/MFV 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.

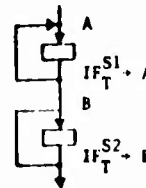
Any number of loops can be created using the LOOP and conditional IF control cards and the associated DESIGN control card. Using the symbolic notation

$$IF_T^S \rightarrow A$$

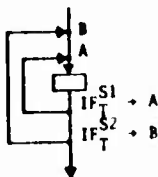
to indicate if the statement S is true go to A, it is apparent that series loops, nested iterative loops, and combined series and nested loops can be constructed. For example:



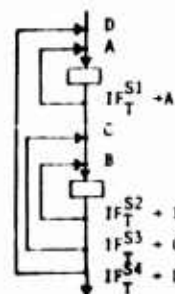
a. SINGLE LOOP



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/MFV 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/MFV 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/MFV 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.

The data base file of information is dynamically constructed by the executive program as the ODIN/MFV 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/MFV 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/MFV approximately 9000 values per second are being retrieved on the CXC 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

2.4.5 Modifying Program Input to Communicate with the Data Base

Development of the ODIN/MFV 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/MFV executive program but 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 prepares automatically 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/MFV 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 accommodated by the general input modification.

Examples:

A. NAMELIST

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

B. RIGID FORMAT

| | | | | | |
|----|--------|-------|----------|-------|--|
| 3 | 'IBAR' | 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/MFV 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.

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/MFV 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/MFV 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/MFV 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 7. 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/MFV 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.

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

| | | |
|--|---|---|
| <p>CDC 6600</p> <pre> MFI,60000. FTM,OPT=0. LGO. 789 {SOURCE DECK} 789 {DATA DECK} 6789 </pre> | <p>IBM 360/67</p> <pre> // EXEC FORTGG //FORT.SYSIN DD * {SOURCE DECK} /* //GO.SYSIN DD * {DATA DECK} /* </pre> | <p>UNIVAC 1108</p> <pre> ▼ F-E MAIN {SOURCE DECK} ▼ XDT MAIN {DATA DECK} ▼ FTN </pre> |
|--|---|---|

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

```

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

```

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

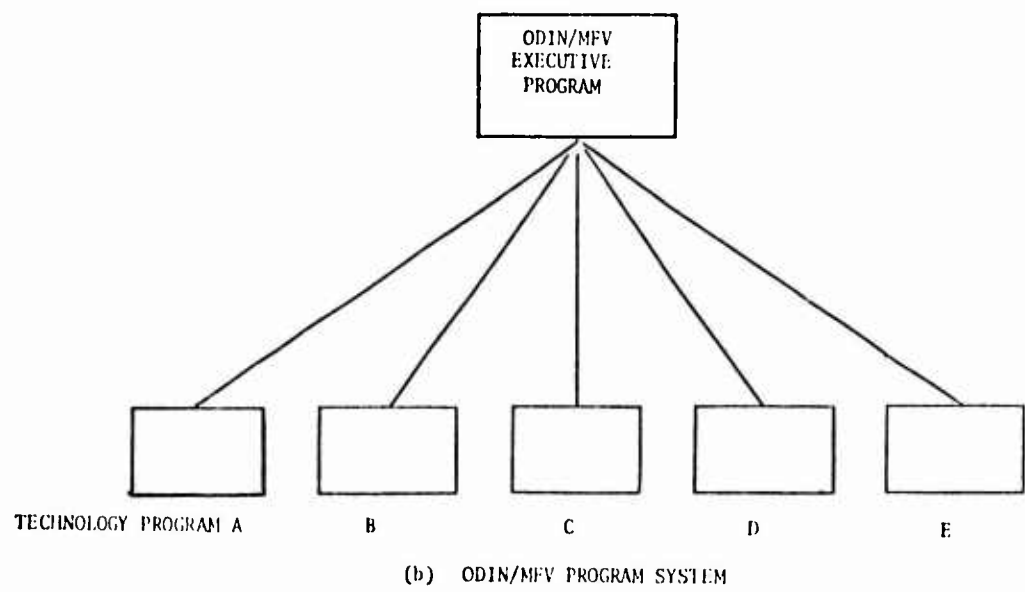
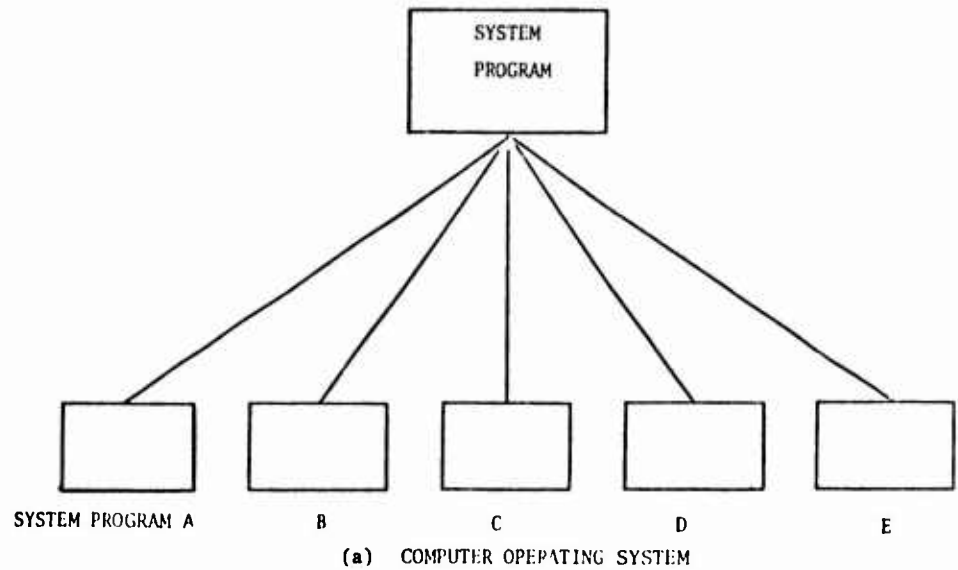


FIGURE 2.4 -1. ANALOGY BETWEEN OPERATING SYSTEM AND ODIN/MFV SYSTEM

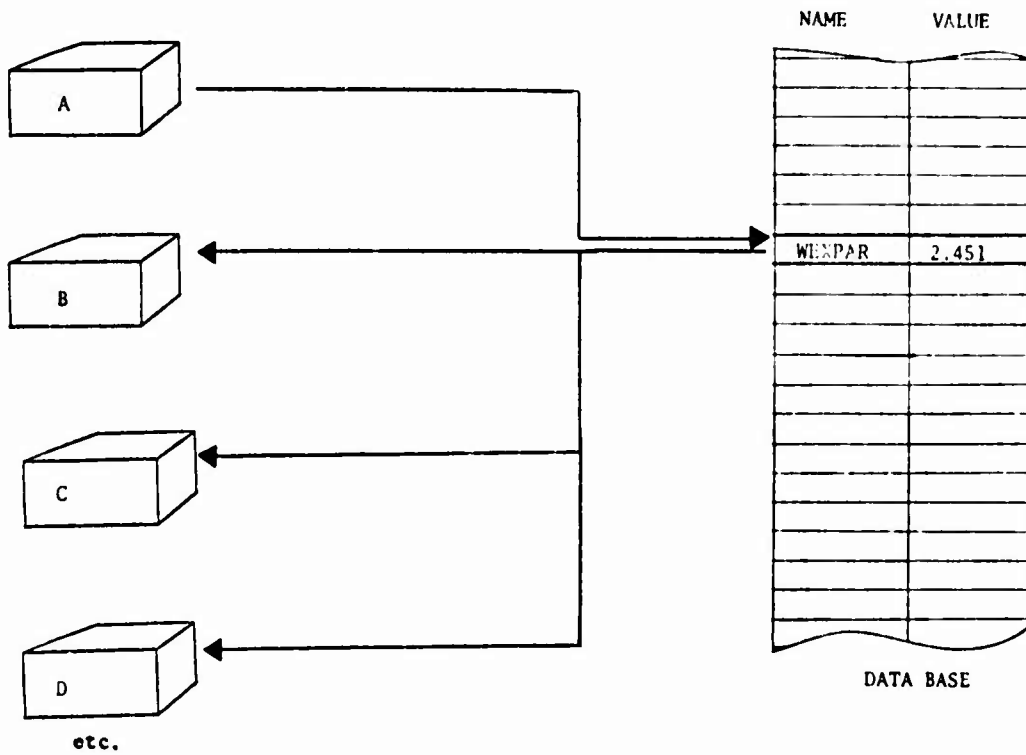


FIGURE 2.4-2. PROGRAM ACCESS TO DATA BASE

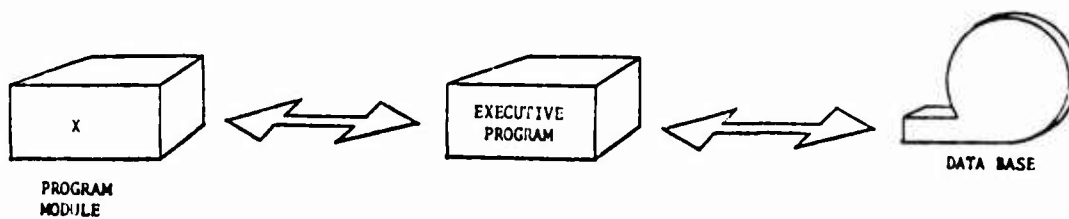


FIGURE 2.4-3. EXECUTIVE PROGRAM CONTROLS ACCESS TO DATA BASE

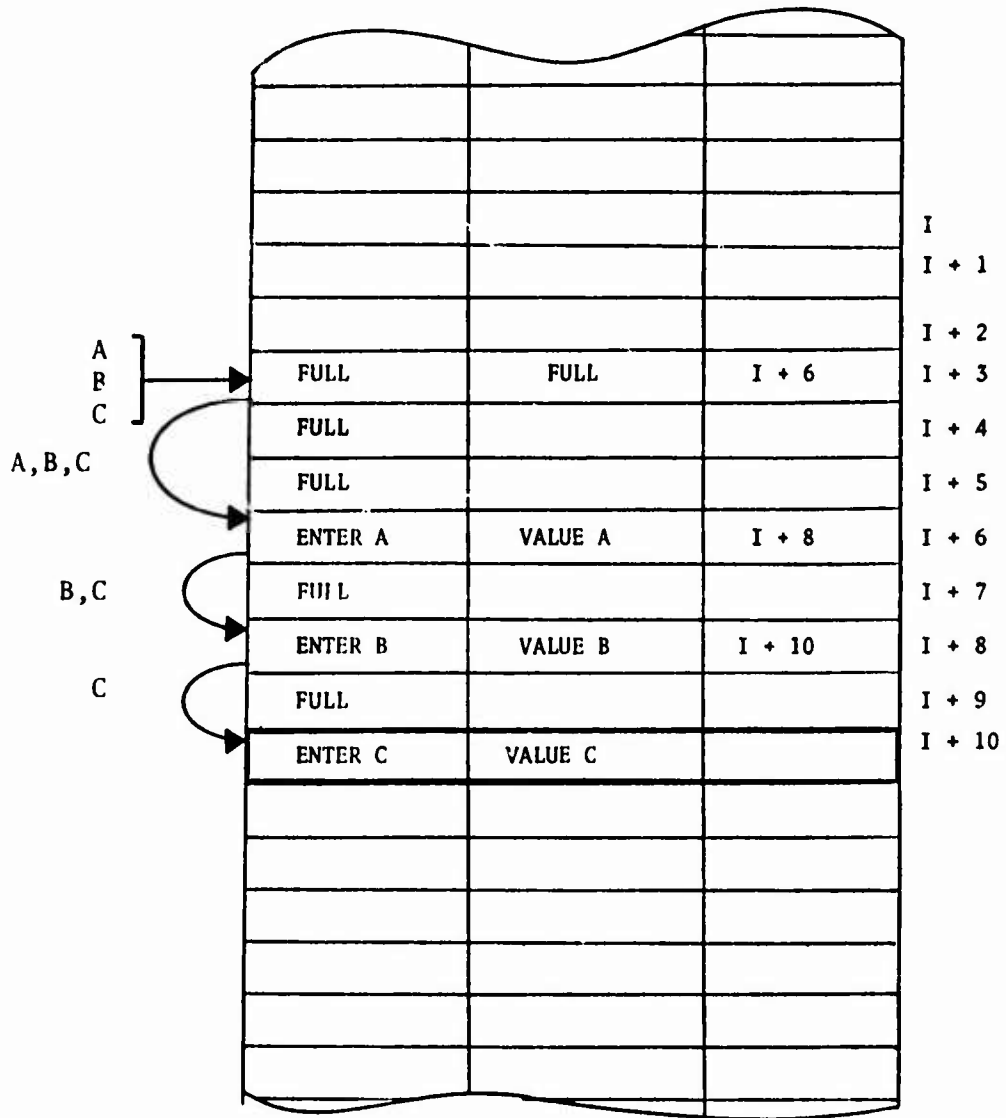


FIGURE 2.4-4. IDEALIZED DATA BASE INFORMATION RETRIEVAL SYSTEM

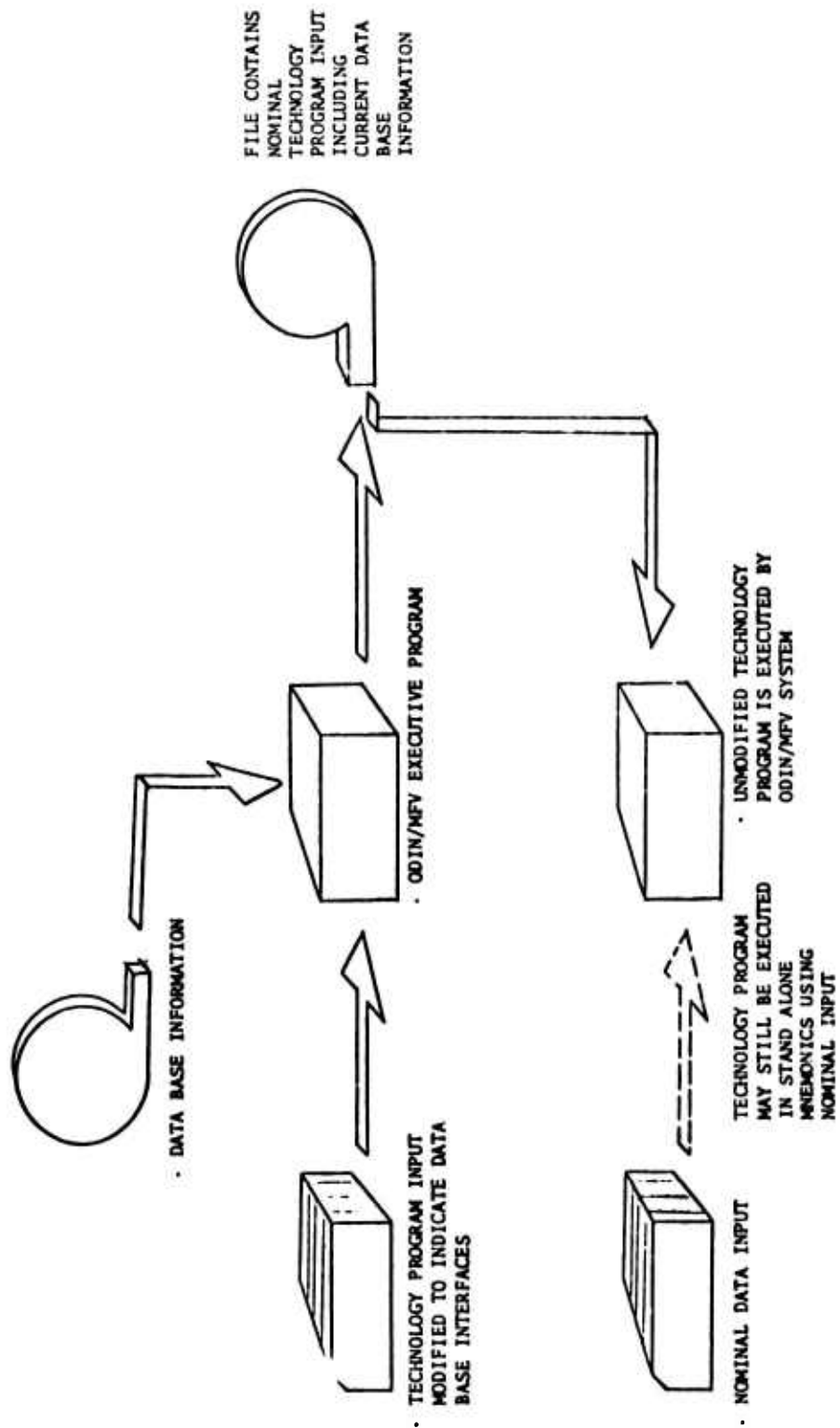


FIGURE 2.4-5 METHOD FOR MODIFYING INPUT DATA WITHOUT PROGRAM MODIFICATION

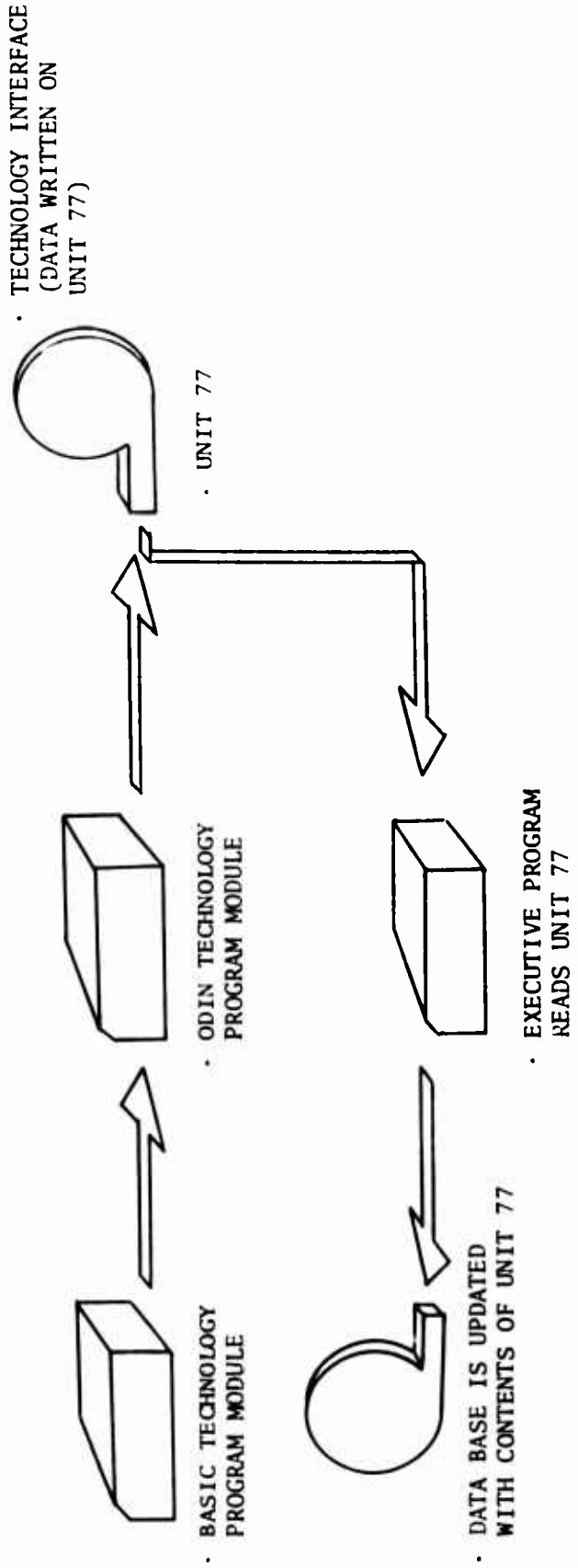


FIGURE 2.4-6. SCHEMATIC OF OUTPUT OF INFORMATION TO DATA BASE

2.5 SUMMARY OF THE ODIN/MFV SYSTEM

The ODIN/MFV system provides a design team with the following capabilities:

1. A basic program library of technology programs for analysis of military flight 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 military flight vehicle characteristics using the program library including computational loops
5. An automated military flight vehicle optimization capability

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

Description of the ODIN/MFV program elements presented by technology area follows.

SECTION 3

GEOMETRY

The ODIN/MFV 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/MFV 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 hypersonic aerodynamics program of the References 1 and 2 geometry package. The third program is a Langley Research Center developed plotting package which interfaces with the aerodynamics programs of References

Construction of separate programs for the geometry definition and graphical displays provides a generalized vehicle geometric definition and graphical display available to all technologies.

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/MFV 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 a military flight 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 consistent 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/MFV technology programs in a unified manner by consistent scaling through the data base.

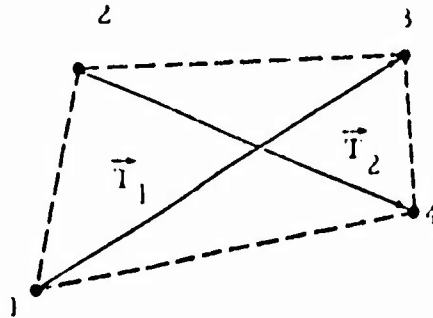
The surface defining mathematics of Reference 1 are straightforward but time consuming for hand calculation. However, the required calculations are rapidly performed on a large scale digital computer.

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

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

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

$$\begin{aligned} T_{1x} &= x_3^i - x_1^i & T_{1y} &= y_3^i - y_1^i & T_{1z} &= z_3^i - z_1^i \\ T_{2x} &= x_4^i - x_2^i & T_{2y} &= y_4^i - y_2^i & T_{2z} &= z_4^i - z_2^i \end{aligned} \quad (3.1.2)$$

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

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

The components of \vec{n} are

$$\begin{aligned} n_x &= (T_{2y} T_{1z} - T_{1y} T_{2z}) / |N| \\ n_y &= (T_{1x} T_{2z} - T_{2x} T_{1z}) / |N| \\ n_z &= (T_{2x} T_{1y} - T_{1x} T_{2y}) / |N| \end{aligned} \quad (3.1.4)$$

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.

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

The original panel defining corner points are now projected parallel to \vec{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

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

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 \vec{n} and \vec{t}_1 is defined by $\vec{t}_2 = \vec{n} \times \vec{t}_1$. The vector \vec{t}_1 defines a local x or ξ axis; \vec{t}_2 defines the y or η axis, and \vec{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

$$\begin{aligned} a_{11} &= t_{1x} & a_{12} &= t_{1y} & a_{13} &= t_{1z} \\ a_{21} &= t_{2x} & a_{22} &= t_{2y} & a_{23} &= t_{2z} \\ a_{31} &= n_x & a_{32} &= n_y & a_{33} &= n_z \end{aligned} \quad (3.1.7)$$

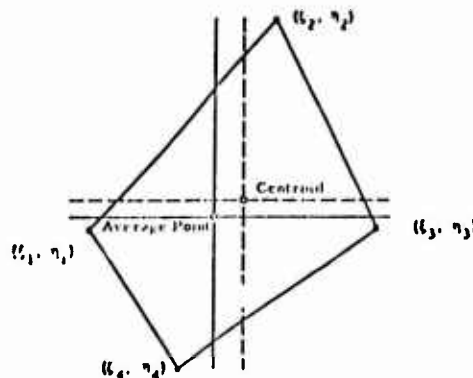
The transformation matrix is

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

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_0, y_0, z_0 . Then, if a point has coordinates x', y', z' in the reference coordinate system and coordinates x, y, z in the element coordinate system,

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = [A] \begin{bmatrix} x' - x_0 \\ y' - y_0 \\ z' - z_0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = [A] \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \quad (3.1.9)$$

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 $\xi_k^*, \eta_k^*, 0$. 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.



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:

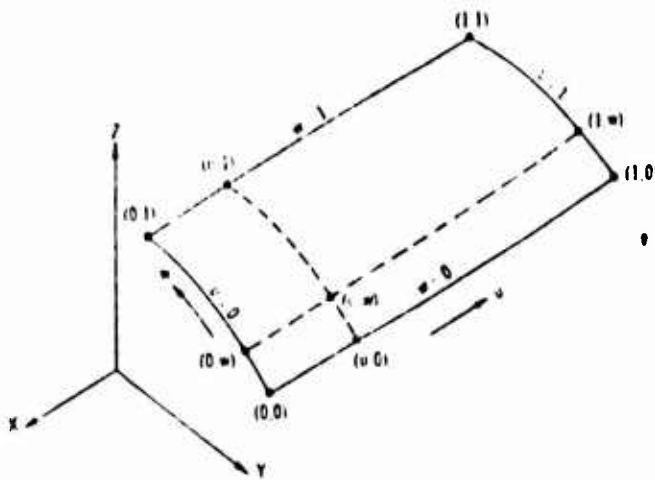
1. Points along a row are input sequentially upward
2. Rows of points are input sequentially to the right

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.

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 $u = 0$)

$$X(u, w) = Aw^3 + Bw^2 + Cw + D$$

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

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

$$C = \frac{\partial X}{\partial w}(0,0) \quad D = X(0,0)$$

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

BLENDED FUNCTIONS

$$F_1(u) = 3u^2 - 2u^3 \quad F_2(w) = 3w^2 - 2w^3$$

$$F_3(u) = 1 - F_1(u) \quad F_4(w) = 1 - F_2(w)$$

SURFACE FORM

$$\begin{aligned} X(u, w) = & X(0, w)F_3(u) + X(1, w)F_1(u) + X(u, 0)F_4(w) \\ & + X(u, 1)F_2(w) - X(0, 0)F_3(u)F_4(w) \\ & - X(0, 1)F_3(u)F_2(w) - X(1, 0)F_1(u)F_4(w) \\ & - X(1, 1)F_1(u)F_2(w) \end{aligned}$$

The basic surface-fit equations and their derivatives are presented in Reference 3 and are outlined in the diagram above.

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_0 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. Gentry, Arvel, Hypersonic Arbitrary Body Aerodynamic Computer Program, Douglas Report DAC 56080, June 1967.
2. Glatt, C. R., PANEL: A Computer Code for Generating Panelled Aerospace Vehicle Surfaces, TN-117, Aerophysics Research Corporation, October 1971.
3. Coons, Steven A., Surfaces for Computer-Aided Design of Space Forms, MAC-TR-41, Massachusetts Institute of Technology, June 1967.

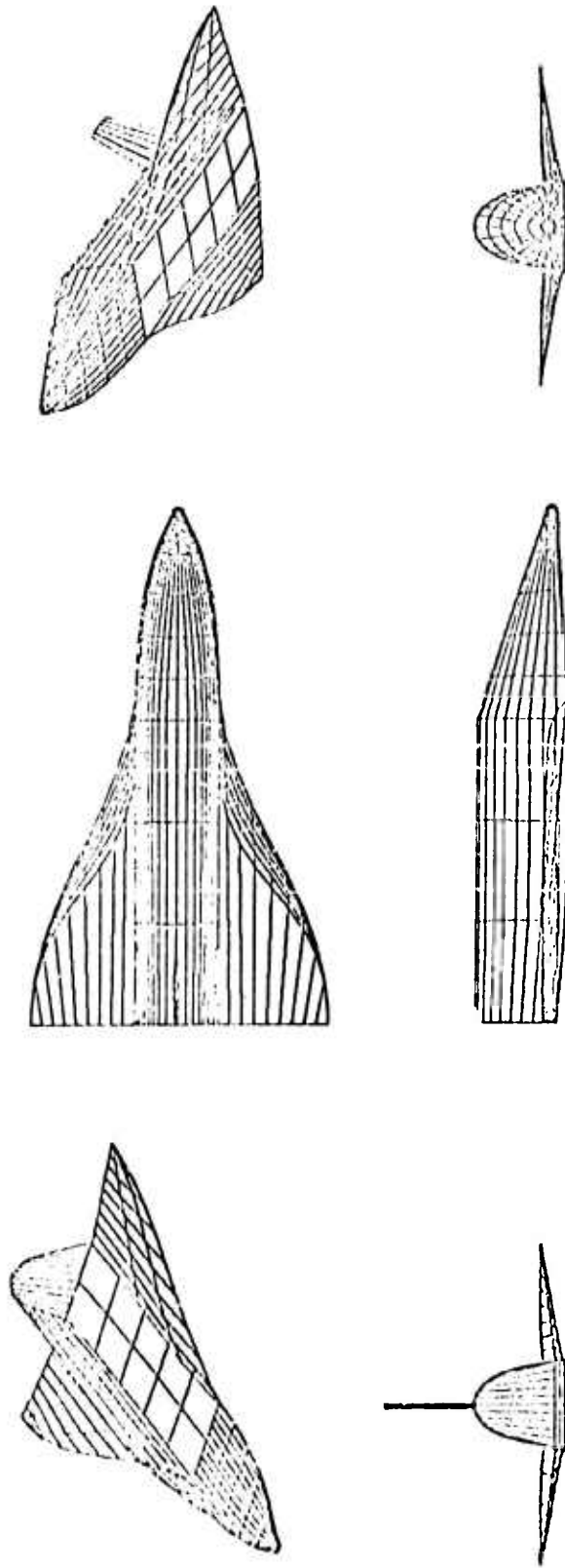


FIGURE 3.1-1. TYPICAL QUADRILATERAL ELEMENT REPRESENTATION OF A VEHICLE SURFACE IN PROGRAM PANEL

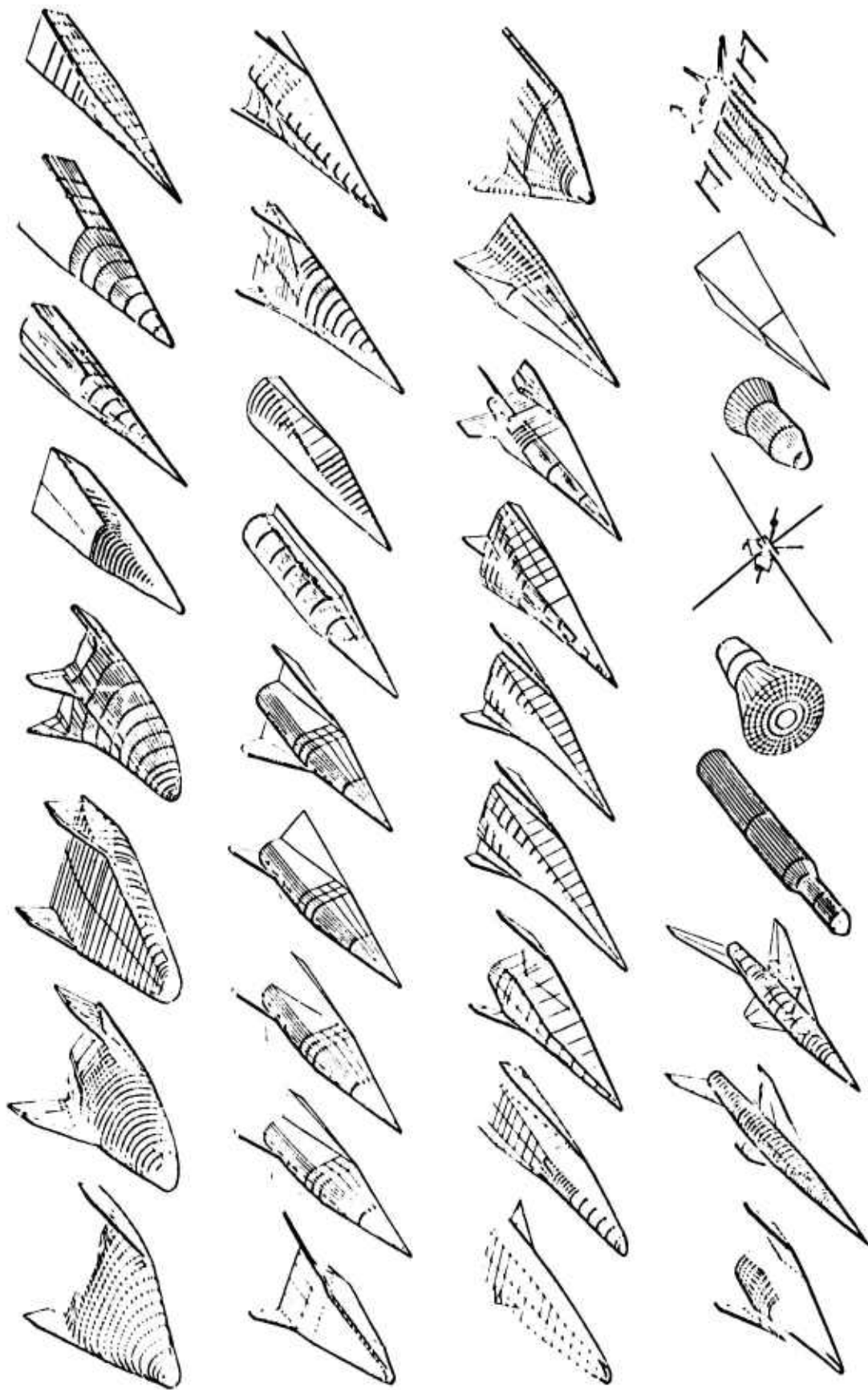
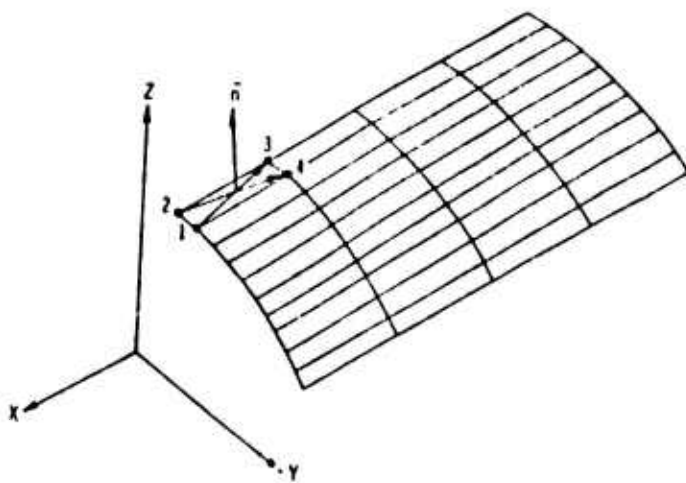


FIGURE 3.1-2 A SELECTION OF CONFIGURATIONS THAT HAVE BEEN STUDIED WITH THE HYPERSONIC ARBITRARY-BODY PROGRAM



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

$$\begin{aligned} \bar{T}_{1x} &= X_3 - X_1 & \bar{T}_{1y} &= Y_3 - Y_1 & \bar{T}_{1z} &= Z_3 - Z_1 \\ \bar{T}_{2x} &= X_4 - X_2 & \bar{T}_{2y} &= Y_4 - Y_2 & \bar{T}_{2z} &= Z_4 - Z_2 \end{aligned}$$

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

$$N_x = \bar{T}_{2y}\bar{T}_{1z} - \bar{T}_{1y}\bar{T}_{2z} \quad n_x = N_x / N$$

$$N_y = \bar{T}_{1x}\bar{T}_{2z} - \bar{T}_{2x}\bar{T}_{1z} \quad n_y = N_y / N$$

$$N_z = \bar{T}_{2x}\bar{T}_{1y} - \bar{T}_{1x}\bar{T}_{2y} \quad n_z = N_z / N$$

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

AVERAGE POINT

$$\bar{x} = 1/4(X_1 + X_2 + X_3 + X_4)$$

$$\bar{y} = 1/4(Y_1 + Y_2 + Y_3 + Y_4)$$

$$\bar{z} = 1/4(Z_1 + Z_2 + Z_3 + Z_4)$$

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

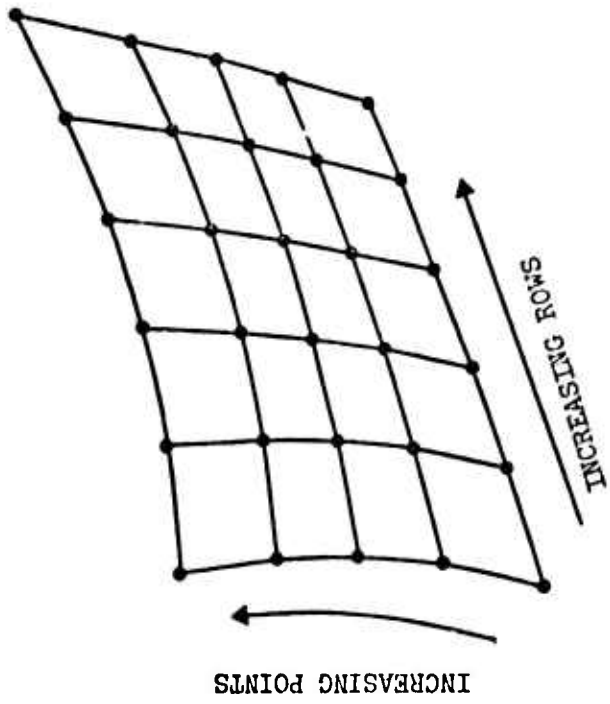
$$X_k = X_k + n_x d_k$$

$$Y_k = Y_k + n_y d_k$$

$$Z_k = Z_k + n_z d_k$$

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

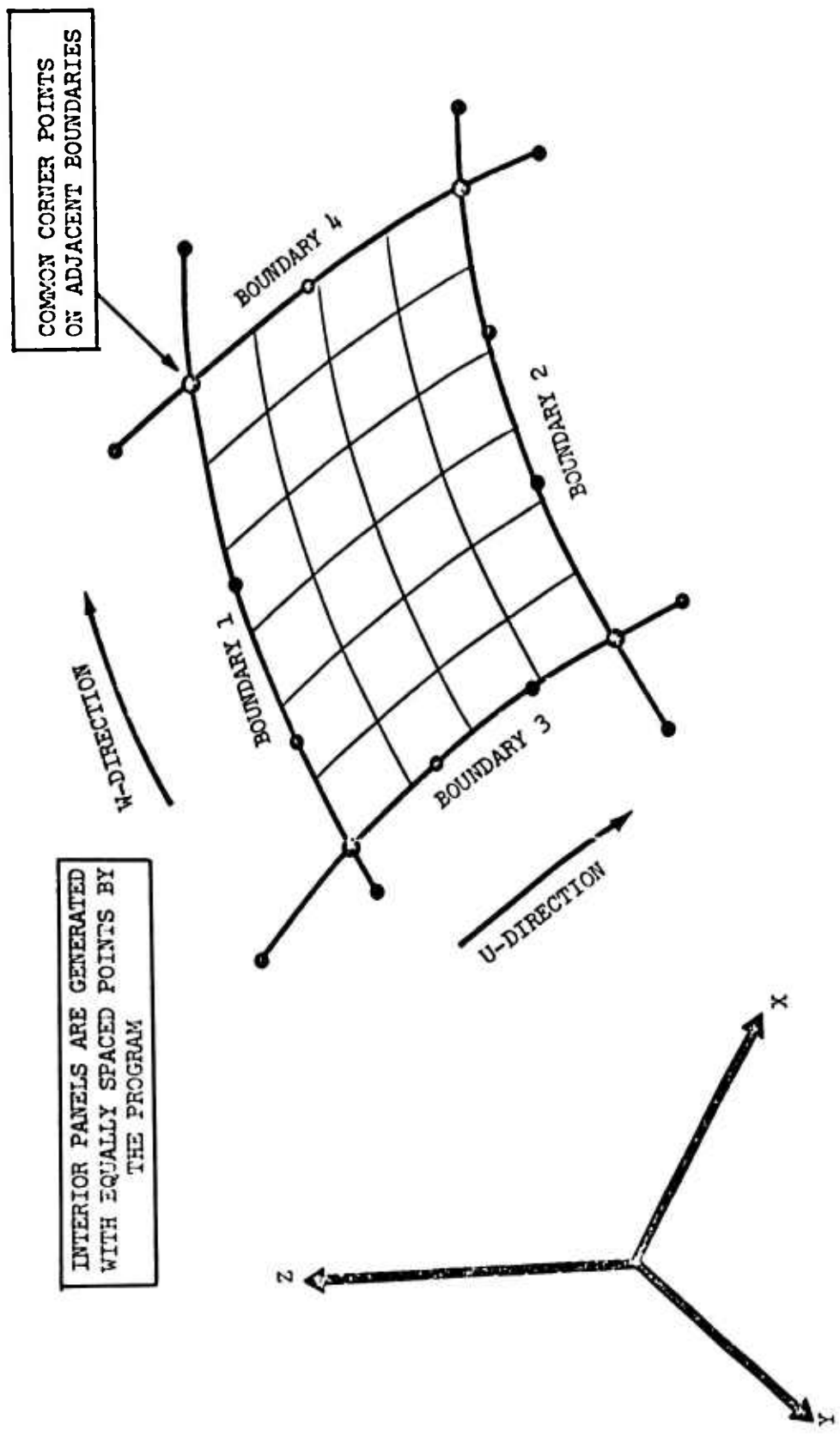


FIGURE 3.1-5. INPUT DESCRIPTION FOR THE CUBIC PATCH TECHNIQUE

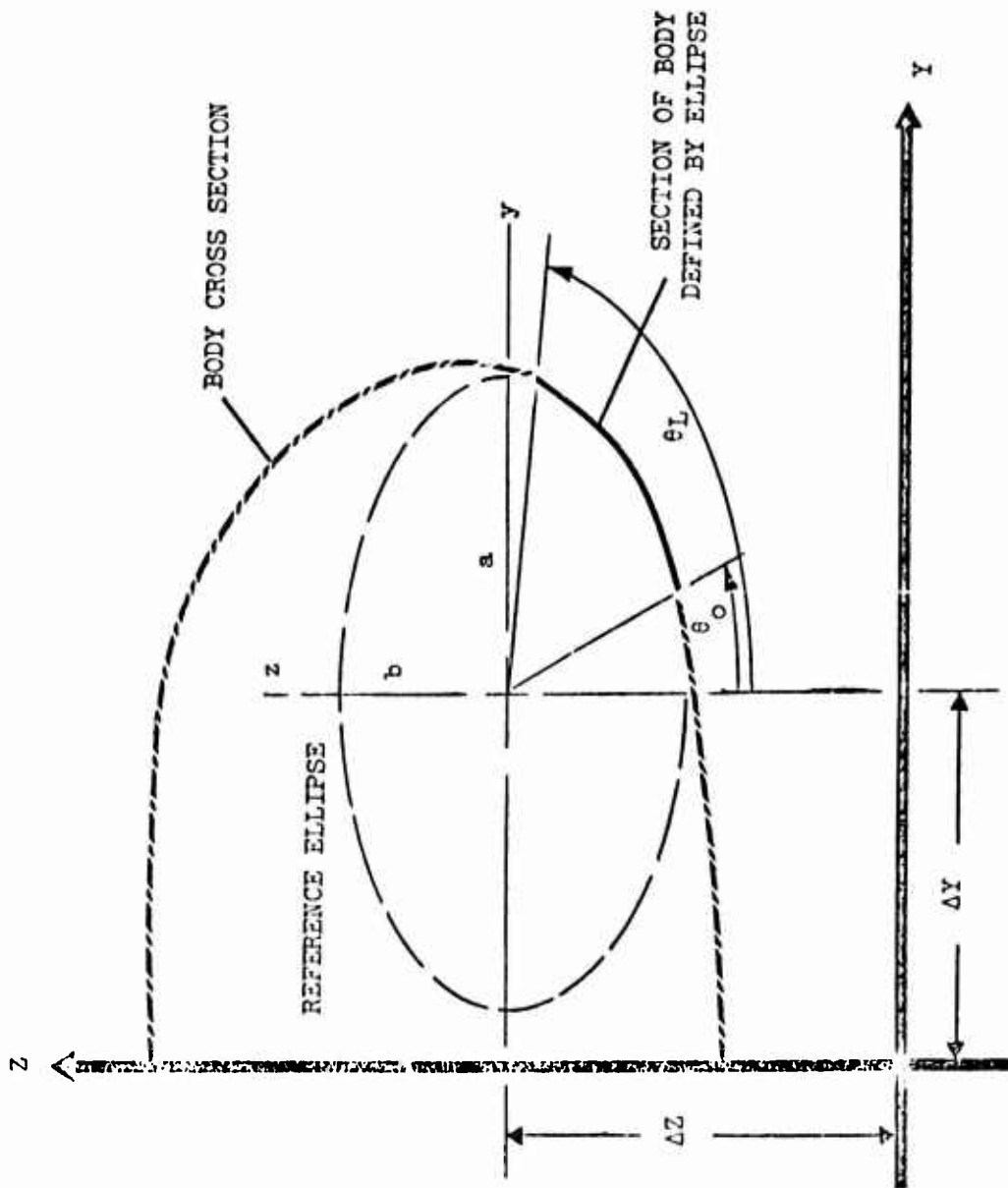


FIGURE 3.1-7. GEOMETRY DEFINITION FOR ELLIPTIC METHOD

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, COMPLIT, 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 corner 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.

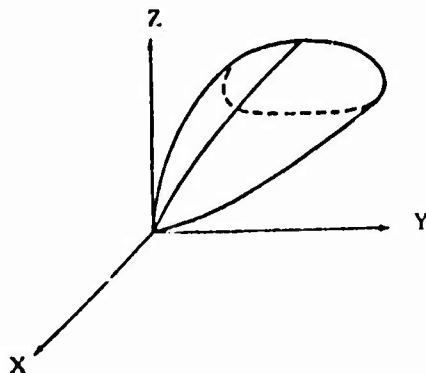
It should be noted that program IMAGE is based on the graphics package of the Reference 1 program. The program has been extended to incorporate display capability on CALCOMP, COMPLIT, and certain cathode ray tube displays at installations which have the necessary software and hardware for these devices.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

3.2.1 Method for Obtaining an Image

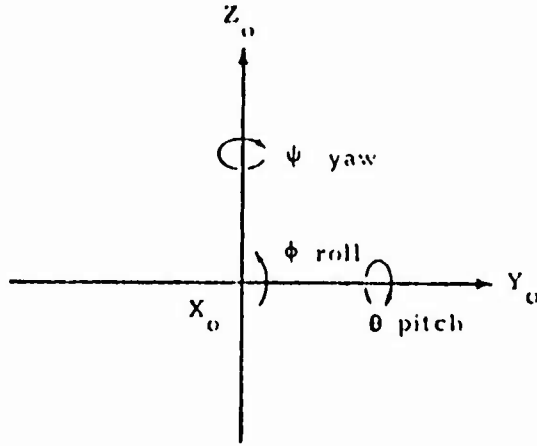
Each point on the surface is described by its coordinates in the body reference coordinate system.

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

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 \$(\psi, \theta, \phi)\$. This rotation is given by the following relationship:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix} \begin{bmatrix} X_o \\ Y_o \\ Z_o \end{bmatrix} \quad (3.2.1)$$

Where the rotation matrices are

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

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

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

or

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = [E] \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} \quad (3.2.5)$$

where

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

Since each point on the surface is given by its coordinates in the X, Y, Z system, its position in the fixed coordinate system (X₀, Y₀, Z₀) may be found by inverting the above process.

$$\begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} = [E]^{-1} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (3.2.7)$$

Carrying out this operation

$$\begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\psi & -\sin\psi\cos\phi+\sin\theta\cos\psi\sin\phi & \sin\psi\sin\phi+\sin\theta\cos\psi\cos\phi \\ \cos\theta\sin\psi & \cos\psi\cos\phi+\sin\theta\sin\psi\sin\phi & -\cos\psi\sin\phi+\sin\theta\sin\psi\cos\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (3.2.8)$$

or

$$X_0 = X(\cos\theta\cos\psi) + Y(-\sin\psi\cos\phi+\sin\theta\cos\psi\sin\phi) + Z(\sin\psi\sin\phi+\sin\theta\cos\psi\cos\phi) \quad (3.2.9)$$

$$Y_0 = X(\cos\theta\sin\psi) + Y(\cos\psi\cos\phi+\sin\theta\sin\psi\sin\phi) + Z(-\cos\psi\sin\phi+\sin\theta\sin\psi\cos\phi) \quad (3.2.10)$$

$$Z_0 = X(-\sin\theta) + Y(\cos\theta\sin\phi) + Z(\cos\theta\cos\phi) \quad (3.2.11)$$

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 Y₀, Z₀ system). With the appropriate SC4020, CALCOMP, COMPLIT, 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 X₀ direction (out of the plane of the paper) may be found from the following equation.

$$n_{x_0} = n_x(\cos\theta\cos\psi) + n_y(-\sin\psi\cos\phi + \sin\theta\cos\psi\sin\phi) + n_z(\sin\psi\sin\phi + \sin\theta\cos\psi\cos\phi) \quad (3.2.12)$$

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

If n_{x_0} is positive, then the surface element is facing the viewer. If n_{x_0} 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. Typical images obtained from the program have been presented previously in Figures 3.1-1 and 3.1-2.

REFERENCES:

1. Gentry, Arvel, Hypersonic Arbitrary Body Aerodynamic Computer Program, Douglas Report DAC 56080, June 1967.

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:

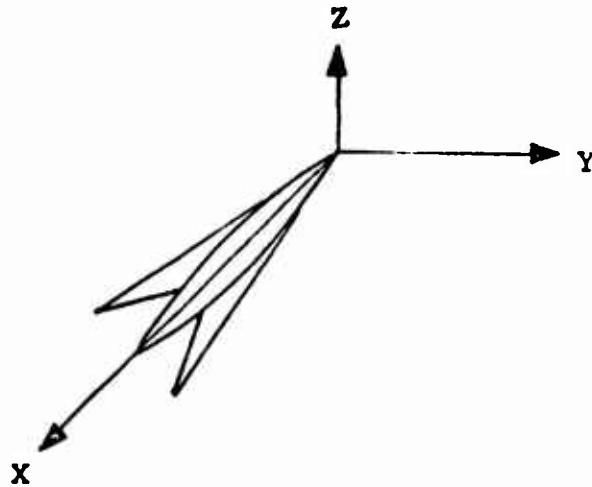
1. Three-views
2. Orthographic, from an arbitrary viewing angle
3. Perspective, from an arbitrary viewing angle
4. Stereoscopic, from an arbitrary viewing angle

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

1. On-line cathode ray tube
2. CALCOMP plotter
3. Houston COMPLIT plotter
4. Gerber plotter
5. Stereoscope

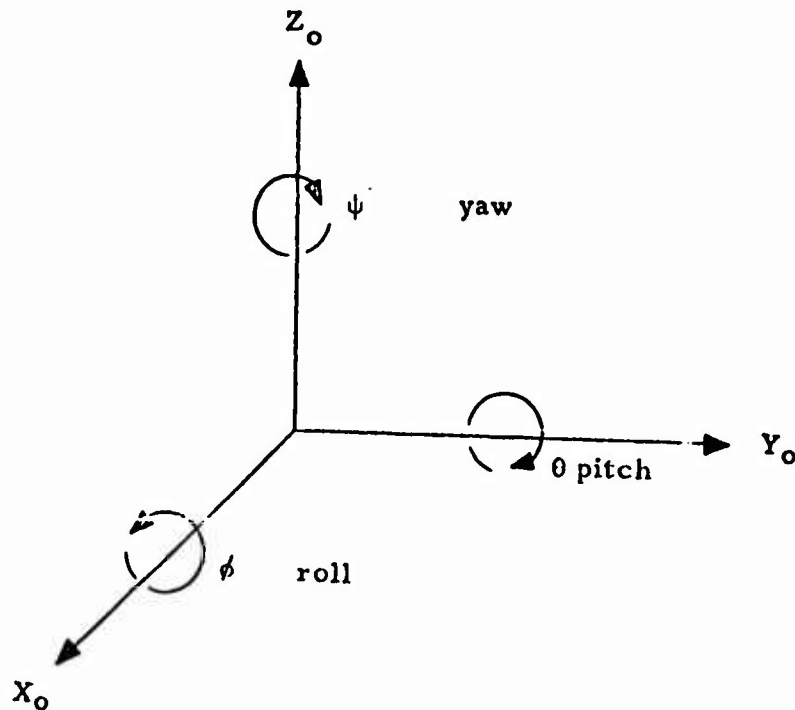
The numerical model of the aircraft configuration may include any combination of components: wing, body, pods, fins, and canards. The wing is made up of airfoil sections; the body is defined by either circular or arbitrary sections. The pods are defined similar to the fuselage, and fins and canards are 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.

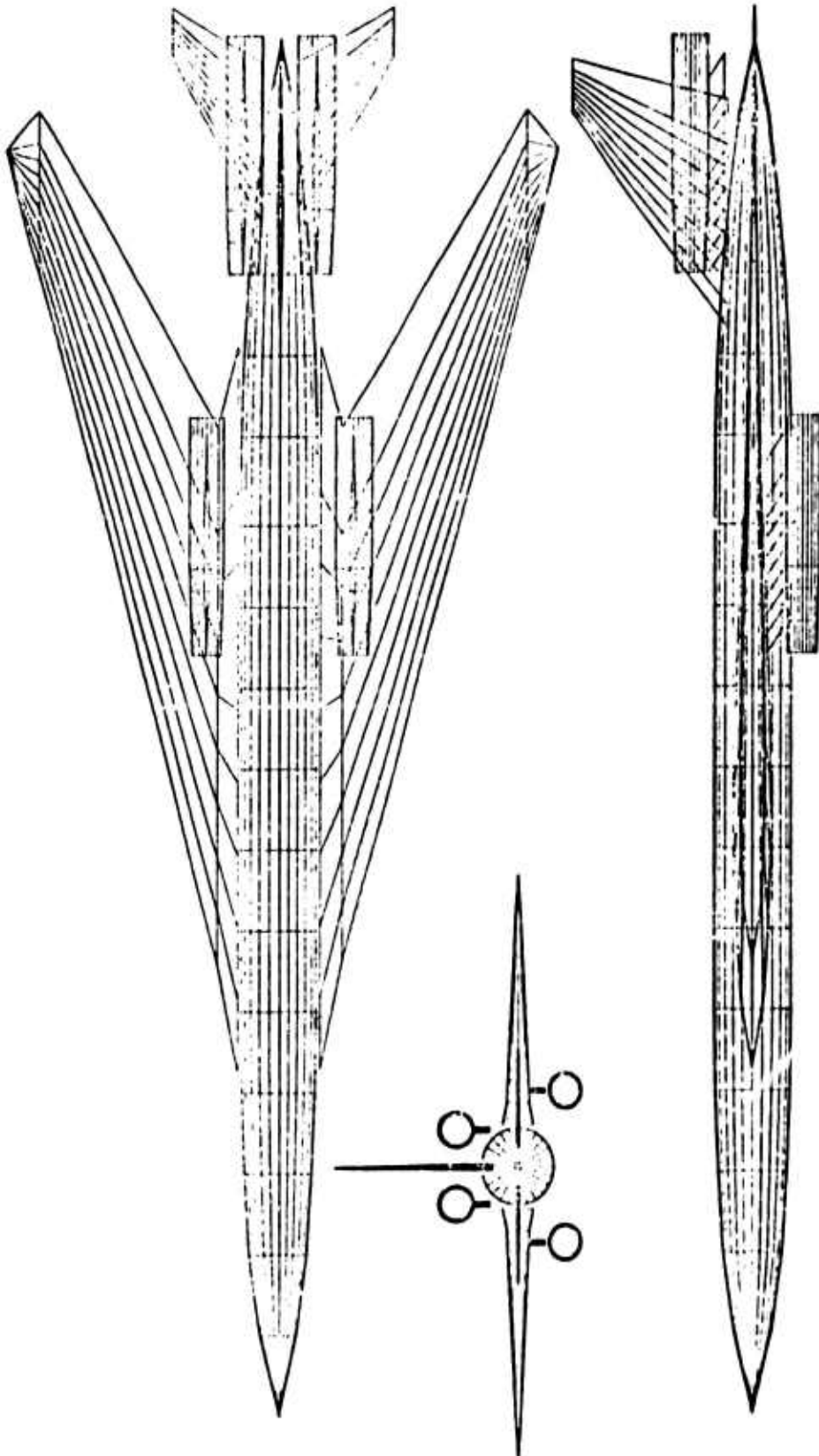


FIGURE 3.3-1. TYPICAL THREE-VIEW ORTHOGRAPHIC REPRESENTATION OF AN AIRCRAFT CONFIGURATION

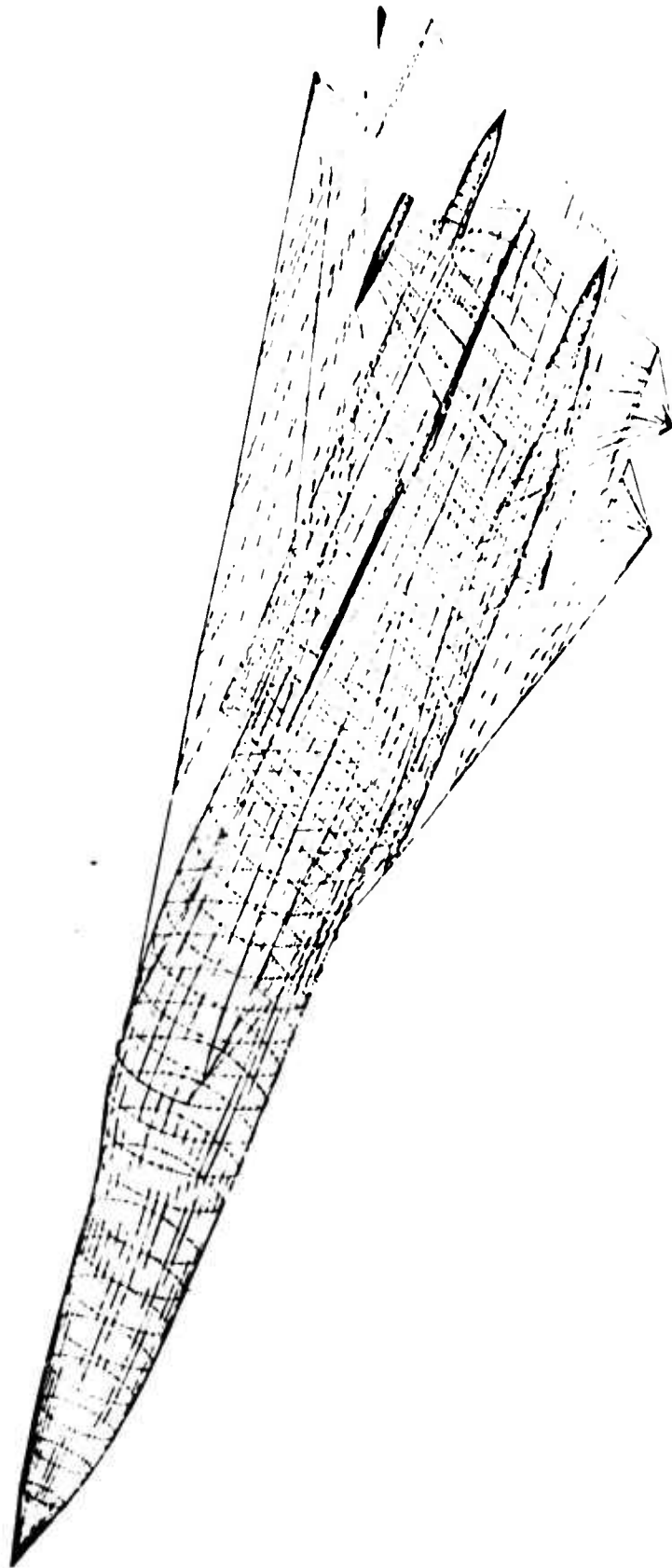


FIGURE 3.3-2. PERSPECTIVE VIEWS OF A REPRESENTATIVE AIRCRAFT CONFIGURATION

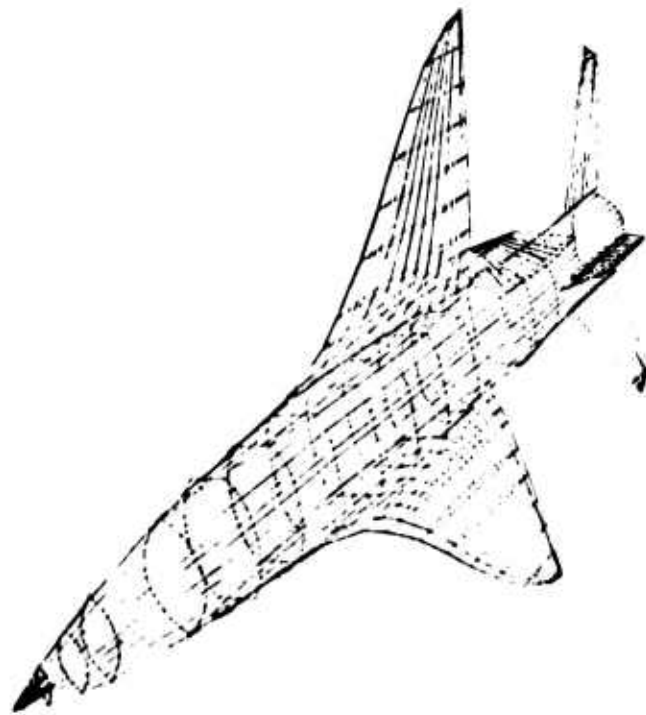
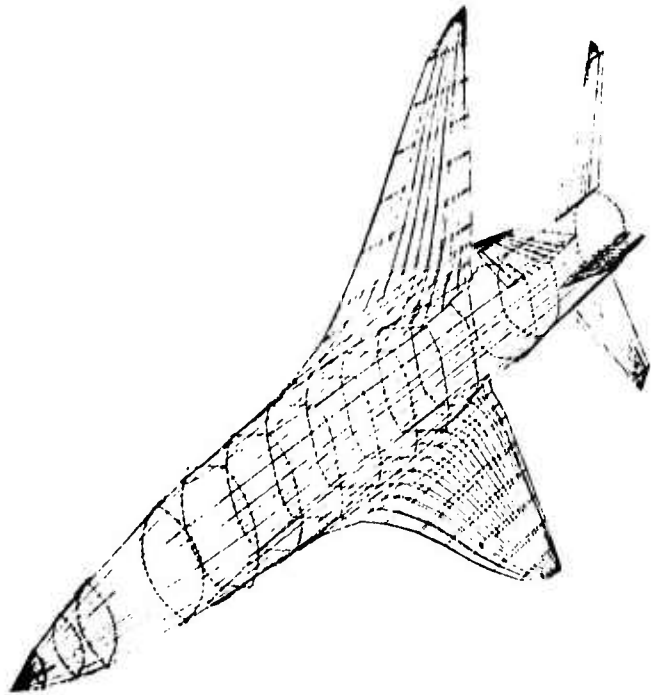


FIGURE 3.3-3. EXAMPLES OF STEREO FRAMES FOR A THREE-DIMENSIONAL VIEW USING AN AIRCRAFT CONFIGURATION

SECTION 4

AERODYNAMICS

The ODIN/MFV 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

In addition, the related ODIN/RLV system installed at NASA's Langley Research Center contains programs for computing

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

Each program now available in the ODIN/MFV system is outlined below; for complete details, reference should be made to the original source documents.

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 References 1 and 2. A CDC 6600 version of this program constructed at Aerophysics Research Corporation contains extensive graphics capability based on elements of the SEAL program, Reference 3. The program of References 1 and 2 treats the vehicle surface as a collection of quadrilateral elements oriented tangentially to the local vehicle surface in the manner of Section 3.1. 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 , $C_{l\alpha}$, C_n , $C_{A\alpha}$, $C_{L\alpha}$, $C_{N\alpha}$, $C_{m\alpha}$, C_{mq} , C_{Aq} , C_{Nq} , $C_{Y\beta}$, $C_{N\beta}$, $C_{l\beta}$, C_{Yr} , C_{Nr} , C_{lr} , $C_{m\delta}$, $C_{l\delta}$, $C_{y\delta}$, $C_{n\delta}$, $C_{N\delta}$, $C_{m\dot{\alpha}}$, $C_{Y\dot{\beta}}$, 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/MFV 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 and 5.

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/MFV 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. This 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 in Reference 1. Some 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 C_p/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. For a discussion of the other available methods, refer to Reference 1.

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 C_p 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 ($C_{p\text{STAG}} = K$) 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

$$\frac{C_p}{C_{p\text{max}}} = \frac{\sin^2 \delta}{\sin^2 \delta_{\text{max}}}$$

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

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.

REFERENCES:

1. Gentry, Arvel, Hypersonic Arbitrary-Body Aerodynamic Computer Program, Volume I, Douglas Report DAC 56080, June 1967.
2. Gentry, Arvel, Hypersonic Arbitrary-Body Aerodynamic Computer Program, Volume II, Douglas Report DAC 56080, June 1967.
3. Jones, R. T., SEAL User's Manual, Boeing Doc. D2-139592-1, 1969.
4. Powers, S. A., et al., A Numerical Procedure for Determining 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.

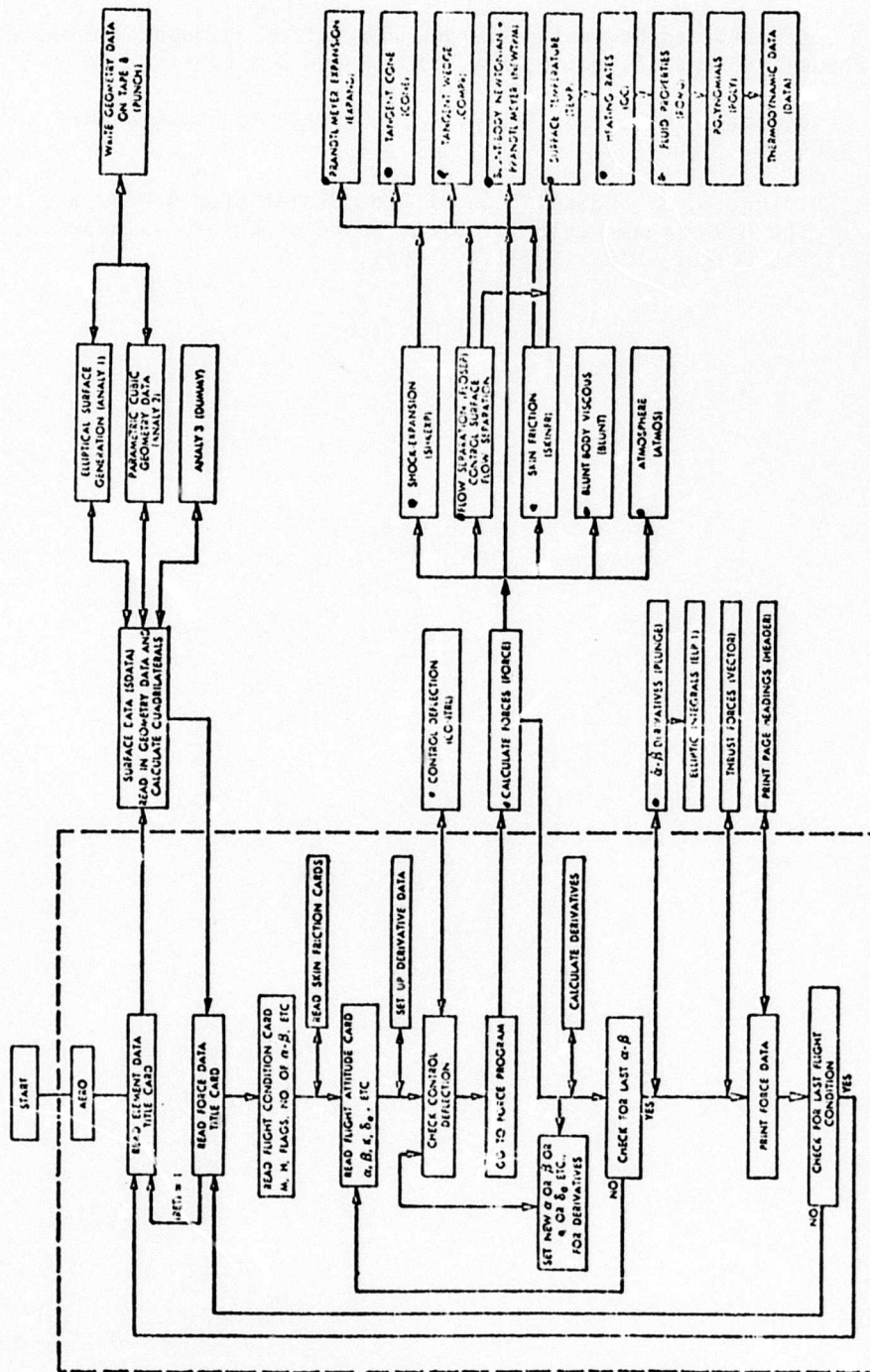


FIGURE 4.1-1. AERODYNAMIC PROGRAM ORGANIZATION

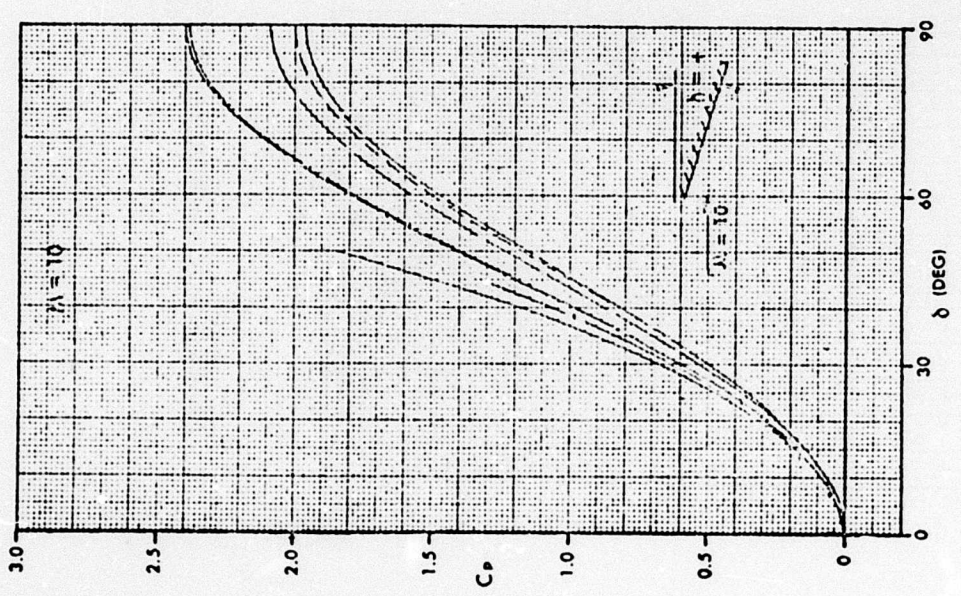
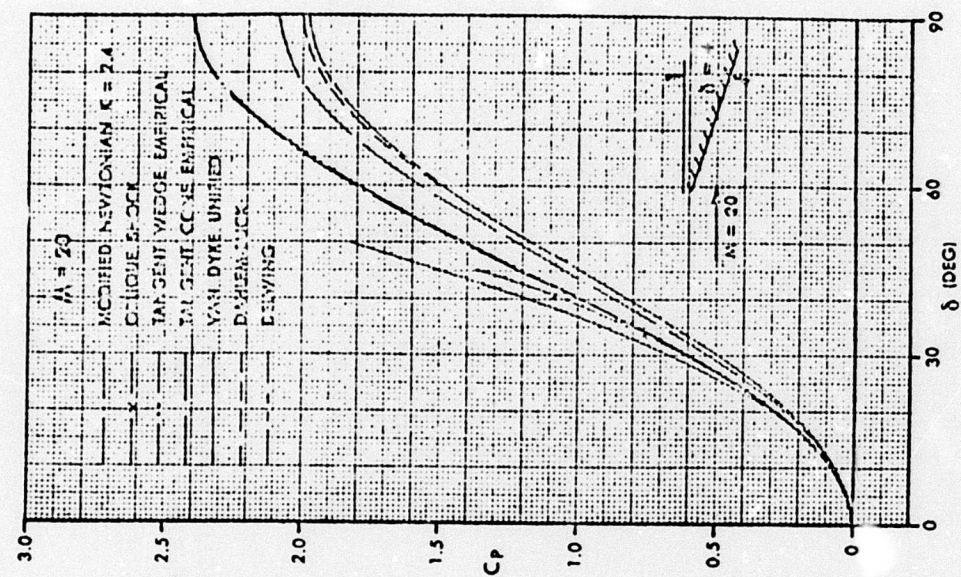


FIGURE 4.1-2. COMPARISON OF PRESSURE PREDICTION METHODS FOR SHARP BODIES IN COMPRESSION TYPE FLOW

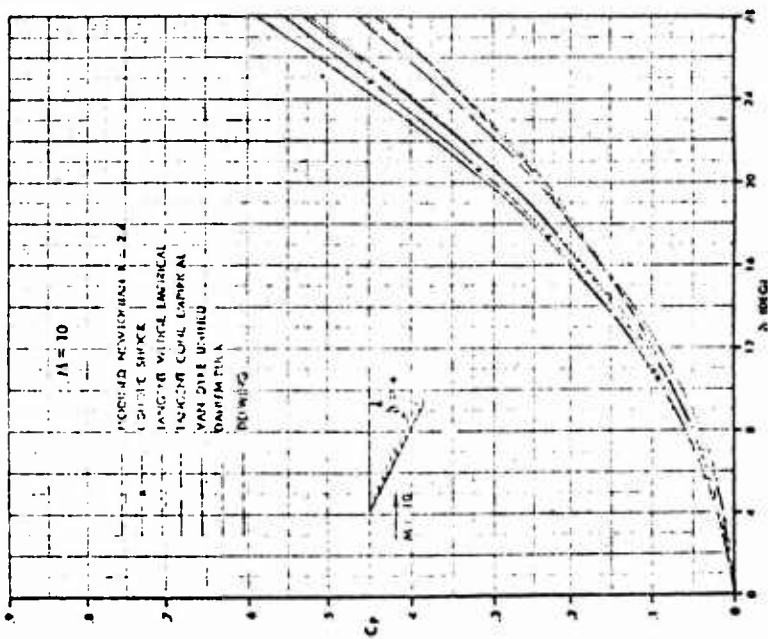
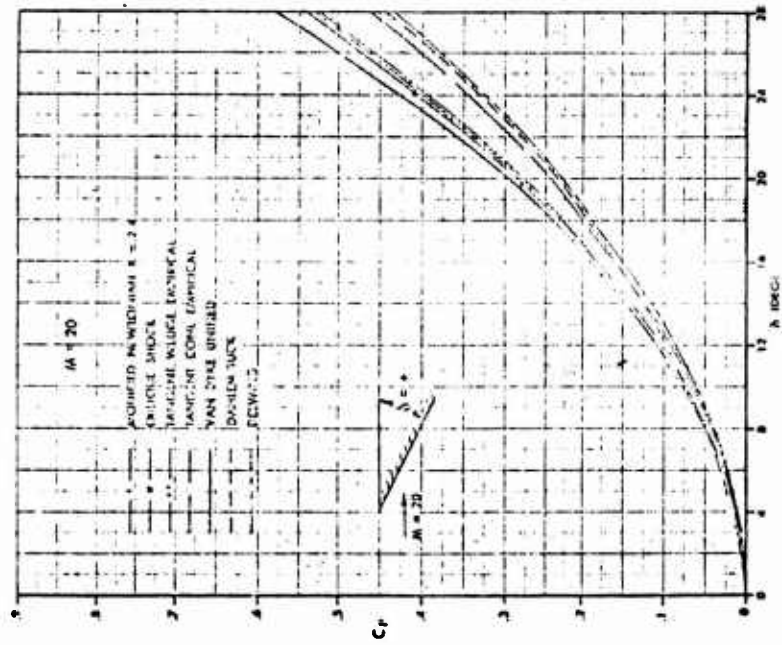


FIGURE 4.1-3. COMPARISON OF PRESSURE PREDICTION METHODS FOR SHARP BODIES IN COMPRESSION TYPE FLOW

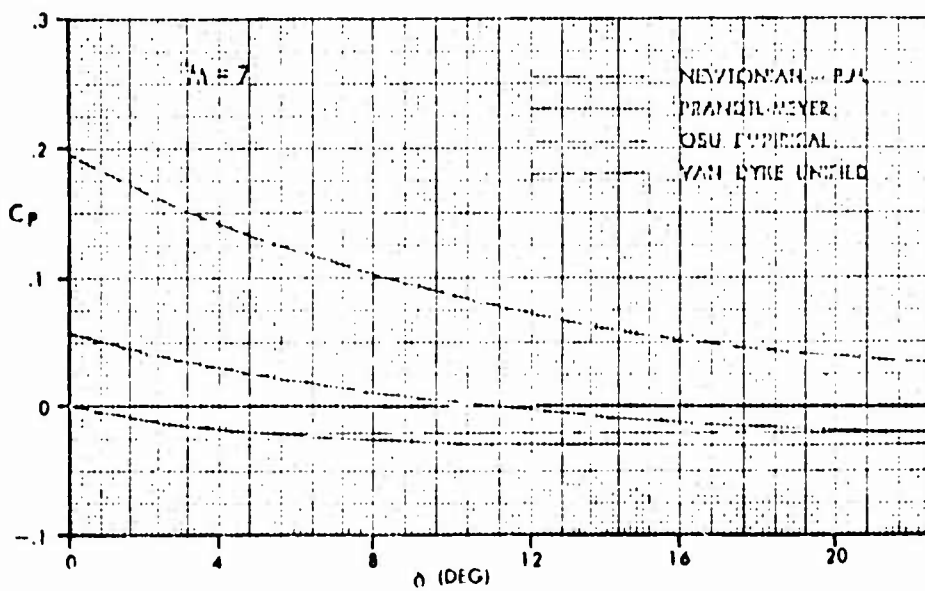
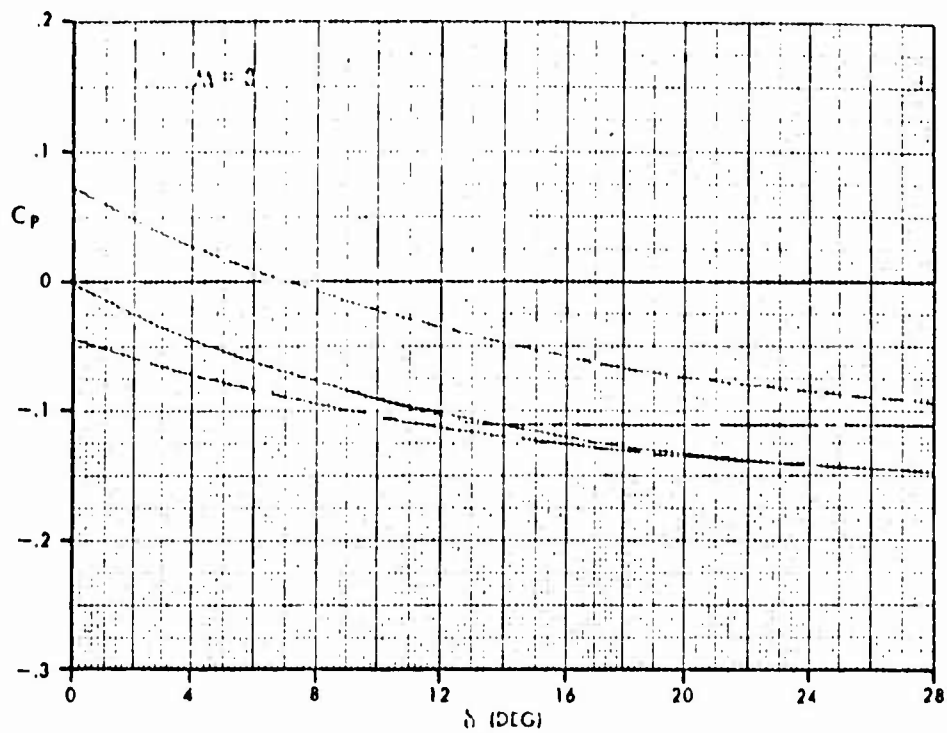


FIGURE 4.1-4. COMPARISON OF PRESSURE PREDICTION TECHNIQUES FOR EXPANSION TYPE FLOW

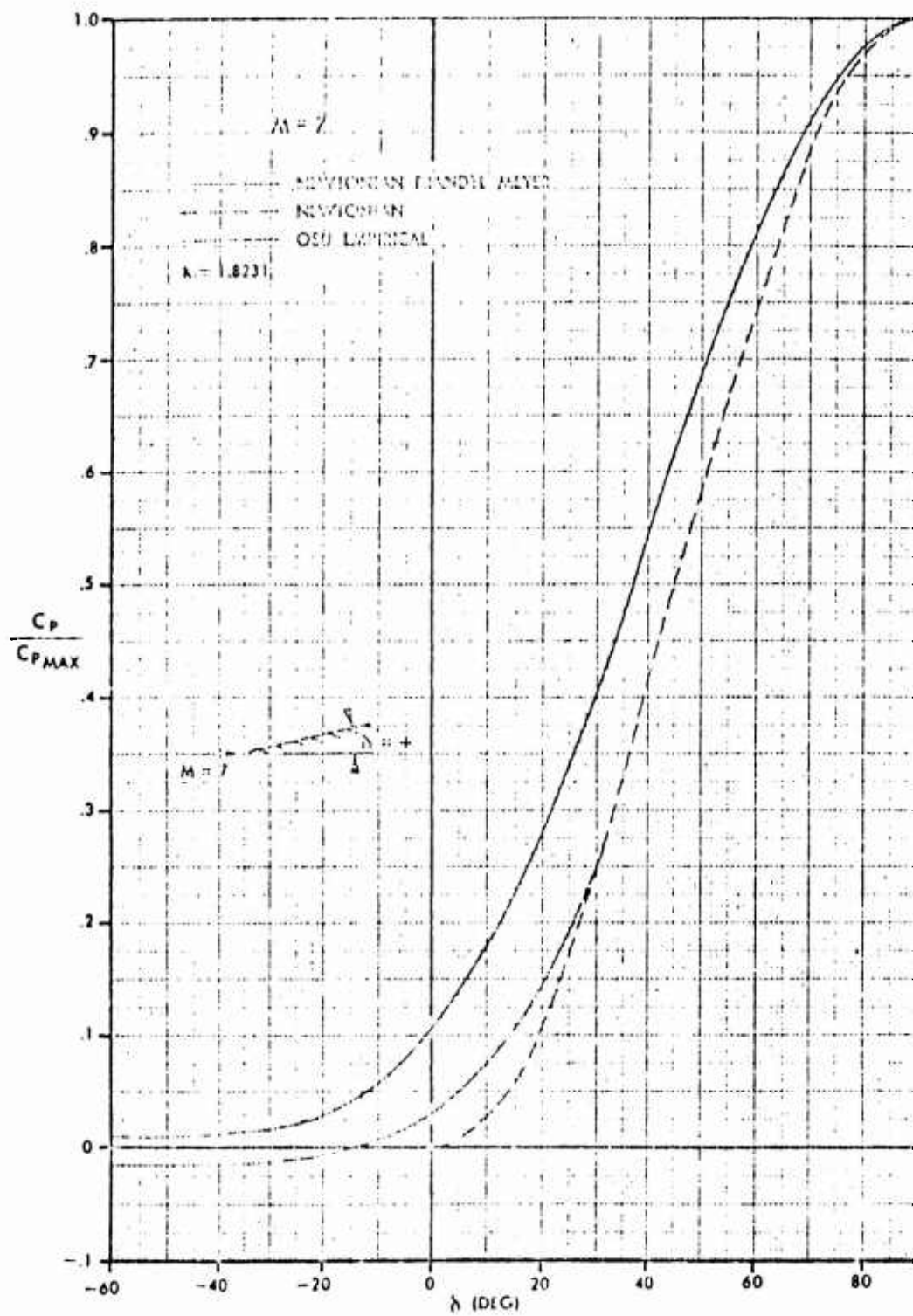


FIGURE 4.1-5. COMPARISON OF BLUNT BODY PRESSURE ESTIMATION TECHNIQUES

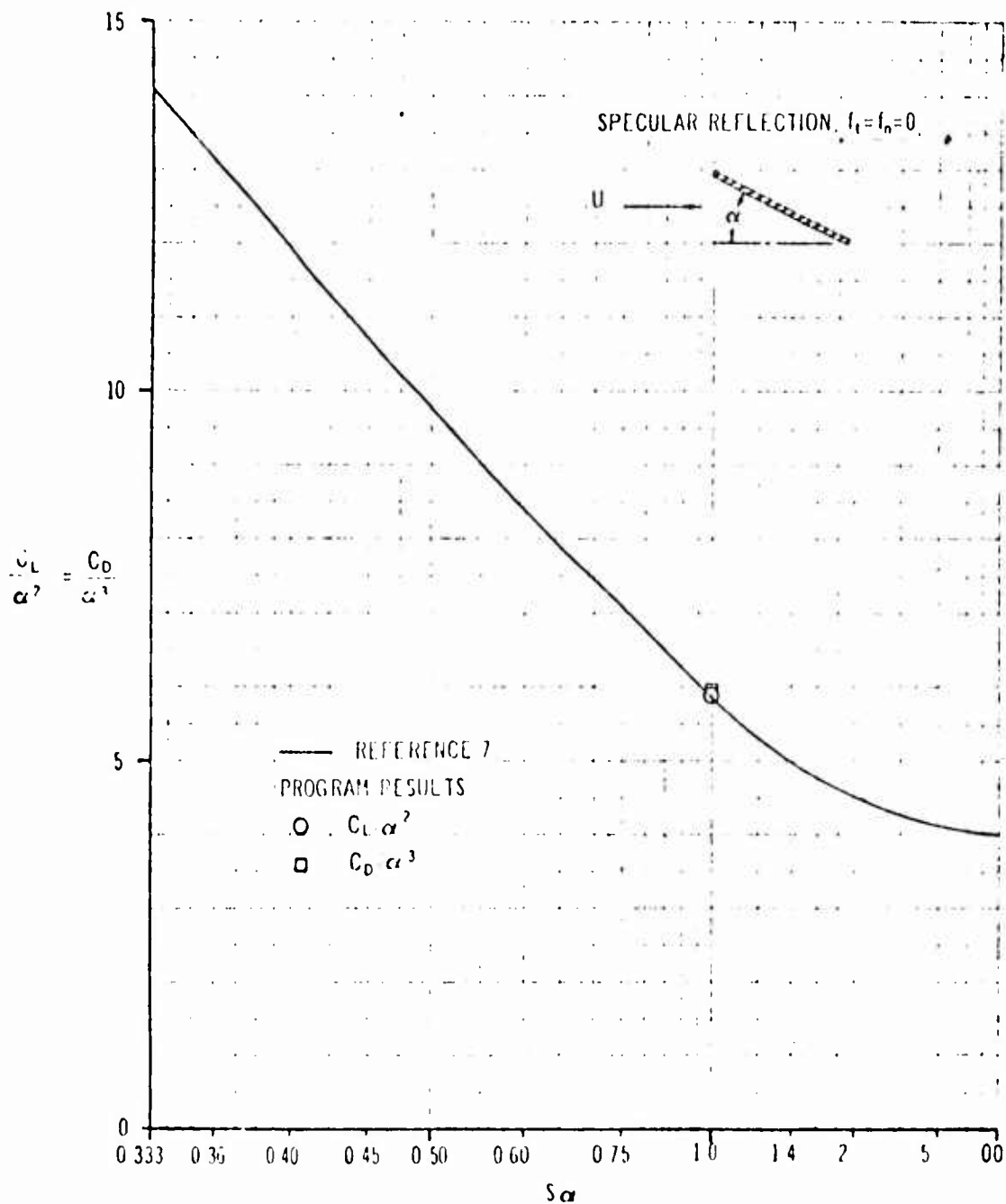


FIGURE 4.1-6. COMPARISON OF FREE MOLECULAR FLOW LIFT AND DRAG FOR A FLAT PLATE WITH SPECULAR REFLECTION

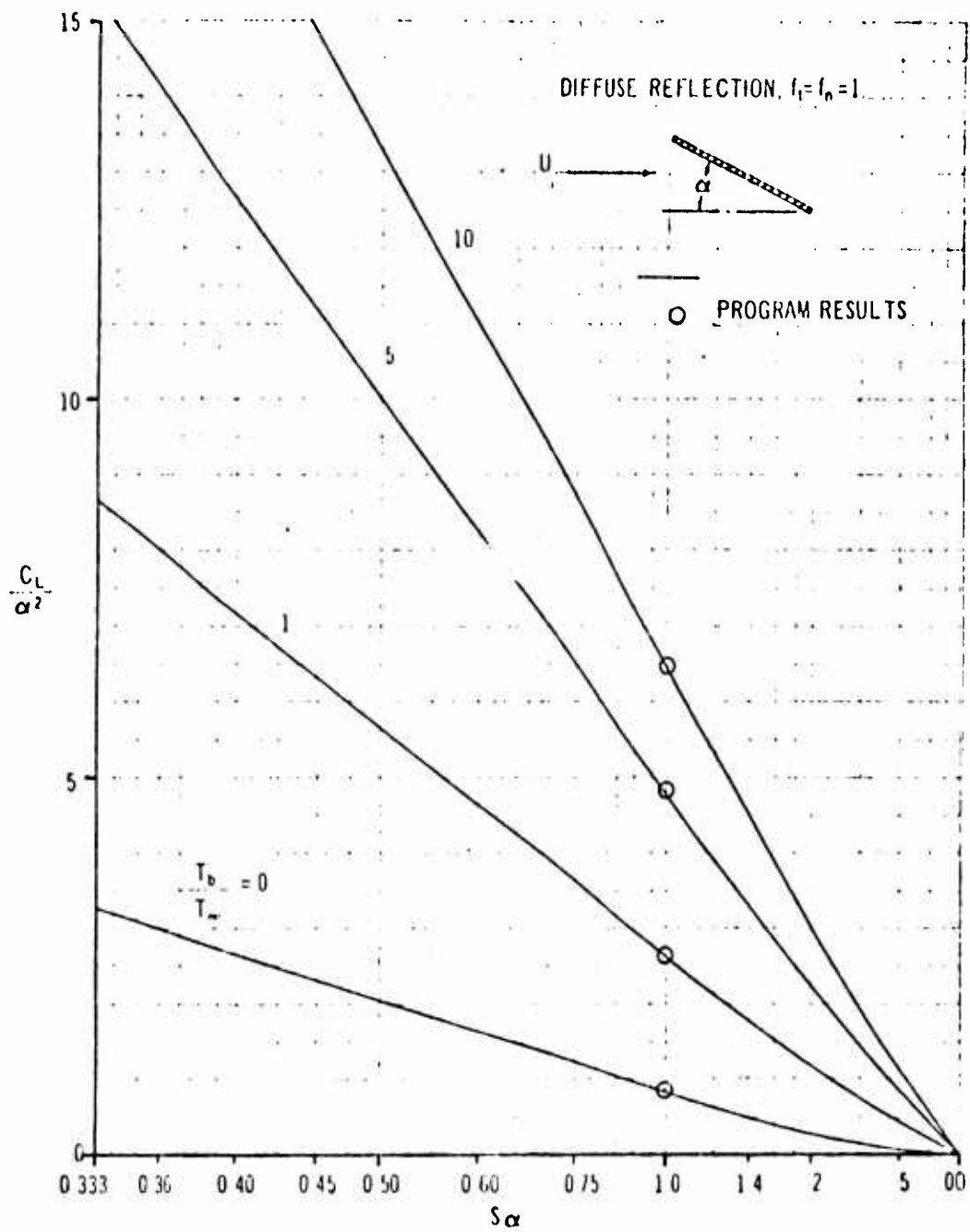


FIGURE 4.1-7. COMPARISON OF FREE MOLECULAR LIFT ON A FLAT PLATE WITH WITH DIFFUSE REFLECTION

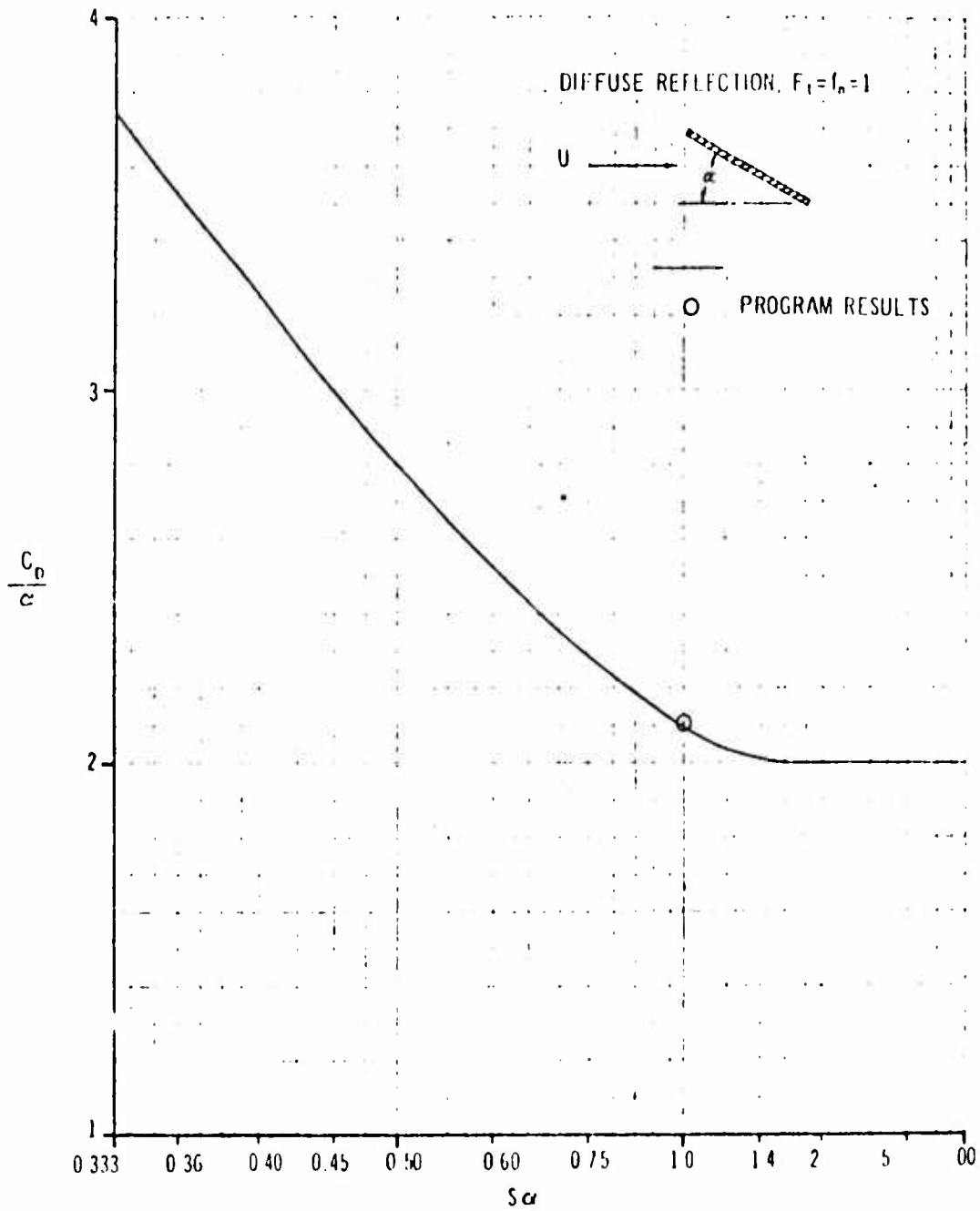


FIGURE 4.1-8. COMPARISON OF FREE MOLECULAR DRAG ON A FLAT PLATE WITH DIFFUSE REFLECTION

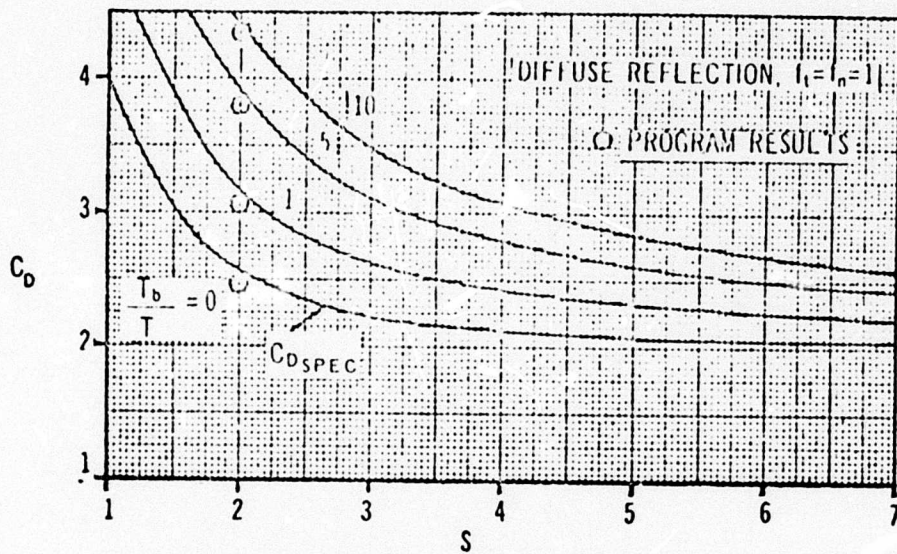


FIGURE 4.1-9. COMPARISON OF FREE MOLECULAR DRAG ON A SPHERE WITH DIFFUSE REFLECTION

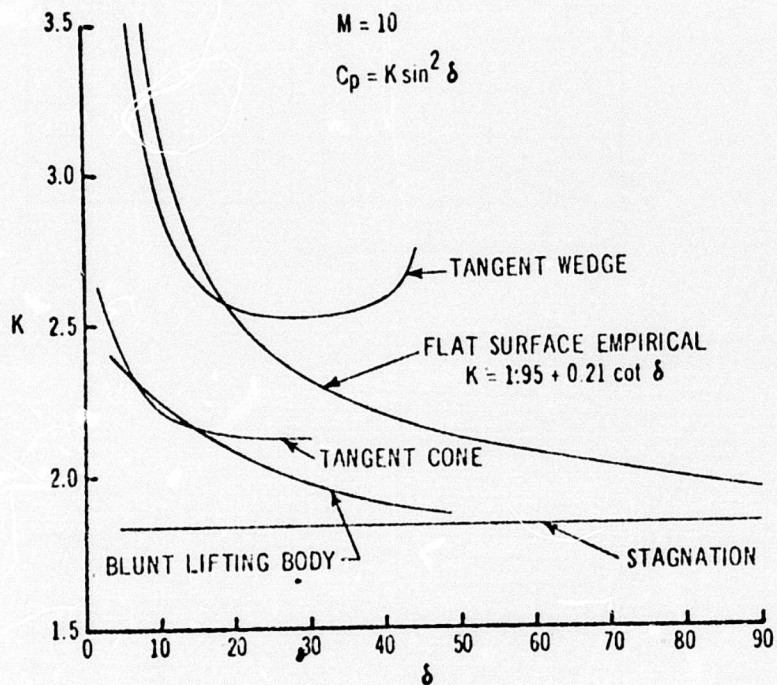


FIGURE 4.1-10. MODIFIED NEWTONIAN CORRELATION FACTORS

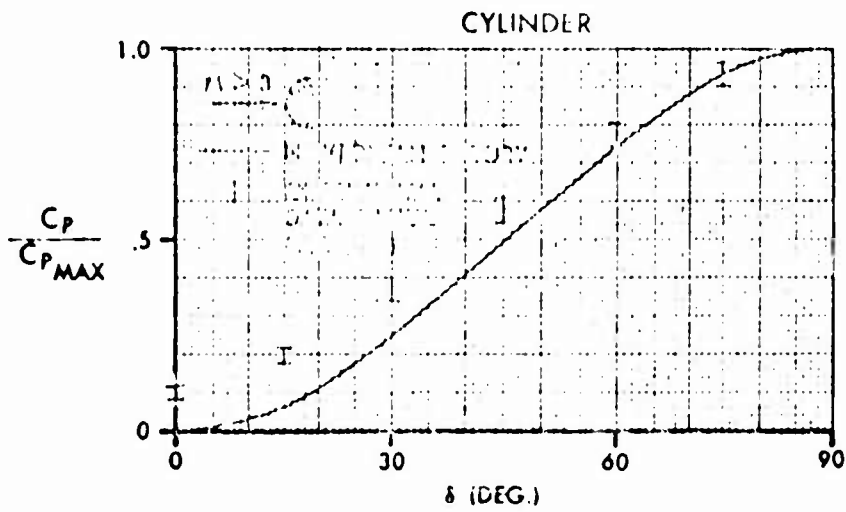
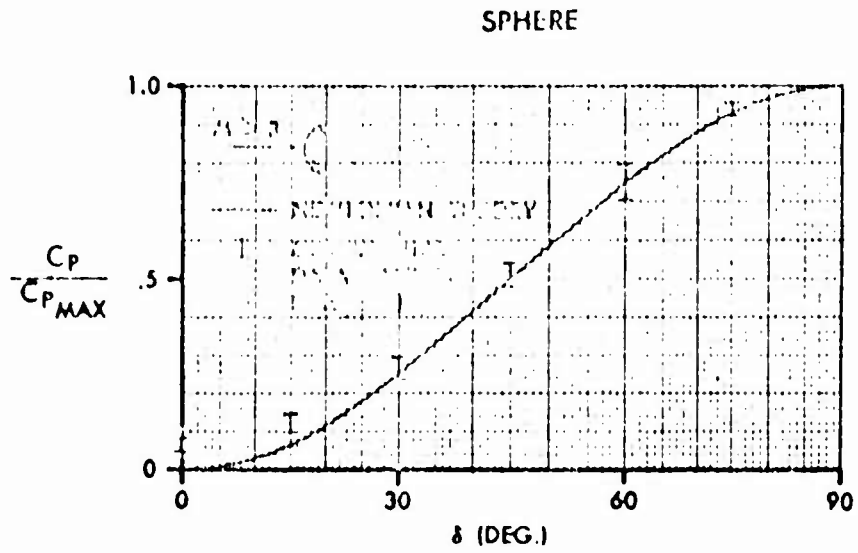


FIGURE 4.1-11. COMPARISON OF EXPERIMENTAL PRESSURES AND MODIFIED NEWTONIAN THEORY FOR SPHERES AND CYLINDERS

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 military 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'Alembert'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 $M = 1.0$ from above, then the expression

$$\left(\frac{D_w}{\rho V^2}\right)_{M=1.0} = -\frac{1}{7\pi} \int_{-L/2}^{L/2} \int_{-L/2}^{L/2} S''(x_1) S''(x_2) \log|x_1 - x_2| dx_1 dx_2 \quad (4.2.1)$$

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 δx , then the flux out of this disc in both cases will be

$$\delta Q = \frac{dS}{dx} \cdot \delta x \quad (4.2.2)$$

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

$$\therefore \frac{D_w}{q} = -\frac{1}{4\pi^2} \int_0^{2\pi} d\theta \int_{-L/2}^{L/2} \int_{-L/2}^{L/2} S''(x_1, \theta, M) S''(x_2, \theta, M) \times \log |x_1 - x_2| dx_1 dx_2$$

(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

$$\begin{aligned} \frac{D_{wv}}{q} = & -\frac{1}{2\pi} \iint_{\text{wing}} T''(\bar{R}_1) T''(\bar{R}_2) \cosh^{-1} \left\{ \frac{|x_1 - x_2|}{|\bar{R}_1 - \bar{R}_2|} \right\} dS_1 dS_2 \\ & + \frac{1}{2\pi} \int_{-L/2}^{L/2} \int_{-L/2}^{L/2} A''(x_1) A''(x_2) \log |x_1 - x_2| dx_1 dx_2 \\ & - \frac{1}{2\pi} \int_{-L/2}^{L/2} \int_{-L/2}^{L/2} [S_F''(x_1) + A''(x_1)] [S_F''(x_2) + A''(x_2)] \\ & \quad \times \log |x_1 - x_2| dx_1 dx_2 \quad (4.2.4) \end{aligned}$$

where

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

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

$$D_{\text{wing-body}} = D_{\text{wing}} - D\{A\} + D\{S_F + A\} \quad (4.2.6)$$

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

$$\frac{D_w}{\rho} = -\frac{1}{4\pi^2} \int_0^{2\pi} d\theta \int_{L_1(\theta)}^{L_2(\theta)} dx_1 \int_{L_1(\theta)}^{L_2(\theta)} \left[S''(x_1, \theta) - \frac{\beta^2}{2q} S'(x_1, \theta) \right] \times \left[S''(x_2, \theta) - \frac{\beta^2}{2q} S'(x_2, \theta) \right] \log|x_1 - x_2| dx_2 \quad (4.2.7)$$

where $l(x, \theta)$ 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

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

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

$$x = \frac{1}{2} (1 - \cos \Theta) \quad (4.2.9)$$

is made.

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

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

so that

$$A_n = \frac{2}{\pi} \int_0^{\pi} S'(x) \sin n\Theta d\Theta \quad (4.2.11)$$

With these assumptions the drag becomes

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

corresponding to an area distribution of

$$S(\alpha) = S(0) + \frac{A_1}{4} \theta + \frac{1}{4} \sum_{n=1}^{\infty} \frac{1}{n} (A_{n+1} - A_{n-1}) \sin n\theta \quad (4.2.13)$$

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 Einton's Method

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

$$S(c), S(1) \text{ and } S(\xi_i) \quad i = 1, 2, \dots, N \quad (4.2.14)$$

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, Einton shows by the method of Lagrangean multipliers the minimum drag is given by

$$\frac{D}{q} = \frac{1}{2} \left[\frac{4}{\pi} \{ S(1) - S(0) \}^2 + \pi \sum_{i=1}^N \sum_{j=1}^N c_i c_j d_{ij} \right] \quad (4.2.15)$$

or in matrix form

$$\frac{D}{q} = \frac{1}{2} \left[\frac{4}{\pi} (S(1) - S(0))^2 + \pi [c] [f] \{c\} \right] \quad (4.2.16)$$

where

$$c_i = S(\xi_i) - S(0) - (S(1) - S(0)) u_i \quad (4.2.17)$$

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

$$[f] = [p]^{-1} \quad (4.2.19)$$

and

$$p_{ij} = -\frac{1}{2} (\xi_i - \xi_j)^2 \log_2 \frac{\xi_i + \xi_j - 2\xi_i \xi_j + 2\sqrt{\xi_i \xi_j (1 - \xi_i)(1 - \xi_j)}}{\xi_i + \xi_j - 2\xi_i \xi_j - 2\sqrt{\xi_i \xi_j (1 - \xi_i)(1 - \xi_j)}} + 2(\xi_i + \xi_j - 2\xi_i \xi_j) \sqrt{\xi_i \xi_j (1 - \xi_i)(1 - \xi_j)} \quad (4.2.20)$$

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 . 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 ARPPII 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 ARPPII program, the method of Eminton is used.

4.2.7.2 Mach Plane Equations

The general equation of a plane is of the form

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

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

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

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 laid 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

$$x_i = a_{1i}y + b_{1i} \quad (4.2.23)$$

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

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

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

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

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

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

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

$$Ax + By + Cz + D_j = 0 \quad (4.2.29)$$

intersects the generator lines

$$x_i = a_{1i}y + b_{1i} \quad (4.2.30)$$

$$z_i = a_{2i}y + b_{2i} \quad (4.2.31)$$

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

$$\bar{y}_{ij} = \frac{Ab_{1i} + Cb_{2i} + D_j}{Aa_{1i} + B + Ca_{2i}} \quad (4.2.32)$$

The z coordinate of the intercept is

$$\bar{z}_{ij} = a_{2i} \bar{y}_{ij} + b_{2i} \quad (4.2.33)$$

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

$$\Delta S_j = \int \bar{z}_{ij} d\bar{y}_{ij} \quad (4.2.34)$$

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 locus of area be y_T , z_T then substituting these values into the Mach plane equation gives the required intercept, i.e.,

$$x_T = x^1 - (By_T + Cz_T) \quad (4.2.35)$$

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

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 through 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 Emlin 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 ARP II, 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 Allessando 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 10, 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. Emlinton, 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.

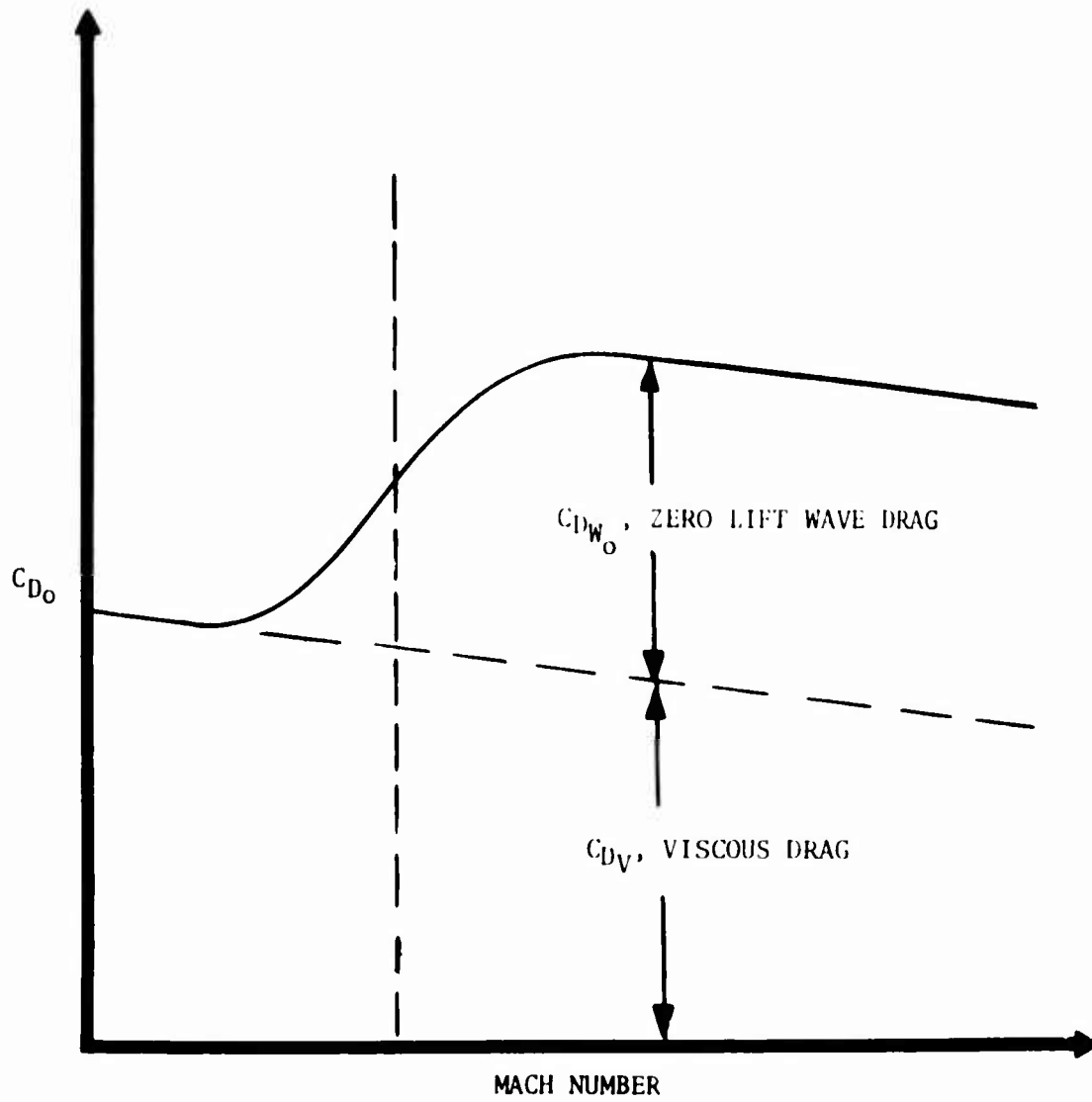


FIGURE 4.2-1. TYPICAL ZERO LIFT DRAG COEFFICIENT

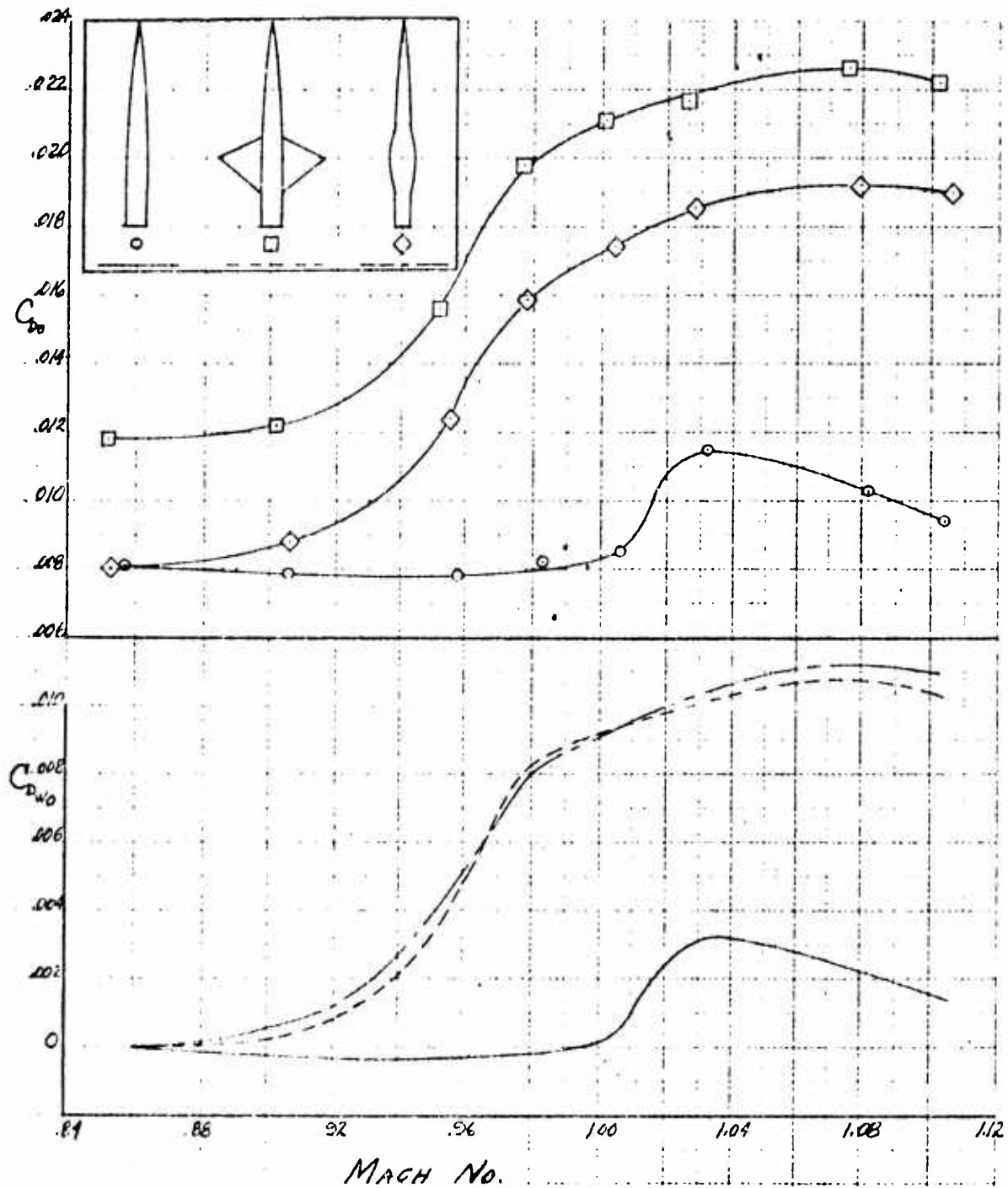


FIGURE 4.2-2(a)

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

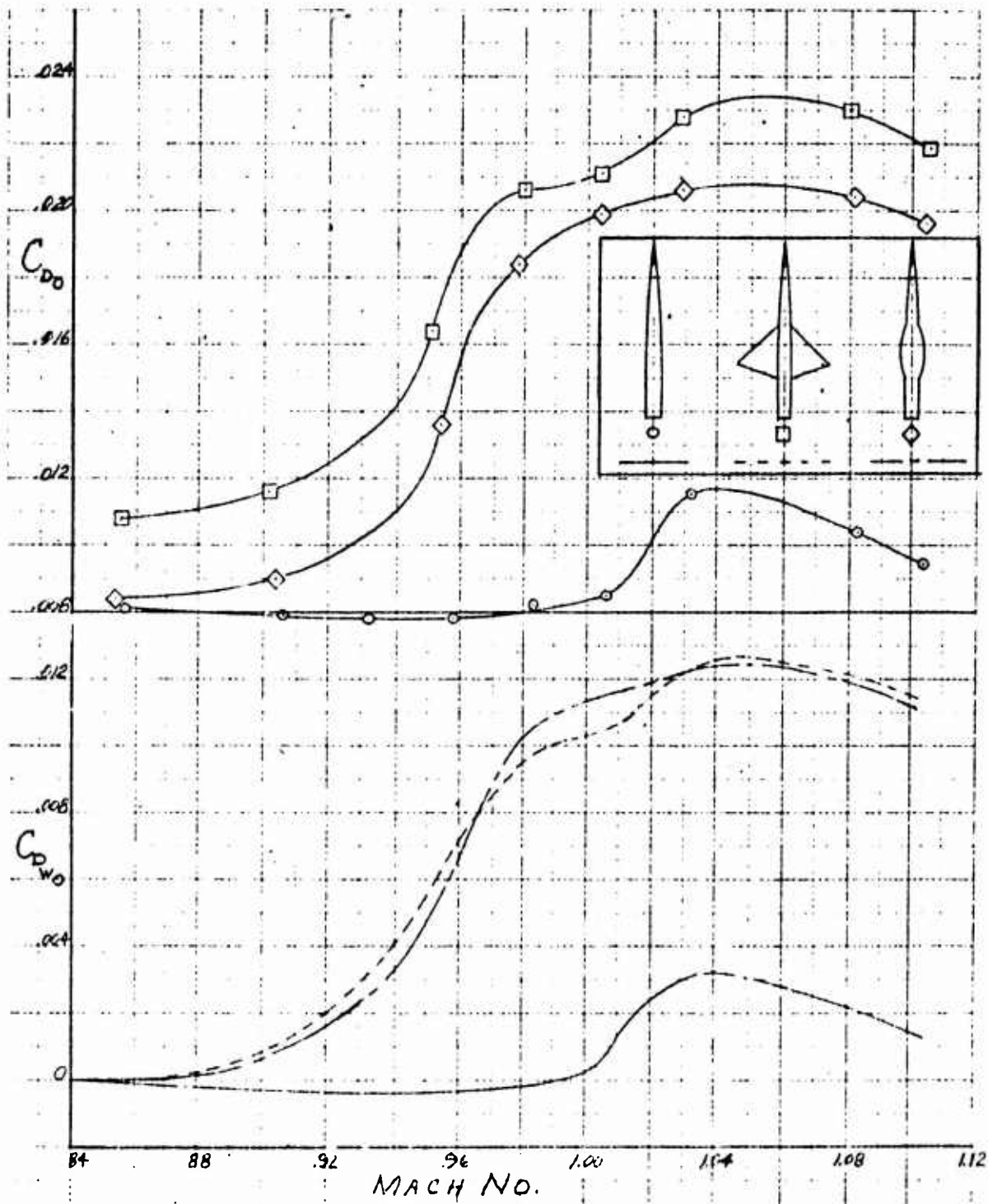


FIGURE 4.2-2(b)

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

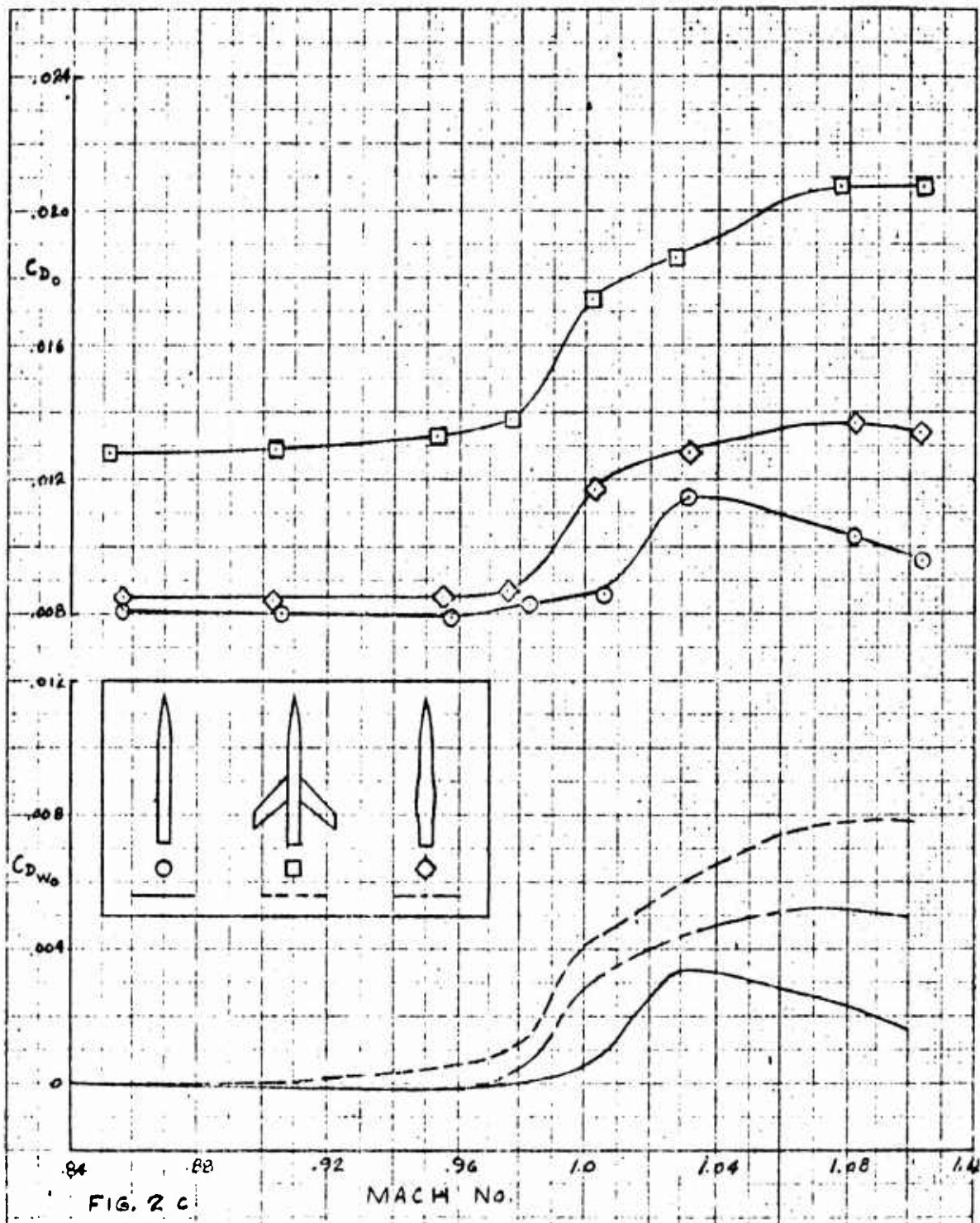


FIGURE 4.2-2 (c)

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

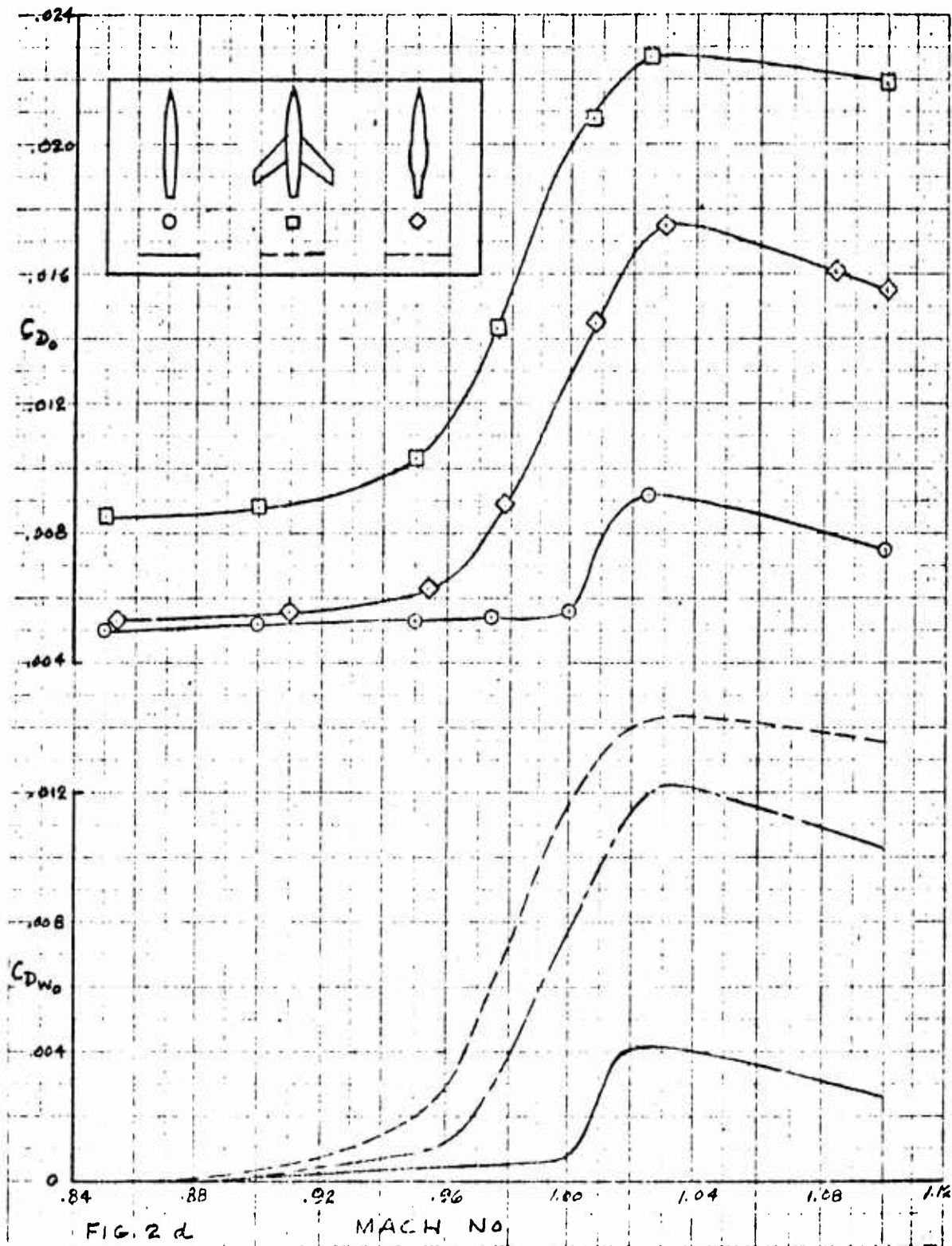


FIGURE 4.2-2(d)

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

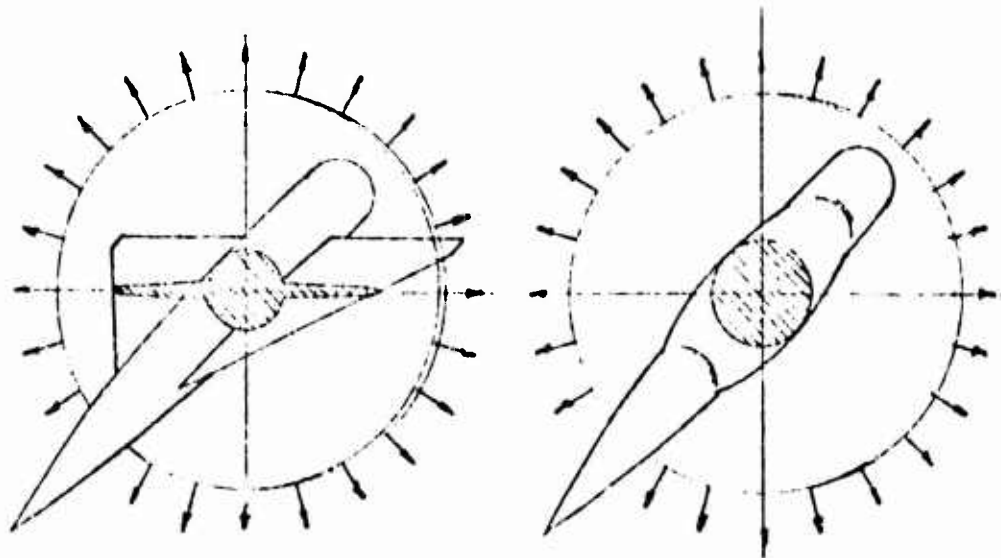


FIGURE 4.2-3. INCOMPRESSIBLE NATURE OF TRANSONIC FLOW FIELD

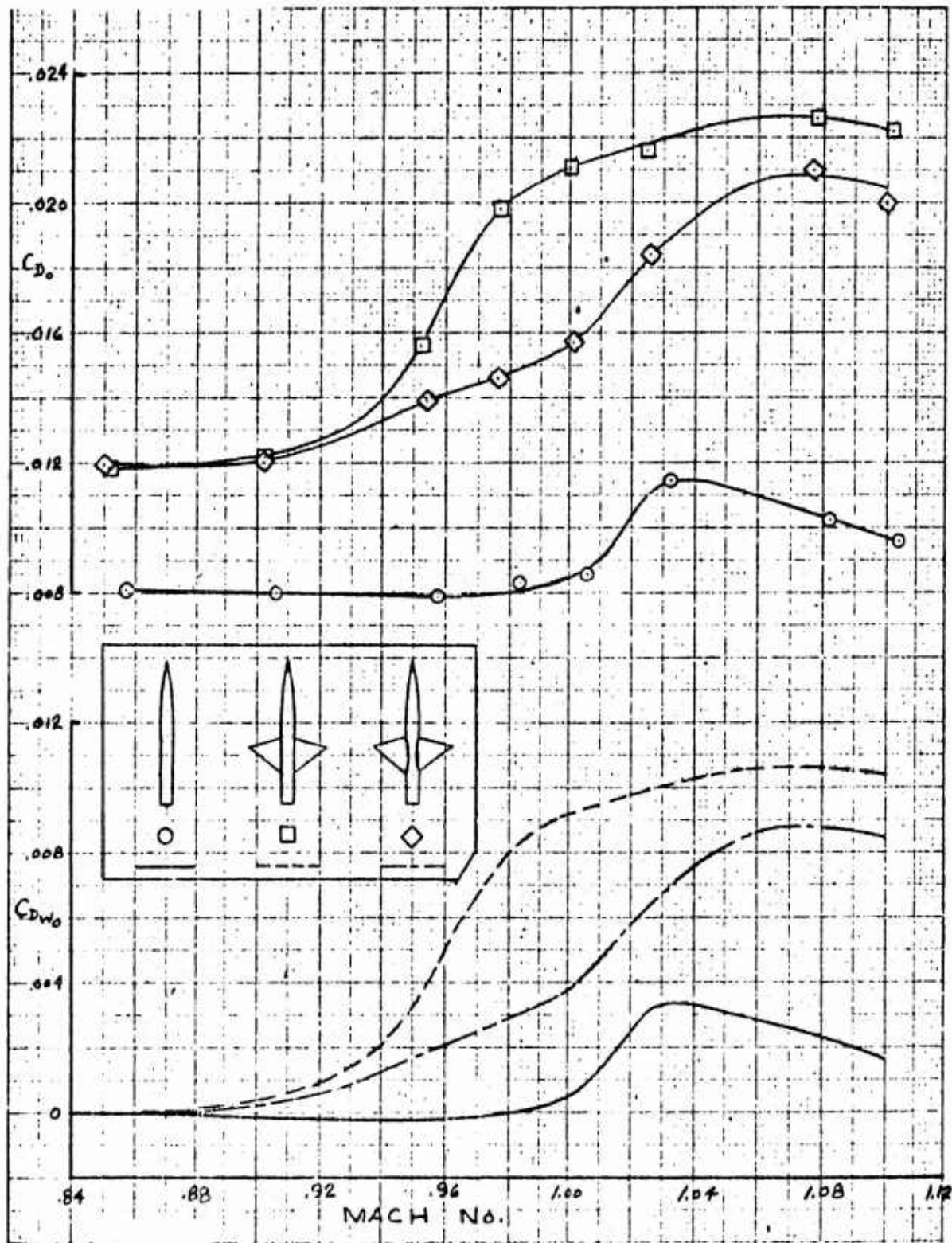


FIGURE 4.2-4(a)

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

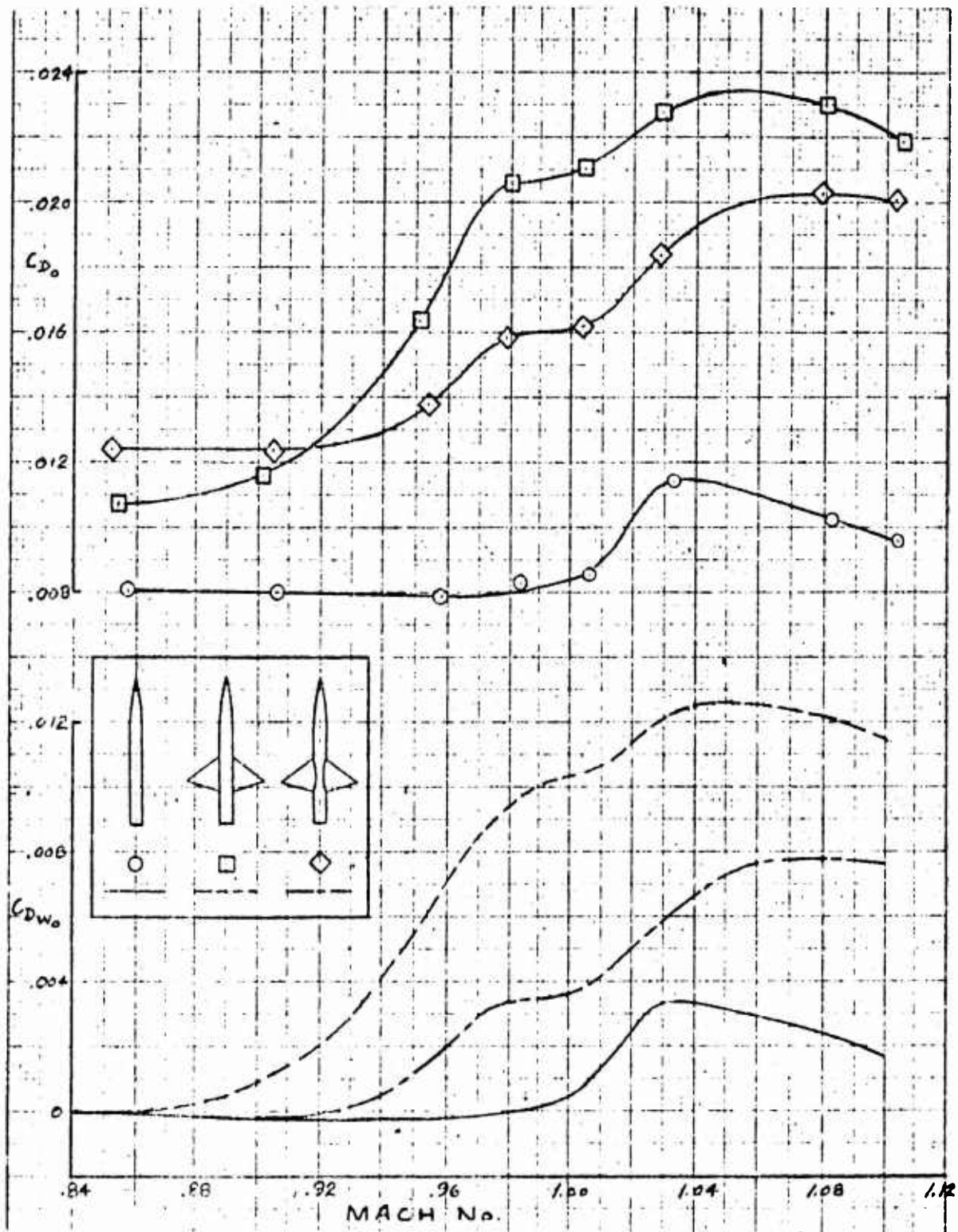


FIGURE 4.2-4(b)

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

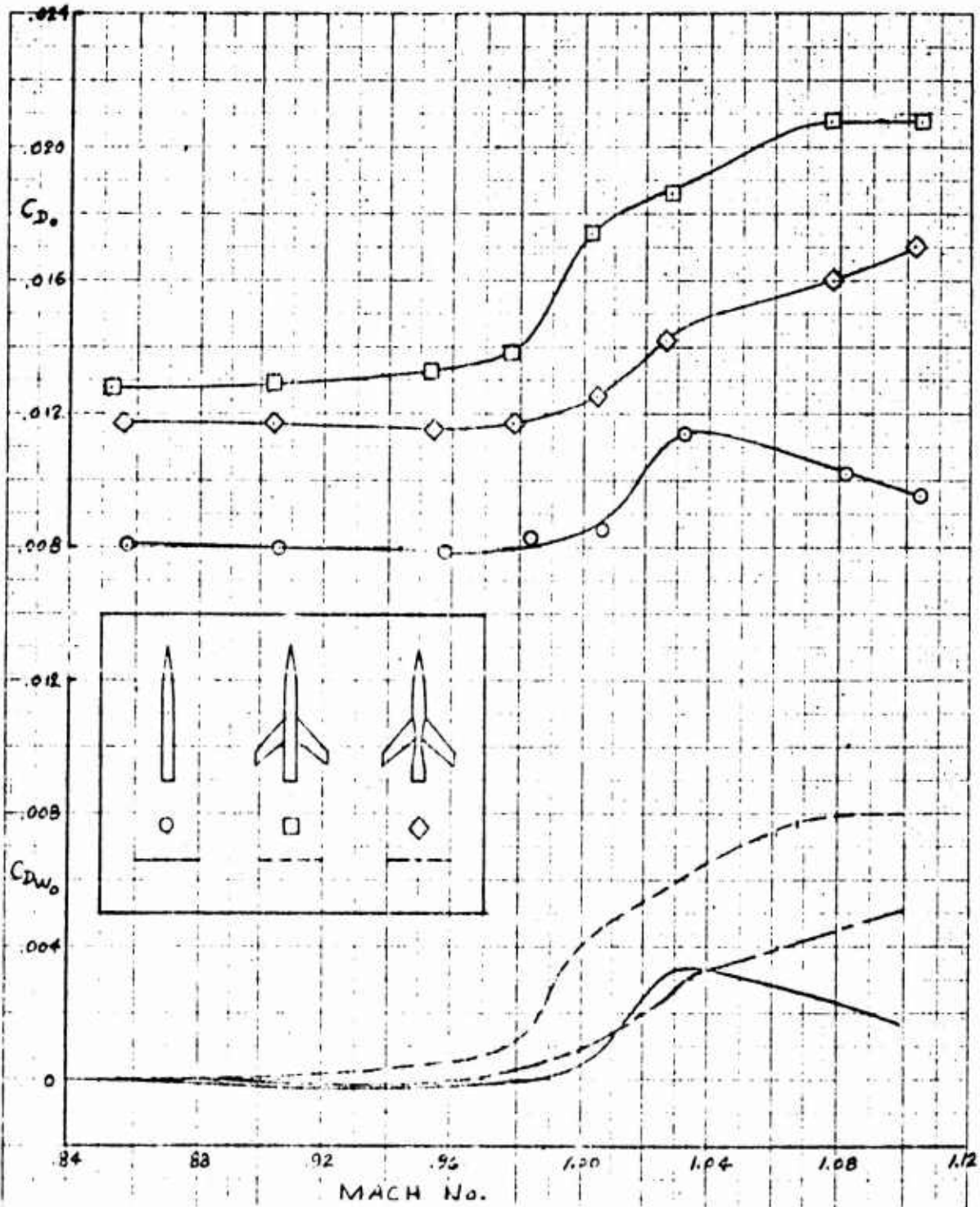


FIGURE 4.2-4(c)

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

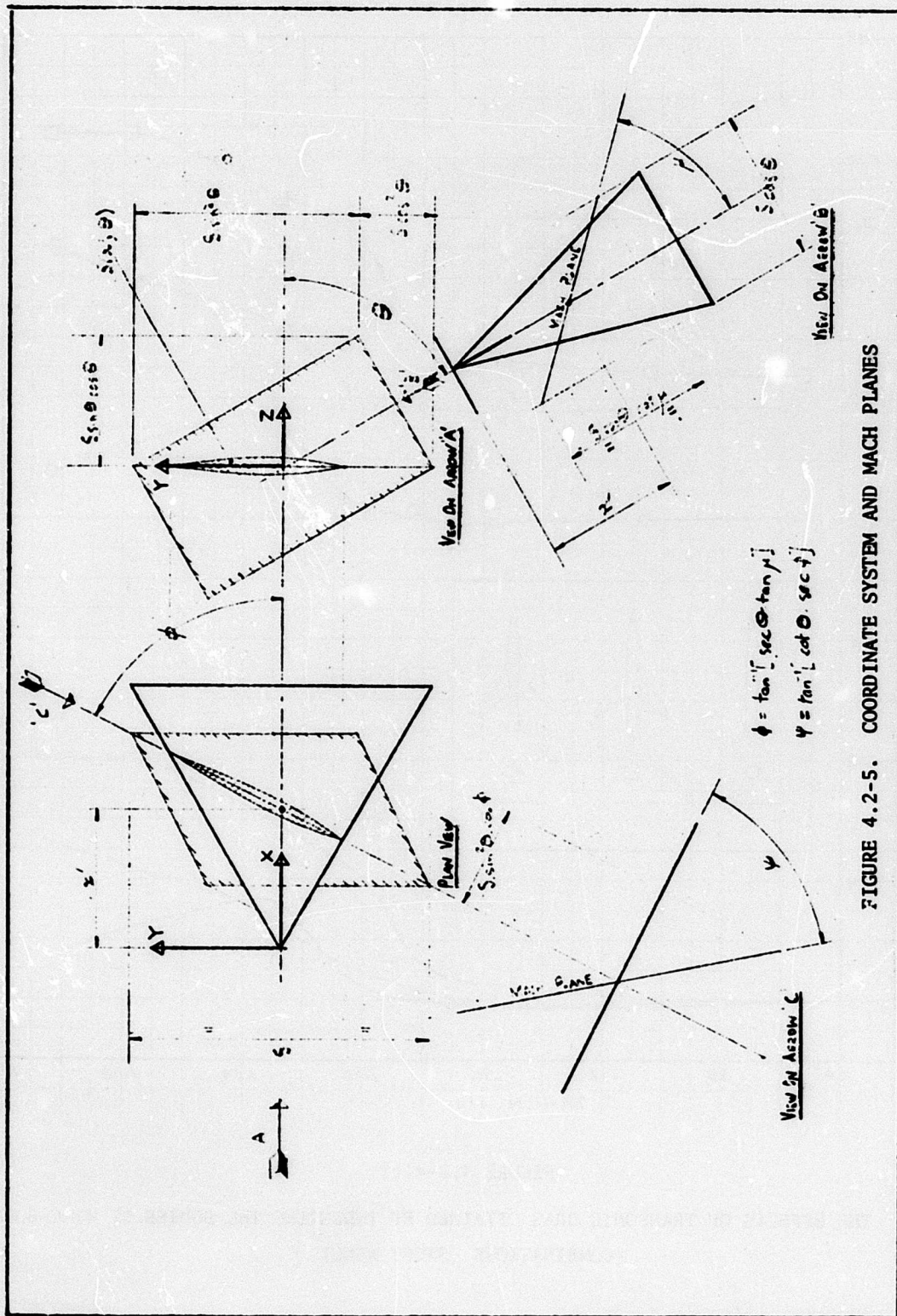


FIGURE 4.2-5. COORDINATE SYSTEM AND MACH PLANES

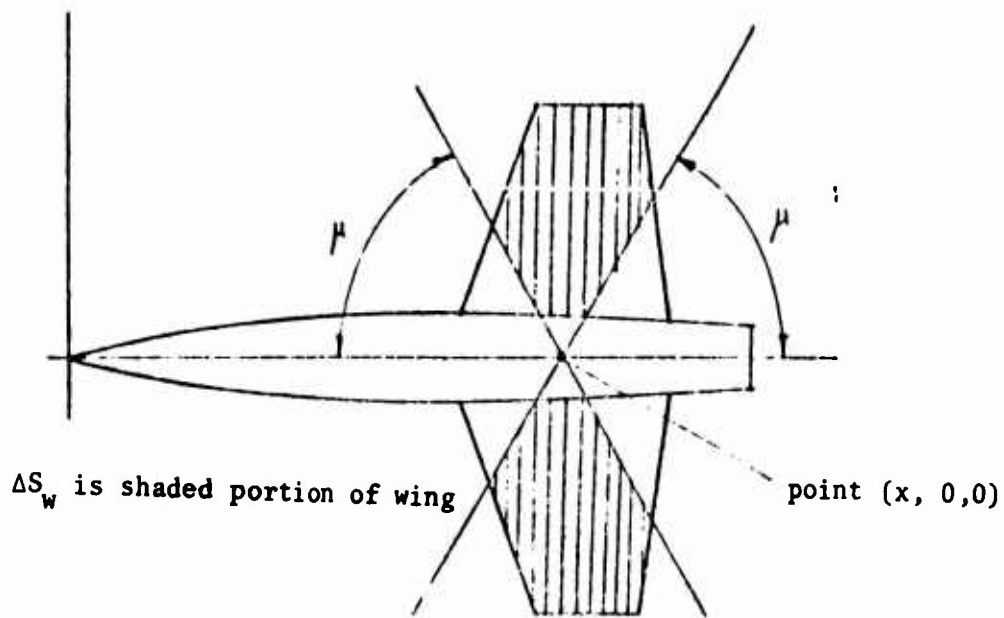


FIGURE 4.2-6. WING REGION FOR OBTAINING $A(x)$.

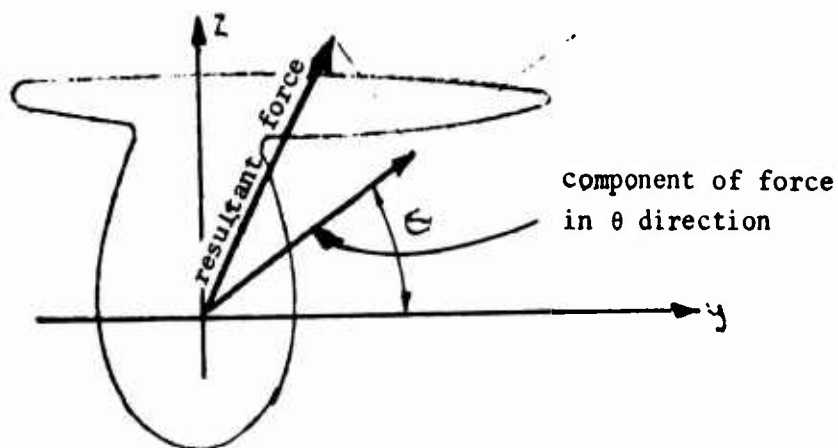


FIGURE 4.2-7. INCLUSION OF LIFT IN SUPERSONIC AREA RULE

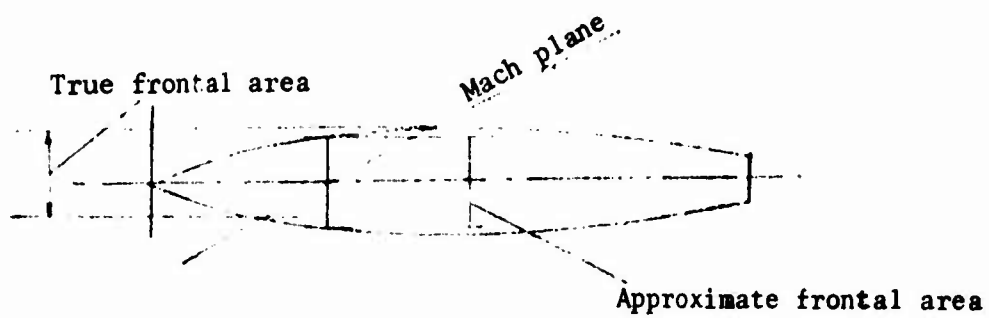


FIGURE 4.2-8. FUSELAGE AREA CONTRIBUTION

Figure 4.2-9a. Layout of Basic Example Aircraft

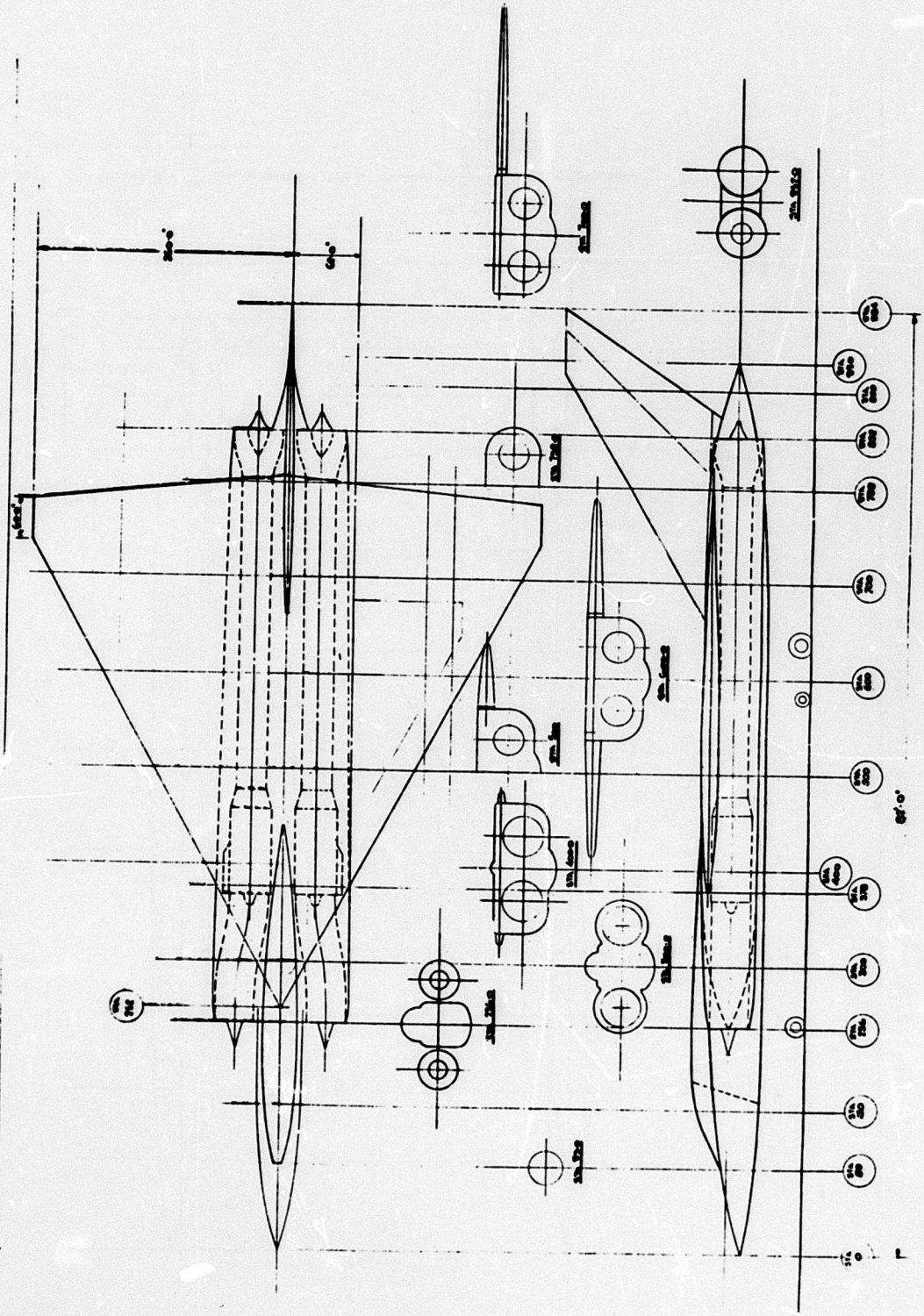
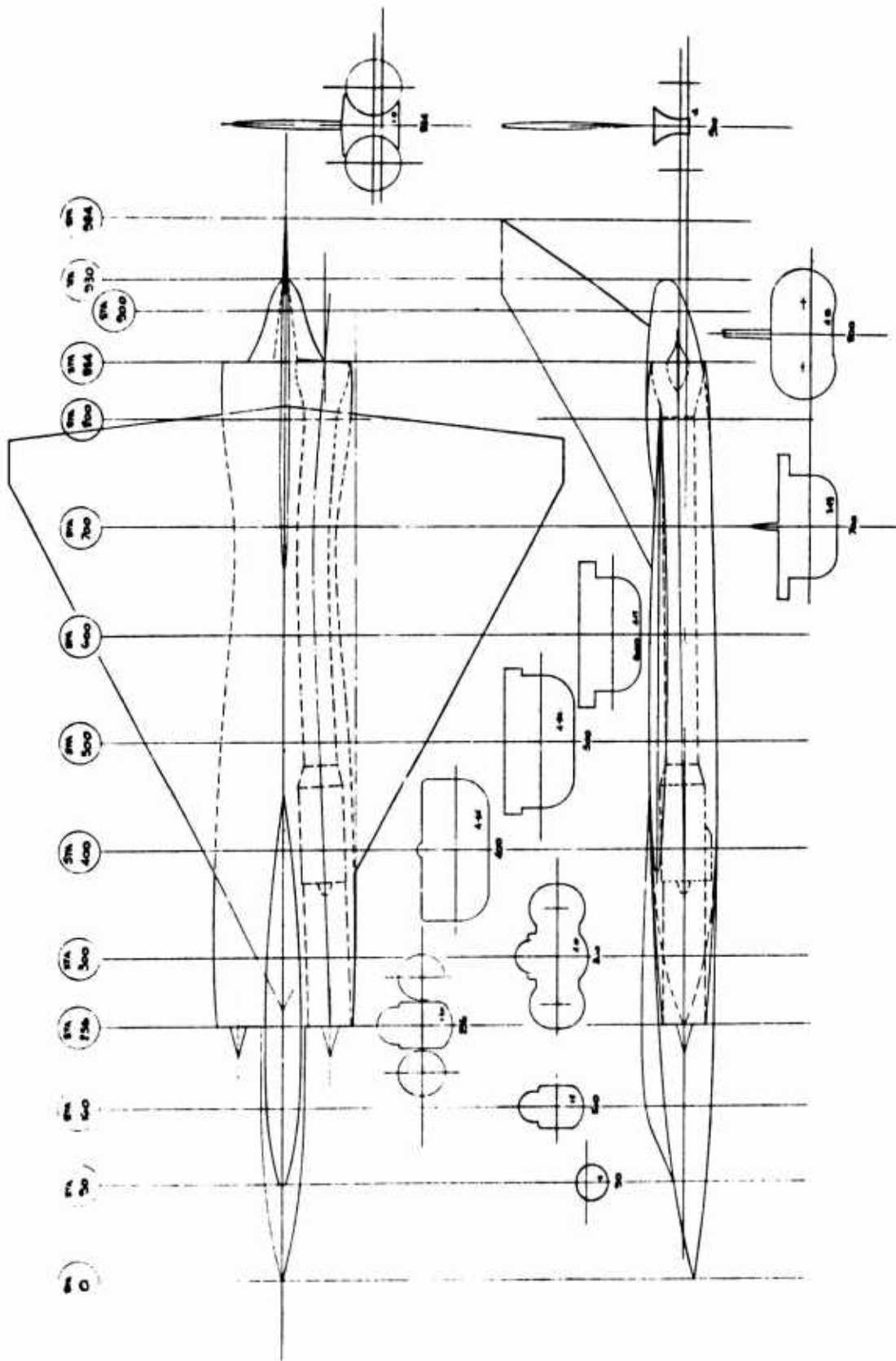


Figure 4.2-9b. Layout of M=1.0 Area Ruled Example Aircraft



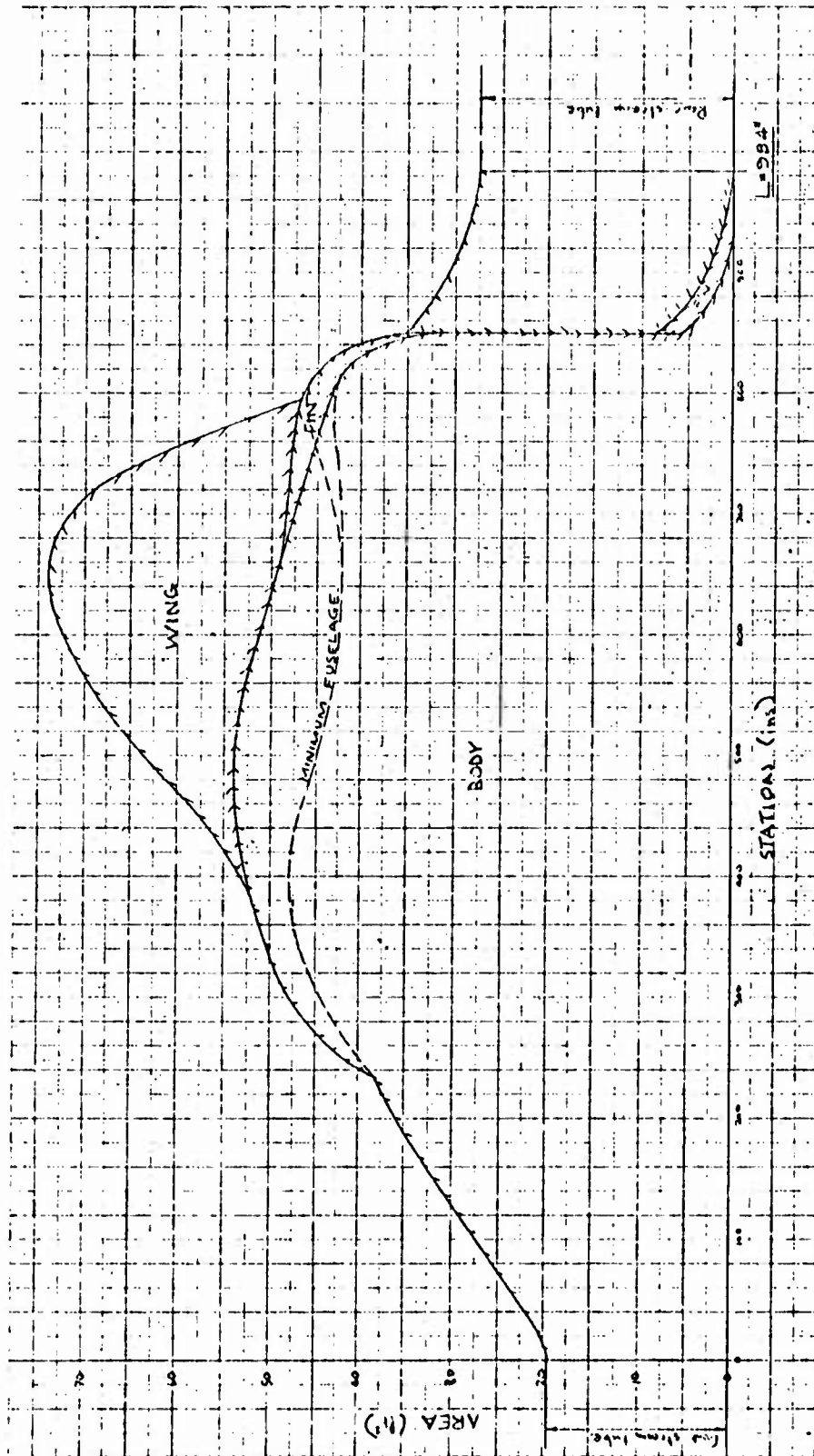


FIGURE 4.2-10 (a). AREA DISTRIBUTION OF BASIC A/C

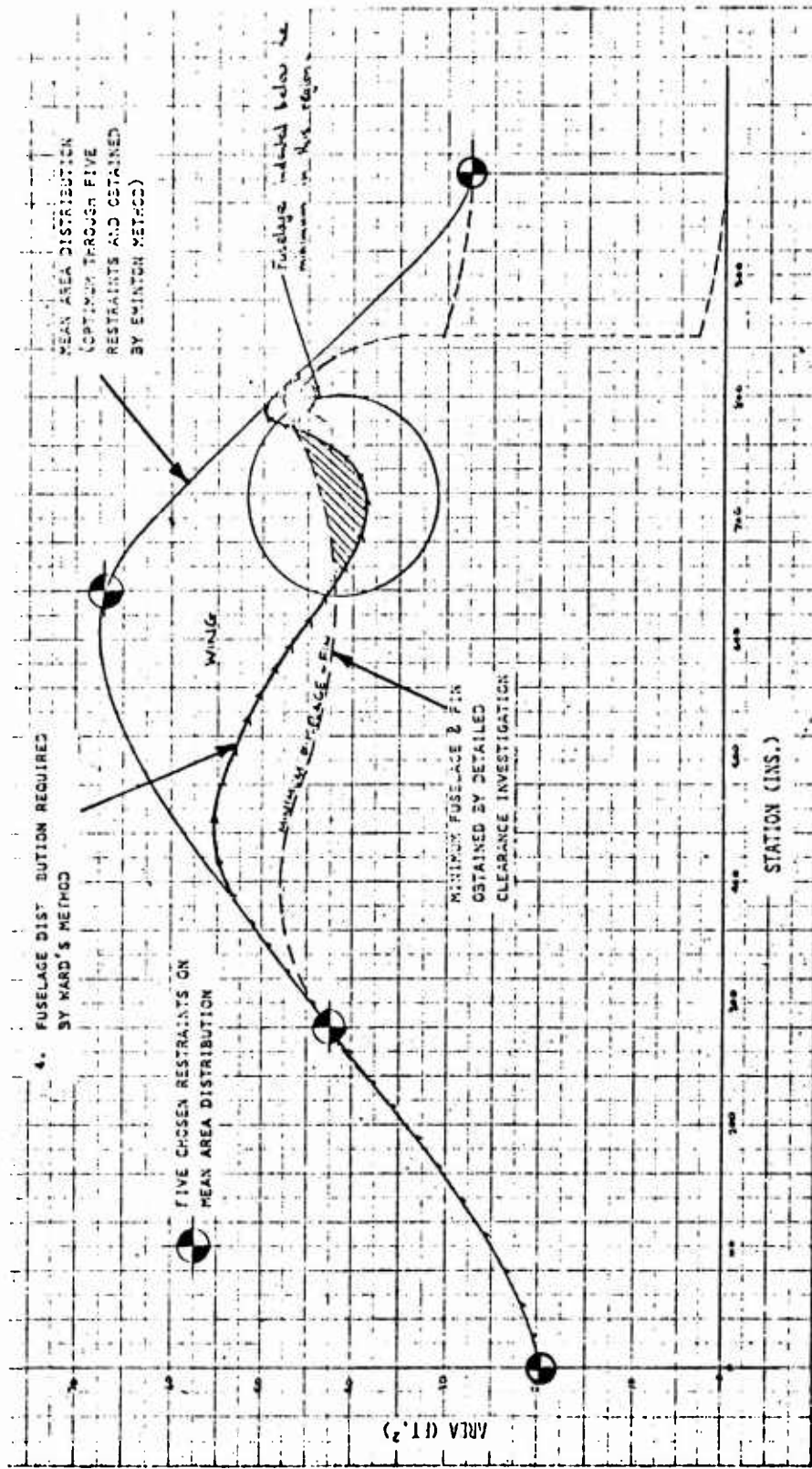


FIGURE 4.2-10(b). AREA DISTRIBUTION OF M = 1.0 AIRCRAFT

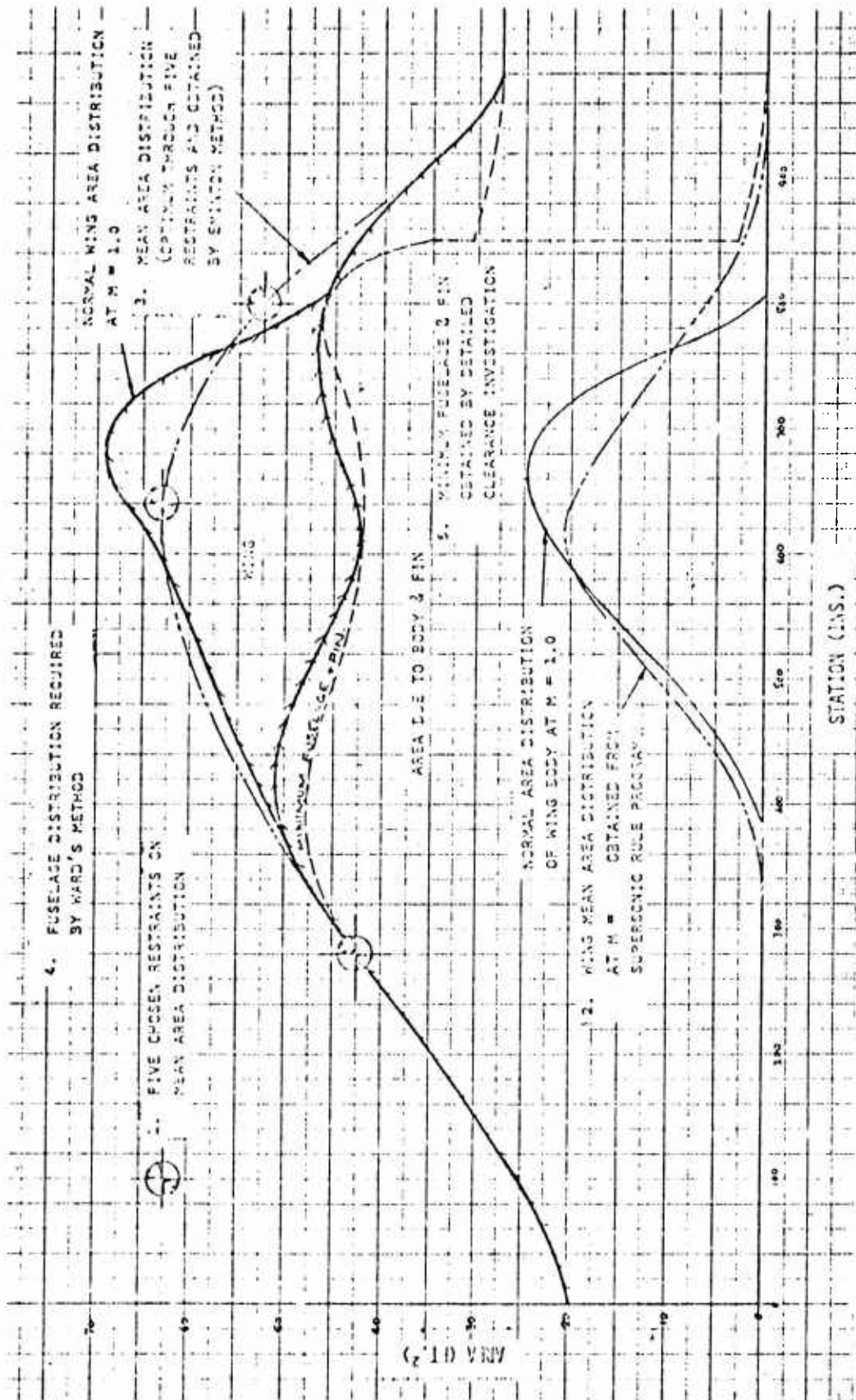


FIGURE 4.2-10(c). AREA DISTRIBUTION OF $M = 1.2$ AIRCRAFT

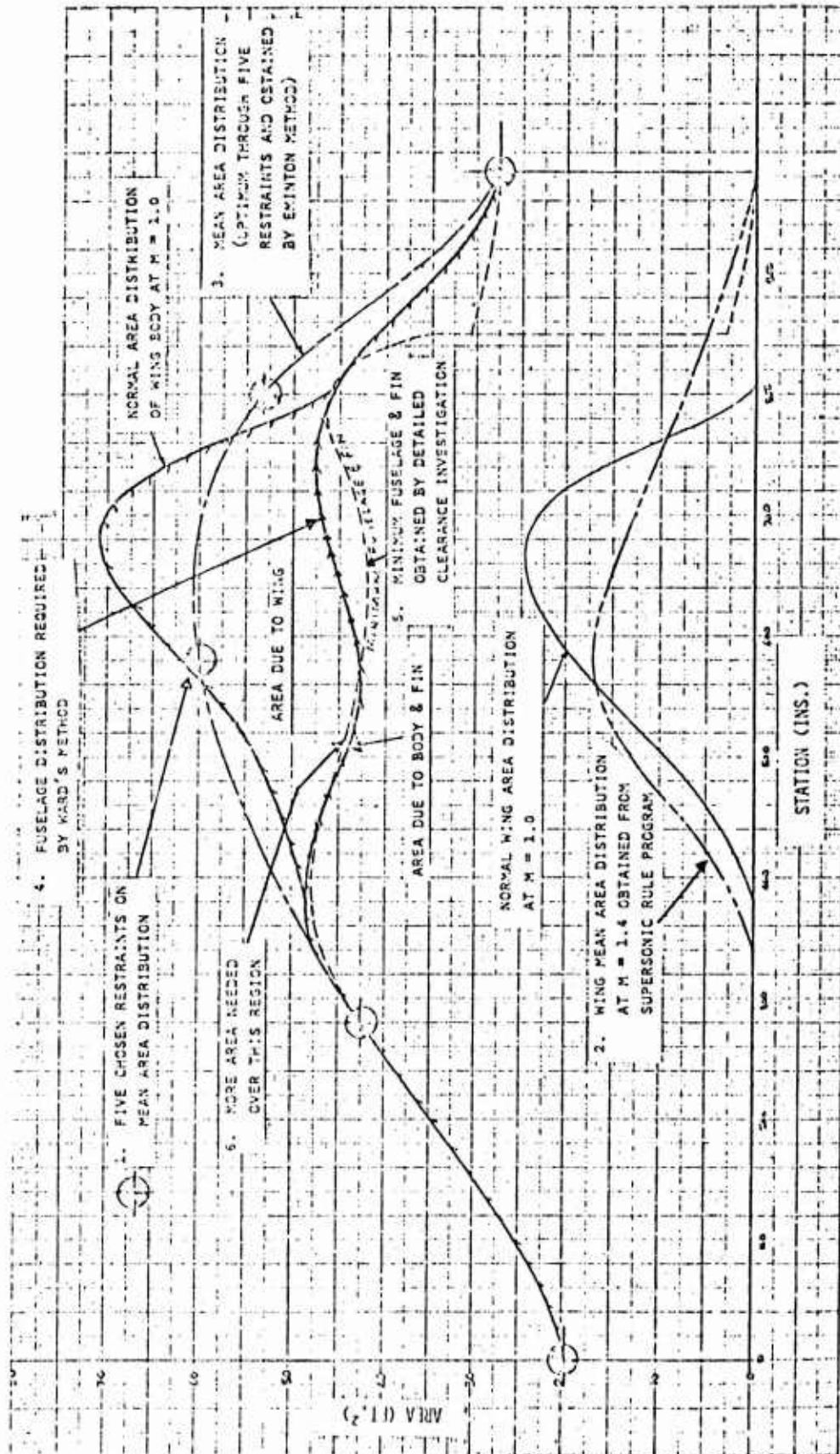


FIGURE 4.2-10(d). AREA DISTRIBUTION OF M = 1.4 AIRCRAFT

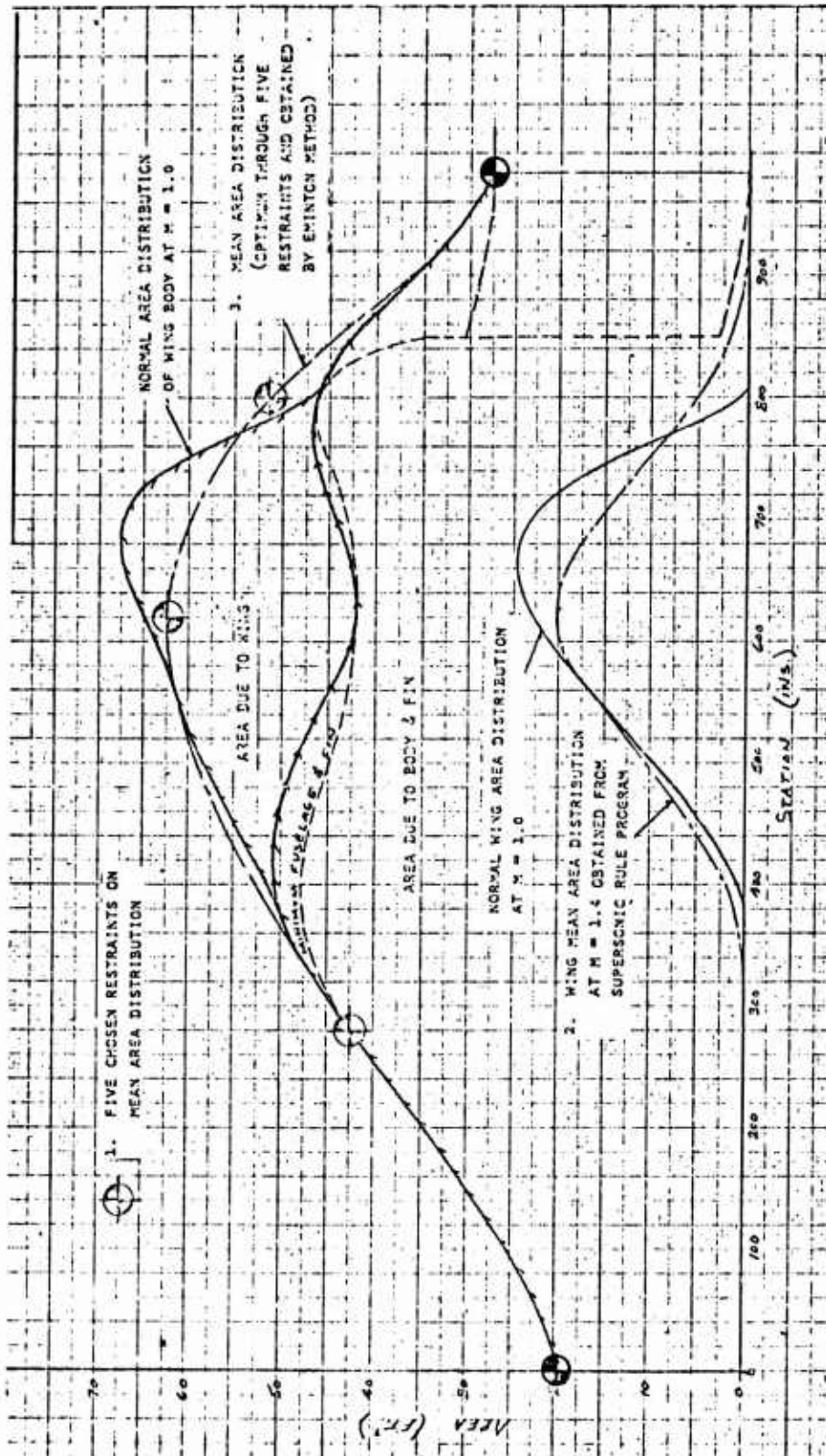


FIGURE 4.2-10(e). AREA DISTRIBUTION OF $M = 1.0$ TO $M = 1.4$ AIRCRAFT

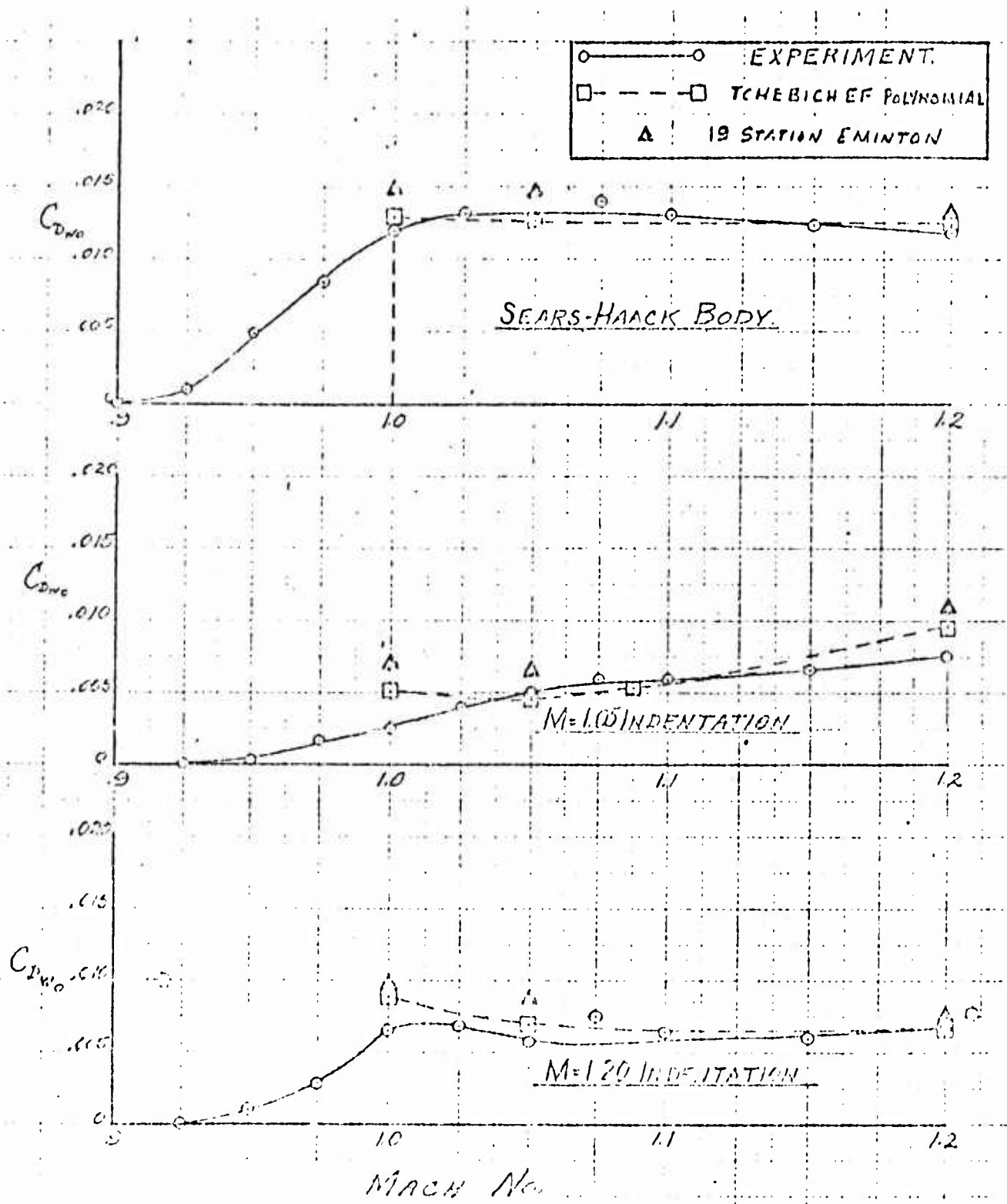


FIGURE 4.2-11(a). EXPERIMENTAL AND COMPUTED ZERO LIFT WAVE DRAG COEFFICIENTS WITH VARIOUS BODIES
 4.2-34

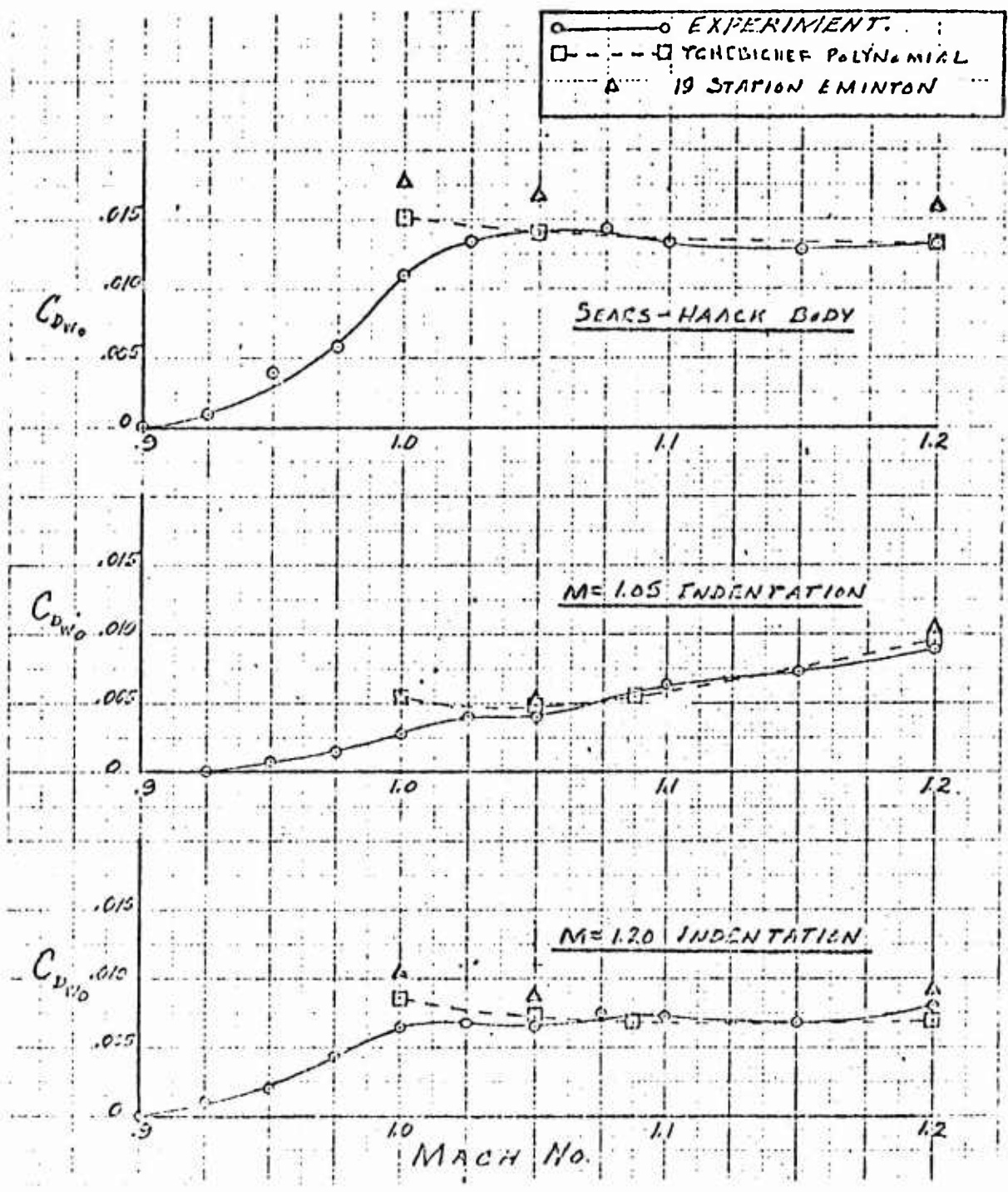


FIGURE 4.2-11(b) EXPERIMENTAL AND COMPUTED ZERO LIFT WAVE DRAG COEFFICIENTS FOR MODIFIED WING WITH VARIOUS BODIES

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 References 1 and 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 *common 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 drags
- 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,

$$\begin{aligned} \Delta P = & \frac{\partial P}{\partial V_1} \Delta V_1 + \frac{\partial P}{\partial V_2} \Delta V_2 + \dots + \frac{\partial P}{\partial V_n} \Delta V_n \\ & + \frac{\partial P}{\partial M} \Delta M + \frac{\partial P}{\partial h} \Delta h + \frac{\partial P}{\partial \delta e} \Delta \delta e + \frac{\partial P}{\partial \alpha} \Delta \alpha \end{aligned} \quad (4.3.1)$$

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.

$$C_{LB} = C_{LP} + C_{LV} \quad (4.3.2)$$

$$C_{LP} = (C_{N\alpha})_b (k_2 - k_1) F_m \alpha' \frac{S_b}{S} \quad (4.3.3)$$

$$C_{LV} = C_{dc} F_m \alpha'^2 \frac{S_p}{S} \quad (4.3.4)$$

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

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

S_b = base area

S_p = planform area

S = reference area

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

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

$$C_{L\alpha_B} = (C_{N\alpha})_b (k_2 - k_1) F_m \frac{S_b}{S} + 2C_{dc} F_m \alpha' \frac{S_p}{S} \quad (4.3.5)$$

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 $C_{L\alpha}$ is based on the expression for normal force curve slope at zero lift contained in the subsection 4.8.1 of DATCOM.

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

$$C_{L_B} = C_{L\alpha_B} \alpha' \quad (4.3.7)$$

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.

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

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

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

$$S_{WE} = \frac{(b_w - W)}{2} (C_{R_{WE}} + C_{T_W}) \quad (4.3.11)$$

$$AR_{WE} = (b_w - W)^2 / S_{WE} \quad (4.3.12)$$

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.

$$C_{L\alpha_H} = C_{L\alpha_{HE}} [K_{B(w)} + K_{w(\delta)}] (1 - d\epsilon/d\alpha) \rho_H/g \quad (4.3.13)$$

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,

$$\Delta C_{L_H} = C_{L\delta_H} \delta_H \quad (4.3.14)$$

$$C_{L\delta_H} = C_{L\alpha_{HE}} [K_{B(w)} + K_{w(\delta)}] \rho_H/g \quad (4.3.15)$$

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

$$\Delta C_{L_e} = C_{L\delta_e} \delta_e \quad (4.3.16)$$

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

$$C_{L_{WB}} = C_{L\alpha_{WB}} \alpha' + \Delta C_{L_H} \quad (4.3.17)$$

$$C_{L\alpha_{WB}} = C_{L\alpha_B} + C_{L\alpha_W} + C_{L\alpha_H} \quad (4.3.18)$$

$$\alpha' = \alpha - \alpha_{L0_{WB}} \quad (4.3.19)$$

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

$$C_{L_{WB}} = (C_{L_{\alpha_B}} + C_{L_{\alpha_H}}) \alpha' + C_{L_{\alpha_W}} (.13963 - \alpha_{L_{0_{WB}}}) + \Delta C_{L_H} \quad (4.3.20)$$

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,

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

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,

$$C_{D_{MIN}} = \sum \Delta C_{D_{MIN}} + C_{D_b} \quad (4.3.22)$$

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

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

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

$$C_f = C_{f_i} \left[\frac{C_f}{C_{f_i}} \right] \quad (4.3.24)$$

$$C_{f_i} = 0.455 / (1.09_{10} RN)^{2.58} \quad (4.3.25)$$

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

$$RN = \frac{a}{\nu} (M l_B) \quad (4.3.26)$$

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

$$RN_{lim} = K_1 \left(\frac{l_B}{A} \right)^{1.0489} \quad (4.3.27)$$

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

| Mach No. | K_1 |
|----------|---------|
| 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.

$$\Delta C_{D_{MIN}} = 2C_{f_i} \left(1 + 2 \frac{k}{c} \right) \frac{S_f}{S} \quad (4.3.28)$$

where C_{f_i} is determined as above with length being replaced by the mean geometric chord in determining Reynolds number

$$\bar{C} = \frac{2}{3} \left[C_{RE} + C_T - \frac{C_{RE} \times C_T}{C_{RE} + C_T} \right] \quad (4.3.29)$$

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

$$C_{D_b} = -C_{P_{B_0}} \frac{S_b}{S} \quad (4.3.30)$$

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

where S_b and the required base geometry are obtained from the input. The term $C_{P_{B_0}} / (\rho / 2\sqrt{\pi S_b})$ 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

$$C_{D_L} = C_L \alpha' \quad (4.3.32)$$

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

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

where K is the drag due to lift factor calculated from

$$K = 1/C_{L\alpha} \quad (4.3.34)$$

and ΔC_L is the polar displacement.

4.3.3.3(b) Subsonic Wing-Body Configurations

The test data indicates that the drag polar of wing-body configurations at a lower lift than that at which wing stall occurs will break. 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

$$C_{L_{BR}} = C_{L\alpha} (.08727 - \alpha_{LOWB}) \quad (4.3.35)$$

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_{LBR}

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

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

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

$$e = e_w [1.0 - (1.18 - .681 C_T/C_{RT}) \frac{W}{b}] \quad (4.3.38)$$

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

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,

$$AR_T = \frac{b^2}{S_T} \quad (4.3.40)$$

where

$$S_T = \frac{b}{2} (C_{RT} + C_T) \quad (4.3.41)$$

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

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

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

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.

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

$$K_T = K \cdot F_{TD} \quad (4.3.45)$$

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

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

$$C_{L(Y/D)_{\max}} = \sqrt{|C_{D_{\min}}/K_T| + \Delta C_L^2} \quad (4.3.47)$$

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 of 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,

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

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 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

$$C_{D_{WB}} = (C_{D_{PN}} + C_{D_{PBT}}) \frac{A_{YZ}}{S} \quad (4.3.48)$$

where

$C_{D_{PN}}$ = function of nose fineness ratio and β

$C_{D_{PBT}}$ = 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.

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

$$\beta C_{D_w} / f(a, \lambda) (t/c)^2 = 0.768 \beta \cot \lambda_{.5c} + 0.0718 (\beta \cot \lambda_{.5c})^2 \quad (4.3.50)$$

The term $f(a, \lambda)$ 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.

$$C_{D_w} = \left\{ 7.85 \cos^2 \lambda_{.25c} - [2.05(M-1)][1 - \tan \lambda_{.25c}] \right\} (t/c)^2 \frac{S_p}{S} \quad (4.3.51)$$

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.

$$C_{D_f} = \sum \Delta C_{D_f} \quad (4.3.54)$$

$$\Delta C_{D_f} = C_f \frac{S_{wet}}{S} \quad (4.3.55)$$

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

$$C_{D_L} = C_L \alpha' \quad (4.3.56)$$

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

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

$$K = 1/C_{L\alpha} \quad (4.3.58)$$

where ΔC_L is the drag polar lift coefficient displacement obtained from the input, and $C_{L\alpha}$ 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, ΔC_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,

$$\beta \cot \Lambda \geq 1, \quad K = 1/C_{L\alpha} \quad (4.3.59)$$

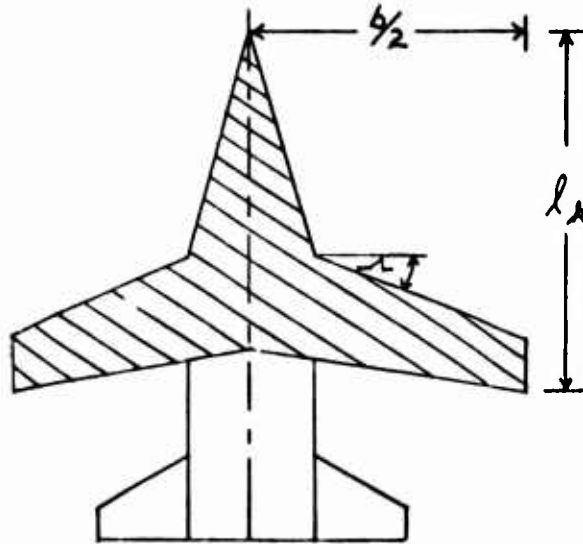
$$\beta \cot \Lambda < 1, \quad K = \frac{P+1}{P(\pi+4)AR} \left[1 + 2\beta^2 \left(\frac{b/2}{l_A} \right)^2 \right] \quad (4.3.60)$$

where

$$P = S_A / b l_A \quad (4.3.61)$$

$$AR = b^2 / S_A \quad (4.3.62)$$

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.

$$C_L = C_N \cos \alpha - C_A \sin \alpha \quad (4.3.63)$$

$$C_D = C_N \sin \alpha + C_A \cos \alpha \quad (4.3.64)$$

$$C_N = C_{N_N} + C_{N_L} + C_{N_{LE}} + C_{N_{LR}} + C_{N_{LER}} + C_{N_e} + C_{N_{LEF}} + C_{N_{B_1}} + C_{N_{B_2}} \quad (4.3.65)$$

$$C_A = C_{A_N} + C_{A_L} + C_{A_{LE}} + C_{A_{LR}} + C_{A_{LER}} + C_{A_e} + C_{A_{LEF}} + C_{A_{B_1}} + C_{A_{B_2}} \quad (4.3.66)$$

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

$$C_{N_L} = K_L S_L / S \sin^2 \alpha \quad (4.3.68)$$

$$C_{N_{LE}} = F_{LE} K_{LE} \sin \alpha [\cos(\lambda_{LE})_E + \cos \lambda_{LE} \cos \alpha] \quad (4.3.69)$$

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

$$C_{N_{LER}} = F_{LER} K_{LER} \sin(\alpha + \Delta_R) \cos \Delta_R [\cos(\lambda_{LER})_E + \cos \lambda_{LER} \cos(\alpha + \Delta_R)] \quad (4.3.71)$$

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

$$C_{N_{LEF}} = -F_{LEF} K_{LEF} \sin^2(\Gamma - \alpha) \cos \Gamma \quad (4.3.73)$$

$$C_{N_{B_1}} = -0.125 F_{B_1} \frac{A_{xy1}}{S} \quad (4.3.74)$$

$$C_{N_{B_2}} = -0.125 F_{B_2} \frac{A_{xy2}}{S} \quad (4.3.75)$$

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

$$C_{A_{LR}} = \frac{G_L}{\sqrt{RN}} \frac{S_{WT}}{S} \left[0.45 \cos \alpha + 4.65 \frac{V_{\infty}}{10,000} \sin \alpha \cos^2 \alpha \right] \quad (4.3.77)$$

$$C_{A_{LE}} = \frac{G_L}{(RN)^2} \frac{S_{WT}}{S} \left[0.048 \sin(4.5\alpha) + 0.7 \frac{V_{\infty}}{10,000} \sin^{1.5} \alpha \cos^{2.25} \alpha \right] \quad (4.3.78)$$

$$C_{A_{LER}} = \frac{1}{2} F_{LER} K_{LER} \cos \lambda_{LER} [\cos(\lambda_{LER})_E + \cos \lambda_{LER} \cos \alpha]^2 \quad (4.3.79)$$

$$C_{A_{LR}} = K_{LR} S_{LR}/S \sin^2(\alpha + \Delta_R) \sin \Delta_R \quad (4.3.80)$$

$$C_{A_{LER}} = \frac{1}{2} F_{LE} K_{LE} \cos \lambda_{LER} \cos \Delta_R \left[\cos(\lambda_{LER}) + \cos \lambda_{LER} \cos(\alpha + \Delta_R) \right]^2 \quad (4.3.81)$$

$$C_{A_e} = K_e S_e/S \sin^2(\alpha + \Delta_e) \sin \Delta_e \quad (4.3.82)$$

$$C_{A_{LEF}} = F_{LE} K_{LE} \sin^2(\Gamma - \alpha) \sin \Gamma \quad (4.3.83)$$

$$C_{A_{B_1}} = 0.196 F_{B_3} \frac{A_{xy1}}{S} \tan \Delta_1 \quad (4.3.84)$$

$$C_{A_{B_2}} = 0.196 F_{B_4} \frac{A_{xy2}}{S} \tan \Delta_2 \quad (4.3.84)$$

$$F_N = \pi R_N^2 / 4S \quad (4.3.86)$$

$$F_{LE} = 4 R_{LE} \lambda_{LE} / 3S \quad (4.3.87)$$

$$F_{LER} = 4 R_{LER} \lambda_{LER} / 3S \quad (4.3.88)$$

$$F_{LEF} = 4 R_{LEF} \lambda_F / 3S \quad (4.3.89)$$

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

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

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

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

$$RN = \frac{V_{\infty} l}{V} \quad (4.3.94)$$

$$\cos(\lambda_{LE}) = \left[1 + \sin^2 \lambda_{LE} \cos^2 \alpha \right]^{1/2} \quad (4.3.95)$$

$$\cos(\lambda_{LER}) = \left[1 + \sin^2 \lambda_{LER} \cos^2(\alpha + \Delta_R) \right] \quad (4.3.96)$$

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 Reference 1 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. 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 II, AFFDL-TR-66-77, October 1966.
3. Hoak, D. E. and Carlson, J. W., Air Force Project Engineers for USAF Stability and Control DATCOM.
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 |
|---|---|
| <p>I - Modified body of revolution with horizontal and vertical tails</p> | <p>Subsonic, supersonic, and hypersonic</p> |
| <p>IB - Type I with a wing</p> | <p>Subsonic and supersonic</p> |
| <p>II - Planar lifting body with elevons and vertical tail</p> | <p>Subsonic, supersonic, and hypersonic</p> |
| <p>IIB - Type II with a wing</p> | <p>Subsonic and supersonic</p> |

TABLE 4.3-2. AVAILABLE SUBSONIC SENSITIVITIES IN PROGRAM TREND

● CONFIGURATION I

| INPUT VARIABLE | STABILITY AND CONTROL DERIVATIVE | | | | | | | | | | | | | |
|------------------|----------------------------------|----------------|-----------------|----------------|-----------------|-----------|---------------|-----------------|-----------|---------------|-----------------|-----------|---------------|-----------------|
| | A.C. | | C _{mp} | | C _{yp} | | | C _{np} | | | C _{sp} | | | C _{yp} |
| | 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 _y | | X | | X | | X | X | | | | | | X | |
| b _x | | X | | X | | | | | | | | | | X |
| b _z | | X | | X | | | | | | | | | X | |
| C _{yβ} | | X | | X | | X | X | | | | | | X | |
| C _{yo} | | X | | X | | | | | | | | | | X |
| C _{zβ} | | X | | X | | | | | | | | | | X |
| r | | | | | | X | | | | | | | X | |
| b _z | | | | | | X | | | | | | | X | |
| β _y | X | X | X | X | | | | X | X | | | | | |
| λ _y | | | | | | | | X | | X | | | X | |
| Δλ _{yβ} | | | | | | | | X | | X | | | X | |
| λ _z | | X | | X | | X | | | X | | | | X | |
| Δλ _{zβ} | | X | | X | | X | | | X | | | | X | |
| λ _β | | X | | X | | X | | | | | | | | X |
| Δλ _{zβ} | | X | | X | | X | | | | | | | | X |
| r _y | | | | | | | | X | | | X | | X | |
| w _y | | | | | | | | | | | | | X | |
| w _z | | X | | X | | | | | | | | | | X |
| b _z | | | | | | X | X | | X | X | | | X | |
| C _z | | | | | | X | X | | X | X | | | | |
| dx/dα | | X | | X | | | | | | | | | | |
| dy/dα | | X | | X | | | | | | | | | | |

● CONFIGURATION II

| INPUT VARIABLE | STABILITY AND CONTROL DERIVATIVE | | | | | | | | | | | | | |
|----------------------------------|----------------------------------|----------------|-----------------|----------------|-----------------|-----------|---------------|-----------------|-----------|---------------|-----------------|----------------|---------------|-----------------|
| | A.C. | | C _{mp} | | C _{yp} | | | C _{np} | | | C _{sp} | | | C _{yp} |
| | BODY | WING-BODY-TAIL | BODY | WING-BODY-TAIL | BODY | WING-BODY | VERTICAL TAIL | BODY | WING-BODY | VERTICAL TAIL | BODY | WING-BODY-TAIL | VERTICAL TAIL | ELLEVON |
| α _c | | | | | | | | | | | X | X | | |
| b _y | X | X | X | X | | | | X | X | | X | X | | X |
| b _x | | X | | X | | | | | | X | | | X | |
| b _z | | X | | X | | | | | | | | | X | |
| C _{yβ} | X | X | X | X | | | | X | X | | X | X | | |
| C _{yo} | | X | | X | | | | | | | | | X | |
| C _{zβ} | | X | | X | | X | | | X | | | | | X |
| C _{zo} | | | | | | | | | | | | | | X |
| C _{zo} | | | | | | | | | | | | | | X |
| β _y | | | | | | X | X | | X | X | | | X | |
| r | | | | | | X | | | | | X | X | | |
| b _z | | | | | | X | | | | | X | X | | |
| λ _z | X | X | X | X | | | | X | X | | X | X | | X |
| Δλ _{zβ} | X | X | X | X | | | | X | X | | X | X | | X |
| λ _y | | | | | | | | X | | X | | | X | |
| Δλ _{yβ} | | | | | | | | X | | X | | | X | |
| λ _β | | X | | X | | X | | | X | | | | X | |
| Δλ _{zβ} | | X | | X | | X | | | X | | | | X | |
| E _{zβ} | | | | | | X | X | | X | X | | | X | |
| E _{zβ} /β _{zβ} | | | | | | X | X | | X | X | | | X | |
| E _{zβ} /β _{zβ} | | | | | | X | X | | X | X | | | X | |
| E _{zβ} | | | | | | X | | | | | | | | X |

TABLE 4.3-3. AVAILABLE SUPERSONIC SENSITIVITIES IN PROGRAM TREND

● CONFIGURATION I

| INPUT VARIABLE | STABILITY AND CONTROL DERIVATIVE | | | | | | | | | | | | | |
|------------------|----------------------------------|-----------------|-----------------|-----------------|-----------------|------------|---------------|-----------------|------------|---------------|-----------------|------------|---------------|------------------|
| | A.C. | | C _{mα} | | C _{yp} | | | C _{yn} | | | C _{ya} | | | C _{ymδ} |
| | BODY | VINC. BODY-TAIL | BODY | VINC. BODY-TAIL | BODY | VINC. BODY | VERTICAL TAIL | BODY | VINC. BODY | VERTICAL TAIL | BODY | VINC. BODY | VERTICAL TAIL | HORIZONTAL TAIL |
| $\dot{\alpha}$ | | X | | X | | X | | | X | | | X | | |
| $\dot{\beta}$ | | | | | | | X | | | X | | | X | |
| $\dot{\gamma}$ | | X | | X | | X | | | X | | | X | | |
| $\dot{\delta}$ | | | | | | | X | | | X | | | X | |
| $\dot{\epsilon}$ | | | | | | X | | X | | | | | X | |
| $\dot{\zeta}$ | | | | | | X | | X | | | | | X | |
| $\dot{\eta}$ | | | | | | X | | X | | | | | X | |
| $\dot{\theta}$ | X | X | X | X | X | X | | X | X | | | | | |
| $\dot{\phi}$ | X | X | X | X | X | X | | X | X | | | | | |
| $\dot{\psi}$ | | X | | X | | | | | | | | X | | |
| $\dot{\chi}$ | | | | | | | X | | | X | | | X | |
| $\dot{\psi}_w$ | | | | | | | | | | | | | X | |
| $\dot{\psi}_w$ | X | X | X | X | | | | | X | | | | | |
| $\dot{\psi}_w$ | X | X | X | X | | X | | | X | | | | | |
| $\dot{\psi}_w$ | | X | | X | | | | | | | | | | X |
| $\dot{\psi}_w$ | | X | | X | | | | | | | | | | X |
| $\dot{\psi}_w$ | | X | | X | | | | | | | | | | X |

● CONFIGURATION II

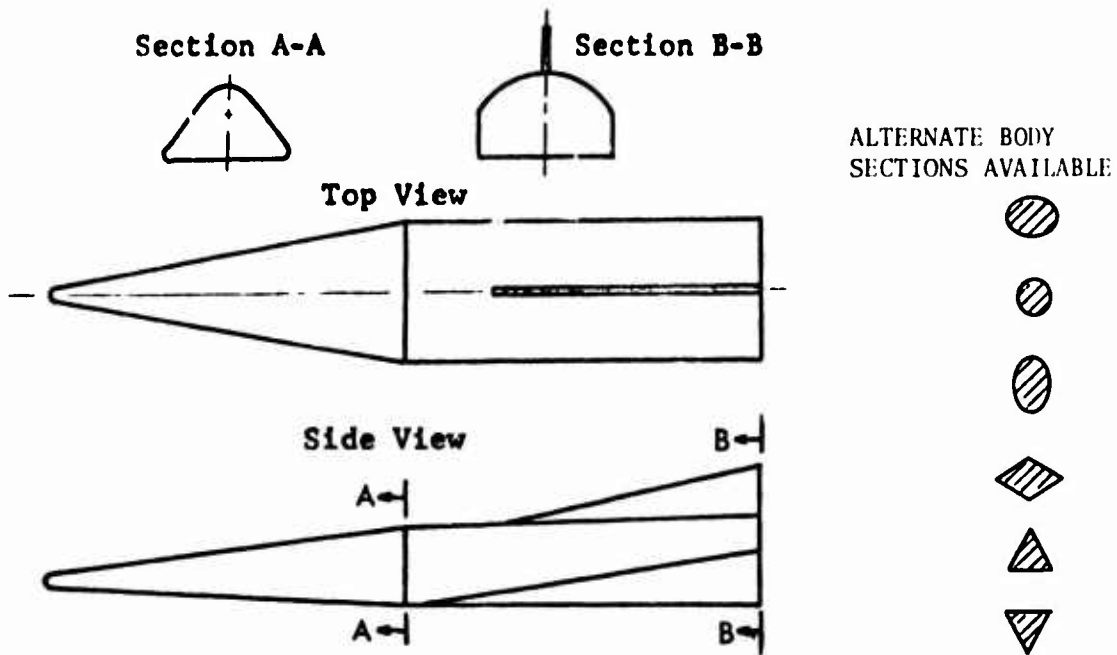
| INPUT VARIABLE | STABILITY AND CONTROL DERIVATIVE | | | | | | | | | | | | | |
|------------------|----------------------------------|------------|-----------------|------------|-----------------|------------|---------------|-----------------|------------|---------------|-----------------|------------|---------------|------------------|
| | A.C. | | C _{mα} | | C _{yp} | | | C _{yn} | | | C _{ya} | | | C _{ymδ} |
| | BODY | VINC. BODY | BODY | VINC. BODY | BODY | VINC. BODY | VERTICAL TAIL | BODY | VINC. BODY | VERTICAL TAIL | BODY | VINC. BODY | VERTICAL TAIL | ELEVONS |
| $\dot{\alpha}$ | | | | | | | | | | | X | X | | |
| $\dot{\beta}$ | | X | | X | | X | | | X | | | X | | |
| $\dot{\gamma}$ | | | | | | | X | | | X | | | X | |
| $\dot{\delta}$ | | X | | X | | X | | | | | | | X | |
| $\dot{\epsilon}$ | | | | | | | | | | | | | | X |
| $\dot{\zeta}$ | | | | | | | | | | | | | | X |
| $\dot{\eta}$ | | | | | | | | | | | | | | |
| $\dot{\theta}$ | | | | | | | | | | | | | | |
| $\dot{\phi}$ | X | X | X | X | X | X | | X | X | | | | | |
| $\dot{\psi}$ | X | X | X | X | X | X | | X | X | | | | | |
| $\dot{\psi}_w$ | | | | | | | | | | | | | | |
| $\dot{\psi}_w$ | | X | | X | | | | | | | | | | |
| $\dot{\psi}_w$ | | X | | X | | | | | | | | | | |
| $\dot{\psi}_w$ | | | | | | | X | | | X | | | X | |
| $\dot{\psi}_w$ | | | | | | | X | | | X | | | X | |
| $\dot{\psi}_w$ | X | X | X | X | | | | | X | | | | | |
| $\dot{\psi}_w$ | X | X | X | X | | X | | | X | | | | | |
| $\dot{\psi}_w$ | | | | | | | | | | | | | | X |
| $\dot{\psi}_w$ | | | | | | | | | | | | | | X |

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

| STABILITY & CONTROL DERIVATIVE | ELEMENT | INPUT VARIABLE | | | | | | | | | | | | | | |
|--------------------------------|---------|----------------|-------|-----|-----|-----|---------------|---------------|---------------|----------|----------|----------|-----------|--------|----------|----------|
| | | α | C_m | X | Y | Z | Δ_{LE} | δ_{LE} | δ_{LE} | δ | Δ | Γ | λ | ϕ | A_{XX} | A_{YY} |
| M | 1 | | X | X | | | | | | | | | | | | |
| | 2 | | | X | | | X | X | X | | | | | | | |
| | 3 | | | X | | | X | X | 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 B | 1 | X | X | X | | X | | | | | | | | | | |
| | 2 | X | | X | | X | X | X | | | | | | | | |
| | 3 | X | | X | | X | | | X | | | | | | | |
| | 4 | 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 _{yp} | 1 | X | X | | | | X | X | X | | | | | | | |
| | 2 | X | | | | | X | X | X | | | | | | | |
| | 3 | X | | | | | X | X | X | X | | | | | | |
| | 4 | X | | | X | X | X | X | X | | | | | | | |
| | 5 | X | | | | X | | | | X | X | | | | | |
| | 6 | X | | | X | X | | | | X | | X | X | | | |
| | 7 | X | | | | X | | | | X | | | | | X | X |
| | 8 | X | | | | X | | | | X | | | | | X | X |
| C _{yp} | 1 | X | X | X | X | | X | X | X | | | | | | | |
| | 2 | X | | X | X | | X | X | X | | | | | | | |
| | 3 | X | | X | X | | X | X | X | X | | | | | | |
| | 4 | X | | X | X | | X | X | X | | | | | | | |
| | 5 | X | | X | X | | | | | X | X | | | | | |
| | 6 | X | | X | X | X | | | | X | | X | X | | | |
| | 7 | X | | X | X | | | | | X | | | | | X | X |
| | 8 | X | | X | X | | | | | X | | | | | X | X |
| C _m AB | 9 | X | | X | | X | | | | | | | | | | X |
| C _m AB | 10 | X | | X | | X | | | | | | | | | | X |

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

CONFIGURATION TYPE I - BODY-ALONE



CONFIGURATION TYPE I - WING-BODY-TAIL

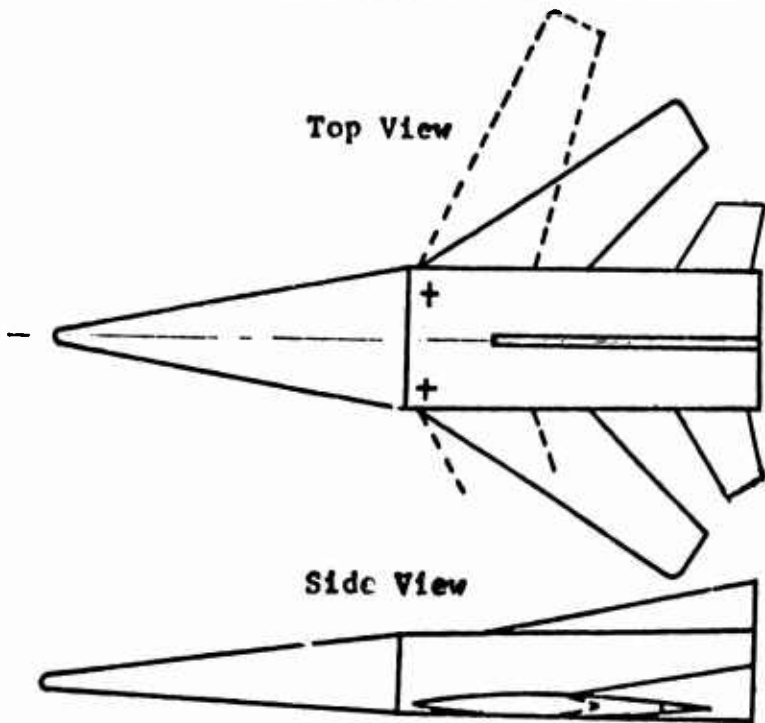
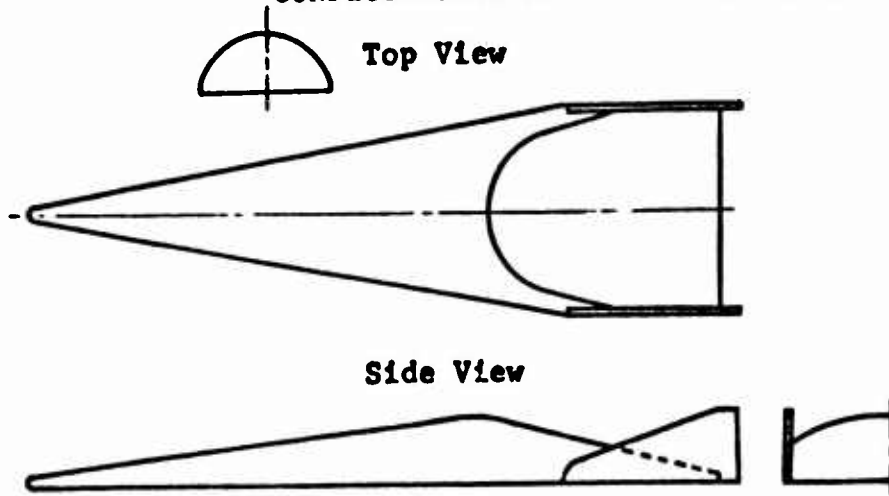


FIGURE 4.3-1. TYPICAL TYPE I CONFIGURATIONS

CONFIGURATION TYPE II - BODY ALONE



CONFIGURATION TYPE II - WING-BODY

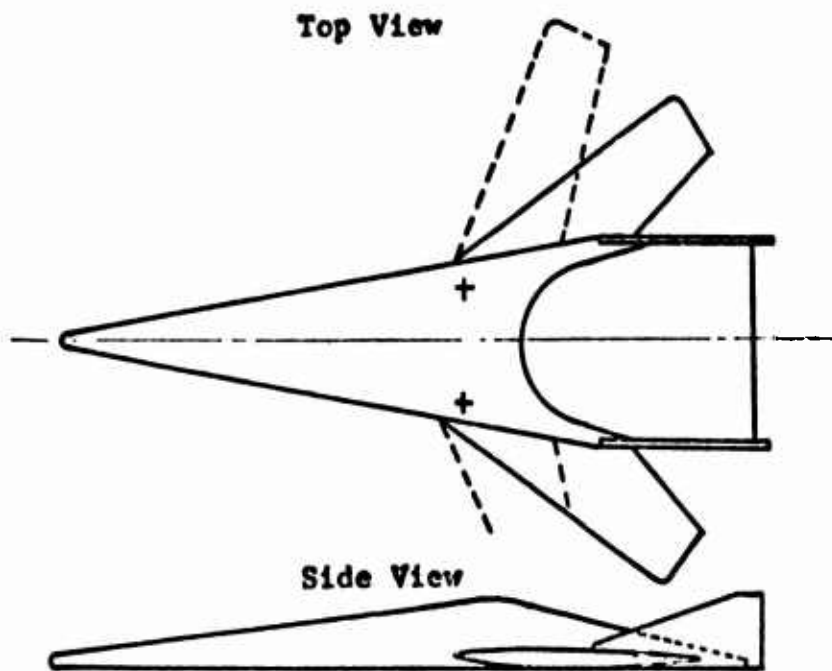


FIGURE 4.3-2. TYPICAL TYPE II CONFIGURATIONS

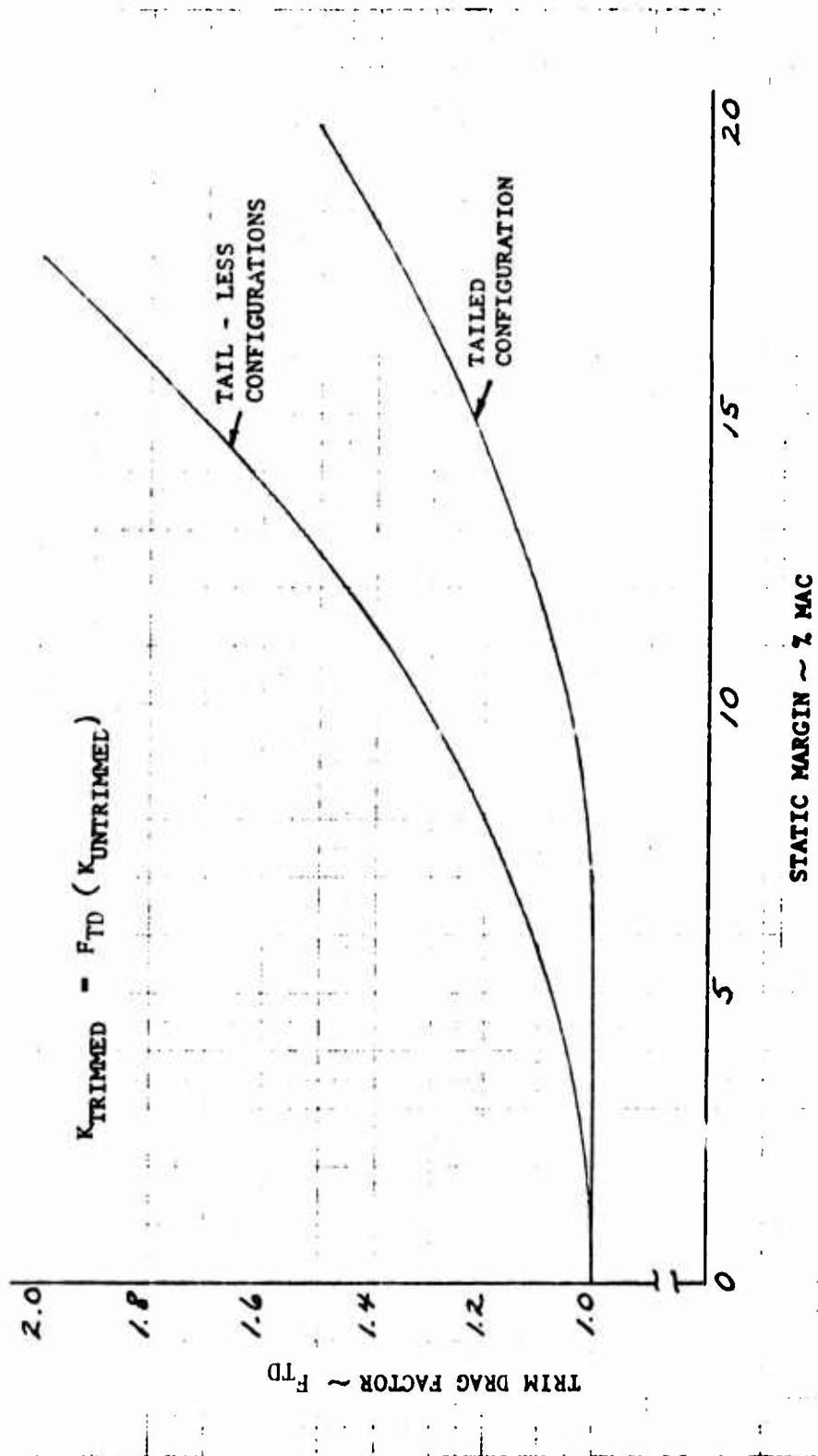


FIGURE 4.3-3 TRIM DRAG CHARACTERISTICS

SECTION 5

PROPULSION

The ODIN/MFV program library installed at Wright-Patterson Air Force Base contains an independent engine cycle analysis program for multi-spool turbojet and turbofan engine cycles with or without afterburners and including off-design point analysis capability. This program, GENENG II, was written at NASA's Lewis Research Center and is a derivative of a United States Air Force Aero Propulsion Laboratory, Wright-Patterson Air Force Base, engine cycle analysis, SMOTE. Complete program documentation for GENENG II is provided in Reference 1.

Two other programs are available for later inclusion in the ODIN/MFV; these programs have the ability to perform the following analyses:

1. Design point performance of single spool turbojet and turbofan engines, Reference 2.
2. Design and off-design performance of one- and two-spool turbojet and turbofan engines, Reference 3.

The GENENG II propulsion program now available in the ODIN/MFV is outlined below. For complete details, reference should be made to the original source document.

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, February 1972.
2. Vanco, Michael R., Computer Program for Design-Point Performance of Turbojet and Turbofan Engines, NASA TM X-1340, February 1967.
3. 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.

5.1 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.

5.1.1 Introduction

Program GENENG II is a derivative of program GENENG of Reference 4. 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 air-streams). 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,

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 5.1.2. The original Fishbach and Koenig GENENG II program was written for the IBM 7094 computer. The GENENG II program contained in the ODIN/MFV program library is a CDC 6600 version constructed at the Naval Air Development Center by Robert Leko.

5.1.2 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.

5.1.2.1 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

$$WA_{F,c} = \frac{WA_F \sqrt{T_2/T_{518.668}}}{P_2/P_{SLS}} \quad (5.1.1)$$

where P_2 and P_{SLS} are atmospheres and P_{SLS} equals 1.0. All symbols are defined in Table 5.1-3 of this section. 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

$$\text{Fan power} = WA_F \times (H_{22} - H_2) \quad (5.1.2)$$

Some fan air may be lost to the cycle as fan bleed Bl_F , which is expressed as a fraction of the fan airflow

$$Bl_F = PC_{Bl,F} \times WA_F \quad (5.1.3)$$

The corrected airflow into the intermediate compressor is

$$WA_{I,c} = \frac{WA_I \sqrt{T_{22}/T_{518.668}}}{P_{22}/1.0} \quad (5.1.4)$$

The remaining air goes through the fan duct where some leakage from the core air may also enter; see Equation 5.3.16.

$$WA_D = WA_F - Bl_F - WA_I + Bl_{DU} \quad (5.1.5)$$

This air, which may be heated by a duct burner to a temperature T_{24} , undergoes a pressure drop

$$P_{25} = P_{24} \times \left[1 - \left(\frac{\Delta P}{P} \right)_{\text{DUCT}} \right] \quad (5.1.6)$$

The air would have been heated by the addition of fuel, which can be expressed as a fuel-air ratio so that

$$WG_{24} = WA_{23} \times [1 + (f/a)_{23}] \quad (5.1.7)$$

The gas is then expanded through a nozzle (Station 29) to produce thrust. The bypass ratio is defined by

$$\text{BYPASS} = \frac{WA_D}{WA_I} \quad (5.1.8)$$

The air going into the intermediate compressor is compressed to the condition at Station 21. The power required is

$$\text{Intermediate-compressor power} = WA_I \times (H_{21} - H_{22}) \quad (5.1.9)$$

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.

$$Bl_I = PC_{Bl,I} \times WA_I \quad (5.1.10)$$

The remainder of the air enters the core compressor

$$WA_C = WA_I - Bl_I \quad (5.1.11)$$

and

$$WA_{C,c} = \frac{WA_C \times \sqrt{T_{21}/T_{518.668}}}{P_{21}^{1.0}} \quad (5.1.12)$$

The air entering the wing duct experiences a pressure drop

$$P_{36} = P_{32} \times \left[1 - \left(\frac{\Delta P}{P} \right)_{WING} \right] \quad (5.1.13)$$

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

$$\text{Core compressor power} = WA_C \times (H_3 - H_{21}) = WA_3 \times (H_3 - H_{21}) \quad (5.1.14)$$

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.

$$Bl_C = BC_{Bl,C} \times WA_3 \quad (5.1.15)$$

$$Bl_{DU} = PC_{Bl,DU} \times Bl_C \quad (5.1.16)$$

$$Bl_{OB} = PC_{Bl,OB} \times Bl_C \quad (5.1.17)$$

$$Bl_{HP} = PC_{Bl,HP} \times Bl_C \quad (5.1.18)$$

$$Bl_{IP} = PC_{Bl,IP} \times Bl_C \quad (5.1.19)$$

$$Bl_{LP} = PC_{Bl,LP} \times Bl_C \quad (5.1.20)$$

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

$$WA_4 = WA_3 - Bl_C \quad (5.1.21)$$

and is heated to a turbine inlet temperature T_4 and goes through a combustor pressure drop $(\Delta P/P)_{\text{COMB}}$. 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

$$WG_4 = WA_4 \times [1 + (f/a)_4] \quad (5.1.22)$$

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

$$WG_4 \times (H_4 - H_{50}) = WA_3 \times (H_3 - H_{21}) + \text{HPEXT} \quad (5.1.23)$$

In addition, the physical speeds must match

$$N_{\text{HP,TURBINE}} = N_{\text{COMP}} \quad (5.1.24)$$

If high pressure turbine bleed air $B1_{\text{HP}}$ is added into the cycle at this point, H_{50} must be readjusted

$$H_{50} = \frac{(B1_{\text{HP}} \times H_3) + WG_4 H_{50}}{WG_4 + B1_{\text{HP}}} = \frac{(B1_{\text{HP}} \times H_3) + WG_4 H_{50}}{WG_{50}} \quad (5.1.25)$$

Similarly,

$$WG_{50} \times (H_{50} - H_5) = WA_I \times (H_{21} - H_{22}) \quad (5.1.26)$$

$$N_{\text{IP,TURBINE}} = N_{\text{INT COMP}} \quad (5.1.27)$$

$$H_5 = \frac{(B1_{\text{IP}} \times H_3) + WG_5 H_{55}}{WG_5 + B1_{\text{LP}}} = \frac{(B1_{\text{LP}} \times H_3) + WG_5 H_{55}}{WG_{55}} \quad (5.1.28)$$

$$WG_5 \times (H_5 - H_{55}) = WA \times (H_{22} - H_2) \quad (5.1.29)$$

$$N_{\text{LP,TURBINE}} = N_{\text{FAN}} \quad (5.1.30)$$

$$H_{55} = \frac{(B1_{\text{LP}} \times H_3) + WG_5 H_{55}}{WG_5 + B1_{\text{LP}}} = \frac{(B1_{\text{LP}} \times H_3) + WG_5 H_{55}}{WG_{55}} \quad (5.1.31)$$

The gas flow WG_{55} then may be heated by an afterburner to a gas temperature T_7 and may undergo a pressure drop.

$$P_7 = P_6 \left[1 - \left(\frac{\Delta P}{P} \right)_{\text{AFTERBURNER}} \right] \quad (5.1.32)$$

The gas flow would be increased by any fuel burned.

$$WG_7 = WG_{55} + WFA \quad (5.1.33)$$

The gas is then expanded through the nozzle (Station 9) to produce the remainder of the total engine thrust.

5.1.2.2 Type b - Two-Spool, Three-Stream, Boosted Fan Turbofan

From Figure 5.1-2 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.

$$N_{\text{INT COMP}} = N_{\text{FAN}} \quad (5.1.34)$$

The power of the low pressure turbine is now

$$WG_{50} \times (H_{50} - H_{55}) = WA_F \times (H_{22} - H_2) + WA_I \times (H_{21} - H_{22}) \quad (5.1.35)$$

$PC_{B1,IP}$ must be zero and H_{55} is readjusted by

$$H_{55} = \frac{(B1_{LP} \times H_3) + WG_{50} H_{55}}{WG_{50} + B1_{LP}} \quad (5.1.36)$$

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.

5.1.2.3 Type c - Two-Spool, Three-Stream Supercharged Compressor Turbofan

Engine c is shown in Figure 5.1-3. 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.

$$N_{COMP} = N_{INT COMP} \quad (5.1.37)$$

$$WG_{50} \times (H_{50} - H_5) = WA_I \times (H_{21} - H_{22}) + WA_C \times (H_3 - H_{21}) + HPEXT \quad (5.1.38)$$

$PC_{B1,HP}$ must be zero and H_5 is readjusted by

$$H_5 = \frac{(B1_{IP} \times H_3) + WG_{50} H_5}{WG_{50} + B1_{IP}} \quad (5.1.39)$$

5.1.2.4 Type d - Three-Spool, Two-Stream Turbofan

Engine d, shown in Figure 5.1-4, 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.

5.1.2.5 Type e - Two-Spool, Two-Stream Turbofan

Engine e is the typical turbofan and is shown in Figure 5.1-5. 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.

5.1.2.6 Type f - Three-Spool, Three-Stream Aft Fan Turbofan

The three-spool, three-stream aft fan engine is shown in Figure 5.1-6. 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

$$WG_{50} \times (H_{50} - H_5) = WA_I \times (H_{21} - H_2) \quad (5.1.40)$$

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.

5.1.2.7 Type g - Two-Spool, Three-Stream Aft Fan Turbofan

Engine g, a counterpart of engine c (Figure 5.3-3) is shown in Figure 5.1-7. The power balance would be

$$WG_{50} \times (H_{50} - H_5) = WA_I \times (H_{32} - H_2) + WA_C \times (H_3 - H_{32}) \quad (5.1.41)$$

5.1.2.8 Type h - Two-Spool, Two-Stream Aft Fan Turbofan

Engine h, a counterpart of engine e, Figure 5.1-5, is shown in Figure 5.1-8. The power balance would be

$$WG_{50} \times (H_{50} - H_5) = WA_C \times (H_3 - H_2) \quad (5.1.42)$$

5.1.2.9 Type i - Three-Spool, Two-Stream Aft Fan Turbofan

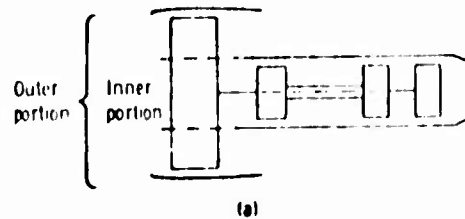
Engine i, a counterpart of engine d, Figure 5.1-4, is shown in Figure 5.1-9. The power balance would be

$$WG_{50} \times (H_{50} - H_5) = WA_I \times (H_{21} - H_2) \quad (5.1.43)$$

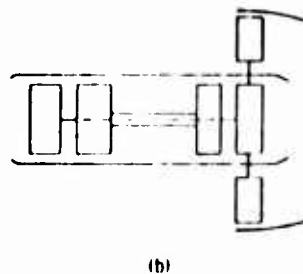
5.1.10 Other Engines

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.



The high bypass ratio turbofan (sketch a) can be simulated by a boosted aft fan engine (sketch b).



5.1.3 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,
- $$ZF = \frac{(\text{Pressure ratio along speed line}) - (\text{Low pressure ratio on speed line})}{(\text{High pressure ratio on speed line}) - (\text{Low pressure ratio on speed line})}$$

2. PCNF - Per cent fan speed or turbine inlet temperature or T4
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 - Per cent inner compressor speed or turbine inlet temperature or T4
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 Reference 4. 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 5.1-1.

5.1.4 Choice of Component Maps - Scaling Laws

The scaling equations used for the compressor maps are

$$PR = \frac{PR_{design} - 1}{PR_{map,design} - 1} (PR_{map} - 1) + 1 \quad (5.1.44)$$

$$WA = \frac{WA_{design}}{WA_{map,design}} \times WA_{map} \quad (5.1.45)$$

$$ETA = \frac{ETA_{design}}{ETA_{map,design}} \times ETA_{map} \quad (5.1.46)$$

Similar equations are used for combustor and turbine map scaling. These equations are found in the appropriate GENENG subroutines, Section 5.1.6. 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 5.1-11. The method of entering such a map into the GENENG program is described in detail, Reference 4.

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 5.1-12. The method of entering a combustor map into GENENG is described in Reference 4.

5.2.4.3 Turbine Maps

Turbine maps are entered into GENENG in a similar manner to fan and compressor maps, Section 5.1.4.1. The parameters of a typical turbine map are illustrated in Figure 5.2-13. Detailed instructions for describing specific turbine maps in the GENENG program are given in Reference 4.

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 5.2-14(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 5.2-14(b) and design afterburner inlet total pressure ratio, Figure 5.2-14(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.

5.1.4.5 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 5.2-15. Detailed input routines for nozzle performance maps are presented in Reference 1.

5.1.5 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

5.1.6 GENENG II Subroutine Functions and Descriptions

A flow chart of the computer program with the subroutines is shown in Figure 5.1-10. The functions of the subroutines are listed here and the purpose of each is described.

| | |
|---------|--|
| 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). |
| OVERLAY | DUMMY routine to restore working part of program to core when using overlay |
| IPTDAT | Performance data for intermediate turbine map (BLOCK DATA) |
| 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).

CMBLAT 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. 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 5.1-1. 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 | $\frac{\text{T F H C A L} - \text{T F F H P}}{\text{T F H C A L}}$ | (a) | $\frac{\text{T F I C A L} - \text{T F F I P}}{\text{T F I C A L}}$ | (a) | (a) | (a) | (b) | (a) | (a) |
| Error 2 | $\frac{\text{D H T C C} - \text{D H T C H P}}{\text{D H T C C}}$ | (a) | $\frac{\text{D H T I C} - \text{D H T C I P}}{\text{D H T I C}}$ | (a) | (a) | (a) | (b) | (a) | (a) |
| Error 3 | $\frac{\text{T F L C A L} - \text{T F F L P}}{\text{T F L C A L}}$ | (a) | (a) | (a) | (a) | (a) | (a) | (a) | (a) |
| Error 4 | $\frac{\text{D H T C F} - \text{D H T C L P}}{\text{D H T C F}}$ | (a) | (a) | (a) | (a) | (a) | (a) | (a) | (a) |
| Error 5 | $\frac{\text{P 2 5 R} - \text{P 2 5}}{\text{P 2 5 R}}$ | (a) | (a) | (a) | (a) | (a) | (a) | (a) | (a) |
| Error 6 | $\frac{\text{P 7 R} - \text{P 7}}{\text{P 7 R}}$ | (a) | (a) | (a) | (a) | (a) | (a) | (a) | (a) |
| Error 7 | $\frac{\text{P 3 8 R} - \text{P 3 8}}{\text{P 3 8 R}}$ | (a) | (a) | $\frac{\text{W A C} - \text{W A I}}{\text{W A C}}$ | ----- | (a) | (a) | ----- | (c) |
| Error 8 | $\frac{\text{T F I C A L} - \text{T F F I P}}{\text{T F I C A L}}$ | ----- | ----- | (a) | ----- | (a) | ----- | ----- | (a) |
| Error 9 | $\frac{\text{D H T I C} - \text{D H T C I P}}{\text{D H T I C}}$ | ----- | ----- | (a) | ----- | (a) | ----- | ----- | (a) |
| Matrix size | 9 × 9 | 7 × 7 | 7 × 7 | 9 × 9 | 6 × 6 | 9 × 9 | 7 × 7 | 6 × 6 | 9 × 9 |

^aSame as error for engine a.

^bSame as error for engine c.

^cSame as error for engine d.

TABLE 5.1-2. INPUTS REQUIRED FOR BASIC CYCLES

| Variable | Units or type | Definition | 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 | | | |
| | | | Turbo-fan | Boosted fan | Super-charged compressor | Turbofan | Aft fan | Super-charged compressor | Aft fan | | |
| PRFDS | ----- | Fan pressure ratio | 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 | $\frac{O_R}{sec}$ | Turbine inlet temperature | ↓ | ↓ | Yes | ↓ | ↓ | Yes | ↓ | ↓ | |
| T FHPDS | $\frac{lb\sqrt{O_R}}{(sec)(psia)}$ | High-pressure-turbine flow function | ↓ | ↓ | No | ↓ | ↓ | No | ↓ | ↓ | |
| CNHPDS | ----- | High-pressure-turbine corrected speed | ↓ | ↓ | No | ↓ | ↓ | No | ↓ | ↓ | |
| ETHPDS | ----- | High-pressure-turbine efficiency | ↓ | ↓ | No | ↓ | ↓ | No | ↓ | ↓ | |
| T FIPDS | $\frac{lb\sqrt{O_R}}{(sec)(psia)}$ | Intermediate-turbine work function | ↓ | No | Yes | ↓ | ↓ | Yes | No | ↓ | |
| CNIPDS | ----- | Intermediate-pressure-turbine corrected speed | ↓ | No | ↓ | ↓ | ↓ | ↓ | No | ↓ | |
| ETIPDS | ----- | Intermediate-pressure-turbine efficiency | ↓ | No | ↓ | ↓ | ↓ | ↓ | No | ↓ | |
| T FL PDS | $\frac{lb\sqrt{O_R}}{(sec)(psia)}$ | Low-pressure-turbine flow function | ↓ | Yes | ↓ | ↓ | Yes | ↓ | Yes | ↓ | |
| CNLPDS | ----- | Intermediate-pressure-turbine corrected speed | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | |
| ETLPDS | ----- | Intermediate-pressure-turbine efficiency | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | |
| DPDUS | ----- | 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 | |
| FIXFANTGMIDDLE | Logical | Boosted fans | .F. | .T. | .F. | .F. | .F. | .F. | .F. | .F. | |
| FIXMIDDLETOCOMP | Logical | Supercharged compressors | ↓ | .F. | .T. | .F. | .F. | .T. | .F. | .F. | |
| DUMMYSPOOL | Logical | No intermediate spool | ↓ | .F. | .F. | ↓ | .T. | .F. | .T. | .F. | |
| AFTFAN | Logical | Aft-fan engines | ↓ | .F. | .F. | ↓ | .F. | .T. | .T. | .T. | |

TABLE 5.1-3. SYMBOLS

| Thermodynamic Properties | | Component Symbols | Engine Symbols |
|--------------------------|------------------------------------|-------------------|----------------|
| AM | Mach number | A, AFT | BL |
| FAR | fuel-air ratio, t/a | B | CN |
| II | enthalpy, Btu/lbm | C | DHT |
| P | total pressure, atm | COM | DHTC |
| PS | static pressure, atm | D | |
| S | entropy, Btu ^o R/lbm | F | DP |
| T | total temperature, ^o R | I | DT |
| TS | static temperature, ^o R | M | ETA |
| V | velocity, ft/sec | MAIN | ETAR |
| | | NOZ | HPEXT |
| | | OB | N |
| | | T | PCBL |
| | | THP | PCN |
| | | TIP | PR |
| | | TLP | TFF |
| | | WDUCT | WA |
| | | WING, WNG | WF |
| | | | WG |
| | | | Z |

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), $(H_{in} - H_{out})/T_{in} \text{ Btu}/^{\circ}\text{R}/\text{lbm}$ |
| 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, $\text{lbm} \sqrt{R}/(\text{psia})(\text{sec})$ |
| WA | airflow, lbm/sec |
| WF | fuel flow, lbm/sec |
| WG | gas flow, lbm/sec |
| Z | ratio of pressure ratios |

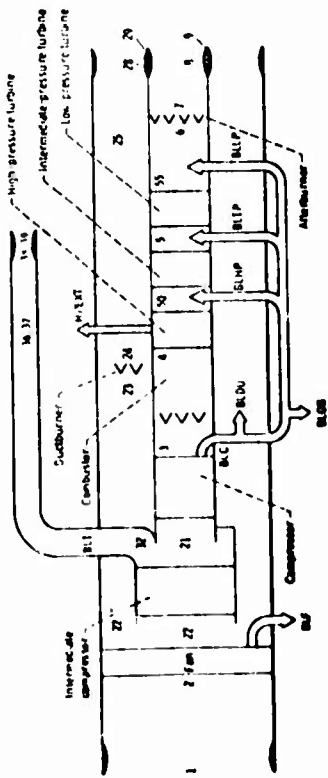


Figure 5.1-1. Three-Spool, Three-Stream Turbofan Engine (Type a)

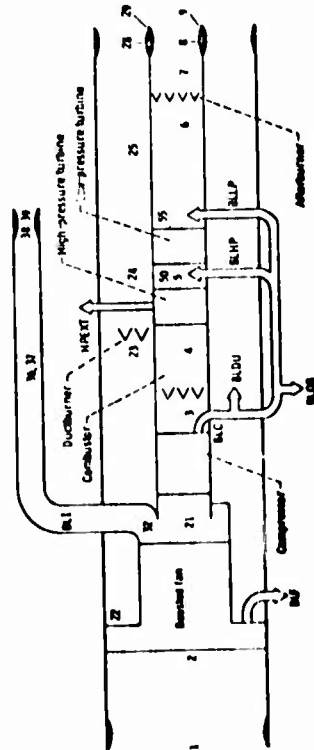


Figure 5.1-2. Two-Spool, Three-Stream Boosted Fan Engine (Type b)

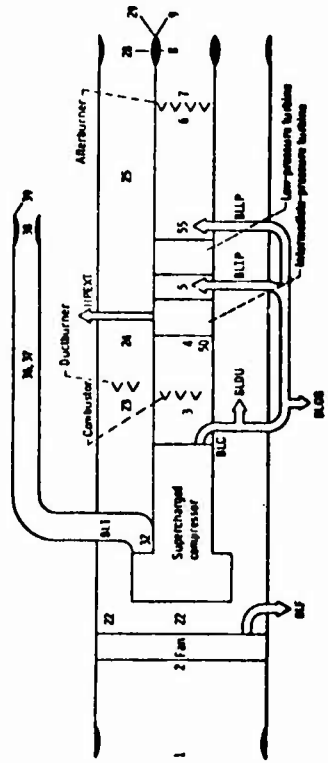


Figure 5.1-3. Two-Spool, Three-Stream, Supercharged Compressor Engine (Type c)

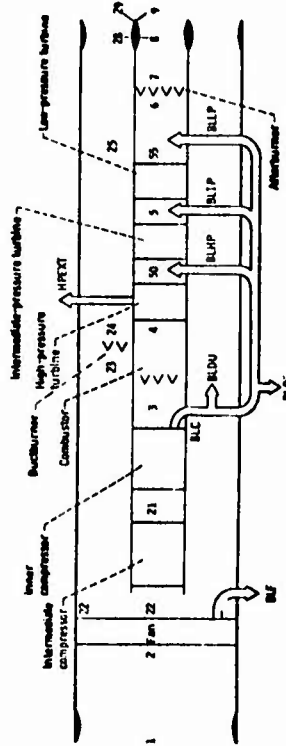


Figure 5.1-4. Three-Spool, Two-Stream Engine (Type d)

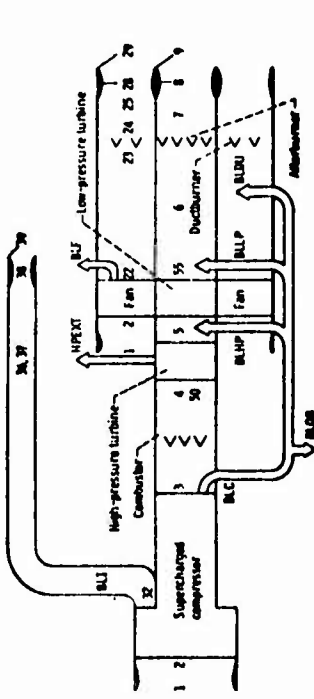


Figure 5.1-7. Two-Spool, Three-Stream, Aft Fan Engine (Type g)

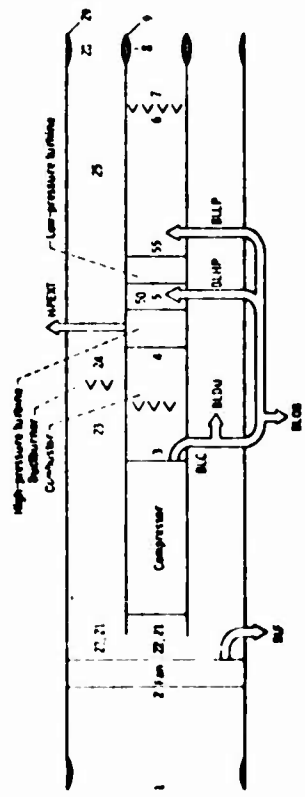


Figure 5.1-5. Two-Spool, Two-Stream Turbofan Engine (Type e)

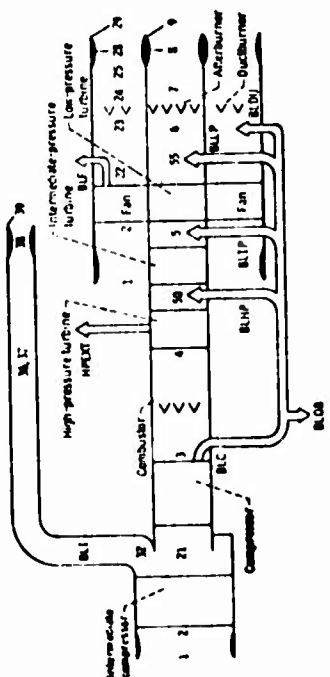


Figure 5.1-6. Three-Spool, Three-Stream, Aft Fan Engine (Type f)

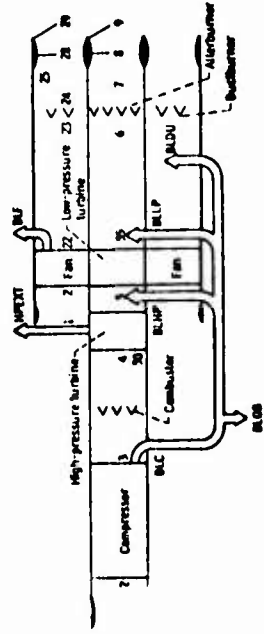


Figure 5.1-8. Two-Spool, Two-Stream, Aft Fan Engine (Type h)

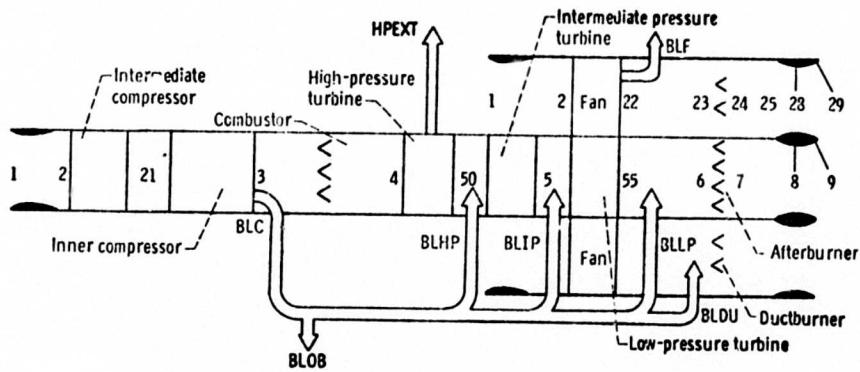


Figure 5.1-9. Three-Spool, Two-Stream, Aft Fan Engine (Type i)

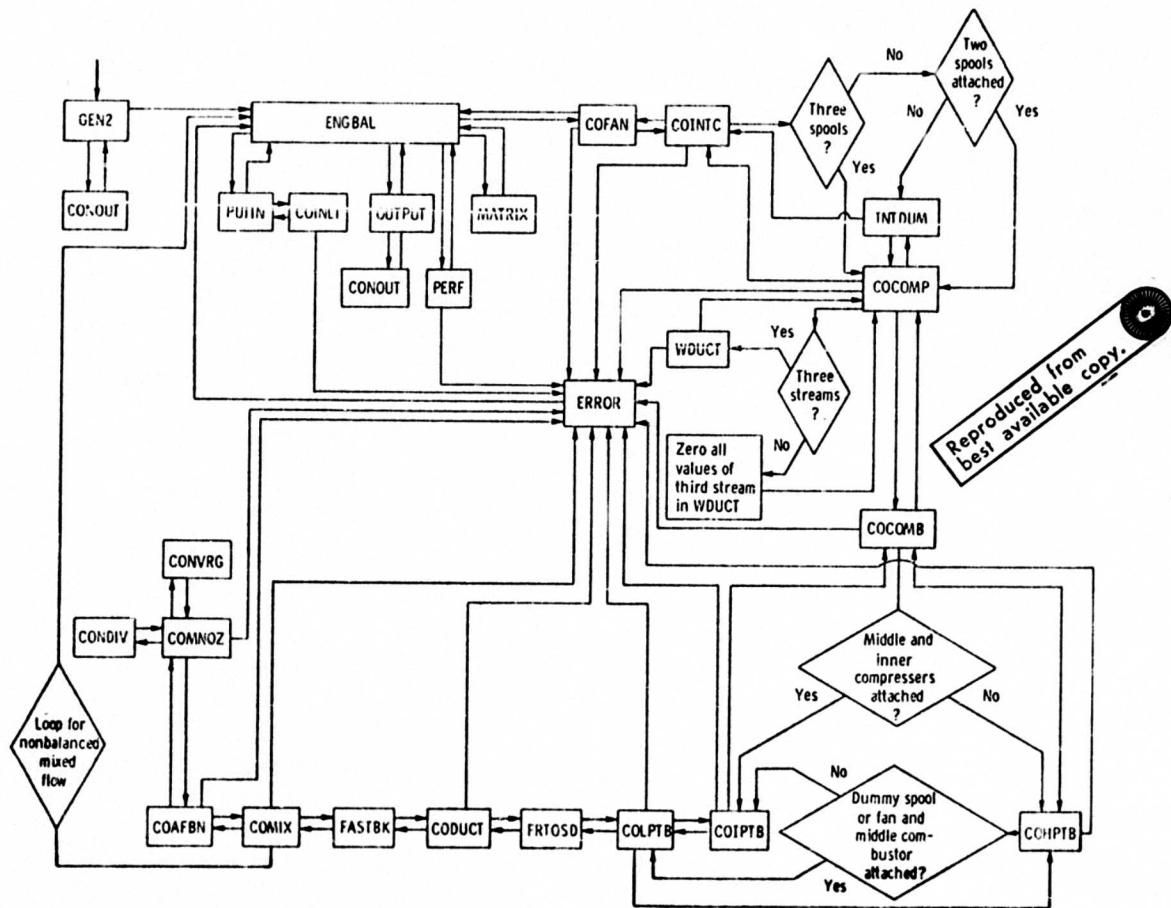


Figure 5.1-10. Flow Chart for GENENG II

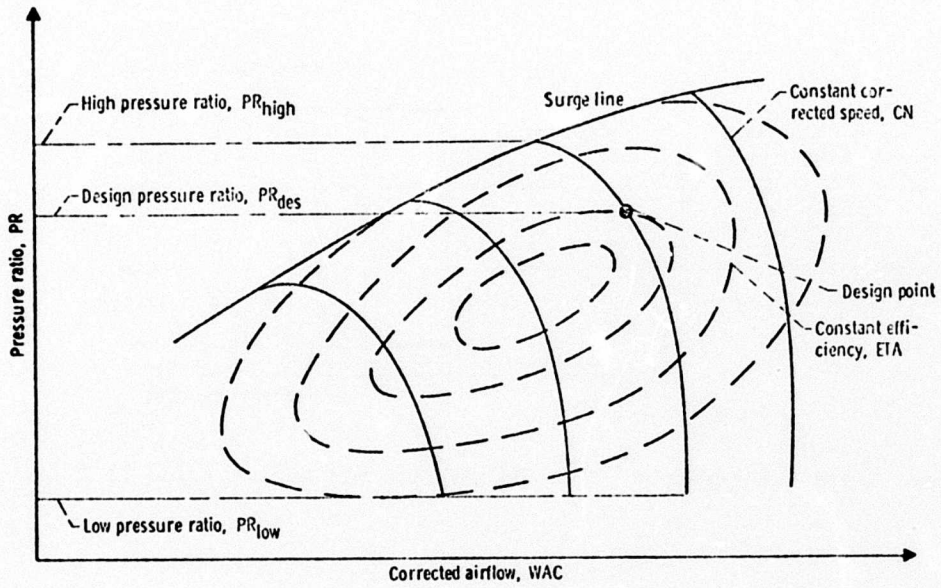


Figure 5.1-11. Example of a Specific Fan-Compressor Map

$$Z = (PR_x - PR_{low}) / (PR_{high} - PR_{low})$$

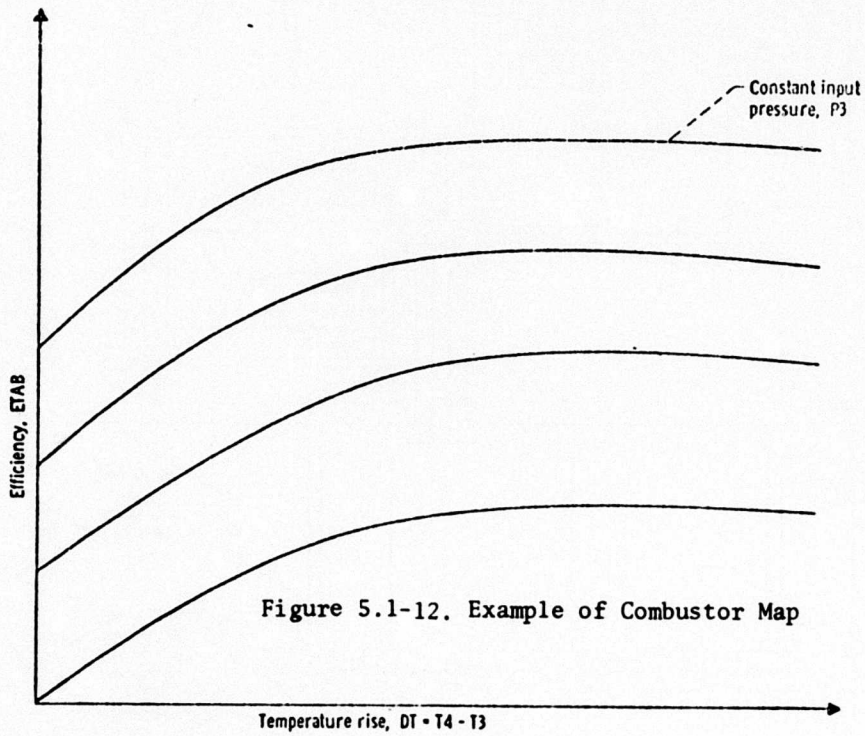


Figure 5.1-12. Example of Combustor Map

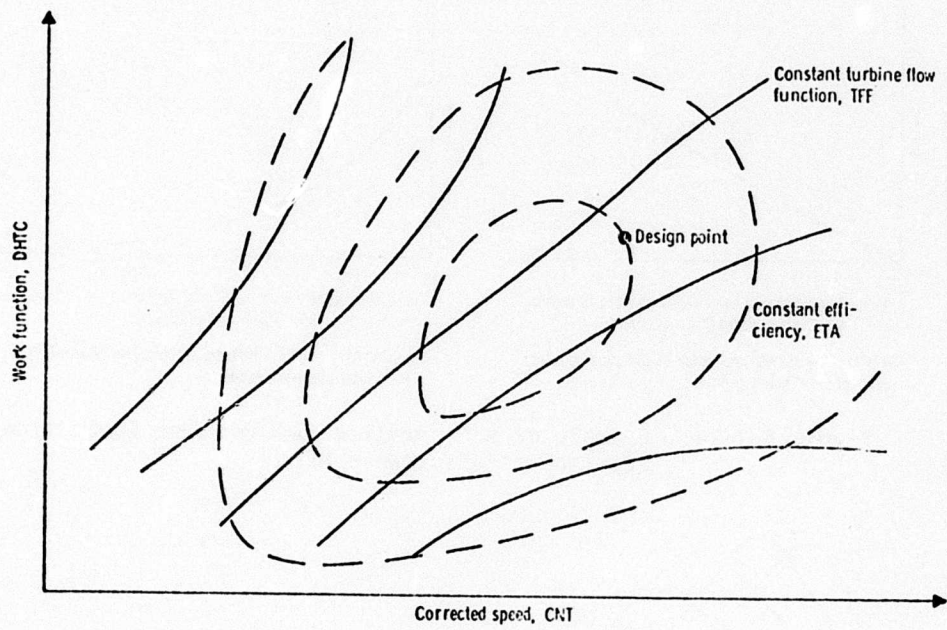
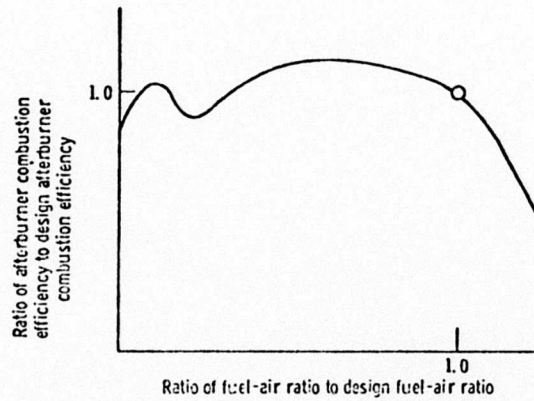
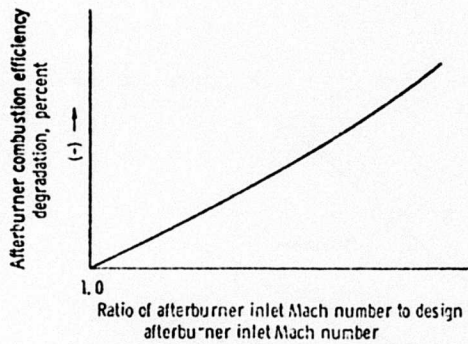


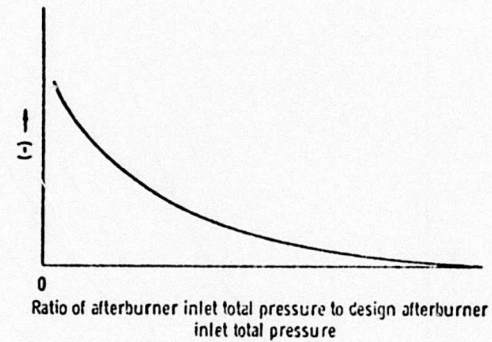
Figure 5.1-13. Example of Specific Turbine Map



(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 5.1-14. Example of a Generalized Afterburner Combustion Efficiency Performance Map

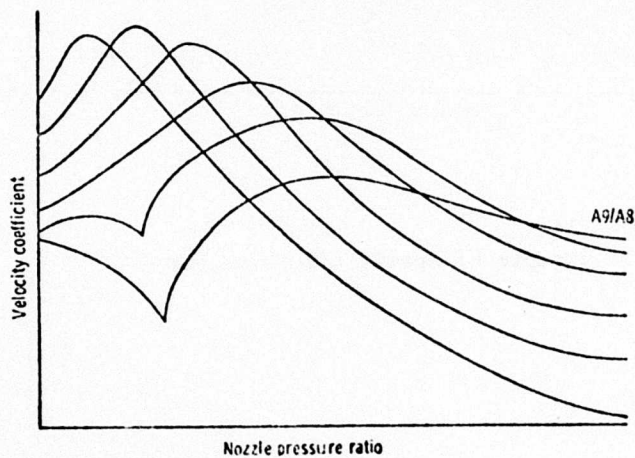


Figure 5.1-15. Performance Map for Nozzle, Giving Velocity Coefficient

SECTION 6

VOLUME AND MASS PROPERTIES

The ODIN/MFV 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 military 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/MFV usage to date has been limited to the volume and mass properties routines.

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 | 8. avionics |
| 2. body structure | 9. crew systems |
| 3. induced environment protection | 10. design reserve |
| 4. launch and recovery | 11. personnel |
| 5. main propulsion | 12. payload |
| 6. orientation controls and separation | 13. propellants |
| 7. power supply, conversion and distribution | |

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

$$WSURF = WWING + WVERT + WHORZ + WFAIR \quad (6.1.1)$$

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:

$$WWING = AC(1) * (WTO * XLF * STSPAN * SW / TROOT) ** AC(78) / 1000 \\ + AC(2) * SWING + AC(3) \quad (6.1.2)$$

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/ft²
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

$$WVERT = AC(4) * SVERT ** AC(89) + AC(5) \quad (6.1.3)$$

where

WVERT = total vertical fin weight, lbs
SVERT = vertical fin planform area, ft²
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

$$WHORZ = AC(6) * ((WTO/SWING) ** .6 * SHORZ ** 1.2 * QMAX **.8) ** AC(90) + AC(7) \quad (6.1.4)$$

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.

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

$$WFAIR = AC(8) * SFAIR + AC(9) \quad (6.1.5)$$

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.

$$AC(8) = WF \cdot KQ \cdot KT \quad (6.1.6)$$

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 of 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

$$WBODY = WBASIC + WSECST + WTHRST \quad (6.1.7)$$

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

$$WBASIC = AC(14) * SBODY + AC(15) ((ELBODY * XLF / HBODY) **.15 * QMAX **.16 * SBODY ** AC(81) + AC(16) \quad (6.1.8)$$

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.

$$AC(15)_{\text{actual}} = AC(15)_{\text{fig.6.1-7}} \times MF \quad (6.1.9)$$

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

$$WSECST = AC(17) * SBODY + AC(18) \quad (6.1.10)$$

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

$$WTHRST = AC(19) * TTOT + AC(20) \quad (6.1.11)$$

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

$$BWBODY = BWINFT + BWINOT + BWBASC + BNSSTR + BWTRST \quad (6.1.12)$$

where

BWINFT = integral fuel tank weight

BWINOT = integral oxidizer tank weight

BWBASC = basic body structure weight

BNSSTR = 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 LO₂/LH₂ vehicles. The equation for integral fuel tank weight is

$$BWINFT = BC(10) * BVFUTK + BC(11) \quad (6.1.13)$$

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 LO₂/LH₂ vehicles. The equation for integral oxidizer tank weight is

$$BWINOT = BC(12) * BVOXTK + BC(13) \quad (6.1.14)$$

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

$$BWBASC = BC(14) * BSBODY + BC(15) * BVBODY + BC(16) \quad (6.1.15)$$

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.

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

$$BWSSTR = BC(17) * BSBODY + BC(18) \quad (6.1.16)$$

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

$$\text{BWTRST} = \text{BC}(19) * \text{BTTOT} + \text{BC}(20) \quad (6.1.17)$$

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

$$\text{WTPS} = \text{WINSUL} + \text{WCOVER} \quad (6.1.18)$$

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, Section

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

$$WINSUL = AC(21) * STPS + AC(76) \quad (6.1.19)$$

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 as discussed in Section

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

$$WCOVER = AC(22) * STPS + AC(77) \quad (6.1.20)$$

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

AC(77) = 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

$$BWTPS = BWINSL + BWCOVER \quad (6.1.21)$$

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

$$BWINSL = BC(21) * BSBODY + BC(76) \quad (6.1.22)$$

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

$$BWCVR = BC(22) * BSBODY + BC(77) \quad (6.1.23)$$

where

BWCVR = 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

$$WGEAR = WLNCH + WLG \quad (6.1.24)$$

where

WLNCH = 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

$$WLNCH = AC(23) * WTO + AC(24) \quad (6.1.25)$$

where

WLNCH = 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

$$WLG = AC(25) * WTO ** AC(101) + AC(26) * WLAND + AC(27) \quad (6.1.26)$$

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

$$WPROPU = WABENG + WRENGS + WFUNCT + WOXCNT + WINSFT + WINSOT + WFUSYS + WOXSYS + WPRSYS + WINLET \quad (6.1.27)$$

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 air-breathing 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.

$$\begin{aligned}
 \text{WABENG} = & (\text{AC}(32) * e ** (\text{AC}(33) * \text{WA}) * ((\text{PT2}-\text{PHIGH})/(\text{PLOW}-\text{PHIGH})) \\
 & + \text{AC}(34) * e ** (\text{AC}(35) * \text{WA}) * ((\text{PT2}-\text{PLOW})/(\text{PHIGH}-\text{PLOW})) \\
 & * \text{ENGINS} + \text{AC}(91) * \text{ENGINS} + \text{WENGMT} \qquad (6.1.28)
 \end{aligned}$$

where

WABENG = total weight of airbreathing engines, lbs.; see Section 6.1.7.2
 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

$$\text{WABENG} = \text{AC}(82) * \text{TTOT} + \text{AC}(83) + \text{WENGMT} \qquad (6.1.29)$$

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 LO₂/LH₂ engine. The weight is scaled as a function of total stage vacuum thrust and area ratio. The equation for rocket engine weight is

$$\begin{aligned} \text{WRENGS} = & \text{AC}(28) * \text{TTOT} + \text{AC}(29) * \text{TTOT} * \text{ARATIO} ** \text{AC}(30) + \text{AC}(31) \\ & * \text{ENGINs} + \text{WENGMt} \end{aligned} \quad (6.1.30)$$

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 ($f_{\text{Thrust and area ratio}}$)
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

$$\text{WENGMt} = \text{AC}(102) * \text{TTOT} + \text{AC}(103) \quad (6.1.31)$$

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 air-breathing 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

$$WFUNCT = AC(36) * (GAL/Tanks) ** .6 * TANKS + AC(37) \quad (6.1.32)$$

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

$$WFUNCT = AC(36) * VFUTK + AC(37) \quad (6.1.33)$$

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

$$WOXCNT = AC(38) * VOXTK + AC(39) \quad (6.1.34)$$

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

$$WINSFT = AC(40) * SFUTK + AC(41)$$

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

$$WINSOT = AC(42) * SOXTK + AC(43)$$

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

$$WBPUMP = \frac{TTOT}{1000} * (1.75 + 0.266 * ENGINES) \quad (6.1.38)$$

where

WBPUMP = total weight of boost and transfer pumps, lbs.
 TTOT = total stage vacuum thrust, lbs.
 ENGINES = 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

$$WDIST1 = ENGINES * AC(104) * (TTOT/ENGINES) ** .5 \quad (6.1.39)$$

where

WDITS1 = total weight of fuel distribution system Part I, lbs.
 ENGINES = 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

$$\text{WDIST2} = 0.255 * \text{GAL} ** .7 * \text{TANKS} ** .25 \quad (5.1.40)$$

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

$$\text{WFCNT} = 0.169 * \text{TANKS} * \text{GAL} ** .5 \quad (6.1.41)$$

where

WFCNT = 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

$$\text{WREFUL} = \text{TANKS} * (3.0 + 0.45 * \text{GAL} ** .333) \quad (6.1.42)$$

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

$$\text{WDRANS} = 0.159 * \text{GAL} ** .65 \quad (6.1.43)$$

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

$$WSEAL = 0.045 * TANKS * GAL / TANKS^{.75} \quad (6.1.44)$$

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

$$WFUSYS = AC(44) * TTOT + AC(45) * ELBODY + AC(46) \quad (6.1.45)$$

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

$$WOXSYS = AC(47) * TTOT + AC(48) * ELBODY + AC(49) \quad (6.1.46)$$

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

$$WPRSYS = 0.0009 * TTOT * TANKS \quad (6.1.47)$$

- 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

$$WPRSYS = AC(50) * VFUTK + AC(51) * VOXTK + AC(52) \quad (6.1.48)$$

- 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

$$W_{INLET} = W_{DUCT} + W_{RAMP} + W_{SPIKE} \quad (6.1.49)$$

where

W_{DUCT} = internal duct weight
 W_{RAMP} = ramp and ramp control weight
 W_{SPIKE} = 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

$$W_{DUCT} = AC(53) * ((ELNLET * XINLET) ** .5 * (AICAPT / XINLET) ** .3334 \\ * PT2 ** .6667 * GEOFACT * FCTMOK) ** AC(54) + AC(105) \quad (6.1.50)$$

where

W_{DUCT} = 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
 $GEOFACT$ = 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 < 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.

$$\text{TMPFCT} = \begin{cases} 1.0, & \text{Mach number} < 3.0 \\ 0.203 * \text{DM} + 0.4, & \text{Mach number} \geq 3.0 \end{cases} \quad (6.1.51)$$

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

$$\text{WVRAMP} = \text{AC}(106) * (\text{ELRAMP} * \text{XINLET} * (\text{AICAPT}/\text{XINLET}) ** .5 * \text{TMPFCT}) ** \text{AC}(107) + \text{AC}(108) \quad (6.1.52)$$

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

$$\text{WSPIKE} = \text{AC}(109) * \text{XINLET} \quad (6.1.53)$$

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

$$\begin{aligned} \text{BWPRPL} = & \text{BWRENG} + \text{BWFCNT} + \text{BWCNT} + \text{BWINSF} + \text{BWINSO} + \text{BWFUSY} \\ & + \text{BWDXS Y} + \text{BWPRSY} \end{aligned} \quad (6.1.54)$$

where

BWRENG = main engine weight including mounts
BWFCNT = non-structural fuel container weight
BWCNT = non-structural oxidizer container weight
BWINSF = fuel tank insulation weight
BWINSO = oxidizer tank insulation weight
BWFUSY = cryogenic fuel system weight
BWDXS Y = 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 LO₂/LH₂ engine. The weight is scaled as a function of total stage vacuum thrust and area ratio. The equation for rocket engine weight is

$$\begin{aligned} \text{BWRENG} = & \text{BC}(28) * \text{BTTOT} + \text{BC}(29) * \text{BTTOT} * \text{BARATO} ** \text{BC}(30) + \text{BC}(31) \\ & * \text{ENGINS} + \text{WBENMT} \end{aligned}$$

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

$$WBENMT = BC(102) * BTTOT + BC(103) \quad (6.1.56)$$

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

$$BWFCNT = BC(36) * BVFUTK + BC(37) \quad (6.1.57)$$

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

$$BWOCNT = BC(38) * BVOXTK + BC(39) \quad (6.1.58)$$

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

$$BWINSF = BC(40) * BSFUTK + BC(41) \quad (6.1.59)$$

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

$$\text{WORNT} = \text{WGIMBL} + \text{WACS} + \text{WACSTK} + \text{WAERO} + \text{WSEP} \quad (6.1.64)$$

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

$$\text{TDEL} = 750 * (\text{TTOT}/\text{ENGINS}/\text{PCHAM}) ** 1.25 \quad (6.1.65)$$

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

$$WGIMBL = AC(55) * TDEL ** AC(110) + AC(56) \quad (6.1.66)$$

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

$$WACS = AC(57) * WTO ** AC(58) + AC(59) \quad (6.1.67)$$

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

$$WACSTK = AC(64) * (WACSFU + WACSOX) + AC(65) \quad (6.1.68)$$

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

$$WAERO = AC(60) * (WTO ** .666 * (ELBODY + GSPAN) ** .25) ** AC(111) + AC(61) \quad (6.1.69)$$

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

$$WSEP = AC(62) * WTO + AC(63) \quad (6.1.70)$$

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

$$BWORNT = BWGIMB + BWSEP \quad (6.1.71)$$

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

$$BWGIMB = BC(55) * BTDEL ** BC(110) + BC(56) \quad (6.1.72)$$

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

$$BWSEP = BC(62) * WTO + BC(63) \quad (6.1.73)$$

where

BWSEP weight of separation system, lbs.
BWTC = 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

$$WPWSY = WELECT + WHYPNU \quad (6.1.74)$$

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 blowers 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

$$WELECT = AC(66) * (WTO **.5 * ELBODY **.25) ** AC(112) + AC(67) \quad (6.1.75)$$

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

$$\text{WHYPNU} = \text{AC}(68) \left[\left((\text{SWING} + \text{SHORZ} + \text{SVERT}) * \text{QMAX}/1000 \right) ** 0.334 + (\text{\#LBODY} + \text{STSPAN}) ** 0.5 * \text{TYTAIL} \right] ** \text{AC}(113) + \text{AC}(69) \quad (6.1.76)$$

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

$$\text{BWPWSY} = \text{BWELEC} + \text{BWHYPN} \quad (6.1.77)$$

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

$$\text{BWELEC} = \text{BC}(66) * \text{BLBODY} + \text{BC}(67) \quad (6.1.78)$$

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

$$BWHYPN = BC(68) * BLBODY + BC(69) \quad (6.1.79)$$

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

$$\text{WAVONC} = \text{AC}(70) * \text{WTO} ** \text{AC}(114) + \text{AC}(71) \quad (6.1.80)$$

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

$$\text{WCPROV} = \text{AC}(74) * \text{WTO} + \text{AC}(80) * \text{CREW} + \text{AC}(75) \quad (6.1.81)$$

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:

$$\begin{aligned} \text{WDRY} = & \text{WSURF} + \text{WBODY} + \text{WTPS} + \text{WGEAR} + \text{WPROPU} + \text{WORNT} \\ & + \text{WPWSY} + \text{WAVONC} + \text{WCPROV} \end{aligned} \quad (6.1.82)$$

This value for dry weight is then used in the equation for contingency and growth which is

$$\text{WCONT} = \text{AC}(98) * \text{WDRY} + \text{AC}(99) \quad (6.1.83)$$

where

WCONT = weight of contingency and growth, lbs.
 WDRY = stage dry weight, lbs.
 $\text{AC}(98)$ = contingency and growth coefficient
 $\text{AC}(99)$ = fixed contingency and growth weight, lbs.

The aircraft weight empty is summed by the equation

$$\text{WEMPTY} = \text{WDRY} + \text{WCONT} \quad (6.1.84)$$

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)

$$\text{BWDRY} = \text{BWBODY} + \text{BWTPS} + \text{BWPRPL} + \text{BWORNT} + \text{BWPWSY} \quad (6.1.85)$$

This value for dry weight is then used in the equation for contingency and growth which is

$$\text{BWCONT} = \text{BC}(98) * \text{BWDRY} + \text{BC}(99) \quad (6.1.86)$$

where

BWCONT = weight of contingency and growth, lbs.
 BWDRY = stage dry weight, lbs.
 $\text{BC}(98)$ = contingency and growth coefficient
 $\text{BC}(99)$ = fixed contingency and growth weight, lbs.

The booster weight empty is summed by the equation

$$\text{BWEMPTY} = \text{BWDRY} + \text{BWCONT} \quad (6.1.87)$$

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

$$W_{CREW} = AC(72) * CREW + AC(73) \quad (6.1.88)$$

where

W_{CREW} = 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

$$WFTRAP = AC(92) * WFUEL + AC(93) \quad (6.1.89)$$

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

$$\text{WOTRAP} = \text{AC}(94) * \text{WOXID} + \text{AC}(95) \quad (6.1.90)$$

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 typical 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

$$\text{WFRESV} = \text{AC}(84) * \text{WFUELM} + \text{AC}(85) \quad (6.1.91)$$

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

$$\text{WORESV} = \text{AC}(86) * \text{WOXIDM} + \text{AC}(87) \quad (6.1.92)$$

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

$$\text{WACSFU} = \text{AC}(96) * \text{WTO} + \text{AC}(97) \quad (6.1.93)$$

and

$$\text{WACSOX} = \text{WACSFU} * \text{OFACS} \quad (6.1.94)$$

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 oxidizer, 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

$$\text{WFUELM} = \text{WPMAIN} / (1. + \text{OF}) \quad (6.1.95)$$

and

$$\text{WOXIDM} = \text{WFUELM} * \text{OF} \quad (6.1.96)$$

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:

$$\text{WFUEL} = \text{WFUELM} + \text{WFRESV} \quad (6.1.97)$$

$$\text{WOXID} = \text{WOXIDM} + \text{WORES V} \quad (6.1.98)$$

The total weight of fuel and oxidizer in the tanks are summed by the equations

$$\text{WFUTOT} = \text{WFUEL} + \text{WFTRAP} \quad (6.1.99)$$

$$\text{WOXTOT} = \text{WOXID} + \text{WOTRAP} \quad (6.1.100)$$

The total weight of propellant tanked is summed by the equation

$$\text{WP} = \text{WFUTOT} + \text{WOXTOT} \quad (6.1.101)$$

The operating weight empty is summed by the equation

$$\text{WOPMTY} = \text{WEMPTY} + \text{WRESID} + \text{WCREW} + \text{WACSFU} + \text{WACSOX} \quad (6.1.102)$$

The zero fuel weight is summed by the equation

$$\text{WZROFU} = \text{WOPMTY} + \text{WPAYLD} \quad (6.1.103)$$

The gross weight is summed by the equation

$$WTO = WZROFU + WPMAIN + WFRESV + WORES V \quad (6.1.104)$$

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

$$BWRES D = BWFTRP + BWOTRP \quad (6.1.108)$$

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

$$BWORES = BC(86) * BWOXM + BC(87) \quad (6.1.110)$$

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:

$$BWPMAN = BWTO * (BMASRT - 1.) / BMASRT \quad (6.1.111)$$

$$BWFULM = BWPMAN / (BMIXRT + 1.) \quad (6.1.112)$$

$$BWOXM = BWPMAN - BWFULM \quad (6.1.113)$$

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

$$BWFUEL = BWFULM + BWFRES \quad (6.1.114)$$

$$BWOXID = BWOXM + BWORES \quad (6.1.115)$$

The total weight of the fuel and oxidizer in the tank are summed by the equations:

$$\text{BWFUTL} = \text{BWFUEL} + \text{BWFTRP} \quad (6.1.116)$$

$$\text{BWOXTL} = \text{BWOXID} + \text{BWOTRP} \quad (6.1.117)$$

The booster operating weight empty is summed by the equation

$$\text{BWOPMT} = \text{BWEEMPTY} + \text{BWRES D} \quad (7.1.118)$$

The booster zero fuel weight (or burnout) is summed by the equation

$$\text{BWZROF} = \text{BWOPMT} + \text{WTO} \quad (6.1.119)$$

The booster gross weight is summed by the equation

$$\text{BWTO} = \text{BWZROF} + \text{BWPMAN} + \text{BWFRES} + \text{BWORES} \quad (6.1.120)$$

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/MFV alternate sources of geometric perturbations must be employed such as the geometry programs of Section 3.

REFERENCES:

1. Adams, J. D., Vehicle Synthesis of High Speed Aircraft, Volume I: Analysis Techniques and User's Instructions, AFFDL-TR-71-40, June 1971.
2. Oman, B., Vehicle Synthesis of High Speed Aircraft, Volume II: Weights and Geometry Analysis, AFFDL-TR-71-40, June 1971.
3. Glatt, C. R., Weight Analysis for Advanced Transportation Systems, Aerophysics Research Corporation LTN-01, 1972.

| 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-38 | 20 | 366 | 42 | 0 | 53 | 481 | 533 | 0.90 |
| 2 | F-204A | 21 | 168 | 197 | 0 | 17 | 403 | 669 | 0.60 |
| 3 | XF-58 | 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 | 430 | 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 | 88 | 486 | 235 | 0 | 0 | 806 | 1373 | 0.59 |

TABLE 6.1-1. TYPICAL FAIRING WEIGHTS

| Fairing Type | WF at $Q = 400 \text{ lbs/ft}^2$ and $T = 400^\circ\text{F}$ |
|----------------------|---|
| 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 6.1-2. SECONDARY STRUCTURE DATA

| TYPE OF SPIKE | AC(109) |
|--------------------------------|---------|
| 1/2 ROUND - FIXED | 35 |
| FULL ROUND - TRANSLATING | 70 |
| FULL TRANSLATING AND EXPANDING | 290 |

TABLE 6.1-3. TYPICAL SPIKE WEIGHTS

| | |
|------------------------|-----------------------------|
| 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 +100 ^o F |
| Acceleration | 2.5 to 15g |

TABLE 6.1-4. AIRCRAFT GIMBAL SYSTEM PARAMETERS

| SYSTEM DESCRIPTION | AC(74) | AC(75) | 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 | - | 60-120 | - |
| Fixed Life Support | - | 10 | - |
| Fixed Emergency Equipment | - | 50 | - |
| Crew Station Controls and Panels | - | 40 | 50 |

TABLE 6.1-5. TYPICAL AIRCRAFT CREW PROVISION INPUTS

| DESCRIPTION | AC(72) | AC(73) |
|----------------------------|---------|--------|
| Crew, Gear and Accessories | 220-290 | --- |
| Crew Life Support | 1-6 | 25-60 |

TABLE 6.1-6. TYPICAL INPUTS FOR CREW AND CREW LIFE SUPPORT

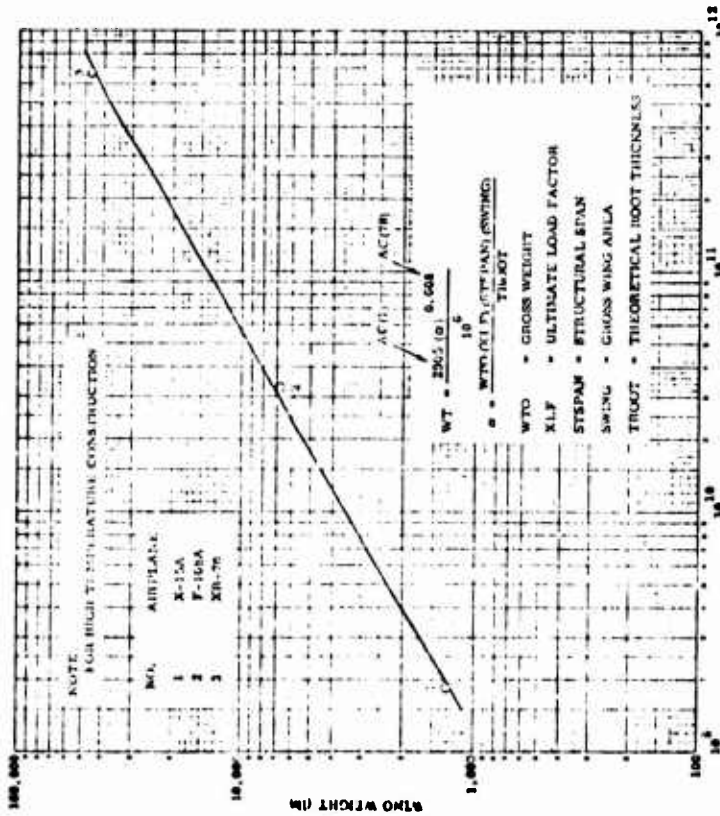


FIGURE 6.1.1. ALUMINUM WING WEIGHTS (NO VARIABLE SWEEP PENALTY INCLUDED)

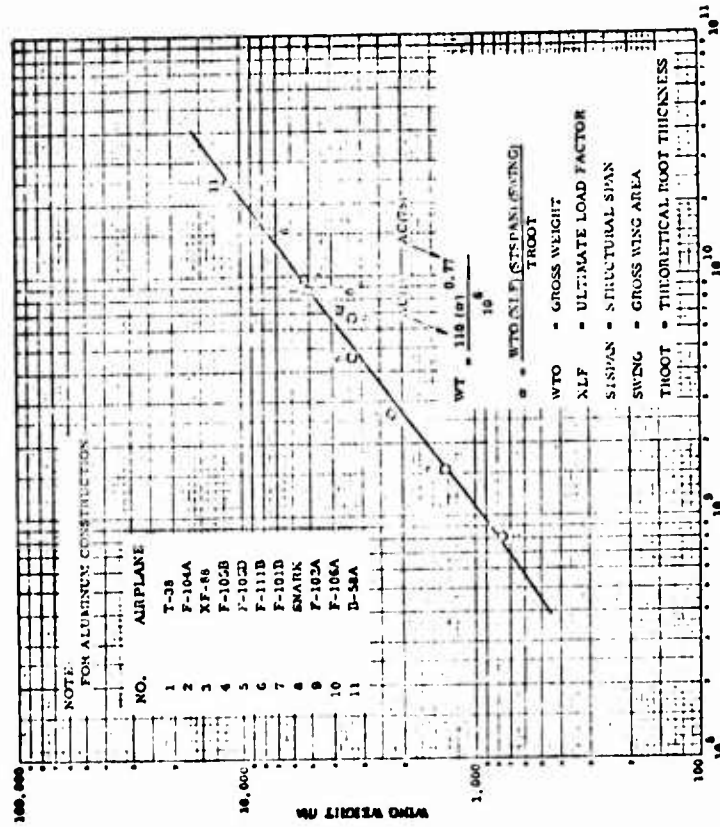


FIGURE 6.1.2. HIGH TEMPERATURE AND THIN SUPERSONIC WING WEIGHTS

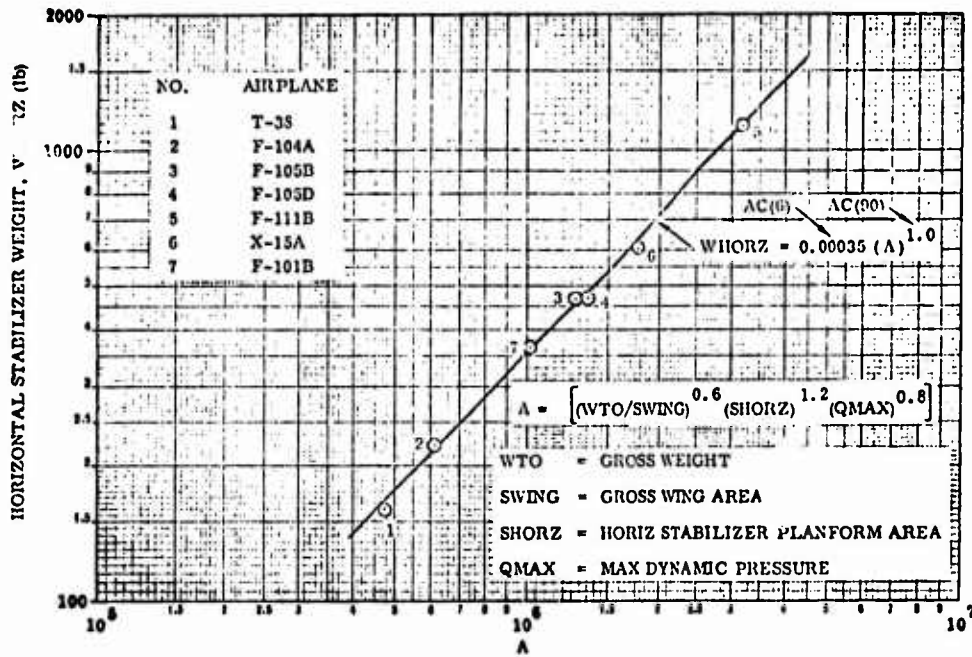


FIGURE 6.1-3. VERTICAL FIN WEIGHTS

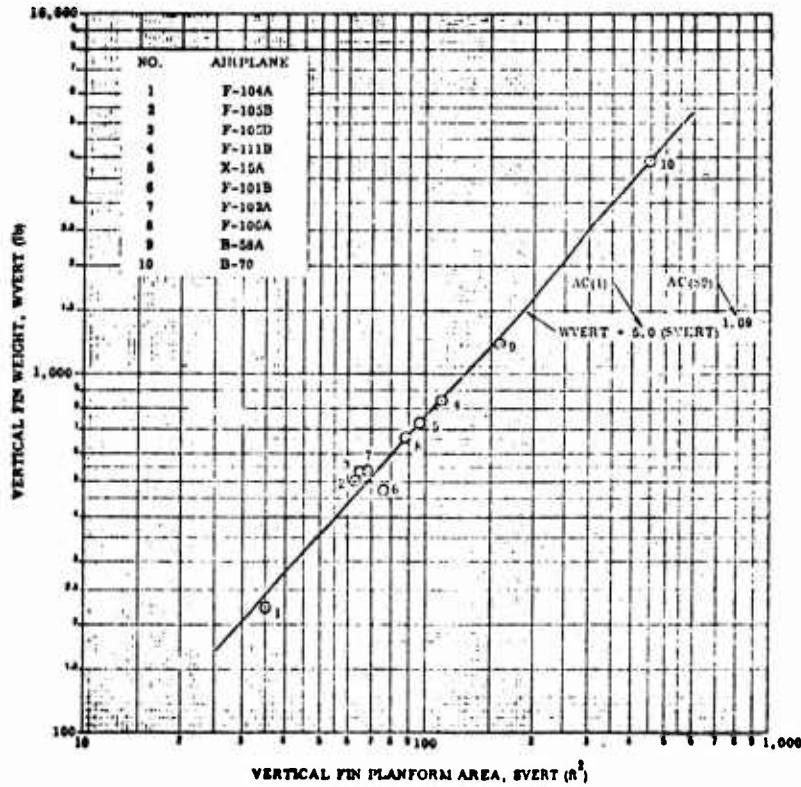


FIGURE 6.1-4. HORIZONTAL STABILIZER WEIGHTS

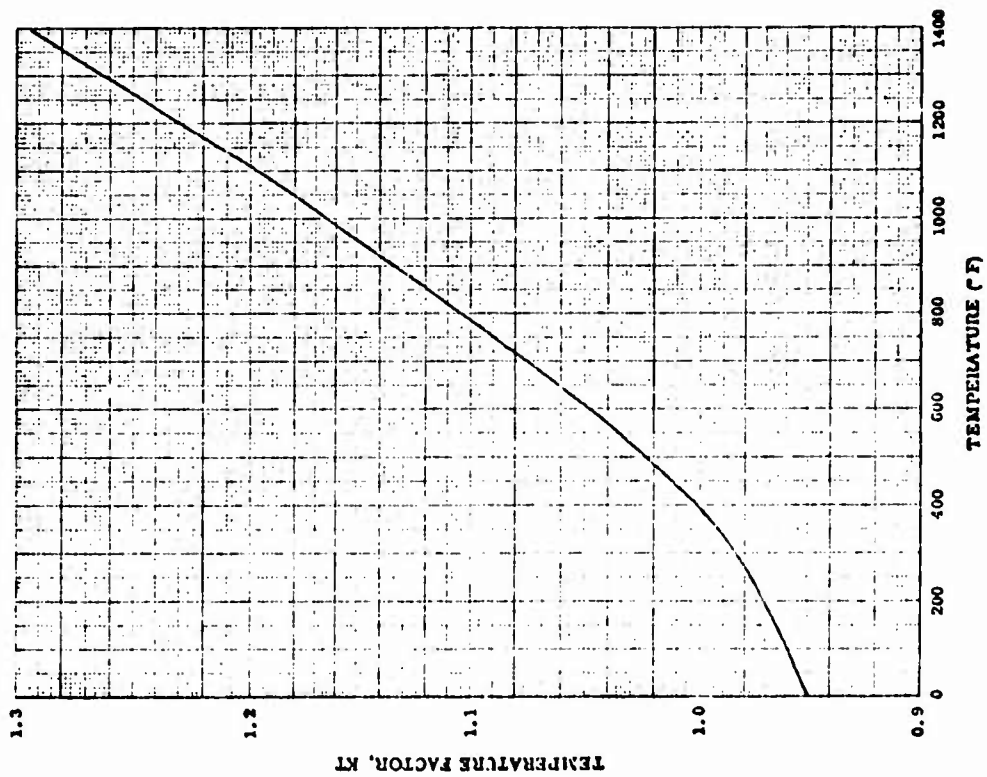


FIGURE 6.1-6. FAIRING WEIGHT TEMPERATURE FACTOR

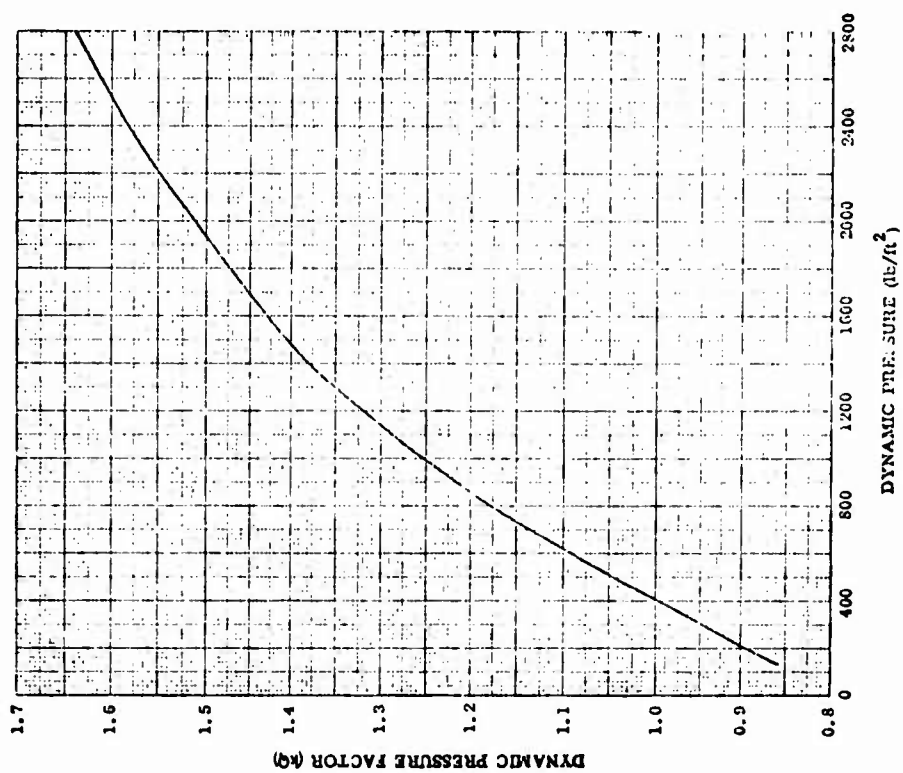


FIGURE 6.1-5. FAIRING WEIGHT DYNAMIC PRESSURE FACTOR

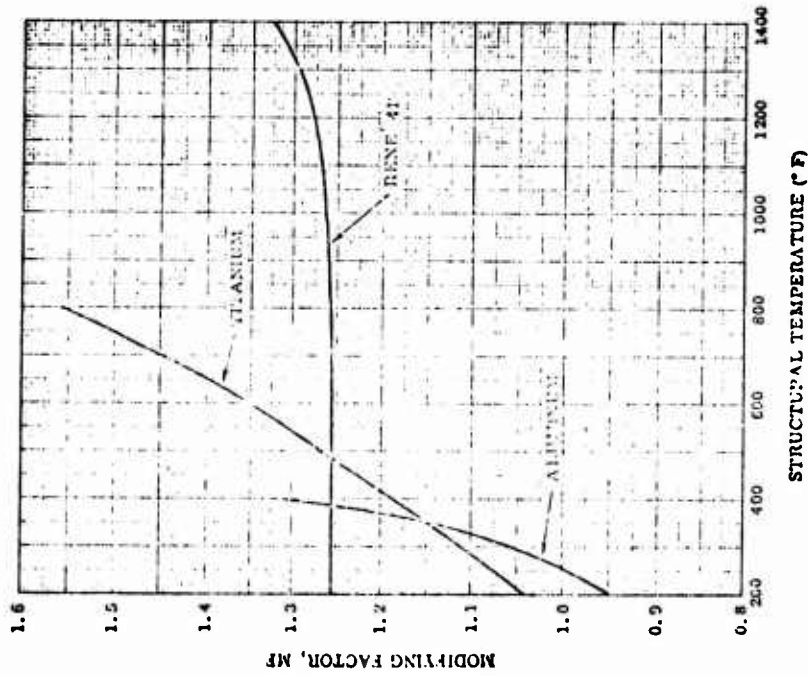


FIGURE 6.1-8. BASIC AIRCRAFT BODY WEIGHT TEMPERATURE FACTOR

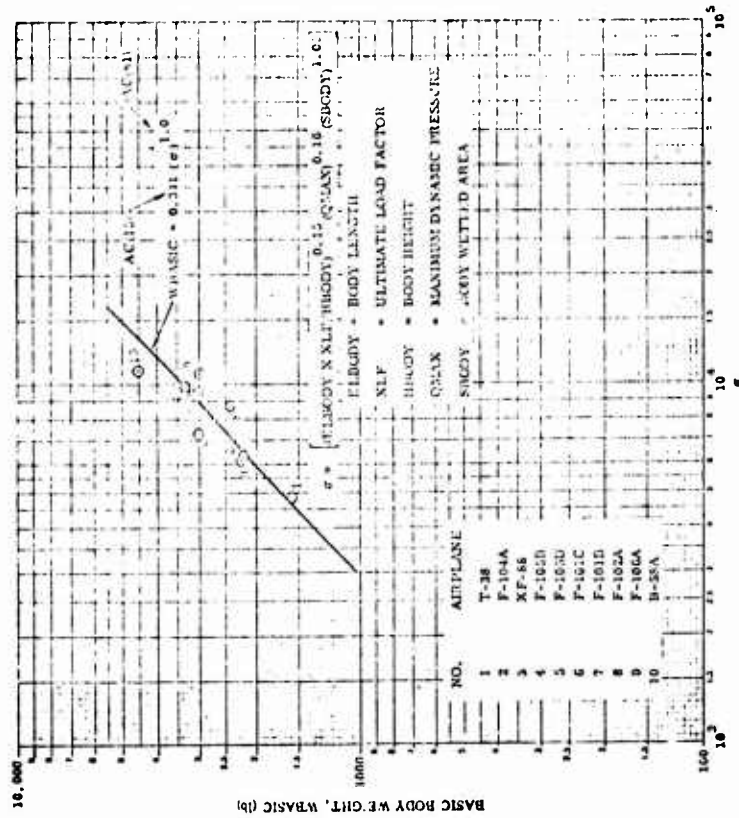


FIGURE 6.1-7. BASIC AIRCRAFT BODY WEIGHTS

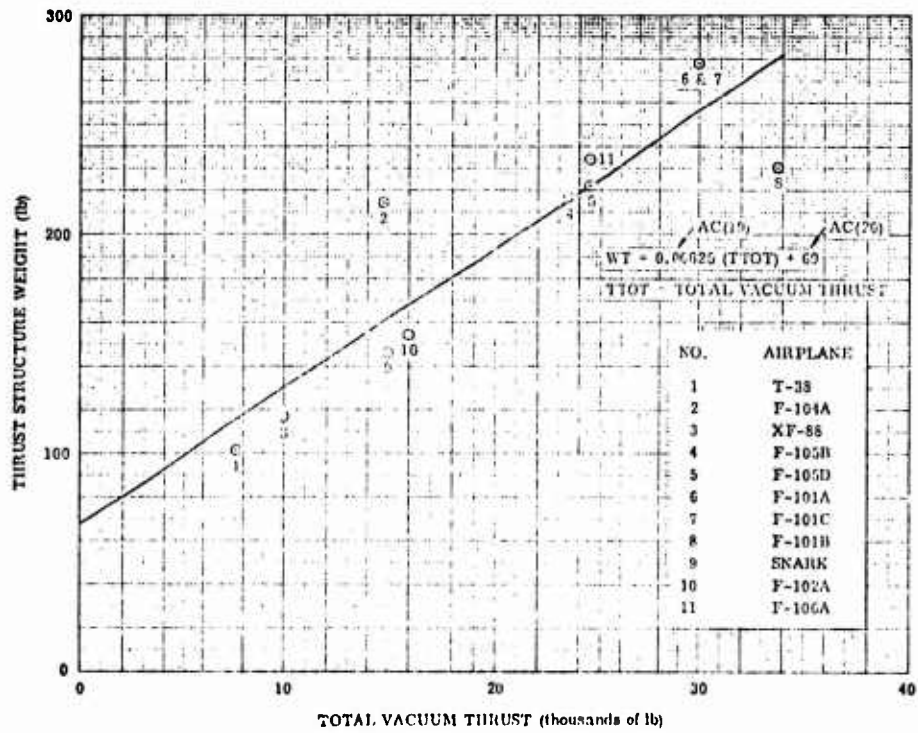
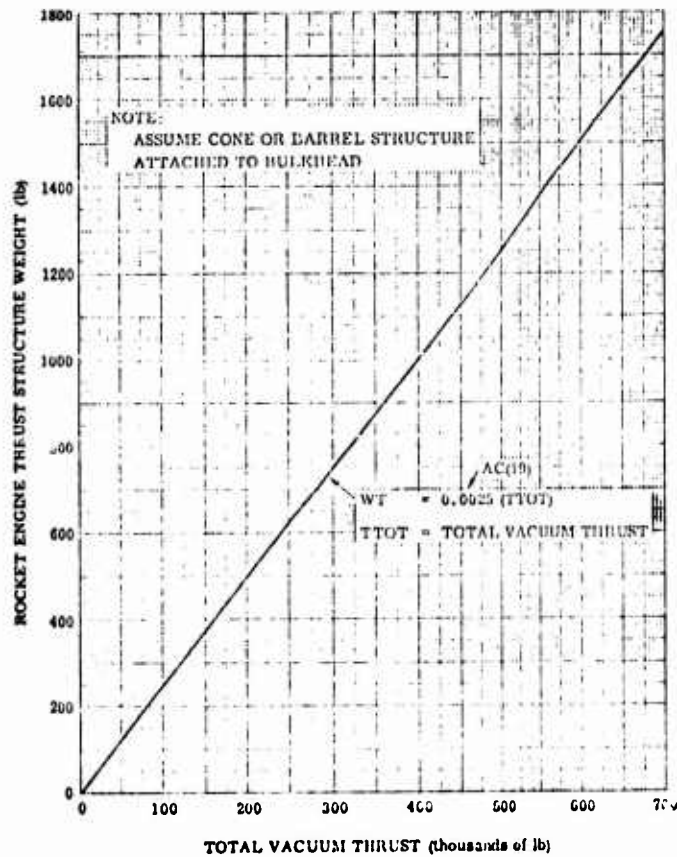


FIGURE 6.1-9. AIRBREATHING THRUST STRUCTURE WEIGHT



6.1-48

FIGURE 6.1-10. ROCKET ENGINE THRUST STRUCTURE

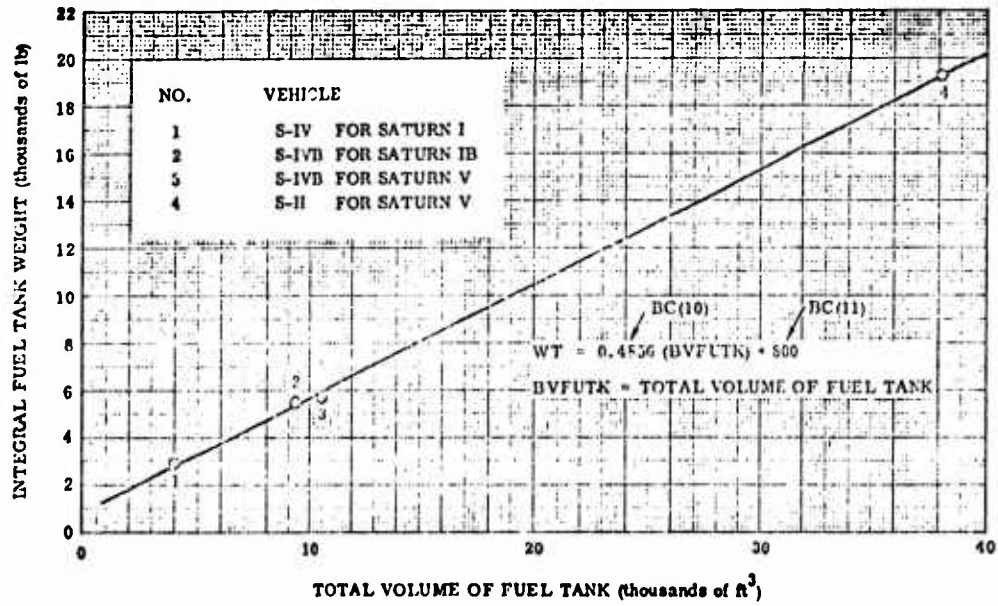


FIGURE 6.1-11. BOOSTER INTEGRAL FUEL TANK WEIGHTS

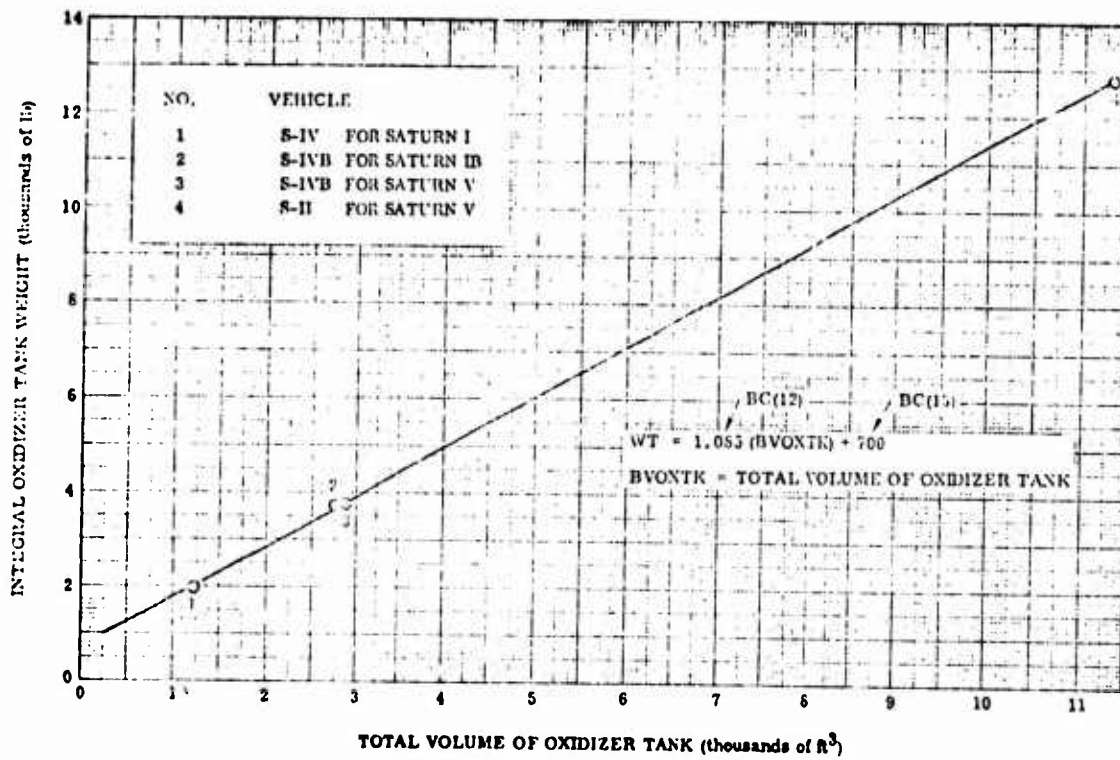


FIGURE 6.1-12. BOOSTER INTEGRAL OXIDIZER TANK WEIGHTS

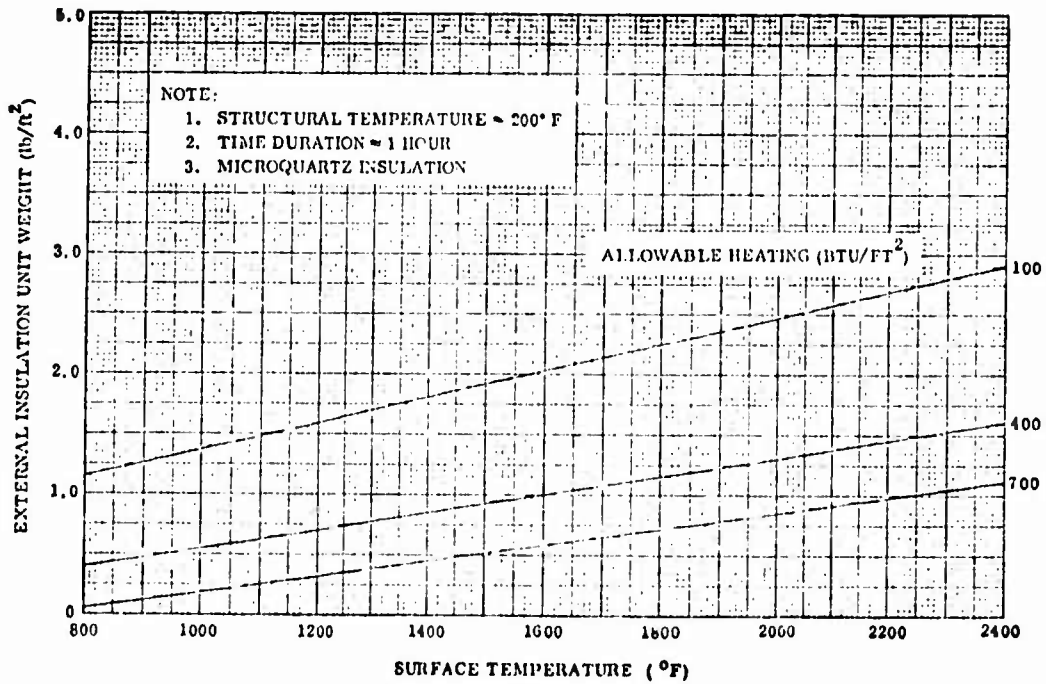


FIGURE 6.1-13. AIRCRAFT INSULATION WEIGHTS

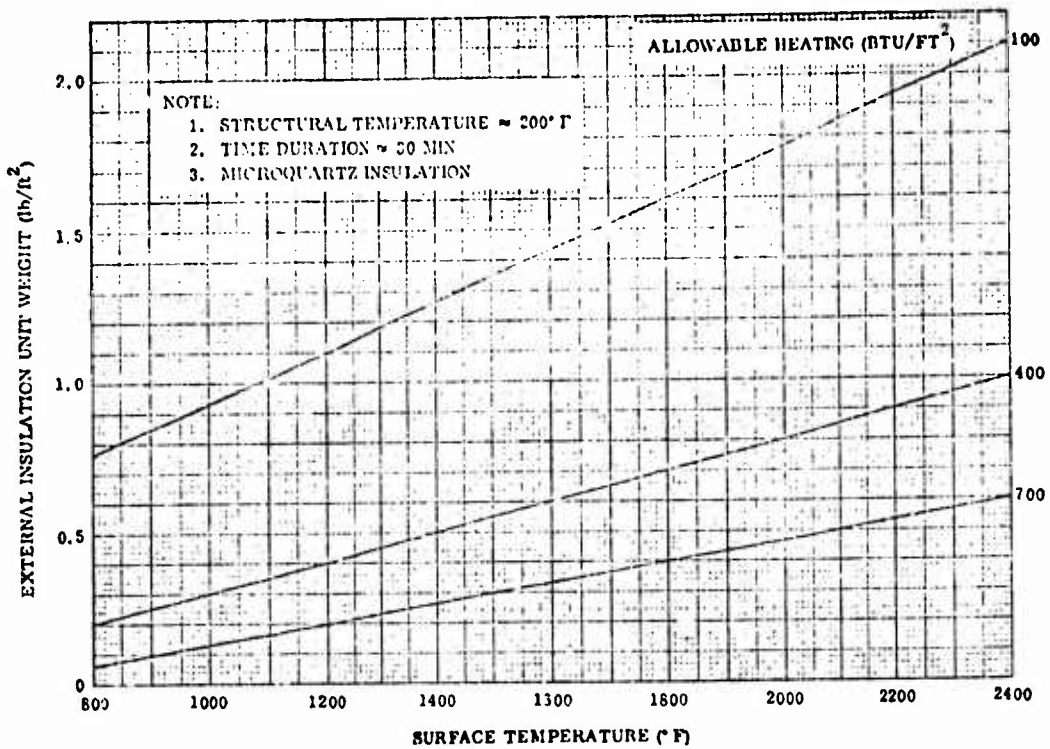


FIGURE 6.1-14. BOOSTER INSULATION WEIGHT

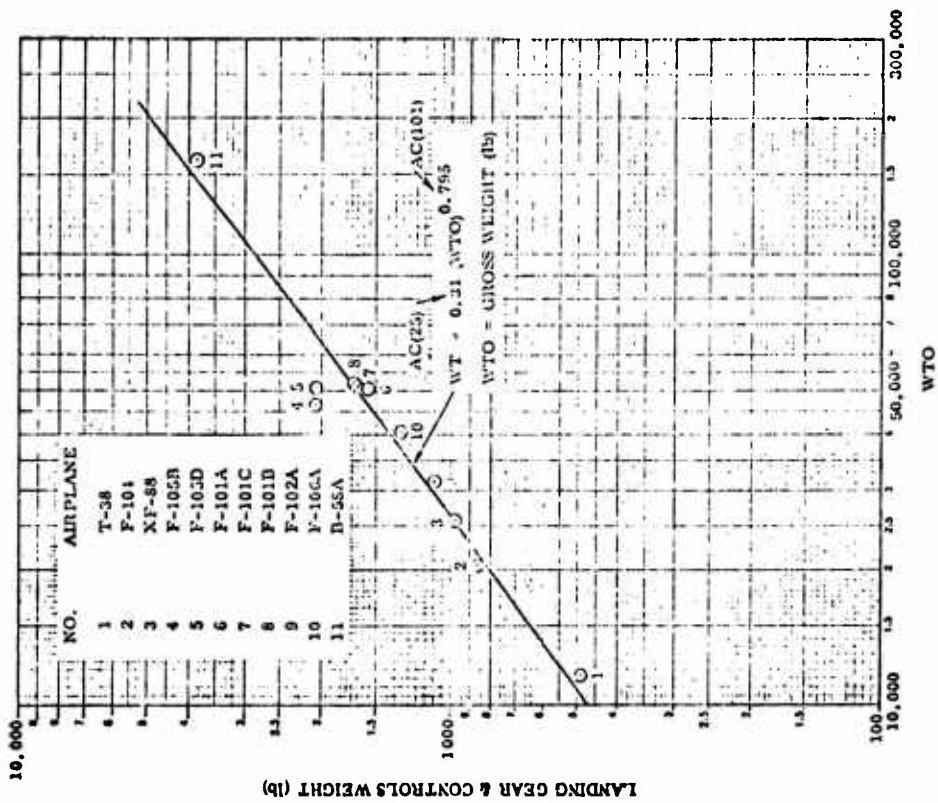


FIGURE 6.1-15. AIRCRAFT LANDING GEAR WEIGHT

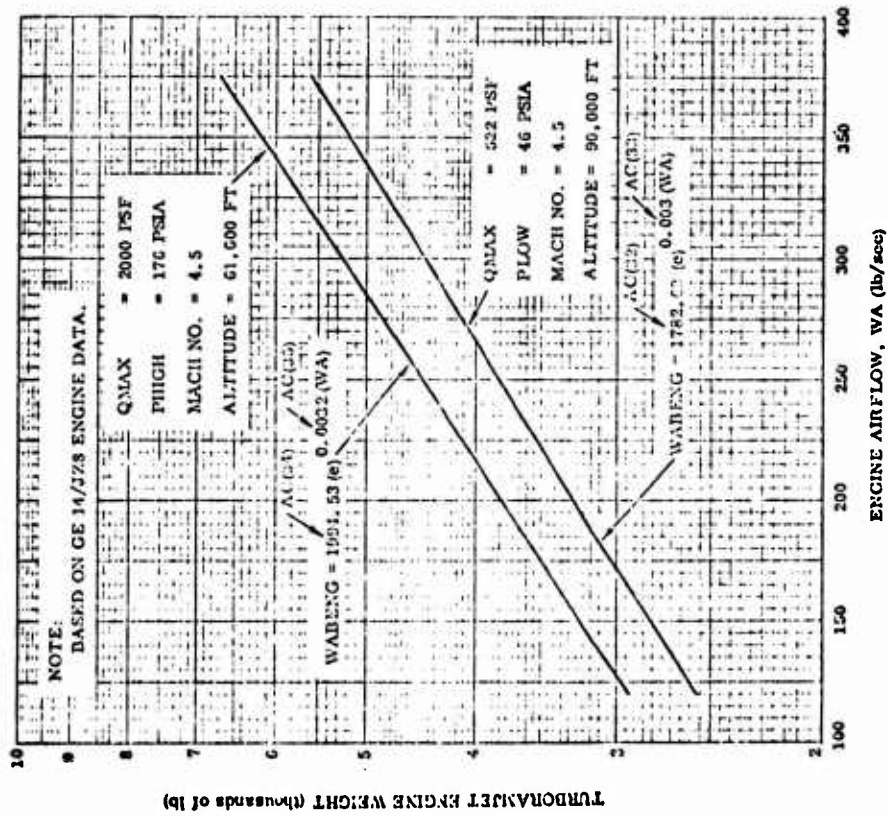


FIGURE 6.1-16. AIRCRAFT TURBORAMJET ENGINE WEIGHT

Reproduced from
 best available copy.

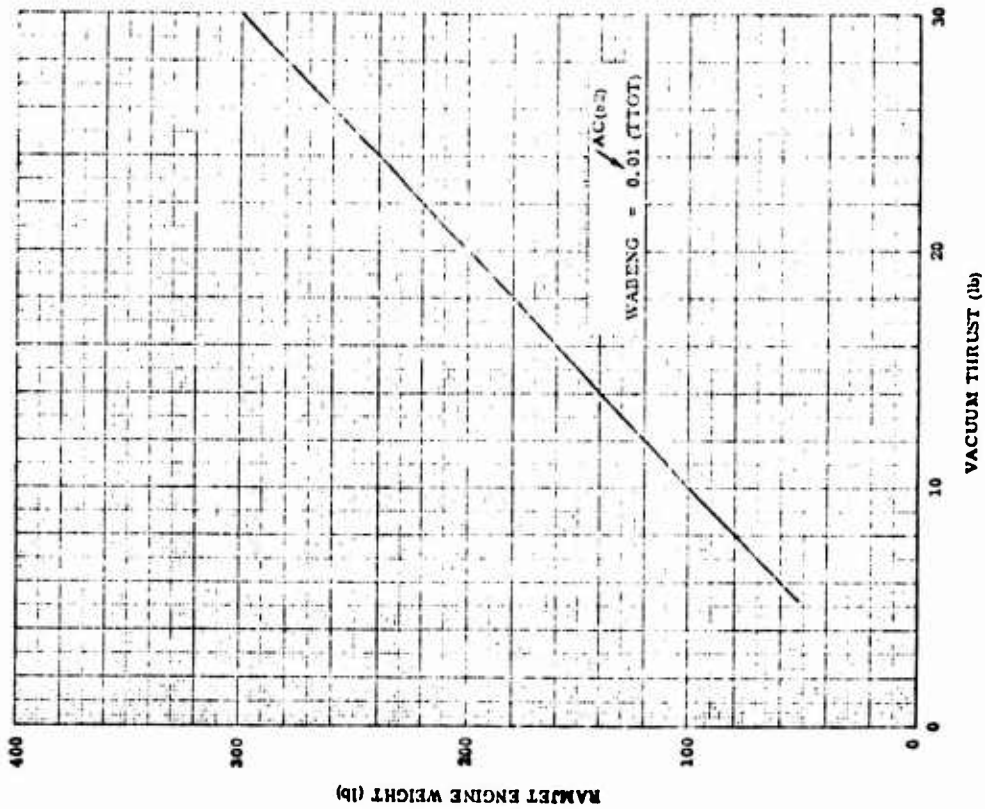


FIGURE 6.1-17. AIRCRAFT RAMJET ENGINE WEIGHT

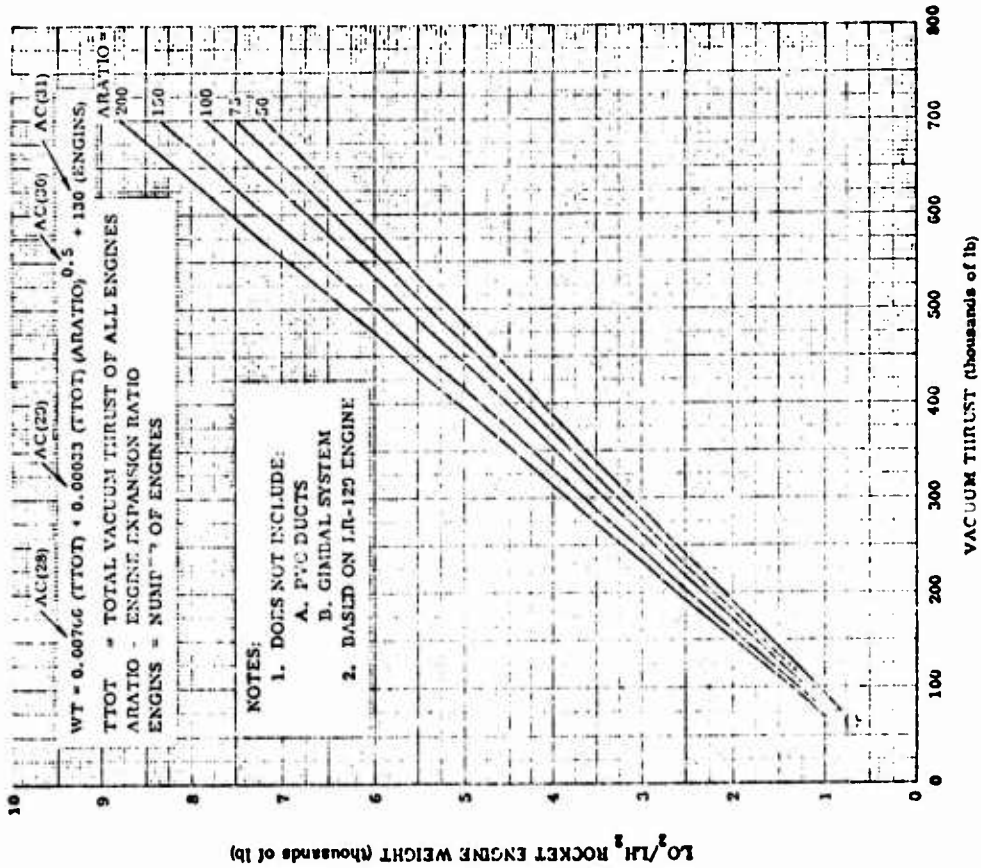


FIGURE 6.1-18. AIRCRAFT ROCKET ENGINE WEIGHT

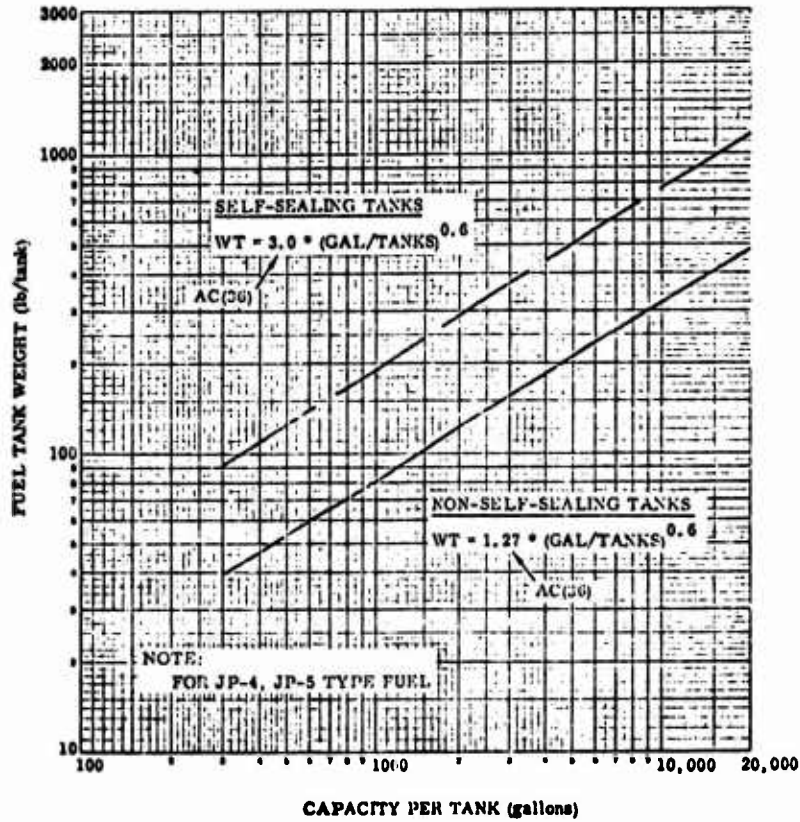


FIGURE 6.1-19. AIRCRAFT FUEL TANK WEIGHT

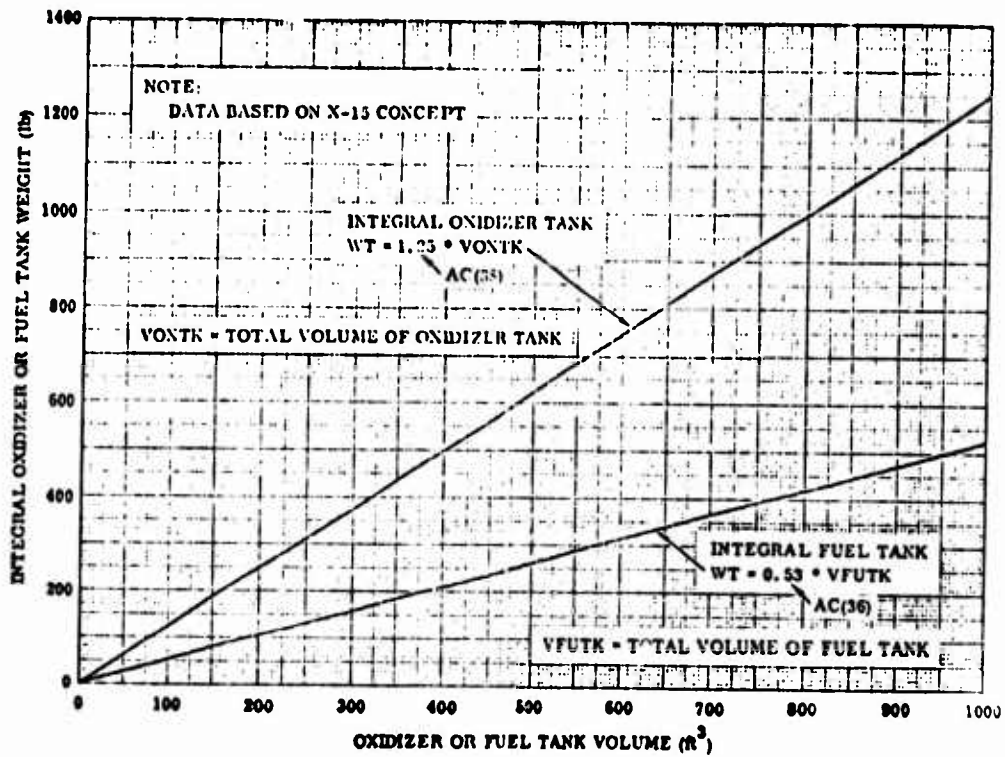


FIGURE 6.1-20. X15 CONCEPT TANK WEIGHTS

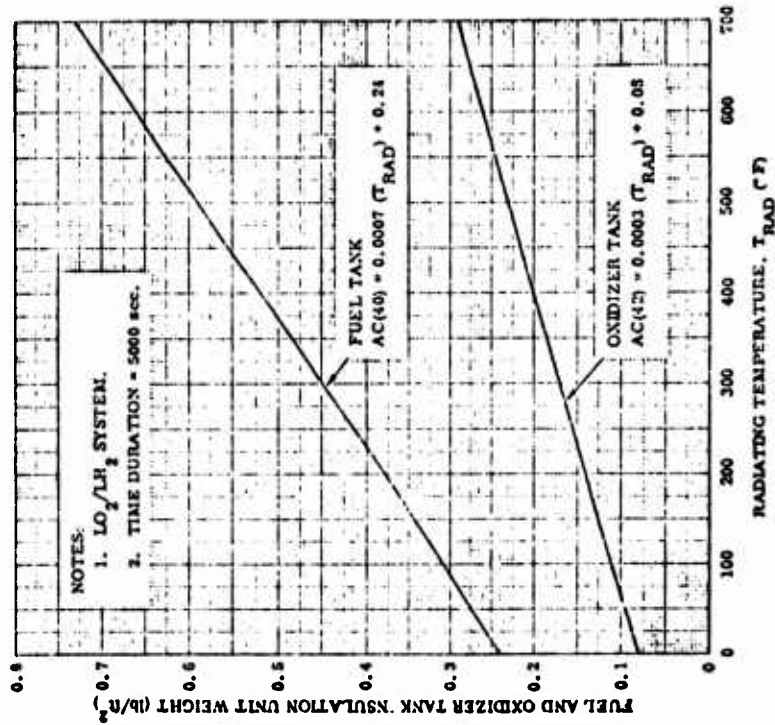


FIGURE 6.1-21. AIRCRAFT AND BOOSTER TANK INSULATION WEIGHTS

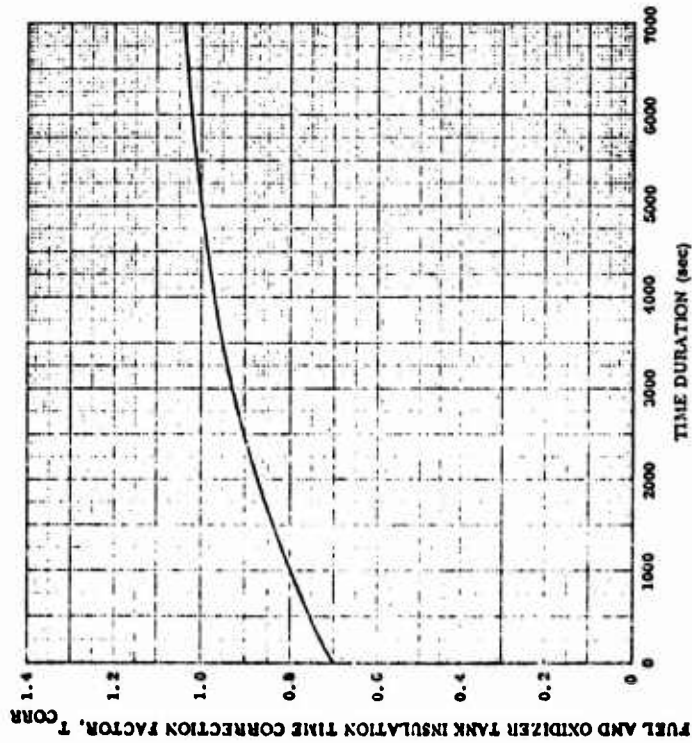


FIGURE 6.1-22. AIRCRAFT AND BOOSTER TANK INSULATION FLIGHT TIME CORRECTION FACTOR

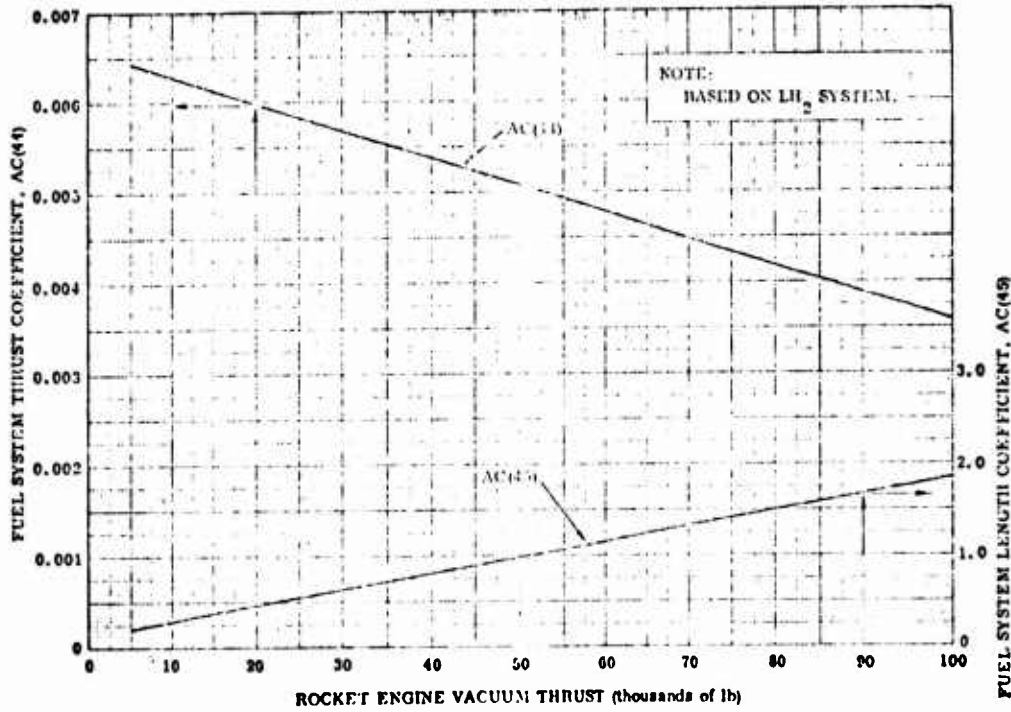


FIGURE 6.1-23. PART I, FUEL DISTRIBUTION SYSTEM WEIGHT

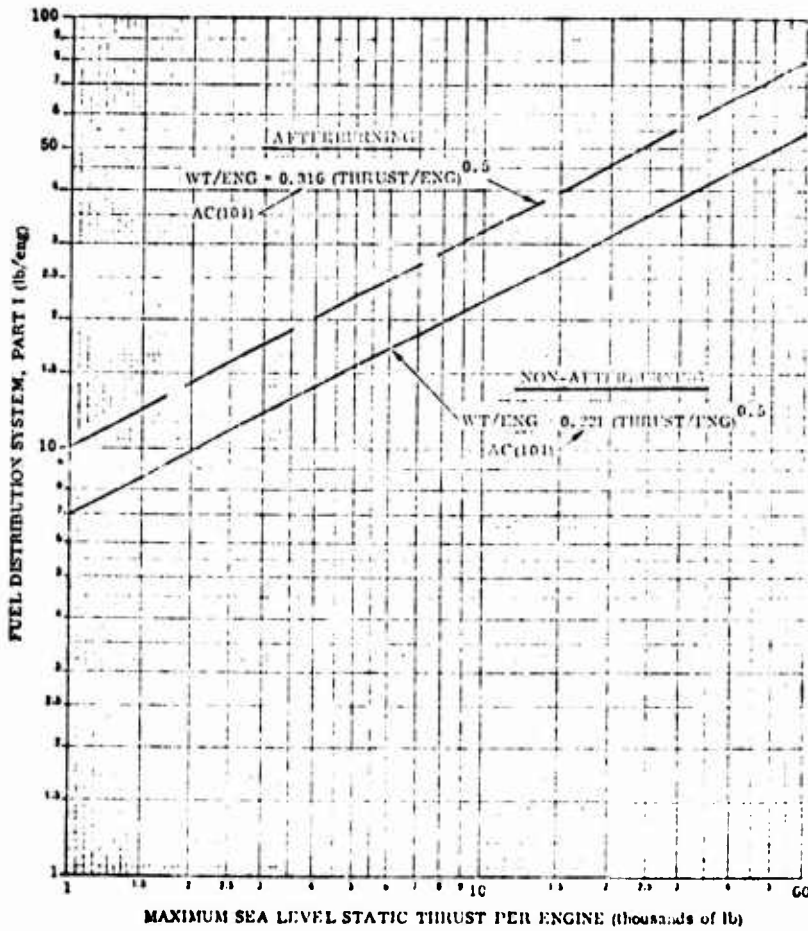


FIGURE 6.1-24.
AIRCRAFT CRYOGENIC
PROPELLANT FUEL SYSTEM
WEIGHTS

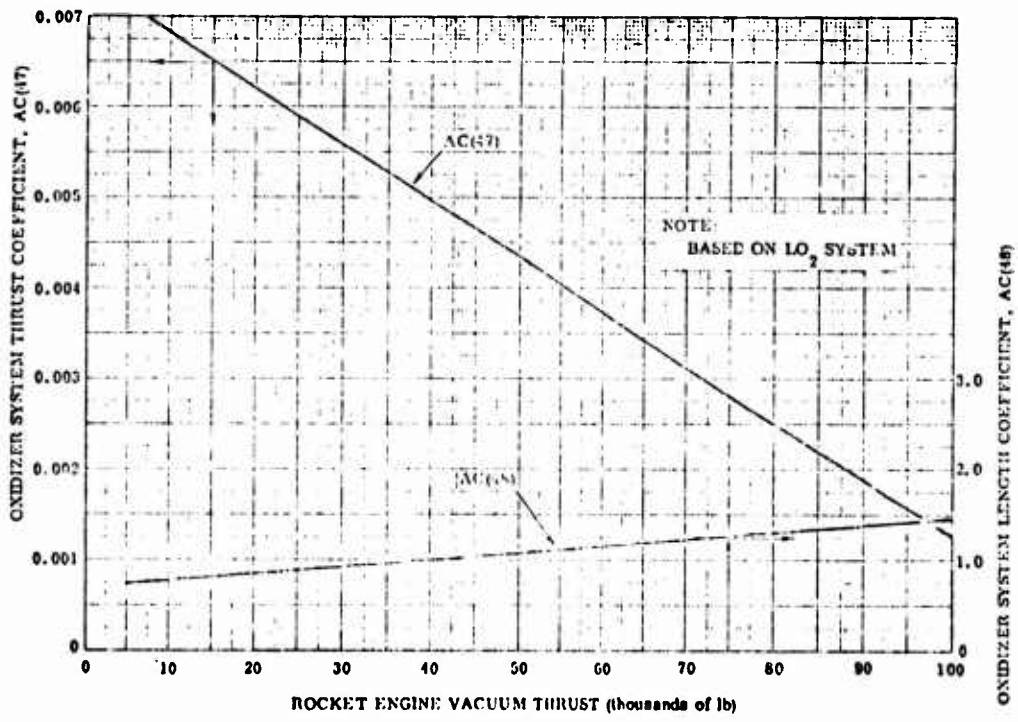


FIGURE 6.1-25. AIRCRAFT CRYOGENIC PROPELLANT OXIDIZER SYSTEM WEIGHTS

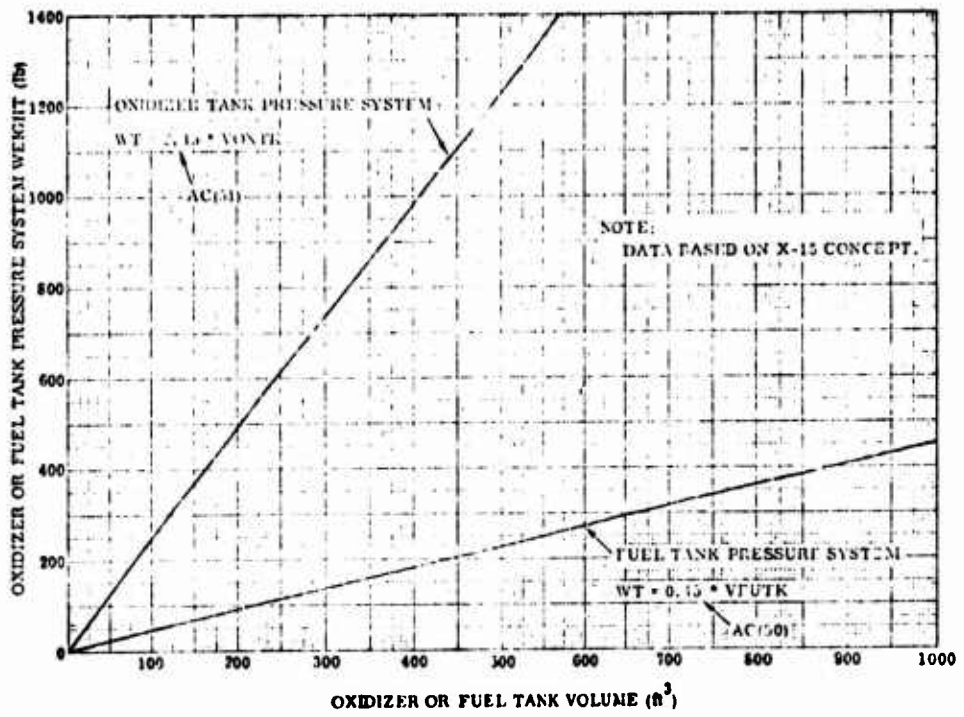


FIGURE 6.1-26. AIRCRAFT CRYOGENIC PROPELLANT PRESSURIZATION SYSTEM WEIGHTS

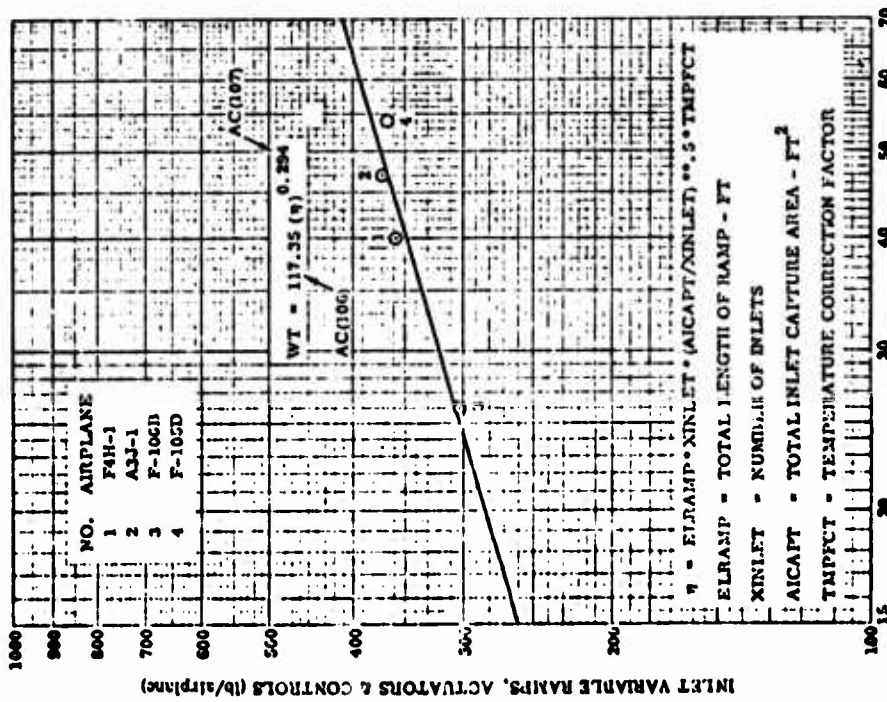


FIGURE 6.1-27. AIRCRAFT INTERNAL DUCT WEIGHT

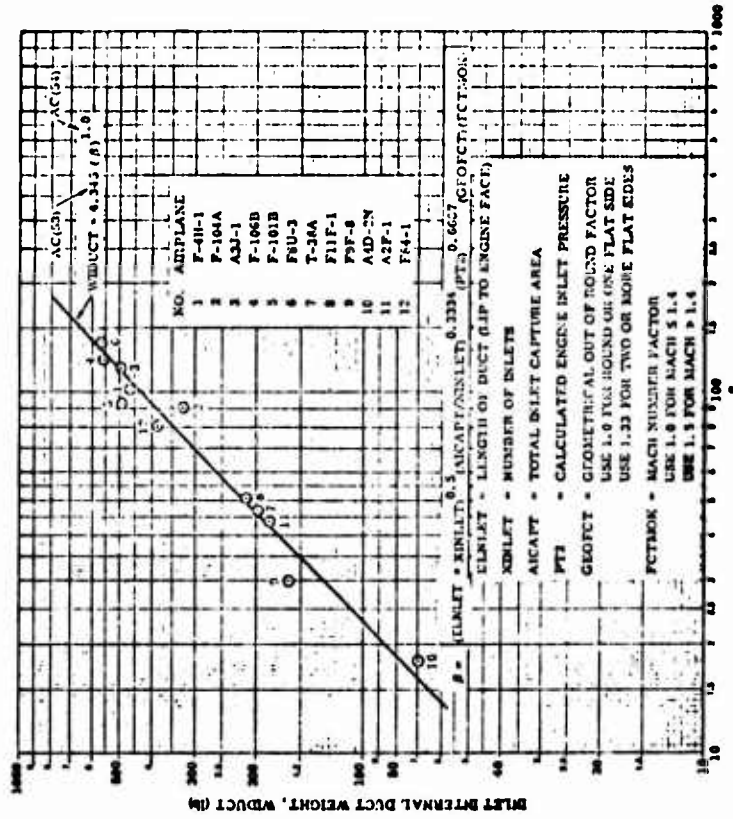


FIGURE 6.1-28. AIRCRAFT INLET VARIABLE RAMPS, ACTUATORS, AND CONTROL WEIGHTS

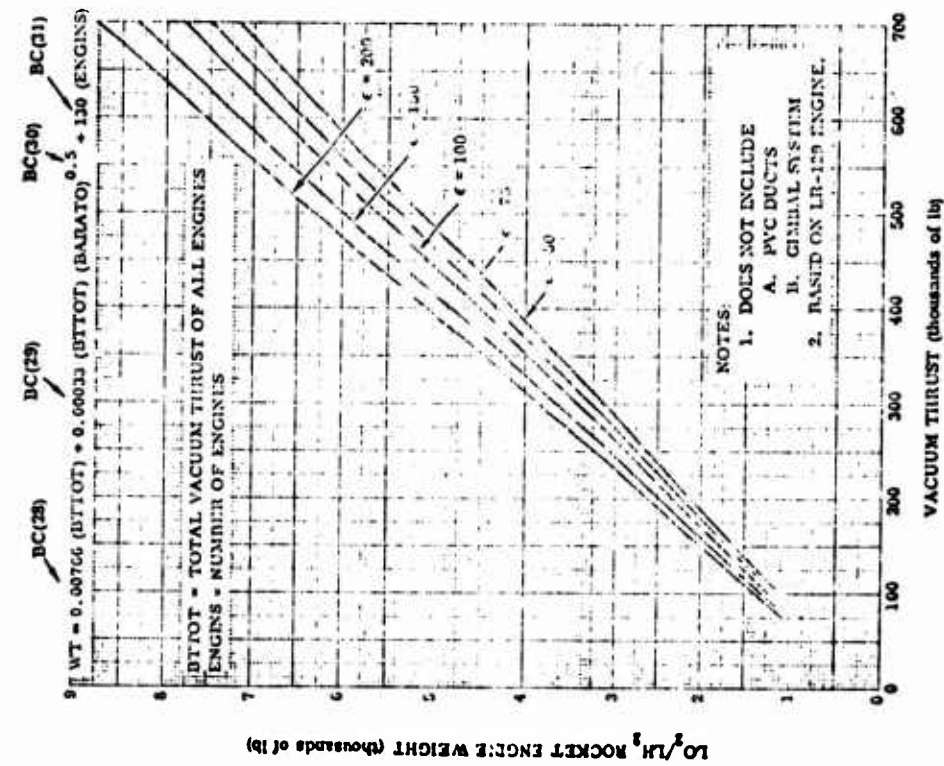


FIGURE 6.1-29. BOOSTER MAIN PROPULSION WEIGHTS

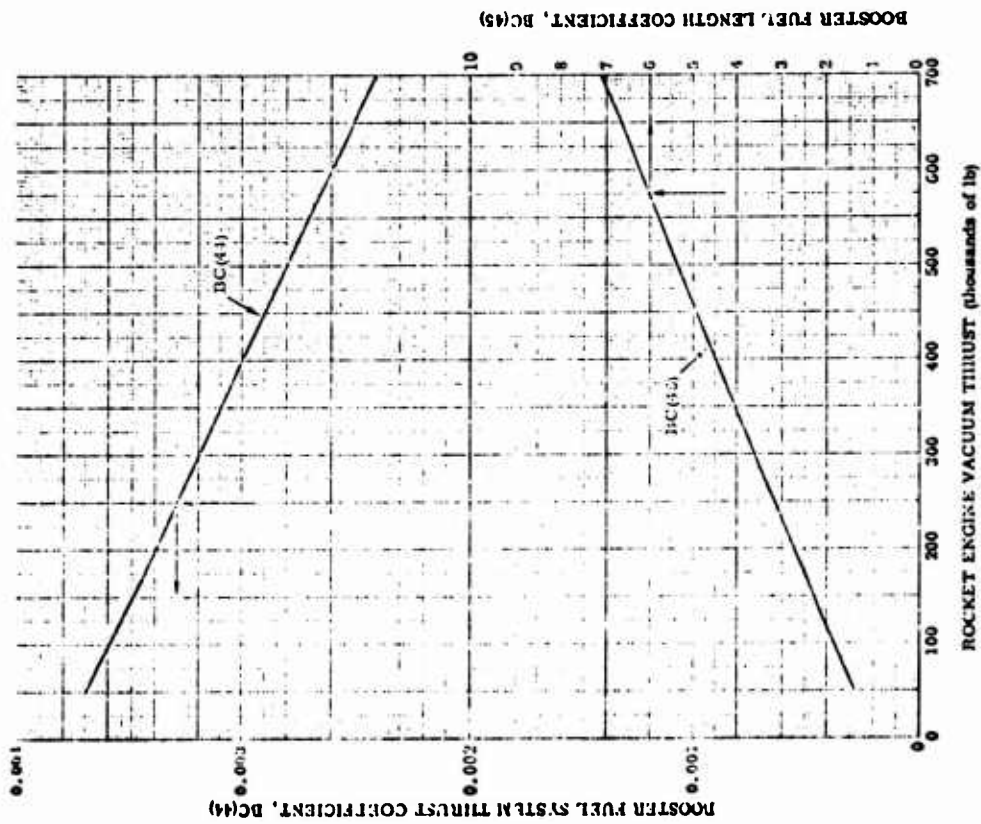


FIGURE 6.1-30. BOOSTER CRYOGENIC PROPELLANT FUEL SYSTEM WEIGHTS

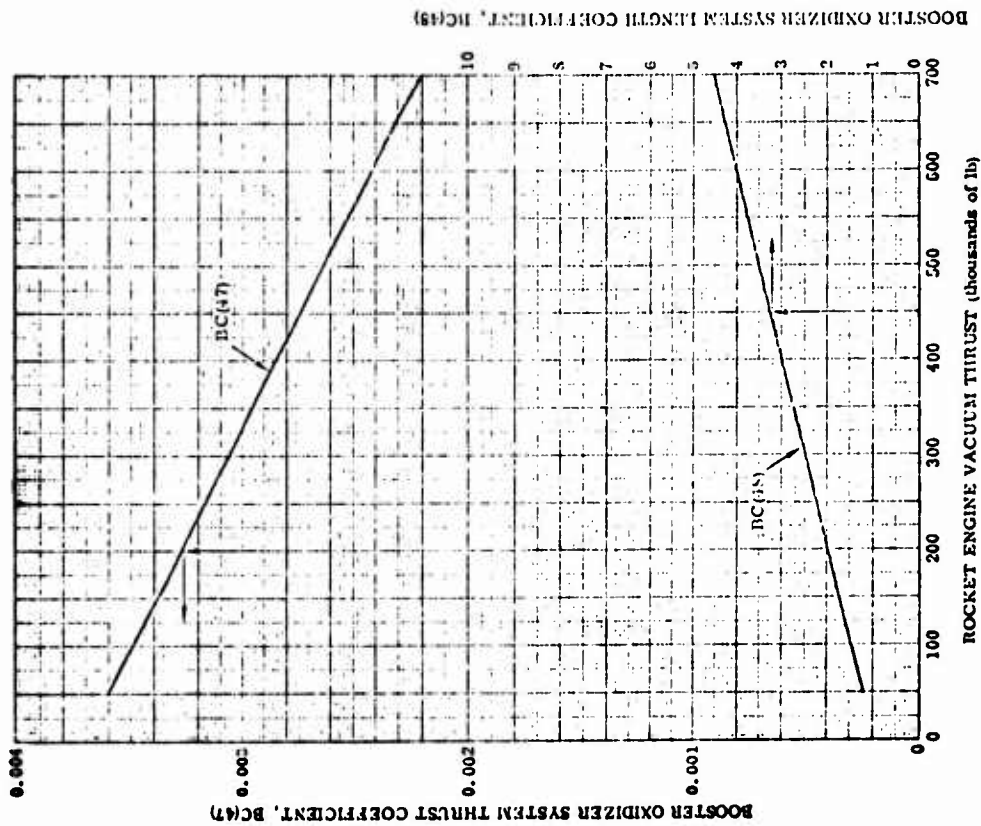


FIGURE 6.1-31. BOOSTER CRYOGENIC PROPELLANT OXIDIZER SYSTEM WEIGHTS

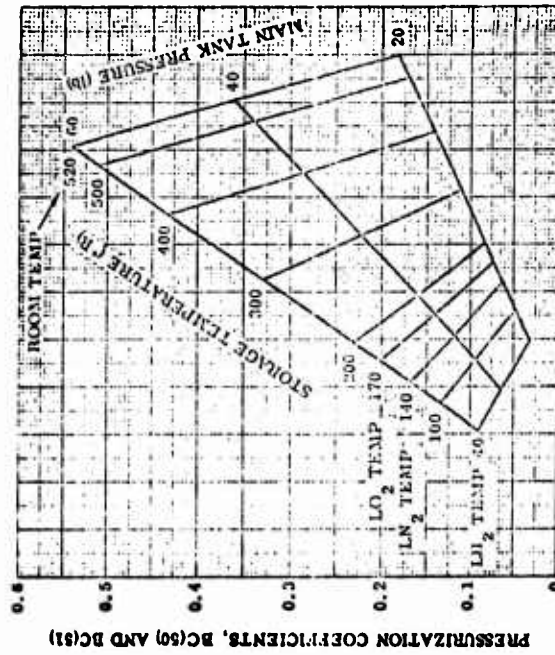


FIGURE 6.1-32. BOOSTER CRYOGENIC PROPELLANT PRESSURIZATION SYSTEM WEIGHTS

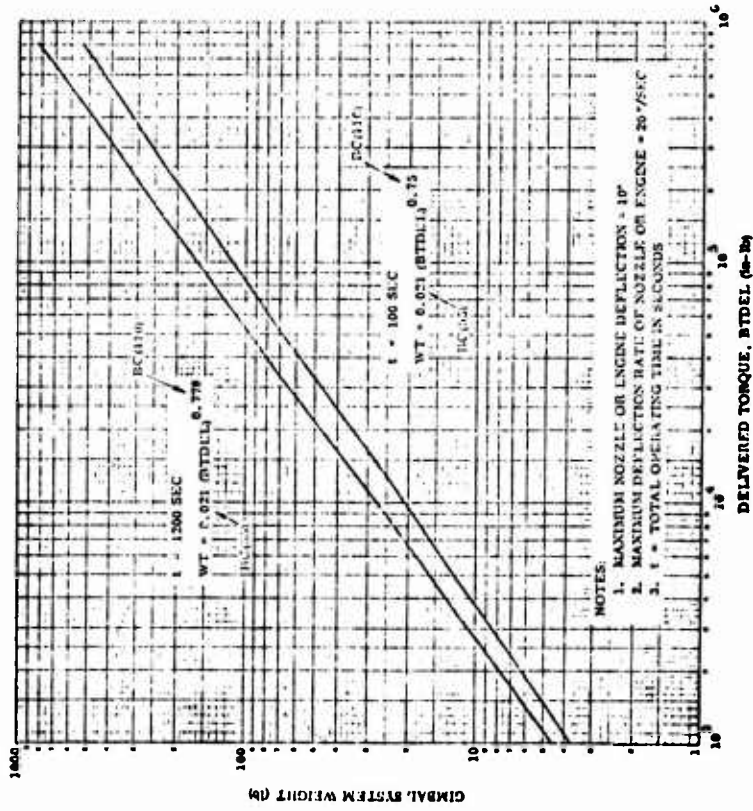


FIGURE 6.1-33. AIRCRAFT AND BOOSTER GIMBAL SYSTEM WEIGHTS

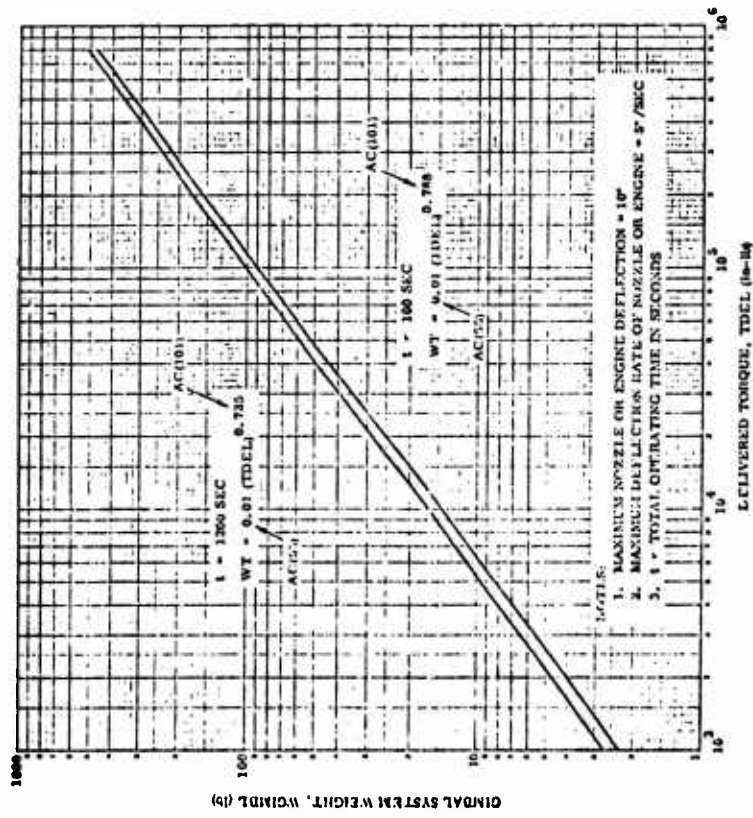


FIGURE 6.1-34. AIRCRAFT GIMBAL SYSTEM WEIGHT
(Five Degrees Per Second)

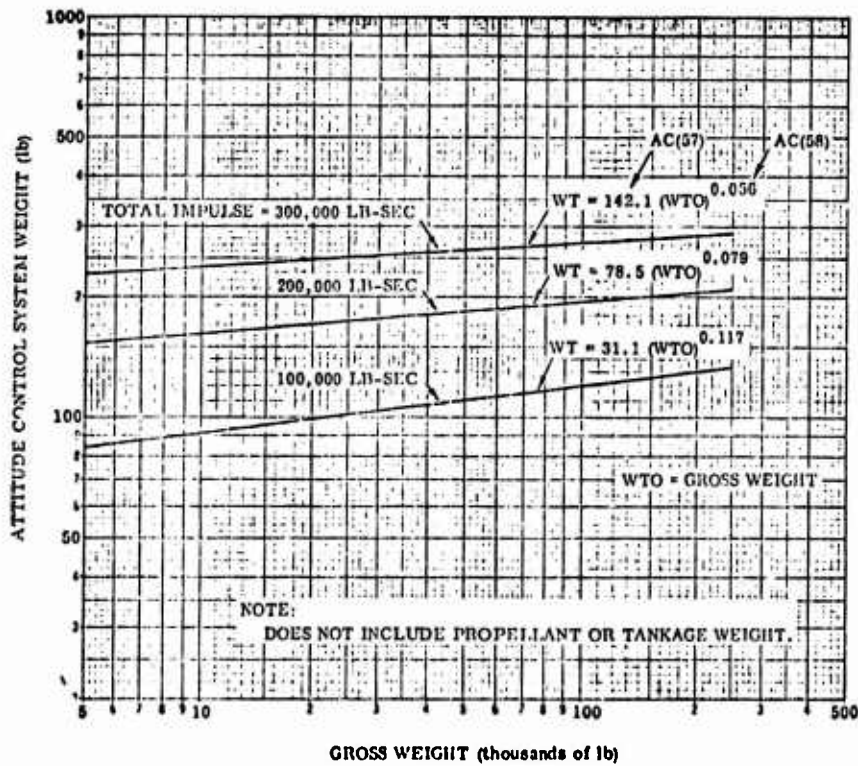


FIGURE 6.1-35. AIRCRAFT ATTITUDE CONTROL SYSTEM WEIGHT

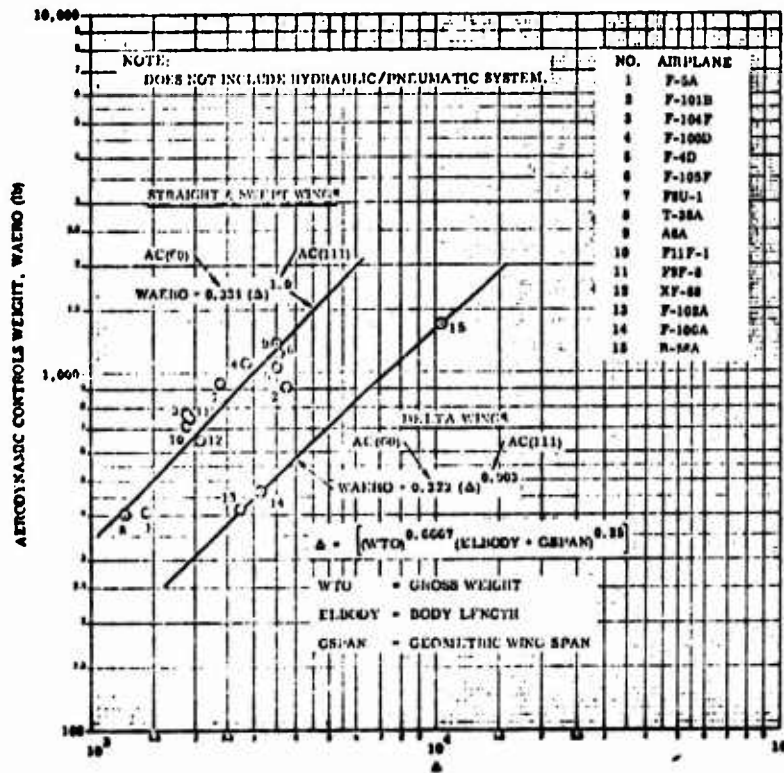


FIGURE 6.1-36. AIRCRAFT AERODYNAMIC CONTROL SYSTEM WEIGHT

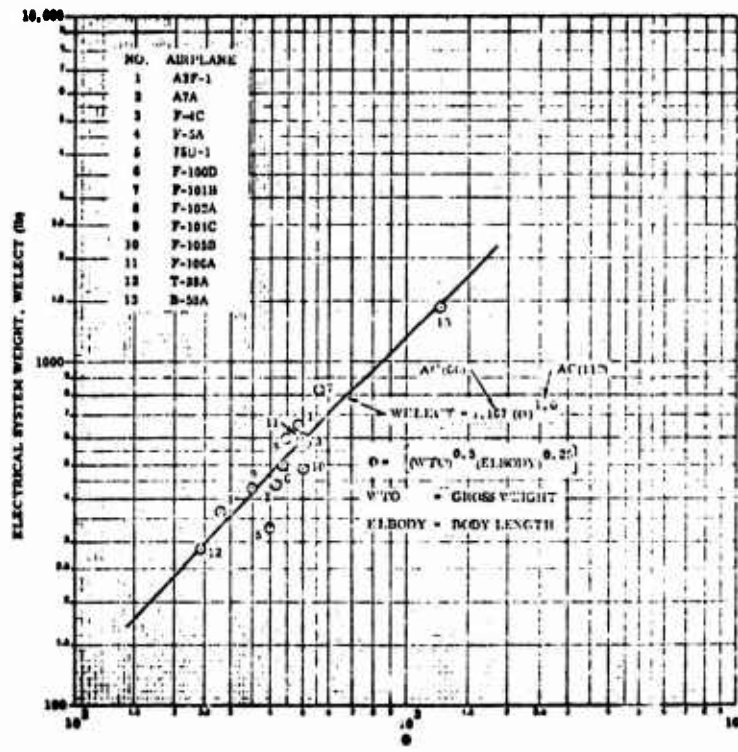
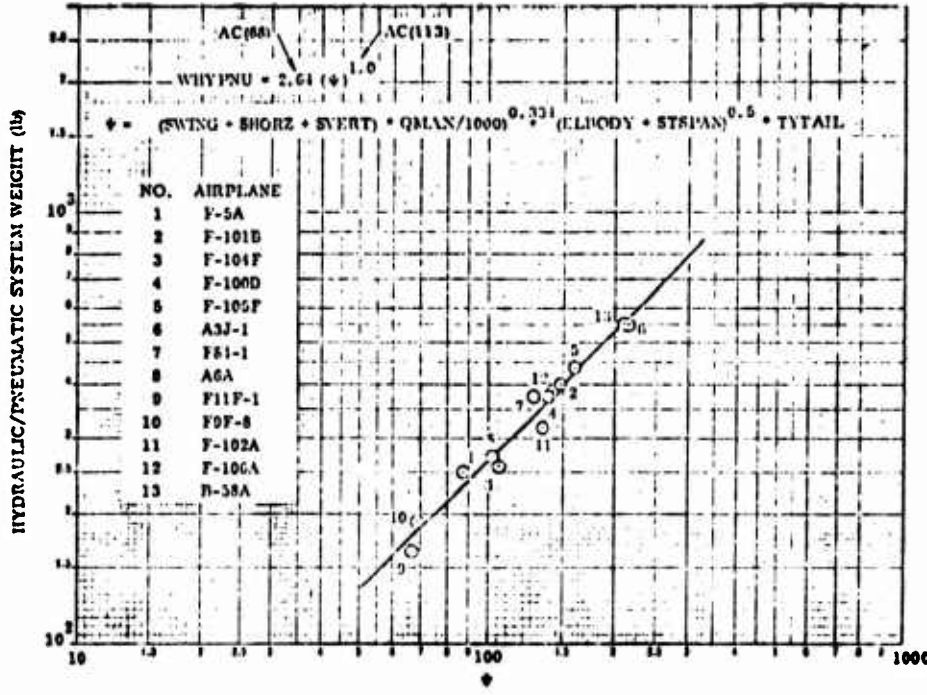


FIGURE 6.1-37. AIRCRAFT ELECTRICAL SYSTEM WEIGHT



6.1-62 FIGURE 6.1-38. AIRCRAFT HYDRAULIC/PNEUMATIC SYSTEM WEIGHT

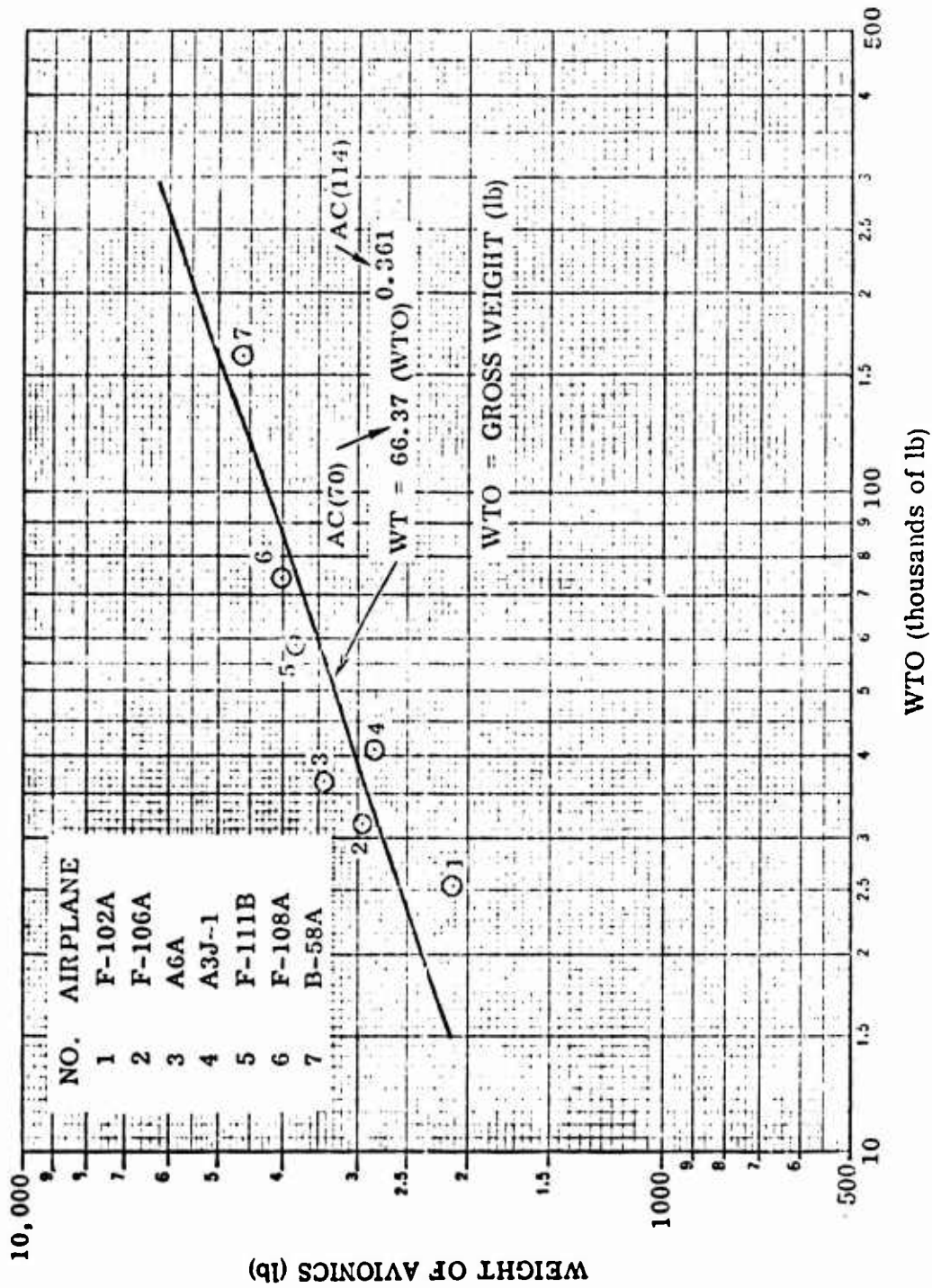


FIGURE 6.1-39. AIRCRAFT AVIONICS WEIGHT

SECTION 7

PERFORMANCE

The ODIN/MFV program library contains three 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

Each program is outlined below; 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.

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/MFV does not exist as an independent code; rather it is an option in the Section 7.2 NSEG 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*

$$\alpha_G = \alpha_{BG} + \alpha_{WB} \quad (7.1.1)$$

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

$$\alpha_R = \alpha_{BMAX} - 1.0 + \alpha_{WB} \quad (7.1.2)$$

The additional symbols are

α_R = wing incidence following rotation
 α_{BMAX} = 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

$$C_{LMAX} = (C_{LMAX})_{BASE} + \Delta C_{LMAX} + \Delta C_{LFLAP} \quad (7.1.3)$$

with a corresponding angle of attack

$$\alpha' = (\alpha_{C_{LMAX}})_{BASE} \quad (7.1.4a)$$

During take-off the maximum angle of attack, α_{MAX} , is limited to

$$\alpha_{MAX} = 0.8 \cdot \alpha'_{MAX} \quad (7.1.4b)$$

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 $C_{L_{\alpha}}$ 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

$$\beta = \sqrt{1-M^2} = 1.0 \quad (7.1.5)$$

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 $M \leq 0.2$ 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

$$\Delta C_{L_{FLAP}} = 10.95 (C_{L_{\alpha}})_{BASE} \left[1.5 \left(\frac{B_F}{B_{WE}} \right) - 0.5 \left(\frac{B_F}{B_{WE}} \right)^2 \right] \times \left(\frac{\bar{C}_F}{\bar{C}_{WE}} \right)^{1/2} \left[.0625 \int_F - .000472 \int_F^2 \right] \quad (7.1.6)$$

where

$(C_{L_{\alpha}})_{BASE}$ = 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

$$C_{LG} = 57.29 C_{L\alpha} \cdot \sin(\alpha_G) \cos^2(\alpha_G) + \left[\frac{C_{LMAX}}{\cos(\alpha_{MAX})} - \frac{57.29}{2} C_{L\alpha} \sin(\alpha_{MAX}) \right] \\ = F(\alpha_G) \sin^2(\alpha_G) \cos(\alpha_G) / \sin^2(\alpha_{MAX}) \quad (7.1.7)$$

where $C_{L\alpha}$ is the linear lift curve slope. The corresponding ground roll drag is taken as

$$C_{DG} = C_{D0} + k \cdot C_{LG} [C_{LG} - C_{L0} - \Delta C_{LFLAP}] + C_{D_{LG}} \quad (7.1.8)$$

where

- C_{D0} = zero lift drag coefficient
- k = induced drag factor
- C_{L0} = lift coefficient at zero wing incidence
- $C_{D_{LG}}$ = landing gear drag coefficient

7.1.1.1(c) Rotation Lift and Drag

The lift coefficient after rotation, C_{LR} , is given by Equation (7.1.7) with α_R replacing α_G ; that is,

$$C_{LR} = F(\alpha_R) \quad (7.1.9)$$

The lift coefficient is subject to the condition that

$$C_{LR} \leq (C_{LMAX}) / (1.1)^2 \quad (7.1.10)$$

This inequality constraint is imposed to prevent buffet or pitch-up problems.

The drag coefficient after rotation is given by

$$C_{DR} = C_{D0} + k C_{LR} [C_{LR} - C_{L0} - \Delta C_{LFLAP}] + C_{D_{LG}} \quad (7.1.11)$$

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.

$$C_{L50} = C_{LR} / (1.1)^2 \quad (7.1.12)$$

The corresponding drag is given by

$$C_{D50} = C_{D0} + k \cdot C_{L50} [C_{L50} - C_{L0} - \Delta C_{LFLAP}] + C_{D_{LG}} \quad (7.1.13)$$

7.1.2 Ground Roll and Rotation

The ground roll distance, X_G , is based on the expression

$$X_G = \frac{13.07 \left(\frac{W_{TO}}{S \cdot C_{LR}} \right)}{\left(\frac{F_{TO}}{W_{TO}} - \mu_G - \frac{1}{2} \left(\frac{C_{DGR}}{C_{LR}} \right) \right)} \quad (7.1.14)$$

where

$$C_{DGR} = \frac{1}{2}(C_{DG} + C_{DR}) \quad (7.1.15)$$

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

$$T_G = 1.1842 X_G / V_R \quad (7.1.16)$$

where the velocity at rotation, V_R , is given by

$$V_R = 17.16 \sqrt{\frac{W_{TO}}{S C_{LR}}} \quad (7.1.17)$$

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

$$C_{DR50} = \frac{1}{2}(C_{DR} + C_{D50}) \quad (7.1.18)$$

The distance covered in clearing the obstacle is given by

$$X_{50} = \frac{50.0 + 2.745 \left(\frac{W_{TO}}{S \cdot C_{LR}} \right)}{\left(\frac{F_{TO}}{W_{TO}} - 1.105 \left(\frac{C_{DR50}}{C_{LR}} \right) \right)} \quad (7.1.19)$$

Time to clear the obstacle after rotation is

$$T_{50} = X_{50} / (1.6889 \times V_R) \quad (7.1.20)$$

Thus, total distance for take-off over 50 foot obstacle is

$$X_{TO} = X_G + X_{50} \quad (7.1.21)$$

The elapsed time is

$$T_{TO} = T_G + T_{50} \quad (7.1.22)$$

Total fuel used is

$$W_{FTO} = F_{TO} \cdot T_{TO} / (I_{SP_{TO}}) \quad (7.1.23)$$

At the 50 foot obstacle the flight path angle is obtained from

$$\sin(\gamma_{50}) = \frac{F_{TO}}{W_{50}} - \frac{C_{D50}}{C_{L50}} \quad (7.1.24)$$

where

$$W_{50} = W_{TO} - W_{FTO} \quad (7.1.25)$$

The corresponding rate of climb is given by

$$RC_{50} = 1.6889 V_{50} \sin(\gamma_{50}) \quad (7.1.26)$$

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

$$\alpha_{TD} = \alpha_{BTD} - 1.0 + \alpha_{WB} \quad (7.1.27)$$

and in the subsequent ground roll

$$\alpha_{LR} = \alpha_{BLR} + \alpha_{WB} \quad (7.1.28)$$

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

$$C_{L50} = C_{LTD} / (1.1)^2 \quad (7.1.29)$$

At touchdown, C_{LTD} , is based on Equation (7.1.7) using α_{TD} ; that is

$$C_{LTD} = F(\alpha_{TD}) + \Delta C_{LFLAP} \quad (7.1.30)$$

where ΔC_{LFLAP} is given by Equation (7.1.6) using the landing flap setting. The inequality

$$C_{LTD} < C_{LMAX} / (1.1)^2 \quad (7.1.31)$$

is used.

Similarly, during the subsequent landing ground roll,

$$C_{L LR} = F(\alpha_{LR}) \quad (7.1.32)$$

Drag coefficient at the 50 foot obstacle, C_{DL50} , 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

$$C_{D LR} = C_{D0} + k \cdot C_{L LR} (C_{L LR} - C_{L0}) + C_{D LG} + C_{D CHUT} \quad (7.1.33)$$

where

$$C_{D CHUT} = \text{landing parachute drag}$$

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

$$V_{TD} = 17.16 \sqrt{\frac{W_L}{S \cdot C_{LTD}}} \quad (7.1.34)$$

The ground distance covered from 50 foot obstacle to touchdown is

$$C_{L50} = \frac{30.0 + 2.745 \left(\frac{W_L}{S \cdot C_{LTD}} \right)}{1.105 \left(\frac{C_{D_{TD50}}}{C_{LTD}} \right) - \left(\frac{F_L}{W_L} \right)} \quad (7.1.35)$$

where

X_{L50} = flight distance from 50 foot obstacle to touchdown

W_L = landing weight

$C_{D_{TD50}}$ = $\frac{1}{2}(C_{D50} + C_{D_{TD}})$, the average drag coefficient

F_L = approach thrust

Rate of sink at the 50 foot obstacle is

$$RS_{50} = 1.69 V_{50} \left[\left(\frac{C_{D_{L50}}}{C_{L_{L50}}} \right) - \left(\frac{F_{L50}}{W_L} \right) \right] \quad (7.1.36)$$

Flight path angle at the 50 foot obstacle is given by

$$\sin(\gamma_{L50}) = \left(\frac{C_{D_{L50}}}{C_{L_{L50}}} \right) - \left(\frac{F_{L50}}{W_L} \right) \quad (7.1.37)$$

The ground roll distance is given by

$$X_{GL} = \frac{13.07 \left(\frac{W_L}{S \cdot C_{LTD}} \right)}{\mu + \frac{1}{2}(C_{D_{LG}} - \mu \cdot C_{L_{LR}}) / C_{LTD}} \quad (7.1.38)$$

Total landing distance is

$$X_L = X_{L50} + X_{GL} \quad (7.1.39)$$

REFERENCES:

1. ———, United States Air Force Stability and Control DATCOM, Prepared by Douglas Division of McDonnell-Douglas Corporation under contracts to the United States Air Force, October 1960.

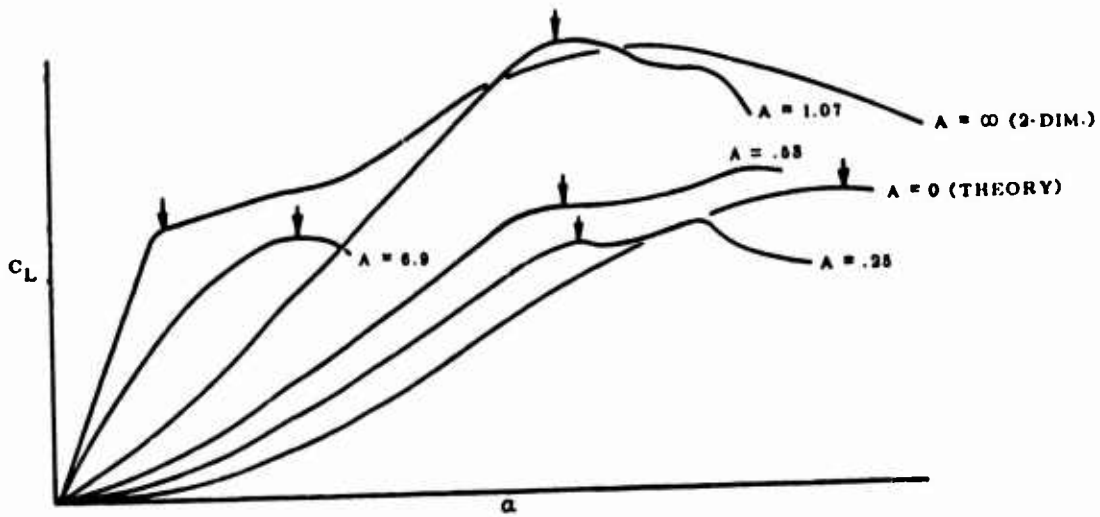


FIGURE 7.1-1. TYPICAL FIRST PEAKS IN LIFT COEFFICIENT VERSUS ANGLE OF ATTACK

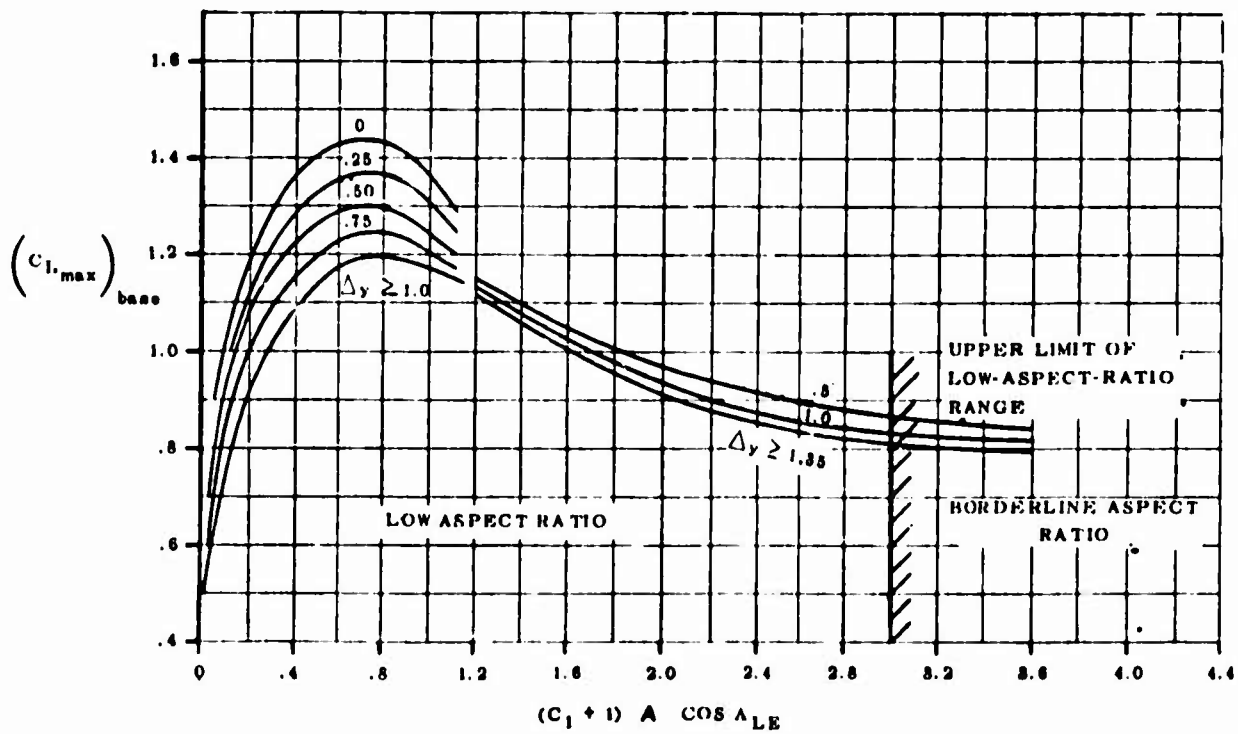


FIGURE 7.1-2. SUBSONIC MAXIMUM LIFT OF WINGS WITH POSITION OF MAXIMUM THICKNESS BETWEEN 35 AND 50 PER CENT CHORD

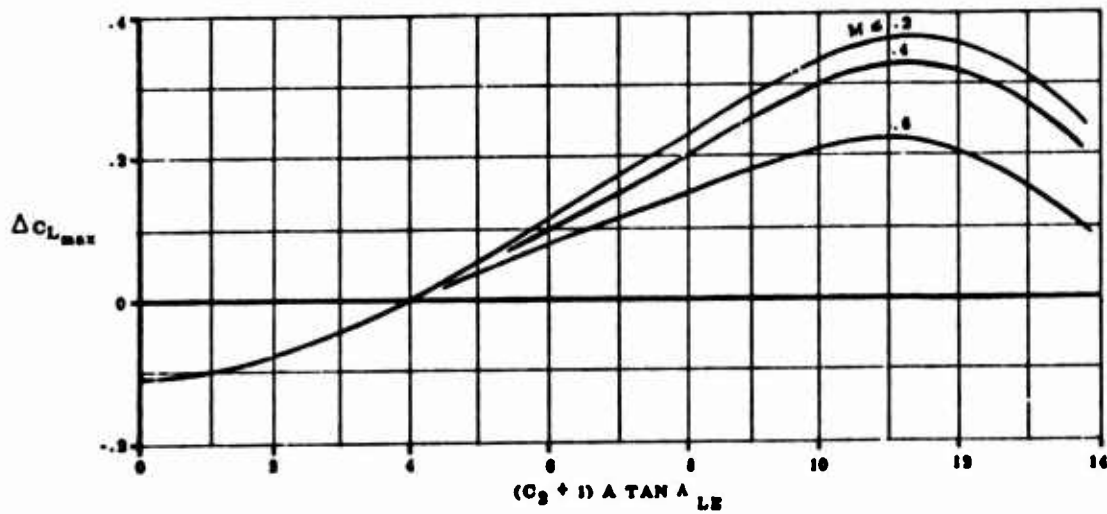


FIGURE 7.1-3. SUBSONIC MAXIMUM LIFT INCREMENT FOR LOW ASPECT RATIO WINGS

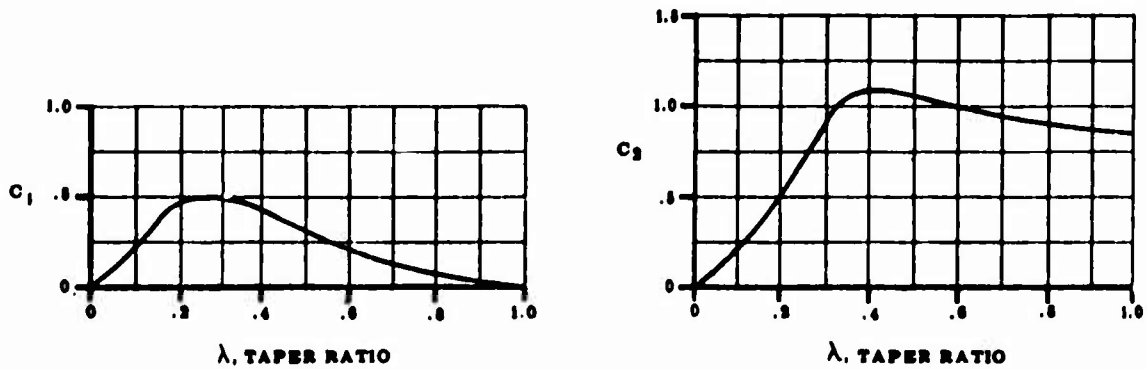


FIGURE 7.1-4. TAPER RATIO CORRECTION FACTORS

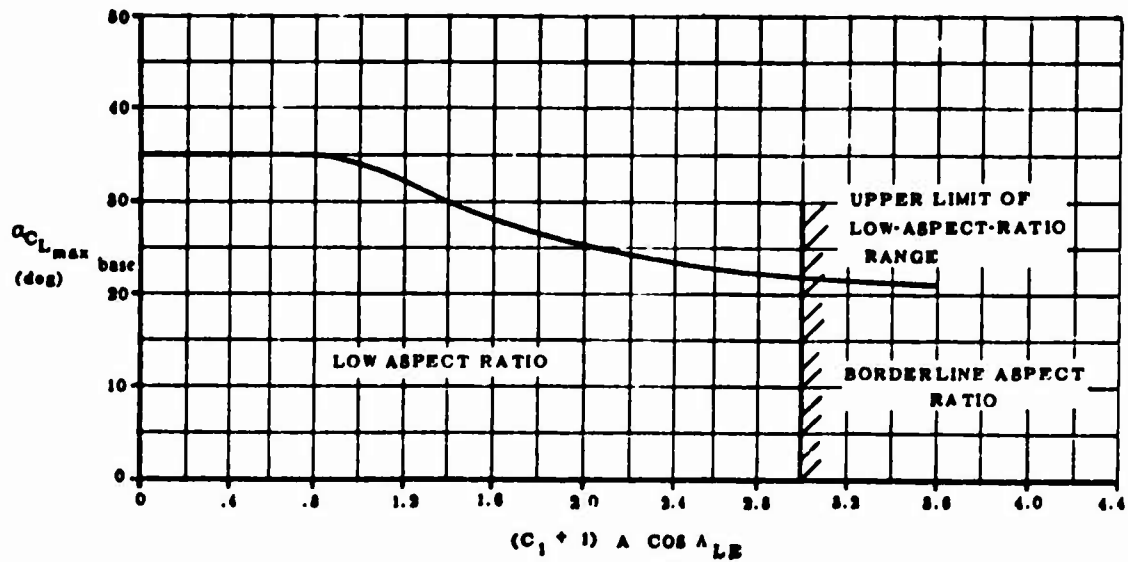


FIGURE 7.1-5. ANGLE OF ATTACK FOR SUBSONIC MAXIMUM LIFT OF LOW ASPECT RATIO WINGS

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.

$$\{X_i\} = \{V, h, \gamma, W, R, t\} \quad (7.2.1)$$

In all flight modes the equations of motion

$$\{X_i\} = \{f_i(V, h, \gamma, W, R, t; \alpha, B_A, N)\} \quad (7.2.2)$$

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

$$\phi = \phi(X_i) \quad (7.2.3)$$

is evaluated.

2. *Vector performance evaluation* where given

$$\{\bar{x}\} = \{\dots, X_i, X_j, \dots\} \quad j = 1, 2, \dots, N_j \quad (7.2.4)$$

the vector

$$\{\phi\} = \{\phi_j\} = \{f(\dots, X_i, X_j, \dots)\} \quad j = 1, 2, \dots, N_j \quad (7.2.5)$$

is evaluated and the maximum or minimum value of ϕ in the region

$$X_{jL} < X_j < X_{jH} \quad (7.2.6)$$

is found by interpolation. That is,

$$\phi_j^* = f(\dots, X_i, X_j^*, \dots) \quad (7.2.7)$$

3. *Map performance evaluation* where given

$$\{\bar{x}\}_{ij} = \{\dots, X_i, X_j, \dots\} \quad \begin{array}{l} i = 1, 2, \dots, N_i \\ j = 1, 2, \dots, N_j \end{array} \quad (7.2.8)$$

the performance array

$$[\phi_{ij}] = [f(\dots, X_i, X_j, \dots)] \quad (7.2.9)$$

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

$$\{x\}_1 \rightarrow T_{12} \rightarrow \{\bar{x}\}_2 \quad (7.2.10)$$

is accomplished.

5. *Mission performance* where given a sequence of mission segments, the successive state transformations

$$\{\bar{x}\}_1 \rightarrow \{\bar{x}\}_2 \rightarrow \dots \rightarrow \{\bar{x}_{N-1}\} \rightarrow \{\bar{x}_N\} \quad (7.2.11)$$

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

$$C_D = C_{D_0} + C_{D_i} \quad (7.2.12)$$

where C_{D_0} is the zero lift drag, and C_{D_i} is the induced drag.

Both C_{D_0} and C_{D_i} may be computed by three component summation; that is,

$$C_{D_0} = C_{D_{01}} + C_{D_{02}} + C_{D_{03}} \quad (7.2.13)$$

$$C_{D_i} = C_{D_{i1}} + C_{D_{i2}} + C_{D_{i3}} \quad (7.2.14)$$

In Equation (7.2.13) each of the three component zero lift drags must be of the form

$$C_{D_{0j}} = C_{D_{0j}}(h, M); \quad j = 1, 2, 3 \quad (7.2.15)$$

Similarly, in Equation (7.2.14) each induced drag component must be of the form

$$C_{D_{ij}} = C_{D_{ij}}(C_L, M); \quad j = 1, 2, 3 \quad (7.2.16)$$

7.2.1.1(b) Polynomial Form

In this aerodynamic option the drag is computed in the component summation form

$$C_D = C_{D_1} + C_{D_2} + C_{D_3} \quad (7.2.17)$$

where

$$C_{D_j} = C_{D_{0j}} + k_{1j} \cdot C_L^2 + k_{2j}(C_L - C_{L_{MIN}}) + k_{3j}C_L^3$$
$$j = 1, 2, 3 \quad (7.2.18)$$

and

$$C_{D0j} = C_{D0j}(M) \quad (7.2.19)$$

$$k_{1j} = K_{1j}(M) \quad (7.2.20)$$

$$k_{2j} = k_{2j}(M) \quad (7.2.21)$$

$$k_{3j} = k_{3j}(M) \quad (7.2.22)$$

$$C_{L\text{MIN}} = C_{L\text{MIN}}(M), \text{ the minimum drag lift coefficient} \quad (7.2.23)$$

7.2.1.2 Store and Pylon Drag

Store and pylon drag is computed in the form

$$C_{DS} = C_{D1} + C_{D2} + C_{D3} \quad (7.2.24)$$

where

$$C_{Dj} = C_{DSj} \cdot NS_j + C_{DSPj} \cdot NSP_j \quad j = 1, 2, 3 \quad (7.2.25)$$

In Equation (7.2.25) the drag of a single type j store pair is

$$C_{DSj} = C_{DSj}(M) \quad (7.2.26)$$

The number of type j store pairs is NS_j . The drag of a single type j store pylon pair is

$$C_{DSPj} = C_{DSPj}(M) \quad (7.2.27)$$

The number of type j store pylon pairs is NSP_j

7.2.1.3 Tank and Pylon Drag

Tank and pylon drag is computed in the form

$$C_{DT} = C_{D1} + C_{D2} + C_{D3} \quad (7.2.28)$$

where

$$C_{Dj} = C_{DTj} \cdot NT_j + C_{DTPj} \cdot NTP_j \quad j = 1, 2, 3 \quad (7.2.29)$$

In Equation (7.2.29) the drag of a single type j tank pair is

$$C_{DTj} = C_{DTj}(M) \quad (7.2.30)$$

The number of type j tank pairs is N_{T_j} . The drag of a single type j tank pylon pair is

$$C_{DTP_j} = C_{DTP_j}(M) \quad (7.2.31)$$

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

$$T_{max} = T_{max1}, T_{max2}, \text{ or } T_{max3} \quad (7.2.32)$$

where

$$T_{maxj} = T_{maxj}(M, h) \quad (7.2.33)$$

A throttle parameter, N , is determined by

$$N = T_{reqd}/T_{maxj} \quad j = 1, 2, 3 \quad (7.2.34)$$

where T_{reqd} is the required thrust. Fuel flow is given by

$$\dot{W}_1 = k \cdot \dot{W}_1(N, M, h) \quad (7.2.35)$$

or

$$\dot{W}_2 = k \cdot \dot{W}_2(N, M, h) \quad (7.2.36)$$

or

$$\dot{W}_3 = k \cdot \dot{W}_3(M, h) \quad (7.2.37)$$

The parameter k is a scalar for adjusting fuel flow to meet various specification requirements. Vehicle thrust, T , is determined by

$$T = T_{reqd} \quad (7.2.38)$$

7.2.3 Vehicle Mass Representation

7.2.3.1 Overall Weight Empty

The vehicle overall weight empty is given by the following equation:

$$\begin{aligned}
W_{\text{OWE}} = & W_{\text{BARE}} + \sum_{i=1}^3 (N_{\text{S}_i} \cdot W_{\text{S}_i})_{\text{fixed}} + \sum_{i=1}^3 (N_{\text{SP}_i} \cdot W_{\text{SP}_i})_{\text{fixed}} \\
& + \sum_{i=1}^3 (N_{\text{T}_i} \cdot W_{\text{T}_i})_{\text{fixed}} + \sum_{i=1}^3 (N_{\text{TP}_i} \cdot W_{\text{TP}_i})_{\text{fixed}} \quad (7.2.39)
\end{aligned}$$

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 pair 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

$$W_{\text{FUEL}} = W_{\text{FINT}} + \sum_{i=1}^3 (N_{\text{T}_i} \cdot W_{\text{FT}_i})_{\text{usable}} \quad (7.2.40)$$

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

$$W_{FT_0} = W_{F_{INT}} + \sum_{i=1}^3 (N_{T_i} \cdot W_{FT_i})_{NPL} \quad (7.2.41)$$

where

W_{FT_0} = 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

$$W_{PL} = W_{PL_{int}} + \sum_{i=1}^3 (N_{S_i} \cdot W_{S_i})_{drop} + \sum_{i=1}^3 (N_{SP_i} \cdot W_{SP_i})_{drop} \\ + \sum_{i=1}^3 (N_{T_i} \cdot W_{T_i})_{drop} + \sum_{i=1}^3 (N_{ST_i} \cdot W_{TP_i})_{drop} \quad (7.2.42)$$

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

$$\{X\}_{T0} \rightarrow \{X\}_{50} \quad (7.2.43)$$

where the suffix T0 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

$$\{X\}_2 = \{X\}_1 + \int_{X_1}^{X_2} \{\dot{X}\} dM = \{X\}_1 + \sum_i \{\Delta X\}_i \quad (7.2.44)$$

where $\{\Delta X\}_i$ is the state change in accelerating from M to $M + \Delta M$.

Given $\{X\}_i = \{V, h, \gamma, W, R, t\}_i$, T_i and D_i , then the velocity change is

$$V = a_s \Delta M \quad (7.2.45)$$

where a_s is the speed of sound at the acceleration altitude and

$$\dot{V}_i = g_0 \left[\frac{T-D}{W} \right]_i \quad (7.2.46)$$

The approximate time to accelerate from M_i to $M_i + \Delta M$ is

$$\Delta t' = \Delta V / \dot{V}_i \quad (7.2.47)$$

The corresponding approximate weight change is

$$\Delta W' = \dot{W}_i \Delta t' \quad (7.2.48)$$

and

$$W_{i+1} = W_i - \Delta W' \quad (7.2.49)$$

and to the first order

$$\dot{V}_{i+1} = g_0 \left[\frac{T-D}{W} \right]_{i+1} \quad (7.2.50)$$

\dot{W}_{i+1} can be obtained at the new Mach number. The mean acceleration is now

$$\dot{\bar{V}} = \frac{1}{2}(\dot{V}_i + \dot{V}_{i+1}) \quad (7.2.51)$$

The time to accelerate is

$$\Delta t = \Delta V / \dot{\bar{V}} \quad (7.2.52)$$

which gives a weight change of

$$\Delta W = \frac{1}{2}(\dot{W}_i + \dot{W}_{i+1}) \Delta t \quad (7.2.53)$$

and a range increment

$$\Delta R = \frac{1}{2}(V_i + V_{i+1}) \Delta t \quad (7.2.54)$$

The state incremental vector $\{\Delta X\}_i$ is therefore given by

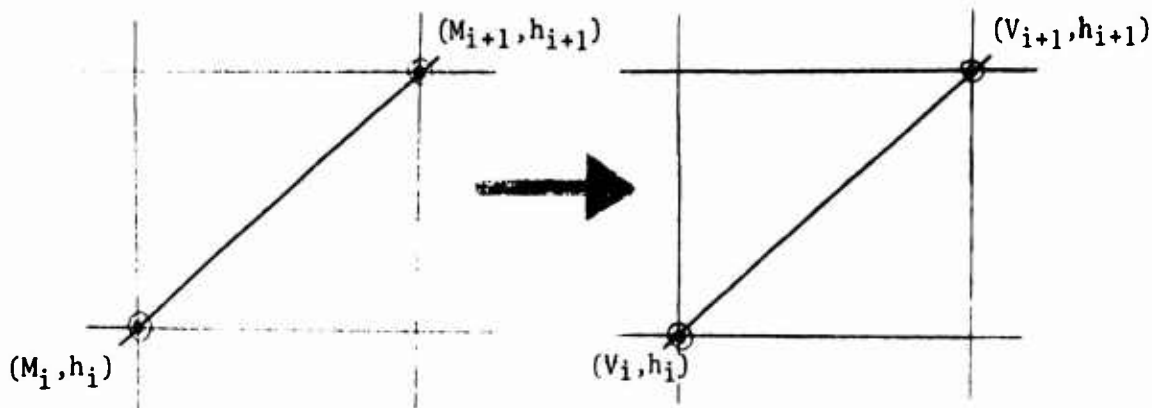
$$\begin{bmatrix} \Delta V \\ \Delta h \\ \Delta \gamma \\ \Delta R \\ \Delta W \\ \Delta t \end{bmatrix} = \begin{bmatrix} a_s \Delta M \\ 0 \\ 0 \\ \frac{1}{2}(\dot{W}_i + \dot{W}_{i+1}) \Delta t \\ \frac{1}{2}(V_i + V_{i+1}) \Delta t \\ \frac{2\Delta V}{(\dot{V}_i + \dot{V}_{i+1})} \end{bmatrix} \quad (7.2.55)$$

7.2.5.2 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

$$h_{i+1} > h_i \quad (7.2.56)$$

The typical arc for a climb path is shown below. The Mach-altitude plane



can be transformed into the velocity-altitude plane as follows:

$$V = V(h, M) \quad (7.2.57)$$

so that

$$\Delta V = \frac{\partial V}{\partial h} \cdot \delta h + \frac{\partial V}{\partial M} \delta M \quad (7.2.58)$$

or

$$\frac{dV}{dh} = \frac{\partial V}{\partial h} + \frac{\partial V}{\partial M} \cdot \frac{dM}{dh} \quad (7.2.59)$$

$$\frac{\partial V}{\partial h} + a \frac{dM}{dh} \quad (7.2.60)$$

where a is the local speed of sound

$$V = aM \quad (7.2.61)$$

Now $\partial V/\partial h$ is the change in velocity with altitude at constant Mach number, and from Equation (7.2.50) with M constant

$$\begin{aligned} \frac{\partial V}{\partial h} &= M \frac{\partial a}{\partial h} \\ &= M \frac{\partial a}{\partial T_R} \cdot \frac{dT_R}{dh} \end{aligned} \quad (7.2.62)$$

where T_R is temperature ratio, T/T_{SL} , so that

$$\frac{\partial V}{\partial h} = \left(\frac{V}{a}\right) \cdot \frac{\partial a}{\partial T_R} \cdot \frac{dT_R}{dh} \quad (7.2.64)$$

and from the atmospheric model

$$a = 1116.45 (T_R)^{1/2} \quad (7.2.65)$$

$$\frac{\partial a}{\partial T_R} = \frac{a}{2T_R} \quad (7.2.66)$$

Substituting into Equation (7.2.64)

$$\frac{\partial V}{\partial h} = \frac{V}{2T_R} \cdot \frac{dT_R}{dh} \quad (7.2.67)$$

Substituting Equation (7.2.67) into Equation (7.2.60)

$$\frac{dV}{dh} = \frac{V}{2T_R} \cdot \frac{dT_R}{dh} + a \frac{dM}{dh} \quad (7.2.68)$$

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

$$\frac{dh}{dt} = RC \quad (7.2.69)$$

or

$$\frac{dh}{RC} = dt \quad (7.2.70)$$

Assuming rate of climb varies linearly with altitude in the elemental arc

$$RC = a + bh \quad (7.2.71)$$

Substituting into Equation (7.2.70) and integrating

$$\int_{h_1}^{h_2} \frac{dh}{(a+bh)} = \int_{t_1}^{t_2} dt \quad (7.2.72)$$

or

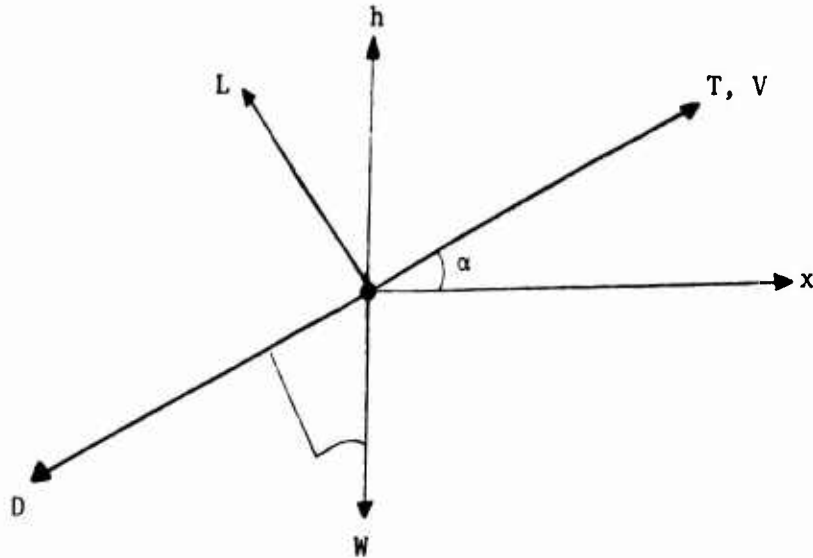
$$t_{i+1} - t_i = \left[\frac{h_{i+1} - h_i}{RC_{i+1} - RC_i} \right] \log \left(\frac{RC_{i+1}}{RC_i} \right) \quad (7.2.73)$$

where

$$a = RC_i \quad (7.2.74)$$

$$b = \frac{RC_{i+1} - RC_i}{h_{i+1} - h_i} \quad (7.2.75)$$

The vehicle rate of climb is computed under the assumption that thrust is aligned along the velocity vector as shown below.



Now

$$m\dot{V} = (T - D) - W \sin\gamma \quad (7.2.76)$$

but

$$m\dot{V} = m \frac{dV}{dh} \cdot \frac{dh}{dt} = \frac{W}{g} \frac{dV}{dh} \cdot V \sin\gamma \quad (7.2.77)$$

Combining Equations (7.2.76) and (7.2.77)

$$\sin\gamma = \frac{T-D}{W \left[\frac{V}{g} \frac{dV}{dh} + 1.0 \right]} \quad (7.2.78)$$

and

$$\cos\gamma = \sqrt{1 - \sin^2\gamma} \quad (7.2.79)$$

so that

$$RC = V \sin\gamma = \left[\frac{(T-D)V}{W} \right] \div \left[\frac{V}{g} \frac{dV}{dh} + 1.0 \right] \quad (7.2.80)$$

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

$$\begin{aligned}
 \Delta V &= V_{i+1} - V_i \\
 \Delta h &= h_{i+1} - h_i \\
 \Delta \gamma &= \gamma_{i+1} - \gamma_i \\
 \Delta R &= \frac{1}{2}[V_{i+1} \cos \gamma_{i+1} + V_i \cos \gamma_i] \\
 \Delta W &= \frac{1}{2}[\dot{W}_{i+1} + \dot{W}_i] \Delta t \\
 \Delta t &= \frac{h_{i+1} - h_i}{RC_{i+1} - RC_i} \log \left[\frac{RC_{i+1}}{RC_i} \right]
 \end{aligned} \tag{7.2.81}$$

7.2.5.3 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

$$\frac{L}{D} \approx \frac{W}{T} \tag{7.2.86}$$

and

$$\Delta R = \frac{V}{SFC} \left(\frac{L}{D}\right) \frac{\Delta W}{W} \quad (7.2.87)$$

On integrating

$$R_{i+1} - R_i = \frac{V}{SFC} \left(\frac{L}{D}\right) \log\left(\frac{W_{i+1}}{W_i}\right) = (RF) \log\left(\frac{W_{i+1}}{W_i}\right) \quad (7.2.88)$$

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

$$W_{i+1} = W_i e^{\left[\frac{R_{i+1}-R_i}{RF}\right]} = W_i e^{\left[\frac{(R_{i+1}-R_i) \cdot (SFC)}{V(L/D)}\right]} \quad (7.2.89)$$

Alternatively, the program can be used with a time increment Δt by using the relationship

$$W_{i+1} = W_i e^{\left[\frac{\Delta t \cdot SFC}{(L/D)}\right]} \quad (7.2.90)$$

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,

$$\Delta R_{\text{cruise}} = \sum_i \Delta R_i$$

$$\Delta T_{\text{cruise}} = \sum_i \Delta T_i$$

or

$$\Delta W_{\text{cruise}} = \sum_i \Delta W_i$$

In all cases the total state increments are summed in the manner

$$\{\Delta X\}_{\text{cruise}} = \sum_i \{\Delta X\}_i \quad (7.2.91)$$

A mean range factor is used in all cruise calculations. The mean range factor, (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.

7.2.5.4 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.

7.2.5.5 Level Flight Acceleration

The approximate time to accelerate from M_i to M_{i+1} in level flight is

$$\Delta t'_i = a_s(M_{i+1} - M_i) / \dot{V}_i \quad (7.2.92)$$

with a corresponding weight change

$$\Delta W' = \dot{W}_i \Delta t' \quad (7.2.93)$$

so that

$$W'_{i+1} = W_i - \dot{W}_i \Delta t' \quad (7.2.94)$$

Therefore, to the first order

$$\dot{V}_{i+1} = g_0 \left[\frac{T-D}{W} \right] \quad (7.2.95)$$

The fuel flow at this point, \dot{W}_{i+1} , can be obtained from the vehicle aerodynamic and propulsion representation.

$$\dot{V}_i = \frac{1}{2} (\dot{V}_i + \dot{V}_{i+1}) \quad (7.2.96)$$

and an improved estimate of the time to accelerate from M_i to M_{i+1} is

$$\Delta t'_i = a_s(M_{i+1} - M_i) / \dot{V}_i \quad (7.2.97)$$

This gives an improved estimate of the weight change

$$\Delta W_i = \frac{1}{2}(\dot{W}_i + \dot{W}_{i+1}) \Delta t_i \quad (7.2.98)$$

and the corresponding range change

$$\Delta R_i = \frac{1}{2}(V_i + V_{i+1}) \Delta t_i \quad (7.2.99)$$

Summarizing, the level acceleration state increment is

$$\begin{aligned} \Delta h &= 0.0 \\ \Delta V &= a_s(M_{i+1} - M_i) \\ \Delta Y &= 0.0 \\ \delta W &= \frac{1}{2}(\dot{W}_i + \dot{W}_{i+1}) \Delta t_i \\ \delta R &= \frac{a_s}{2}(M_{i+1} + M_i) \Delta t_i \\ \Delta t_i &= \frac{a_s}{\partial g_0} (M_{i+1} - M_i) / \left[\left(\frac{T-D}{W}\right)_i + \left(\frac{T-D}{W}\right)_{i+1} \right] \end{aligned} \quad (7.2.100)$$

7.2.5.6 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 must rapidly build 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

$$E_1 = V_1^2/\partial g + h_1 \quad (7.2.101)$$

and find specific energy

$$E_2 = V_2^2/\partial g + h_2 \quad (7.2.102)$$

and divide the energy change $(E_2 - E_1)$ into N equal increments

2. Search at each incremental energy level

$$E_i = E_1 + i \cdot \Delta E \quad i = 1, 2, \dots, N \quad (7.2.103)$$

to find the point of maximum specific energy derivative, (M_i, h_i) where

$$\dot{E}_i = (T_i - D_i)V_i/W_i \quad (7.2.104)$$

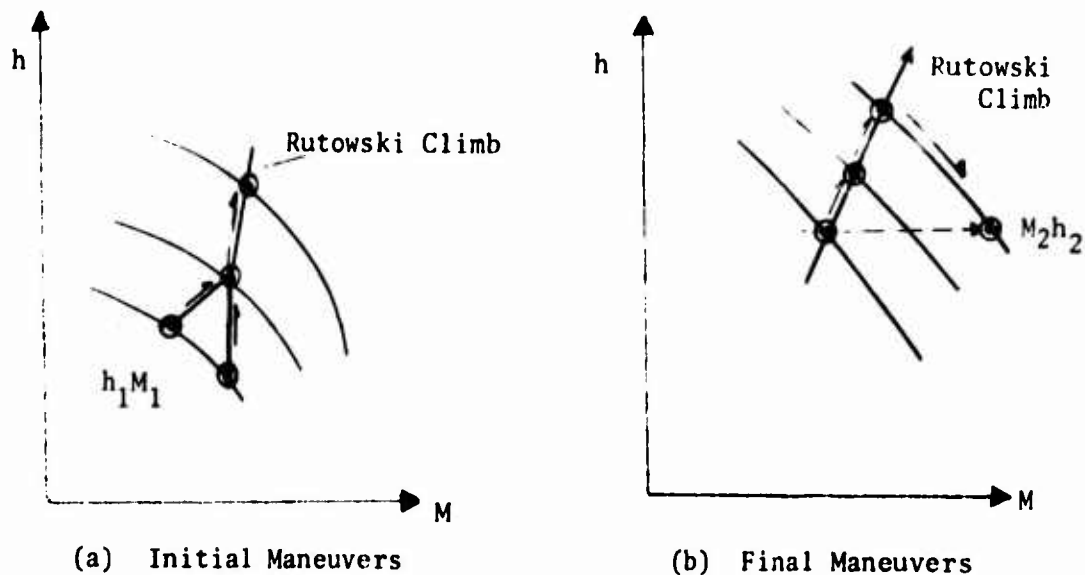
The \dot{E} calculation is carried out for specified weight and load factor.

3. Fly a sequence of linear Mach-altitude flight increments joining the point $(M_i h_i)$ 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

$$h = h_1 + \Delta h \quad (7.2.105)$$

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 $M_2 h_2$. 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 $M_2 h_2$ occurs. These terminal maneuvers are sketched below.



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 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 E is carried out at the constant altitudes

$$h = h_1 + i \cdot \Delta h \quad i = 1, 2, \dots, N \quad (7.2.106)$$

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 E is carried out at the constant altitudes

$$M = M_1 + i \cdot \Delta M \quad i = 1, 2, \dots, N \quad (7.2.107)$$

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 Cl_{max} and maximum dynamic pressure constraints of the Rutowski path. In addition, the condition

$$\Delta E_{i+1} \geq \Delta E_i \quad (7.2.108)$$

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

$$\phi = \text{Maximum } [E/M] = \text{Max} \left[\frac{dE/dt}{dM/dt} \right] = \text{Max} \left[\frac{dE}{dM} \right] \quad (7.2.109)$$

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

$$\phi = \text{Maximum [L/D]} \quad (7.2.110)$$

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:

$$E = h + V^2/2g \quad (7.2.111)$$

and

$$m \left(\frac{dv}{dt} \right) = T - D - W \sin \gamma \quad (7.2.112)$$

Now

$$R = \int dR = \int \frac{dR}{dE} dE = \int \frac{dR}{dt} \cdot \frac{dt}{dE} \cdot dE \quad (7.2.113)$$

There from Equation (7.2.111)

$$\begin{aligned} R &= \int \frac{V \cos \gamma \cdot dE}{\frac{dh}{dt} + \frac{V}{g} \cdot \frac{dV}{dt}} \\ &= \frac{\cos \gamma \cdot dE}{\sin \gamma + \frac{1}{g} \frac{dV}{dt}} \end{aligned} \quad (7.2.114)$$

But from Equation (7.2.112)

$$\frac{1}{g} \cdot \left(\frac{dV}{dt} \right) = \frac{T-D}{W} - \sin \gamma \quad (7.2.115)$$

So that

$$R = \int \frac{W \cos \gamma \cdot dE}{T-D} \quad (7.2.116)$$

Assuming that range biased ascents occur at small flight path angles with $L \approx W$, Equation (7.2.116) becomes

$$R \approx \int \frac{L}{T-D} dE \quad (7.2.117)$$

Therefore, an energy-like approximation for a range biased ascent 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.

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

$$h_i = h_1 + i \cdot \Delta h \quad i = 1, 2, \dots, N \quad (7.2.118)$$

At each altitude the Mach number for maximum rate of climb using the angle of attack for $C_{L_{MAX}}$ is found

$$M_i = M_{i_{MAX RC}} \quad (7.2.119)$$

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 ; $j = J_1, J_2, \dots, J_N$ equal to the total range over the combined mission segments S_k ; $k = K_1, K_2, \dots, K_N$. That is

$$R = \sum_j R_j = \sum_k R_k \quad (7.2.120)$$

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

$$W_i = W_1 + i \cdot \Delta W \quad i = 1, 2, \dots, 5 \quad (7.2.121)$$

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,

$$W_f = a + bR + cR^2 \quad (7.2.122)$$

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

$$W_{FR} \geq W_{FA} \quad (7.2.123)$$

2. Distance flow is equal to maximum refuelling range
3. Minimum in-flight weight of the vehicle receiving fuel is reached where

$$W_{MIN} = W_{OWE} + W_{PL} + W_F \cdot k_F \quad (7.2.124)$$

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

$$W_{MAX} = W_{TO} \quad (7.2.125)$$

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 ,

$$\{X\}_j = [P]_{ij} \{X\}_i \quad (7.2.126)$$

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:

$$\phi = \text{Min } [M_1, M_2] \quad (7.2.127)$$

$$\phi = \text{Max } [M_1, M_2] \quad (7.2.128)$$

$$\phi = \text{Min } [h_1, h_2] \quad (7.2.129)$$

$$\phi = \text{Max } [h_1, h_2] \quad (7.2.130)$$

$$\phi = \text{Min } [W_1, W_2] \quad (7.2.131)$$

$$\phi = \text{Max } [W_1, W_2] \quad (7.2.132)$$

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

$$W_{i+1} = W_i - \Delta W \quad (7.2.133)$$

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

$$\Delta M = M_{i+1} - M_i \quad (7.2.134)$$

$$\Delta h = h_{i+1} - h_i \quad (7.2.135)$$

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*

$$\underset{h}{\text{Max}}[\text{RF}; M, W] \quad (7.2.36)$$

2. *Ceiling for a specified rate of climb at given Mach number and weight*

$$\underset{h}{\text{Max}}[\text{RC}; M, W] \quad (7.2.37)$$

3. *Mach number for maximum lift coefficient at given weight and altitude*

$$\underset{M}{\text{Max}}[C_L; W, h] \quad (7.2.38)$$

4. *Mach number for specified lift coefficient given weight and altitude*

$$\underset{M}{\text{Find}}[C_L; W, h] \quad (7.2.39)$$

5. *Maximum endurance Mach number given altitude, weight, and maximum lift coefficient*
- $$\underset{M}{\text{Min}}[\dot{W}; W, h, C_{L_{\text{max}}}] \quad (7.2.140)$$
6. *Maximum Mach number at given weight and altitude*
- $$\underset{M}{\text{Max}}[M; W, h] \quad (7.2.141)$$
- 7a. *Mach number for maximum rate of climb at given weight and altitude*
- $$\underset{M}{\text{Max}}[RC; W, h] \quad (7.2.142)$$
- 7b. *Mach number for maximum rate of climb per pound of fuel at given weight and altitude*
- $$\underset{M}{\text{Max}}[dh/dW; W, h] \quad (7.2.143)$$
8. *Approximate Mach number and altitude for maximum range factor given weight*
- $$\underset{(M,h)}{\text{Max}}[RF; W] \quad (7.2.144)$$
9. *Mach number for maximum range factor given altitude and weight*
- $$\underset{M}{\text{Max}}(RF; h, W] \quad (7.2.145)$$
10. *Various energy maneuverability parameters at specified load factor given Mach, altitude, and weight*
- The required lift coefficient
 - Specific excess power
- $$P_S = \dot{E} = (T-D)V/W$$
- Specific excess power divided by fuel flow
- $$P_S/\dot{W} = \dot{E}/\dot{W} = (T-D)V/(W \cdot \dot{W}) \quad (7.2.146)$$
- Specific excess power divided by fuel flow and multiplied by fuel remaining (ΔE capability) measure

$$P_S \frac{W_F}{W} = \frac{\dot{E} W_F}{\dot{W}} = \frac{dE}{dW} \cdot W_F \approx \Delta E \quad (7.2.147)$$

e. Specific energy

$$E_S = h + V^2/2g \quad (7.2.148)$$

f. Load factor at $P_S=0.0$

g. Steady state turn radius computed as follows:

$$C_L = C_L', \text{ for given load factor}$$

Now for given bank angle, B_A

$$W = qSC_L \cdot \cos(B_A) \quad (7.2.149)$$

and the centrifugal force is

$$L \sin(B_A) = \frac{WV^2}{Rg} = \frac{L \cos(B_A) V^2}{Rg} \quad (7.2.150)$$

$$\therefore R = \frac{V^2}{g \tan \cdot B_A} \quad (7.2.151)$$

but from Equation (7.2.149)

$$\cos B_A = \frac{W}{qSC_L} \quad (7.2.152)$$

$$\therefore \tan B_A = \sqrt{\left(\frac{qSC_L}{W}\right)^2 - 1.0} \quad (7.2.153)$$

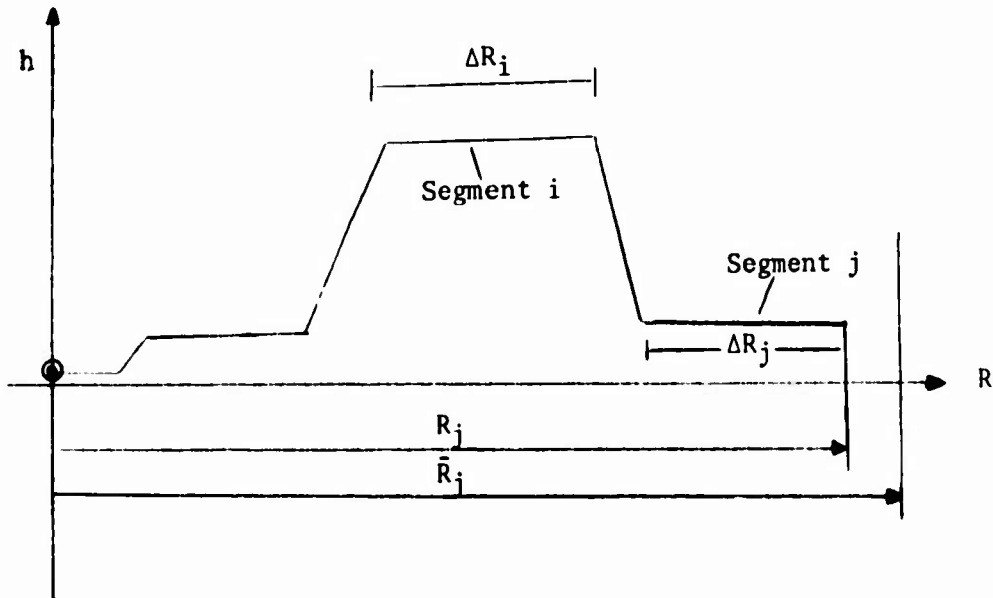
Substituting Equation (7.2.153) and (7.2.151)

$$R = \frac{V^2}{g} \sqrt{\frac{1.0}{\left(\frac{qSC_L}{W}\right)^2 - 1.0}} \quad (7.2.154)$$

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 option 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.

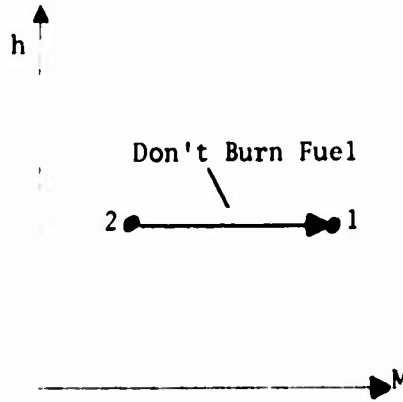
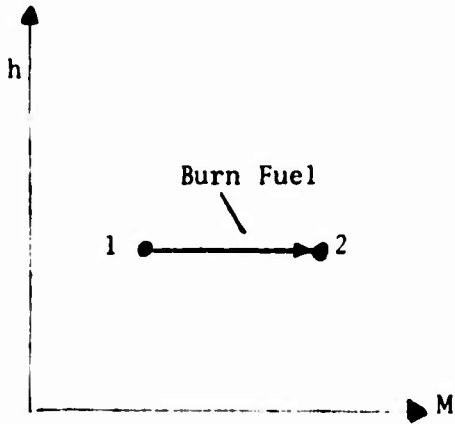
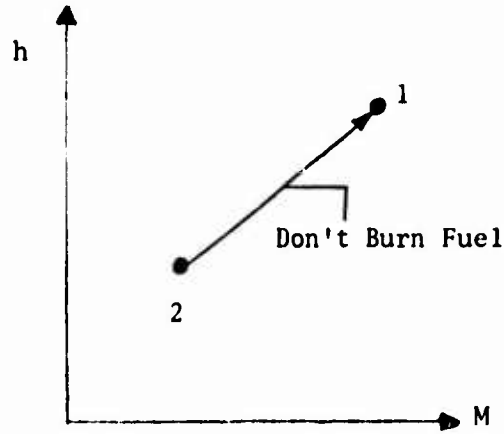
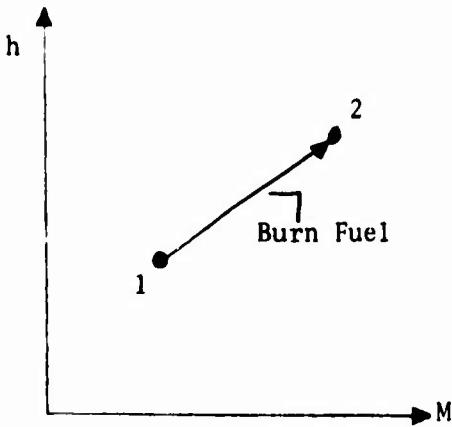
Fuel burning decisions are made according to Mil-C rules while going from *condition 1 to 2*. Thus, fuel is burned if

$$h_2 > h_1$$

or if

$$M_2 > M_1 \quad \text{and} \quad h_2 = h_1$$

This behavior is illustrated below.



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

$$W_{i+1} = W_i - \Delta W \quad (7.2.155)$$

$$T_{i+1} = T_i + \Delta T \quad (7.2.156)$$

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.

$$L = \bar{n} \cdot W \quad (7.2.157)$$

where \bar{n} is the load factor. The centrifugal force is

$$F_R = \sqrt{L^2 - W^2} \quad (7.2.158)$$

and the turn radius is

$$R = \frac{WV^2}{g\sqrt{L^2 - W^2}} \quad (7.2.159)$$

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, $T_{\max j}$; $j = 1, 2, 3$. These tables are referenced as follows:

$T_{\max 1}$ = maximum dry thrust

$T_{\max 2}$ = maximum wet thrust

$T_{\max 3}$ = maximum power

Throttling may only be used for $T_{\max 1}$ and $T_{\max 2}$. In using the various mission segment options an appropriate choice of thrust must be made. The options are

1. $T = D$
2. $T = \text{maximum dry}$
3. $T = \text{maximum wet}$
4. $T = \text{maximum power}$
5. $T = \text{thrust for given power setting, dry.}$

7.2.8 Flight Envelope Calculations

Several gross flight envelope calculations may be performed. All flight envelope computations are subject to the conditions

$$C_L \leq C_{L\text{lim}}, \text{ lift coefficient limit}$$

$$M \leq M_{\text{lim}}, \text{ Mach number limit}$$

$$q < q_{\text{lim}}, \text{ dynamic pressure limit.}$$

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

$$P_1 = P_1(M_1, 0.0) \quad (7.2.160)$$

to

$$P_2 = P_2(M_2, h_{\text{MAX RF}}) \quad (7.2.161)$$

where

$$h_{\text{MAX RF}} = \text{altitude for best range factor at } M_2$$

Alternatively, P_2 may be selected as

$$P_2 = P_2(M_{\text{MAX RF}}, h_{\text{MAX RF}}) \quad (7.2.162)$$

The calculation is performed in ten equal altitude increments from P_1 to P_2 . Climb paths are generated for N distinct weights

$$W_i = W_0 + i \cdot \Delta W; \quad i = 0, 1, \dots, N-1 \quad (7.2.163)$$

7.2.8.2 Endurance versus Weight at Various Altitudes

The endurance is calculated at a given altitude for the weights $W_i = W_0 + i \cdot \Delta W$; $i = 0, 1, \dots, N-1$. Mach number selected is for best endurance.

The calculation may be repeated for any number of altitudes, $h = h_0 + i \cdot \Delta h$; $i = 0, 1, \dots$

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

$$M_i = M_0 + i \cdot \Delta M; \quad i = 0, 1, 2, \dots \quad (7.2.164)$$

$$h_j = h_0 + j \cdot \Delta h; \quad j = 0, 1, 2, \dots \quad (7.2.165)$$

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_{ij}^k , 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

$$\dot{E}(M, h) = (T - D)V/W \quad (7.2.166)$$

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

$$\dot{E}/\dot{m} = \frac{dE/dt}{dm/dt} = \frac{dE}{dm} \quad (7.2.167)$$

The map illustrates an aircraft's ability to convert fuel into energy at specified flight conditions.

The point calculation employed is

$$\dot{E}/\dot{m} = (T - D)V/(W\dot{m}) \quad (7.2.168)$$

where 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 lg 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 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.

$$R_F = \left(\frac{V}{SFC}\right) \left(\frac{L}{D}\right) \quad (7.2.169)$$

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 lg 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

$$E = h + V^2/2g \quad (7.2.170)$$

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

$$R \approx \int \frac{L}{T-D} dE \quad (7.2.171)$$

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

$$R = w \left(\frac{V^2}{g} \right) \left[\frac{1.0}{(qSC_L)^2 - W^2} \right]^{1/2} \quad (7.2.172)$$

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.

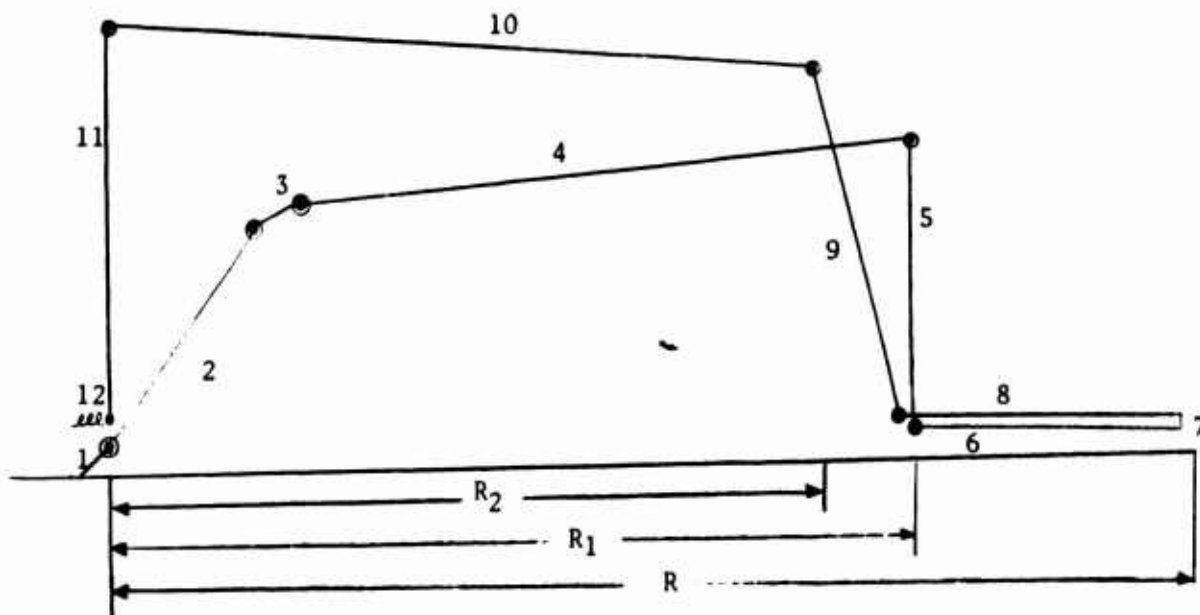
7.2.9.12 Time to Turn (INDMAP=12)

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

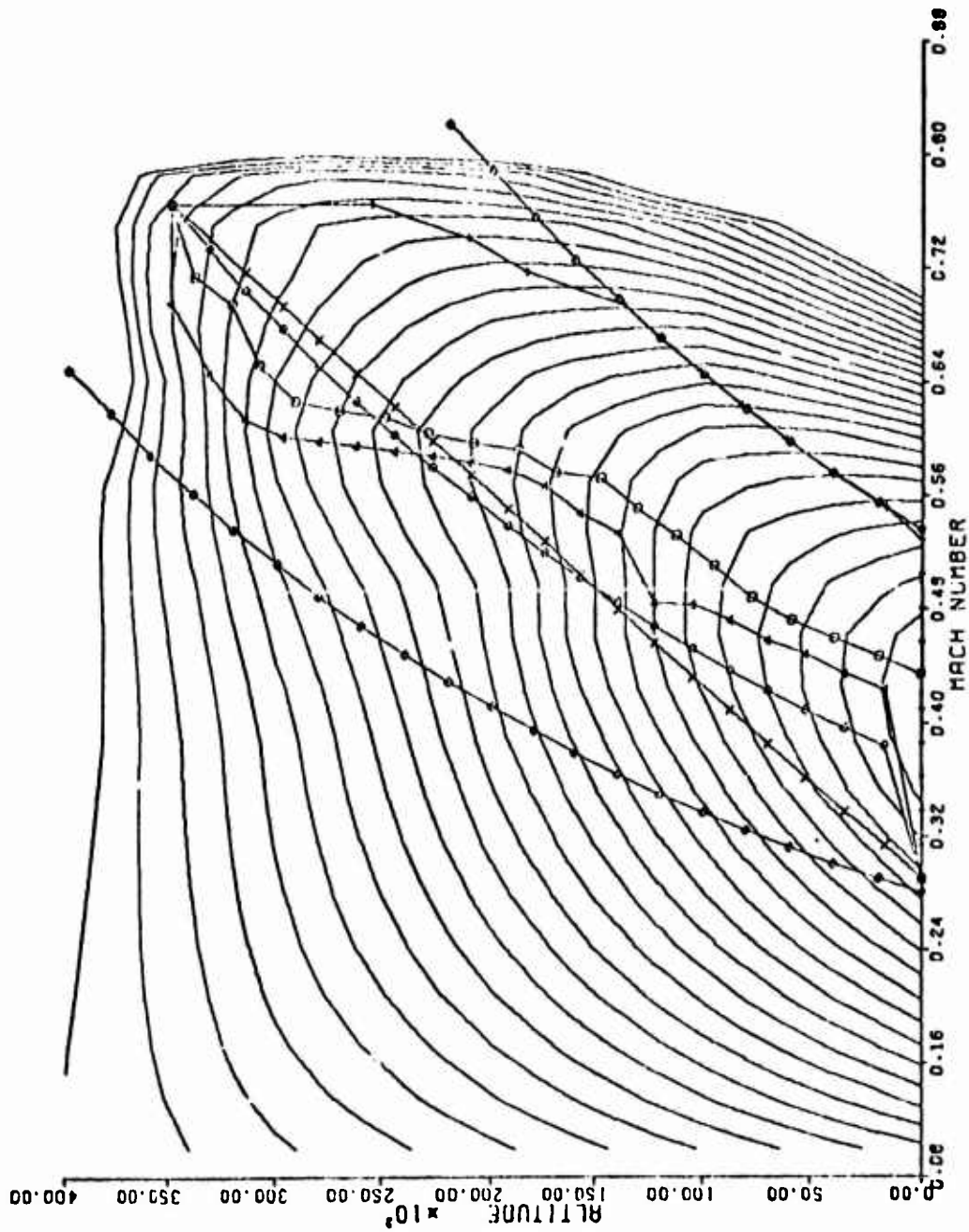


FIGURE 7.2-2. TYPICAL FLIGHT POINT PERFORMANCE MAPS
(SPECIFIC ENERGY AT VARIOUS LOAD FACTORS, WEIGHTS, AND THRUSTS)

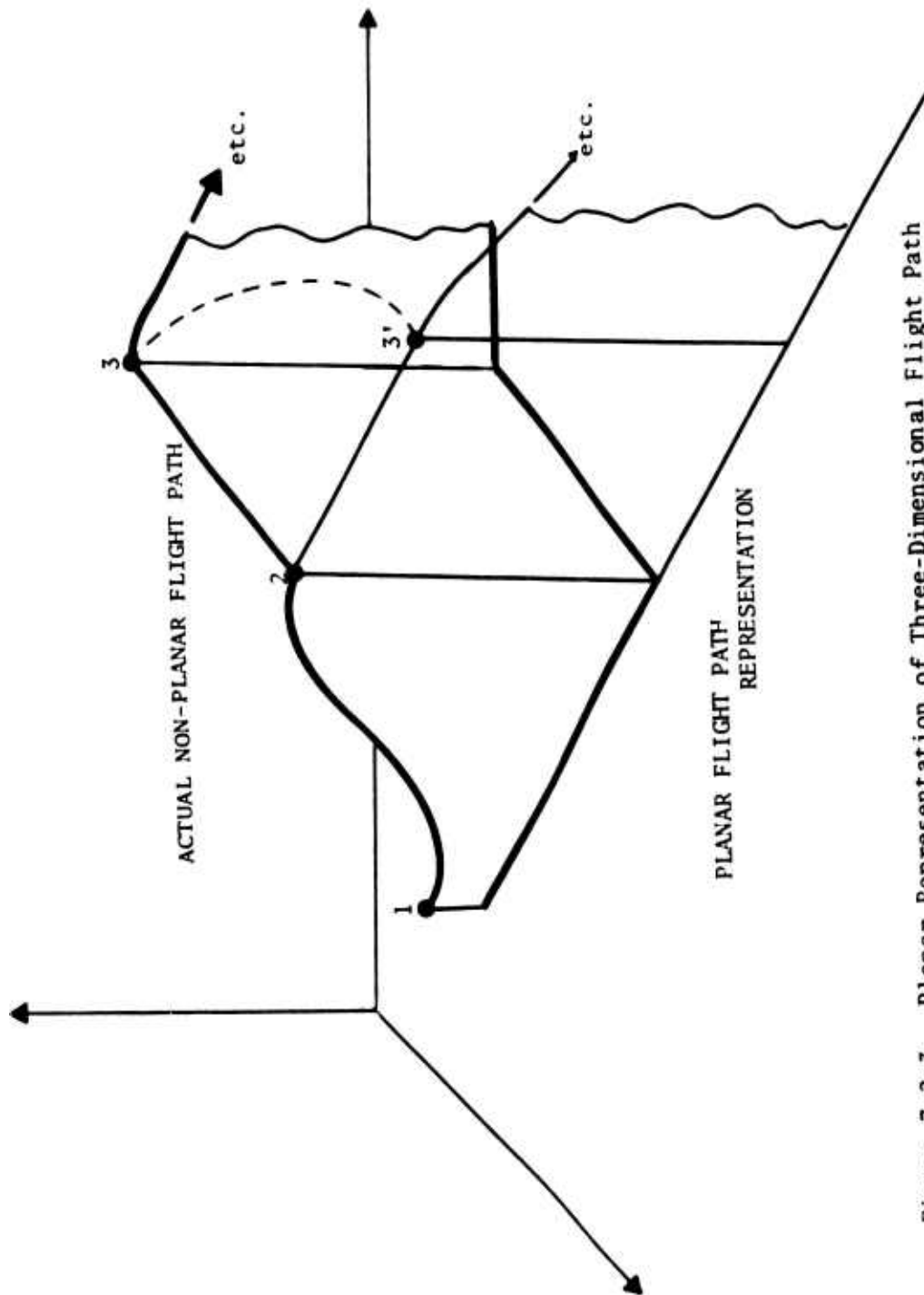


Figure 7.2-3. Planar Representation of Three-Dimensional Flight Path

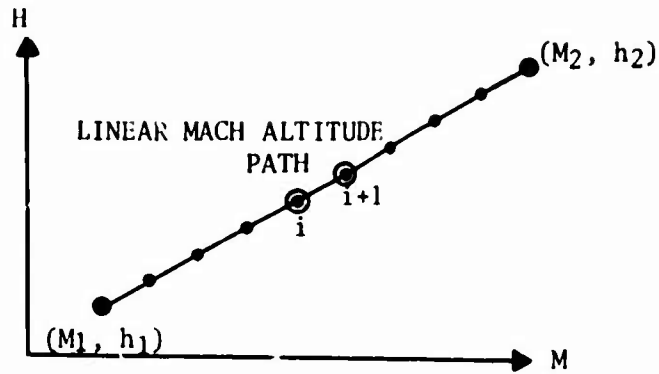


Figure 7.2-4. Linear Mach Altitude Mach Segment

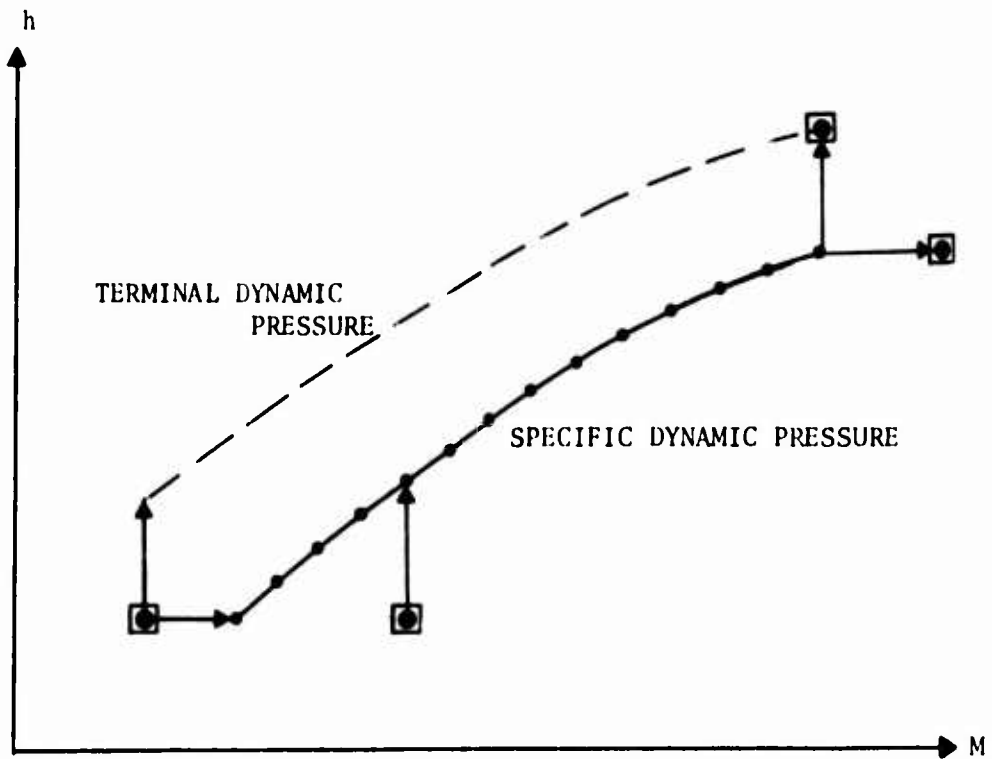


Figure 7.2-5. Constant Dynamic Pressure Segment

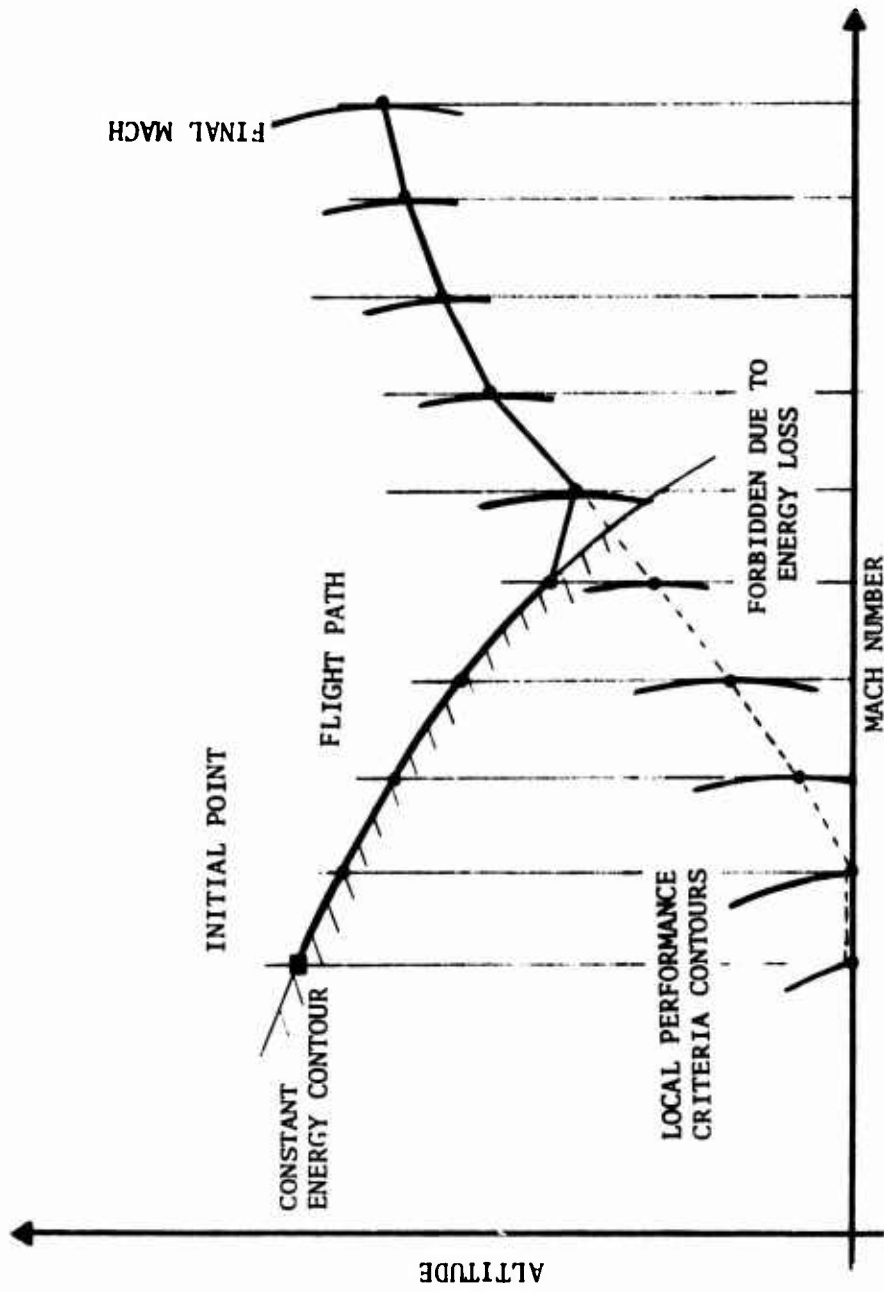


FIGURE 7.2-6. TYPICAL NSEG II ACCELERATION PATH

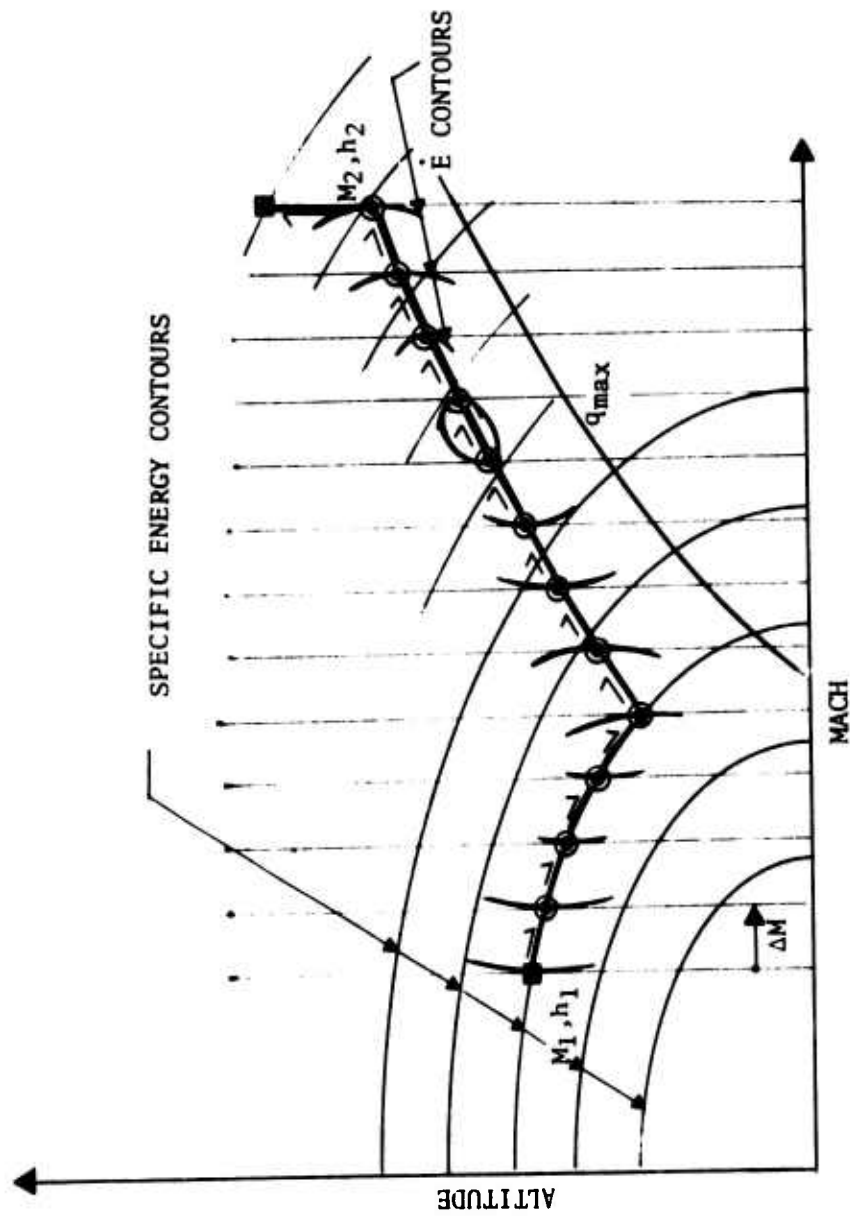


Figure 7.2-7. Energy Constraint Imposed on Maximum Acceleration Path

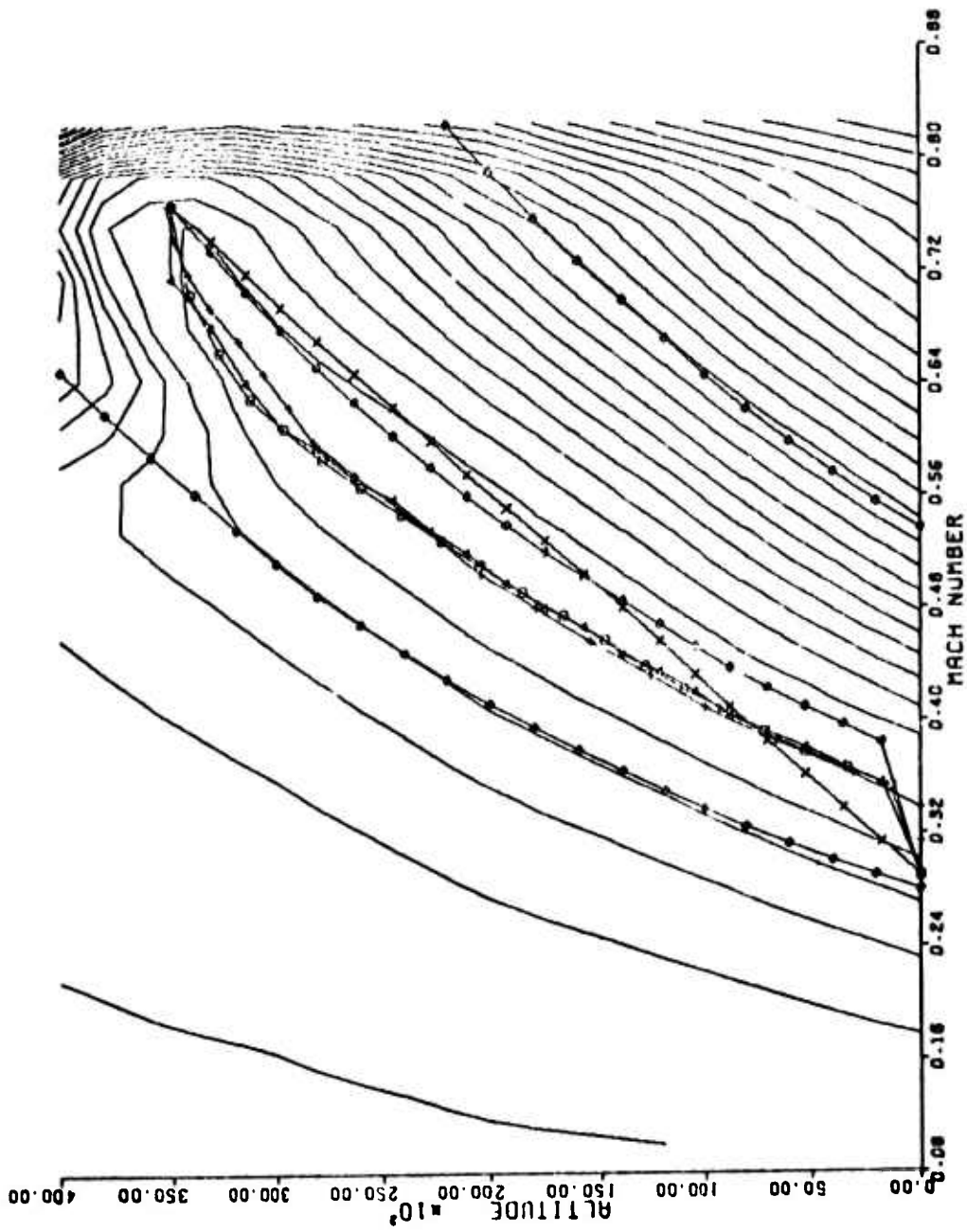


FIGURE 7.2-8. LIFT/DRAG MAP

MAX THRUST
GLEVEL = 1.0
Wt = 575000 pounds
EGGT MAP

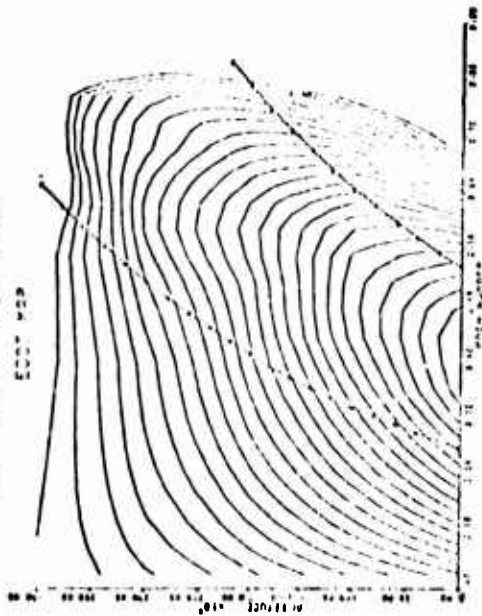


Figure 1a.

75% MAX THRUST
GLEVEL = 1.0
Wt = 575000 pounds
EGGT MAP

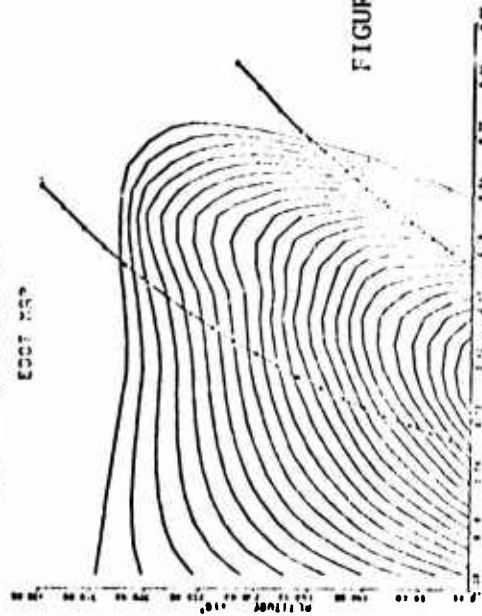


Figure 1c.

MAX THRUST
GLEVEL = 1.5
Wt = 575000 pounds
EGGT MAP

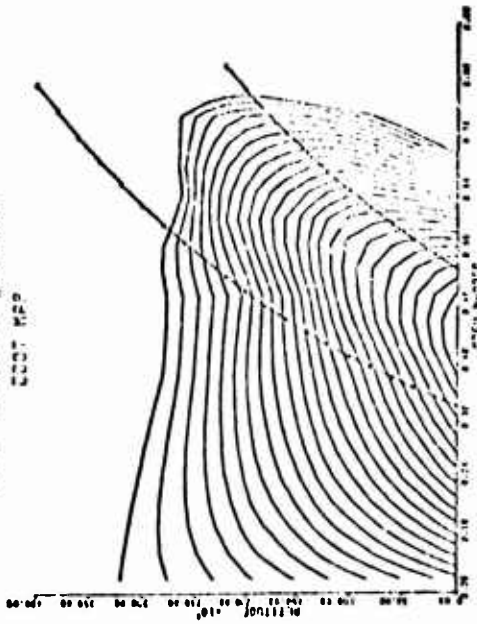


Figure 1b.

MAX THRUST
GLEVEL = 1.0
Wt = 660000 pounds
EGGT MAP

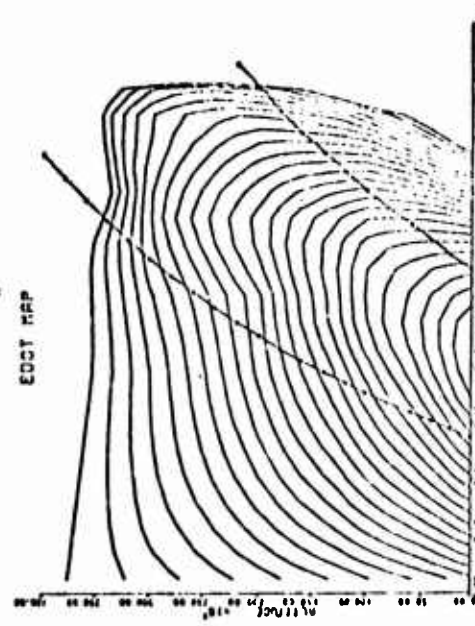


Figure 1d.



FIGURE 7.2-9. E MAPS

FIGURE 7.2-10

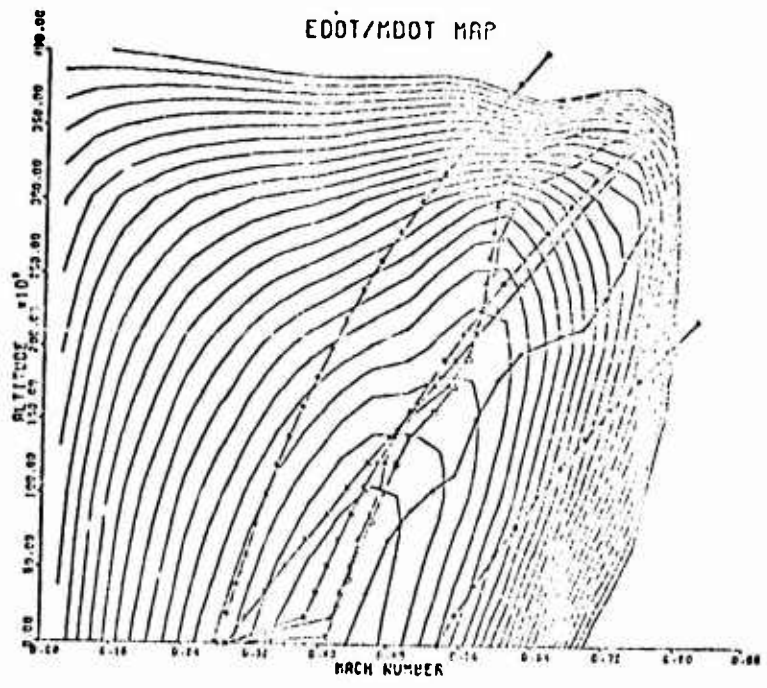


FIGURE 7.2-11

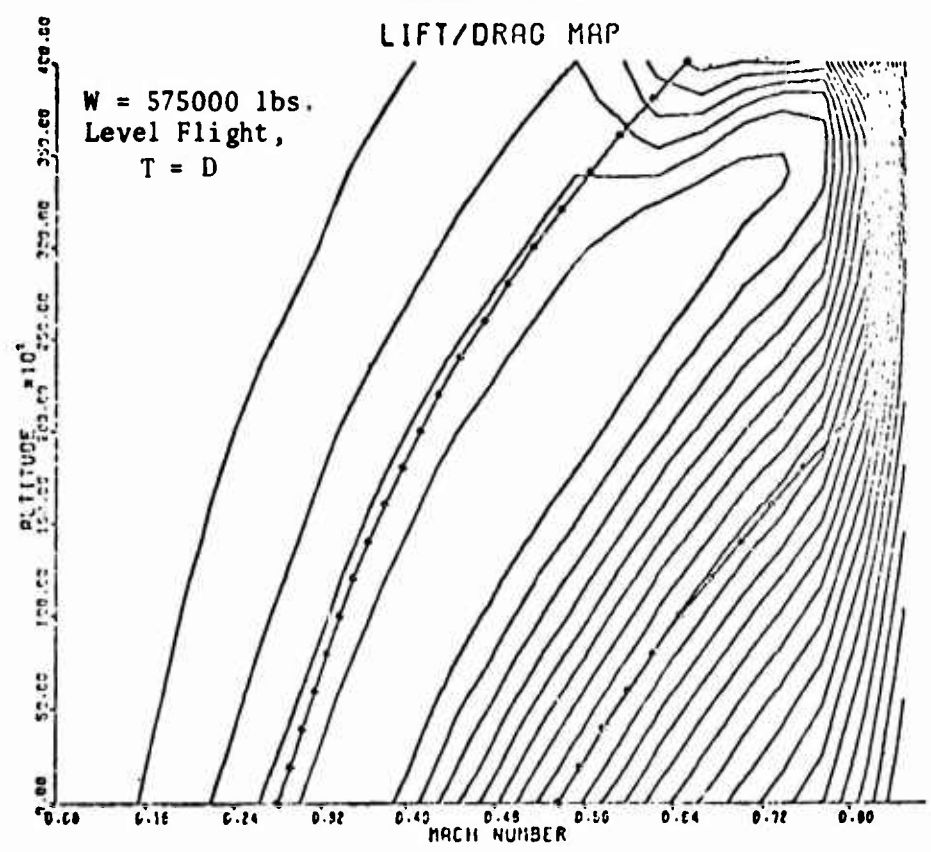


FIGURE 7.2-12.
RANGE FACTOR MAP

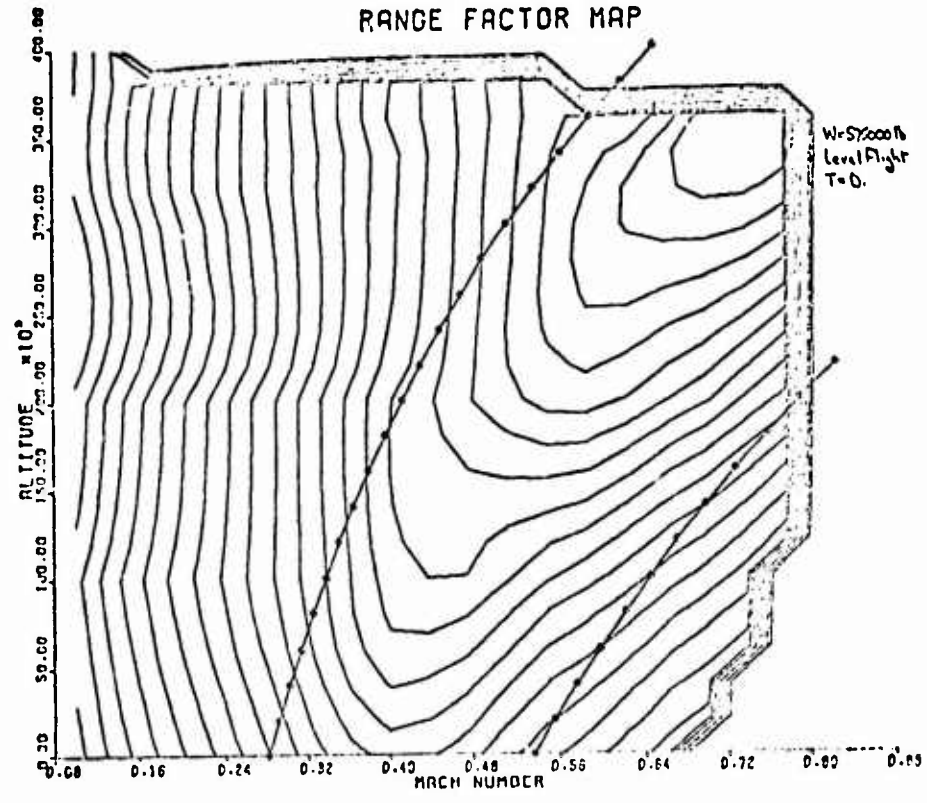


FIGURE 7.2-13
THRUST MAP

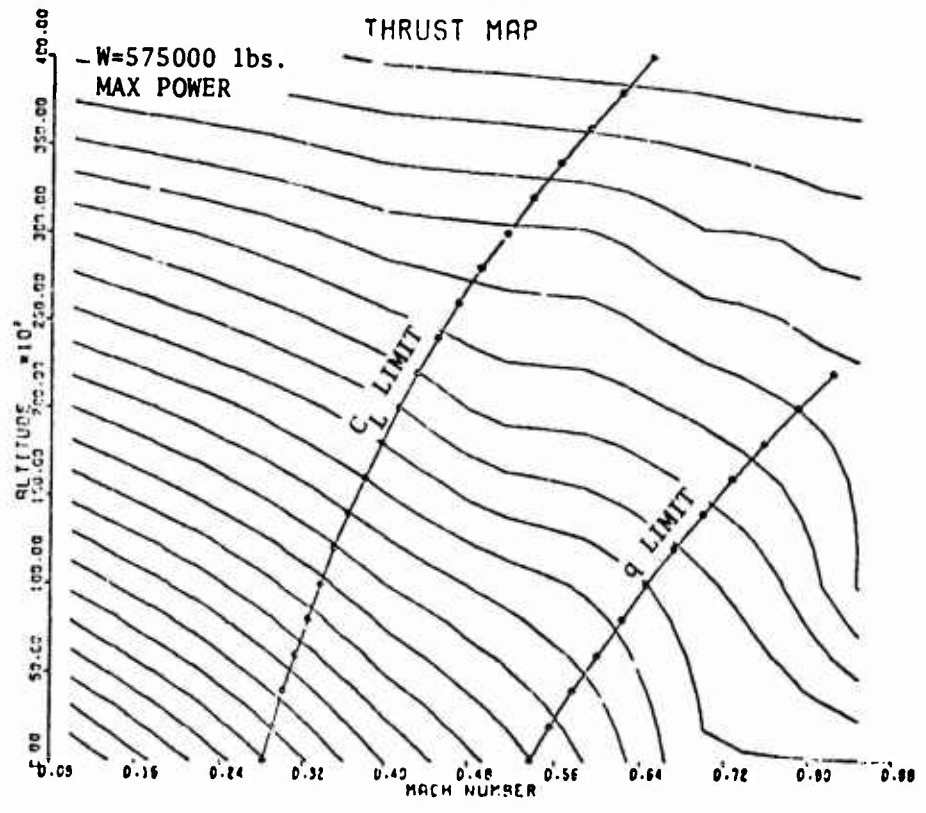


FIGURE 7.2-15. SPECIFIC FUEL CONSUMPTION MAP

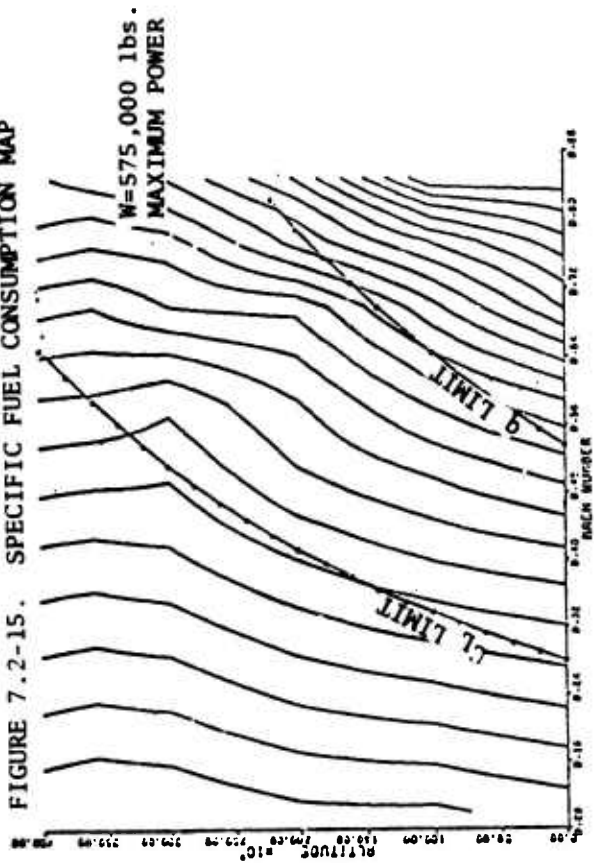


FIGURE 7.2-16. FUEL FLOW RATE MAP

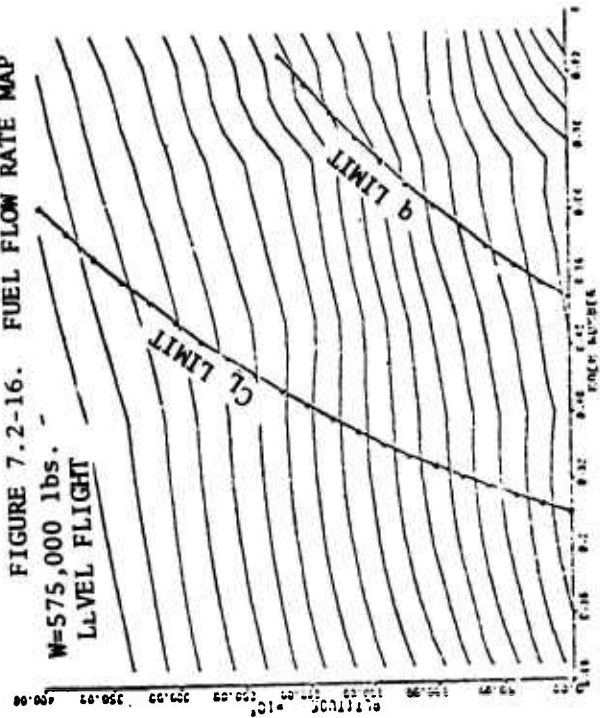


FIGURE 7.2-14. DRAG MAP

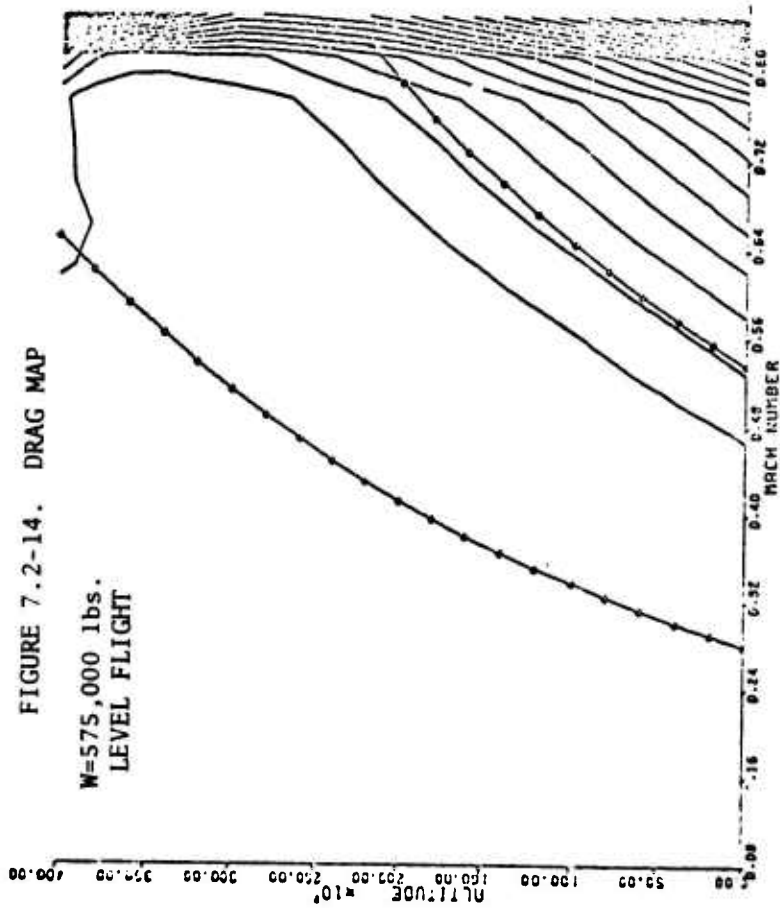


FIGURE 7.2-17. SPECIFIC ENERGY MAP

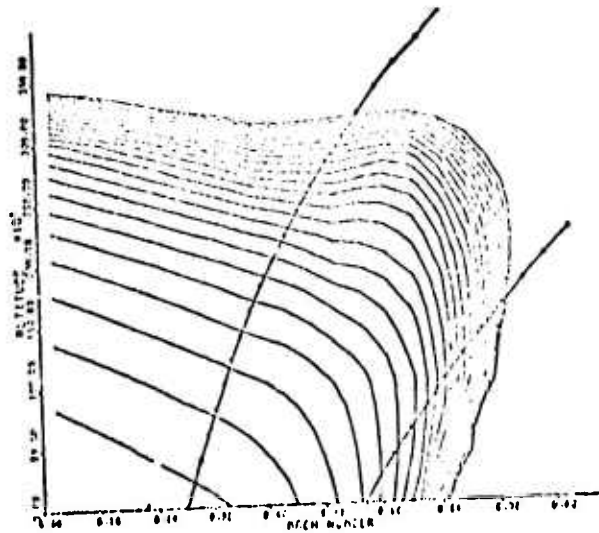


FIGURE 7.2-18. LIFT/(THRUST-DRAG)

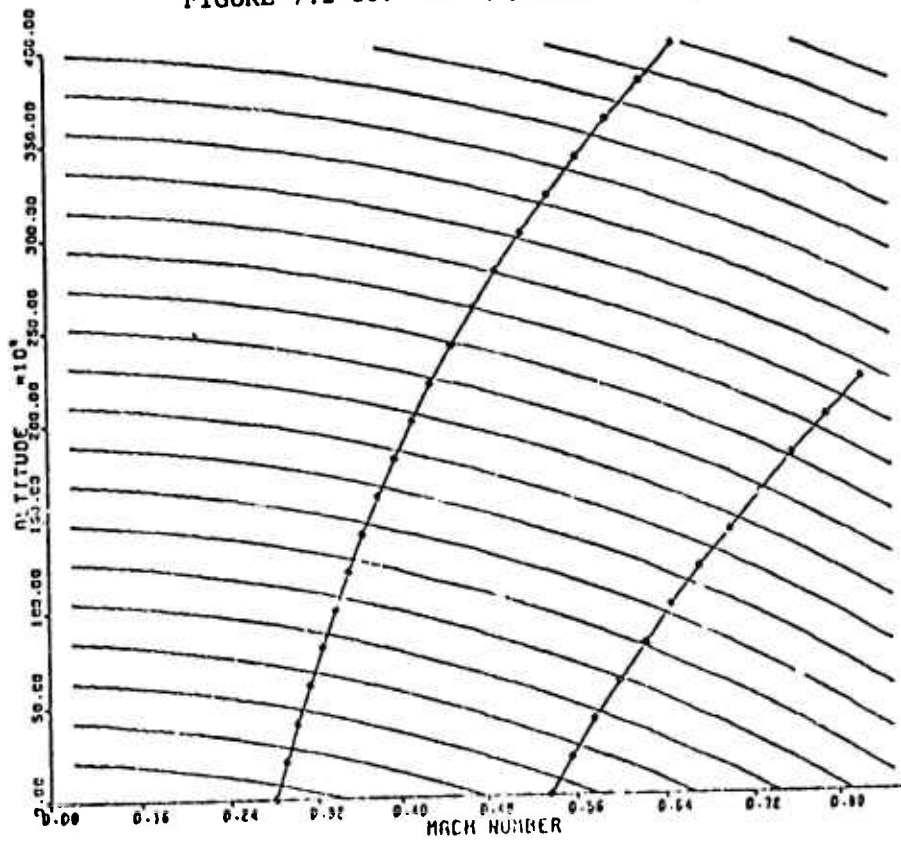


FIGURE 7.2-19 RADIUS OF TURN MAP

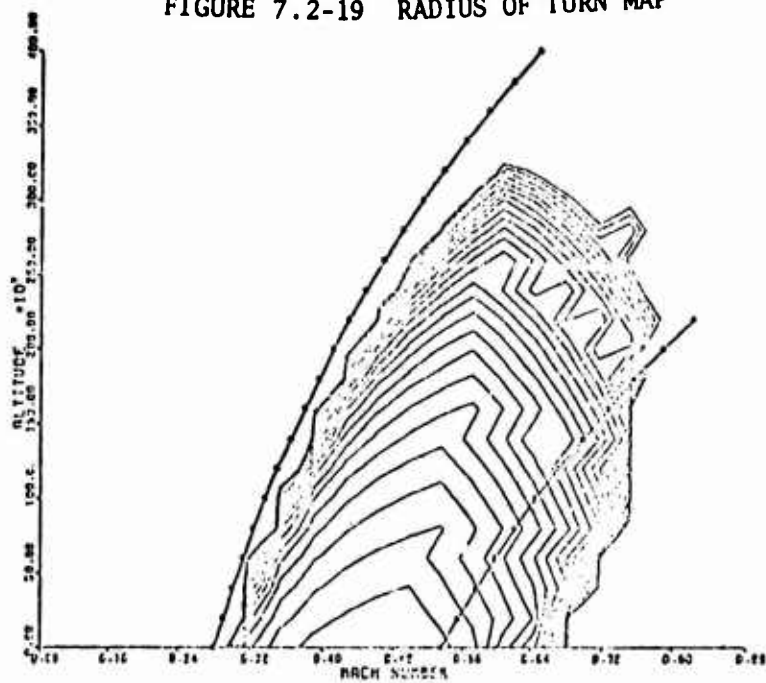
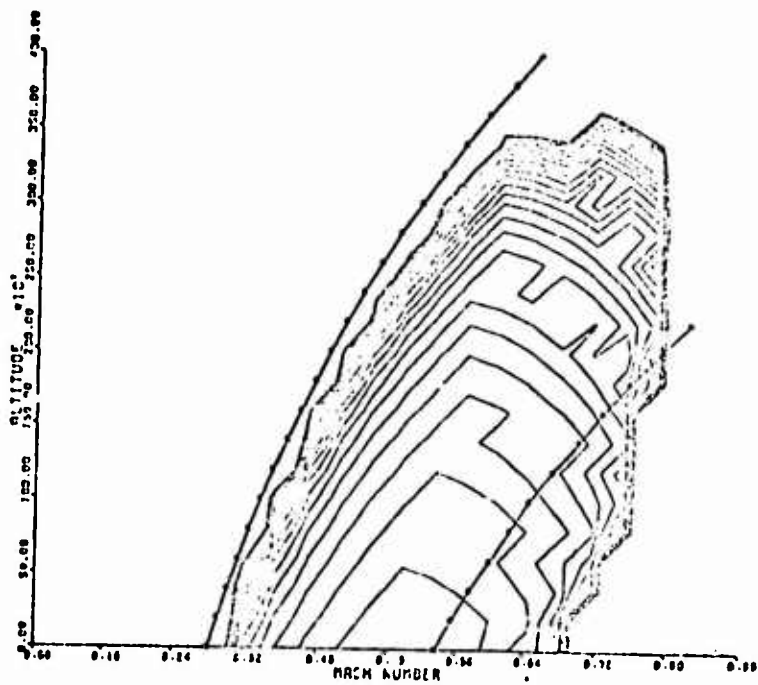


FIGURE 7.2-20. MAP OF TIME TO TURN 180 DEGREES



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.

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 $t = t_0$. 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

$$|\mathbf{R}| = \sqrt{X_e^2 + Y_e^2 + Z_e^2} \quad (7.3.1)$$

The angle between \mathbf{R} and the North pole is given by

$$\phi' = 90 - \phi_L \quad (7.3.2)$$

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,

$$\frac{\mathbf{F}}{m} = \left(\frac{d^2 \mathbf{R}}{dt^2} \right)_e + 2\omega_p \times \left(\frac{d\mathbf{R}}{dt} \right)_e + \omega_p \times \omega_p \times \mathbf{R} \quad (7.3.3)$$

Here \mathbf{F} is the total force acting on the vehicle; m is the vehicle mass, and ω_p is the planet's rotation rate. This vector equation can be expressed in component form using the relationships

$$\frac{F_{X_e}}{m} = \ddot{X}_e + 2\omega_p \dot{Y}_e - \omega_p^2 X_e \quad (7.3.4)$$

$$\frac{F_{Y_e}}{m} = \ddot{Y}_e - 2\omega_p \dot{X}_e - \omega_p^2 Y_e \quad (7.3.5)$$

$$\frac{F_{Z_e}}{m} = \ddot{Z}_e \quad (7.3.6)$$

Alternatively, by introducing the vehicle velocity components, Equations (7.3.4) to (7.3.6) can be reduced to the first order form

$$\dot{X}_e = u_e \quad (7.3.7)$$

$$\dot{Y}_e = v_e \quad (7.3.8)$$

$$\dot{Z}_e = w_e \quad (7.3.9)$$

$$\dot{u}_e = \frac{F_{x_e}}{m} - 2\omega_p v_e + \omega_p^2 X_e \quad (7.3.10)$$

$$\dot{v}_e = \frac{F_{y_e}}{m} + 2\omega_p u_e + \omega_p^2 Y_e \quad (7.3.11)$$

$$\dot{w}_e = \frac{F_{z_e}}{m} \quad (7.3.12)$$

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

$$\dot{m} = \dot{m}(\bar{x}_n(t), \bar{a}_m(t), t) \quad (7.3.13)$$

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 $\bar{a}(t)$ 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

$$\alpha = \tan^{-1} \left(\frac{w}{u} \right) \quad (7.3.14)$$

Sideslip angle, β , is the angle between the velocity vector and the reference axis when looking down on the vehicle platform, that is, along the z axis. In this case,

$$\beta = \tan^{-1} \left(\frac{v}{u} \right) \quad (7.3.15)$$

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

$$C = \theta_{L0} - \theta_L \quad (7.3.16)$$

Angle C is related to the vehicle position by the expression

$$C = \tan^{-1} \left(\frac{Y_e}{X_e} \right) \quad (7.3.17)$$

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 $(180^\circ + C)$, then rotate about Y_g through the angle $(90^\circ - \phi_L)$. The complete transformation can be reduced to the single transformation matrix

$$\begin{vmatrix} \bar{1}_{X_g} \\ \bar{1}_{Y_g} \\ \bar{1}_{Z_g} \end{vmatrix} = \begin{vmatrix} -\sin \phi_L \cos C & -\sin \phi_L \sin C & -\cos \phi_L \\ \sin C & -\cos C & 0 \\ -\cos \phi_L \cos C & -\cos \phi_L \sin C & \sin \phi_L \end{vmatrix} \begin{vmatrix} \bar{1}_{X_e} \\ \bar{1}_{Y_e} \\ \bar{1}_{Z_e} \end{vmatrix} \quad (7.3.18)$$

which defines a direction cosine set (i, j, k) by the equation

$$\begin{vmatrix} \bar{1}_{X_g} \\ \bar{1}_{Y_g} \\ \bar{1}_{Z_g} \end{vmatrix} = \begin{vmatrix} i_1 & j_1 & k_1 \\ i_2 & j_2 & k_2 \\ i_3 & j_3 & k_3 \end{vmatrix} \begin{vmatrix} \bar{1}_{X_e} \\ \bar{1}_{Y_e} \\ \bar{1}_{Z_e} \end{vmatrix} \quad (7.3.19)$$

Planet referenced velocity in the local-geocentric coordinate system is given by

$$\begin{vmatrix} \dot{x}_g \\ \dot{y}_g \\ \dot{z}_g \end{vmatrix} = \begin{vmatrix} i_1 & j_1 & k_1 \\ i_2 & j_2 & k_2 \\ i_3 & j_3 & k_3 \end{vmatrix} \begin{vmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{z}_e \end{vmatrix} \quad (7.3.20)$$

and

$$v_g = \sqrt{\dot{x}_g^2 + \dot{y}_g^2 + \dot{z}_g^2} \quad (7.3.21)$$

Flight path angles are computed by

$$\sigma = \tan^{-1} \left(\frac{\dot{y}_g}{\dot{x}_g} \right) \quad (7.3.22)$$

and

$$\gamma = \sin^{-1} \left(\frac{-\dot{z}_g}{v_g} \right) \quad (7.3.23)$$

Here σ is the heading angle, and λ is the flight path angle.

3.3.3.2 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 ($\dot{x}_{gw}, \dot{y}_{gw}, \dot{z}_{gw}$), then

$$V_A = \sqrt{(\dot{x}_g - \dot{x}_{gw})^2 + (\dot{y}_g - \dot{y}_{gw})^2 + (\dot{z}_g - \dot{z}_{gw})^2} \quad (7.3.24)$$

$$\gamma_A = \sin^{-1} \left[-(\dot{x}_g - \dot{x}_{gw}) / V_A \right] \quad (7.3.25)$$

$$\sigma_A = \tan^{-1} \left[(\dot{y}_g - \dot{y}_{gw}) / (\dot{x}_g - \dot{x}_{gw}) \right] \quad (7.3.26)$$

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, λ_A (defined above); and bank angle, B_A .

The complete transformation from local geocentric horizon coordinates to wind axes is

$$\begin{vmatrix} X_A \\ Y_A \\ Z_A \end{vmatrix} = \begin{vmatrix} \cos \gamma_A \cos \sigma_A & \cos \gamma_A \sin \sigma_A & -\sin \gamma_A \\ -\sin \sigma_A \cos B_A & \cos \sigma_A \cos B_A & \cos \gamma_A \sin B_A \\ \sin \sigma_A \cos B_A & \sin \gamma_A \sin \sigma_A \sin B_A & \cos \gamma_A \cos B_A \\ \sin \sigma_A \sin B_A & -\cos \sigma_A \sin B_A & \cos \gamma_A \cos B_A \\ + \sin \gamma_A \cos \sigma_A \sin B_A & + \sin \gamma_A \sin \sigma_A \cos B_A & \end{vmatrix} \begin{vmatrix} X_G \\ Y_G \\ Z_G \end{vmatrix} \quad (7.3.27)$$

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_e) coordinate system becomes

$$\begin{vmatrix} F_{X_e} \\ F_{Y_e} \\ F_{Z_e} \end{vmatrix} = \begin{vmatrix} o_1 & o_2 & o_3 \\ p_1 & p_2 & p_3 \\ q_1 & q_2 & q_3 \end{vmatrix} \begin{vmatrix} F_{X_A} \\ F_{Y_A} \\ F_{Z_A} \end{vmatrix} + \begin{vmatrix} m g_{X_e} \\ m g_{Y_e} \\ m g_{Z_e} \end{vmatrix} \quad (7.3.31)$$

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 7.3-8 involves rotation first through angle of attack, α , then through an auxiliary angle, β' .

The complete transformation is

$$\begin{vmatrix} X_A \\ Y_A \\ Z_A \end{vmatrix} = \begin{vmatrix} \cos \beta' \cos \alpha & \sin \beta' & \cos \beta' \sin \alpha \\ -\sin \beta' \cos \alpha & \cos \beta' & -\sin \beta' \sin \alpha \\ -\sin \alpha & 0 & \cos \alpha \end{vmatrix} \begin{vmatrix} x \\ y \\ z \end{vmatrix} \quad (7.3.32)$$

which defines the (u, v, 2) direction cosines

$$\begin{vmatrix} X_A \\ Y_A \\ Z_A \end{vmatrix} = \begin{vmatrix} u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \\ w_1 & w_2 & w_3 \end{vmatrix} \begin{vmatrix} x \\ y \\ z \end{vmatrix} \quad (7.3.33)$$

and the force coefficient transformation

$$\begin{vmatrix} -C_D \\ C_{Y_A} \\ -C_L \end{vmatrix} = \begin{vmatrix} u_1 & u_2 & u_3 \\ v_1 & v_2 & v_3 \\ w_1 & w_2 & w_3 \end{vmatrix} \begin{vmatrix} -C_A \\ C_Y \\ -C_N \end{vmatrix} \quad (7.3.34)$$

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, ω_{pt} .

The transformation from planet referenced velocities to inertial velocities is

$$\begin{vmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{vmatrix} = \begin{vmatrix} \cos \omega_{pt} & \sin \omega_{pt} & 0 \\ -\sin \omega_{pt} & \cos \omega_{pt} & 0 \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} \dot{X}_e + \omega_p Y_e \\ \dot{Y}_e - \omega_p X_e \\ \dot{Z}_e \end{vmatrix} \quad (7.3.35)$$

The components of inertial velocities are used to calculate the inertial speed of the body as

$$v_I = \sqrt{\dot{X}^2 + \dot{Y}^2 + \dot{Z}^2} \quad (7.3.36)$$

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 λ and heading angle σ are calculated with respect to the local geocentric coordinates. By definition λ_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 λ_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 7.3-9 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

$$\frac{X^2}{R_e^2} + \frac{Z^2}{R_p^2} = 1 \quad (7.3.37)$$

It is apparent from Figure 7.3-9 that the most significant difference between the geocentric referenced position and the geodetic position is the distance \bar{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 \bar{OC} .

The flight path and heading angles corrected to the local geodetic latitude are computed by

$$\gamma_D = \text{Sin}^{-1} \left(\frac{-\dot{z}_g - \{\dot{x}_g(\phi_g - \phi_L)\}}{V_g} \right) \quad (7.3.38)$$

and

$$\sigma_D = \text{Sin}^{-1} \left(\frac{\dot{y}_g}{\sqrt{\{\dot{x}_g + \dot{z}_g(\phi_g - \phi_L)\}^2 + \dot{y}_g^2}} \right) \quad (7.3.39)$$

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

$$R_D = \int_{t_1}^{t_2} \frac{R_{\phi_L}}{R} V_g \text{Cos } \gamma \, dt \quad (7.3.40)$$

The flight path angle, λ , 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

$$R_g = \left[\frac{R_{\phi_L} + R_{\phi_{L_0}}}{2} \right] \cos^{-1} \left[\sin \phi_L \sin \phi_{L_0} + \cos \phi_L \cos \phi_{L_0} \cos(\theta_L - \theta_{L_0}) \right] \quad (7.3.41)$$

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_0 , ϕ_{L_0} , and θ_{L_0} , 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 .

$$X_D = \left(\frac{R_{\phi_L} + R_{\phi_{L_0}}}{2} \right) \cdot \cos^{-1} \left(\frac{\cos LF}{\cos(\sin^{-1}(\sin LF \sin \xi))} \right) \quad (7.3.42)$$

$$Y_D = \left(\frac{R_{\phi_L} + R_{\phi_{L_0}}}{2} \right) \sin^{-1} (\sin LF \sin \xi) \quad (7.3.43)$$

where

$$\xi = \zeta - \sigma_p \quad (7.3.44)$$

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:

$$v_{\text{theo}} = \int_{t_1}^{t_2} \frac{T_{\text{VAC}}}{m} dt \quad (7.3.45)$$

Speed Loss Due to Gravity:

$$V_{\text{grav}} = \int_{t_1}^{t_2} -g_z \sin \gamma \, dt \quad (7.3.46)$$

Speed Loss Due to Aerodynamic Drag:

$$V_D = \int_{t_1}^{t_2} \frac{D}{m} \, dt \quad (7.3.47)$$

Speed Loss Due to Atmospher Back Pressure Upon the Engine Nozzle:

$$V_P = \int_{t_1}^{t_2} - \frac{P_{A_e}}{m} \, dt \quad (7.3.48)$$

Maneuvering Losses:

$$V_{\text{ML}} = \int_{t_1}^{t_2} \left(\frac{T_{\text{VAC}} - P_{A_e}}{m} \right) (\cos \alpha - 1) \, dt \quad (7.3.49)$$

The resultant velocity $V'_g(t_2)$ is obtained by adding the components computed to the initial value $V'_g(t_1)$

$$V'_g(t_2) = V'_g(t_1) + V_{\text{theo}} + V_{\text{grav}} + V_D + V_P \quad (7.3.50)$$

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

$$\sigma_I = \tan^{-1} \left(\frac{\dot{Y}_g + \omega_p |R| \cos \phi_L}{\dot{X}_g} \right) \quad (7.3.51)$$

$$\gamma_I = \sin^{-1} \left(\frac{\dot{Z}_g}{|V_I|} \right) \quad (7.3.52)$$

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

$$\cos i = \cos \phi_L \sin \sigma_I \quad (7.3.53)$$

The difference in longitude between the vehicle and the ascending node, ν , is given by

$$\tan \nu = \sin \phi_L \tan \sigma_I \quad (7.3.54)$$

The inertial longitude is given by

$$\theta_I = \theta_L - \omega_p t \quad (7.3.55)$$

and the inertial longitude of the ascending node by

$$\Omega = \theta_I - \nu \quad (7.3.56)$$

It is convenient to know the central angle, u , in the orbital plane. Measuring from the ascending node,

$$\tan u = \frac{\tan \phi_L}{\cos \sigma_I} \quad (7.3.57)$$

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_0} , by the expression

$$\phi_s = \phi_{s_0} + \omega_s t \quad (7.3.58)$$

The satellite angular velocity is obtained from the satellite inertial velocity, V_{c_s} , where

$$V_{c_s} = \sqrt{\frac{\mu_g}{(R_e + h_s)}} \quad (7.3.59)$$

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

$$\omega_s = \frac{V_{c_s}}{R_e + h_s} \quad (7.3.60)$$

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

$$L = q(V,h) S C_L(V,h,\alpha,\beta) \quad (7.3.61)$$

$$D = q(V,h) S C_D(V,h,\alpha,\beta) \quad (7.3.62)$$

$$Y = q(V,h) S C_Y(V,h,\alpha,\beta) \quad (7.3.63)$$

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.

$$\begin{aligned} C_L = & C_{L_0} + C_{L_\alpha} \alpha + C_{L_{\alpha^2}} \alpha^2 + C_{L_\beta} |\beta| \\ & + C_{L_{\beta^2}} \beta^2 + C_{L_{\alpha\beta}} \alpha |\beta| \end{aligned} \quad (7.3.64)$$

$$\begin{aligned} C_D = & C_{D_0} + C_{D_\alpha} |\alpha| + C_{D_{\alpha^2}} \alpha^2 + C_{D_\beta} |\beta| \\ & + C_{D_{\beta^2}} \beta^2 + C_{D_{\alpha\beta}} |\alpha| |\beta| \end{aligned} \quad (7.3.65)$$

$$\begin{aligned} C_Y = & C_{Y_0} + C_{Y_\alpha} |\alpha| + C_{Y_{\alpha^2}} \alpha^2 + C_{Y_\beta} \beta \\ & + C_{Y_{\beta^2}} \beta |\beta| + C_{Y_{\alpha\beta}} |\alpha| \beta \end{aligned} \quad (7.3.66)$$

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:

$$T = \text{Max} [T_{\text{vac}} - PA_e, 0] \quad (7.3.67)$$

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

$$T = \epsilon_{13} T_{VAC} + \epsilon_{14} \quad (7.3.68)$$

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

$$T_x = T \cos \lambda_T \quad (7.3.69)$$

$$T_y = -T \sin \lambda_T \cos \phi_T \quad (7.3.70)$$

and

$$T_z = -T \sin \lambda_T \sin \phi_T \quad (7.3.71)$$

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, m , is set equal to the negative of the total mass flow rate, $-\dot{m}_t$. 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

$$W_T = 32.174 \cdot m \quad (7.3.72)$$

The propellant consumed is computed as

$$m_f = m_0 - m \quad (7.3.73)$$

where m_0 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.

| <u>Layer</u> | <u>H_b-Lower Altitude</u> (Geopotential Meters) | <u>Upper Altitude</u> (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:

$$H_{gp} = \frac{.3048h}{1+.3048h/6356766} \quad \text{Meters} \quad (7.3.74)$$

$$T_m = (T_M)_b [1 + K_1(H_{gp}-H_b)] \quad ^\circ R \quad (7.3.75)$$

$$T = T_M [A - B \tan^{-1} \left(\frac{H_{gp}-C}{D} \right)] \quad ^\circ R \quad (7.3.76)$$

$$P = P_b [1 + K_1(H_{gp}-H_b)]^{-K_2} \quad \text{Lb/Ft}^2 \quad (7.3.77)$$

$$\rho = \rho_b [1 + K_1(H_{gp} - H_b)]^{-(1+K_2)} \quad \text{Slugs/Ft}^3 \quad (7.3.78)$$

$$V_s = 49.021175(T_M)^{1/2} \quad \text{Ft/sec} \quad (7.3.79)$$

$$v = 2.269681 \times 10^{-8} \left[\frac{T^{3/2}}{(T+198.72)\rho} \right] \quad \text{Ft}^2/\text{sec} \quad (7.3.80)$$

For the isothermal layers 2, 4, and 6, the following changes are made

$$P = P_b e^{-K_3(H_{gp}-H_b)} \quad (7.3.81)$$

$$\rho = \rho_b e^{-K_3(H_{gp}-H_b)} \quad (7.3.82)$$

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

$$U = \frac{\mu_E}{R} \left[1 + \frac{\gamma}{3} \left(\frac{R_e}{R} \right)^2 P_2 + \frac{H}{5} \left(\frac{R_e}{R} \right)^3 P_3 + \frac{K}{30} \left(\frac{R_e}{R} \right)^4 P_4 + \dots \right] \quad (7.3.83)$$

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:

$$\mu_g = 1.407698 \times 10^{16} \quad \text{ft} / \text{sec}^2$$

$$R_e = 20,925,631. \quad \text{ft.}$$

$$J = 1623.41 \times 10^{-6}$$

$$K = 6.37 \times 10^{-6} \quad (7.3.89)$$

It should be noted that these constants and equations pertain to the planet Earth; however, it is possible to use these same equations for any other planet.

REFERENCES:

1. Hague, D. S., Three-Degree-of-Freedom Problem Optimization Formulation--Analytical Development, Part 1, Volume III, FDL-TDR-64-1, October 1964.
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 Astroynamics 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.

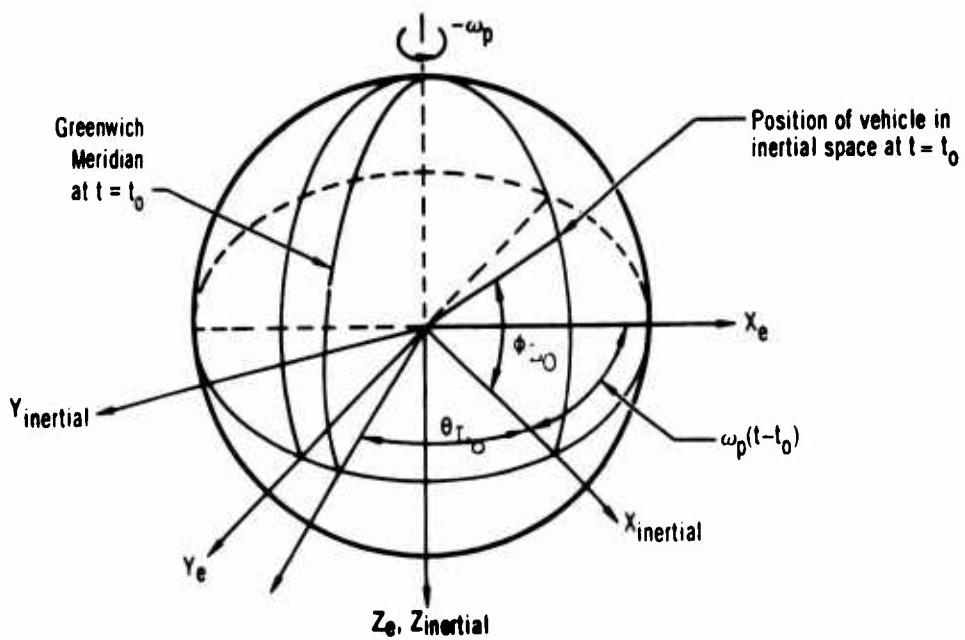


FIGURE 7.3-1. BASIC COORDINATE SYSTEM

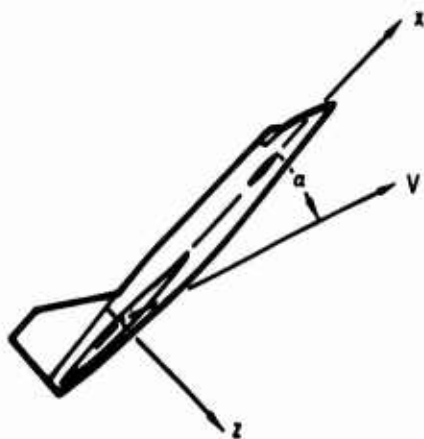


FIGURE 7.3-2. ANGLE OF ATTACK

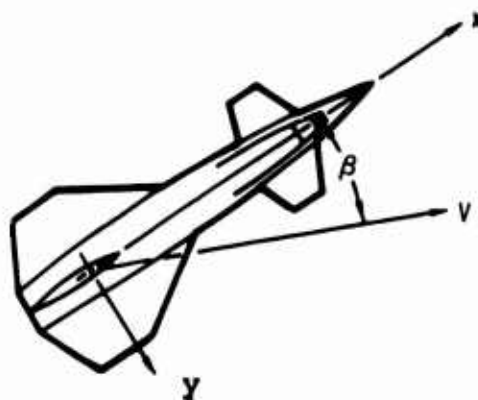


FIGURE 7.3-3. SIDESLIP ANGLE

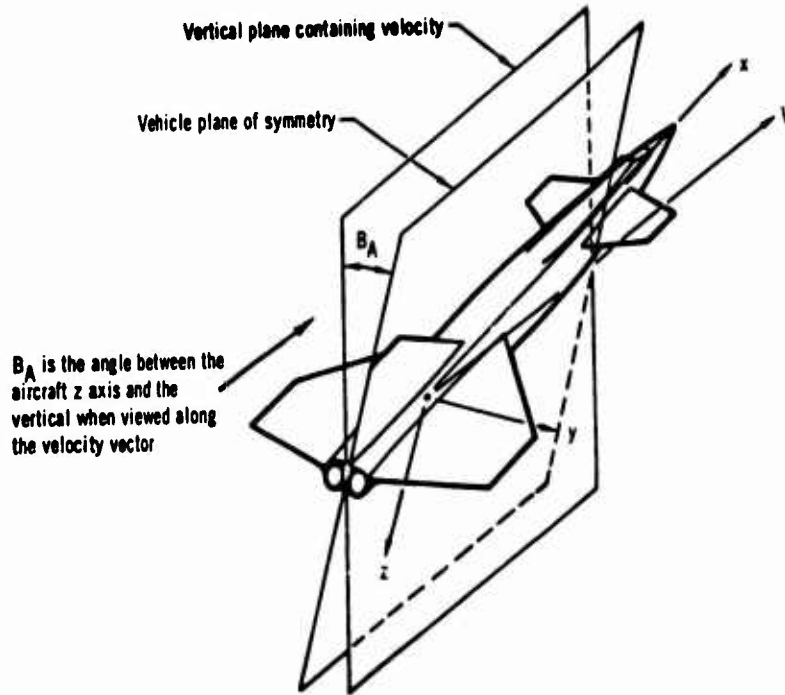


FIGURE 7.3-4. BANK ANGLE

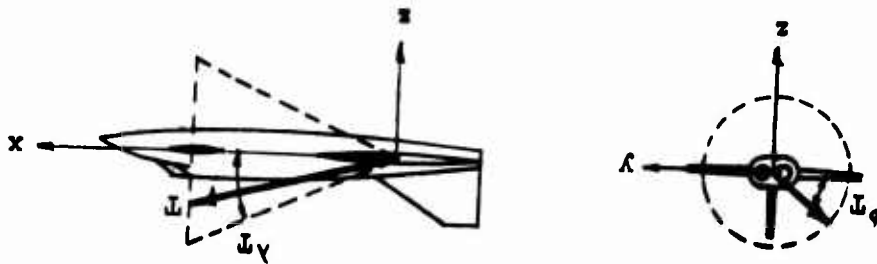


FIGURE 7.3-5. THRUST ANGLES

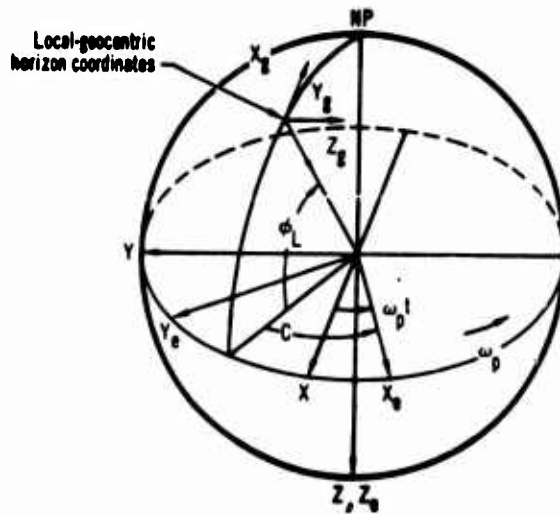


FIGURE 7.3-6. RELATION BETWEEN LOCAL-GEOCENTRIC, INERTIAL AND EARTH-REFERENCED COORDINATES FOR POINT MASS PROBLEMS

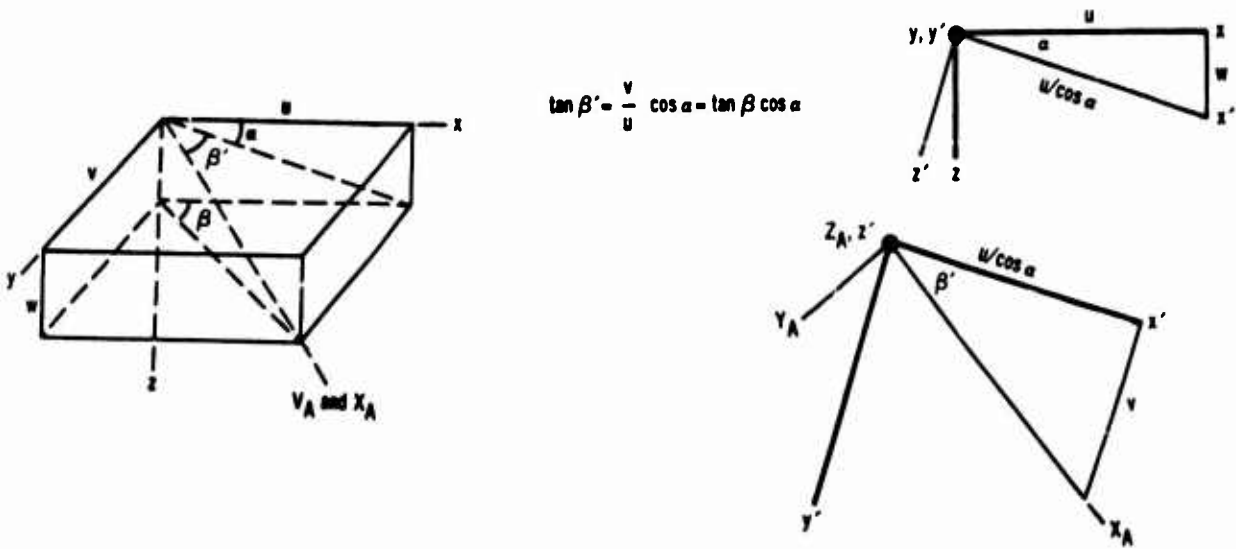


FIGURE 7.3-7. RELATIONSHIP BETWEEN BODY AXES AND WIND AXES

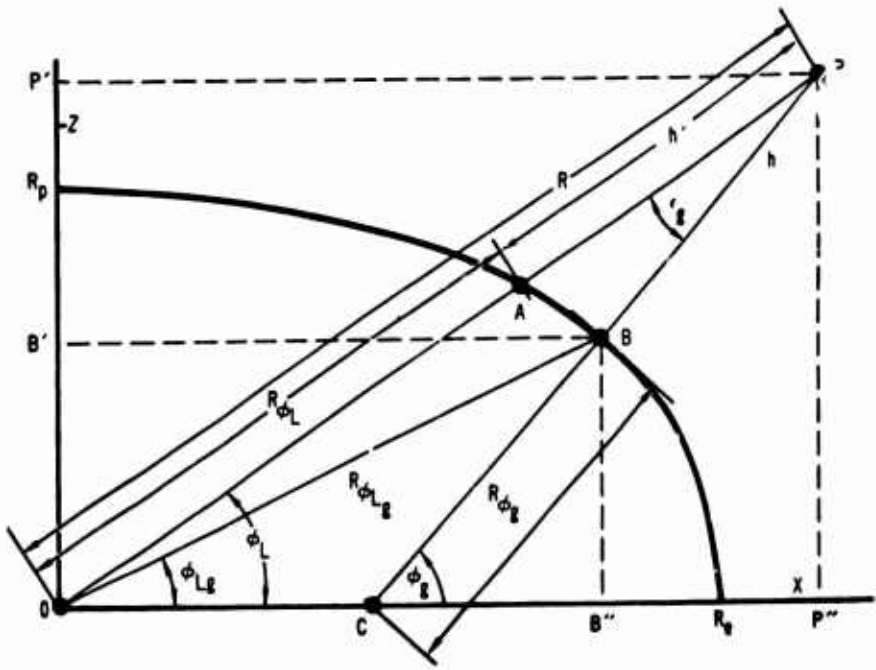


FIGURE 7.3-8. PLANET-OBLATENESS EFFECT ON LATITUDE AND ALTITUDE

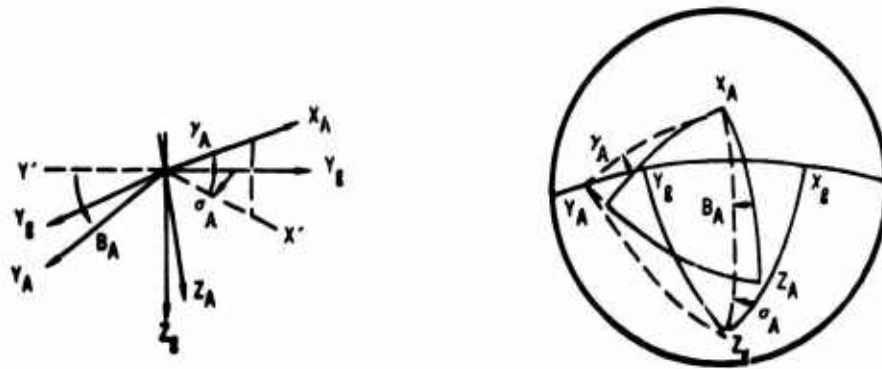


FIGURE 7.3-9. RELATIONSHIP BETWEEN LOCAL-GEOCENTRIC AXES AND WIND AXES

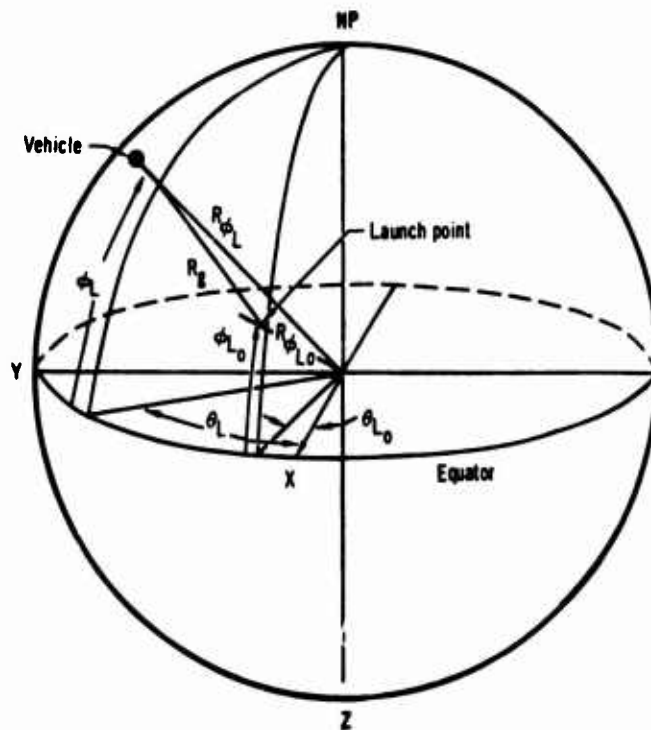


FIGURE 7.3-10. GREAT-CIRCLE RANGE

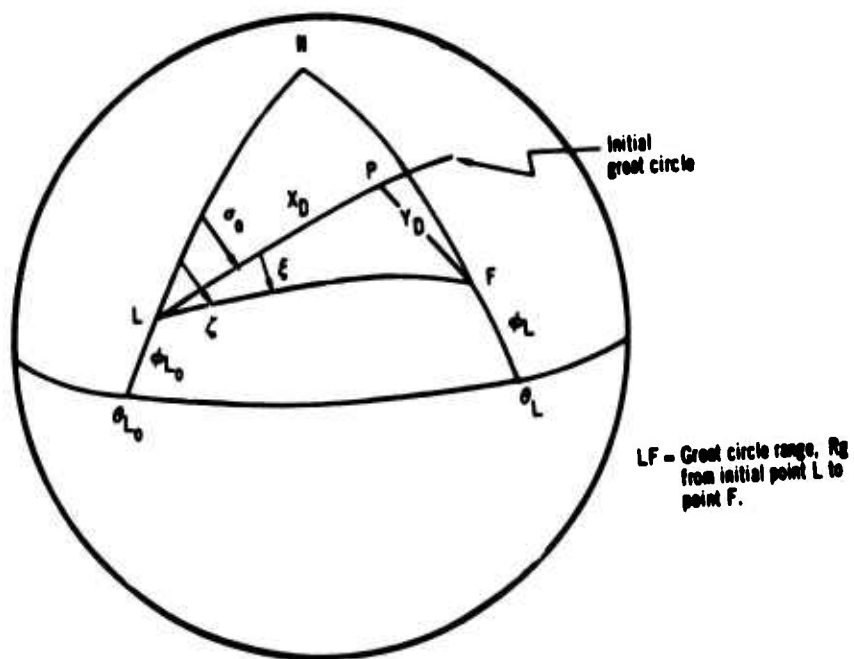


FIGURE 7.3-11. DOWNRANGE AND CROSSRANGE GEOMETRY

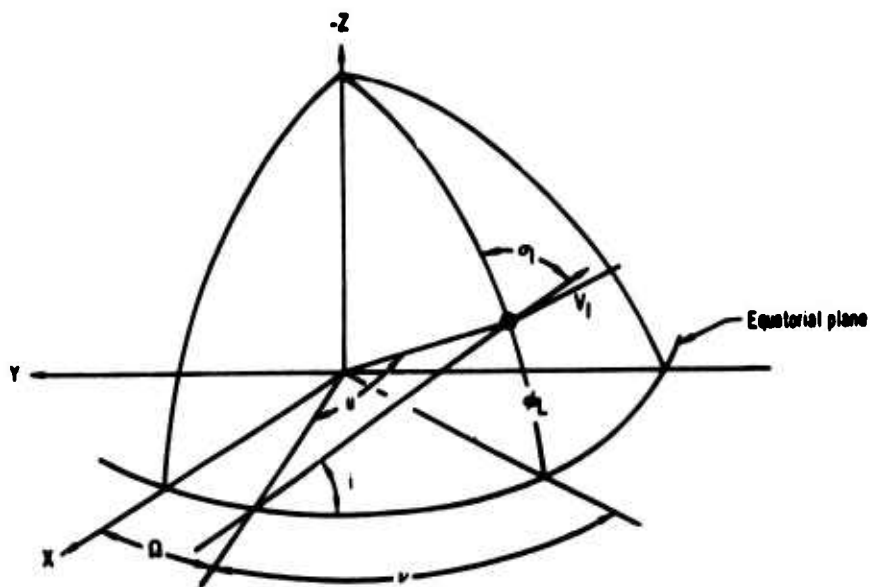


FIGURE 7.3-12. ORBITAL PLANE GEOMETRY

SECTION 8

STRUCTURES

The ODIN/MFV program as installed at Wright-Patterson Air Force Base 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.

8.1 PROGRAM SSAM: SWEEP 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.

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

$$-\sum_0^n \left(\frac{w}{v}\right)_n = \alpha_f \quad (8.1.1)$$

where w/v is the induced downwash angle of the three-quarter chord and α_f is the section airfoil free air angle of attack. In matrix notation

$$\left\{\frac{w}{v}\right\}_{3c/4} = \frac{1}{4\pi v} [S_1] \{l\} \quad (8.1.2)$$

leading to the basic equation

$$\left[\frac{1^\circ}{4q\bar{m}_0} \right] [S_1] \{\ell\} = \{\alpha_f\} \quad (8.1.3)$$

where

$\left[\frac{1^\circ}{4q\bar{m}_0} \right]$ = 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

$$\alpha_f = \alpha_s + \alpha_r + \alpha_g \quad (8.1.4)$$

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,

$$\alpha_{s\sigma} = \int_0^\sigma \frac{m \, M \, ds}{EI} + \int_0^\sigma \frac{t \, T \, ds}{GJ} \quad (8.1.5)$$

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 :

$$\{\alpha_s\} = [S_2] \{l\} \quad (8.1.6)$$

where

$[S_2]$ = 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

$$\begin{aligned} L_{\text{FUSELAGE}} &= qS[CL_{F_0} + (CL_F)_\alpha \alpha_r] \\ M_{\text{FUSELAGE}} &= qS\bar{c}[Cm_{F_0} + (Cm_F)_\alpha \alpha_r] \end{aligned} \quad (8.1.7)$$

where

CL_{F_0}, Cm_{F_0} = aerodynamic C_L and C_m for $\alpha_r = 0$

$(CL_F), (Cm_F)$ = 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_0 = slope of lift coefficient
 ac = airfoil aerodynamic center position
 α_0 = 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

$$\{\Delta W_i\}_l = k_1\{\Delta W_i^1\} + k_2\{\Delta W_i^2\} + \dots + k_n\{\Delta W_i^n\} \quad (8.1.8)$$

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 symmetrically. The upper and lower beam bending material are treated as equal areas. A factor K_0 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_0 = total material area of one segment of bending material
- K_0 = skin segment area/total segment area = $t_s w / A_0$.

The general equation for representing the combined maximum wing stress in terms of the material area A_0 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

$$I = A_0 d_e^2 / 2.0$$

$$A_B = d_a w$$

The average skin thickness in terms of A_0 is

$$t_s = k_0 A_0 / w \quad (8.1.9)$$

The maximum wing bending and torsion stress in terms of the applied bending moment M and torsion moment T becomes

$$\sigma_b = \frac{My}{I} = \frac{Md}{A_0 d_e^2} \quad (8.1.10)$$

$$\sigma_t = \frac{T}{2ABt_s} = \frac{T}{2A_0 K_0 d_a} \quad (8.1.11)$$

The shear loads V also produce a shear stress, σ_{sw} , in the spar webs, Here,

$$\sigma_{sw} = \frac{V}{d_e t_s} = \frac{Vw}{d_e K_0 A_0} \quad (8.1.12)$$

The maximum combined principal stress from basic structure considerations is expressed as

$$\sigma = \frac{\sigma_b}{2} + \sqrt{\left(\frac{\sigma_b}{2}\right)^2 + (\sigma_{sw} \pm \sigma_t)^2} \quad (8.1.13)$$

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_0 ,

$$\sigma = \frac{1}{2A_0} \left[\frac{Md}{d_e^2} + \sqrt{\left(\frac{Md}{d_e^2}\right)^2 + \frac{w^2}{K_0^2} \left(\frac{V}{d} \pm \frac{T}{wd_a}\right)^2} \right] \quad (8.1.14)$$

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_0 where

$$A_0 = \frac{1}{2\sigma_A} \left[\frac{Md}{d_e^2} + \left(\frac{Md}{d_e^2}\right)^2 + \frac{w^2}{K_0^2} \left(\frac{V}{d} \pm \frac{T}{wd_a}\right)^2 \right] \quad (8.1.15)$$

The above general equation is used with the proper sign to calculate A_0 based upon the allowable bending stress for the front spar, rear spar, and maximum spar depth. These areas are then averaged with weighting factor, 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

$$[S_2] = f_1 \begin{matrix} \text{(wing planform} \\ \text{geometry} \end{matrix} \begin{matrix} 0 \\ [\frac{1}{EI}] \end{matrix} + f_2 \begin{matrix} \text{(wing planform} \\ \text{geometry} \end{matrix} \begin{matrix} 0 \\ [\frac{1}{GJ}] \end{matrix}$$

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_0 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_0 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_0 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_0 values to compute a new S_2 matrix. Here

$$[\frac{1}{EI}] = \frac{A_0 d_e^2}{2E}$$

$$[\frac{1}{GJ}] = \frac{1.0}{[\frac{d_{FS} + d_{RS}}{2tw}] (\frac{d_a}{w})^2 G}$$

where

$$t_s = K_0 A_0 / w$$

$$t_w = k_w t_s$$

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).

THE ESTIMATED WING FLEXIBILITY FOR THE FIRST
 ITERATION IS AS FOLLOWS.

| ETA | RFLEX | TFLEX |
|------------|--------------|--------------|
| .9500 | .368460 | .523520 |
| .8500 | .141080 | .183250 |
| .7500 | 6.171000E-02 | 9.539000E-02 |
| .6500 | 3.182000E-02 | 4.923000E-02 |
| .5500 | 1.991000E-02 | 3.226000E-02 |
| .4500 | 1.187000E-02 | 1.681000E-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 | 1.940000E-03 | 0. |

WHERE -

- ETA WING STATION (SPAN FRACTION)
- RFLEX 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)

TABLE 8.1-1a. INITIAL GUESS AT WING FLEXIBILITIES

FLIGHT LOAD CONDITION 1.

ALTITUDE (FT) 0.
EQUIVALENT AIRSPEED (KT) 280.00
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) 138.95
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 | CN | ALP-F | ALPHA | SHEAR | MOM | TOR | DEFL |
|------|------|-------|--------|--------|--------|--------|-------|--------|-------|
| .950 | .183 | .802 | -.0044 | 8.620 | 7.675 | 7.08 | .17 | .01 | 96.22 |
| .850 | .267 | .981 | -.0238 | 9.473 | 8.017 | 22.98 | 1.54 | .13 | 74.03 |
| .750 | .345 | 1.095 | -.0266 | 10.392 | 8.673 | 43.58 | 4.69 | .33 | 54.15 |
| .650 | .406 | 1.132 | -.0283 | 10.518 | 9.607 | 51.44 | 9.69 | -.94 | 37.80 |
| .550 | .447 | 1.108 | -.0293 | 12.232 | 10.639 | 75.26 | 15.71 | -.39 | 24.63 |
| .450 | .500 | 1.109 | -.0333 | 13.226 | 11.610 | 83.50 | 25.17 | -1.60 | 14.63 |
| .350 | .564 | 1.103 | -.0461 | 13.776 | 12.481 | 107.58 | 33.81 | -1.20 | 7.70 |
| .250 | .621 | 1.071 | -.0820 | 14.417 | 13.134 | 127.54 | 44.83 | -2.32 | 3.38 |
| .150 | .665 | 1.027 | -.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) 13.836 -.5327F
BALANCING TAIL LOAD (LG/G) -38397 -17157
AIRPLANE LIFT COEFF. (TAIL-OFF) 1.0583 2.23226E-02
PITCHING MOMENT COEFF. (0.25 MAC) -5.61116E-02 -6.15654E-02

TABLE 8.1-1b. FINAL DEFLECTED WING, LOAD CASE 1

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 LR) 301.90
LOAD FACTOR 2.5000
CENTER OF GRAVITY (MAC FRACTION) .16200
AIR DENSITY RATIO .37410

CENTERLINE AIRLOADS (ONLY)
SHEAR (1000 LR) 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 | 89.90 |
| .850 | .233 | .573 | -.0052 | 4.054 | 2.599 | 19.05 | 1.21 | .20 | 70.46 |
| .750 | .310 | .658 | -.0267 | 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 | -.1323 | 6.285 | 4.692 | 68.33 | 13.91 | -1.24 | 24.40 |
| .450 | .514 | .764 | -.1548 | 7.383 | 5.767 | 75.62 | 23.27 | -3.66 | 14.50 |
| .350 | .599 | .785 | -.1750 | 8.107 | 6.812 | 103.69 | 32.40 | -5.16 | 7.57 |
| .250 | .679 | .785 | -.2088 | 8.942 | 7.659 | 129.68 | 44.24 | -8.71 | 3.32 |
| .150 | .742 | .767 | -.2362 | 9.546 | 8.253 | 149.81 | 56.71 | -15.50 | 1.05 |
| .050 | .783 | .733 | -.2543 | 10.031 | 8.747 | 173.88 | 67.79 | -32.14 | 0.00 |

.082 SIDE OF BODY LOADS 166.11 64.21 -26.76
.082 STREAMWISE LOADS 55.83 -41.50

AIRPLANE LIFT CURVE SLOPE (PER DEG) 7.11354E-02
BALANCING TAIL LOAD (LG/G) -13315
GUST VELOCITY, IF USED (FT/SEC) -0.
THRUST PER ENGINE (10000 LR) .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 (LG/G) -64515 -31228
AIRPLANE LIFT COEFF. (TAIL-OFF) .71449 2.72341E-02
PITCHING MOMENT COEFF. (0.25 MAC) -9.72546E-02 -7.51139E-02

TABLE 8.1-1c. FINAL DEFLECTED WING, LOAD CASE 12

| ETA | SKINL | ID | ASEG | ASTR | TSKIN | TWEB | TSTIF | AMAT | VMAT | WMAT |
|------|--------|------|--------|---------|-------|-------|-------|--------|----------|---------|
| .950 | 0.0000 | TS | 6.816 | 18.072 | .0640 | .0704 | 4.42 | 14.92 | 772.26 | 78.00 |
| .850 | 0.0000 | PS | 10.305 | 26.890 | .0919 | .0732 | 6.03 | 22.26 | 1924.49 | 194.37 |
| .750 | 0.0000 | PS | 15.350 | 40.221 | .1350 | .0995 | 7.80 | 33.29 | 2875.71 | 290.45 |
| .650 | 0.0000 | PS | 26.640 | 66.345 | .1743 | .1204 | 13.21 | 56.52 | 4566.36 | 461.20 |
| .550 | 0.0000 | PS | 32.171 | 78.918 | .1734 | .0999 | 15.92 | 67.23 | 6224.51 | 628.68 |
| .450 | 0.0000 | -MAX | 39.696 | 98.032 | .1801 | .1290 | 19.57 | 83.52 | 7582.30 | 765.81 |
| .350 | 0.0000 | RS | 45.206 | 109.952 | .1757 | .1142 | 21.92 | 94.67 | 8912.13 | 900.13 |
| .250 | 0.0000 | PS | 53.848 | 126.955 | .1804 | .1249 | 25.44 | 113.22 | 10151.48 | 1025.30 |
| .150 | 0.0000 | PS | 64.366 | 148.796 | .1933 | .1717 | 28.71 | 138.63 | 11815.93 | 1193.41 |
| | | SH | | 163.583 | | | | | 9061.93 | 915.26 |
| .050 | | CS | 46.607 | 100.140 | .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
 - RS REAR SPAR
 - SH 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 (LBS/SIDE)

TABLE 8.1-1d. FINAL WING, WEIGHT SUMMARY

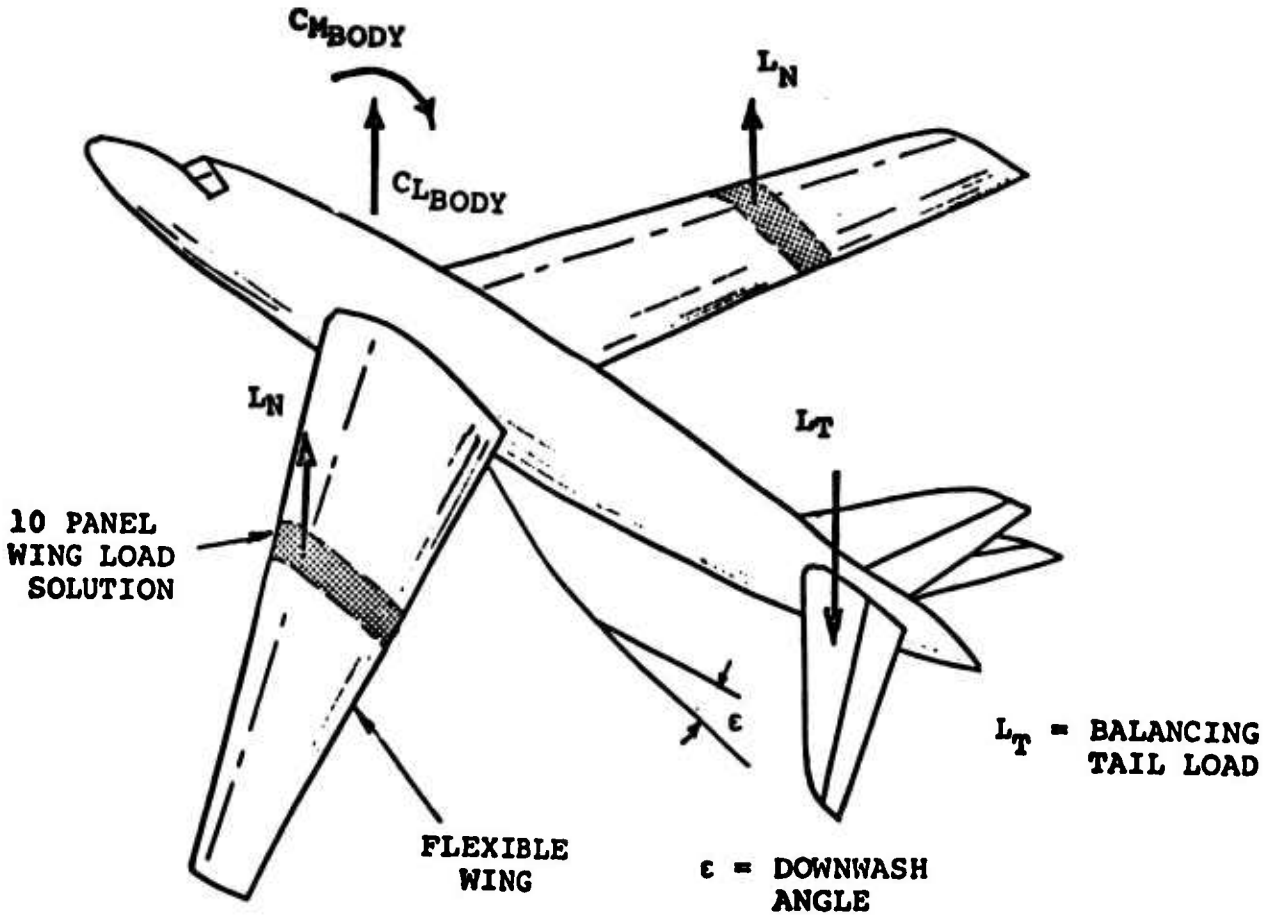
| ETA STA. | FINAL STIFFNESS | | | |
|-------------|-----------------|--------------|------------------|------------------|
| | EI(10E-9) | GJ(10E-9) | EI(10E-9) BOX | GJ(10E-9) BOX |
| .95 | 2.749152E+00 | 1.533305E+00 | 2.749152E+00 | 1.533305E+00 |
| .85 | 8.332625E+00 | 5.145732E+00 | 7.055567E+00 | 4.116586E+00 |
| .75 | 2.043949E+01 | 1.191455E+01 | 1.406709E+01 | 9.546919E+00 |
| .65 | 3.283945E+01 | 2.150318E+01 | 2.720675E+01 | 1.974580E+01 |
| .55 | 4.869643E+01 | 3.668477E+01 | 4.581095E+01 | 3.209517E+01 |
| .45 | 8.125445E+01 | 6.435236E+01 | 7.643399E+01 | 5.571633E+01 |
| .35 | 1.297903E+02 | 1.055615E+02 | 1.260100E+02 | 9.006952E+01 |
| .25 | 2.250999E+02 | 1.662610E+02 | 2.149951E+02 | 1.569982E+02 |
| .15 | 4.398154E+02 | 3.276241E+02 | 4.276954E+02 | 3.276241E+02 |
| .05 | 5.396119E+02 | 3.884961E+02 | 5.279960E+02 | 3.884961E+02 |

| ETA STA. | FLEXIBILITY | | | |
|-------------|-----------------|--------------|------------------------|--------------|
| | FINAL ITERATION | | NEXT TO LAST ITERATION | |
| | 10E9/EI | 10E9/GJ | 10E9/EI | 10E9/GJ |
| .95 | 3.637485E-01 | 6.521858E-01 | 3.637485E-01 | 6.521858E-01 |
| .85 | 1.200112E-01 | 1.943358E-01 | 1.201474E-01 | 1.945579E-01 |
| .75 | 4.892493E-02 | 8.393096E-02 | 4.897950E-02 | 8.402457E-02 |
| .65 | 3.045202E-02 | 4.653475E-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 | 1.231979E-02 | 1.555561E-02 |
| .35 | 7.704737E-03 | 9.473153E-03 | 7.714739E-03 | 9.485450E-03 |
| .25 | 4.442473E-03 | 6.014638E-03 | 4.447178E-03 | 6.021009E-03 |
| .15 | 2.278862E-03 | 3.052278E-03 | 2.280929E-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

TABLE 8.1-1e. FINAL WING FLEXIBILITIES



PROGRAM SOLVES FOR WING PANEL LOADS AND
BALANCES AIRPLANE FOR EACH SPECIFIED
FLIGHT CONDITION

(12 EQUATIONS)

$$\begin{bmatrix} \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} \\ \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} \\ \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} \\ \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} \\ \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} \\ \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} \\ \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} \\ \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} \\ \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} \\ \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} \\ \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} \\ \text{---} & \text{---} & \text{---} & \text{---} & \text{---} & \text{---} \end{bmatrix} \times \begin{bmatrix} L_1 \\ L_2 \\ L_3 \\ \vdots \\ L_{10} \\ \text{LIFT} \\ \text{MOM.} \end{bmatrix} = 0$$

FIGURE 8.1-1. PROGRAM MODEL AND BASIC EQUATIONS

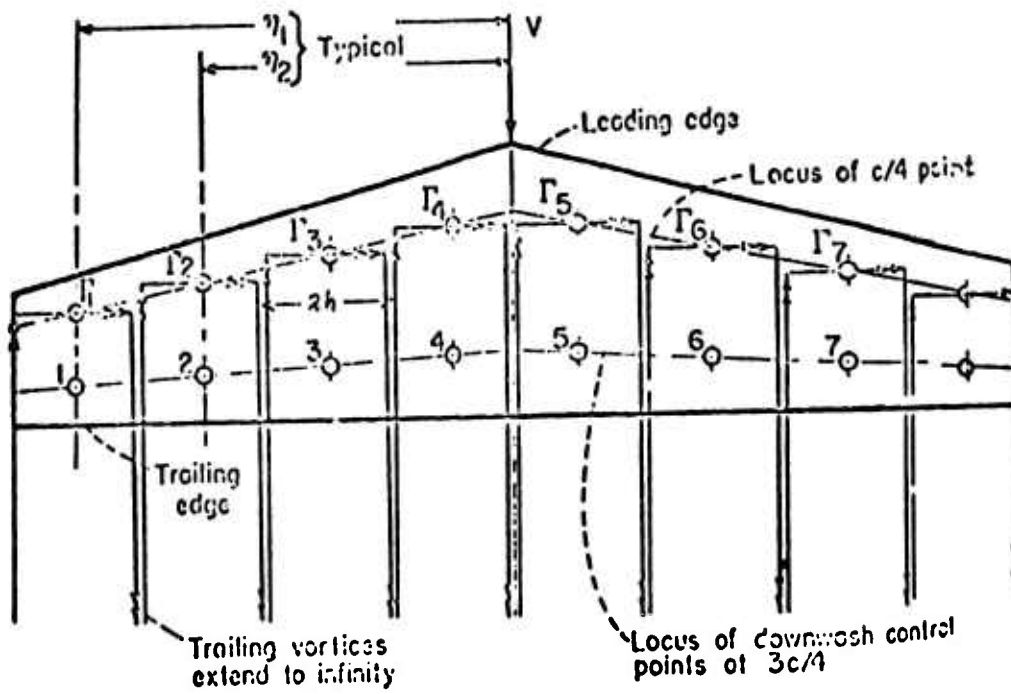


FIGURE 8.1-2. WEISSINGER SUBSONIC STEADY STATE AERODYNAMIC MODEL

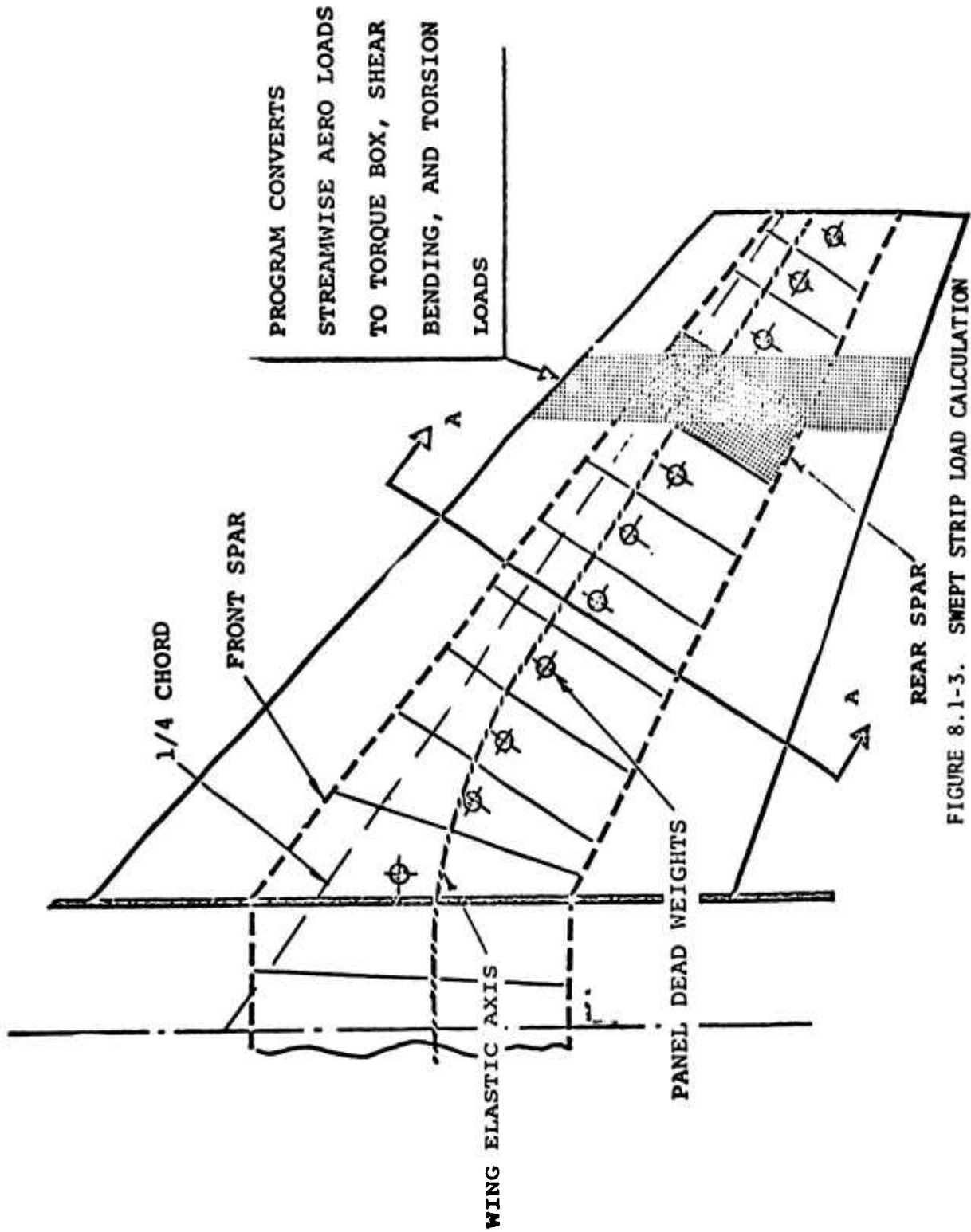


FIGURE 8.1-3. SWEEP STRIP LOAD CALCULATION

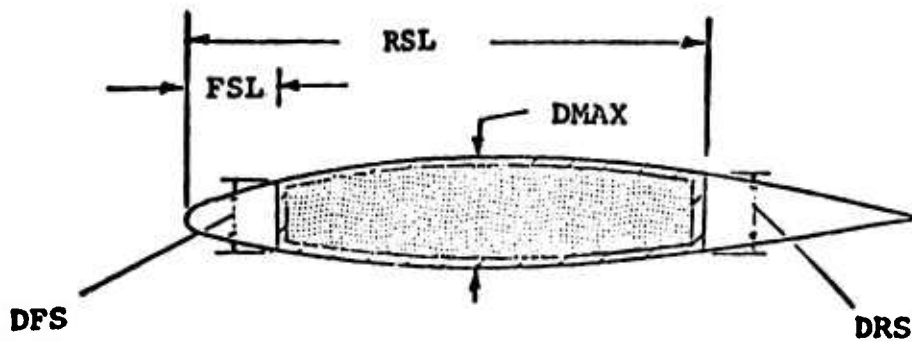


FIGURE 8.1-4. WING BOX SIZE DIMENSIONS

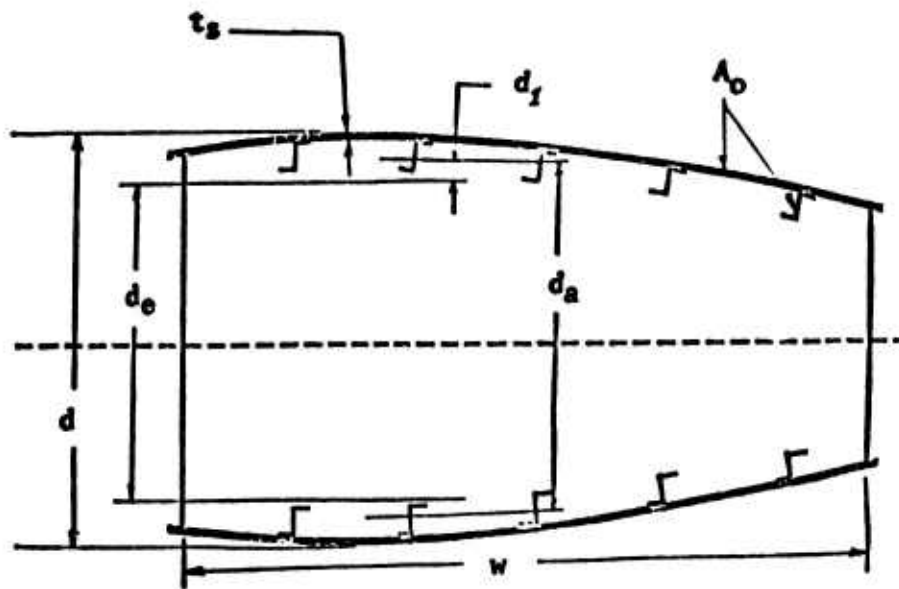


FIGURE 8.1-5. WING BOX SECTION GEOMETRY

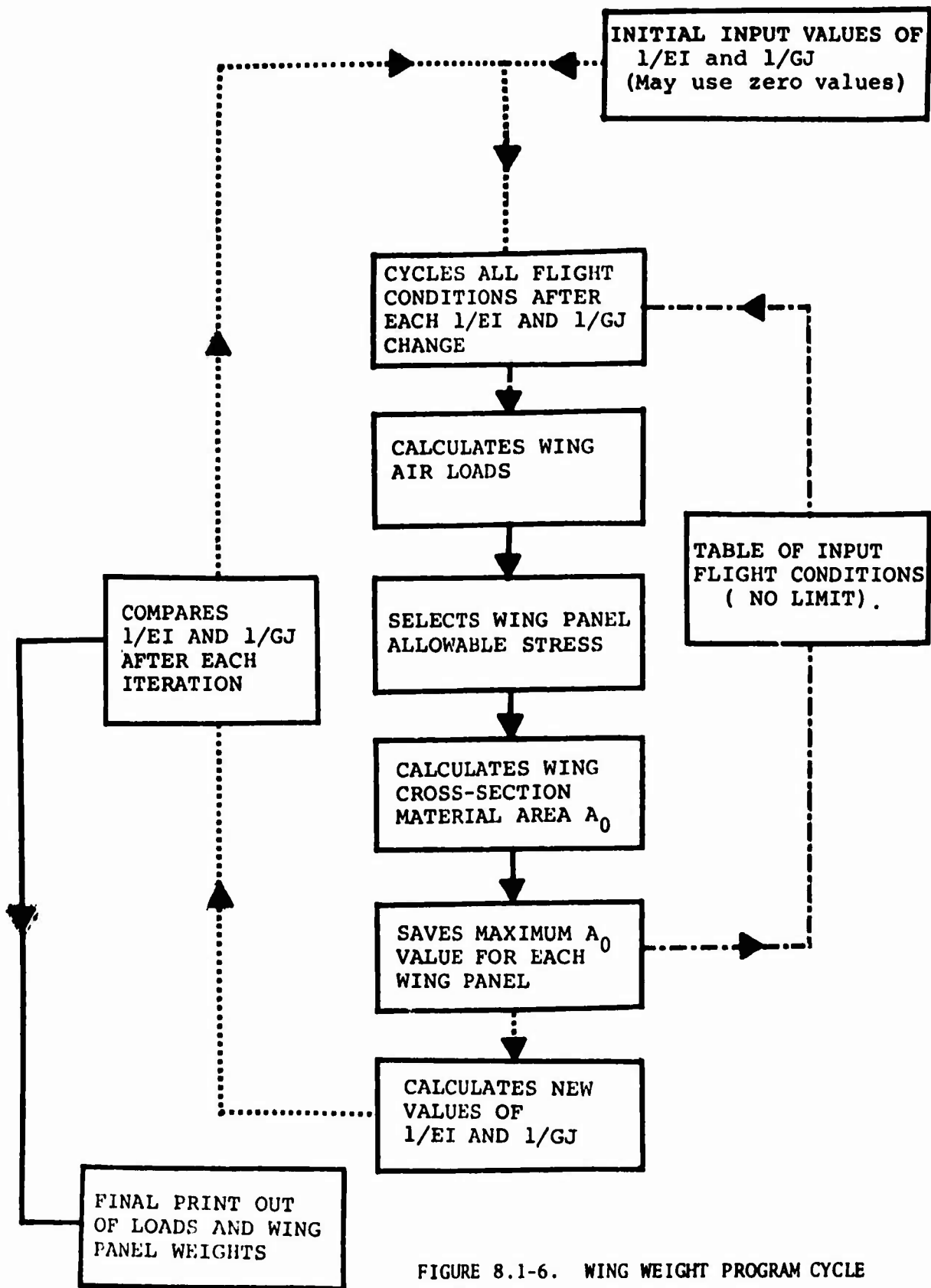


FIGURE 8.1-6. WING WEIGHT PROGRAM CYCLE

SECTION 9

ECONOMICS

Two cost estimation models are available in the ODIN/MFV 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/MFV 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 was constructed by Aerophysics Research Corporation during the ODIN study effort.
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 and is the model summarized below.

REFERENCES:

1. Boren, H.E., Jr., DAPCA: A Computer Program for Determining Aircraft Development and Production Costs, The Rand Corporation, Memorandum RM-5221-PR, February 1967.
2. Wilcox, Darrell E., A Model for Estimating Total Program Cost of Aircraft, Spacecraft, and Reusable Launch Vehicles, NASA OART, Advanced Concepts and Missions Division, Working Paper MA-71-3, August 1971.

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.

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 (Reference 2) 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 3 to 6, 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 7, although conformance is not exact because the cost model accommodates aircraft data which have not been reported to this WBX. 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 8, 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 10 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 establishment 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 cost 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 3, 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. Body Structure. 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.

| | |
|---|-----------------------|
| $CADAPT = 1730 (WADAPT) \cdot 678 (CFADAP)$ | , Adapters |
| $CFWD = 1730 (WFWD) \cdot 678 (CFFWD)$ | , Forward skirts |
| $CAFT = 1730 (WAFT) \cdot 678 (CFAFT)$ | , Aft skirts |
| $CINTK = 1730 (WINTK) \cdot 678 (CFINTK)$ | , Intertank structure |
| $CTHRST = 3400 (WTHRST) \cdot 678 (CFTHRS)$ | , Thrust structure |
| $COTANK = 7400 (WOTANK) \cdot 565 (CFOXTK)$ | , Oxidizer tank |
| $CFTANK = 7400 (WFTANK) \cdot 565 (CFFUTK)$ | , Hydrogen fuel tank |
| $CFTANK = 5770 (WFTANK) \cdot 565 (CFFUTK)$ | , Storable fuel tank |
| $CNOSE = 1730 (WNOSE) \cdot 678 (CFNOSE)$ | , Nose structure |

- b. For spacecraft

| | |
|---|---------------------|
| $CCOMPT = 20130 (WCOMPT) \cdot 631 (CFCOMPT)$ | , Crew compartment |
| and | |
| $CSERV = 8800 (WSERV) \cdot 631 (CFSERV)$ | , Cargo compartment |
| $CADAPT = 2100 (WADAPT) \cdot 631 (CFADAP)$ | , Adapter |

- c. For aircraft

| | |
|--|------------------|
| $CBODY = 56100 (WBODY) \cdot 451 (CFBODY)$ | , Body structure |
|--|------------------|

2. Aerodynamic Surfaces.

| | |
|--|---|
| $CWING = 36000 (WWING) \cdot 451 (CFWING)$ | , Aircraft wing |
| $CEMP = 10230 (WEMP) \cdot 451 (CFEMP)$ | , Aircraft empennage |
| $CFAIR = 1730 (WFAIR) \cdot 678 (CFFAIR)$ | , 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

$CDPLOY = 1340 (WDPLY)^{.766} (CFDPLY)$, Deployable recovery gear

$CRECOV = 42100 (WRECOV)^{.7064} (CFRRCV)$, 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

$CENG. = [35000 + 450 (T)^{.8}] (EN)^{zetap}$, 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 = 6650 (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 propel-
lant utilization
system

CLUBE = 59000 (WLUBE)^{.43} CFLUBE)

, space vehicle lubri-
cating 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 = 5660 (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 (WGNV)^{.485} (CFGNAV) , spacecraft

CGNAV = 22200 (WGNV)^{.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 3 to 6, 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.

$$DDEL = 3145 (WA)^{.5825} (RE) (CONFIG) \quad , \text{ subsonic prototypes}$$

$$DDEL = 23.85 (WA)^{.5825} (VMAX)^{.771} (RE) (CONFIG) \quad , \text{ advanced aircraft}$$

$$DDEL = 207 (WA)^{.931} (RE) (CONFIG) \quad , \text{ subsonic production aircraft}$$

$$DDEL = 348 (WA)^{.931} (RE) (CONFIG) \quad , \text{ Mach 2 aircraft}$$

$$DDEL = 115000 (WA)^{.509} (RE) (CONFIG) \quad , \text{ spacecraft}$$

, expendable VTO
launch vehicles

$$DDEL = [1930000 (WA)^{.484} + 16.65 (NENG)^{.26} (TPEREN)^{.14}] (CONFIG)$$

4. Subsystem development.

$$\text{SUBSYS} = 35000 (\text{WACS} + \text{WECS} + \text{WPOWER} + \text{WPOWCD} + \text{WDPLOY} + \text{WCANDP} + \text{WAERO} + \text{WRECOV} + \text{WPPROV}) (\text{XNEW})$$

5. Avionics development.

$$\text{AD} = [5.3 \times 10^6 (\text{WGNV})^{.439} + 2.19 \times 10^6 (\text{WCOMM})^{.439} + 0.55 \times 10^6 (\text{WINST})^{.439}] (\text{XAVD})$$

6. Propulsion development.

$$\text{PDTJ} = 29.5 \times 10^6 \left(\frac{T}{1000}\right)^{.55} (\text{MACH})^{.66} [(\text{NV} + \text{NFV}) (\text{EN}) (1. + \text{ENSPAR})]^{.1}$$

, turbine engines

$$\text{PDCSJ} = 204 \times 10^6 (\text{ASJMOD})^{.41}$$

, ramjet and scramjet

a. For liquid rocket engines the following equations are used.

For regenerative cooled, pump fed, oxygen/hydrogen engines,

$$\text{PDROCK} = 50 \times 10^6 + 1.405 \times 10^6 (T)^{.422};$$

for regenerative cooled, pump fed, storable propellant engines,

$$\text{PDROCK} = 50 \times 10^6 + 8.65 \times 10^5 (T)^{.422};$$

for ablative cooled, pressure fed, storable propellant engines,

$$\text{PDROCK} = 10 \times 10^6 + 8.4 \times 10^4 (T)^{.678}; \text{ and}$$

for radiation cooled, pressure fed, storable propellant engines,

$$\text{PDROCK} = 5 \times 10^6 + 4.86 \times 10^4 (T)^{.678}.$$

7. Flight test hardware.

$$\text{FV} = (\text{AV} + \text{AMFG}) (\text{NFV})^{\text{ZETA}} + (\text{PROPU}) (\text{NFV})^{\text{ZETAP}}$$

$$\text{ZETA} = 1 + \frac{\ln [.01 (\text{LEARN})]}{\ln 2}$$

and

$$\text{ZETAP} = 1 + \frac{\ln [.01 (\text{LEARNP})]}{\ln 2}$$

$$\text{FS} = .20 (\text{FV})$$

8. Ground test hardware.

$$GTS = .10 (GTV)$$

9. Tooling and special test equipment.

$$\begin{aligned} TST &= 6.19 (RT) (WA)^{1.062} (TOOLC) && , \text{conventional aluminum} \\ TST &= 45.0 (RT) (WA)^{1.062} (TOOLC) && , \text{VTO launch vehicles} \\ TST &= 0.0267 (RT) (WA)^{.99} (VMAX)^{1.21} && , \text{advanced aircraft} \\ & \quad (RATE)^{.4} (TOOLC) (NFV)^{.14} \end{aligned}$$

10. Flight test operations.

$$\begin{aligned} FTO &= 0.75 (NFV)^{1.1} (WG)^{.08} (VMAX)^{0.9} && , \text{typical aircraft} \\ & && \text{certification} \\ & && \text{flight test} \\ FTO &= (ROPF) (NFTEST)^{ZETA} && , \text{other flight test} \\ & && \text{programs} \end{aligned}$$

11. Ground support equipment.

$$AGEP = .05 (ADDE) + .15 (FV)$$

12. Technical data.

$$TDP = .02 (FV)$$

13. Basic RDT&E cost.

$$\begin{aligned} RDTE &= CF + CD + DDEL + AD + PD + SUBSYS + FV + FS + GTV + GTS + \\ & \quad TST + FTO + AGEP + TDTP \end{aligned}$$

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 trailing, 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.

$$\begin{aligned} \text{CAF} &= (\text{AMFG}) [(\text{NVHF})^{\text{ZETA}} - (\text{NFV})^{\text{ZETA}}] && , \text{ airframe} \\ \text{AVO} &= (\text{AV}) [(\text{NVHF})^{\text{ZETA}} - (\text{NFV})^{\text{ZETA}}] && , \text{ avionics} \\ \text{PO} &= (\text{PROPU}) [(\text{NVHF})^{\text{ZETAP}} - (\text{NFV})^{\text{ZETAP}}] && , \text{ engines} \\ \text{CVO} &= \text{CAF} + \text{AVO} + \text{PO} && , \text{ total} \end{aligned}$$

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.

$$\text{AGEO} = 0.15 (\text{OV})$$

3. Spares.

$$\text{OS} = 0.13 (\text{OV})$$

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.

$$\text{SE} = (\text{DDEL}) [(\text{NV})^{.20} - 1]$$

6. Sustaining tooling.

$$\text{ST} = (\text{TST}) \left[\left(\frac{\text{NV}}{\text{NFV}} \right)^{.14} - 1 \right]$$

7. Miscellaneous equipment.

$$\text{MEQ} = 500 \text{ (NPER)}$$

8. Training equipment.

$$\text{OT} = .1442 \text{ (OV) (NV)}^{-.4525} \quad , \text{ aircraft}$$

$$\text{OT} = 0.2088 \text{ (CSTRUC)}^{1.3822} \quad , \text{ spacecraft}$$

9. Initial training.

$$\text{IT} = 50000 \text{ (NPL)}$$

10. Initial transportation.

$$\text{TRI} = 0.005 \text{ (OV + OS + MEQ + OT + AGEQ)}$$

11. Basic acquisition.

$$\text{IV} = \text{OV} + \text{AGEQ} + \text{OS} + \text{FAC} + \text{SE} + \text{ST} + \text{MEQ} + \text{OT} + \text{IT} + \text{TRI}$$

12. Fee.

$$\text{AQFEE} = \text{(IV) (FEE)}$$

13. Project management.

$$\text{AQMGMT} = 0.01 \text{ (IV)}$$

14. Total acquisition cost.

$$\text{AQ} = \text{IV} + \text{AQFEE} + \text{AQMGMT}$$

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.

$$\begin{aligned} \text{NPL} &= 2 \text{ (NCREW) (NV)} && \text{, flight crew number} \\ \text{NMTPS} &= (\text{SURF}) (\text{HRPFT2}) (\text{XREPL}) (\text{LPM}) / 160 && \text{, TPS personnel} \\ \text{NMPR} &= \text{NMAINT} + \text{NMTPS} && \text{, total support personnel} \\ \text{WSA} &= 500,000 \text{ (NPL)} + 270,000 \text{ (NSPR)} && \text{, total 10 year support} \\ &&& \text{and crew cost} \\ \text{CLO} &= 58700 (\text{WDY}) \begin{matrix} .524 \\ [12(\text{LPM})] \end{matrix} \begin{matrix} -.891 \\ [120(\text{LPM})] \end{matrix} && \text{, launch vehicle 10 year} \\ &&& \text{cost} \\ \text{CLSUP} &= 0.598 \text{ (CLO)} \begin{matrix} 1.204 \end{matrix} && \text{, launch support 10 year} \\ &&& \text{cost} \\ \text{WSA} &= \text{CLO} + \text{CLSUP} && \text{, total 10 year launch} \\ &&& \text{vehicle cost} \end{aligned}$$

2. Vehicle maintenance.

$$\begin{aligned} \text{VMTPS} &= (\text{XMTPS}) (\text{CTPS}) \begin{matrix} \text{ZETA} \\ [120(\text{LPM}) - \text{NV}] \end{matrix} && \text{, TPS maintenance} \\ \text{VM} &= \text{VMTPS} + 270,000 \text{ (NMPR)} && \text{, total maintenance} \end{aligned}$$

or

$$\text{VM} = 120 \text{ (LPM) (XMRA) (CSTRUC)}$$

3. Vehicle retrieval.

$$\text{WTR} = 1.98 \times 10^7 \text{ (WE/1000)} \begin{matrix} .585 \\ (\text{LPM}) \end{matrix} && \text{, water retrieval}$$

4. Propellants.

$$\text{PF} = 120 \text{ (LPM) (CFUEL + COXID + CAUXP + CGASPR)}$$

5. Miscellaneous expendables.

$$\text{MFL} = 120 \text{ (LPM) (EXPEND)}$$

6. Facilities maintenance.

$$\text{FM} = 0.4 \text{ (FAC)}$$

7. GSE maintenance.

$$\text{AGEM} = 0.3 \text{ (AGEO)}$$

8. Miscellaneous equipment maintenance.

$$\text{MFM} = 550 \text{ (NPER)}$$

9. Training.

$$TO = 400,000 \text{ (NPL)}$$

10. Transportation.

$$TRO = 0.15 \text{ (PF + MFL + FM + AGEM + MEM)}$$

11. Total operations.

$$RO = WSA + VM + WTR + PF + MFL + FM + AGEM + MEM + TO + TRO$$

12. Fee.

$$ROFEE = 0.5 \text{ (RO) (FEE)}$$

13. Project management.

$$ROMGMT = 0.057 \text{ (RO)}$$

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:

$$\text{TOTAL} = \text{TRDTE} + \text{AQ} + \text{ROTOT} \quad , \text{ total cost of program}$$

9.1.8 Conclusion

It is again emphasized that the above description merely summarizes program PRICE; for complete details refer to Reference 1. 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-Estimating Relationships for Aircraft Airframes," Rand Corp. Memo RM-4845-PR, Feb. 1966.
2. Anon., "Methods of Estimating Fixed-Wing Airframe Costs," Planning Research Corporation, Report R-547A, April 1967.
3. Anon., "Optimized Cost Performance Design Methodology," McDonnell Douglas Corporation Report G975, April 1969.
4. 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.
5. Knapp, M. I., "Costs of Reusable and Expendable Spacecraft Systems," Institute for Defense Analysis Research Paper P-426, June 1969.
6. Campbell, H. G., and Dreyfuss, D. J., "Manned Spacecraft Cost Estimating Relationships," Rand Corporation Memo RM-5317-NASA, June 1967.
7. NASA, Office of Advanced Manned Missions, "Specification for Contractor Presentation of Cost and Schedule Plans for New Space Projects," Feb. 17, 1969.
8. Anon. "Mass Properties Control Requirements for Missile and Space Vehicles," USAF Military Specification MIL-M-38310, 15 Dec. 1964.
9. 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.
10. Wilcox, D. E., "A Computer Program for Estimating Total Program Cost of Aircraft, Spacecraft, and Reusable Launch Vehicles," Working Paper MA-71-2, August 2, 1971.
11. Anon., "Influence of Structure and Material Research on Advanced Launch Systems," NASA Contractor Report CR-1116, July 1968, North American Rockwell Corp.
12. Morris, R. E., and Williams, N. B., "Study of Advanced Air Breathing Launch Vehicles with Cruise Capability," Vol. 5, NASA CR-73198, Lockheed California Company, Feb. 1968.
13. Anon., "Integral Launch and Reentry Vehicle," General Dynamics/Convair, First Monthly Progress Report, Contract NAS 9-9207, April, 1969.
14. Large, J. P., "Estimating Aircraft Turbine Engine Costs," Rand Corporation Memo RM-6384/1-PR, September 1970.

15. Anon., "Study of Navigation and Guidance of Launch Vehicles Having Cruise Capability," Final Report Contract NAS 2-3691, The Boeing Company, April 10, 1967.
16. Anon., "Gas Turbine Engine Market 1970-1974," DMS Inc., April 1970.
17. Anon., "Launch Vehicle Component Costs Study," Lockheed Missiles and Space Co., Report 895429, Contract NAS 8-11368, June 1965
18. Johnson, D. R., 'Design Considerations of Reusable Launch Vehicles," Vol. VI, Final Report, Contract NAS 2-3191, Douglas Aircraft Company, Oct. 1966.
19. Personal communication with Mr. J. Henry, NASA Langley Research Center.
20. Schlosser, K. P., "Cargo Aircraft Cost Model," Directorate of Operations Research, WPAFB, 2 Nov. 1965.
21. Anon., "Aircraft Systems Cost Factors," Comptroller Office, Aeronautical Systems Division, WPAFB, Dec. 1969.
22. Anon., "Supersonic Transport Development and Production Cost Analysis Program," Booz-Allen Applied Research Inc., Final Report, Contract FA-SS-66-13, Dec. 1966.
23. Anon., "USAF Cost and Planning Factors," Air Force Manual 172-3 (CONFIDENTIAL)
24. Woolsey, J. P., "American Seeks Flight Training Reductions," Aviation Week and Space Technology, July 13, 1970, p. 41
25. Anon., "Optimized Cost Performance Design Methodology," Final Oral Briefing Contract NAS 2-5022, 2 Feb. 1970.
26. Anon., "Supersonic Transport Development Program, Phase III Proposal," The Boeing Company, Report V6-B2707-2, 6 Sept. 1966.
27. Anon., "Standard Method of Estimating Comparative Direct Operating Costs of Turbine Powered Transport Airplanes," Air Transport Association of America, Dec. 1967.
28. Raymond, A. P., "Cost Analysis of Current Launch Systems," Vol. IV, Final Report, Contract NAS w-1740, Chrysler Corp. 15 Feb. 1969.

Table 9.1-1.
Type of Material and Construction Complexity Factors

| Type Material | Type Construction 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 areas on the vehicle. Ten values to 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. |
| WDFLOY | 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. |
| WENG5 | Weight of all main engines and accessories, except scramjets | lb. |
| WENG52 | Weight of all secondary and tertiary engines and accessories | lb. |
| WEQEC5 | 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. |
| WPROV | 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 ₂ O ₄ /N ₂ H 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 manu- facturing 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 | |
| NOENGL | 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 | |
| PDEN21* | 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. | |
| CFACK* | 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. | |

* 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 |
| AQMGMGT | 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=29700., MWING=13260., WPRESS=15260.,
 WLG=4800., WDPLOY=0., WDOCK=550., WBLANCH=0., WRECOV=0., WENGS=6875.,
 WENG2=3692., WINLET=0., WNACEL=1122., WFUTK=12890., WXTK=0.,
 WFUTK2=1727., WXTK2=586., WINSTK=602., WFUSYS=1237., WXSYS=1383.,
 WPRSYS=4405., WPUSYS=323., WLUBE=0., WAERO=1095., WAUT=0.,
 WACS=2573., WACSTK=0., WPOWER=649., WELCAD=1465., WHYCAD=1644.,
 WGNV=790., WINST=885., WCUMM=165., WEQCS=0.,
 WPCS=1123., WINCOM=0., WPRUV=200., WCANDP=460.,
 WABORT=0., WFUTOT=47200., WXTOT=282600., WAUXP=7012., WGASPR=1170.,
 WGROSS=439971., WDRY=92662., WPAYL=15400.,
 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., PUDOCI=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, NTEST=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., 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. SAMPLE CASE

MINI SHUTTLE ORBITER 002 (3-27-70)
 ICLASS=5, NENG=2., TPEREN=297000., WMING=13260., WPRESS=15260.,
 WLG=4800., WDPLOY=0., WDOCK=550., WANCH=0., WRECOV=0., WENGS=6875.,
 WENGS2=3692., WINLET=0., WNACEL=1122., WFUTK=12890., WDXTK=0.,
 WFUTK2=1727., WDXTK2=586., WINSTK=602., WFUSYS=1237., WOXSYS=1383.,
 WPRYS=4405., WPUSYS=323., WLUBE=0., WAERO=1095., WAUXT=0.,
 WACS=2573., WACSTK=0., WPOWER=649., WELCAD=1465., WHYCAD=1644.,
 WGNV=790., WINST=885., WCOMM=165., WEQCS=0.,
 WPECS=1123., WINCOM=0., WPPROV=200., WCANDP=460.,
 WABORT=0., WFUTOT=47200., WDXTOT=282600., WAUXP=7012., WGASPR=1170.,
 WGRSS=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, ICONF=5, TOOLC=7.2,
 IPOWER=0, XFASSY=.20, ITANK2=2, PDEN2I=121., HRPFT2=16.,
 XREPL=.054, NTEST=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., 200., 100., 4X0.,
 SPANEL=100., 100., 4., 7X100.,
 *
 CFWING=2.2, CFCOMP=2.8, CFTK=1.5, *

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 DOLLARS) | |
|---|----------------------------|-------------|
| | 0.24003E 04 | 0.68896E 03 |
| RESEARCH, DEVELOPMENT, TEST, AND EVALUATION | | |
| AIRFRAME DESIGN AND DEVELOPMENT ENGINEERING | | |
| CONCEPT FORMULATION | 0.11812E 02 | |
| CONTRACT DEFINITION | 0.35000E 02 | |
| AIRFRAME ENGINEERING | 0.64215E 03 | |
| MISCELLANEOUS SUBSYSTEM DEVELOPMENT | | |
| AVIONICS DEVELOPMENT | | 0.32231E 03 |
| PROPULSION DEVELOPMENT | | 0.13058E 03 |
| DEVELOPMENT SUPPORT | | 0.12100E 03 |
| (MANUFACTURING--FIRST UNIT) | | 0.73738E 03 |
| AIRFRAME | (0.10935E 03) | |
| AVIONICS PROCUREMENT | (0.90200E 02) | |
| PROPULSION PROCUREMENT | (0.84558E 01) | |
| FLIGHT VEHICLES(1.0) | (0.10694E 02) | |
| GROUND TEST VEHICLES(1.0) | 0.10935E 03 | |
| GROUND TEST SPARES | 0.90200E 02 | |
| FLIGHT TEST SPARES | 0.90200E 01 | |
| TOOLING AND SPECIAL TEST EQUIPMENT | 0.21871E 02 | |
| FLIGHT TEST OPERATIONS | 0.96444E 02 | |
| AGE | 0.35745E 03 | |
| TECHNICAL DATA | 0.50851E 02 | |
| | 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 | | 0.72017E 03 |
| OPERATING(10 YEAR) | | 0.35382E 03 |
| WAGES, SALARIES, ALLOWANCES | | 0.24974E 03 |
| VEHICLE MAINTENANCE | | 0.10031E 02 |
| PROPELLANTS | | 0.0 |
| MISCELLANEOUS EXPENDABLES | | 0.0 |
| | 24 PILOTS | 488 MAINT. |
| | 1265 SUPPORT PERS) | |

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 9.1.1-1. Total Program Cost Element Structure

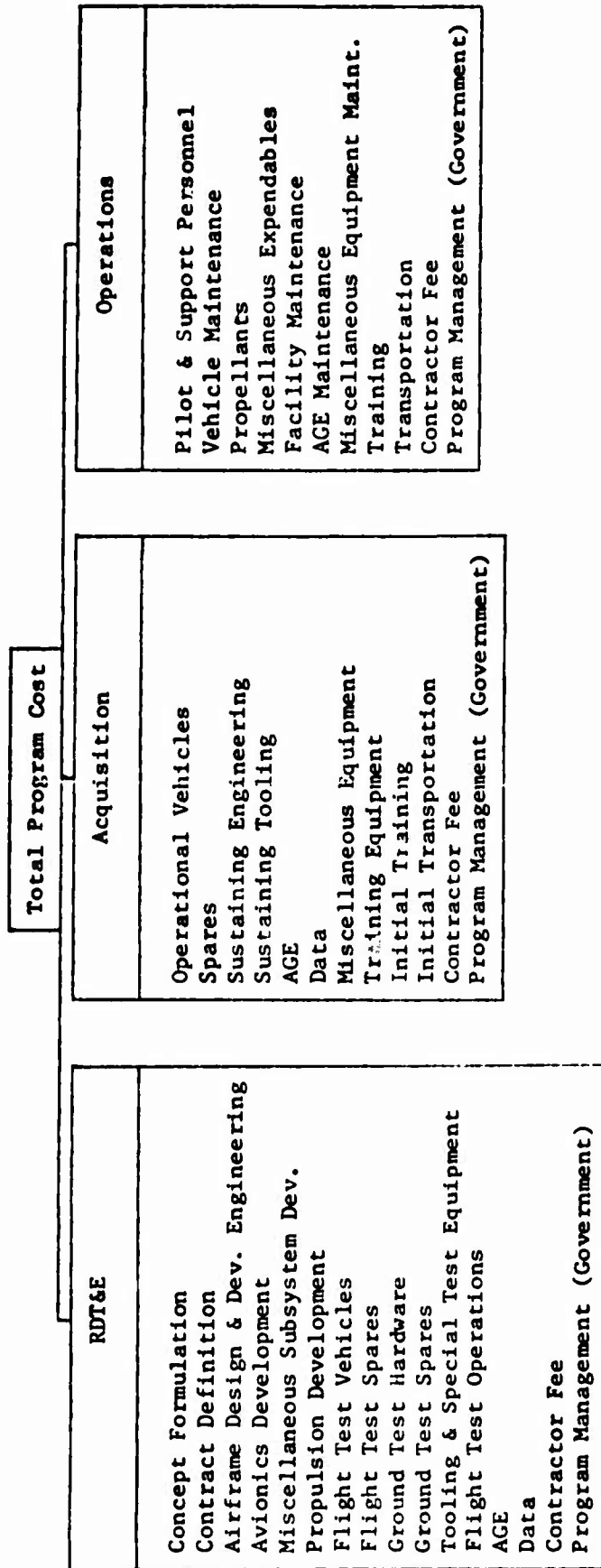


Figure 9.1-2. First Unit Manufacturing Cost Element Structure

First Unit Manufacturing Cost

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

Abort System

Final Assembly & Checkout

Figure 9.1-3. Body Structure Cost Elements

| <u>VTO Launch Vehicle</u> | <u>Spacecraft</u> | <u>Aircraft</u> |
|---------------------------|-------------------|-----------------|
| 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 | | |

EFFECT OF LEARNING RATE

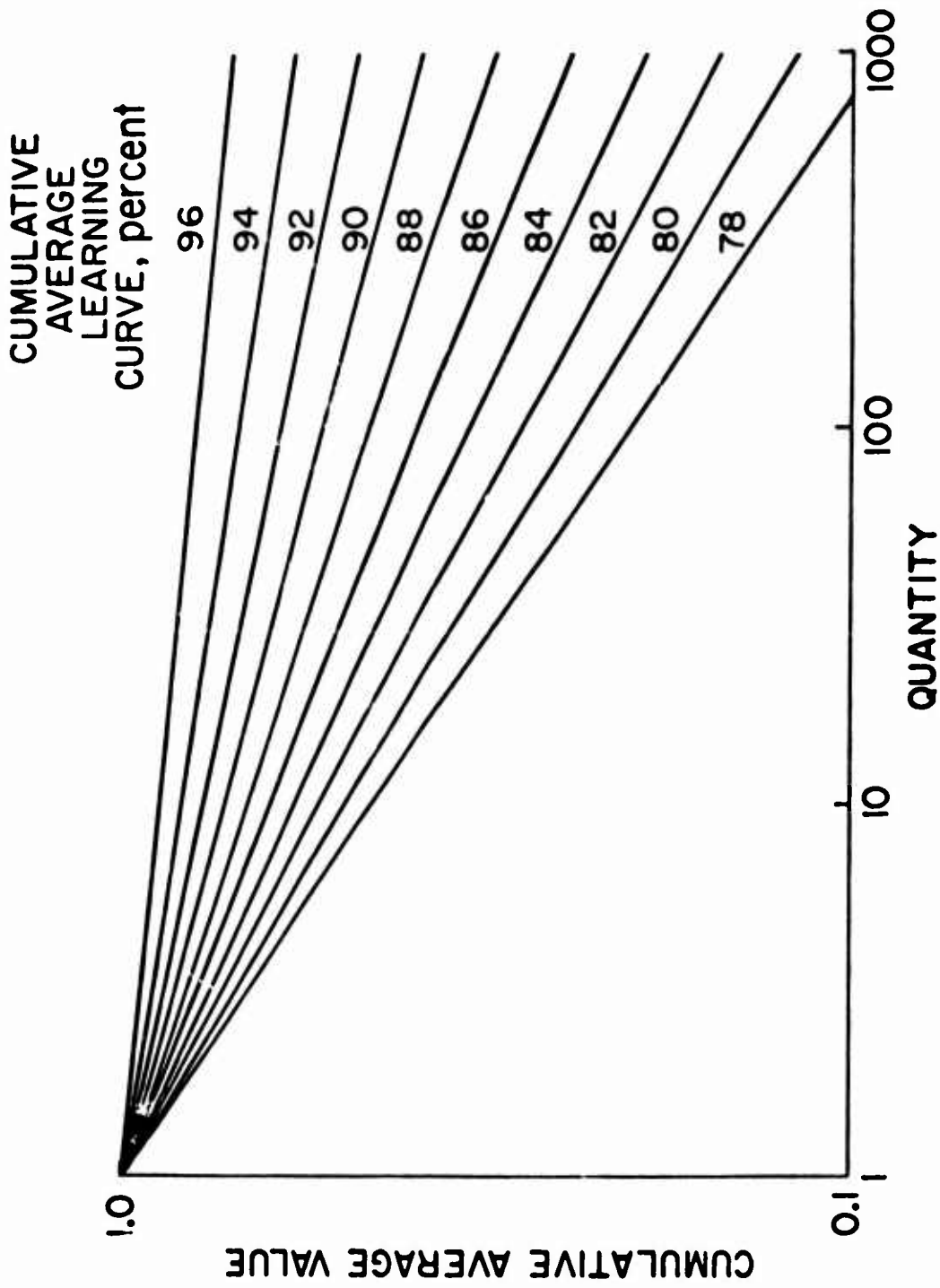


Figure 9.1-4. Effect of Learning Rate

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/MFV and also as an integral part of the ATOP II program of section 7.3.

REFERENCES:

1. Hague, D. S., Three Degree of Freedom Problem Optimization Formulation, Part 1, Volume III--Analytical Development, McDonnell-Douglas Corporation October 1964.
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.
4. Hague, D. S. and Glatt, C. R., A Guide to the Automated Engineering and Scientific Optimization Program--AESOP, NASA CR-73201, April 1968.
5. 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.

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

$$\begin{aligned} \dot{x}_n(t) &= f(x_n(t), \alpha_m(t), t) \\ n &= 1, 2 \dots \dots \dots N \\ m &= 1, 2 \dots \dots \dots M \end{aligned} \quad (1)$$

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

$$\begin{aligned} \psi_p &= \psi_p(x_n(T), T) = 0 \\ p &= 1, 2 \dots \dots \dots P \end{aligned} \quad (2)$$

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

$$\Omega = \Omega(x_n(T), T) = 0 \quad (3)$$

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

$$\phi = \phi(x_n(T), T) \quad (4)$$

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,

$$DP^2 = \int_{t_0}^T \left[\delta \alpha(t) \right] \left[W(t) \right] \left\{ \delta \alpha(t) \right\} dt \quad (5)$$

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

$$\left\{ \alpha_m(t) \right\}^{\text{nominal}} = \left\{ \bar{\alpha}_m(t) \right\} \quad (6)$$

and

$$\left\{ x_n(t) \right\}^{\text{nominal}} = \left\{ \bar{x}_n(t) \right\} \quad (7)$$

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

$$\left\{ \alpha(t) \right\} = \left\{ \bar{\alpha}(t) \right\} + \left\{ \delta\alpha(t) \right\} \quad (8)$$

and

$$\left\{ x(t) \right\} = \left\{ \bar{x}(t) \right\} + \left\{ \delta x(t) \right\} \quad (9)$$

The nominal state variable and perturbed state variable histories can also be written as

$$\left\{ \bar{x}(t) \right\} = \left\{ x(t_0) \right\} + \int_{t_0}^t \left\{ f(\bar{x}(t), \bar{\alpha}(t), t) \right\} dt \quad (10)$$

$$\left\{ x(t) \right\} = \left\{ x(t_0) \right\} + \int_{t_0}^t \left\{ f(\bar{x} + \delta x, \bar{\alpha} + \delta\alpha, t) \right\} dt \quad (11)$$

Subtracting Equation (10) from Equation (11) and using Taylor's expansion to first order,

$$\left\{ x(t) \right\} - \left\{ \bar{x}(t) \right\} = \int_{t_0}^t \left\{ \frac{\partial \bar{f}}{\partial x_n} \cdot \delta x^n + \frac{\partial \bar{f}}{\partial \alpha_m} \cdot \delta \alpha^m \right\} dt = \left\{ \delta x(t) \right\} \quad (12)$$

where

$$\bar{f} = f(\bar{x}(t), \bar{\alpha}(t), t) \quad (13)$$

and where the repeated index indicates a summation over all possible values. Differentiation leads to

$$\frac{d}{dt} \left\{ \delta x(t) \right\} = \left\{ \frac{\partial \bar{f}}{\partial x_n} \delta x^n + \frac{\partial \bar{f}}{\partial \alpha_m} \delta \alpha^m \right\} \quad (14a)$$

or in matrix form

$$\frac{d}{dt} \{ \delta x(t) \} = [F] \{ \delta x \} + [G] \{ \delta \alpha \} \quad (14b)$$

where

$$F_{ij} = \frac{\partial \bar{f}_i}{\partial x_j} \quad \text{and} \quad G_{ij} = \frac{\partial \bar{f}_i}{\partial \alpha_j} \quad (15)$$

Here the (i,j) th element lies in the i^{th} row and j^{th} column of the matrices; F is an $N \times N$ matrix and G is an $N \times M$ 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

$$[\dot{\lambda}(t)] = -[F(t)]' [\lambda(t)] \quad (16)$$

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

$$[\lambda]' \cdot \left\{ \frac{d}{dt} (\delta x) \right\} + [\dot{\lambda}]' \{ \delta x \} = [\lambda]' [F] \{ \delta x \} + [\lambda]' [G] \{ \delta \alpha \} - [\lambda]' [F] \{ \delta x \} \quad (17)$$

which may be written as

$$\left\{ \frac{d}{dt} (\lambda' \delta x) \right\} = [\lambda]' [G] \{ \delta \alpha \} \quad (18)$$

Integrating Equation (18) over the trajectory

$$\{ \lambda' \delta x \}_T - \{ \lambda' \delta x \}_{t_0} = \int_{t_0}^T [\lambda]' [G] \{ \delta \alpha \} dt \quad (19)$$

Now define three distinct sets of λ functions by applying the following boundary conditions at $t = T$:

$$\{ \lambda(T) \} = \left\{ \frac{\partial \phi}{\partial x_1} \right\}_T = \{ \lambda_{\phi}(T) \} \quad (20a)$$

$$\{ \lambda(T) \} = \left\{ \frac{\partial \Omega}{\partial x_1} \right\}_T = \{ \lambda_{\Omega}(T) \} \quad (20b)$$

$$[\lambda(T)] = \left[\frac{\partial \psi_j}{\partial x_1} \right]_T = [\lambda_{\psi}(T)] \quad (20c)$$

Equation (16) may now be integrated in the reverse direction (i.e., from T to t_0) 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

$$\left[\lambda_\phi(T) \right] \{ \delta x \} = \left[\frac{\partial \phi}{\partial x} \right] \{ \delta x \} = \delta \phi_{t=T} \quad (21a)$$

$$\left[\lambda_\Omega(T) \right] \{ \delta x \} = \left[\frac{\partial \Omega}{\partial x} \right] \{ \delta x \} = \delta \Omega_{t=T} \quad (21b)$$

$$\left[\lambda_\psi(T) \right] \{ \delta x \} = \left[\frac{\partial \psi_i}{\partial x_j} \right] \{ \delta x \} = \{ \delta \psi_{t=T} \} \quad (21c)$$

It follows that

$$\delta \phi_{t=T} = \int_{t_0}^T \left[\lambda_\phi \right] \left[G \right] \{ \delta \alpha \} dt + \left[\lambda_\phi(t_0) \right] \{ \delta x(t_0) \} \quad (22a)$$

$$\delta \Omega_{t=T} = \int_{t_0}^T \left[\lambda_\Omega \right] \left[G \right] \{ \delta \alpha \} dt + \left[\lambda_\Omega(t_0) \right] \{ \delta x(t_0) \} \quad (22b)$$

$$\{ \delta \psi \}_{t=T} = \int_{t_0}^T \left[\lambda_\psi \right] \left[G \right] \{ \delta \alpha \} dt + \left[\lambda_\psi(t_0) \right] \{ \delta x(t_0) \} \quad (22c)$$

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, $T + \Delta T$. In this case, the total change in the above quantities becomes

$$d\phi = \int_{t_0}^T \left[\lambda_\phi \right] \left[G \right] \{ \delta \alpha \} dt + \left[\lambda_\phi(t_0) \right] \{ \delta x(t_0) \} + \dot{\phi}(T) \Delta T \quad (23a)$$

$$d\Omega = \int_{t_0}^T \left[\lambda_\Omega \right] \left[G \right] \{ \delta \alpha \} dt + \left[\lambda_\Omega(t_0) \right] \{ \delta x(t_0) \} + \dot{\Omega}(T) \Delta T \quad (23b)$$

$$\{ d\psi \} = \int_{t_0}^T \left[\lambda_\psi \right] \left[G \right] \{ \delta \alpha \} dt + \left[\lambda_\psi(t_0) \right] \{ \delta x(t_0) \} + \{ \dot{\psi}(T) \} \Delta T \quad (23c)$$

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.

$$\therefore \Delta T = -\frac{1}{\dot{\Omega}(T)} \left(\int_{t_0}^T [\lambda_{\Omega}] [G] \{\delta\alpha\} dt + [\lambda_{\Omega}(t_0)] \{\delta x(t_0)\} \right) \quad (24)$$

Substituting Equation (24) into Equations (23a) and (23c)

$$d\phi = \int_{t_0}^T [\lambda_{\phi\Omega}] [G] \{\delta\alpha\} dt + [\lambda_{\phi\Omega}(t_0)] \{\delta x(t_0)\} \quad (25a)$$

$$\{d\psi\} = \int_{t_0}^T [\lambda_{\psi\Omega}]' [G] \{\delta\alpha\} dt + [\lambda_{\psi\Omega}(t_0)]' \{\delta x(t_0)\} \quad (25b)$$

where

$$\{\lambda_{\phi\Omega}\} = \{\lambda_{\phi}\} - \frac{\dot{\phi}(T)}{\dot{\Omega}(T)} \{\lambda_{\Omega}\} \quad (26a)$$

$$[\lambda_{\psi\Omega}]' = [\lambda_{\psi}]' - \frac{\dot{\psi}(T) \{\lambda_{\Omega}\}}{\dot{\Omega}(T)} \quad (26b)$$

Equations (25) reveal the significance of the λ functions, originally defined by Equations (16) and (20). At time t_0 , $\lambda_{\phi\Omega}$ gives the sensitivity of $\phi(T)$ to small perturbations in the state variables at t_0 . 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

$$\begin{aligned}
 U = & \int_{t_0}^T [\lambda_{\phi\Omega}] [G] \{\delta\alpha\} dt + [\lambda_{\phi\Omega}(t_0)] \{\delta x(t_0)\} \\
 & + [\nu] \left\{ \int_{t_0}^T [\lambda_{\psi\Omega}]' [G] \{\delta\alpha\} dt + [\lambda_{\psi\Omega}(t_0)]' \{\delta x(t_0)\} \right\} \\
 & + \mu \int_{t_0}^T [\delta\alpha] [W] \{\delta\alpha\} dt
 \end{aligned} \tag{27}$$

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

$$\{\delta\alpha\} = \{A(t)\} k, \text{ or } [\delta\alpha] = [A(t)] k \tag{28}$$

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

$$\begin{aligned}
 \bar{U} = & k \int_{t_0}^T [\lambda_{\phi\Omega}] [G] \{A(t)\} dt + k [\nu] \int_{t_0}^T [\lambda_{\psi\Omega}]' [G] \{A(t)\} dt \\
 & + k^2 \mu \int_{t_0}^T [A(t)] [W] \{A(t)\} dt
 \end{aligned} \tag{29}$$

So that

$$\begin{aligned}
 \frac{\partial \bar{U}}{\partial k} = & \int_{t_0}^T [\lambda_{\phi\Omega}] [G] \{A(t)\} dt + [\nu] \int_{t_0}^T [\lambda_{\psi\Omega}]' [G] \{A(t)\} dt \\
 & + 2k\mu \int_{t_0}^T [A(t)] [W] \{A(t)\} dt
 \end{aligned} \tag{30}$$

or

$$\begin{aligned} \delta \bar{U} &= \int_{t_0}^T \left([\lambda_{\phi\Omega}] [G] \{ \delta k \cdot A(t) \} + [\nu] [\lambda_{\psi\Omega}]' [G] \{ \delta k \cdot A(t) \} \right. \\ &\quad \left. + 2\mu [k \cdot A(t)] [W] \{ \delta k \cdot A(t) \} \right) dt \\ &= \int_{t_0}^T \left[[\lambda_{\phi\Omega}] [G] + [\nu] [\lambda_{\psi\Omega}]' [G] + 2\mu [\delta\alpha] [W] \right] \{ \delta(\delta\alpha) \} dt \end{aligned} \quad (31)$$

where it has been noted from Equation (28) that

$$\delta(\delta\alpha) = A(t) \delta k \quad (32)$$

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_0 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_0 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_0 to T . Hence, the coefficient of $\delta(\delta\alpha)$ must be identically zero in the whole interval $t_0 \leq t \leq T$. This argument is essentially based on that presented by Goldstein, Reference 14. It follows that

$$\left[[\lambda_{\phi\Omega}] + [\nu] [\lambda_{\psi\Omega}]' \right] [G] = -2\mu [\delta\alpha] [W] \quad (33)$$

Transposing, noting that W is symmetric, and solving for δa ,

$$\{\delta a\} = -\frac{1}{2\mu} [W]^{-1} [G] \left\{ \lambda_{\phi\Omega} \right\} + [\lambda_{\psi\Omega}] \{ \nu \} \quad (34)$$

Substituting Equation (34) into Equation (25b)

$$\{d\beta\} = -\frac{1}{2\mu} \left\{ I_{\psi\phi} \right\} + [I_{\psi\psi}] \{ \nu \} \quad (35a)$$

where

$$\{d\beta\} = \{d\psi\} - [\lambda_{\psi\Omega}(t_0)]' \{ \delta x(t_0) \} \quad (35b)$$

and

$$[I_{\psi\psi}] = \int_{t_0}^T [\lambda_{\psi\Omega}] [G] [W]^{-1} [G]' [\lambda_{\psi\Omega}] dt \quad (36a)$$

$$\{I_{\psi\phi}\} = \int_{t_0}^T [\lambda_{\psi\Omega}] [G] [W]^{-1} [G]' \{ \lambda_{\phi\Omega} \} dt \quad (36b)$$

For subsequent use define the integral

$$I_{\phi\phi} = \int_{t_0}^T [\lambda_{\phi\Omega}] [G] [W]^{-1} [G]' \{ \lambda_{\phi\Omega} \} dt \quad (36c)$$

The multipliers ν can be expressed in terms of the multipliers μ by Equation (35a)

$$\{ \nu \} = -[I_{\psi\psi}]^{-1} \left\{ 2\mu \{d\beta\} + \{I_{\psi\phi}\} \right\} \quad (37)$$

Substituting Equation (34) into Equation (5)

$$DP^2 = \frac{1}{4\mu^2} \left(I_{\phi\phi} + [I_{\psi\phi}] \{ \nu \} + [\nu] \{ I_{\psi\phi} \} + [\nu] [I_{\psi\psi}] \{ \nu \} \right) \quad (38)$$

Transposing the second term in the right hand side bracket

$$DP^2 = \frac{1}{4\mu^2} \left(I_{\phi\phi} + 2 [\nu] \{ I_{\psi\phi} \} + [\nu] [I_{\psi\psi}] \{ \nu \} \right) \quad (39)$$

Substituting Equation (37) in Equation (39)

and noting that $[I_{\psi\psi}]^{-1}$ is symmetrical gives

$$4\mu^2 DP^2 = I_{\phi\phi} - [I_{\psi\phi}] [I_{\psi\psi}]^{-1} \{I_{\psi\phi}\} + 4\mu^2 [d\beta] [I_{\psi\psi}]^{-1} \{d\beta\} \quad (40)$$

So that

$$2\mu = \pm \sqrt{\frac{I_{\phi\phi} - [I_{\psi\phi}] [I_{\psi\psi}]^{-1} \{I_{\psi\phi}\}}{DP^2 - [d\beta] [I_{\psi\psi}]^{-1} \{d\beta\}}} \quad (41)$$

Substituting Equation (41) into Equation (37), the remaining Lagrangian multipliers are obtained in the form

$$\{\nu\} = -[I_{\psi\psi}]^{-1} \left\{ \{I_{\psi\phi}\} \pm \sqrt{\frac{I_{\phi\phi} - [I_{\psi\phi}] [I_{\psi\psi}]^{-1} \{I_{\psi\phi}\}}{DP^2 - [d\beta] [I_{\psi\psi}]^{-1} \{d\beta\}}} \{d\beta\} \right\} \quad (42)$$

The optimum control perturbation is found by substituting Equations (41) and (42) back into Equation (34) and is

$$\begin{aligned} \{\delta\alpha\} = & \frac{1}{2} [W]^{-1} [G] \left\{ \{\lambda_{\phi\Omega}\} - [\lambda_{\psi\Omega}] [I_{\psi\psi}]^{-1} \{I_{\psi\phi}\} \right\} \\ & \times \sqrt{\frac{DP^2 - [d\beta] [I_{\psi\psi}]^{-1} \{d\beta\}}{I_{\phi\phi} - [I_{\psi\phi}] [I_{\psi\psi}]^{-1} \{I_{\psi\phi}\}}} \\ & + [W]^{-1} [G] [\lambda_{\psi\Omega}] [I_{\psi\psi}]^{-1} \{d\beta\} \end{aligned} \quad (43)$$

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.

$$d\phi = \pm \sqrt{\left(I_{\phi\phi} - [I_{\psi\phi}] [I_{\psi\psi}]^{-1} \{ I_{\psi\phi} \} \right) \left(DC^2 - [d\beta] [I_{\psi\psi}]^{-1} \{ d\beta \} \right) + [I_{\psi\phi}] [I_{\psi\psi}]^{-1} \{ d\beta \} + [\lambda_{\phi\Omega}(t_0)] \{ \delta x(t_0) \}} \quad (44)$$

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

$$DC^2 = \int_{t_0}^T [\delta\alpha(t)] [W(t)] \{ \delta\alpha(t) \} dt + \sum_{s=1}^S v_s^2 \Delta T_s^2 \quad (45)$$

Equations which are analogous to those of (39) and (35a) are obtained

$$\begin{aligned} \mu^2 \cdot DC^2 - (J_{\phi\phi} + I_{\phi\phi}) - 2 \left[[J_{\psi\phi}] + [L_{\psi\phi}] \right] \{ \vartheta \} \\ - [\vartheta] \left[[J_{\psi\psi}] + [L_{\psi\psi}] \right] \{ \vartheta \} = 0 \end{aligned} \quad (46)$$

$$2\mu \{ \delta\Gamma \} + \left\{ \{ J_{\psi\phi} \} + \{ L_{\psi\phi} \} \right\} + \left[[J_{\psi\psi}] + [L_{\psi\psi}] \right] \{ \vartheta \} = 0 \quad (47)$$

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

$$\begin{aligned} \{ \Delta T_s \} = \mp [v_s]^{-1} [B_s^N]^T \left\{ \{ \bar{\lambda}_{\phi\Omega N} \} - [\bar{\lambda}_{\psi\Omega N}] [J_{\psi\psi} + L_{\psi\psi}]^{-1} \{ J_{\psi\phi} + L_{\psi\phi} \} \right\} \\ \times \sqrt{\frac{DC^2 - [\delta\Gamma] [J_{\psi\psi} + L_{\psi\psi}]^{-1} \{ \delta\Gamma \}}{(J_{\phi\phi} + L_{\phi\phi}) - [J_{\psi\phi} + L_{\psi\phi}] [J_{\psi\psi} + L_{\psi\psi}]^{-1} \{ J_{\psi\phi} + L_{\psi\phi} \}}} \\ + [v_s]^{-1} [B_s^N]^T [\bar{\lambda}_{\psi\Omega N}] [J_{\psi\psi} + L_{\psi\psi}]^{-1} \{ \delta\Gamma \} \end{aligned} \quad (48)$$

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., Three-Degree-of-Freedom Problem Optimization Formulation, Part 1, Volume III--Analytical Development, McDonnell-Douglas Corporation, October 1964.
2. Mobley, R. L. and Vorwald, R. R., Three-Degree-of-Freedom Optimization Formulation, Part 2, Volume III--User's Manual, FDL-TDR-64-1, McDonnell-Douglas Corporation, 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, McDonnell-Douglas Corporation, October 1964.
4. Seubert, F. W. and Usher, Newell E., Six Degree of Freedom Flight Path Study Generalized Computer Program, Part 2, User's Manual, 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, edited by Lavi and Vogl, 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. Petersen, L. D., Trajectory Optimization by the Method of Steepest-Descent, Volume I, AFFDL-TR-67-108, April 1968.
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., Glatt, C. R., and Jones, R. T., Integration of Aerospace Vehicle Performance and Design Optimization, AIAA Paper No. 72-948, Presented at AIAA Second Flight Mechanics Conference, September 11-13, 1972.

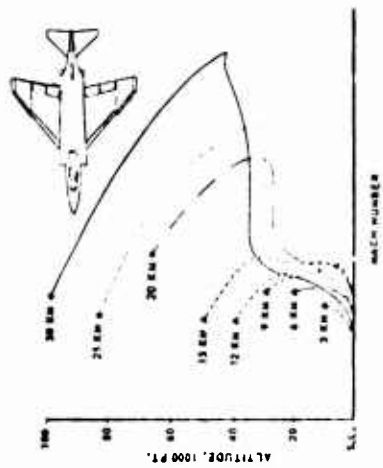


FIGURE 1. F-4B TIME-TO-CLIMB RECORD FLIGHTS

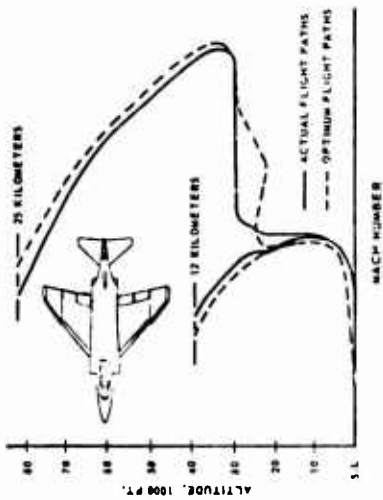


FIGURE 2. COMPARISON OF ACTUAL AND CALCULATED OPTIMUM FLIGHT PATHS FOR TWO F-4B TIME-TO-CLIMB RECORD FLIGHTS

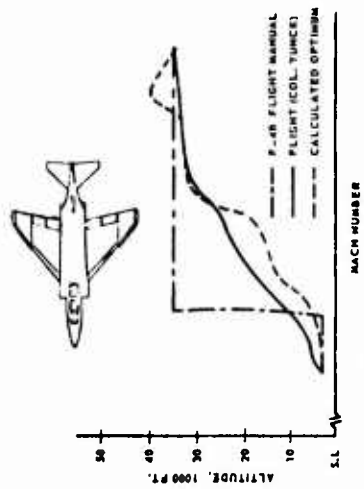


FIGURE 3. FLIGHT PATH COMPARISON

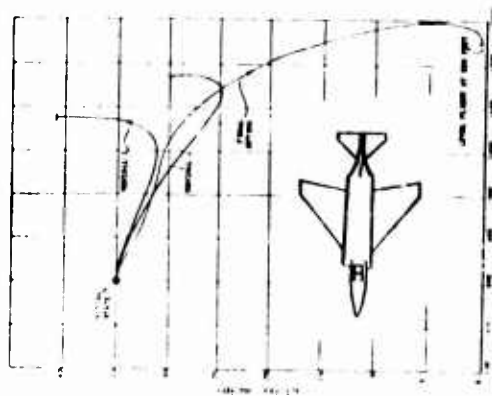


FIGURE 4(a). MAXIMUM VELOCITY DESCENT TO LEVEL FLIGHT

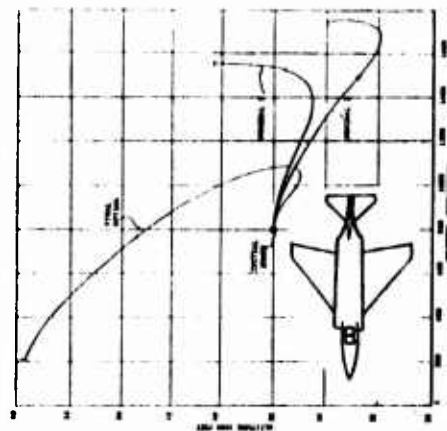


FIGURE 4(b). MAXIMUM LEVEL FLIGHT ALTITUDE IN FIXED TIME

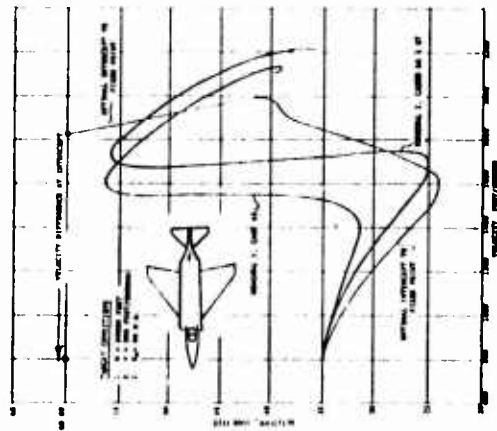


FIGURE 5. MINIMUM TIME (MAXIMUM RANGE) INTERCEPT

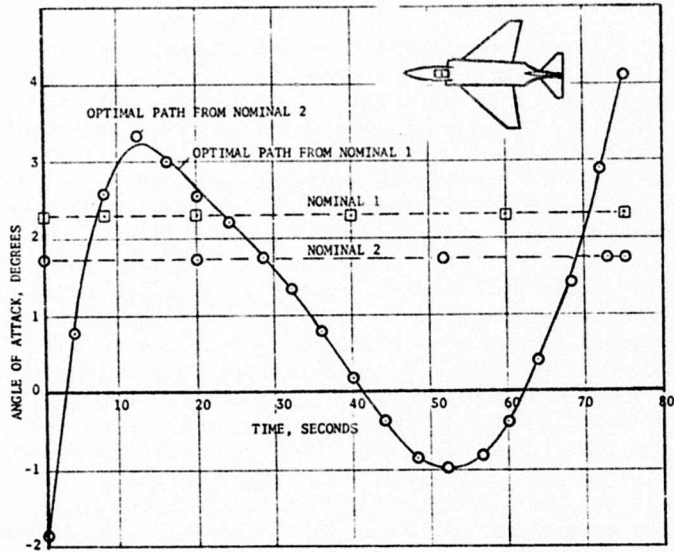
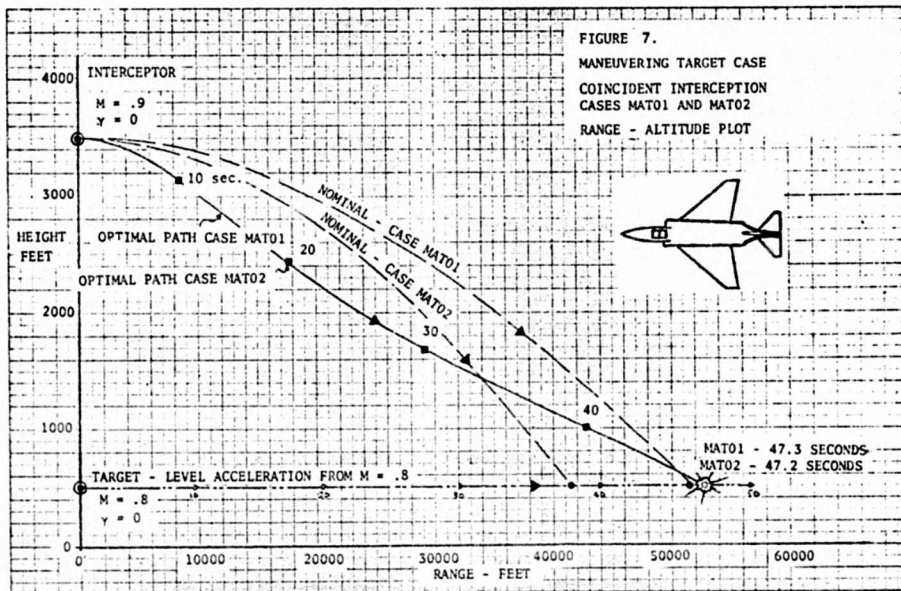


FIGURE 6. MAXIMUM VELOCITY DESCENT CONTROL HISTORY



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:

- . AERODYNAMICS
- . 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 com-

puter aided design techniques such as multivariable optimization. The remainder of this paper discusses these techniques and their 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 multi-variable 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, $\bar{\alpha}$, a unique set of performance characteristics \bar{F} , are obtained.

$$\bar{F} = \bar{F}(\bar{\alpha}) \quad (10.2.1)$$

In the design process one of these characteristics will be selected for minimization or maximization. This is the payoff function

$$\phi = \phi(\bar{\alpha}) \quad (10.2.2),$$

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

$$V = V(\bar{F}) \quad (10.2.3)$$

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, $\bar{\psi}$, are selected from the performance characteristics.

These constraints can always be defined such that

$$\bar{\psi} = \bar{\psi}(\bar{\alpha}) = \bar{0} \quad (10.2.4)$$

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, $\bar{\phi}$, is constructed, in the minimization case

$$\bar{\phi} = \phi + \sum_{i=1}^M W_i \psi_i^2 \quad (10.2.5)$$

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

$$\begin{aligned} \psi_j &= F_i^2; F_i > 0 \\ &= 0; F_i \leq 0 \end{aligned} \quad (10.2.6)$$

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

$$\alpha^L \leq \alpha \leq \alpha^H \quad (10.2.7)$$

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 essence, 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, $\bar{\alpha}_0$, 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, \bar{F} , to the optimizer. Based on inspection of these characteristics, a new design, $\bar{\alpha}$, is supplied to the system, and the process repeats in iterative fashion until the optimal performance $\phi = \phi^*$ for the constraint levels $\bar{\psi} = \bar{U}$ 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.

Sectioning search 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.

Creeping search 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 non-dimensionalization 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 Jones¹¹, Lomax¹², and Heaslett¹² 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 Miele¹⁵.

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 Gellatley¹⁶, Venkayya¹⁷, and others. A summary of much of this activity is provided by the recent Air Force sponsored Conference on Matrix Methods¹⁸.

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 Rozendaal¹⁹. 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 Rutowski²⁰, Bryson²¹, Hague²², and Landgraf²³. 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 program⁷, 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 step-size 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 history⁶. 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. Callatly, 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 Rozendall, 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., "Three-Degree-of-Freedom Problem Optimization Formulation, FDL-TDR-64-1, Part I, Vol. 3, 1964.
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).

FIGURE 10.2-1.

HYPERSONIC TRANSPORT SYNTHESIS

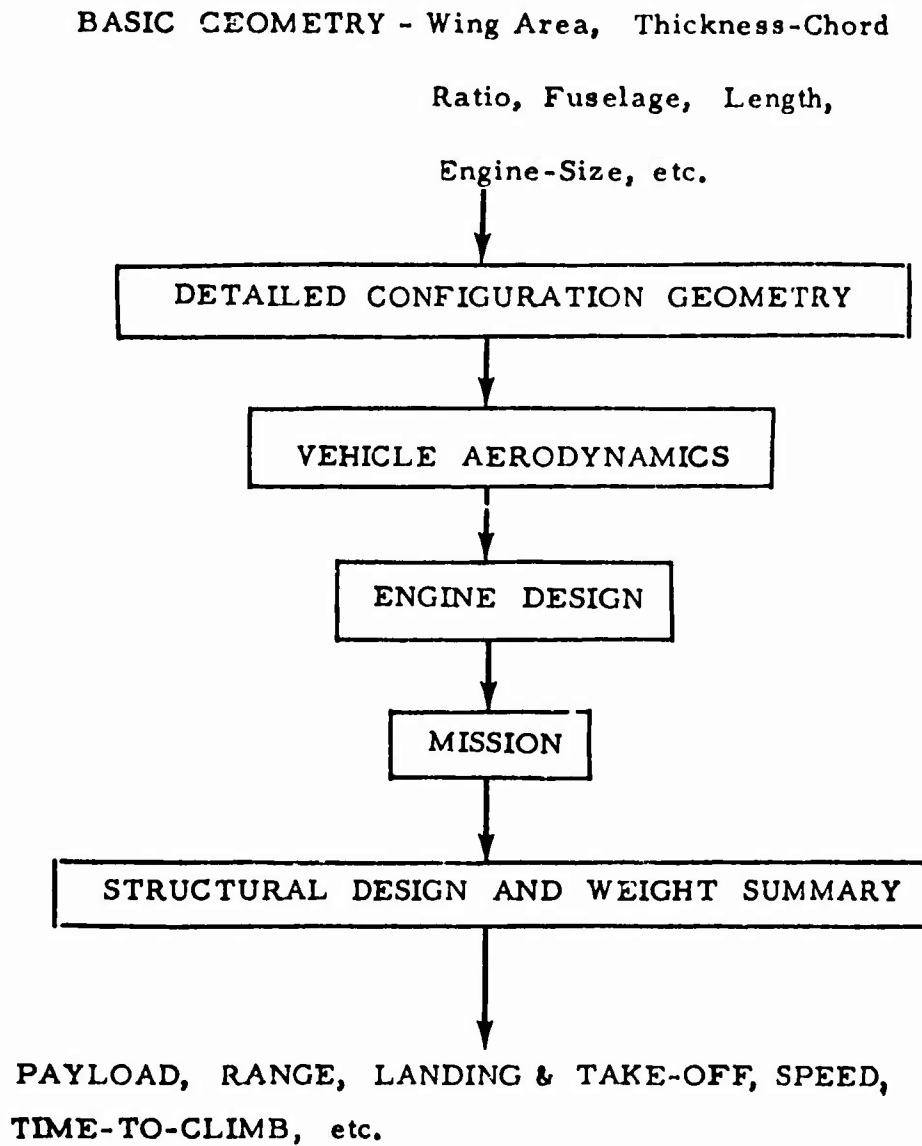


FIGURE 10.2-2.

OPTIMIZER SCHEMATIC

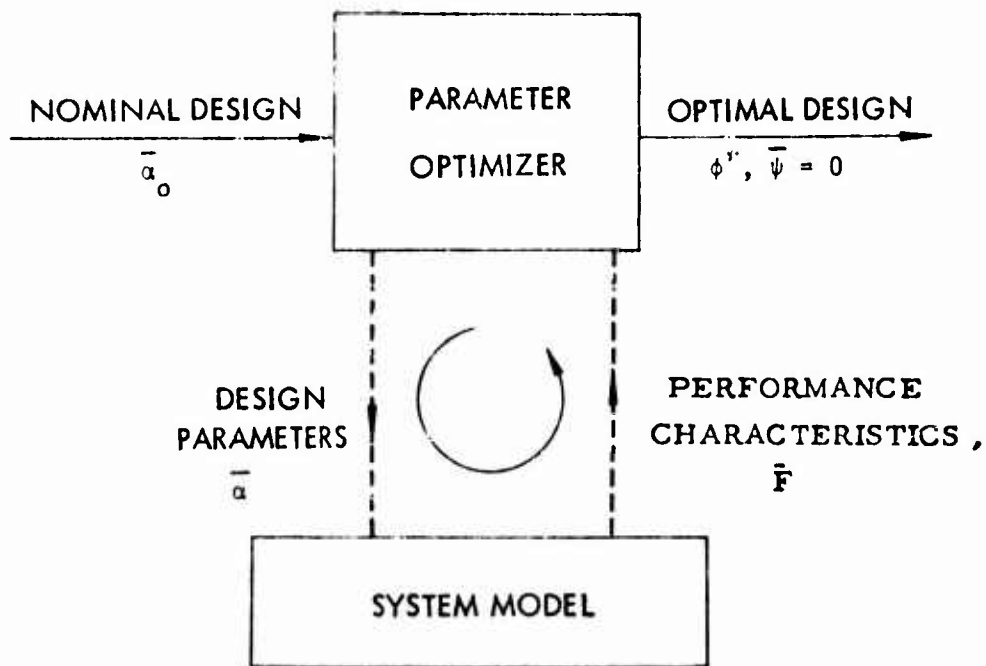


FIGURE 10.2-3.

ENGINEERING PROBLEMS SOLVED BY AESOP

- **AERODYNAMIC AND AEROTHERMODYNAMIC SHAPING**
AIRFOILS
WINGS
BODIES
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 PROCPULSION MODULE DESIGN
STOL VEHICLE DESIGN

FIGURE 10.2-4.

ALGORITHMS FOR OPTIMIZATION

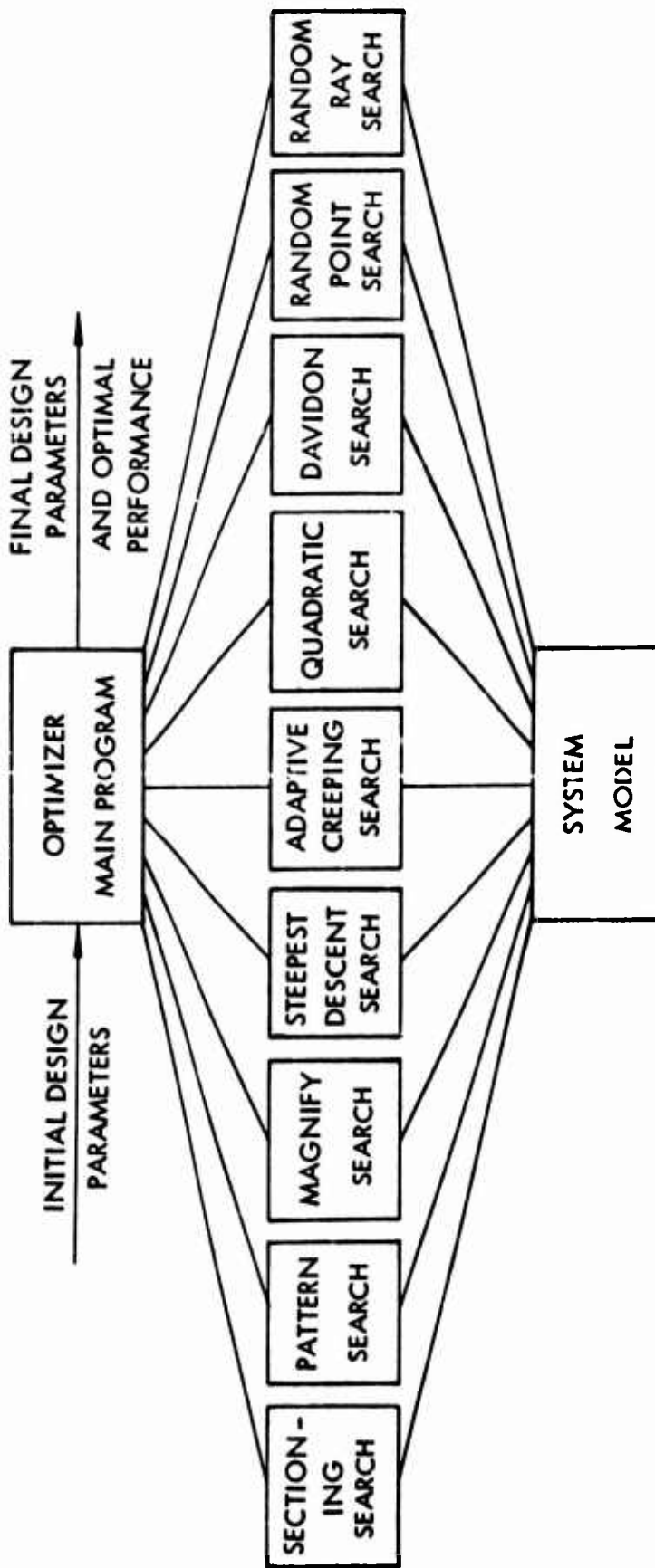


FIGURE 10.2-5.

NOMINAL VEHICLE CONFIGURATION

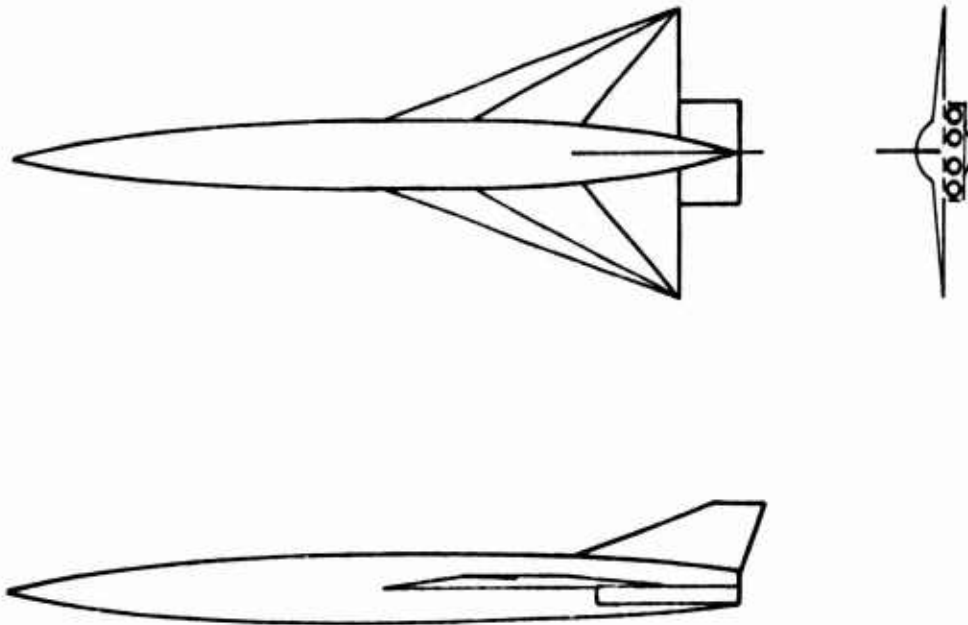


FIGURE 10.2-6.

OPTIMIZED HRA GENERAL ARRANGEMENT

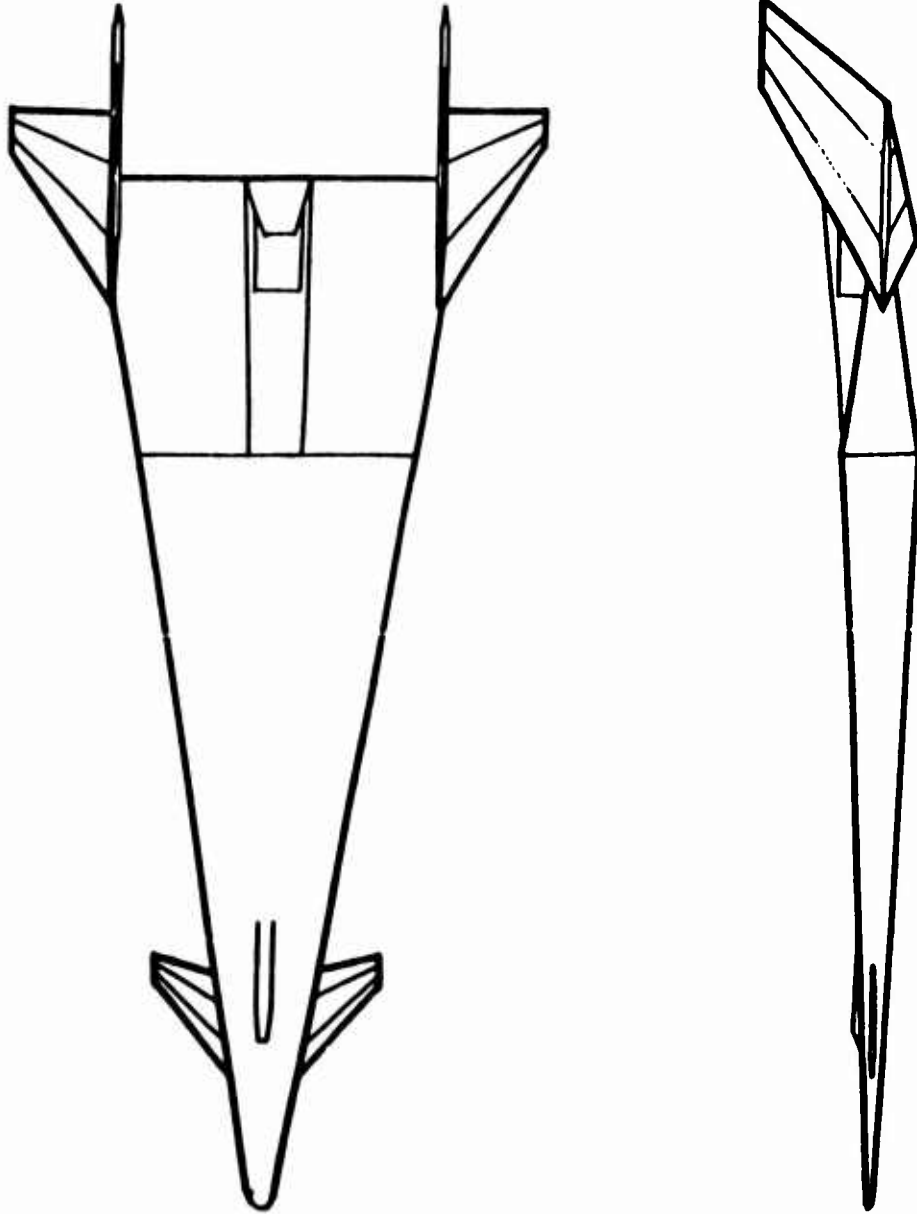
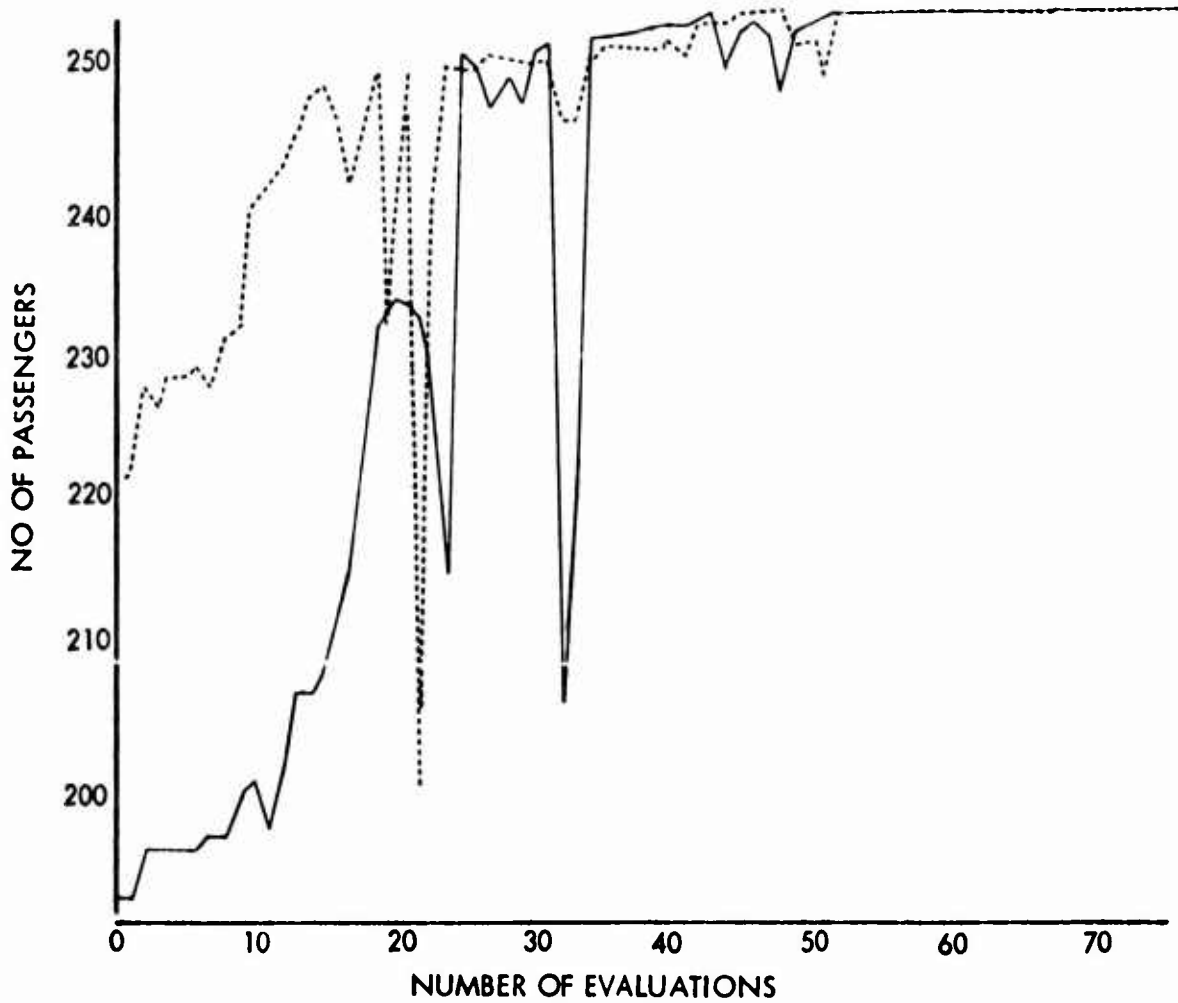


FIGURE 10.2-7.
OBJECTIVE FUNCTION CONVERGENCE
(NO. OF PASSENGERS)



| 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 |

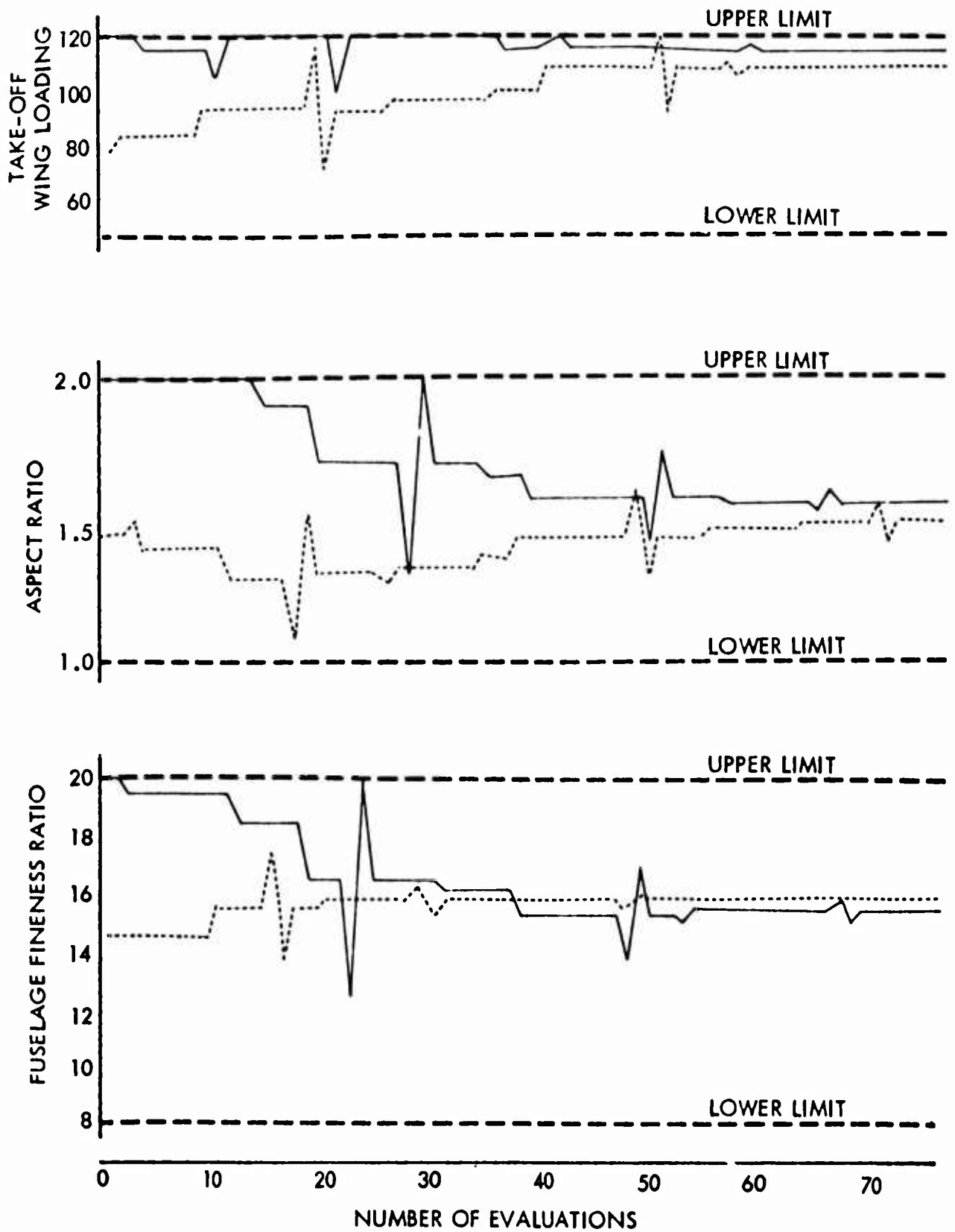


FIGURE 10.2-8. AERODYNAMIC PARAMETER CONVERGENCE - FIVE VARIABLE PROBLEM

FIGURE 10.2-9.

ENGINE PARAMETER CONVERGENCE - FIVE VARIABLE PROBLEM

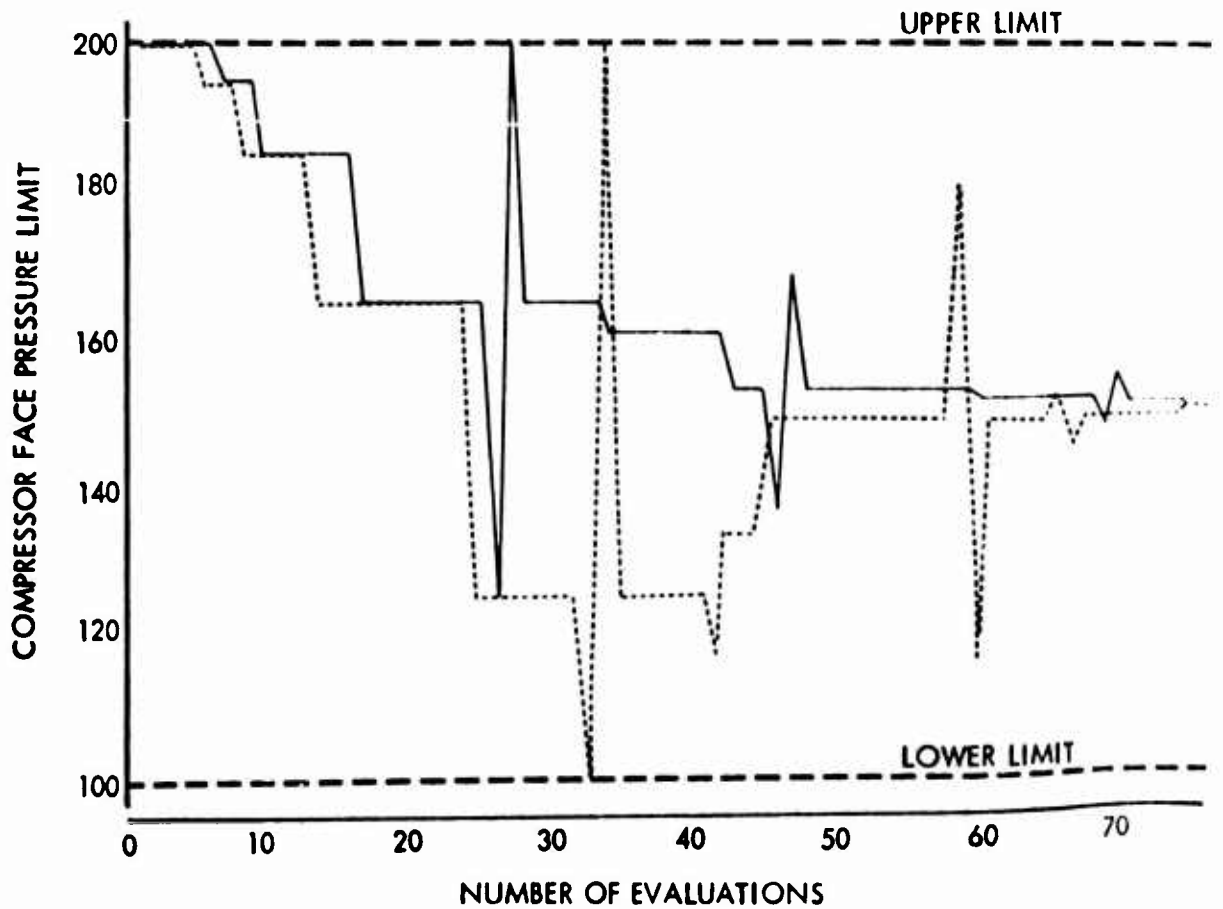
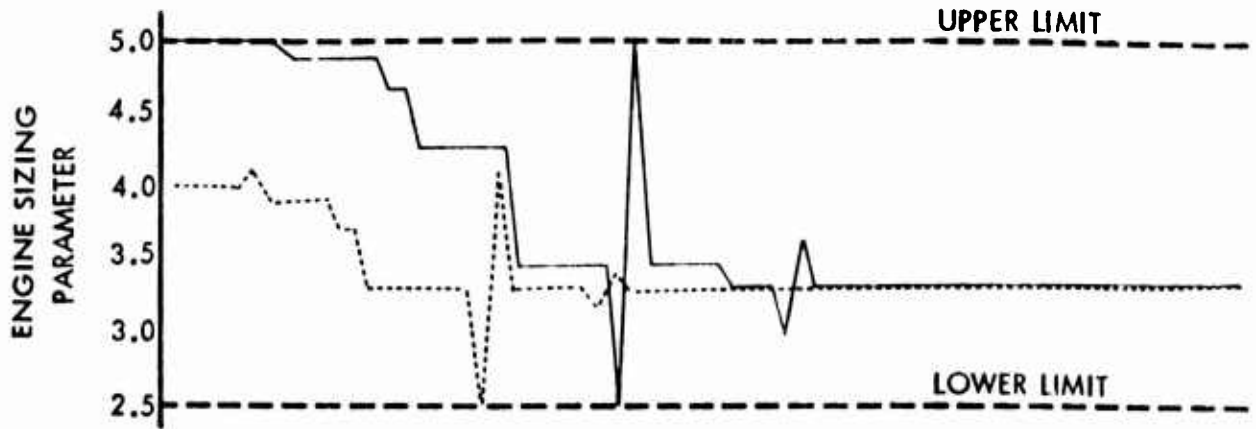


FIGURE 10.2-10.

NOMINAL AND OPTIMAL SENSITIVITIES ENGINE PRESSURE LIMIT

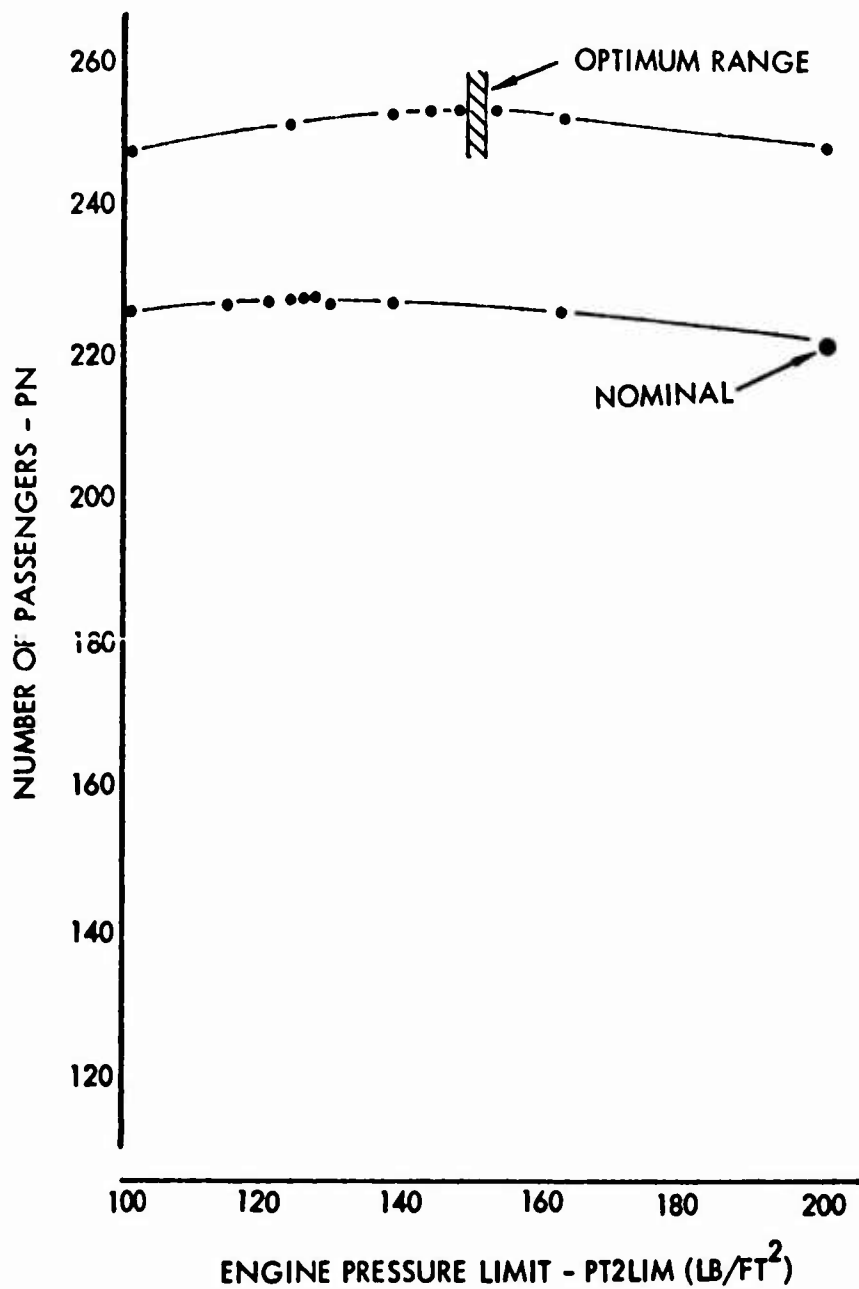


FIGURE 10.2-11.

NOMINAL AND OPTIMAL SENSITIVITIES FUSELAGE FINENESS RATIO

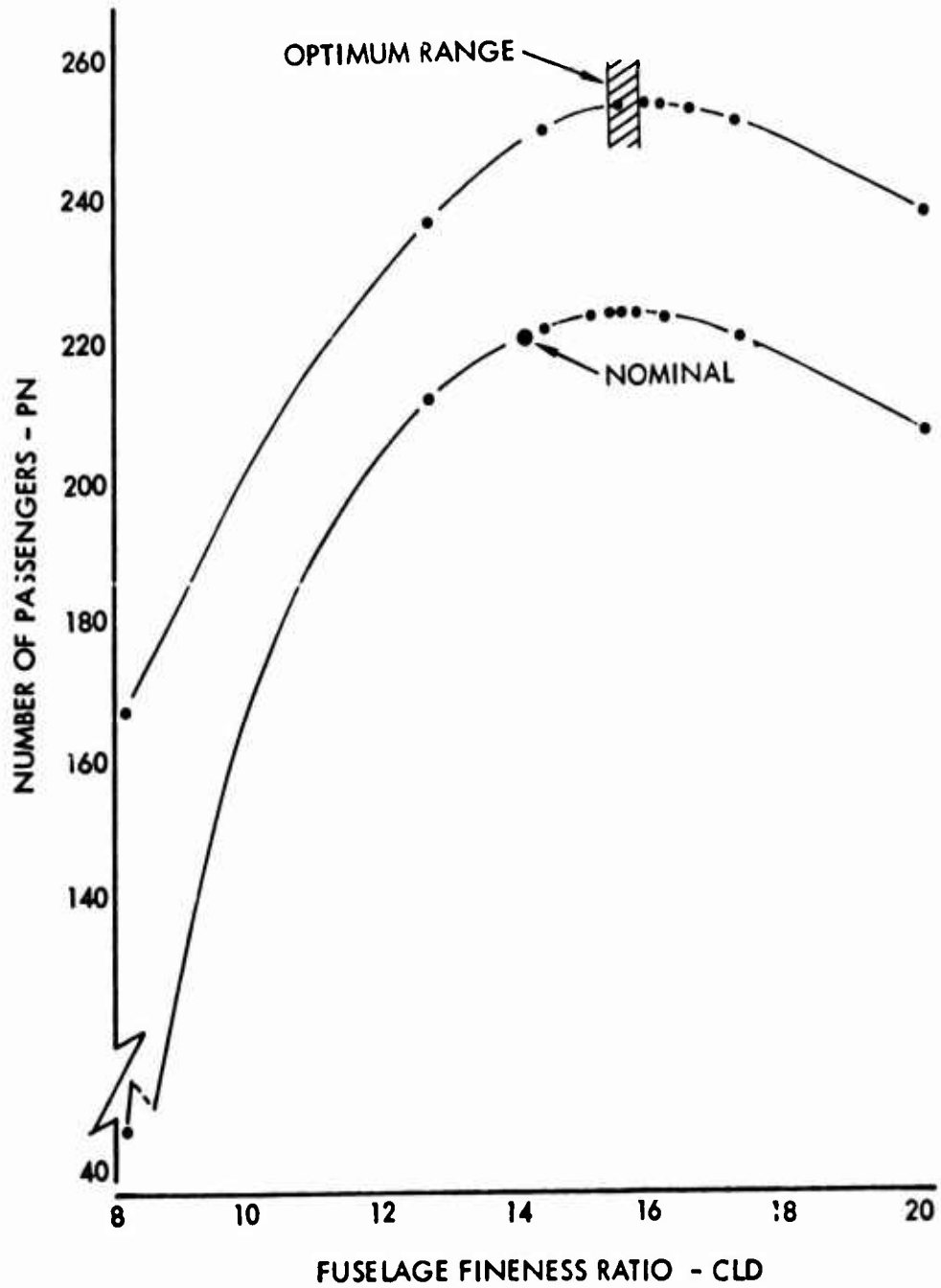


FIGURE 10.2-12.

NOMINAL AND OPTIMAL SENSITIVITIES WING ASPECT RATIO

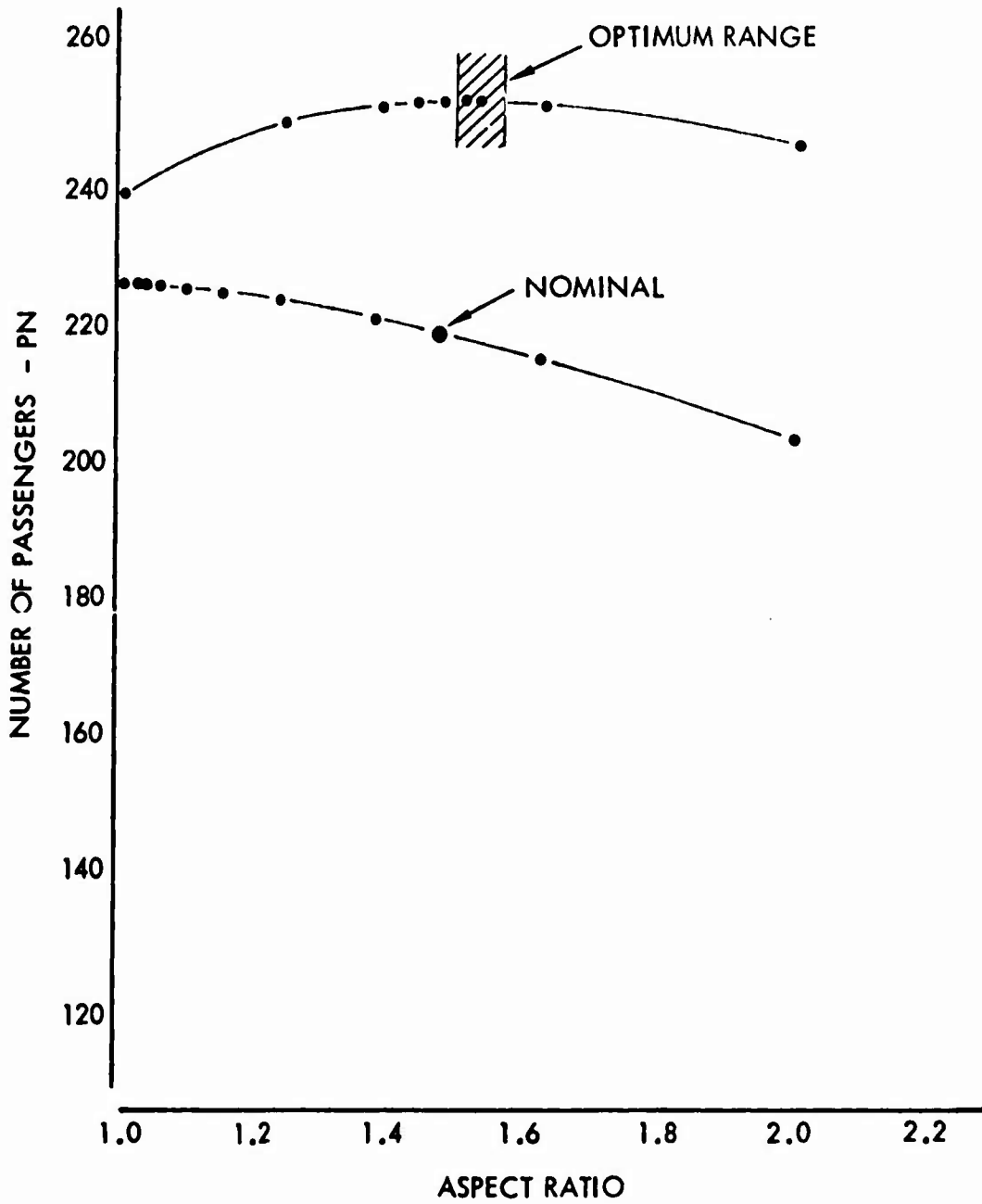


FIGURE 10.2-13.

**PAYOFF FUNCTION CONVERGENCE
TEN VARIABLES - ADAPTIVE SEARCH**

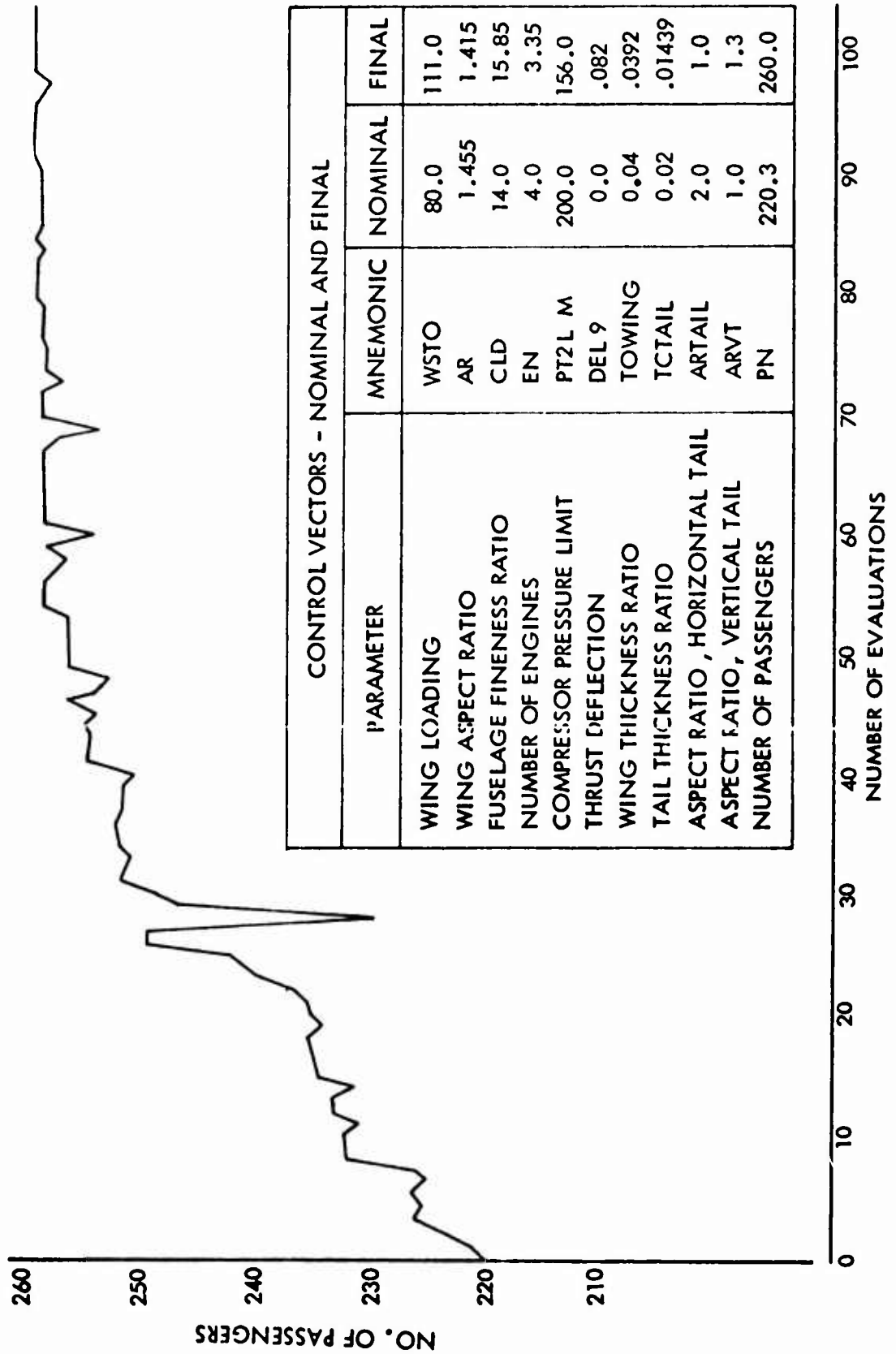


FIGURE 10.2-14.

UNCONSTRAINED OPTIMAL HYPERSONIC CRUISE VEHICLE FLIGHT PATH $0.4 \leq M \leq 6.0$

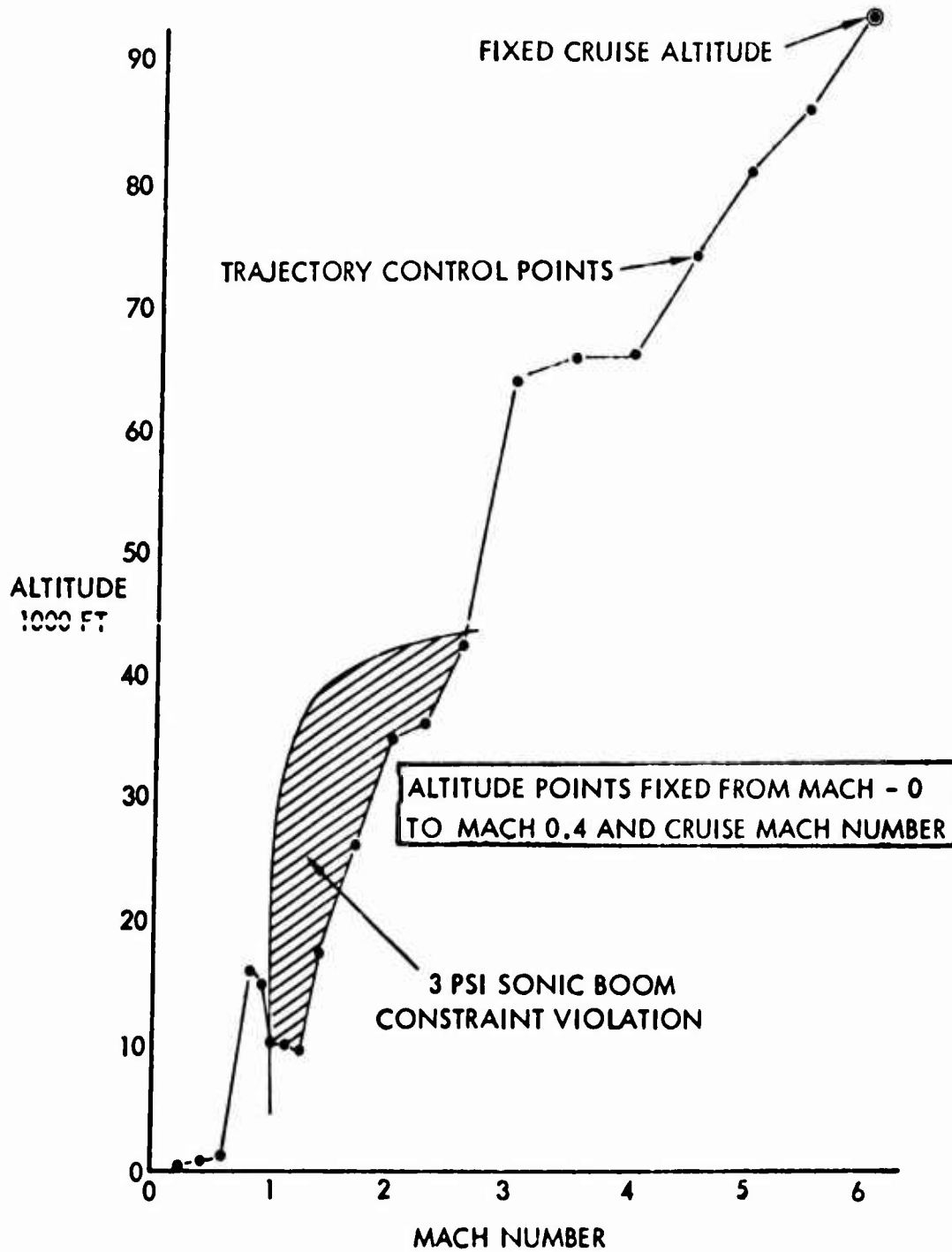


FIGURE 10.2-15.

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 |

SECTION 11

PRECOMPILER TECHNIQUES

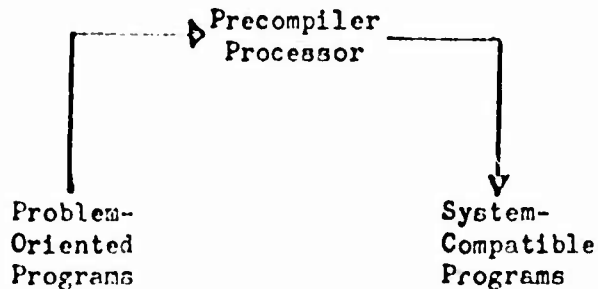
The ODIN system contains a generalized precompiler program, MACRO FORTRAN. This string processor allows the user to construct his own programming language, for example, extended FORTRAN. The MACRO FORTRAN program was obtained under Aerophysics Research Corporation funds from The Boeing Company.

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 pre-compiler 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 MAC Precompilers.

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.

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):

| | |
|--------------|--|
| <u>XXXXX</u> | 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)
- 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.
- proto-
type** 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 var₁ , name D₁) , ..., (int var_n , name D_n)

IDENTIFY β .EPCS. (V) \$d\$ prototype

lβ \$d\$ string expression

•β \$d\$ string expression

\$β \$d\$ string expression

BUILD β .E. \$d\$ string expression

SUBSTRING β .EPCS. INTO .E. ON .EPCSL.

COMPARE β .EPCS. (V) .EPCSL.

MOVE int expression FROM .E₁ . \$d\$ INTO .E₂ . \$d\$

MOVE .P. { TO } .P.
 { INTO }

COMPRESS .E.

COMPRESS .E. ALL BUT integer

CONVERT int expression to .E.

CONVERT .E. TO integer variable

DEFINE { INTEGER
 REAL
 LABEL } .EP.

ORIGIN { INTEGER
 REAL
 LABEL } integer expression

WARNINGS any appropriate diagnostic note

ERRORS any appropriate diagnostic note

ABORTS any appropriate diagnostic note

WARNINGS β \$d\$ string expression
ERRORS β \$d\$ string expression
ABORTS β \$d\$ string expression
LEVEL CANCEL
RESTORE
LINK TO name
DELIMIT STRING any one character (excluding blank)
COLUMN
SWITCH FIND integer variable
TYPE
REINIT MAC
END
FORMAT β (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 $\beta, V, d, etc., are explained in Section 1.8.$

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 Example.

```
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 CONTROL BLOCK card.

```
PROGRAM NAME(F1, NFIRST, L,F2, F3, F4, F5,..., FN,
             TAPE8, TAPE13, TAPE18, TAPE5=F2, TAPE6=F3,
             TAPE7=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.

Note: 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 these 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 is 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 β .S. (ν) prototype

where .S. is an EPCS string, ν is a statement label, and prototype is any combination of literals, column delimiters, and strings that satisfy Rule 3 below. (- α) means transfer to α if unsuccessful, (+ α) or (α) means transfer to α 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 opt1/ opt2/ opt3
```

where opt₁ is the name of the plotting device to be used, opt₂ is the type of plot desired, and opt₃ 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 (IS, S)
STRING (IL,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

bbb42bbbPLOTb6ONbDEVICEbA/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 | bbb42b |
| LL would be set to six | |
| A would be filled with | bDEVICEbA |
| LA would be set to nine | |
| B would be filled with | bLINEAR |
| LB would be set to seven | |
| C would be filled with | bX,bY,bXX,bYY |
| LC would be set to thirteen | |

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 7 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.

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)..,lH.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 lH
- The Jth character of string T
- A right parenthesis

Example 2 - `* / §1§ .L. §7§ GO / 10 / .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 *, l, 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

| <u>B (4 substrings)</u> | <u>C (4 substrings)</u> |
|-------------------------|-------------------------|
| 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

| <u>W (4 substrings)</u> | <u>R (3 substrings)</u> |
|-------------------------|-------------------------|
| 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 ANGLEbO7bATTACK  
and substring .B=3=. contains ANGLEbO7bATTACK
```

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. TO
INTO .C.

This form is best explained by the following example:

MOVE .A(I,J). TO .B(K,L).

case a) $(J - I + 1) \geq (L - K + 1)$

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 means 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 pppppppppppppppR0002 (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

| | |
|-----------------------|-------|
| <u>ORIGIN LABEL</u> | I-2*J |
| <u>ORIGIN REAL</u> | 32 |
| <u>ORIGIN INTEGER</u> | 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\$ 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.
WARNING\$

ERROR\$ 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 any other errors in the user program may be detected. No transformed program will be produced.
ERROR\$

ABORT\$ note... After this type of diagnostic note is printed for the user, his precompilation is aborted. No transformed program is produced.
ABORT\$

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).

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,Y),(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 ,5HCA 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 auxiliary 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

```

FORMAT  $\beta$  ( 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 character 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

| | | |
|-------------------------|--------------------------------------|-------------------|
| <u>(PRINT)</u> | \$d\$ | string expression |
| <u>(READ)</u> | \$d1\$,d2\$.E1. \$d3,d4\$.E2. etc. | |
| <u>SET (PRINT) UNIT</u> | | integer variable |
| <u>SET (READ) UNIT</u> | | 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 variable₁
SWITCH TYPE integer variable₂

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.

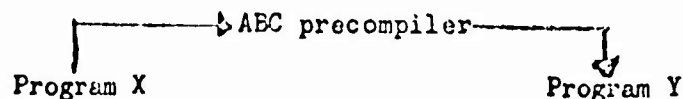
The FIND SWITCH indicates whether or not a successful IDENTIFY has occurred within a macro block. Each time a source statement is placed in the image string so that it may be examined by the various macro blocks, the find switch is set to zero. The control

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.



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 IBCMAP 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 less 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)
```

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 (IS,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 (IS,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
```

into the following:

```
      SUBROUTINE GAMMA
      DIMENSION JO001(2)
      DATA JO001/8HA,B,CALC/
      DIMENSION JO002(3)
      DATA JO002/13H(C(1),1=1,10)/
      .
      .
90001  FORMAT (1HO, 1P8E14.5)
90002  FORMAT (1HO, 11A10)
      WRITE (6,90002) JO001
      WRITE (6,90001) A,B,CALC
      .
      .
12    WRITE (6,90002) JO002
      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,1P8E14.5)
      * $1$ +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. $7$ 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. $7$ 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.
L A R E L , M A C M A C , R , V S N = X X X X .
RFL,30000.
R E W I N D (M A C M A C)
C O P Y R F (M A C M A C , M A C F O R , 1)
C O P Y R F (M A C M A C , M A C L I R , 1)
C O P Y R F (M A C M A C , A U T O L A Y , 1)
RFL,120000.
M A C F O R .
RFL,50000.
R E W I N D (Q X C O N)
R U N (S , , , Q X C O N)
A U T O L A Y (N E W P G M , L G O , M A C L I R)
R E W I N D (N E W P G M)
RFL,7777 (AS REQUIRED BY NEWPGM)
N E W P G M .
000000000000000000000000
S O U R C E D E C K C O N T A I N I N G M A C S T A T E M E N T S)
000000000000000000000000
A U T O L A Y T E X T C A R D S
(SEE AEROPHYSICS RESEARCH CORPORATION TN-122)
000000000000000000000000
D A T A D E C K T O B E P R O C E S S E D B Y N E W P G M)
000000000000000000000000
000000000000000000000000

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 |

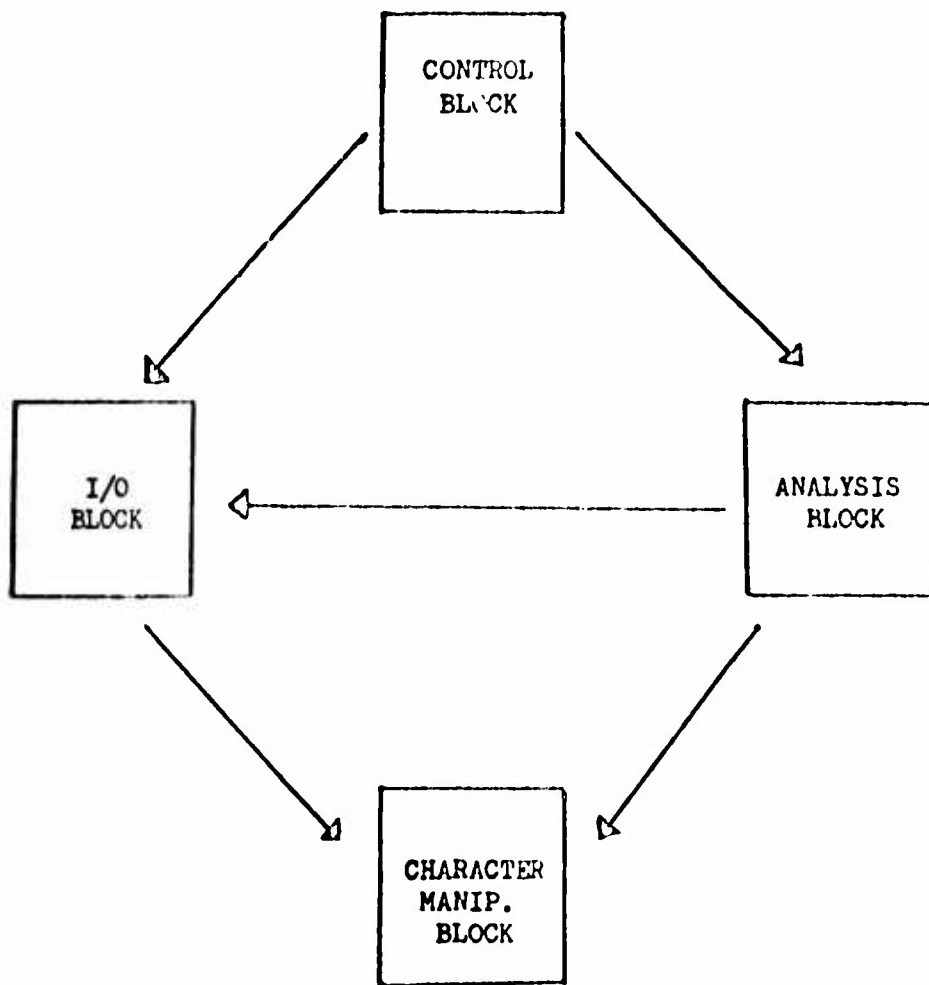


FIGURE 11.1-1 LOGICAL MAP OF A MAC PRECOMPILER.

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 COMPLIT 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.

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.

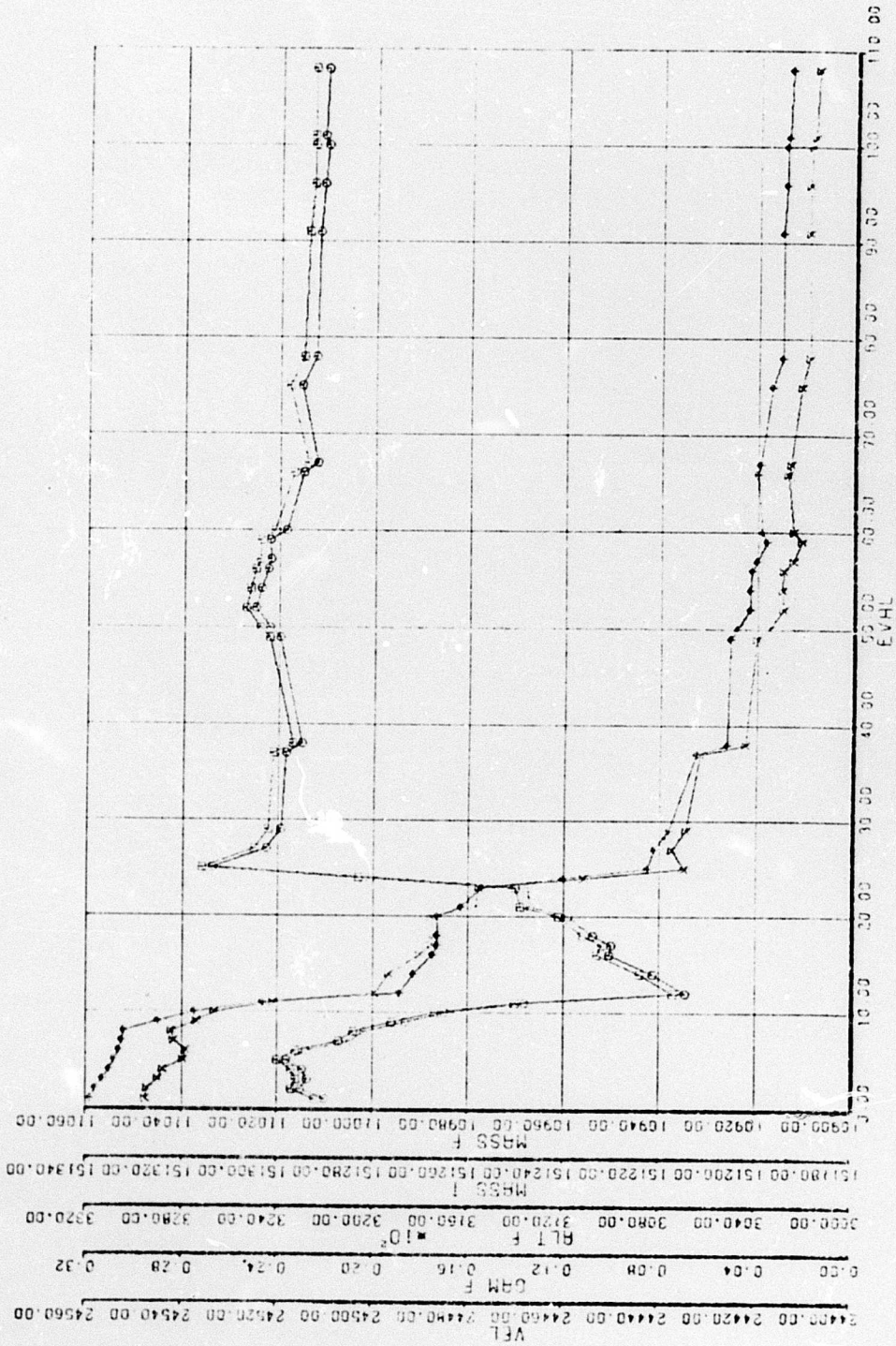
PLOTIN NAMELIST DATA

| NAME | DEFAULT(S) | DESCRIPTION OF INPUT |
|--------|------------|--|
| NPAGE | 0 | PAGE NUMBER OF THE PLOT (PRINTER ONLY) |
| ALTVAL | .FALSE. | LOGICAL VARIABLE, IF .TRUE. DATA WILL BE READ AS OBSERVATION FUNCTIONS. OTHERWISE DATA WILL BE READ AS PLOT ARRAYS FOR ALTVAL=.FALSE. PLOT ARRAYS ARE LOADED, A(1),--,A(N),R(1),--R(N),C(1)--C(N). FOR ALTVAL=.TRUE. PLOT ARRAYS ARE LOADED, A(1),R(1),C(1),-----,A(N),R(N),C(N). |
| 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 |
| CALCOM | .FALSE. | LOGICAL VARIABLE, IF .TRUE. CALCOMP PLOTS WILL BE GENERATED. |
| PRINTR | .FALSE. | LOGICAL VARIABLE, IF .TRUE. PRINTER PLOTS WILL BE GENERATED. |
| NOBSEP | NONE | NUMBER OF OBSERVATION FUNCTIONS (OR PLOT ARRAYS) |
| NAMES | .FALSE. | LOGICAL VARIABLE, IF .TRUE. NOBSEP SIX-CHARACTER NAMES WILL BE READ IN 12A6 FORMAT FOLLOWING THE REMARKS (IF ANY). THESE ARE THE PLOT ARRAY TITLES. |
| ORSPLT | 12(*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-OBS FUNCTIONS A ZERO INDICATES THE END OF ONE CHART, MORE CHARTS CAN BE GENERATED UP TO A LIMIT OF 12; ENTRIES IN THE ORSPLT ARRAY. USER SHOULD ENTER A VALUE OF 7777 AFTER THE LAST CHART. |
| ORSTH | NONE | PLOT DATA IN ONE OF THE FOLLOWING FORMATS. FOR ALTVAL=.FALSE. PLOT ARRAYS ARE LOADED, A(1),--,A(N),B(1),--R(N),C(1)--C(N). FOR ALTVAL=.TRUE. PLOT ARRAYS ARE LOADED, A(1),R(1),C(1),-----,A(N),R(N),C(N). |
| INPUT | 0 | ZERO FOR READING PLOT DATA FROM CARDS .GT.0 FOR READING PLOT DATA FROM UNIT 15 |
| NUMP | NONE | NUMBER OF PLOT POINTS PER FUNCTION (OR ARRAY) THERE IS AN INTERNAL LIMIT OF 213 POINTS. |

| | | |
|---------|---------|--|
| STOP | .FALSE. | LOGICAL VARIABLE. IF .TRUE. PROGRAM WILL STOP |
| SCALEF | 100*1.0 | SCALE FACTOR ARRAY, ONE FOR EACH PLOT ARRAY OR OBSERVATION FUNCTION |
| XSIZE | 6.0 | X-AXIS LENGTH IN INCHES |
| YSIZE | 8.0 | Y-AXIS LENGTH IN INCHES |
| MYX | .FALSE. | LOGICAL VARIABLE. IF .TRUE. USER MAY INPUT STARTX AND SCALEX |
| STARTX | 0.0 | STARTING VALUE FOR X-AXIS |
| SCALEX | 1.0 | UNITS PER INCH FOR X |
| MYY | .FALSE. | LOGICAL VARIABLE. IF .TRUE. USER MAY INPUT STARTY AND SCALEY |
| STARTY | 0.0 | STARTING VALUE FOR Y-AXIS |
| SCALEY | 1.0 | UNITS PER INCH FOR Y |
| 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.F. LINTYP=4 MEANS EVERY FOURTH POINT LINTYP=0 STRAIGHT LINES WITH SPECIFIED SYMBOL AT THE END OF THE LINE (SEE INTE0) LINTYP=+ LINES AND SYMPOLS LINTYP=- SYMPOLS ONLY |
| INTE0 | 1 | INTEGER FROM 0 TO 22 INDICATING THE PLOT SYMBOL. THE FIRST 20 ARE STANDARD NASA SYMBOL SET. (21 AND 22 ARE . AND +) |
| REMSIZ | 0.21 | HEIGHT OF TITLE FOR CALCOMP SHOULD BE MULTIPLE OF 0.07 |
| XMESH | NONE | ARRAY OF POINTS DEFINING THE X-AXIS OF A CONTOUR PLOT |
| YMFESH | NONE | ARRAY OF POINTS DEFINING THE Y-AXIS OF A CONTOUR PLOT |
| CONTOUR | .FALSE. | LOGICAL VARIABLE. IF .TRUE. A CONTOUR PLOT WILL BE GENERATED FROM ORSTH DATA CALCOM MUST BE SET .TRUE. |
| NZCUTS | NONE | NUMBER OF CONTOURS REQUESTED |
| ZCUTS | NONE | VALUES OF THE CONTOURS |
| DIALOG | .FALSE. | LOGICAL VARIABLE. IF .TRUE. DATA BASE OUTPUT ROUTINE WILL BE CALLED |

| | | |
|--------|---------|--|
| XMOVE | 8.5 | X-DISTANCE BETWEEN PLOT ORIGINS |
| YMOVE | 0.0 | Y-DISTANCE BETWEEN PLOT ORIGINS |
| NSKIPR | 0 | NUMBER OF OBSERVATION FUNCTIONS TO SKIP ON UNIT 15 BEFORE STARTING TO READ DATA |
| BLOCK | .FALSE. | LOGICAL VARIABLE, IF .TRUE., BINARY BLOCKING OF THE OBSERVATION UNIT (15) WILL BE EXPECTED. |
| REWIND | .FALSE. | LOGICAL VARIABLE, IF .TRUE., OBSERVATION TAPE (UNIT 15) WILL BE REWOUND BEFORE READING FROM IT. |
| NSKIPF | 0 | NUMBER OF OBSERVATION FUNCTION FILES THAT WILL BE SKIPPED ON UNIT 15 BEFORE STARTING TO READ DATA FROM IT. |
| NAXIS | 1 | NUMBER OF TIMES THE CALCOMP PEN WILL TRACE THE X- AND Y-AXES (INCL ANOTATION) |
| GRID | .FALSE. | LOGICAL VARIABLE, IF .TRUE., ONE INCH GRID WILL BE GENERATED ON CALCOMP PLOTS. |
| NUMP15 | .FALSE. | LOGICAL VARIABLE, IF TRUE, NUMP WILL BE READ FROM UNIT 15 |

FIGURE 12.1-1.
CONVERGENCE FROM APPROX VAR. SOLUTION



Reproduced from
best available copy.

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)

789 End of File Card

'EXECUTE MYPROGRAM'

(Insert data for compiled program
at this point)

789 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.

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 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*, *I1*, *I2*)

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 or 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, 50000g will be sufficient field length. AUTOLAY will abort if the following internal tables overflow:

| <u>Name of Table</u> | <u>Size</u> |
|--------------------------------|----------------------|
| 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 40500g 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 1000g 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=ARCL,  
       CN=ARCL,MD=ARCL,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

APPENDIX I

AN INTRODUCTION TO DIALOG:

A DESIGN INTEGRATION AND ANALYSIS LANGUAGE

ABSTRACT

A design integration and analysis language has been implemented in a CDC 6000 series computer code called DIALOG. 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

| | <u>Page</u> |
|---------------------------------------|-------------|
| 1. SUMMARY | I-1 |
| 2. INTRODUCTION | I-6 |
| 3. GENERAL DESCRIPTION OF DIALOG | I-8 |
| 3.1 ODIN Concept | I-8 |
| 3.2 DIALOG Executive | I-8 |
| 3.3 Control Card Data Base | I-11 |
| 3.4 Design Data Base | I-12 |
| 3.5 Information Storage and Retrieval | I-12 |
| 3.6 Units Conversion and Scaling | I-12 |
| 3.7 Report Generation | I-14 |
| 4. THE DIALOG LANGUAGE | I-15 |
| 4.1 Control Directives | I-17 |
| 4.2 Communication Commands | I-20 |
| 4.2.1 Replacement command | I-20 |
| 4.2.1.1 Simple replacement | I-21 |
| 4.2.1.2 Replacement with scaling | I-21 |
| 4.2.1.3 Array replacement | I-22 |
| 4.2.2 Action Commands | I-22 |
| 4.2.2.1 ADD command | I-25 |
| 4.2.2.2 The DELETE command | I-26 |
| 4.2.2.3 The DEFINE command | I-26 |
| 4.2.2.4 Special ADD-like commands | I-26 |
| 4.2.2.5 The . (comment) command | I-27 |
| 4.2.3 Logic Commands | I-27 |
| 4.3 Report Generation and Graphics | I-29 |
| 4.3.1 Stylized reports | I-30 |
| 4.3.2 Graphics | I-30 |
| 5. DIALOG USAGE EXAMPLES | I-36 |
| 6. REFERENCES | I-45 |

TABLE OF CONTENTS

| | <u>Page</u> |
|---|-------------|
| APPENDIX A: DATA BASE CONSTRUCTION | I-A1 |
| 1. Data Base Size | I-A1 |
| 2. Description of Direct Access Method | I-A2 |
| 2.1 Hash Coding the Key | I-A2 |
| 2.2 Resolving Collisions | I-A3 |
| 2.3 Storage and Retrieval Method | I-A4 |
| APPENDIX B: MULTIPLE PROGRAM EXECUTION | I-B1 |
| 1. Description of CCLINK | I-B1 |
| 2. Use of CCLINK | I-B2 |
| 3. System Interface Requirements for CCLINK | I-B3 |
| APPENDIX C: ODIN SIMULATION EXAMPLE | I-C1 |
| 1. Summary | I-C1 |
| 2. Introduction | I-C3 |
| 3. The ODIN/RLV Mission and Vehicle Model | I-C4 |
| 3.1 Trajectory Profile | I-C5 |
| 3.2 Vehicle Characteristics and Constraints | I-C6 |
| 3.3 Weights and Geometry | I-C7 |
| 3.4 Other Limitations on the Simulations | I-C8 |
| 4. Payload Optimization | I-C11 |
| 4.1 Selection of Engine Vacuum Thrust | I-C11 |
| 4.2 Selection of Booster and Engine Size | I-C13 |
| 4.3 Cost Sensitivity | I-C17 |
| 5. Data Intercommunication Between SSSP and AESOP | I-C18 |
| 6. Sample Output from DIALOG | I-C21 |
| 7. Conclusions | I-C27 |
| 8. References | I-C28 |

LIST OF FIGURES

| <u>Figure</u> | | <u>Page</u> |
|---------------|--|-------------|
| 1 | Schematic of the ODIN Concept | I-2 |
| 2 | ODIN Control Logic | I-4 |
| 3 | DIALOG Executive Functions | I-9 |
| 4 | ODIN/DIALOG Functions | I-13 |
| 5 | The DIALOG Schematic | I-16 |
| 6 | Control Directives | I-18 |
| 7 | ODIN/RLV Control Directives | I-19 |
| 8 | Rules for Using Communication Commands | I-23 |
| 9 | Action Commands | I-24 |
| 10 | Output from Logical Command 'DBUMP' | I-27 |
| 11 | Example Input and Output for Report | I-31 |
| 12 | Example Output from Printer Plot Option | I-32 |
| 13 | Example Output from CALCOMP Plot Option | I-34 |
| 14 | Example of CDC 250 Hard Copy Option | I-35 |
| 15 | A Sample Optimization Using ODIN | I-37 |
| 16 | Example of Construction and Storage of Control Card Data Base | I-38 |
| 17 | Graphical Presentation of Data from Any Analysis Program | I-39 |
| 18 | Use of FORTRAN Language to Augment Analysis | I-41 |
| 19 | Geometry Perturbation for the 040A Orbiter | I-42 |
| 20 | Comparison of Batch and Interactive Modes | I-43 |
| A1 | Composition of the Directory, FSL | I-A5 |
| B1 | Example Use of CCLINK | I-B4 |
| C1 | Schematic of Sample ODIN/RLV Model | I-C2 |
| C2 | The Two-Stage Space Shuttle Concept | I-C4 |
| C3 | Shuttle Engine Sizing Study | I-C12 |
| C4 | Mass Distribution and Engine Size Optimization | I-C14 |
| C5 | Booster Sizing Study | I-C15 |
| C6 | Payload Contours Stage Mass Distribution and Engine Size Parameters | I-C16 |
| C7 | Data Base Configuration after Initialization | I-C22 |
| C8 | Data Base Configuration after SSSP | I-C23 |
| C9 | Data Base Configuration after DAPCA (Booster) | I-C24 |
| C10 | Data Base Configuration after DAPCA (Orbiter) | I-C25 |

LIST OF FIGURES

| <u>Figure</u> | | <u>Page</u> |
|---------------|-------------------------------------|-------------|
| C11 | Data Base Configuration after AESOP | I-C26 |

APPENDIX I

AN INTRODUCTION TO DIALOG: A DESIGN INTEGRATION AND ANALYSIS LANGUAGE WITH OPEN-ENDED GROWTH CAPABILITY

by C. R. Glatt, D. A. Watson, R. T. Jones, and D. S. Hague
Aerophysics Research Corporation

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.

DIALOG is the control and communication executive computer program which implements a *community of programs* concept. Figure 1 shows the relationship between DIALOG and the ODIN library of independent technology modules. DIALOG 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 DIALOG executive.

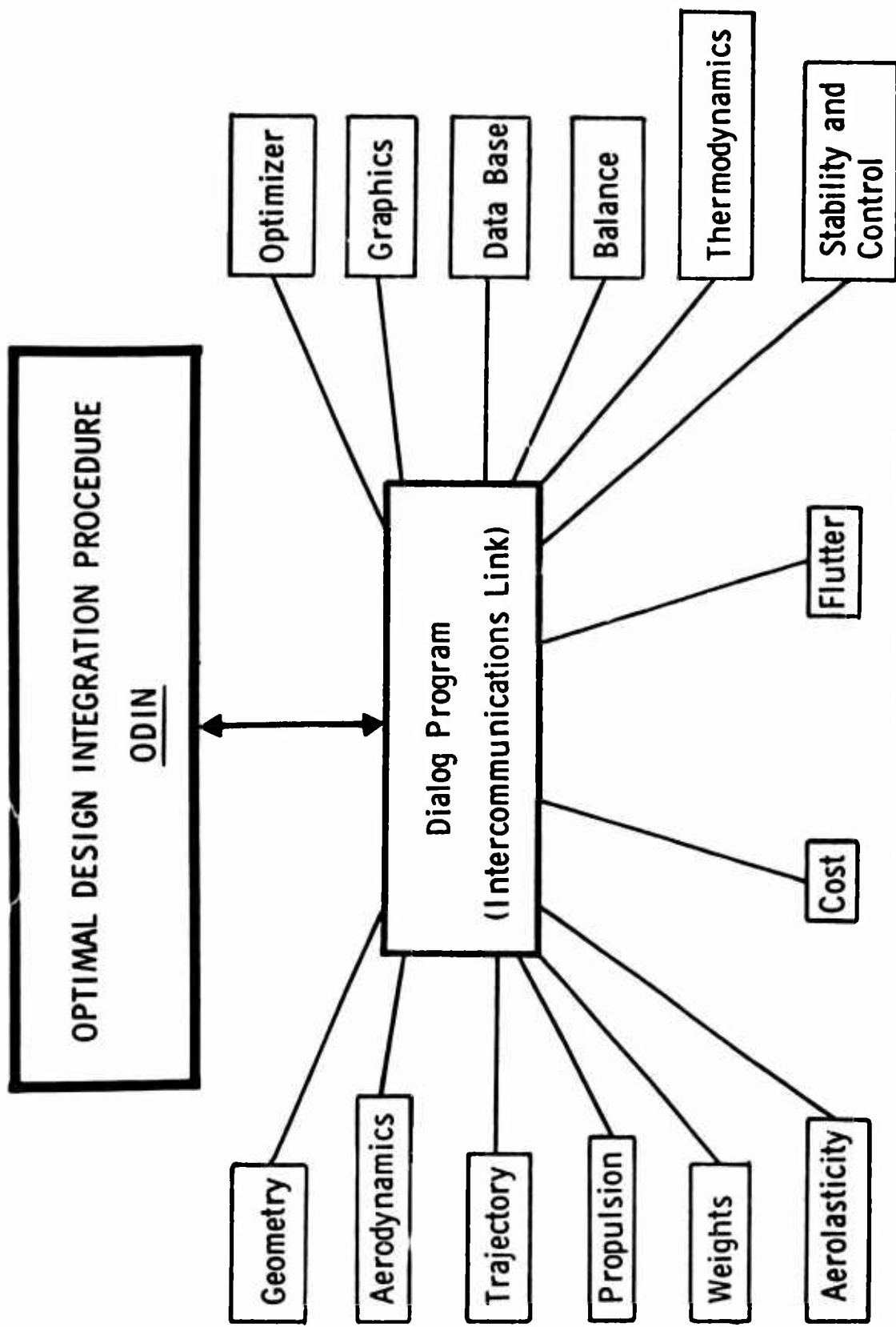


Figure 1. Schematic of the ODIN Concept

As a result of the development of DIALOG, 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 the DIALOG executive. 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.*

CONTROL LOGIC

PROGRAM FLOW

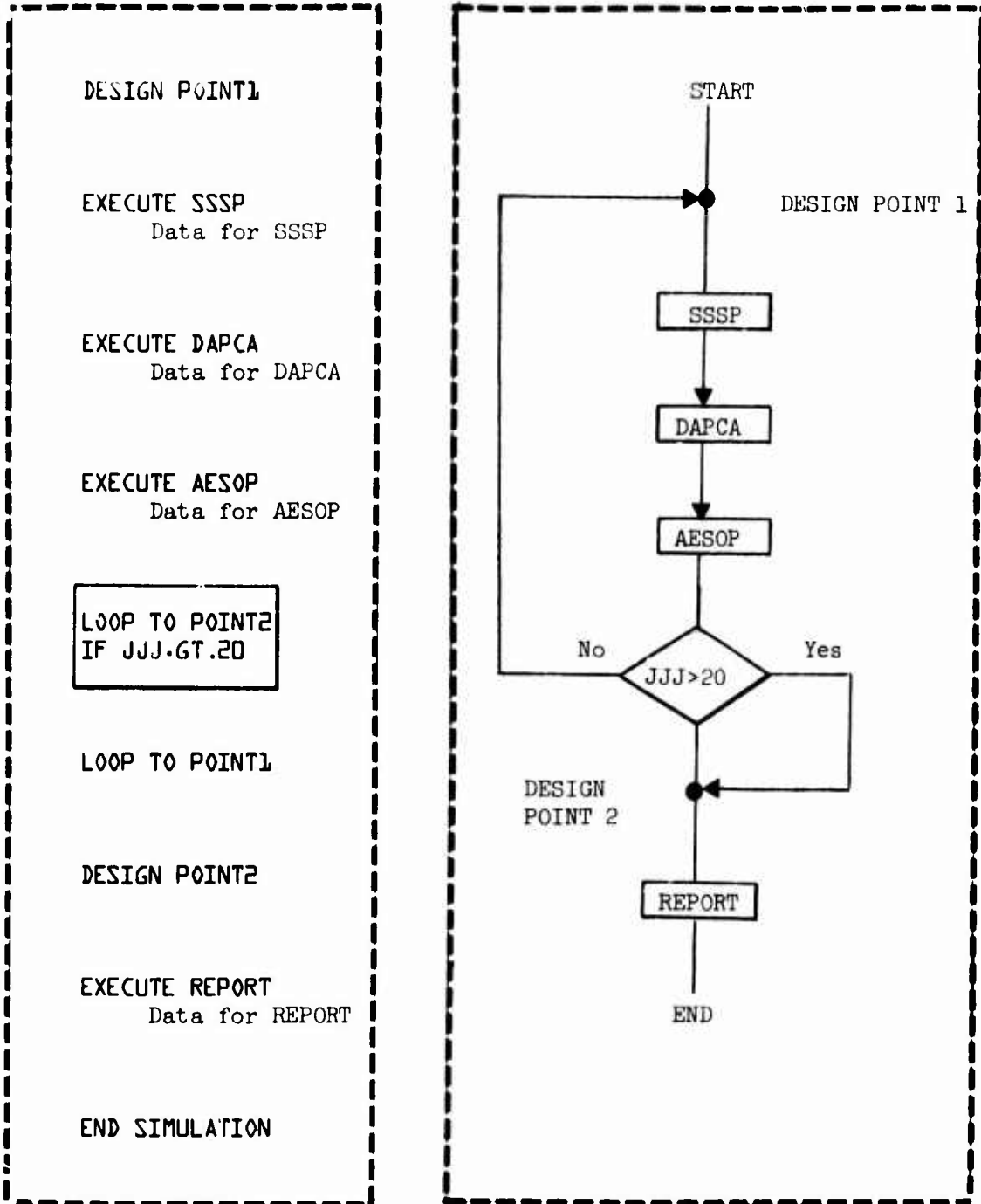


Figure 2. ODIN Control Logic

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 DIALOG. 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 DIALOG executive is an effort to overcome most of the shortcomings inherent in its predecessors.

3. GENERAL DESCRIPTION OF DIALOG

An independent executive computer program called DIALOG 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 DIALOG. 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 DIALOG executive computer program.

3.2 DIALOG Executive

The DIALOG functions are shown schematically in Figure 3. It pre-processes 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, DIALOG is repeatedly executed after each design program. At each successive execution DIALOG extracts selected information from the design program output and installs or updates the information in the data base. Additionally, DIALOG interrogates the input stream of the succeeding design program and merges data base information with it. In practice, the DIALOG program is essentially transparent

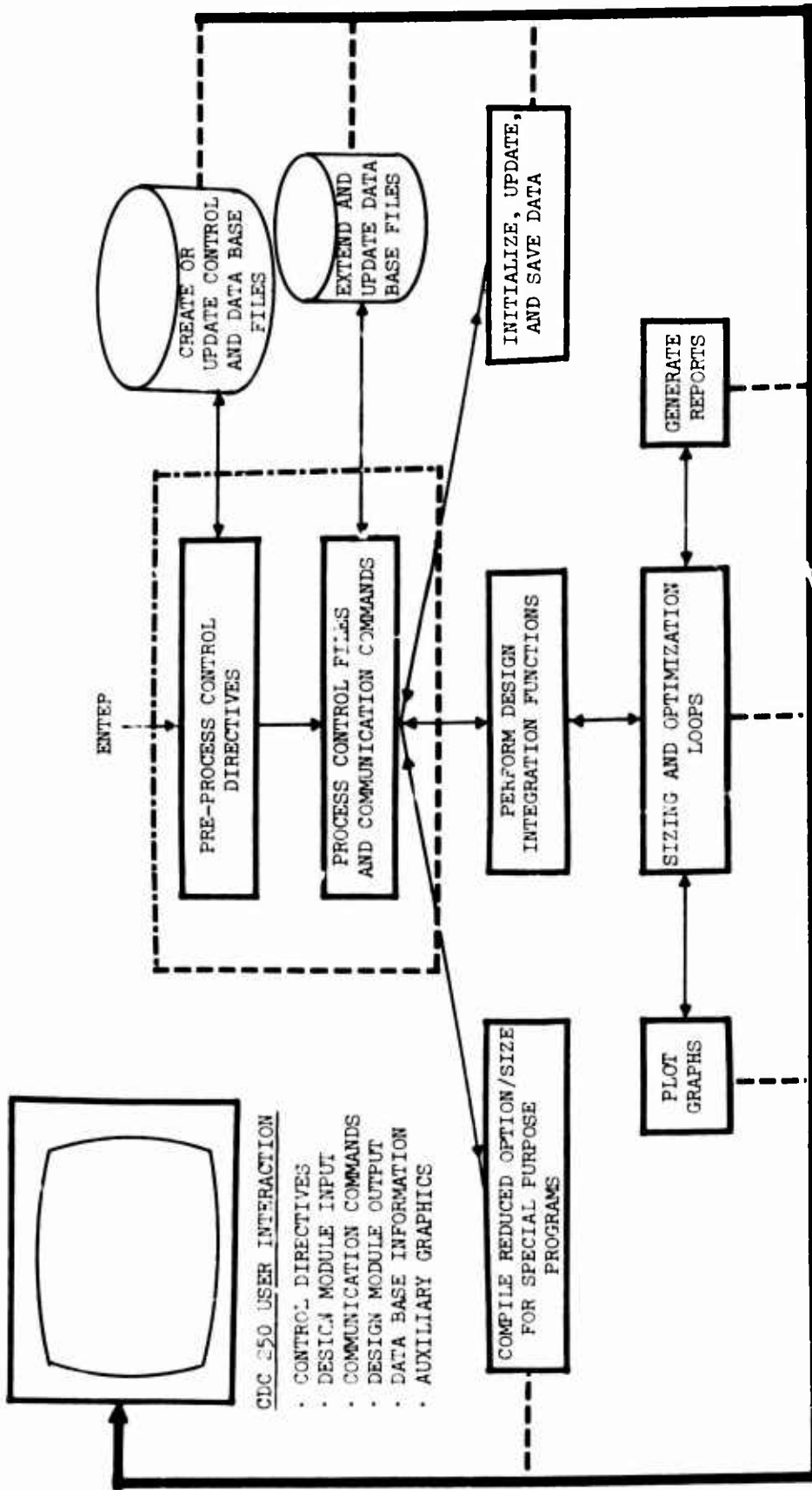


Figure 3. DIALOG Executive Functions

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 DIALOG 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. DIALOG 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 DIALOG 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. DIALOG combines the capabilities of RANDAC and CCLINK to form a unified approach to vehicle design synthesis.

3.3 Control Card Data Base

DIALOG 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 DIALOG language.

3.4 Design Data Base

DIALOG 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 DIALOG extracts selected information from the design program output and installs or updates the information in the data base. Additionally, DIALOG interrogates the input stream of the succeeding design program and merges data base information with it. In practice, the DIALOG program is essentially transparent to the user, appearing only as an *input language* which augments the input of existing design programs.

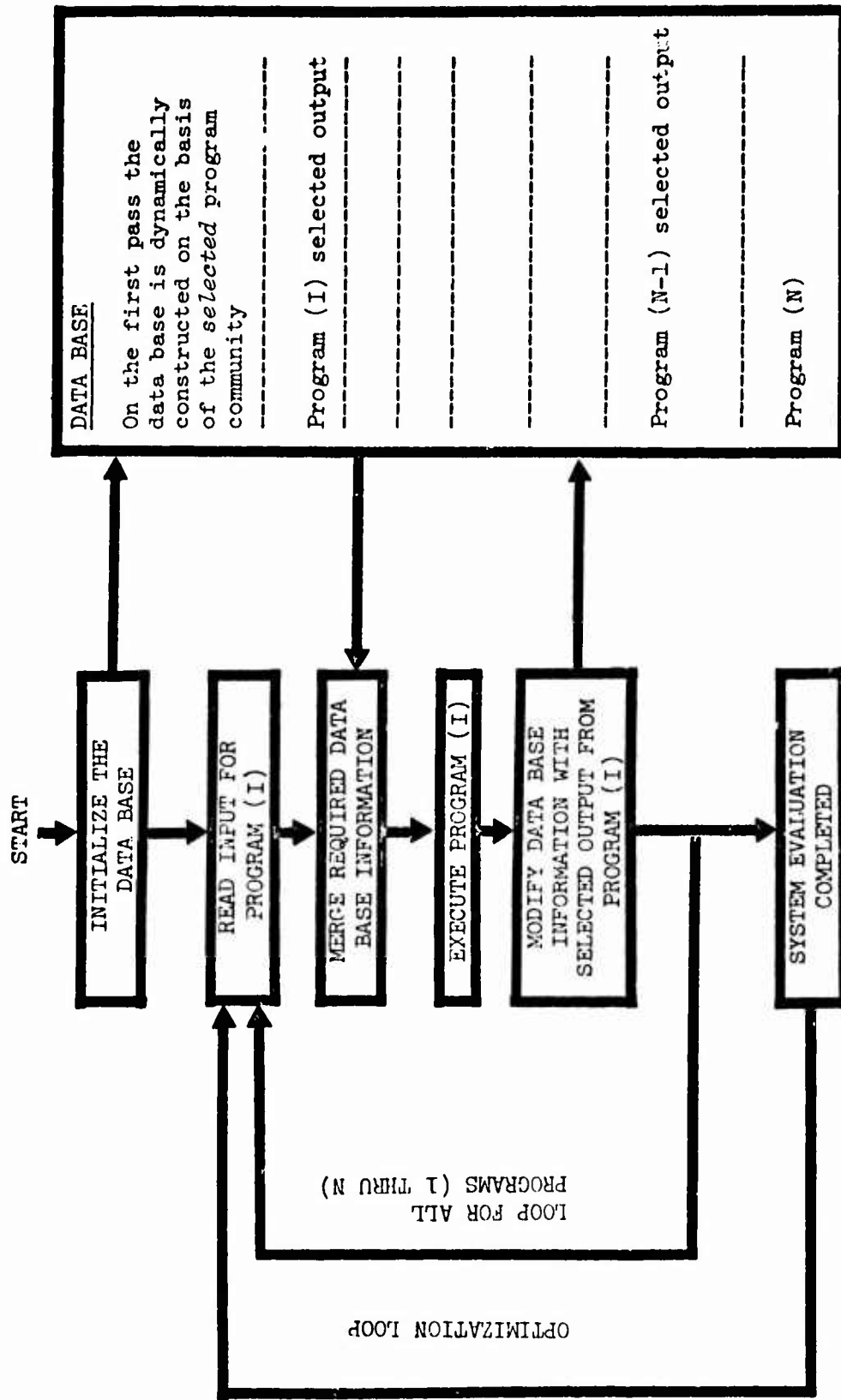
3.5 Information Storage and Retrieval

DIALOG 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 DIALOG 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/DIALOG FUNCTIONS



3.7 Report Generation

The DIALOG 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 DIALOG LANGUAGE

The DIALOG 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 DIALOG 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 DIALOG. 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.

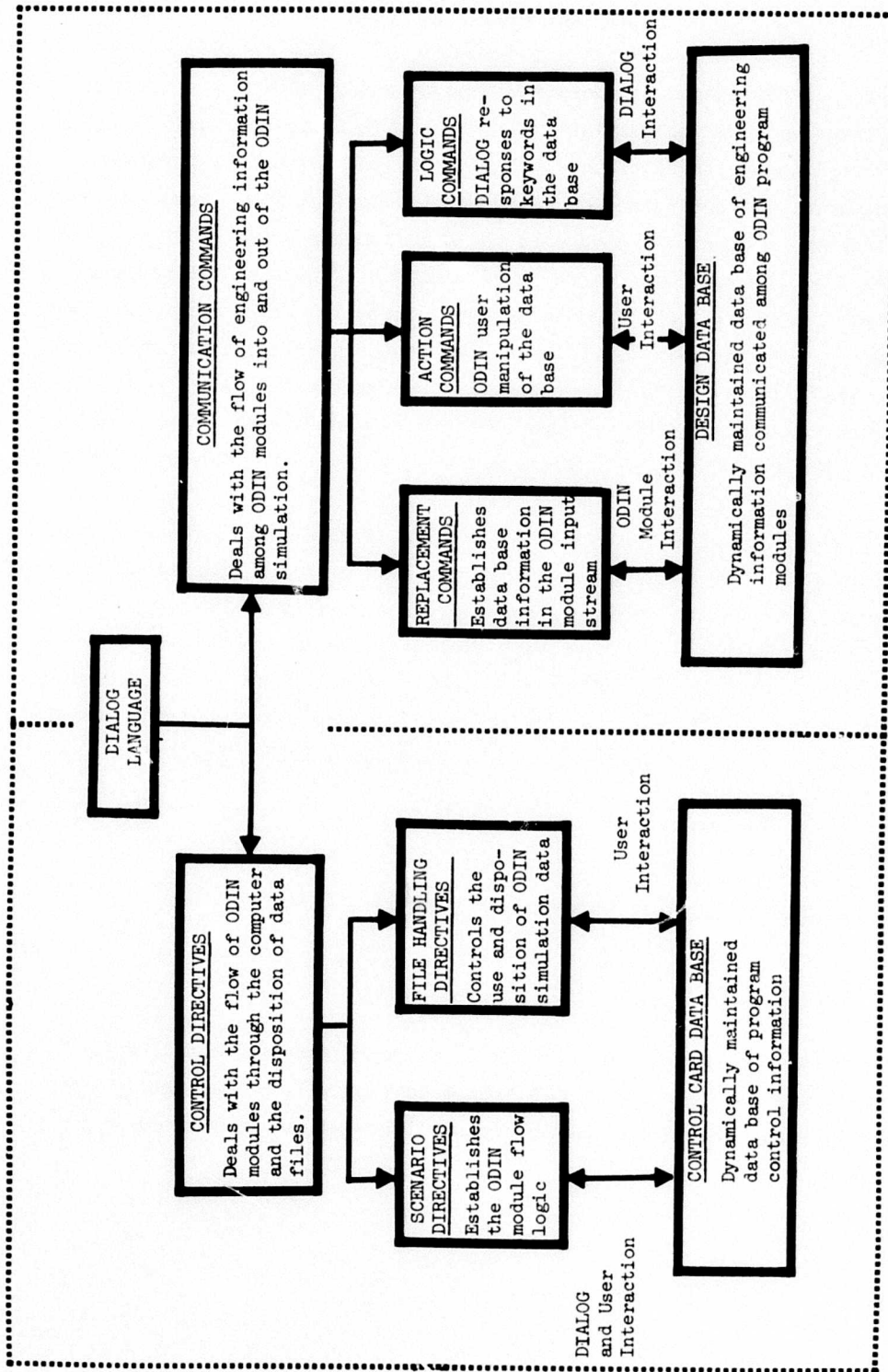


Figure 5. The DIALOG Schematic

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* DIALOG 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 INITIAL 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 <i>after</i> 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 DIALOG.

4.2.1 Replacement command. Among the DIALOG 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 DIALOG, and the communication commands are interpreted. Based on the instructions encountered, DIALOG 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 DIALOG 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.

DIALOG 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 DIALOG 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, DIALOG 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 DIALOG delimiters. DIALOG 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 Action Commands. Action commands consist of five types: ADD, DELETE, DEFINE, INITIAL, 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 DIALOG, 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

- '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 if omitted. description is the hollerith information associated with the variable A.
- 'INITIAL 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. DIALOG 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

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. DIALOG 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. DIALOG 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 Logic Commands. Logic commands control the flow of information within the DIALOG program. They are in the form of data base entries or keywords which are interrogated at each pass through DIALOG. 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 |
| AUILD | 19 | 1 | 0 | DEFINE | DYNAMIC DB BUILD OPTION |
| CRPSKP | 1 | 1 | 32 | ADD | CARD SKIP OPTION |
| DBDUMP | 31 | 1 | | OFFINE | CAUSES DATA BASE DUMP |
| DISPLAY | 13 | 1 | WPAYLO | INITIAL | SCOPE DISPLAY FUNCTION |
| ELTIME | 23 | 1 | 28.0920000 | DEFINE | TOTAL ELAPSED TIME |
| JJ4EST | 25 | 1 | 1 | AESOUT | BEST PERFORMANCE EVALUATION |
| JJJ | 17 | 1 | 0 | AESOUT | AESOP EVALUATION COUNTER |
| WJXR | 3 | 1 | 2.71999999999999209 | ADD | PROPELLANT MIXTURE RATIO |
| RUNID | 21 | 1 | *ODIN JAN 22, 72 | OFFINE | IDENTIFICATION FOR DIALOG RUN |
| TCOSTH | 35 | 1 | 48573.95593780655658 | ADD | TOTAL COST OF BOOSTER |
| TCOSTN | 37 | 1 | 0.19037115923077E+05 | DATAOUT | TOTAL COST OF ORBITER |
| THRST | 5 | 1 | 525000.000000000000 | ADD | ENGINE VACUUM THRUST |
| VSTGR | 29 | 1 | 0.91501785518072E+04 | WTOUT | BOOSTER STAGING VELOCITY |
| VSTGO | 33 | 1 | 0.25853097798483E+05 | WTOUT | ORBITER STAGING VELOCITY |
| WGRJSP | 11 | 1 | 0.26662821532874E+07 | WTOUT | GROSS WEIGHT OF BOOSTER |
| WGRJSO | 15 | 1 | 0.71802039902725E+06 | WTOUT | GROSS WEIGHT OF ORBITER |
| WPAYLO | 27 | 1 | 0.3184776197751E+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, DIALOG 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. OUTDMF 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 DIALOG 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 DIALOG executive at any point in the simulation. The report is generated by the control directive 'EXECUTE REPORT'. DIALOG 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. Larger 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 DIALOG 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. DIALOG 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.

1
0

SUMMARY REPORT FOR ODIN/RLV

(# IJJ# CYCLE(S) ELAPSED TIME = #ELTIME# CPU SECONDS

#ADD A=TCOSTN+TCOSTB#

#ADD KN=1.689#

#ADD V=0.45359#

| | | | | | |
|----------------------|-------|------------|------------|------------|-----------|
| MAXIMUM PAYLOAD | | #WPAYLO | # LBS | #WPAYLO*K# | KILOGRAMS |
| BOOSTER WEIGHT | | #WGMOSH | # LBS | #WGMOSH*K# | KILOGRAMS |
| ORBITER WEIGHT | | #WGROSO | # LBS | #WGROSO*K# | KILOGRAMS |
| TOTAL COST | | #A | # MILLIONS | | |
| BOOSTER COST | | #TCOSTB | # MILLIONS | | |
| ORBITER COST | | #TCOSTN | # MILLIONS | | |
| STAGING VELOCITY | | #VSTGH | # FPS | #VSTGR/KN# | KNOTS |
| ENGINE VACUUM THRUST | | #THRUST | # LBS | | |
| BOOSTER MASS RATIO | | #ALPHA(2)# | | | |

#ADD N=JJEST#

BEST CYCLE=#JJEST#, THE #N*2-1#TH RECORD ON FILE OUTSAV. N=#N#

END OF ODIN/RLV SUMMARY

REPORT

00000000000000000000

SUMMARY REPORT FOR ODIN/RLV

1 CYCLE(S)

ELAPSED TIME = 58.26500 CPU SECONDS

| | | | | | |
|----------------------|-------|-------------|----------|------------|-----------|
| MAXIMUM PAYLOAD | | 31245.8096 | LBS | 14444.8500 | KILOGRAMS |
| BOOSTER WEIGHT | | 2571117.97 | LBS | 1211592.40 | KILOGRAMS |
| ORBITER WEIGHT | | 713186.621 | LBS | 323494.319 | KILOGRAMS |
| TOTAL COST | | 267409.5935 | MILLIONS | | |
| BOOSTER COST | | 34845.2395 | MILLIONS | | |
| ORBITER COST | | 218974.3540 | MILLIONS | | |
| STAGING VELOCITY | | 9202.29158 | FPS | 5448.36683 | KNOTS |
| ENGINE VACUUM THRUST | | 525000.000 | LBS | | |
| BOOSTER MASS RATIO | | 2.72500000 | | | |

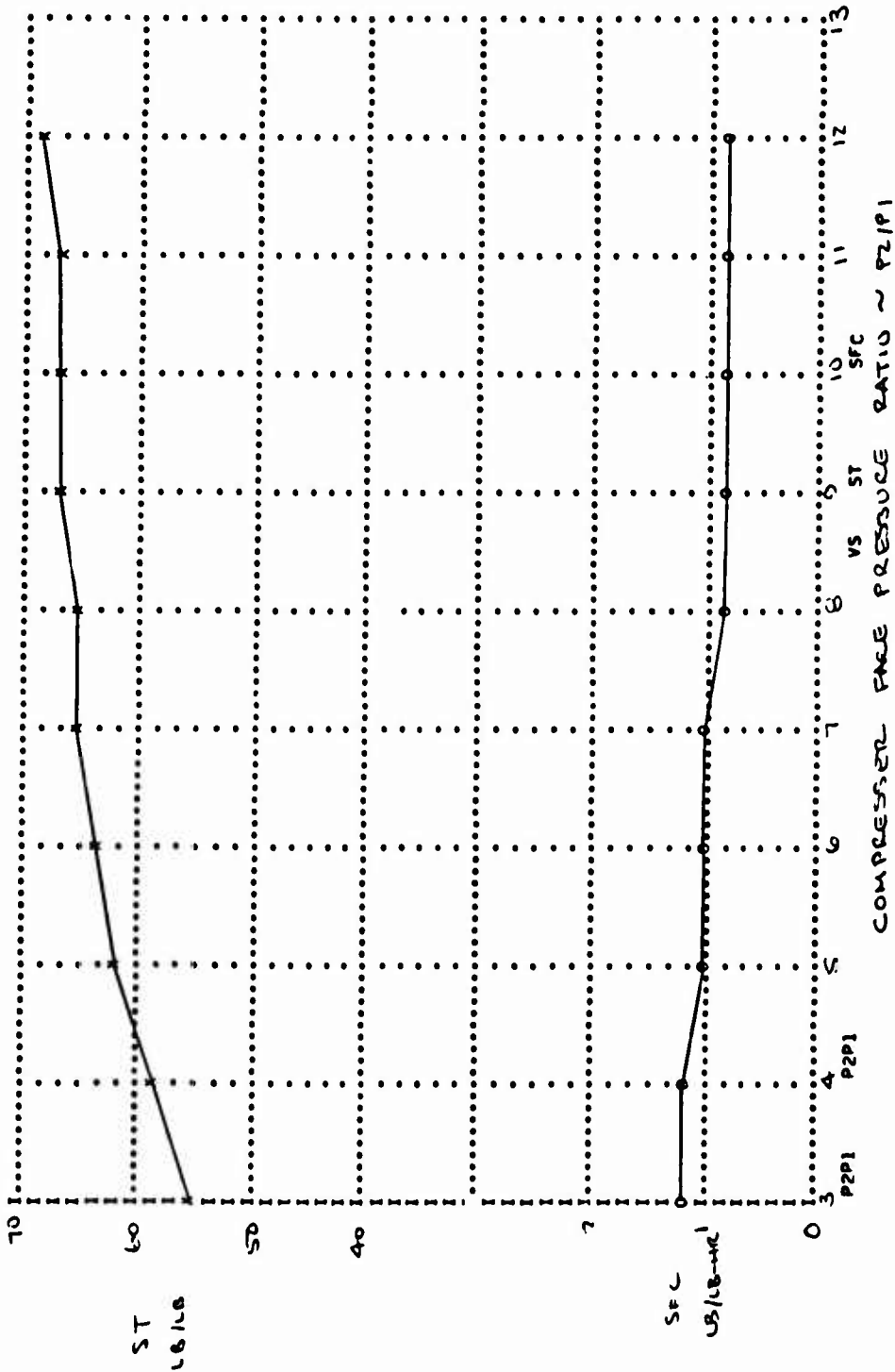
BEST CYCLE# 1, THE 1TH RECORD ON FILE OUTSAV. N= 1

END OF ODIN/RLV SUMMARY REPORT

Figure 11. Example Input and Output for Report

AEROPHYSICS RESEARCH CORPORATION PLOT PROGRAM

DRY TURBOFAN ENGINE 8=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

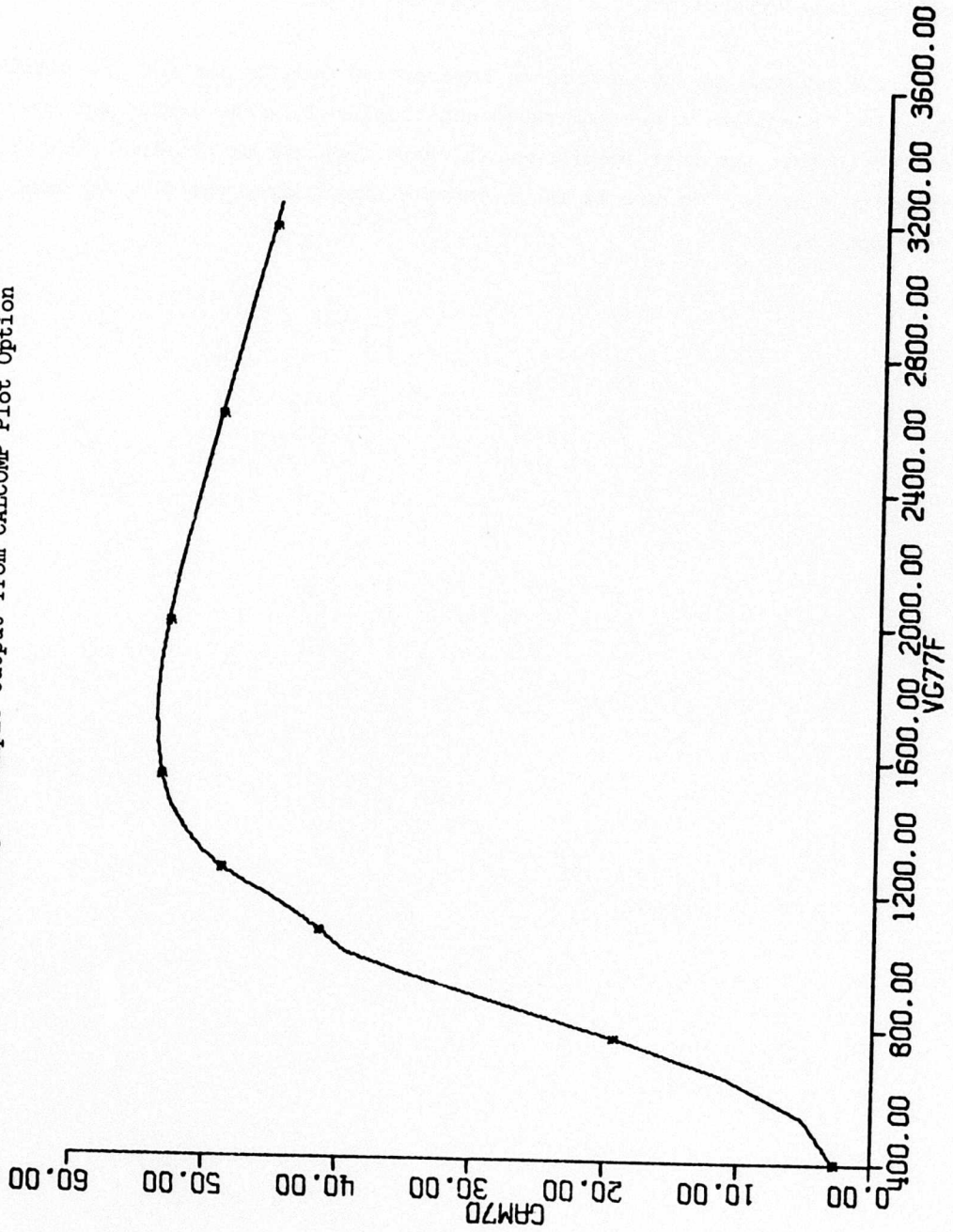
FRAME 4

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



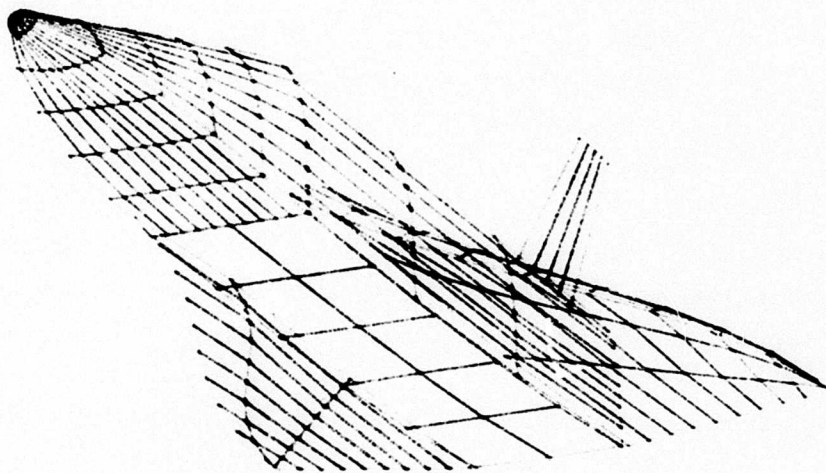


Figure 14. Example of CDC 250 Hard Copy Option

5. DIALOG USAGE EXAMPLES

The ODIN simulation concept requires only five control cards to initiate regardless of the type of synthesis to be performed.

```
JOB-----  
USER-----  
FETCH(A3682,SPR---,BOTH,ODINRLV,CCDATA)  
CCLINK(ODINRLV)  
789                      end of record
```

All program control is handled through the DIALOG *control directives*. All data intercommunication is handled through DIALOG *communication commands*. These DIALOG functions are discussed in Section 4.

Five sample cases using the DIALOG 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 DIALOG 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 DIALOG 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 'INITIAL 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

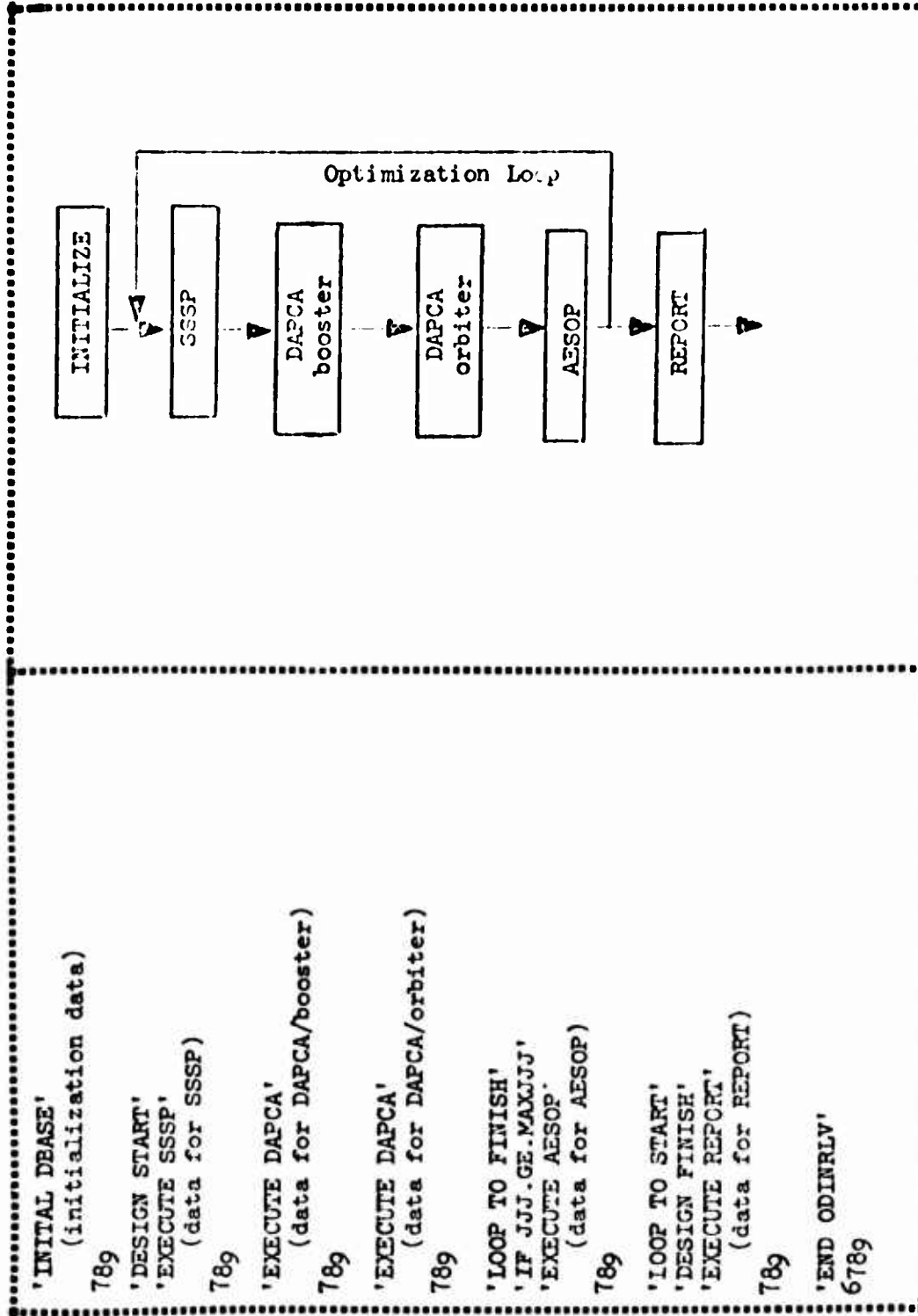


Figure 15. A Sample Optimization Using ODIN

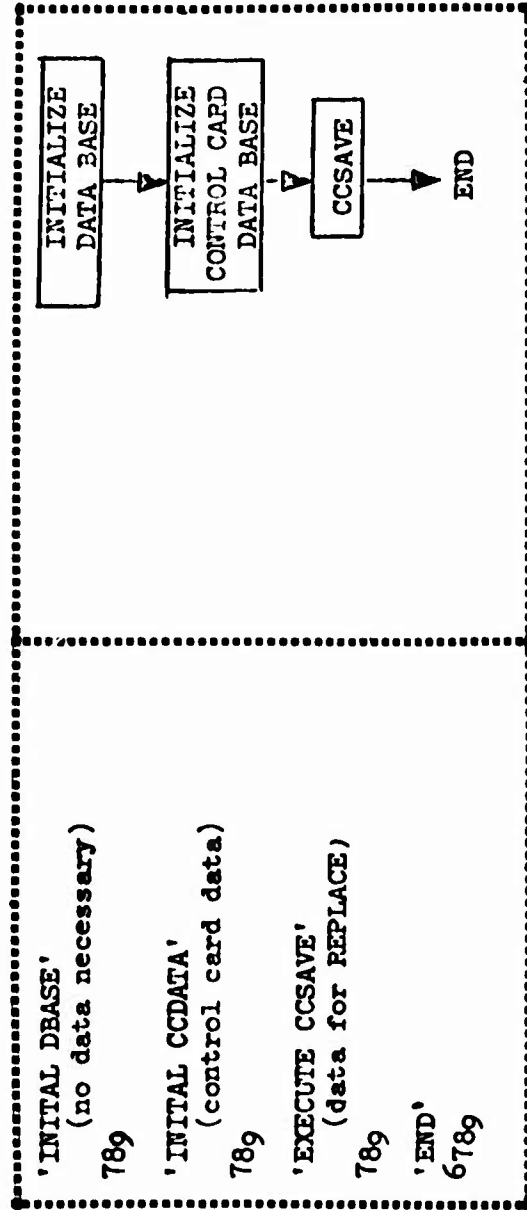


Figure 16. Example of Construction and Storage of Control Card Data Base

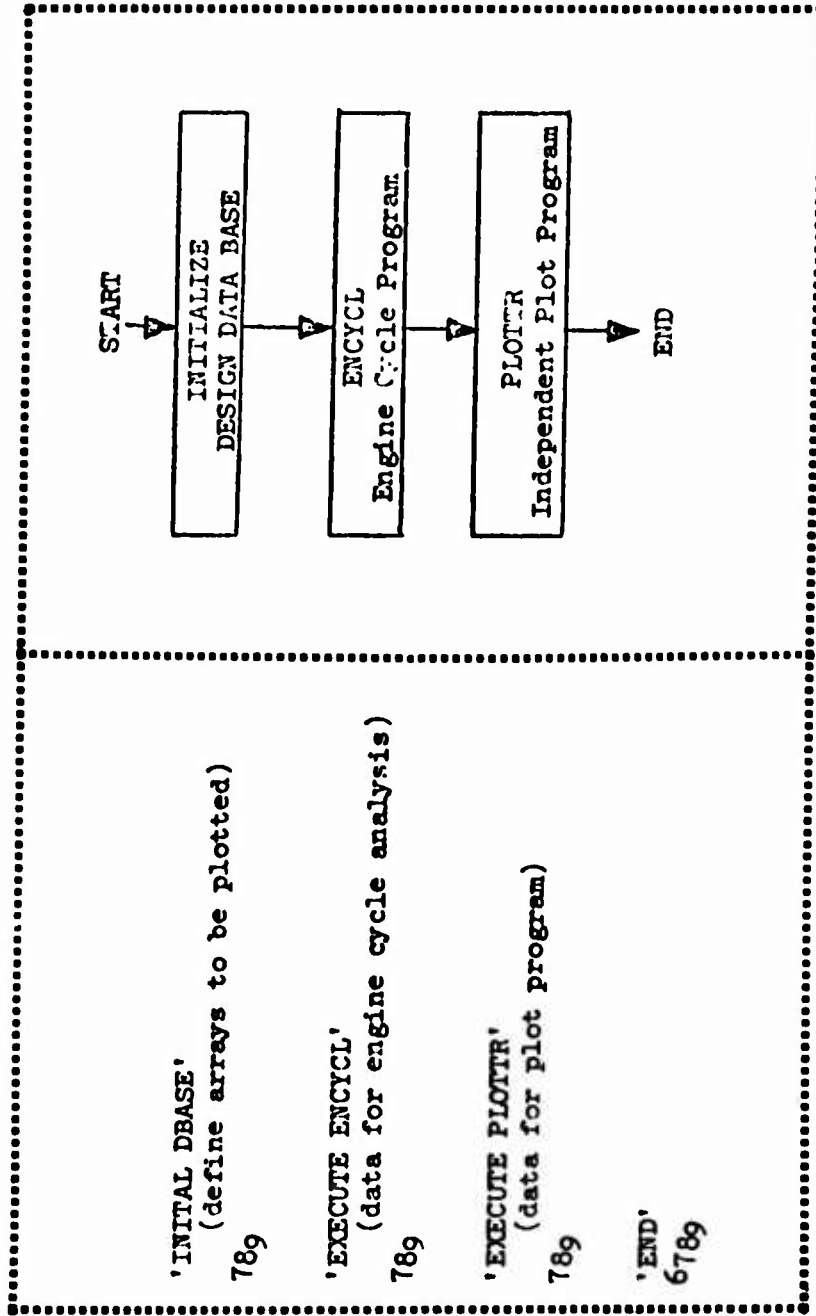


Figure 17. Graphical Presentation of Data from Any Analysis Program

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 DIALOG 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 DIALOG 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

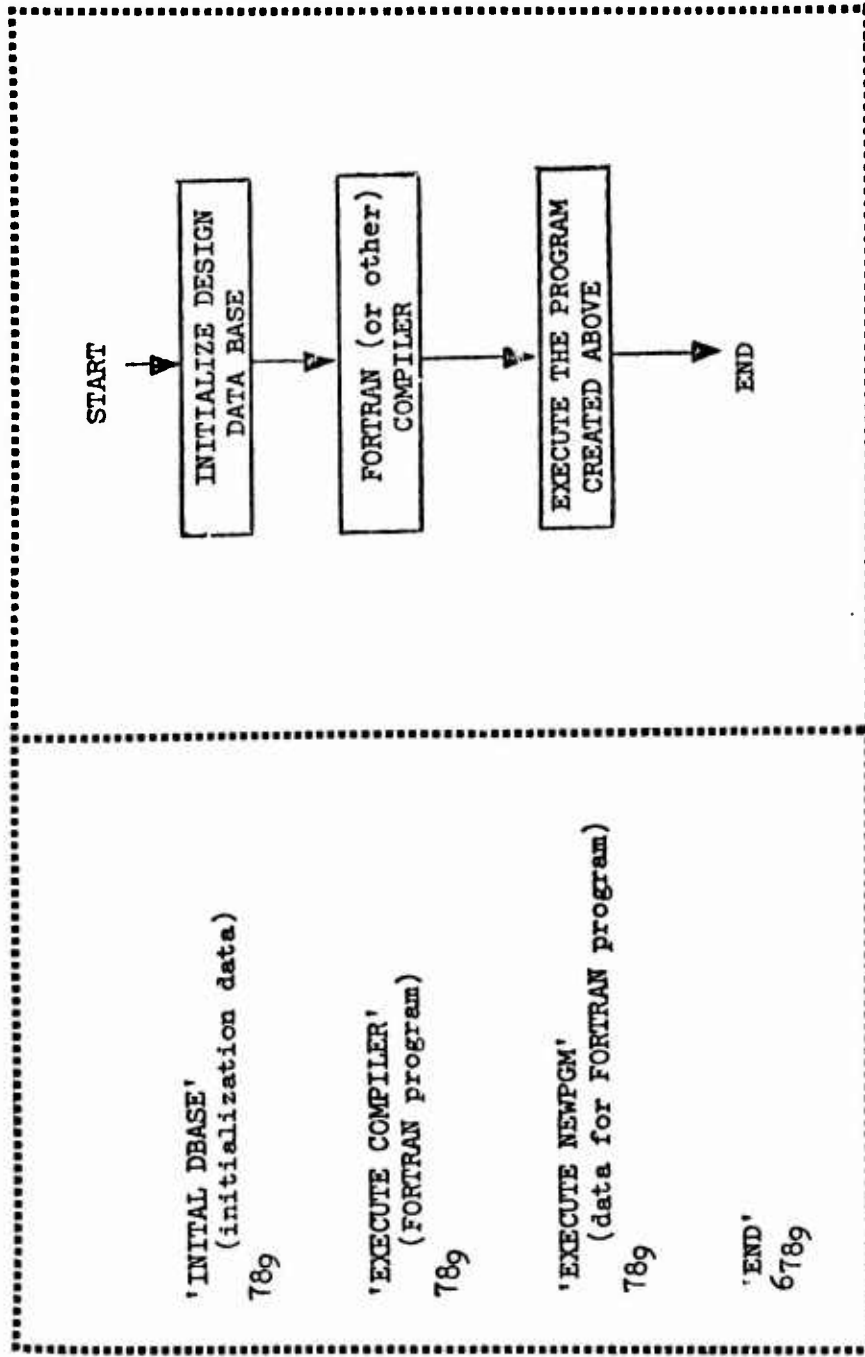


Figure 18. Use of FORTRAN Language to Augment Analysis

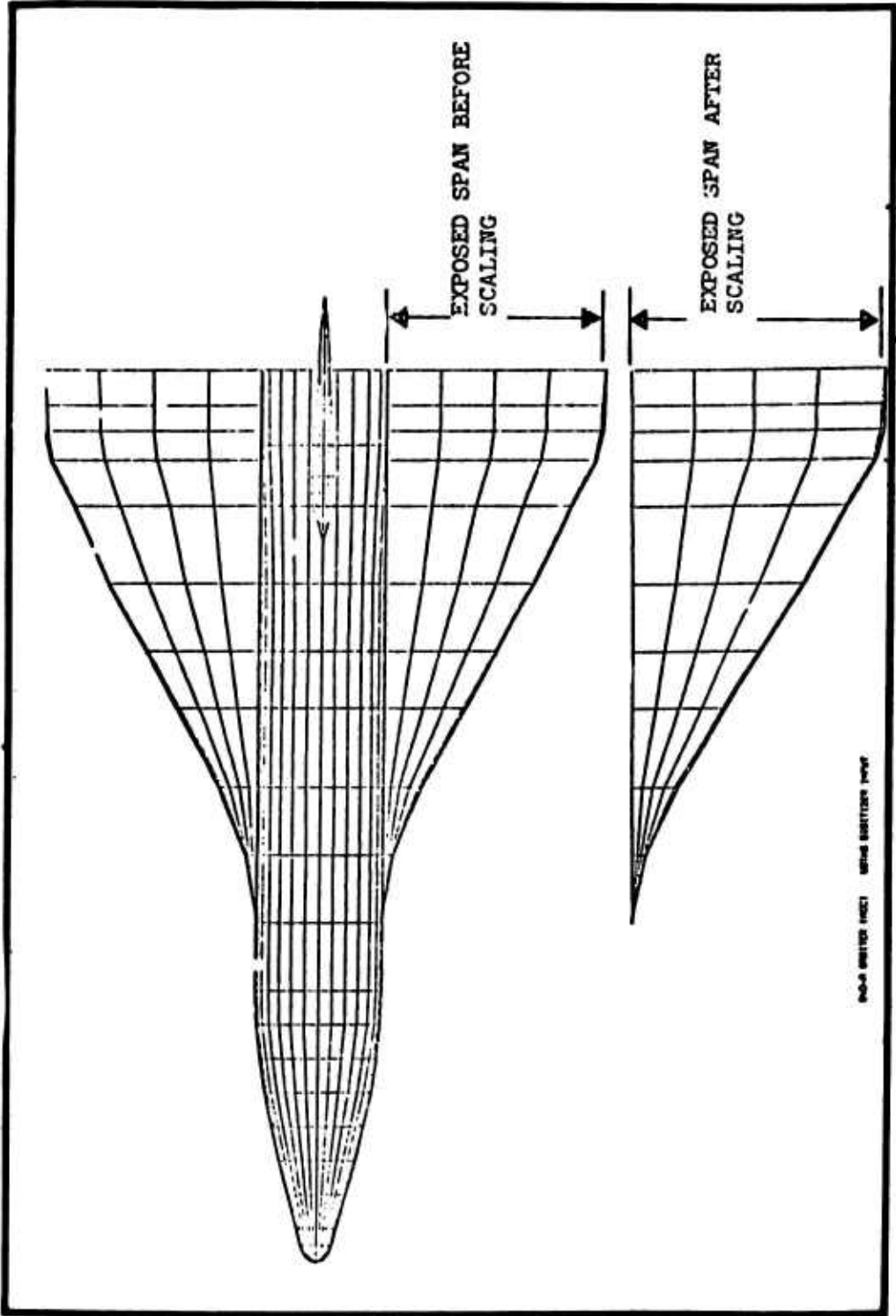


Figure 19. Geometry Perturbation for the O40A Orbiter

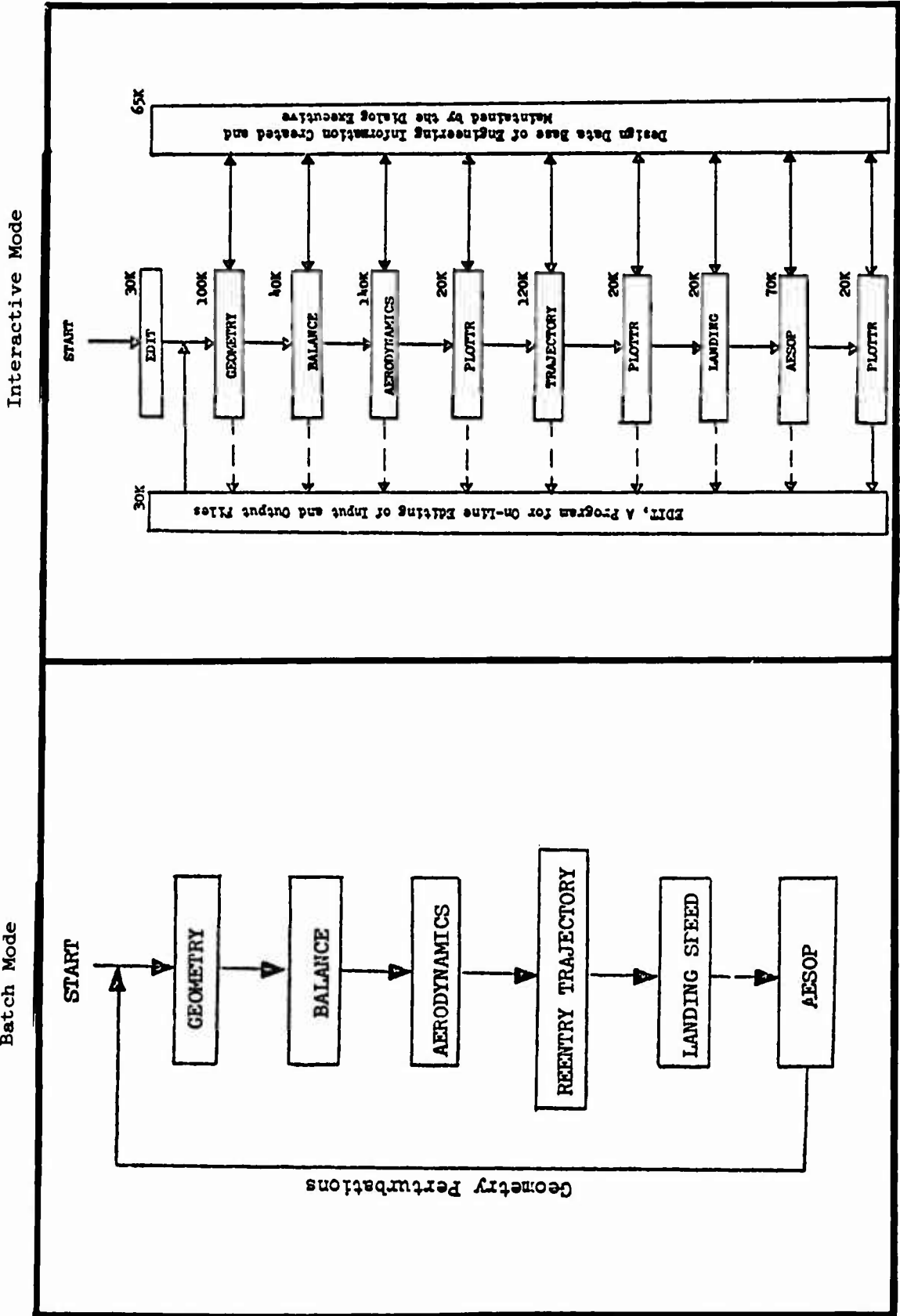


Figure 20. Comparison of Batch and Interactive Modes

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. DIALOG makes extensive use of RANDAC for communicating information to and from the data base.

1. DATA BASE SIZE

The DIALOG 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 DIALOG 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 locked 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 <i>chain</i> 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

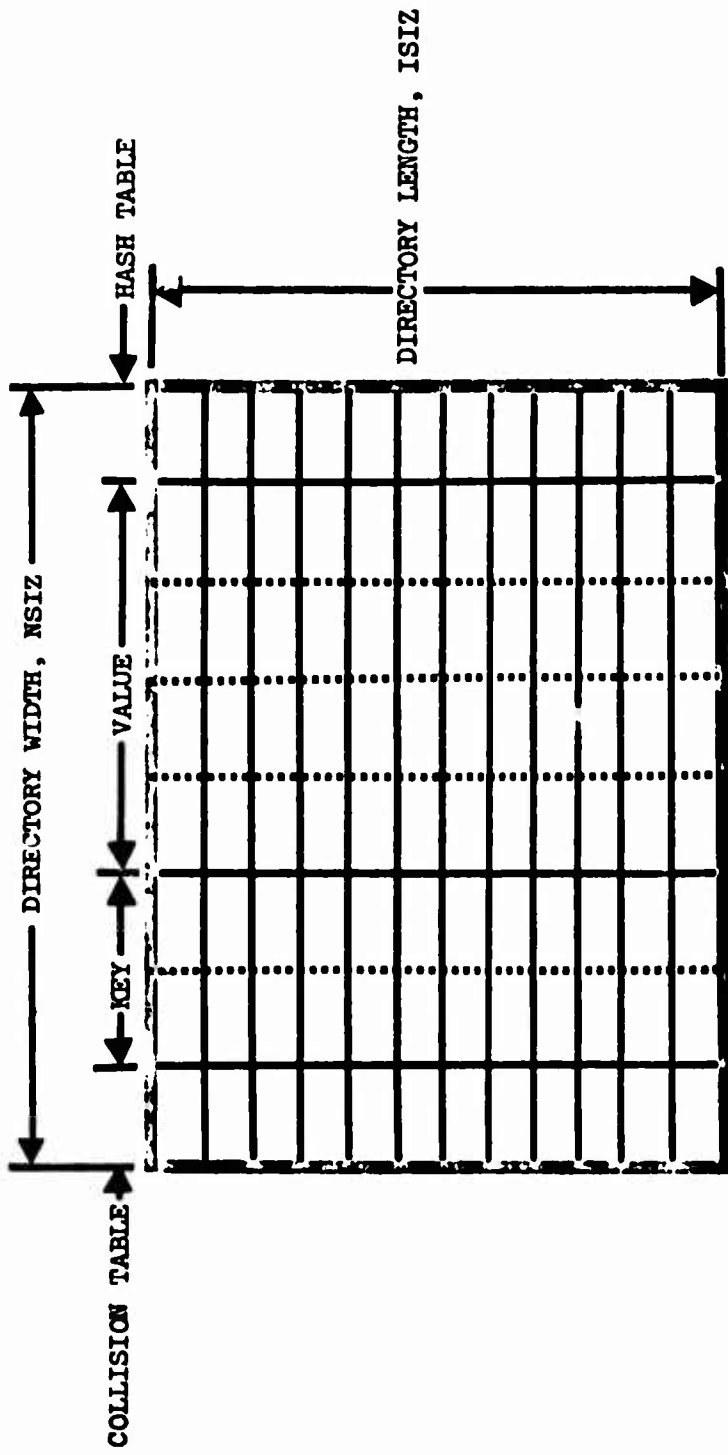


Figure A1. Composition of the Directory, FSL

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 DIALOG 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 DIALOG 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 DIALOG 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 | <div style="border: 1px solid black; padding: 5px; width: fit-content;"> </div> | | Control cards required to access, load and execute one or more user programs. |
| | | SETINDEX,+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 LAJ 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 DIALOG. 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.

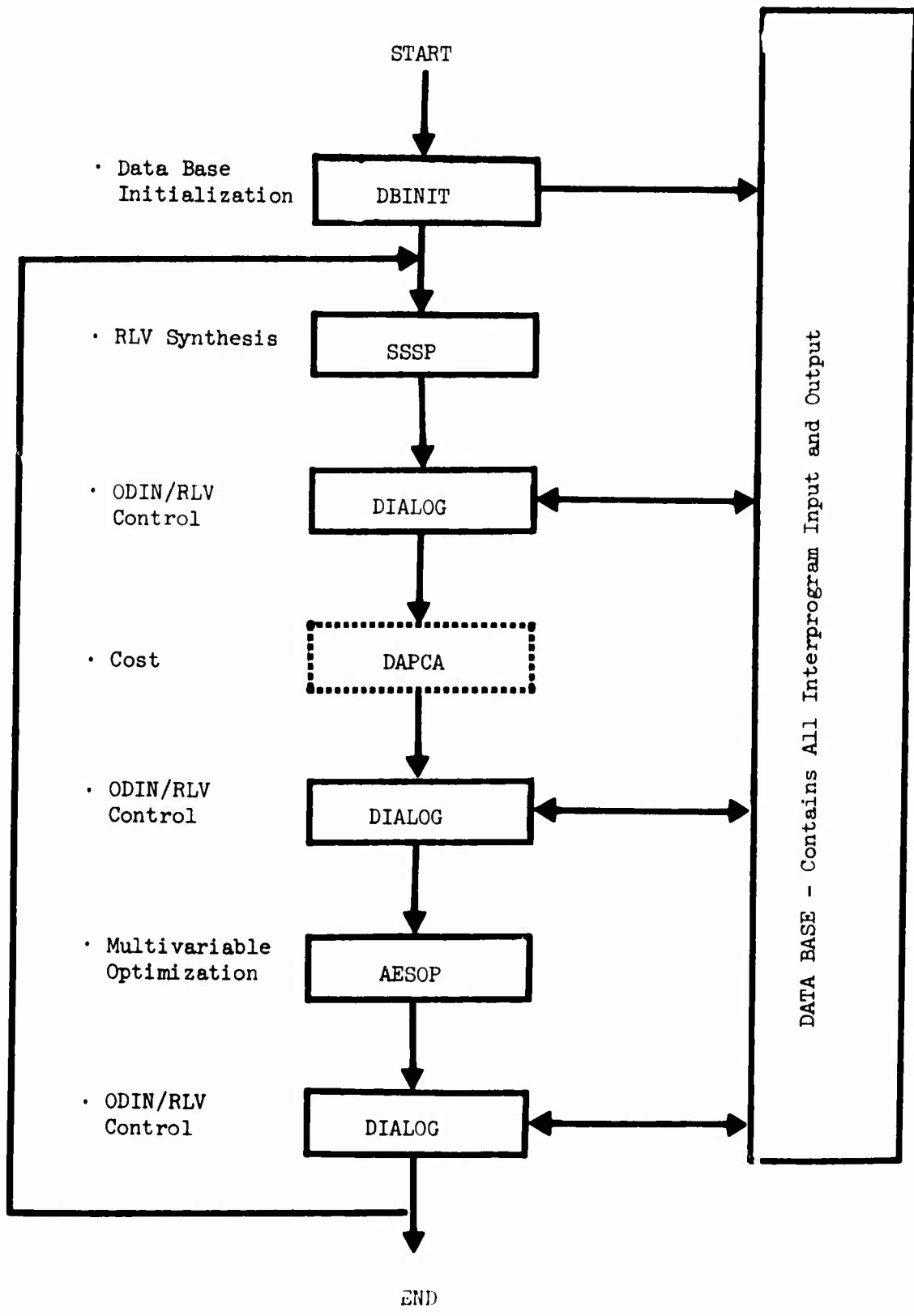


Figure 1. Schematic of Sample ODIN/RLV Model

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 DIALOG system described in Reference C4.

DIALOG 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 DIALOG 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 DIALOG system for forming very large synthesis programs.

3. The CDIN/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.

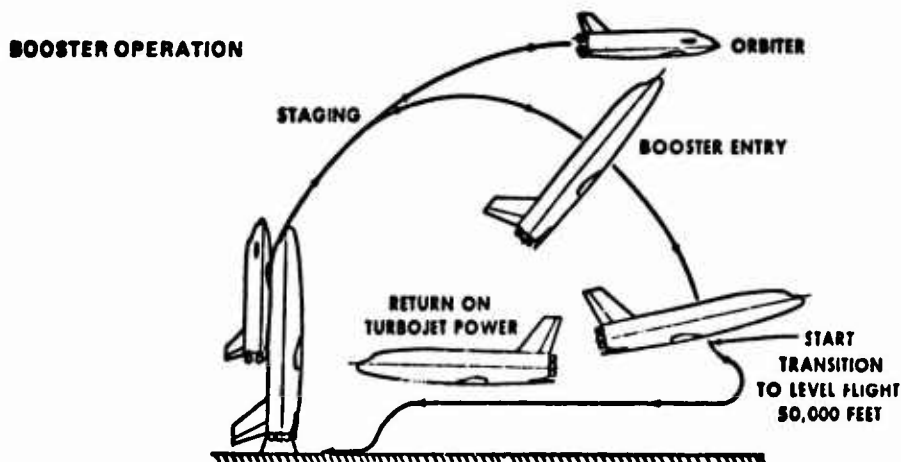


Figure 2. The Two-Stage Space Shuttle Concept

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 the present 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 aq 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 LO₂/LH₂ 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 wing 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/ft² on the upper surface and greater than 1.75 lb/ft² 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 LH₂ 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 ft³. 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 specified 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.

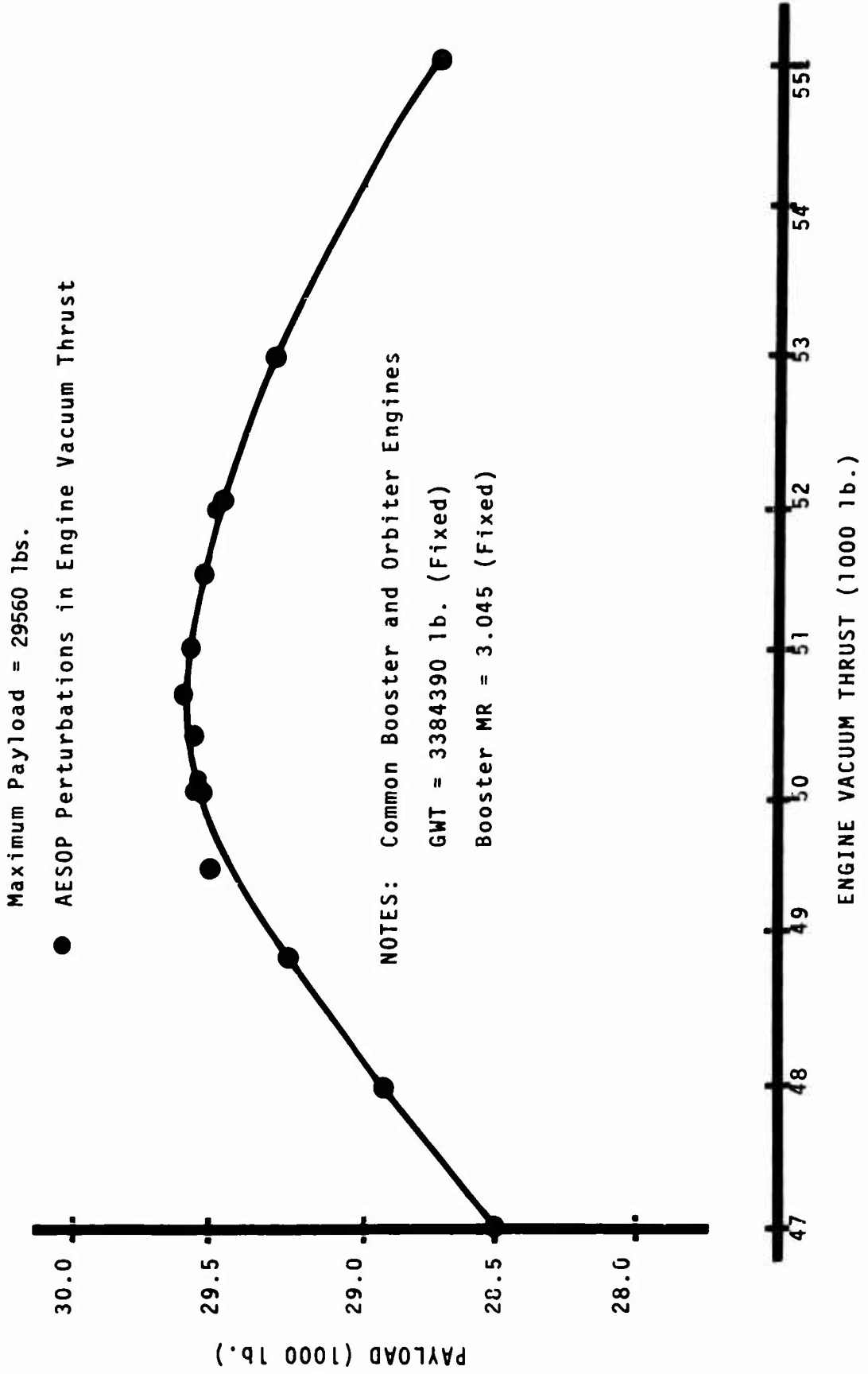
4.1 Selection of Engine Vacuum Thrust.

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

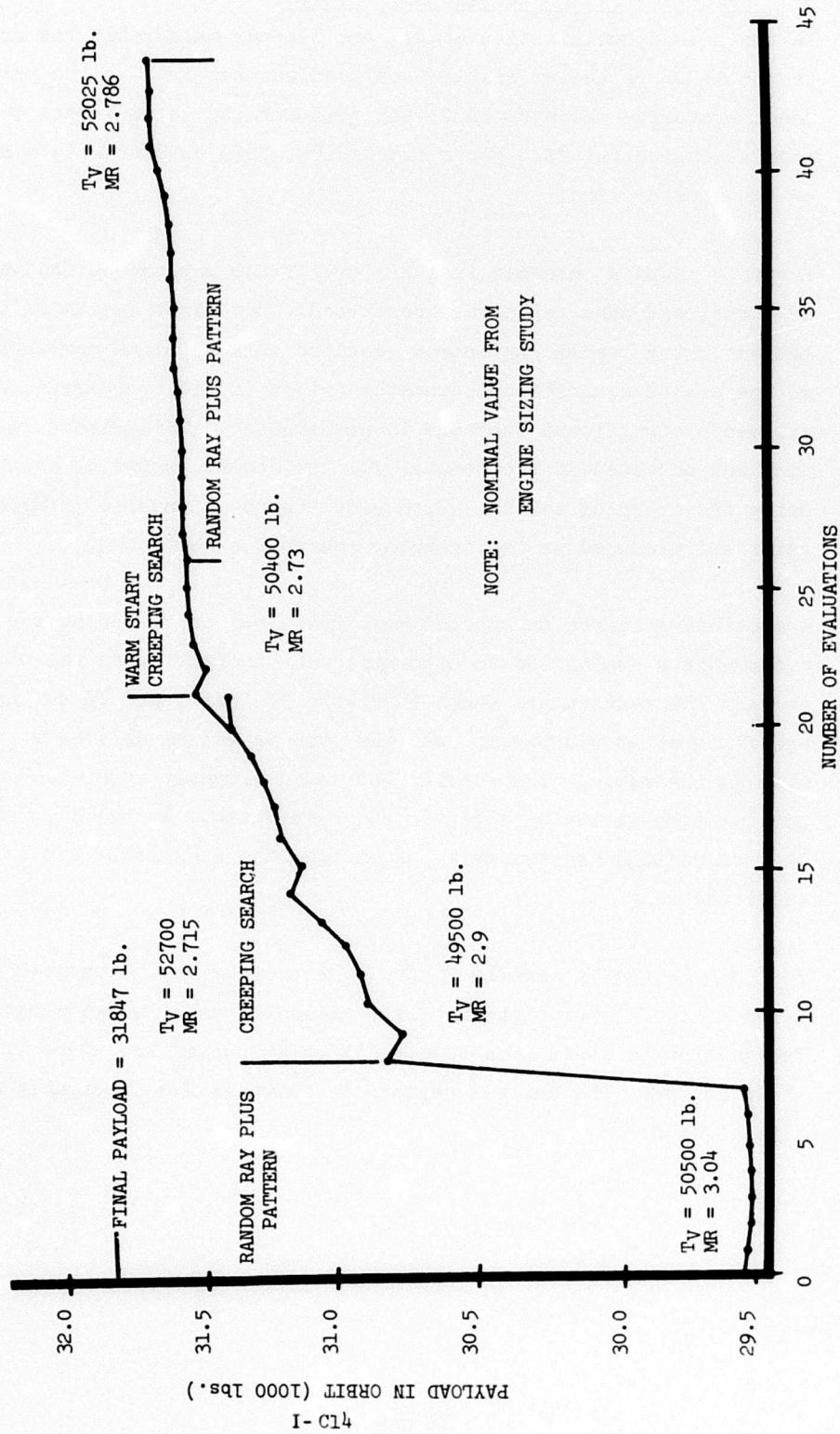


FIGURE 5

BOOSTER SIZING STUDY

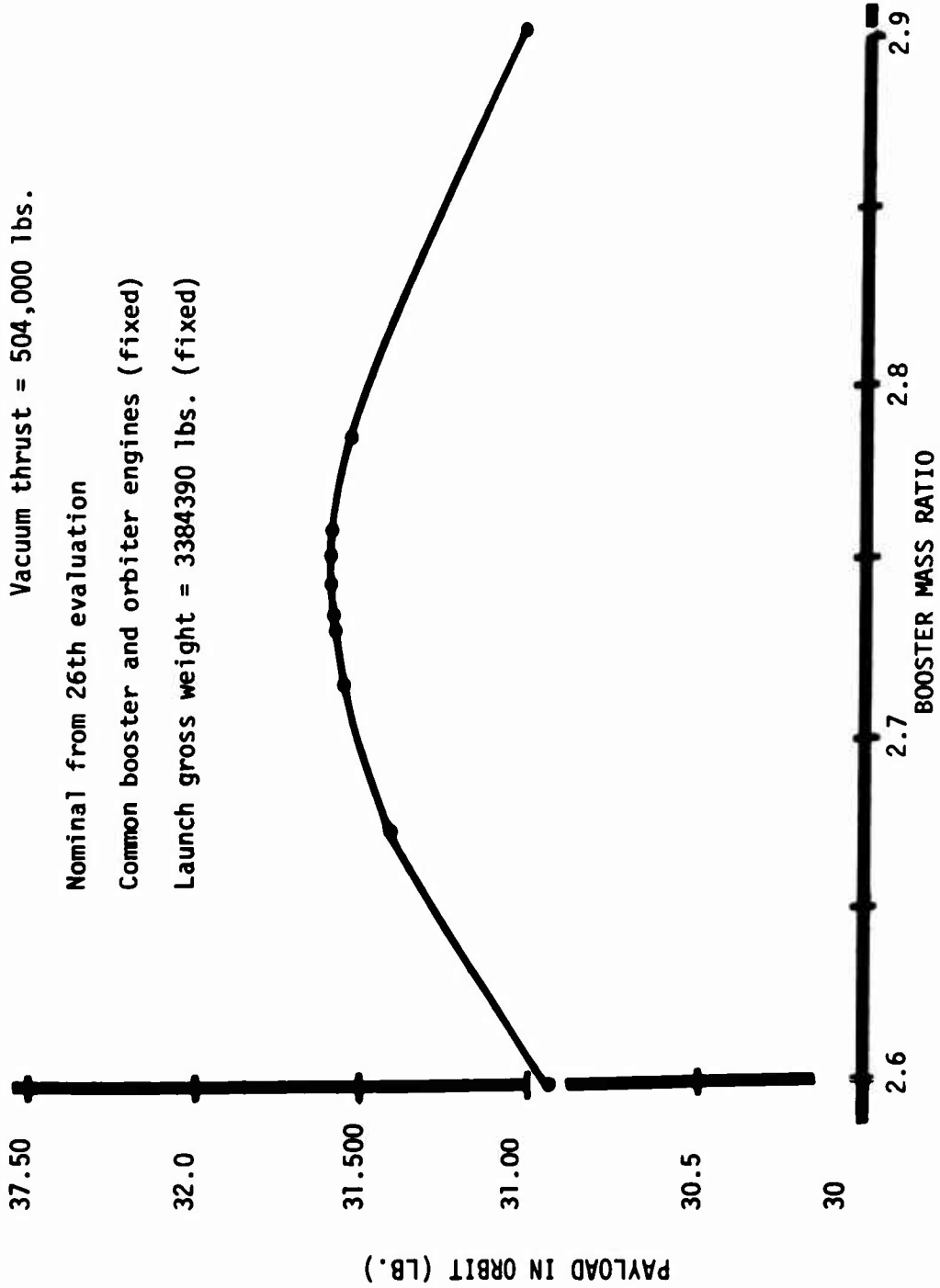
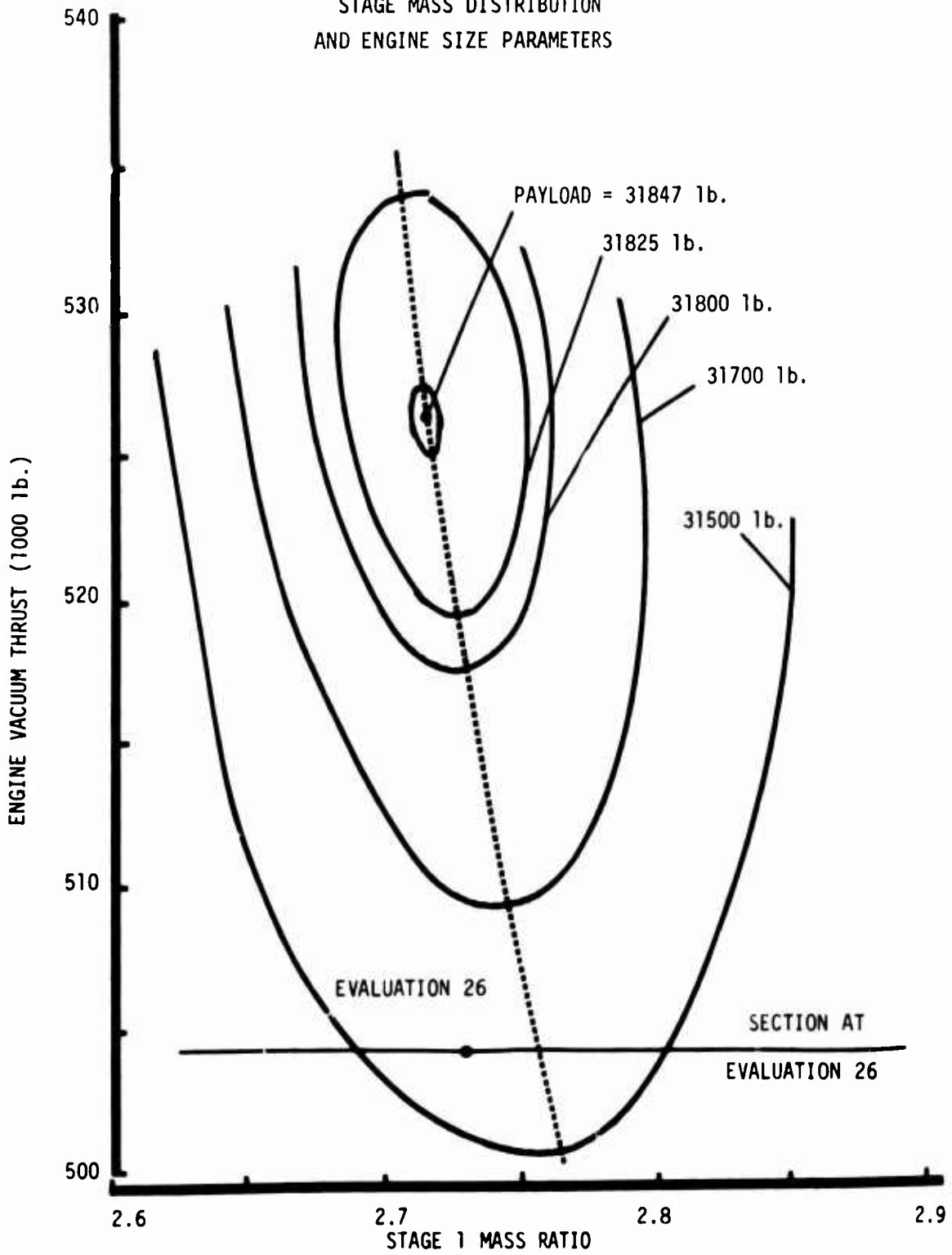


FIGURE 6.
PAYLOAD CONTOURS
STAGE MASS DISTRIBUTION
AND ENGINE SIZE PARAMETERS



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.

| TVAC 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, DIALOG *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 DIALOG merges it in the data base. AESOP extracts the payload and generates parameter perturbations, and DIALOG 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 DIALOG, 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@,

DIALOG 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 DIALOG 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 DIALOG 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 DIALOG. 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 DIALOG 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. DIALOG 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 DIALOG. DIALOG 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 DIALOG

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 |
|---------|----------|-----------|----------------------|---------|-------------------------------|
| AFSOP | 39 | 1 | | DEFINE | AESOP USED IN SIMULATION |
| ALPHA | 7 | 2 | 525000. 2.71 | INITIAL | AESOP CONTROL PARAMETERS |
| BUILD | 19 | 1 | 0 | DEFINE | DYNAMIC DR BUILD OPTION |
| CRDSKP | 1 | 1 | 250 | ADD | CARD SKIP OPTION |
| DRDUMP | 31 | 1 | | DEFINE | CAUSFS DATA BASE DUMP |
| DISPLAY | 13 | 1 | WPAYLO | INITIAL | SCOPE DISPLAY FUNCTION |
| ELTIME | 23 | 1 | 4.30200000 | DEFINE | TOTAL ELAPSED TIME |
| JJREST | 25 | 1 | | DEFINE | BEST PERFORMANCE EVALUATION |
| JJJ | 17 | 1 | | DEFINE | AFSOP EVALUATION COUNTER |
| MIXR | 3 | 1 | 2.7149999999989200 | ADD | PROPELLANT MIXTURE RATIO |
| RUNID | 21 | 1 | *00IN JAN 22. 72 | DEFINE | IDENTIFICATION FOR DIALOG RUN |
| TCOSTR | 35 | 1 | | DEFINE | TOTAL COST OF BOOSTER |
| TCOSTH | 37 | 1 | | DEFINE | TOTAL COST OF ORBITER |
| THRUST | 5 | 1 | 525000.0000000000000 | ADD | ENGINE VACUUM THRUST |
| VSTGR | 29 | 1 | | DEFINE | BOOSTER STAGING VELOCITY |
| VSTGO | 33 | 1 | | DEFINE | ORBITER STAGING VELOCITY |
| WGROSR | 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. | INITIAL | AESOP CONTROL PARAMETERS |
| | | | 2.71 | | |
| BUILD | 19 | 1 | 0 | DEFINE | DYNAMIC DR BUILD OPTION |
| CRD5KP | 1 | 1 | 250 | ADD | CARD SKIP OPTION |
| DRDUMP | 31 | 1 | | DEFINE | CAUSES DATA RASE DUMP |
| DISPLAY | 13 | 1 | WPAYLO | INITIAL | SCOPE DISPLAY FUNCTION |
| FLTIME | 23 | 1 | 4.30200000 | DEFINE | TOTAL ELAPSED TIME |
| JJREST | 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 | DEFINE | IDENTIFICATION: FOR DIALOG RUN |
| TCOSTB | 35 | 1 | JAN 22. 72 | DEFINE | TOTAL COST OF BOOSTER |
| TCOSTN | 37 | 1 | | DEFINE | TOTAL COST OF ORBITER |
| THRUST | 5 | 1 | 525000.00000000000000 | ADD | ENGINE VACUUM THRUST |
| VSTGR | 29 | 1 | 0.91501785518072E+04 | WTOUT | BOOSTER STAGING VELOCITY |
| VSTGO | 33 | 1 | 0.25853097798483E+05 | WTOUT | ORBITER STAGING VELOCITY |
| WGROSR | 11 | 1 | 0.26662821532874E+07 | WTOUT | GROSS WEIGHT OF BOOSTER |
| WGROSO | 15 | 1 | 0.71802039902725E+06 | WTOUT | GROSS WEIGHT OF OPBITER |
| WPAYLO | 27 | 1 | 0.3184576197751E+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 |
|---------|----------|-----------|----------------------|---------|-------------------------------|
| AFSOP | 39 | 1 | | DEFINE | AESOP USED IN SIMULATION |
| ALPHA | 7 | 2 | 525000. 2.71 | INITIAL | AESOP CONTROL PARAMETERS |
| BUILD | 19 | 1 | 0 | OFFINE | DYNAMIC DR BUILD OPTION |
| CROSSP | 1 | 1 | 85 | ADD | CARD SKIP OPTION |
| DRDUMP | 31 | 1 | | DEFINE | CAUSES DATA BASE DUMP |
| DISPLAY | 13 | 1 | WPAYLO | INITIAL | SCOPE DISPLAY FUNCTION |
| ELTIME | 23 | 1 | 25.6240000 | OFFINE | TOTAL ELAPSED TIME |
| JJREST | 25 | 1 | | OFFINE | BEST PERFORMANCE EVALUATION |
| JJJ | 17 | 1 | | OFFINE | AESOP EVALUATION COUNTER |
| MIXP | 3 | 1 | 2.714999999999985200 | ADD | PROPELLANT MIXTURE RATIO |
| FUNID | 21 | 1 | *OCIN JAN 22, 72 | OFFINE | IDENTIFICATION FOR DIALOG RUN |
| TCOSTA | 35 | 1 | | OFFINE | TOTAL COST OF BOOSTER |
| TCOSTN | 37 | 1 | 0.48673855937808E+05 | DAPOUT | TOTAL COST OF ORBITER. |
| THRUST | 5 | 1 | 525000.000000000000 | ADD | ENGINE VACUUM THRUST |
| VSTGR | 29 | 1 | 0.91501786518072E+04 | WTOUT | BOOSTER STAGING VELOCITY |
| VSTGO | 33 | 1 | 0.25853097798483E+05 | WTOUT | ORBITER STAGING VELOCITY |
| WGROSS | 11 | 1 | 0.26672821532874E+07 | WTOUT | GROSS WEIGHT OF BOOSTER |
| WGROSSO | 15 | 1 | 0.71802039902725E+06 | WTOUT | GROSS WEIGHT OF ORBITER |
| WPAYLO | 27 | 1 | 0.3184576197751E+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 | | OFFINE | AESOP USED IN SIMULATION |
| ALPHA | 7 | 2 | 525000. 2.71 | INITIAL | AESUP CONTROL PARAMETERS |
| BUILD | 19 | 1 | 0 | DEFINE | DYNAMIC DR BUILD OPTION |
| CROSSK | 1 | 1 | 85 | ADD | CARD SKIP OPTION |
| CRDUMP | 31 | 1 | | OFFINE | CAUSES DATA BASE DUMP |
| DISPLAY | 13 | 1 | WPAYLO | INITIAL | SCOPE DISPLAY FUNCTION |
| ELTIME | 23 | 1 | 26.8380000 | OFFINE | TOTAL ELAPSED TIME |
| JJREST | 25 | 1 | | DEFINE | BEST PERFORMANCE EVALUATION |
| JJJ | 17 | 1 | | OFFINE | AESOP EVALUATION COUNTER |
| MTXR | 3 | 1 | 2.714999099999989200 | ADD | PROPELLANT MIXTURE RATIO |
| FUNID | 21 | 1 | *00IN JAN 22, 72 | DEFINE | IDENTIFICATION FOR DIALOG RUN |
| TCOSTR | 35 | 1 | 49673.8559372066565H | ADD | TOTAL COST OF BOOSTER |
| TCOSTN | 37 | 1 | 0.19037115923077E+05 | DAPOUT | TOTAL COST OF ORRITER |
| THRUST | 5 | 1 | 525000.000000000000 | ADD | ENGINE VACUUM THRUST |
| VSTGR | 29 | 1 | 0.91501786518072E+04 | WTOUT | BOOSTER STAGING VELOCITY |
| VSTSO | 33 | 1 | 0.25853097798483E+05 | WTOUT | ORBITER STAGING VELOCITY |
| WGROSH | 11 | 1 | 0.25662821532874E+07 | WTOUT | GROSS WEIGHT OF BOOSTER |
| WGROSO | 15 | 1 | 0.71802039902725E+06 | WTOUT | GROSS WEIGHT OF ORRITER |
| WPAYLO | 27 | 1 | 0.3184576197751E+05 | WTOUT | PAYLOAD OF THE LAUNCH SYSTEM |

Figure 10. Data Base Configuration After DAFCA (Orbiter)

***** THERE ARE CURRENTLY 19 DATA BASE ENTRIES AS FOLLOWS *****

| NAME | LOCATION | DIMENSION | CURRENT VALUE(S) | ORIGTN | DESCRIPTION |
|---------|----------|-----------|----------------------|---------|-------------------------------|
| AESOP | 39 | 1 | 0.525E+06 | DEFINE | AESOP USED IN SIMULATION |
| ALPHA ● | 7 | 2 | 0.2715E+01 | AESOUT | AESOP CONTROL PARAMETERS |
| BUILD | 19 | 1 | 0 | DEFINE | DYNAMIC DB BUILD OPTION |
| CARDSKP | 1 | 1 | 32 | ADD | CARD SKIP OPTION |
| CAUSEP | 31 | 1 | | DEFINE | CAUSES DATA BASE DUMP |
| DISPLAY | 13 | 1 | WPAYLO | INITIAL | SCOPE DISPLAY FUNCTION |
| FLTIME | 23 | 1 | 28.0920000 | DEFINE | TOTAL ELAPSED TIME |
| JJREST | 25 | 1 | 1 | AESOUT | BEST PERFORMANCE EVALUATION |
| JJJ | 17 | 1 | 0 | AESOUT | AESOP EVALUATION COUNTER |
| MIXR | 3 | 1 | 2.714999999999989200 | ADD | PROPELLANT MIXTURE RATIO |
| PRUNID | 21 | 1 | *ODIN JAN 22. 72 | DEFINE | IDENTIFICATION FOR DIALOG RUN |
| TCOSTA | 35 | 1 | 48673.85593780655658 | ADD | TOTAL COST OF BOOSTER |
| TCOSTN | 37 | 1 | 0.19037115923077E+05 | DA'OUT | TOTAL COST OF ORBITER |
| THRUST | 5 | 1 | 525000.000000000000 | ADD | ENGINE VACUUM THRUST |
| VSTGR | 29 | 1 | 0.91501786512072E+04 | WTOUT | BOOSTER STAGING VELOCITY |
| VSTSO | 33 | 1 | 0.25853097798432E+05 | WTOUT | ORBITER STAGING VELOCITY |
| WGROSSR | 11 | 1 | 0.26662821532874E+07 | WTOUT | GROSS WEIGHT OF BOOSTER |
| WGROSSO | 15 | 1 | 0.71802039902725E+06 | WTOUT | GROSS WEIGHT OF ORBITER |
| WPAYLO | 27 | 1 | 0.3184776197751E+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 DIALOG 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 , Appendix I-C

- C1. ———., Space Shuttle Synthesis Program (SSSP), Volume I, Part 1, Report GDC-DBB70-002, General Dynamics Corporation, December 1970.
- C2. Hague, Donald S. and Glatt, C. R., An Introduction to Multi-variable 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. and Watson, D. A., Introduction to the DIALOG System: A Concept of Multiple Program Execution with Mutual Data Intercommunication, Aerophysics Research Corporation TN-108, August 1971.
- C5. Wilcox, Darrell E., A Model for Estimating Total Program Cost of Aircraft, Spacecraft, and Reusable Launch Vehicles, Ames Research Center, Working Paper MA-71-3, August 1971.