AEROSPACE REPORT NO.
ATR-76(7361)-1, VOL IV
(Formerly ATR-74(7341)-6)

DRA

Manned Systems Utilization Analysis (Study 2.1)
Final Report
Volume IV: Program Manual and Users Guide for the LOVES Computer Code

Prepared by STANLEY T. WRAY, JR.
Information Processing Division

1 September 1975

Prepared for OFFICE OF MANNED SPACE FLIGHT
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Washington, D. C

Contract No. NASW-2727

Systems Engineering Operations
THE AEROSPACE CORPORATION

(NASA-CR-145836) MANNED SYSTEMS UTILIZATION ANALYSIS (STUDY 2.1). VOLUME 4: PROGRAM MANUAL AND USERS GUIDE FOR THE LOVES COMPUTER CODE Final Report (Aerospace Corp., El Segundo, Calif.) 151 p HC $6.75 — N76-14847 — Unclas — G3/61 15046

Report No.
ATR-76(7361)-1, Vol IV
(Formerly ATR-74(7341)-6)

MANNED SYSTEMS UTILIZATION ANALYSIS (STUDY 2.1)

FINAL REPORT

Volume IV: Program Manual and Users Guide for the LOVES Computer Code

Prepared by

Stanley T. Wray, Jr.
Data Processing Subdivision
Information Processing Division
Engineering Science Operations

1 September 1975

Systems Engineering Operations
THE AEROSPACE CORPORATION
El Segundo, California

Prepared for

OFFICE OF MANNED SPACE FLIGHT
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Washington, D. C.

Contract No. NASW-2727

Report No.
ATR-76(7361)-1, Vol IV
(Formerly ATR-74(7341)-6)

MANNED SYSTEMS UTILIZATION ANALYSIS (STUDY 2.1)

FINAL REPORT

Volume IV Program Manual and Users Guide for the LOVES Computer Code

Prepared by

Stanley T. Wray, Jr.
Staff Engineer
Evaluation and Planning Systems
Data Processing Subdivision

Approved by

L. Sashkin, Director
Data Processing Subdivision
Information Processing Division
Engineering Science Operations

R. R. Wolfe
NASA Study 2.1 Director
Advanced Mission Analysis Directorate
Advanced Orbital Systems Division

FOREWORD

The LOVES computer code was developed to investigate the concept of space servicing operational satellites as an alternative to replacing expendable satellites or returning satellites to earth for ground refurbishment. In addition to having the capability to simulate the expendable satellite operation and the ground refurbished satellite operation, the program is designed to simulate the logistics of space servicing satellites using an upper stage vehicle and/or the earth to orbit shuttle. The program not only provides for the initial deployment of the satellite but also simulates the random failure and subsequent replacement of various equipment modules comprising the satellite. The program has been used primarily to conduct trade studies and/or parametric studies of various space program operational philosophies.

The program was developed in the CDC 6400/7600 computer complex at The Aerospace Corporation, El Segundo, California, for implementation on a UNIVAC 1108 computer. It is coded in SIMSCRIPT 1.5 and FORTRAN IV. SIMSCRIPT (simulation of a program used for design and development purposes) is a simulation language originally developed at the Rand Corporation and now available from Consolidated Analysis Centers, Inc., (C.A.C.I.) in Santa Monica, California. FORTRAN IV (Formula Translation System) is a standard scientific programming language in common use in computer programs.

There are five volumes to this final report which are as follows:

Volume I:	Executive Summary, ATR-76(7361)-, Vol I
Volume II:	Manned Systems Utilization, ATR-76(7361)-1, Vol II
Volume III:	LOVES Computer Simulations, Results and Analyses, ATR-76(7361)-1, Vol III
Volume IV:	Program Manual and Users Guide for the LOVES Computer Code, ATR-76(7361)-1, Vol IV (formerly ATR-74(7341)-6)
Volume V:	Program Listing for the LOVES Computer Code, ATR-76(7361)-1, Vol V (formerly ATR-74(7341)-7)

This volume (Vol IV) is an updated version of the Program Manual and Users Guide. It was issued to incorporate definitions and explanations of the latest modifications that were made to the final version of the program.

Design of the program was initiated by The Aerospace Corporation in FY 74 under Study 2.1, Operations Analysis, Payload Designs for Space Servicing (contract NASW 2575). It was completed in FY 75 under Study 2.1, Manned Systems Utilization Analysis (contract NASW 2727). The NASA Study Director for FY 74 and part of FY 75 was Mr. V. N. Huff, NASA Headquarters, Code MTE. The NASA Study Director for the balance of FY 75 was Dr. J. W. Steincamp, MSFC, Code PD 34.

ACKNOWLEDGMENTS

Many people have participated in the design, implementation, and usage of the LOVES Computer Program. For technical direction, credit is due R. R. Wolfe, NASA Study 2.1 Director for The Aerospace Corporation, and V. N. Huff, NASA Study 2.1 Task Monitor (Code MTE), NASA Headquarters, Washington D. C. Problem definition was provided by J. B. Carey, S. B. Miller, L. T. Stricker, and M. G. Wolfe of The Aerospace Corporation. Programmers responsible for coding the LOVES Computer Program include W. J. Swartwood, G. W. Timpson, and S. T. Wray, Jr. To date, the program has been used primarily by Messrs. Stricker and Wolfe of The Aerospace Corporation, V. N. Huff of NASA, and Dr. J. Steincamp, NASA Manned Space Flight Center, Huntsville, Alabama.

CONTENTS

CONTENTS (cont.)

CONTENTS (cont.)

CONTENTS (cont.)

CONTENTS (cont.)

TABLES

FIGURES

1. INTRODUCTION

This document provides the potential user with the information necessary to use the LOVES Computer Program in its existing state or to modify the program to include studies not properly handled by the basic model. The report is divided into a Users Guide, a Programmers Manual, and several supporting appendices. To achieve a full understanding of the LOVES Computer Program will require at least a perusal of the entire document.

The Users Guide defines the basic elements assembled together to form the model for servicing satellites in orbit. As the program is a simulation, the method of attack is to disassemble the problem into a sequence of events, each occurring instantaneously and each creating one or more other events in the future. The main driving force of the simulation is the deterministic launch schedule of satellites and the subsequent failure of the various modules which make up the satellites.

The user will find a description of the events in the simulation along with the properties of satellite systems, satellites, modules, orbits where satellites "live," and vehicles (upper stages, Shuttles, and the Solar Electric Propulsion Stage). The phasing algorithm is described as it pertains to visiting several payloads positioned at different locations within an orbit. The loading queue is discussed as the means of choosing when a launch shall occur. There is also a discussion on the detail of the data cards contained in the input data deck.

The LOVES Computer Program uses a random number generator to simulate the failure of module elements and therefore operates over a long span of time — typically 10 - 15 years. The sequence of events is varied by making several runs in succession with different random numbers resulting in a Monte Carlo technique to determine statistical parameters of minimum value, average value, and maximum value.

The parameters collected are described in the paragraphs on program output.

The Programmers Manual presents a series of flow charts showing the interconnections between all the subroutines and events which the program comprises. Each subroutine and event is then described in some detail. It should be noted that experience in working with the program has demonstrated that changes can rarely be localized to one routine but rather must be integrated into the entire conglomerate.

Appendix A describes the queues (waiting lines) used to retain a record of satellite launchings, payloads ready to fly (loading queue), and the next event to occur in the simulation.

Appendix B defines the variables used in the LOVES Computer Program and provides some insight into the internal operations.

Appendix C provides a sample printout which should be produced by the original program (the code is in another document) if the user inputs the data included in the appendix.

Appendix D provides a description of various optional features of the program and what action the program will take in response to certain inputs. This is intended as an aid to the user in understanding how the program performs the simulation.

2. USERS GUIDE

This section is intended to acquaint the user with the capabilities of the LOVES Computer Program. The section contains descriptions of the events being simulated; the construction of systems, satellites, and modules; the vehicles used to transfer to orbits; and the algorithms used in the Monte Carlo simulation.

2.1 SIMULATION ELEMENT INTERRELATIONSHIPS

2.1.1. Event Flow

The LOVES Computer Program is a simulation of an on-orbit servicing process. The simulation model has been set up as a series of events with each event having a time of occurrence. A typical sequence of events is shown in the list below:

- BEGIN - initiates the simulation; reads in data to define the span of years for the simulation; causes all data to be loaded; sets up the next event, START; and never occurs again.
- START - initializes data for the Monte Carlo cycle; sets up all required NWSAT events from input data; and sets up the last event, TERM.
- NWSAT - confirms that a satellite is available for launch and is placed in the loading queue and sets up the mandatory launch event, LAUNC.
- LAUNC - signifies that a launch must occur now. The payloads are removed from the loading queue and arrival events, ARRIV, are set up for each payload. The event REFVE is set up. The events REMOV and BACK may be optionally set up.
- ARRIV - indicates that a payload has arrived at its designated position and become operational. The optional events (FAIL, WARN, SATDN, NWSAT, and RETRI) may be set up for each module or the satellite.

REFVE	-	shows that the vehicle has completed flight and refurbishment and is now ready for use. The loading queues are checked for flights not flown because of lack of vehicles (in which case, the functions of LAUNC are performed at this time).
WARN	-	provides notice that a warning of impending module failure has been received. A replacement module is put into the loading queue. Any payload put into the loading queue may create a condition where a vehicle could not be flown (in which case, the functions of LAUNC are performed on a portion of the loading queue). The mandatory launch event, LAUNC, for this module is optionally set up.
FAIL	-	signals that a module failure has occurred. A replacement module is put into the loading queue if a warning has not been received previously. The mandatory launch event, LAUNC, for this module is optionally set up.
QWAIT	-	is a delay event that occurs after a FAIL or WARN event. This event permits the simulation of the delay incurred in removing a replacement from the shelf and putting it through checkout and repair before making it available for launch.
RETRI	-	notes that retrieval of a satellite is scheduled to occur now. The request for retrieval is entered into the loading queue (which will force the REMOV and BACK events).
SATDN	-	is used when a satellite is permanently deactivated at the end of its life.
REMOV	-	indicates the satellite is physically docked with a vehicle as the first step of retrieval.
BACK	-	shows when the satellite has reached the earth's surface.
TERM	-	is the last event to occur. It may decide to terminate the run or reset the simulation clock and set up the event START for another integration through the sequence.

The typical simulation run consists of 15-200 initial setups of the event NWSAT which cascade into arrivals, warnings, and failures at the rate of 1-25 modules/satellite. The SIMSCRIPT system performs the booking function of selecting the new next event to occur, even if two or more occur at the same time. For more detail on options, see the input data description and the Programmers Manual portion of this document.

In addition to the basic sequence of events shown, there are three more options available, although none of them has been required in the simulations performed to date. The three additional events are:

- REFMO - occurs when a module completes refurbishment and is available for reuse.
- REFSA - provides notification of the completion of satellite refurbishment.
- NEWME - confirms that a new mission equipment module is available for launch. The module is placed in the loading queue, and a mandatory launch event, LAUNC, is set up.

2.1.2 Satellite Systems

Satellite systems usually consist of a number of satellites located in various orbits or at various positions around an orbit so that they can be separated in angle. A system can also have all of its satellites in the same orbit and may have them clustered together in the orbit. The user has the option of defining any mix between the two extremes.

If a system is defined as having four satellites at more or less unique positions, the program reserves four satellite positions in orbit as belonging to that system. During a span of say 12 years, the program may satisfy the user's intentions by deploying a total of 15 satellites to the 4 positions. Occasionally, due to an input data error, the summary printout may show no satellites deployed to a specific position in a system.

A system can be said to be operational if some number of the satellites are each operational. This definition can be interpreted to mean a system is operational if all satellites are operational with active spares (the spare can be temporarily operative, inoperative due to module failure or end of satellite life, or not yet deployed).

The system has the property of a lifetime which is measured from the time the first satellite becomes operational. At the end of the system lifetime, all satellites are disabled, and no more deployments should be made to any satellite position in the system.

A low development rate and a limited lifetime for each satellite in the system can lead to a premature termination of the system, thereby yielding a shorter than expected system lifetime. Over the period of the lifetime, statistical information is gathered for evaluation of the system performance in conjunction with whatever policies and constraints the user has imposed on the run. For each system, the minimum, average, and maximum values for the following parameters have been printed as shown in Appendix C:

a. System Total Flights - the total of all the load factors for modules and satellites as chargeable to this system in a Monte Carlo cycle. It is interpreted as the total equivalent flights of the uppermost vehicle in the delivery cycle.

b. Percentage System Available - the percentage ratio of the total of all time intervals that the system was operational to the actual lifetime for each Monte Carlo cycle.

c. Delay Interval to Restore - the interval in days between the moment the system became inoperative and the moment following when the system returned to operational status for each and every outage interval.

2.1.3 Satellites

The satellites are used as members of the systems, and a satellite may be used as many times as necessary in any number of systems. The program retains, as a part of the system information, which satellite is at each position. Thus, the user need only define the satellite once and then may refer to the same data as many times as necessary. For example, on page C-8 in Appendix C, the system NNDIAB contains the satellite NNDIA, NNDIB, NNDIA, and NNDIB. These data defined NNDIA and NNDIB previously, and the system specification makes reference to NNDIA and NNDIB each twice, thereby causing the program to generate two of each of the two satellites at the satellite positions in orbit.

Similarily, the satellites are made up of modules each defined in a module table. Each satellite can use as many modules as necessary, some of them repetitively for the module can be manufactured in lots. The user defines the module once and thereafter can refer to the data by name as many times as necessary. The data for each module on each satellite at each satellite position are maintained by the program.

A satellite can be said to be operational if a single strand module and all redundant subsystems are operational. Redundant subsystems contain active spares only. The satellite has the property of a lifetime which is measured from the time the satellite becomes operational after deployment of an entire satellite. At the end of the satellite lifetime, the satellite and all of its modules are disabled. The satellite position can be reactivated by a new deployment, resulting in a new life span at the satellite position. This feature permits the user to define a long-lived system using short-lived satellites.

Satellite deployments are performed by two features in the program. The user can stipulate via input data the date at which each satellite becomes available for launch. This is required for the initial satellite launch to each satellite position, as the delivery schedule is an integral part of the mission model being simulated. Some satellite systems have holes in operational schedules during which, for budgetary or other reasons, no active satellites are available. Where applicable, the user can request automatic replacement of satellites over the span of the systems. The parameter POLIC is a property of each satellite.

A normal value of 0 (is equivalent to 1) means no replacement. If POLIC is 2 or 3, the replacement satellite becomes available WAIT1 days after termination. If WAIT1 is negative, the new satellite could replace the previous satellite with no system outage. If POLIC is 3 or 4, the old satellite will become available for retrieval after the interval WAIT1 + WAIT2 after the termination. WAIT2 is typically +1 day, depending on the users preference of deployment over retrieval in conjunction with outage.

For each satellite on each system, the minimum, average, and maximum values for the following parameters have been printed as shown in Appendix C:

a. Satellites Deployed - the average number of satellites deployed only.

b. Satellite Deployment Flights - the total of chargeable flights for satellite deployment only.

c. Satellite Total Flights - the total of all the load factors as chargeable to this satellite position in a Monte Carlo cycle.

d. Percentage Satellite Available - the percentage ratio of the total of all time intervals that the satellite was operational to the actual lifetime for each Monte Carlo cycle.

e. Delay Interval to Restore - the interval in days between the moment the satellite position became inoperative and the moment following when the satellite position returned to operational status for each and every outage interval.

2.1.4 Modules

The basic elements of a satellite are modules which can be classified as SRU or NRU. An SRU (space replaceable unit) is a module that is packaged in a single unit and can be physically removed and replaced in the satellite. Examples of SRUs are attitude control units, electrical power systems, sensors, on-board computers, telemetry and data communication units, etc. An NRU (non-replaceable unit) might be the structural shell of the satellite, the wiring harness connecting all modules together electrically, or a replaceable unit mounted in the interior of the satellite where it would be impractical to replace. The major distinction between SRUs and NRUs is that an SRU replacement amounts to delivery of typically 200 pounds whereas an NRU failure forces replacement of the entire satellite amounting to typically 1500-5000 pounds.

Regardless, a module has the properties of weight, lifetime, and the Weibul parameters, α and β, for both failures and warnings. The lifetime is an interval at the end of which the module is inoperative, and in the case of an SRU, it will be replaced. This property is applicable to batteries and propellant bottles which should be replaced at known intervals. A module failure can be deduced from telemetry data or other observations, and similarily a degradation in performance can be observed down to a limiting criteria (warning). No two modules behave the same way, and therefore the program predicts the failure and warning of each module by selecting a random number and transforming it to a time of failure and warning by means of the Weibul function:

$$R = e^{-\left(\frac{t}{\alpha}\right)^{\beta}}$$

where

- t is the time elapsed since the module became operative,
- R is the probability of failure or warning, and
- α and β are properties of the module.

Some modules do not give warnings of degradation (or the user may not wish to respond). This is accomplished by setting the α for warning to zero. Similarily, no failures will occur for modules with the α for failure set to zero. This is important for interplanetary satellites where, for the purposes of determining flights, it is undesirable to attempt to replace a module and the satellite will terminate eventually anyway.

A module on a satellite has an operative or inoperative status. The occurrence of a warning does not affect the module status. A module becomes operational when the satellite is deployed in orbit or when the module is replaced. The occurrence of a failure makes the module inoperative. Whenever a failure or a warning occurs, the replacement is scheduled except when the flight would be too near the end of the life of the satellite (which would be a wasted flight). Again, there is an exception in which an NRU failure or warning can force the replacement of the satellite thereby extending the life at the orbital position.

The program does recognize subsystems as consisting of a redundant set in which n out of m modules are required for the subsystem to be operational ($n < m$). Also, certain non-critical modules (some scientific experiments for example) bear no relationship to the primary functions of the satellite. Failure of these modules does not affect the status of the satellites. All other module outages, single strand and subsystems, can force the satellite to become inoperative.

The launch of the modules is facilitated by a service unit weighing approximately 400 pounds with a length of 8 feet and capable of holding 16 modules. All the numerical values can be found in the sample input in Appendix C. As many service units as necessary will be put on the flight. The flag PDOWN controls the retrieval of modules; if it is 0, all service units and modules are returned to the ground, and, if it is not 0, the items are left as a group at the last orbit position serviced by the flight. The flag EXMOD controls the SRU/NRU replacement; if it is 0, the program is as has been described. If it is 100, SRU failures suddenly act like NRU failures, and, if it is any other value, the NRUs act like SRUs.

The program is to be capable of permitting the user to specify the change of a module known as the mission equipment upgrade. Based on the premise that with the passage of time, better, more reliable units will evolve, the user can stipulate the replacement of a module on a specific satellite in orbit on a specific day.

Two classes of statistics are gathered for modules. Each satellite position in orbit has a string of modules attached to it describing the makeup of the satellite. For each module at each satellite position, the minimum, average, and maximum values are computed for the following parameters: the number of times a replacement module was launched in this Monte Carlo cycle, and the load factors for deployment of the replacement module including proration of the service unit on the flight. The other class is the module table in which the module appears only once. The minimum, average, and maximum values accrued over a Monte Carlo cycle are computed for the number of warnings, failures, and actual replacements.

2.1.5 Orbits

Each satellite is assigned to a specific orbit, and many different satellites can be assigned the same orbit; i.e., geosynchronous. Orbits are identified by a name and have specific properties. Vehicles are assigned to fly transfer trajectories from the earth's surface via Shuttle to near-earth orbit and thence via an upper stage to final orbit or via an upper stage plus the Solar Electric Propulsion Stage (SEPS). The required Δ V for the upper stage to fulfill on the last leg is provided. The Δ V should include any necessary plane changes.

Statistics are gathered for the activity of each orbit. The program will report the average number of flights to the orbit and the average total payload weight deployed on those flights.

2.1.6 Outage and Availability

The LOVES Program measures the parameters of outage and availability at both the system and the satellite levels. Consider the diagram in Figure 1 for the Astronomy 1D Satellite system from the printout on page C-13 of Appendix C. Both satellites and the system

Date	Member 1	Member 2	Action or Event
	ASTID	ASTID	
1/2/83			Both satellites become available for launch.
2/27/83			The two are launched on the same flight.
2/27/83			The two become operational and the system is operational (6 hours lag for flight).
8/20/83			Module TTC5 fails. Member No. 1 inoperative and system inoperative.
11/20/83			Module TTC5 launched and replaced. All elements operational.
0/0/84			All elements retired from operational status due to termination of model.

NOTE: Interval C is when the second member of the system was operational. Due to the nature of the example, intervals A and B are when both the first member and the system were operational.

Figure 1. Diagram of Astronomy 1D Satellite System Activity

became operational at the same time. This is an exception, for generally the members of a system arrive in a staggered order. The module TTC5 failed which resulted in the satellite and the system becoming inoperative. The outage period between intervals A and B is charged to both the satellite and the system as a delay interval to restore (90 days). Member 2 has no delay intervals to restore for this Monte Carlo cycle. Member 2 is 100 percent available in the sample, and both member 1 and the system have an availability of $100 \cdot (A + B) / C$ which is 66 percent.

The more usual situation is one in which several modules on the same satellite may fail at nearly the same time but not be replaced at the same time. The outage for the satellite is measured from the moment the satellite becomes inoperative until the moment it returns to operational status. Subsystems containing redundant module groups tend to improve availability and decrease outage as the satellite may be operative with a failed module in a particular subsystem. The outage on systems is complicated by overlapping inoperative intervals for satellites and by delays in the initial deployment of the necessary number of operative satellites (first arrival starts the system clock).

2.2 OPERATIONS AND PHASING

2.2.1 Vehicles

The LOVES program divides vehicles into three classes and handles each class uniquely.

The Shuttle must be defined on at least one data card. The user can define more than one Shuttle to provide for orbital maneuvering system (OMS) kits or flights to other than the nominal parking orbit. The important parameters of Shuttles are the total weight delivered to orbit, the maximum length of the payload in the bay, and the number of days required for refurbishment.

The upper stage class can consist of one or more of the following typical vehicles: Centaur, transtage, transtage with one or two solid kick stages, tandem Tug, and the full-capability Tug. The important parameters are gross weight of vehicle, dry stage weight, unused propellant, engine I_{sp}, refurbishment time, maximum payload length, number of stages, and solid/liquid stage indicator.

The SEPS class is restricted to only one vehicle via input, and only one is ever active in the simulation (fleet size is one). This feature of the program is implemented but is not completely operational as yet.

Each class of vehicles is treated as an aggregate; that is, if there is a fleet of 10 Centaurs and transtages, then the mix of vehicles can vary within the run from all Centaurs to all transtages as the program moves year by year. The statistics gathered for summation at the end of the run include the number in each class used in each year with a total over all years. The average number of expended upper stages is shown. If a vehicle is the uppermost from earth in the flight sequence, then its class gets credit for the delivery. The printout shows the average flights and average total payload weight delivered to each orbit by each class. If the program is used with small fleet sizes, flight delays are associated with the class of vehicle by showing the average delay for when no vehicle was available for a flight and the percentage of unavailability in each class of vehicle.

2.2.2 Vehicle Modes of Operation

The Shuttle is used to deliver Shuttle-only payloads and payloads requiring the additional performance of an upper stage. For the Shuttle-only flights, the program checks end-to-end packed payloads against the weight and length constraint for the deploy mode only. The down payload traffic is not checked.

The upper stage vehicle can be composed of one to three stages which can be combinations of expendable, reusable, solid, or liquid stages. However, the combination of expendable lower stages with reusable upper stages is not permitted nor can solid lower stages be combined with liquid upper stages.

The program takes a payload group, determines a delivery sequence including phasing maneuvers, and delivers the list of payload weights with corresponding Δ V's to the performance computation routines. These routines use the simple rocket equation to compute the propellant requirements by processing the list in reverse order, resulting in the total propellant required and the total weight of the vehicle, including payloads and propellant. The propellant required is constrained by input to a maximum value, and the gross weight of the vehicle is constrained by the Shuttle capability. The end-to-end packing of payloads on the front end of the upper stage is constrained to a value that permits the upper stage to fit in the Shuttle bay with the payloads.

The program operates on a fly-on-demand basis and does not support multiple payload-vehicle combinations in the Shuttle bay nor does it support the concept of two Shuttle flights for a tandem Tug with payload and on-orbit docking. The only on-orbit activity permitted is the separation of payload (maybe with upper stage) and rendezvous with a returning upper stage with payload.

2.2.3 The SEPS Vehicle

A portion of the program is intended to perform deployment, servicing, and retrieval missions to synchronous equatorial orbit using a Tug-type upper stage with a continuous low-thrust upper stage known as a Solar Electric Propulsion Stage (SEPS). The SEPS ferries payloads back and forth between an intermediate orbit and synchronous orbit and performs the necessary servicing maneuvers in synchronous orbit. The Tug carries payloads between the Orbiter and the intermediate orbit, deploys fully fueled SEPS vehicles, and retrieves exhausted SEPS vehicles when and if required. The SEPS is assumed to operate in the ground-based mode; that is, the SEPS is initially launched with a specified amount of fuel, and, when that fuel is exhausted (or nearly so), the SEPS is either returned to earth for refurbishment or abandoned in space. In this usage, the time and fuel remaining on a SEPS vehicle are monitored, and Tug flights are automatically initiated to launch new SEPS vehicles as required and return expended ones, if required. Missions which are found to be within the capability of the Tug alone are performed without the aid of SEPS, and payloads which cannot be boosted to a circular orbit of at least 8000-nmi altitude by the Tug are not launched, since the SEPS is not allowed to operate in the region of the Van Allen radiation belts (due to rapid solar cell degradation).

This feature is not fully operational at this time. For a description of the equations and logic, see Aerospace Report No. ATR-74 (7341)-2, Operations Analysis (Study 2.1), Program Sepsim (Solar Electric Propulsion Stage Simulation), by T. F. Lang.

2.2.4 The Phasing Algorithm

The Tug is currently being designed to have an on-orbit life of seven days, which is a fixed constraint. The phasing algorithm attempts to provide service at several different points spaced around the orbit but always in the seven-day time frame by controlling the ΔV required for the individual transfer orbits. Thus, a typical trajectory sequence would be an ascent, one or more phasing maneuvers, and a descent. Of course, there are sequences without phasing maneuvers or the descent trajectory. The procedure for computing the phasing is divided into two parts. The phasing sequence is described in the writeup on the Subroutine PASER in the Programmers Manual. The ΔV for the transfer orbit from one orbit position to another is done iteratively, position by position.

$$n = \frac{|\alpha_i|}{T} \cdot D_r + 0.2$$

where n is the number of revolutions in the transfer orbit ($n > o$).

α_i is the i^{th} phase angle,

T_α is the total of the remaining phase angles, and

D_r is the number of days remaining to phase through T.

The 0.2 is a rounding adjustment more or less empirically derived.

$$P_t = P \cdot \left(1 - \frac{\alpha_i}{360 \cdot n}\right)$$

where P_t is the period of the transfer orbit ($P_t < P$ or $P_t > P$).

P is the period of the basic orbit.

$$T_p = P_t \cdot n$$

where T_p is the flight time for the phasing maneuver with no allowance for rendezvous.

The time remaining is reduced by T_p, and if $D_r < -0.5$

days, the flight is declared unfeasible because of the 7-day time constraint. If $P_t < 0.3535\ P$, the flight is declared unfeasible because the transfer orbit is impossible.

$$R_{pt} = R_a \cdot \left(2 \cdot \left(\frac{P_I}{P}\right)^{2/3} - 1\right)$$

where R_{pt} is the perigee radius of the transfer orbit,

R_a is the apogee radius of the basic orbit.

$$V_{cp} = V_{co} \cdot \sqrt{\frac{R_o}{R_{pt}}}$$

where V_{cp} is the perigee velocity of the transfer orbit,

V_{co} is the equivalent circular velocity of the basic orbit, and

R_o is the equivalent circular radius of the basic orbit.

Then

$$\Delta V = 2 \cdot V_{cp} \cdot \left(\sqrt{\frac{R_p}{R_a}} - \sqrt{\frac{2\ R_p}{R_a\ \left(1 + \frac{R_a}{R_p}\right)}}\right)$$

This series of expression equations is done for each position in orbit, and the result is a series of ΔV's or the sequence is not feasible in the seven-day constraint.

2.2.5 The Loading Queue

When a new satellite arrives at the launch facility, it is entered into a loading queue for subsequent scheduling of a flight. The loading queues are ranked by time, thereby assuring that earlier payloads in the queues are the first ones to be flown. Each loading queue is destined for a specific orbit. The program logic does not support a multi-transfer orbit sequence but rather makes a direct ascent to the orbit and then performs phasing maneuvers within that orbit. Therefore, performance computations are restricted to that queue designated for a specific orbit.

A loading queue is permitted to fill up as a rationale of increasing the upper stage load factors. The criteria for emptying the queue are:

a. When a payload is put into the queue, all payloads in the queue are flown by the performance routines on a trial basis. If the flight is possible, the queue is placed in a flight-ready state. If the flight is not possible, the previous group of payloads (which the program remembered) is flown immediately, if the required vehicles are available. This removes some of the payloads from the queue. If the vehicles are not available, the queue remains in the flight-ready state.

b. When a payload is put into the queue, a mandatory launch event is scheduled for 90 days later. (The 90-day period is an input parameter.) When the mandatory event occurs at the end of 90 days, the loading queue is checked to see if the payload is still there. If so, the payload is scheduled to be flown with all other similar waiting payloads based on vehicle performance computations. This usually results in emptying the queue. However, if the vehicles are not available, the payloads in the queue will wait.

c. When a vehicle completes its refurbishment cycle, a check is made of payloads waiting for vehicles. The first group of payloads for which all vehicle elements are suitable will be flown. The queue may or may not be emptied, depending on the circumstances.

A Sortie payload is treated differently. When a Sortie arrives, there is no wait unless a vehicle is not available. Sortie payloads fly alone always, but Sortie payloads can be in the queue with other payloads; however, the other payloads will be ignored when the Sortie is in the queue.

The LOVES Program attempts to launch all payloads, even if it is necessary to expend reusable vehicles to deliver payloads too heavy to deliver in a reusable mode. The increased deployment capability in the expend mode is used to fly a higher payload weight than normal, and other payloads are included in the flight. No retrieved payloads are flown in this flight, but modules plus service units are carried and expended.

2.3 PROGRAM DATA FLOW

2.3.1 Input Data Formats

The data input to the LOVES Program consists of two parts. The first part is what is referred to as SIMSCRIPT initialization data which define the sizes of the arrays used by the program and input some simple constants. The second part of the input provides the data for vehicles, orbits, modules, satellites, satellite systems, deterministic launch schedules, and scheduled mission equipment upgrades. The complete deck setup is shown in Figure 2.

Data formats can be summarized as alphabetic entries which are left justified and integer entries which are right justified. Numerical entries containing decimal points must be written with the decimal point in the correct column.

The two EQUIP cards (see page C-5) are required for the CDC systems but are not used on the UNIVAC 1108 systems.

The first physical card of the deck is the Run Parameter card. This card does not change unless a new variable is added to the program with an array number greater than 285. Then the value 285 must be increased to the new value.

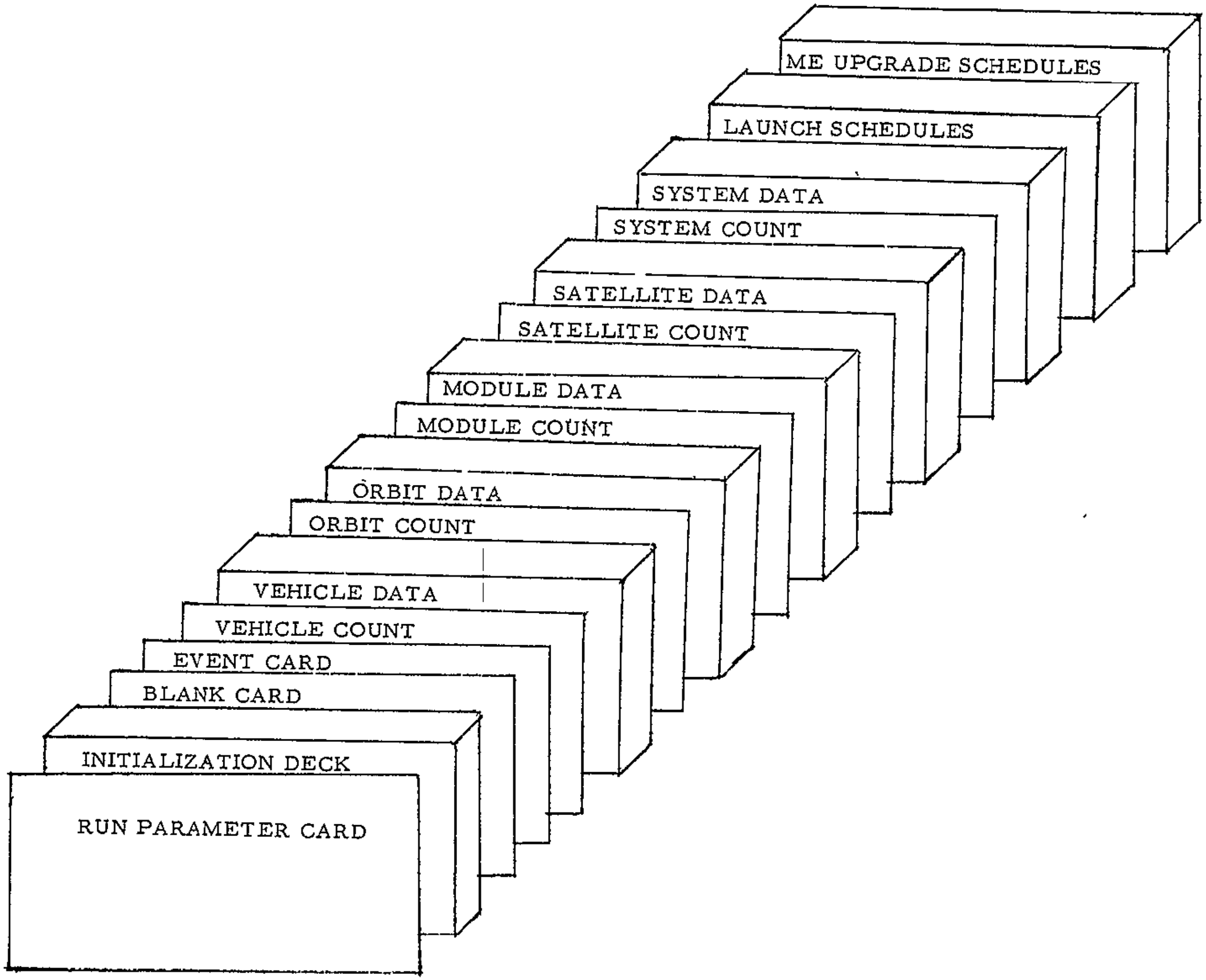


Figure 2. Configuration of Input Data Deck

2.3.1.1. Initialization Deck

The initialization deck is sometimes referred to as the "front end." It is a relatively static deck, changing only when the simple constants are changed in parameter sensitivity studies and when the model grows by the addition of new modules, satellites, etc., to the data deck such that the amount of memory must be increased.

The front end card is defined as follows:

Columns	Data Description
2 - 4	The array number for variables (as defined in Appendix B) which are input in strict numerical sequence.
6 - 8	A second array number whose presence implies a first array number through and including the second array number.
10	Dimensionality of array numbers. This entry is either 0 or 1 and must be as shown in Appendix C (C-5 and C-6). (It is checked against the definition portion of the program code.)
12	Read column is either blank or R. An R implies data in columns 50 - 66.
13	Zero entry column is either blank or Z. These two columns 12 and 13 must have one entry between them to stipulate a read or 0 operation. A Z will preset memory to 0.
16 - 18	An array number previously defined by a read-in card. This states which variable has the dimensionality constant for these arrays. These columns are used for dimensionality of 1 (column 10).
19 - 22	A four-digit vector length constant which must agree with the value input into the array number referenced in columns 16 - 18.
50 - 66	Numerical inputs except the format 7 (A6) which should be left alone. The numerical values can be anywhere in these columns and must have or have not the decimal point as shown. If the decimal point is left out, the value is an integer which is quite different from a value with a decimal point. The value would be very near 0.

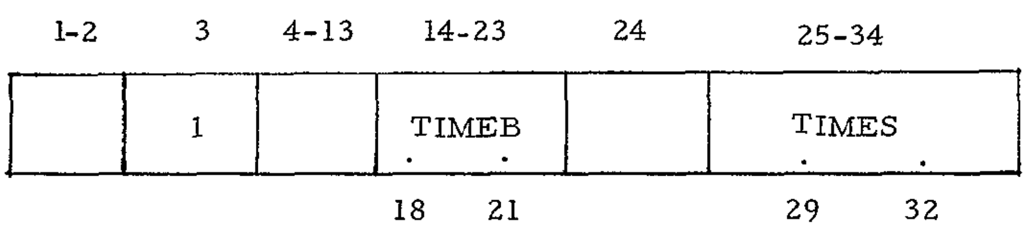
67 - 80	Commentary area used to associate names with array numbers.

The front end deck is terminated by a blank card.

2.3.1.2 Event Card

The event card is used to trigger the simulation and contains two data entries, the start and stop times of the run. The structure of an event card is:

Columns	Name	Data Description
1 - 2		Blank
3		1
4 - 13		Blank
14 - 23	TIMEB	Year, month, and day with the decimal point in columns 18 and 21. Month 0, day 0 = January 1.
24		Blank
25 - 34	TIMES	Year, month, and day with the decimal point in columns 29 and 32.



The remainder of the data deck consists of groups of data in the form of tables describing vehicles, orbits, modules, satellites, satellite systems, satellite launch schedules, and mission equipment upgrade schedules.

2.3.1.3 Vehicle Data

The first group of data cards is for vehicle data. The structure of the first card is:

Columns	Name	Data Description
1 - 3	COUNT	The right justified count of the number of different vehicles available to the program. This entry will be referred to as an I3 format for later groups of data.

1 - 3	4 - 80
#	Not Used - Available for Comments

The data description of each vehicle will occupy one vehicle card. The format of the card is fixed and applies equally to Shuttles, upper stages, and SEPS vehicles. The actual type of each vehicle is determined by its usage in the orbit table that follows the vehicle data. The structure of a vehicle data card is:

Columns	Name	Data Description
1 - 6	NAMEV	The alphabetic name of the vehicle. For the SEPS, the only name of the vehicle is SEPS.
7 - 13	DAYSV	The maximum total flight time of the vehicle in days including phasing maneuvers with the decimal point in column 12. For the SEPS, this entry is the maximum thrust time in days.
14 - 20	ISP	The effective specific impulse of the vehicle in seconds with the decimal point in column 19.

Column	Name	Data Description
21 - 27	WDV	The burnout weight of the vehicle in pounds with the decimal point in column 26.
28 - 34	WPNUV	The non-useable propellant of the vehicle in pounds with the decimal point in column 33. For the SEPS, this entry is the reserve propellant in pounds.
35 - 41	WCONV	The gross weight in the Shuttle in pounds with the decimal point in column 40. For an upper stage, this is the maximum propellant available in pounds. For the SEPS, this entry is the power level in watts.
42 - 48	REFTV	The refurbishment cycle time of the vehicle in days with the decimal point in column 47. For the SEPS, this entry is the weight of the usable propellant.
49 - 55	EXPV	If it is not 0, the vehicle is to be expended. The decimal point is in column 54.
56 - 62	PAYLV	The maximum length of stacked payloads that can be put on top of the vehicle. For the Shuttle, it is the available length of the payload bay. The entry is in feet with the decimal point in column 61. For the SEPS, this entry is the percentage propulsion efficiency.
63 - 64	NSTAG	The number of stages of a multistage vehicle. The data cards are ordered Stage 1, followed by Stage 2, followed by Stage 3 (if any).
65 - 66	SOLID	If it is not 0, the stage is a solid.

1 - 6	7 - 13	14 - 20	21 - 27	28 - 34	35 - 41	42 - 48	49 - 55	56 - 62	63 - 64	65 - 66
NAMEV	DAYSV	ISP	WDV	WPNUV	WCONV	REFTV	EXPV	PAYLV	NSTAG	SOLID
	. 12	. 19	. 26	. 33	. 40	. 47	. 54	. 61		

The number of vehicle cards must agree with the vehicle count on the first card. A stage is considered a vehicle and can be used as the second stage of a two-stage vehicle and as a separate vehicle in its own right.

2.3.1.4 Orbit Data

The next grouping is for orbit data. The structure of the first card is the same as that of the first card of the vehicle grouping; i.e., the count of orbits is in the I3 format.

The data for each orbit will occupy one orbit card. The structure of an orbit data card is:

Columns	Name	Data Description
1 - 6	ORBID	The alphabetic name of the orbit.
7 - 13	ORBDV	The ΔV required to go from the parking orbit into the final orbit (two-burn minimum) with the decimal point in column 12.
14 - 20	ORBPD	The period of the orbit in hours with the decimal point in column 19.
21 - 27	ORBRA	The apogee radius of the orbit in nautical miles with the decimal point in column 26.
28 - 34	ORBVC	The equivalent circular velocity of the orbit in feet per second with the decimal point in column 33.
35 - 40	RQUP	The alphabetic name of the required upper stage, if any.
41 - 46	RQSEP	The alphabetic entry SEPS if the SEPS vehicle is to be used.

Column	Name	Data Description
47 - 52	RQSUT	The alphabetic name of the Shuttle assigned to deliver to parking orbit.
53 - 59	DVI	The ΔV for the first burn of the tran-stage with kick solids with the decimal point in column 58.

1-6	7-13	14-20	21-27	28-34	35-40	41-46	47-52	53-59
NAME	DV .	Period .	R_a .	V_c .	RQUP	RQSEP	ROSUT	DV_1 .
	12	19	26	33				58

The number of orbit cards must agree with the orbit count on the first card.

2.3.1.5 Module Data

The next grouping is for the data pertaining to modules. The first card of the group is the count of modules in I3 format, and the warning factor FACT is in columns 4 - 8 with the decimal point in column 5.

1	2	3	4 - 8	9 - 80
			FACT .	Not Used
			5	

Then the individual module cards are entered, one card per module. The number of cards must agree with the input count of modules. The structure of a module card is:

Columns	Name	Data Description
1 - 6	MNAME	The alphabetic name of the module.
11 - 15	ALPF	The Weibul failure parameter with the decimal point in column 13.
16 - 20	BETAF	The Weibul failure parameter with the decimal point in column 18.
21 - 24	TTFMD	The module truncation time with the decimal point in column 24.
25 - 30	MODWT	The module weight in pounds with the decimal point in column 30.
31 - 35	MDVOL	The module volume in cubic feet with the decimal point in column 34.
36 - 41	MCLAS	The alphabetic module classification.
45 - 49	ALPW	The Weibul warning parameter with the decimal point in column 47. If this entry is 0, it is replaced by ALPF*FACT.
50 - 54	BETAW	The Weibul warning parameter β with the decimal point in column 52.
55 - 57	R	The reliability (90.) for the module with the decimal point in column 57.
58 - 62	TAU	The period of time at which R is true with the decimal point in column 60.

1 - 10	11 - 15	16 - 20	21 - 25	26 - 30	31 - 35	36 - 41	45-49	50-54	55-57	58-62
Name	α_f .	β_f .	T_T .	WT	Vol .	Class	α_ω .	β_ω .	R .	τ .
	13	18	24	30	34		47	52	57	60

The number of module cards must agree with the module count on the first card.

2.3.1.6 Satellite Data

The next grouping is for satellite data. The structure of the first card is the same as that of the first card for the vehicle grouping; i.e., the count of satellites in the I3 format.

The data for each satellite will occupy three or more cards per satellite. The first card for the data for a satellite is:

Columns	Name	Data Description
1 - 10	SNAME	The alphabetic name of the satellite.
11 - 15	SWT	The unused satellite weight in pounds with the decimal point in column 15.
16 - 20	SVOL	The satellite volume in cubic feet with the decimal point in column 18.
21 - 25	PRIOR	The satellite priority with the decimal point in column 25.
26 - 30	INCL	The orbit inclination with the decimal point in column 30.
31 - 36	ORBIT	The alphabetic name of the orbit.
37 - 70	Unused.	
71 - 75	NMOD	The integer count of the number of modules comprising this satellite.
16 - 80	TTSAT	The termination time of the satellite with the decimal point in column 80.

1 — 10	11 — 15	16 — 20	21 — 25	26 — 30	31 — 36	37 — 70	71 — 75	76 — 80
Name	wt .	Vol .	Pr .	i .	Orbit	SPARE	NMOD	T_s
	15	18	25	30				

The second card of a satellite group is:

Columns	Name	Data Description
1	POLDN	The policy (1, 2, 3, 4) for replacing the satellite on an NRU failure.
2 - 6	SORTE	If not 0, the payload is a Sortie to be aloft this many days.

The remaining card or cards of the satellite group list the modules by name which the satellite comprises. The format of the module list card is:

Columns 1 - 10 are blank.

The remainder of the card is divided into 7 fields of 10 columns each. Each 10-column field contains a 6-column name of a module in the module table and a 4-column blank or nonblank entry. If the entry is 2 - 15, the module is the first member of a redundant group. The entry on the following module tells how many operational modules are required to call the group operational. If the four-column entry is otherwise not blank, the module cannot be replaced. Seven modules may be entered per module card to provide the number of modules given in column 71 - 75 of the first card of the satellite group by entering the necessary number of cards.

1 — 10	11 — 19	20	21 — 29	30	31 — 39	40	41 ....
Blank	Name		Name		Name		

The number of groups of cards (each group is a satellite) must agree with the satellite count on the first card.

2.3.1.7 Satellite Systems Data

The next group is for the description of the satellite systems. The first card in the group is the count of the satellite systems, again in I3 format. Each system is described by one or more data cards. The structure of the first card for each system is:

Columns	Name	Data Description
1 - 6	SYNAM	The alphabetic name of the system.
7 - 11	NFUP	The integer number of active satellites required to have an active system (nine or less).
12 - 16	NS	The integer number of satellites in the system (nine or less).
17 - 20	TTSYS	The termination time of the system with the decimal point in column 19.
21 - 30	STNAM	The alphabetic name of a satellite in the satellite group.
31 - 40	PHASE	The longitude of the satellite with the decimal point in column 35.
41 - 50	STNAM	The alphabetic name of the second satellite in the system.
51 - 60	PHASE	The longitude of the second satellite with the decimal point in column 55.
61 - 70	STNAM	The alphabetic name of the third satellite in the system.
71 - 80	PHASE	The longitude of the third satellite with the decimal point in column 75.

1 — 10	11 — 15	16 — 20	21 — 30	31 — 40	41 — 50	51 — 60	61 — 70	71 — 80
Name	#	#.	Name	Long.	Name	Long.	Name	Long.
		19		35		55		75

Should more than three satellites comprise a system, other satellites are entered on subsequent cards with the same format except that columns 1 - 20 are blank. The number of satellite systems input with one or more cards per system must agree with the count entered on the first card. The program has a limit of a maximum of nine satellites in a system.

2.3.1.8 Satellite Launching Data

The schedule of new satellite launchings is the next group of cards. Each card can designate up to four satellite launchings. The card is, therefore, divided into 4 groups of 20 columns each. The 20-column group is:

Columns	Name	Data Description
1	N	The number of satellites in a satellite system.
2 - 10	SYSNAM	The name of the satellite system for which the satellite is intended.
11 - 20	DATE	The date of availability with the decimal point in column 15 and including the fractional part of the year.

1	2 — 10	11 — 20
N	SYSNAM	DATE (. at 15)

The schedule input is terminated whenever column 1 of a card contains satellite number 0 of a system. For user convenience, the program requires data in the first 20 columns; the others are optional.

2.3.1.9 Mission Equipment Upgrade Data

The schedule of launching of mission equipment upgrades is the next group of cards. Each card can designate only one upgrade. The structure of a card is:

Columns	Name	Data Description
1 - 6	SYNAM	The name of the system containing the mission equipment to be upgraded.
7 - 10	NS	The number of the satellite in the system.
11 - 16	MEOLD	The name of the old module.
17 - 20	NM	The number of duplicates to be replaced.
21 - 30	Date	The date at which the new modules become available. The entry is of the form, 1984.01.01 for year, month, and day. The decimal points are in columns 25 and 28.
31 - 36	MENEW	The name of the new module intended to replace the old one.

1 - 6	7 - 10	11 - 16	17 - 20	21 - 30	31 - 36	37 - 80
SYNAM	NS	MEOLD	NM	Date (. . at 25 28)	MENEW	Not Used

The upgrade input is terminated whenever column 1 - 6 of a card contains a blank system name.

2.3.2 Naming Conventions

All names input into the program are restricted to six characters in length. Some naming conventions used in other applications are cited as examples for the user:

VEHICLES	TUG and XTUG for the Tug to be used in the reusable mode versus the expended mode. SHUTL1, SHUTL2, etc. to define the Shuttle performance to different altitudes. SEPS is reserved for the said vehicle.
ORBITS	SYNC/0 - Synchronous, equatorial. 100/35 - 100 nmi circular, 35-degree inclination. 1-2/30 - 100 x 200 nmi, 30-degree inclination. ESC1 - Interplanetary injection orbit No. 1.
MODULES	AVCS1, AVCS2, etc. for altitude control modules. EPS1, EPS2, etc. are electrical power modules.
SATELLITES	ASTIC, EOP4, NNDIA, NNDIB are satellite designations found in the current mission model shortened to six characters and using the letters A and B for different versions.
SYSTEMS	Systems names usually match the name of the satellite members or the name of the program, agency, or whatever - NNDIAB, COMSAT, EOL, SURV, EARTH, etc.

The name SEPS is the only reserved name in the program; all others are at the discretion of the user. The user is reminded to avoid duplicate names in the input tables, as the program will tend to ignore the duplicate name occurring after the first definition.

2.3.3 Input Data Errors

The computer code will detect three classes of errors: mispunched data cards, invalid dimensioning, and an invalid model.

2.3.3.1 Mispunched Data Cards

The SIMSCRIPT system requires that the decimal point be in the column stipulated by the code. This is the most frequent error made by a user, misalignment of data entries. In such cases, the message generated is unfortunately insufficient as it identifies the routine detecting the error and little more. The user must then look at a listing of the data cards and visually locate the misalignment. An impatient user can look at the output of the program (which also shows the input data) and, by a careful perusal, with intelligence, deduce the errors. The following list of routines with the name of the data table they process may be an aid to error detection.

LDVEH	-	Vehicle Data
LDORB	-	Orbit Data
LDMOD	-	Module Data
LDSAT	-	Satellite Data
LDSYS	-	Systems Data
LDSCH	-	Satellite Launch Schedule
LDME	-	Mission Equipment Upgrade Schedule

Errors of this type can cause other errors later. Therefore care is required in diagnosing subsequent errors (especially nearby).

2.3.3.2 Invalid Dimensioning:

For example, the vehicle data may be preceded by a number that states the number of vehicle cards being input. The program checks the number against the value intended into array number 85, and, if it is greater, an error message will be printed. The solution is to increase the value in array number 85.

2.3.3.3. Invalid Model

The user can make errors in his model by referencing, for example, a module by a name that is misspelled or doesn't exist in the module data. The program will identify the erroneous entry, and the user will have to figure out what he intended to do. This is the only checking done; the consistency and content of the model are the responsibility of the user. Three clues for inconsistency of the model are: no satellites ever initially launched to a particular orbit position are shown as a 0 in the total launched for that position, bad satellite launch schedules are shown by satellites becoming available after the system goes permanently disabled (NG) or by high maximum delays and low minimum availabilities.

2.3.4 Program Output

This section describes the computer output in Appendix C beginning on page C-5. The printout logically separates into the following groups of pages listed in order of occurrence.

2.3.4.1. Initialization Cards (Pages C-5, C-6)

This is an exact line for line copy of the front end portion of the input deck with header lines of card column numbers. The first card with the series of numbers (285, 30, 12, etc.) is the specification card and should be left as is. The remaining cards comprise the front end including the blank terminating card.

2.3.4.2 Input Data (Pages C-7, C-8)

This is a print back of the input data deck with reformatting of the data. In producing this printout, a modification was made to the routine LDMOD to force the module termination time to be 20 years (which reduced the output because no modules were terminated).

2.3.4.3 Synopsis of Input (Page C-9)

As the launch schedules were read in, the systems were tagged as being in use. Therefore, the program detected an unused system in the input. The user must decide on the criticality of the message, since an unused item consumes memory. A check was made by computing the necessary satellite/system position needed by the launch schedules, and that value is printed with the front end entry in array number 230 (which offers a potential savings in memory). The number of required systems is printed with the entry in array number

200. The unused satellites are listed, and the number of required satellites is printed with the entry in array number 180. The unused modules are listed, and the required number is printed with the entry in array number 150.

2.3.4.4 Chronological Time History of Base Cycle (Pages C-10 through C-12)

As the program executes the first cycle from year 1 to year 3000, this printout is produced showing all events as they occurred. The column titles are: Time in years, months, and days with clock counting from zero (January 1, 1980=1980.0.0); system name with operational status; module name with operational status; satellite name with operational status; and vehicle name and number with availability status. The small number after the system identifies the member of the system, which is necessary if the same satellite is used several times in the system. For instance, the user might read, "on 1980.3.28, the satellite NNDIA became available for launch." It is the third member of the inoperative system NNDIAB. On 1980.3.28, a launch of Shuttle 20 and Tug 10 occurred, because the Tug could not carry both payloads. All payloads listed between "launch now" and the dashed line were launched together. On 1980.3.29, the satellite NNDIA is placed in operational mode at its orbital position. On 1980.4.14, Shuttle 20 and Tug 10 completed refurbishment and became available for use again. On 1980.7.22, the module TTC1 failed, causing the satellite NNDIA and the system NNDIAB to be inoperative. The module was replaced on 1980.9.2. In this sample, there are launches occurring because of exceeding vehicle performance (1980.3.28 twice) and mandatory after 90 days (1980.8.26-NNDIB). These are launches of single- and multiple-satellite deployments (1983.2.27) and multiple-module replacements with a definite delivery sequence (1982.6.18). There is an instant launch of a Sortie (1982.1.2) with its subsequent return after seven days in orbit. For each launch, the total payload weight and length including service units are shown. At the end of the cycle, all satellites are operationally terminated and left in orbit. Five days later, the last Shuttle and Tug complete refurbishment. This printout is used to verify the interactions between the events and to show difficulties in the mission model of too small a fleet, forced expenditure of a vehicle, or excessively overweight payload.

2.3.4.5 Chronological Time History of Satellite Position in Orbit (Page C-13, C-14)

This is a rearrangement of the preceding report with the vehicle and launch information deleted. The information is arranged so as to provide a history of all activity pertaining to a specific satellite at a specific orbital position. Thus, the user is informed about the birth, life, and death of each satellite. The user can trace the system activity as all members of the system are grouped together, yet separately.

2.3.4.6 Statistical Summary for 25 Monte Carlo Cycles (Page C-15)

Between this page and the preceding page, the program has executed but not printed 24 additional cycles, none of which were identical to any of the others. This is effected by the use of a random number generation which can be negated by the variable TIMEC. This page is the flight summary for the three classes of vehicles: Shuttles, upper stages (called Tugs), and the SEPS. For example, in the year 1982, the Shuttle flew a minimum of 2 flights on one of the Monte Carlo cycles, averaged 3.2 flights over all cycles, and flew a maximum of 5 flights on another Monte Carlo cycle. The Tug flew one less flight in that year because the Sortie operates on the Shuttle only. The totals of all years are computed for each Monte Carlo cycle, and the values obtained are used to determine the minimum, average, and maximum total flights. The user is cautioned against attempting to add the minimum or maximum columns for a check; i.e., the sum of the minimums is not the minimum of the sums. The other two lines are connected with vehicle availability: percentage of time the vehicle was unavailable and the number of delays in days resulting from vehicles being unavailable. This sample had a very large fleet and therefore did not suffer any shortages of vehicles. There is one more line showing the average number of expended upper stages which is 0 in the sample (suppresses printing of line). Note that the SEPS was not used in the run.

2.3.4.7 Orbit Traffic Summary (Page C-16)

This report quotes the average number of flights and average delivered payload weights for each orbit separately. For example, the Tug made the final delivery to the orbit SNC/O with 11.7 flights and 3179 pounds per flight. The 11.7 agrees with the previous page. There is no traffic to the orbit 1-1/28. The Shuttle delivered the 10,000-pound Sortie once each cycle to the orbit SORT for an average Shuttle load factor of 16 percent.

2.3.4.8 Statistics for System - ASTID (Pages C-17 through C-20)

This report summarizes the statistics for each module on each satellite, each satellite in each system, and each system. The system name appears as above. The modules for the satellites in the system are summarized individually. For example, the first AVCS7 module was replaced a maximum of once in the 25 cycles. The average count is too small to show, and the maximum of 1 required a maximum of 0.25 equivalent vehicle flights (the maximum of 1 could have occurred 5 or 6 times). The first Astronomy 1D Satellite was deployed exactly once in each Monte Carlo cycle, yielding an average of 0.49 equivalent flights in a band from 0.40 to 0.76. Including the module flights, this yields an average of 0.77 flights flown in support of this satellite. The satellite was operational or available 85.05 percent of the time and averaged 61.93 days of outage during the 25 cycles. The second Astronomy 1D Satellite of the system is shown, and, in this case, the system statistics are also shown. The system required a minimum of 0.88 Tug flights but suffered operationally as badly as approximately 30 to

75 percent, while it took a minimum of 2.99 days to restore the system to service. It is very likely that the three values did not occur in the same cycle, particularly since there was one cycle in which the system was fully operational. A dash line indicates the end of the statistics for a system.

2.3.4.9 Module Summary (Page C-21)

This report summarizes the warning, failure, and replacement activity of each module in the original module table. For example, the module AVCS7 is a frequently used module, appearing 24 times in the previous report. The module had no warning mechanism but failed an average of 39 times per Monte Carlo cycle in a band of 2 to 8. But an average of 3.5 out of 4 were replaced as modules failing too close to satellite, system, or run termination. The module MODULE has no failure mechanism, and NASTIC is too reliable (long lived) to have failed.

3. PROGRAMMERS MANUAL

This section is provided as documentation for the LOVES Computer Program Code and as a convenient reference should the need arise to modify the code or the basic procedures in the program. It will also provide further understanding for the user who takes the time to peruse this part of the document.

The section is divided into three parts. First is a description of the SIMSCRIPT 1.5 part of the program with detailed definitions for each event and subroutine. The second part provides detailed definitions of a group of subroutines coded in FORTRAN and used to pass data along into labeled common areas for the FORTRAN coded performance subroutines. The third part of the section includes a general description of the performance computation subroutines, also in FORTRAN. Following this section are supporting appendices which describe and illustrate each set and queue (Appendix A) and each of the variables defined globally in the program (Appendix B).

A series of flow charts are provided in Figures 3 through 5 which illustrate how the LOVES Computer Program is linked together and the interconnections between the various events and subroutines. The connections (or calls) to subroutines are shown by the solid lines, and the connections marked by the dashed lines indicate which events and subroutines initiate other events. The events are differentiated from the subroutines by asterisks following each event name.

Figure 3 is divided into an input section and a Monte Carlo section. The input section is executed only once at the beginning of the run. The Monte Carlo section contains the event START to initiate the Monte Carlo cycle and the event TERM to terminate the cycle. The other subroutines have to do with the chronological printout of all activity for the first cycle only for each individual satellite, the gathering of statistics at the end of each cycle, and the final statistical output (TERMO).

The heart of the simulation is illustrated in Figure 4, beginning with the events START and NWSAT. This section of the simulation is concluded with the subroutine PROP which is the transition to the next illustration (Figure 5). Figure 4 is extremely complex, and the most important point in this phase of the simulation is not readily apparent from the diagram--most of the activity flows through or around the two subroutines SHIP and STATUS.

The subroutines shown in Figure 5 are all coded in FORTRAN except for the transition subroutine PROP. All of these subroutines have to do with vehicle performance and phasing considerations.

3.1 ELEMENTS OF SIMSCRIPT 1.5

3.1.1 Overall Logic Construction

A simulation program written in SIMSCRIPT has no main program per se. The SIMSCRIPT simulation system provides the main program for the user and consists of subroutines and events. SIMSCRIPT subroutines are programmatically identical to FORTRAN subroutines with parameter lists and return statements, and FORTRAN subroutines can be called from SIMSCRIPT routines. Events are routines called by the SIMSCRIPT system, and they are divided into two classes: external events (exogenous) and internal events (endogenous). External events are triggered by a card in the input data deck, and internal events are set up and triggered by the code of the various events and subroutines.

An event differs from a subroutine in that an event occurs or is called at a specific time on the simulation clock. At the completion of an event, control is passed back to the main SIMSCRIPT subroutine which finds the next closest event in the immediate future. That event then occurs or is called.

This program consists of one external event (used to start the simulation) and 15 internal events for the simulation, including start and stop events. The program does a simulation between the bounds of the start and stop events which is repeated n times in a Monte Carlo

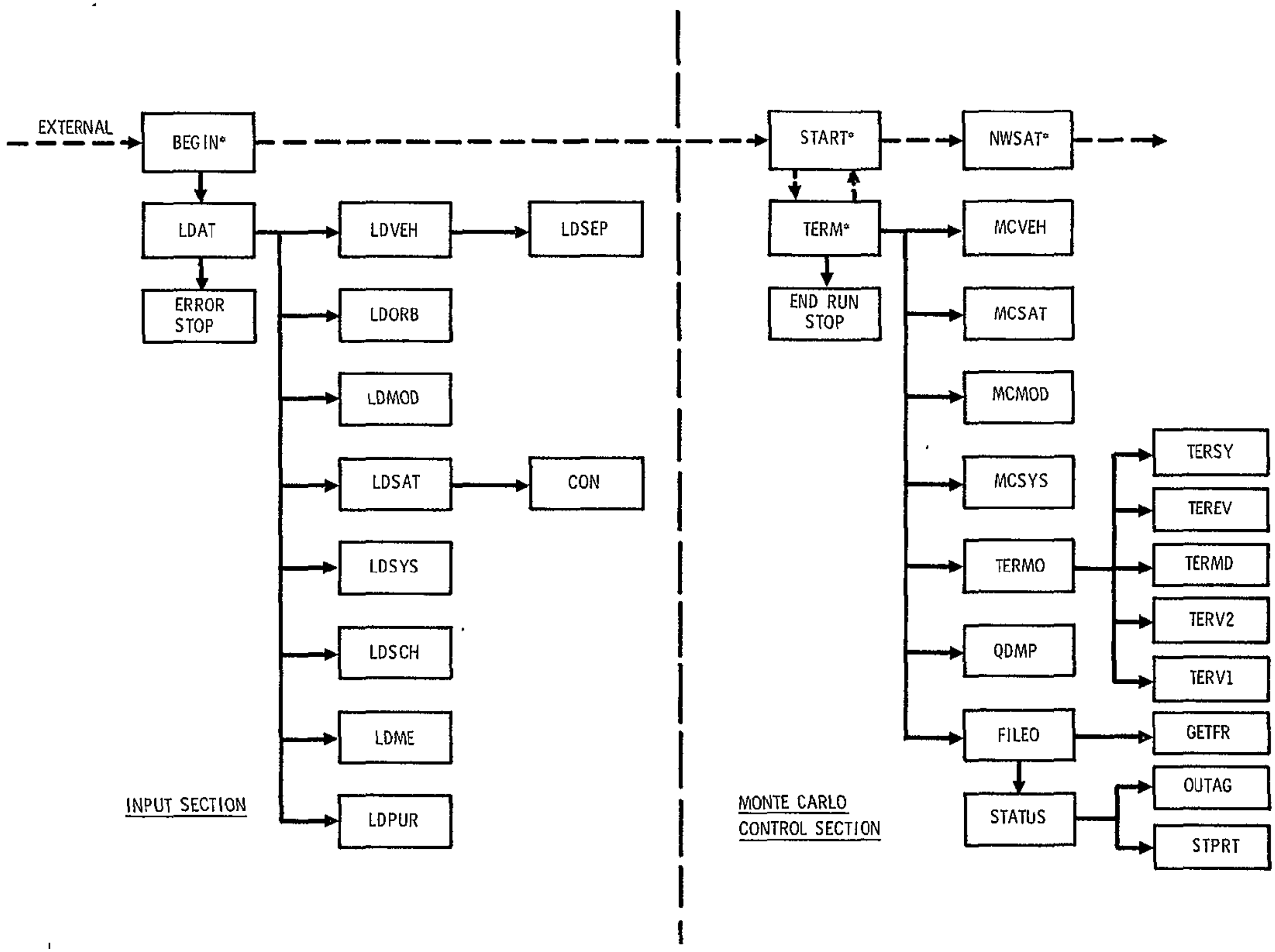


Figure 3. The Input Section and Monte Carlo Control Section

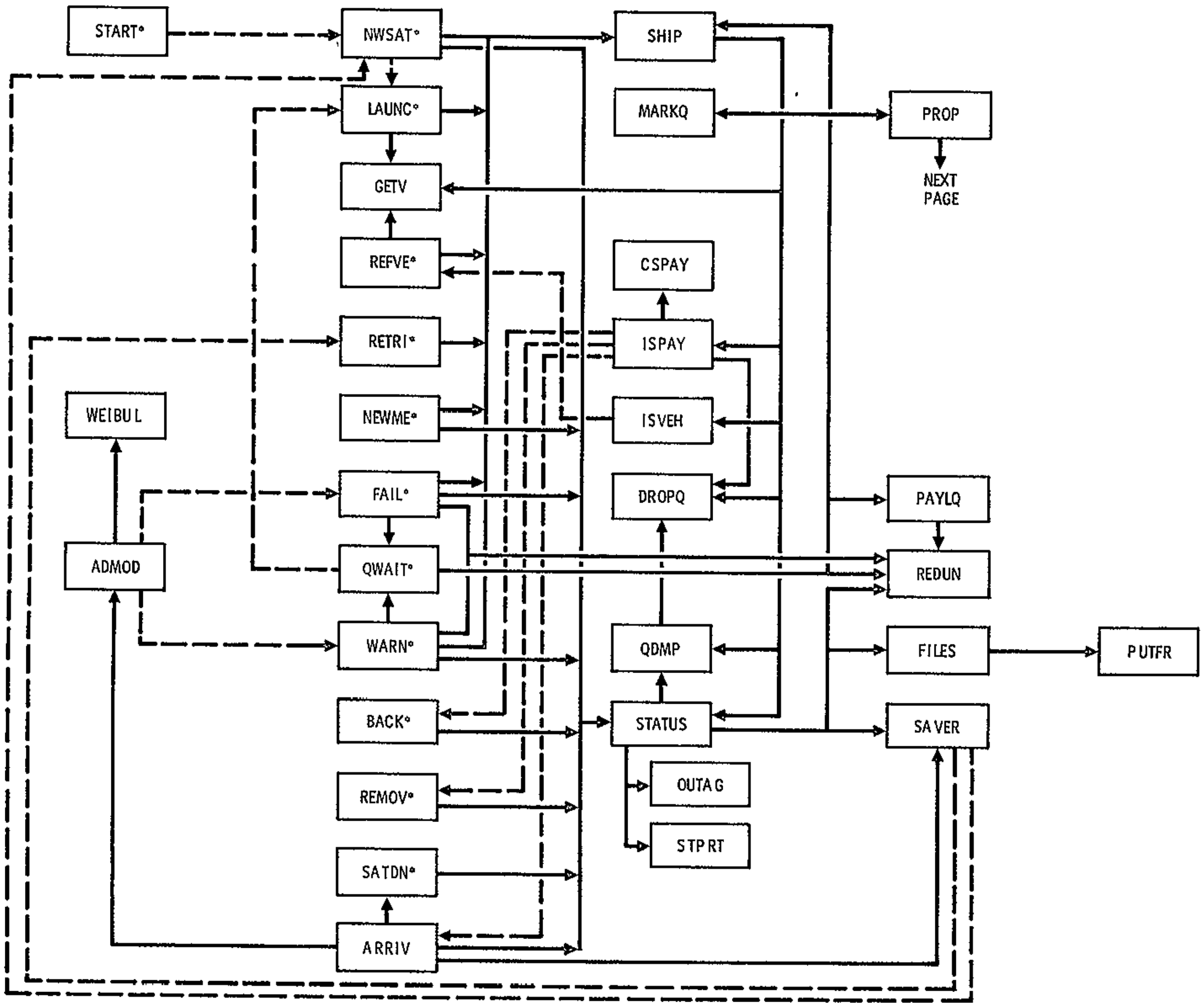


Figure 4. Monte Carlo Cycle Area, Heart of the Simulation

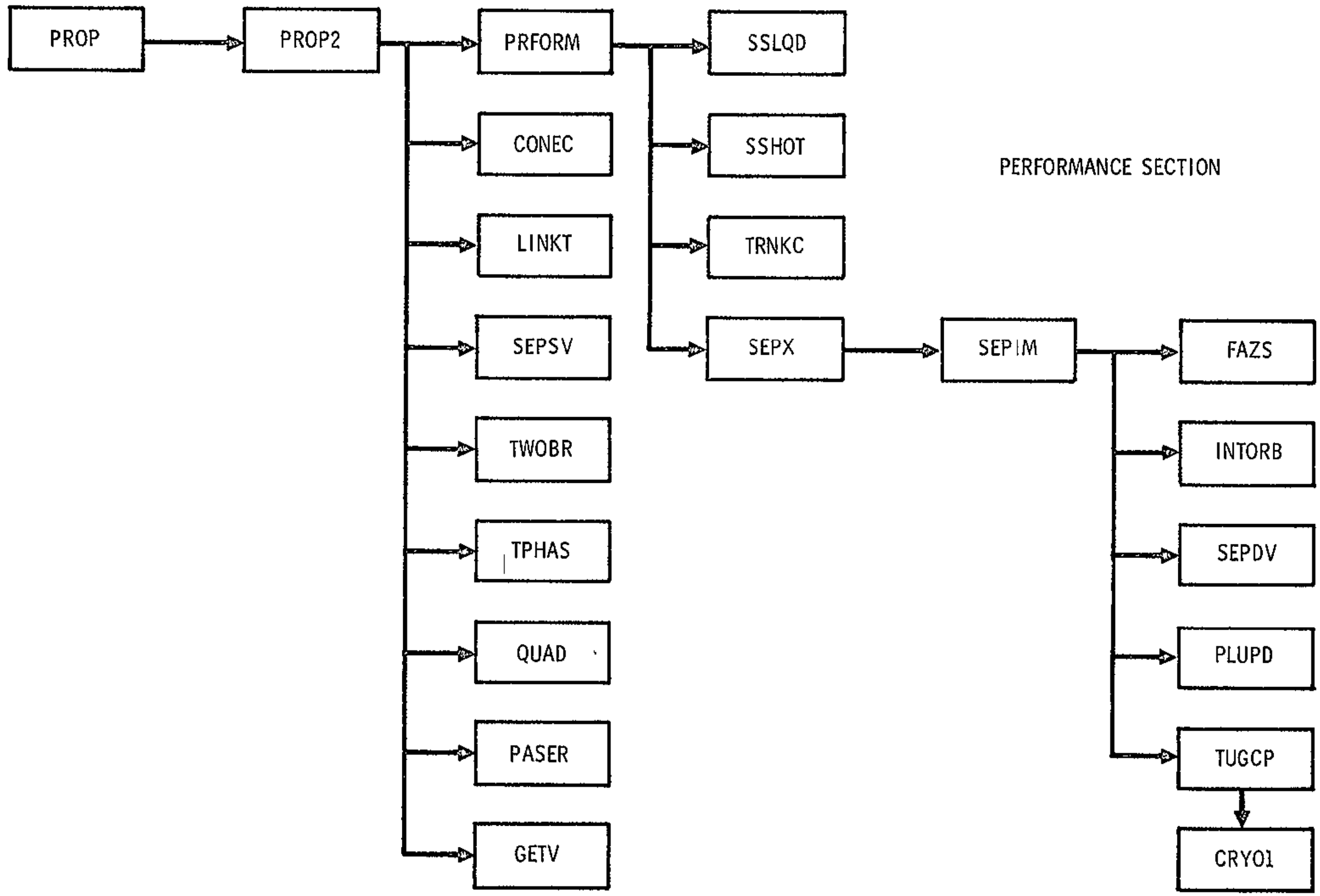


Figure 5. FORTRAN Linkage and Performance Subroutines

fashion. Each cycle is different according to the random sequence output by a random number generator.

SIMSCRIPT also provides the capability of processing waiting lines (referred to as queues, holding queues, or sets). Queues can be first in-first out (supermarket checkout lines), last in-first out (stack of plates), or ranked in ascending or descending order. Queues are more fully discussed in Appendix A.

3.1.2 Definition Table

The first part of a SIMSCRIPT program is the Definition Table in which variables, events, entities, and attributes are defined, typed, and dimensioned. The format of the cards in the definition table can be found in the SIMSCRIPT manual. The variables, etc. listed on these cards are retained by the compiler as it processes the remainder of the program. Thus, these variables, etc. are accessible by any SIMSCRIPT routine. A more exact definition of these items can also be found in the appendices..

3.1.3 Events List

The events list begins with the word "EVENTS," and then informs SIMSCRIPT which events are external (exogenous), which are internal (endogenous), and the number of each. In this program, there is one external event BEGIN which is keyed as number 1. The digit 1 appears on the event card in the data deck requesting the SIMSCRIPT event-sequencing routine to execute event 1 (BEGIN) at the time entered on the card.

There is a list of 15 internal events. These two lists (see Table 1) are used by the compiler to define the code for the event-sequencing routine. The events list is terminated with an "END" card.

Table 1. Internal - External Events List

INTERNAL		EXTERNAL	
1.	ARRIV	1.	BEGIN
2.	BACK		
3.	FAIL		
4.	LAUNC		
5.	NEWME		
6.	NWSAT		
7.	QWAIT		
8.	REFMO		
9.	REFSA		
10.	REFVE		
11.	REMOV		
12.	RETRI		
13.	SATDN		
14.	START		
15.	TERM		
16.	WARN		

3.1.3.1 Event ARRIV

This event occurs whenever a satellite or a module arrives at its destination in orbit. If the payload is a satellite, the number deployed and the number at the position are increased by 1. The satellite system and satellite clocks are started if necessary. All the modules are activated, and the failure and warnings are predicted via the subroutine ADMOD. If the satellite termination policy is not 0, the subroutine SAVER is called for retrieval and new satellite delivery. If the satellite is not a Sortie, the satellite termination event SATDN is set up.

If the payload is a module, the satellite status is checked. If it is out, the subroutine returns. The single module is activated by the subroutine ADMOD, and the module status line is printed by the subroutine STATUS. The number of times the module has been replaced is increased by 1.

3.1.3.2 Event BACK

This event occurs when a satellite is retrieved from orbit and actually reaches the ground. The routine STATUS is used to print the line signifying the end of the retrieval operation in the chronological summary printout.

3.1.3.3 Event BEGIN

This event is triggered by the event card in the input data deck, and the routine obtains the times TIMEB and TIMES from the event card. Since the primary function of BEGIN is to initiate the simulation, it therefore sets up the first occurrence of the event START and calls the subroutine LDAT to load the remainder of the input data. Time intervals input in days are rescaled to years, and the minimum variables of statistical parameters collected during the Monte Carlo cycles are set to arbitrary high values. Control is then returned to the SIMSCRIPT system which will then cause the event START to occur.

3.1.3.4 Event FAIL

This event occurs whenever a module fails as determined by the random number process used by the subroutine WEIBUL when the module was previously activated in orbit. If the satellite is out of service and must be replaced, the failure is ignored; otherwise the subroutine STATUS is used to print the status line. The number of times this module has failed is then increased by 1. If the failure is too close to the end of life for the satellite, the failed module is not replaced; otherwise, a check is made to see if a warning has occurred. If so, the module is already scheduled to be replaced; otherwise, the module is entered into the loading queue and the mandatory launch event LAUNC is scheduled to force the payload to fly by a specified date.

3.1.3.5 Event LAUNC

This event occurs at that time when a payload must be launched (mandatory launch event). The orbit for the payload is obtained, and these data are used to scan the orbit queue to make sure the payload is still in the queue. If the payload is not in the queue, the event takes no action. If the payload is still in the queue but there is no vehicle available, a message is printed, and statistics are collected on vehicle unavailability. Otherwise, if the payload is in the queue and a vehicle is available, the subroutine SHIP is called to force the flight to go.

3.1.3.6 Event NEWME

This event occurs when a mission equipment upgrade is scheduled by input data. The new module is put into the loading queue, and the subroutine STATUS is used to print the status line for "ME UPGRADE."

3.1.3.7 Event NWSAT

This event occurs whenever a complete satellite becomes available for launch. The routine STATUS is used to print the status line for "AVAILABLE." The routine will return if the arrival occurs after the system (to which this satellite belongs) has terminated its activity. The satellite is scheduled for launch, and the mandatory launch event is set up under the conditions that the payload is not a Sortie, and the event does not occur after system termination.

3.1.3.8 Event QWAIT

This event occurs after a suitable delay(WAIT4) after the failure or warning of a module. If the module is critical (satellite has failed), the module is placed directly in the loading queue and the routine exits. Otherwise the module is passed to the routine SHIP and if the module was put into the loading queue, the corresponding mandatory launch event LAUNC is set up for the module.

3.1.3.9 Event REFMO

This event is intended to occur when a module completes refurbishment. The creation of the event elsewhere in the code has not been implemented, and the operations to be performed by this event have not been defined.

3.1.3.10 Event REFSA

This event is intended to occur when a satellite completes refurbishment. The creation of the event elsewhere in the code has not been implemented, and the operations to be performed by this event have not been defined.

3.1.3.11 Event REFVE

This event occurs when a vehicle completes refurbishment and is thereby available again for a flight. On the first Monte Carlo cycle, the vehicle refurbishment event is reported as a part of the chronological history. The vehicle is identified as a Shuttle, Tug, or SEPS, and the appropriate fleet is checked for no vehicles available. The refurbished vehicle is made available for use; if there were vehicles previously available, the event returns. Otherwise the loading queues attached to each orbit are scanned to see if returning this vehicle to the fleet permits a pending flight to be launched. The flight is launched via a call to the subroutine SHIP.

3.1.3.12 Event REMOV

This event occurs when the vehicle docks with a satellite being retrieved. The satellite is thus removed from service and from its orbit position at this instant. The subroutine STATUS is used to print the line in the chronological time history. The subroutine QDMP is then used to remove any modules scheduled for replacement on this satellite from the loading queue if this is the only satellite at this position.

3.1.3.13 Event RETRI

This event occurs when it is time to schedule in the loading queue the retrieval of a satellite. The subroutine SHIP is used to put the retrieve request in the loading queue. There is no required time of completion, and therefore no mandatory launch event is set up.

3.1.3.14 Event SATDN

This event occurs when the lifetime of a satellite is reached. At this time, the satellite is retired from service and can only be replaced by another satellite. The subroutine STATUS is used to print the line in the chronological time history.

3.1.3.15 Event START

This event is first set up by the event BEGIN as the beginning of the first Monte Carlo cycle and thereafter set up by the event TERM for the beginning of a subsequent cycle. For each satellite position in orbit and for each module assigned to each position, the statistical parameters to be accumulated over a Monte Carlo cycle are set to 0. If this is the first cycle, the chronological time history titling is printed. The set NEWS (Appendix A) is processed to retrieve all the necessary NWSAT events as defined by the input data cards. On the first cycle, the start simulation cycle event time is printed out. The vehicle fleets are all placed in a state of availability. The system statistical parameters to be accumulated are zeroed out. The event TERM is set up for the arbitrarily chosen year 3000 (all events in the simulation will be done by then). The module statistical parameters are set to 0.

3.1.3.16 Event TERM

This event occurs at the end of each Monte Carlo cycle to provide a reference point for collection of statistics and a point for chosing whether or not to terminate or continue the simulation. At the end of the first cycle only, the terminate simulation cycle event time is printed out and the subroutine FILEO is called. The Monte Carlo cycle counter TRIG (Appendix B) is advanced, and, at the end of each cycle, statistics are gathered by calling the subroutines MCVEH, MCMOD, MCSAT, and MCSYS. If the end of the run has been reached (TRIG ≥ TRIGS), the event calls the subroutine TERMO and stops. Otherwise, time is set back to 0, and the event START is set up for the next cycle.

3.1.3.17 Event WARN

This event occurs whenever a module warns of an impending failure as determined by the random number process used in the subroutine WEIBUL when the module was previously activated in orbit. If the satellite is out of service and must be replaced, then the warning is ignored. If the satellite is in service, the subroutine STATUS is used to print the status line, and the number of warnings for this module is increased by 1. However, if the warning is too close to the end of life for the satellite, the replacement module is not shipped. Otherwise, the module is put into the loading queue for shipment, and the mandatory launch event LAUNC is set up to force the payload to fly by a specified date.

3.1.4 Subroutines List

The following paragraphs describe the various individual subroutines coded in SIMSCRIPT. They have been arranged in alphabetical order for the convenience of the user.

3.1.4.1 Subroutine ADMOD

This subroutine is called whenever an individual module is replaced on a satellite and for each module on a satellite when the satellite is presumed released from the delivery vehicle. The purpose of the subroutine is to predict failures and set up a sequence of events for module failures and warnings from the module based on a random number input to the Weibul function. The random number and the Weibul function are obtained from the subroutine WEIBUL.

In normal usage, the subroutine ADMOD is called with the parameters IS and IM where IS is the pointer to the specific satellite in an orbit position and IM is the pointer to the item MODSY in the queue Mod (Appendix A) identifying the specific module. The subroutine ADMOD is divided into three logical sections or blocks of code:

a. First, the module on the satellite is placed in an active state, the characteristics of the module are found, and the subroutine WEIBUL is used to determine the predicted module warning time (TW) and failure time (TF), each a random interval of time from now into the future.

b. The second section has to do with warnings. Any warning event associated with this module on this satellite, is removed from the timing queue. If the variable TW is 0, no warning events can be generated and the remainder of the section is bypassed. A warning can be detected either by telemetry data (random) or predicted from known failure characteristics of the module less a lead time for delivery. The minimum value of the two is taken to predict the event. The event cannot occur after the predicted termination of the satellite. The event is defined and entered into the timing queue, and its location in the timing queue is recorded in the module data of the satellite for later reference.

c. The third section deals with the pending failure of the module. The processing for a failure is identical to that of a warning except there can be no lead time on a failure that can occur any time up to termination of the module.

3.1.4.2 Subroutine CSPAY

This subroutine computes the statistics associated with the launching of a set of payloads, and it is called from the subroutine ISSUE. For each payload scheduled for the current flight, the following functions are performed:

a. The total payload weight of the flight is accumulated, and, if the payload item is a module, the number of times the module is flown is increased by 1.

b. The number of service units (each holding a number of modules) is determined, and the weight of the service units is added to the total weight and prorated equally among all the modules on the flight.

c. The weight factors (individual payload weight divided by the total weight on the vehicle) are accumulated for the total traffic to a satellite position, the total traffic of whole satellites only to a satellite position, and the total traffic for each module at each satellite position.

3.1.4.3 Subroutine DROPQ

The function of this subroutine is to remove a payload from the orbit queue and the loading queue. The subroutine DROPQ is called with the parameters J and IO where J is the pointer to the payload in the loading queue and IO is the pointer to the orbit being processed. The orbit queue is searched, and, when the payload is located, it is deleted from both the orbit queue and the loading queue.

3.1.4.4 Subroutine FILEO

This subroutine is called only once from the event TERM at the end of the first Monte Carlo cycle. The purpose of the subroutine is to print the status information summarized for each satellite position. This is done by reading back the data stored on tapes 11-20 one tape at a time. The data is then filed in the ranked set FRS (Appendix A). The set is ranked by satellite position and retains its first-in, first-out character with respect to time; therefore, the set is chronological for each satellite position. As each item is processed, the necessary program variables are reestablished to the instant the item was put into the set. The subroutine STATUS is used to reprint the status line that corresponds to the item in the set. The net result is a time history of activity relative to each satellite position. As the items in the set are processed, the memory storage space used by each item is returned to the SIMSCRIPT System.

3.1.4.5 Subroutine FILES

This subroutine is called from the subroutine STATUS after the current status of a particular module, satellite, and system has been determined. The purpose of the routine is to retain the status information for printout at a later time. The subroutine FILES is called with the parameters IS, IM, and IST where IS is the pointer to the satellite position arrays, IM is the pointer to the module on the satellite (if IM is not 0 indicating the item is a satellite), and IST is the status indicator 1 - 9. The member for the set FRS (Appendix A) is built, but instead of being filed in the set, it is written to a temporary disc file identified by tapes 10-19 according to satellite position. Sufficient information is carried in the data written to tape to be able to recreate the print line at a later time.

3.1.4.6 Subroutine GETV

This subroutine assigns the necessary vehicles to the next flight and a flag is set non-zero when vehicles are not available. The subroutine GETV is called with the parameter IGO where IGO is a flag with the following values:

0	All necessary resources are available.
1	Shuttle not available.
2	Tug not available.
3	SEPS not available.
4	Launch pad not available.

The Shuttle fleet availability array is then searched to find available vehicles, and those available are noted. However, if none are available, the flag IGO is set, and the subroutine returns. If a Tug is not required, the subroutine also returns. Otherwise, a Tug is located in the Tug fleet by the same process. In the same manner, if a SEPS is not required, the subroutine returns. Otherwise a SEPS is found. In a similar manner, an available launch pad will be located in the list of launch pads.

3.1.4.7 Subroutines ISPAY and ISVEH

This pair of subroutines are called from the subroutine SHIP when the decision has been made to launch a group of payloads. There are no parameters shown here as all necessary information is available in the definition table for the variables IORB, ILOAD, NQ, ISHUT, ITUG, and ISEPS (Appendix B).

The subroutine ISSUE collects statistics on the interval of time a class of vehicles (Shuttle, Tug, or SEPS) was unavailable. The total weight and length of the payload group are computed and printed on the first Monte Carlo cycle. The subroutine CSPAY is called to gather statistics on the individual payloads. For all payloads being launched, the following events are set up and put into the event queue:

a. If the payload is being deployed, the event ARRIV is generated, and the occurrence of the launching of the payload is recorded in the chronological time history by the subroutine STATUS.

b. If the payload is to be retrieved, the events REMOV and BACK are generated.

c. In the special case of a Sortie flight, the payload will be launched, arrive, and be removed.

For each vehicle used in the launch, the following is done:

1. The event REFVE is generated for when the vehicle becomes available again after completion of the flight and its associated refurbishment period.

2. The specific vehicle in its respective fleet is tagged as currently unavailable.

3. The number of flights for each vehicle involved is increased by 1.

4. If the vehicle is the uppermost stage, the weight of the payload group is accumulated for that vehicle and the number of times that vehicle is an uppermost stage is increased by 1.

3.1.4.8 Subroutine LDAT

The subroutine LDAT (load data) is called once at the beginning of the run from the event BEGIN. The function of this subroutine is to call the necessary subroutines to load the individual data groups from cards into the memory. After the subroutine LDAT is called, the input error flag IRFLG is set to 0, and the following subroutines are called: LDVEH, LDORB, LDMOD, LDSAT, LDSYS, LDSCH, LDME, and LDPUR, each with the input error flag IRFLG as a parameter. If the error flag is still 0 (meaning no known input errors), control is returned to the event BEGIN. Otherwise, the run is stopped with the message: --------RUN STOPPED DUE TO DATA ERROR--------

3.1.4.9 Subroutine LDME

The subroutine LDME is called once from the subroutine LDAT for the purpose of loading the data for the mission equipment upgrade schedules from cards. When the subroutine LDME (IRFLG) is called, IRFLG is the input data error flag. A non-zero value implies a data error was detected.

This subroutine reads a variable number of cards, and the extent of the card group is marked by a blank last card. The format of the data card was previously described in detail under input data in the User Guide section of this document. The data cards contain the system identifier, the system member, the name of the old mission equipment, the number of times that the item is repeated on the satellite, date of the upgrade, and the name of the new mission equipment. The old equipment module is located in the system on the satellite, and the event NEWME is set up for the date of upgrade and put into the event queue. Error messages are printed if the system cannot be found, when neither mission equipment module is identified as mission equipment, or if the mission equipment module is not on the satellite.

3.1.4.10 Subroutine LDMOD

The subroutine LDMOD is called once from the subroutine LDAT for the purpose of loading the data for the modules from cards. IRFLG is again the input data error flag, and a non-zero value indicates a data error. The subroutine LDMOD reads the number of input modules from a card. The number is then compared against the maximum capacity variable (see Appendix B), MITAB and, if excessive, an error message is printed along with setting IRFLG to non-zero. If the error did occur, the flight would probably fail. The module cards are read in (one module per card), and each is promptly printed for user reference and verification.

3.1.4.11 Subroutine LDORB

The subroutine LDORB is called once from the subroutine LDAT for the purpose of loading the data for the orbits from cards. As before, IRFLG is the input data error flag and a non-zero value indicates a data error.

This subroutine reads the number of orbits from a card. The number is compared against the maximum capacity variable NORBS (see Appendix B), and, if excessive, an error message is printed along with setting the IRFLG flag to non-zero. Such an error would probably cause the flight to fail. The orbit data cards are read in (one orbit per card), and each is promptly printed for user reference and verification. The vehicle name on the orbit data card is checked against the vehicle input table, and the name is replaced by a pointer if the vehicle is not found in the vehicle input table.

3.1.4.12 Subroutine LDPUR

The subroutine LDPUR is called once from the subroutine LDAT for the purpose of cross checking the input data. The user is informed of unused modules, satellites, and systems and the appropriate values to use on table sizes. (This reduces memory to a minimum and may permit the run to be completed.)

As the subroutine LDSCH processes the satellite launch schedules, the variable MARKS (see Appendix B) is set non-zero to indicate the satellite was used. In this routine, each system is scanned for required satellites, and the satellite input table and module input table have the needed satellites and modules marked by special non-zero entries. Unused systems are identified in the printout, and, at the end of the scan of the systems, a message is printed as to the number of systems this model actually uses. Then the satellite and module input tables are scanned to list unused satellites and modules and provide the true required number of satellites and modules.

3.1.4.13 Subroutine LDSAT

The subroutine LDSAT is called once from the subroutine LDAT for the purpose of loading the data for the satellites from cards. IRFLG is again the input data error flag, and a non-zero value indicates a data error.

The subroutine LDSAT reads the number of input satellites from a card, and the number is compared against the maximum capacity variable SITAB (see, Appendix B). If the number is excessive, an error message is printed along with setting the IRFLG to non-zero. If the error did occur, the program would probably crash. The satellite data cards are read in groups with two cards for the basic satellite data and one or more cards listing the modules on the satellite. The entire group is then promptly printed for user reference and verification, and each satellite data group is also printed for user verification and reference. In addition, certain input names are replaced by pointers to tables for later use by the program. Each satellite is assigned to an orbit which must be described in the orbit input table, and similarly each module must be in the module input table. For each module, an item will be filed in the set MDS (Appendix A) for the i^{th} satellite.

3.1.4.14 Subroutine LDSCH

The subroutine LDSCH is called once from the subroutine LDAT for the purpose of loading the data for the deterministic launch schedules from cards. IRFLG continues to be the input data error flag, and a non-zero value indicates a data error was detected.

The subroutine LDSCH reads a variable number of schedule cards where the extent of the card group is marked by a last blank card. The format of the data card is described in detail under input data in the Users Guide section of this document. The data consists of a system name, a member number, and the date of launch. After the system name is located and the member number identified, that information with the date of launch are filed in the set NEWS (see Appendix B) for every item to be launched. Error messages are printed if the system cannot be found or the member number is invalid.

3.1.4.15 Subroutine LDSYS

The subroutine LDSYS is called once from the subroutine LDAT for the purpose of loading the data for the systems from cards. IRFLG is still the input data error flag, and a non-zero value indicates a data error.

This subroutine reads the number of input systems from a card, and the number is compared against the maximum capacity variable STSTB. If the number is excessive, an error message is printed along with setting the IRFLG to non-zero. If the error did occur, the program would probably crash. The system data cards are read in groups of one or two cards, and each is printed for user reference and verification. The satellites that are members of the system are found in the data cards and located in the satellite input table. The names are replaced by pointers to the satellite input table unless the satellite was not found, in which case an error message is printed.

3.1.4.16 Subroutine LDVEH

The subroutine LDVEH is called once from the subroutine LDAT for the purpose of loading the data for the vehicles from cards. IRFLG continues to be the imprint data error flag, and a non-zero value indicates a data error.

This subroutine reads the number of input vehicles from a card, and the number is compared against the maximum capability variable NVEH. If the number is excessive, an error message is printed along with setting the IRFLG to non-zero. If the error did occur, the programs would probably crash fatally. The vehicle data cards are read in, one vehicle per card, and each is promptly printed for user reference and verification.

3.1.4.17 Subroutine MARKQ

This subroutine takes all the payloads in the loading queues scheduled for delivery to a single orbit and assigns them to the next flight. All items in the orbit queue are entered into the flight queue (the array ILOAD) up to the maximum capacity of the flight queue IL (see Appendix A).

3.1.4.18 Subroutine MCMOD

This subroutine consolidates the statistics for modules at the end of each Monte Carlo cycle. For each module in the module input table, the totals are accumulated, and the maximum and minimum counts are found for modules replaced or failed and for warnings of impending failure received.

3.1.4.19 Subroutine MCSAT

This subroutine consolidates the statistics for satellites at the end of each Monte Carlo cycle. For each satellite position in orbit,, the total weight factors are accumulated, and the maximum and minimum values are found for each wholly delivered or retrieved satellite, all traffic to the satellite, and each module belonging to the satellite. In addition, the totals, maximums, and minimums are found for the percentage availability of each satellite position in orbit.

3.1.4.20 Subroutine MCSYS

This subroutine consolidates the statistics for satellite systems at the end of each Monte Carlo cycle. For each satellite system, the totals, maximums, and minimums are accumulated for the total weight factors in the system and the percentage availability of the system.

3.1.4.21 Subroutine MCVEH

This subroutine consolidates the statistics for the vehicle fleets at the end of each Monte Carlo cycle. For the flights each year, the totals, maximums, and minimums are accumulated for the Shuttle, Tug, and SEPS vehicles. This same information is also provided for the total flights over all years.

3.1.4.22 Subroutine PASER

This subroutine accepts a list of payloads destined for the same orbit and reorders the list into a sequence of phase angle positioning around the orbit. The rules for ordering are:

a. Payloads are ordered by increasing phase angle.

b. The biggest gap in the circle is located, and the necessary payloads have their phasing positions decreased by 360 degrees such that when the list is again in increasing sequence, the gap is outside (on the end) of the list.

c. If there is a satellite in the list, the one nearest the beginning of the list is moved to the beginning of the list.

d. The logic can reorder the list in the reverse direction if the satellite is at the "wrong end."

The parameters are accessible in the definition table in the variables NQ and ILOAD (Appendix B). The ILOAD array points to the individual payloads in the loading queue and is therefore rearranged in the desired sequence.

3.1.4.23 Subroutine PAYLQ

The function of this subroutine is to put a payload into its orbit queue and the loading queue. The subroutine PAYLQ is called with the parameters IS and IM where IS is the pointer to the specific satellite in an orbit position and IM is the pointer to the element MODSY (Appendix A) identifying the specific module. If it is 0, the item is a satellite.

The elements ITORB and PAYLD (Appendix A) are created, values for all members of each element are defined as necessary, and the elements are filed into the orbit queue and loading queue.

3.1.4.24 Subroutines PROP and PROP2

These subroutines have the responsibility for computing the propellant required to deliver a group of payloads to their respective orbital destinations on a specified upper stage. The subroutine PROP is called with the parameter IGO where IGO is a vehicle availability flag. If IGO is not 0, then the flight cannot be flown at this time.

The other parameters are accessible in the definition table and are the portion of the loading queue as defined by variables NQ and ILOAD (Appendix B) and vehicle data. The subroutine returns the flight configuration in terms of sequence and time of arrival after launch or a no-go status which indicates the payload group cannot be delivered due to propellant restrictions or that the group is of excessive volume.

The functions performed within this subroutine can be separated into options for Sortie, Shuttle only, SEPS, and standard upper stage delivery.

The Sortie option checks the loading queue to determine how many Sorties are waiting. If no Sorties are in the queue, the routine looks for other options. If only one Sortie exists, all other payloads are ignored, and the next available launch is assigned to carry a single Sortie payload. If more than one Sortie is in the queue, the flight is rejected. No volume check is performed as the delivery is a single payload.

The Shuttle-only option determines the weight of payloads to be delivered and to be retrieved. If either value exceeds the variable WCONS as input on the appropriate Shuttle card, the flight is declared no go. A volume check is also performed.

The SEPS option is used for orbits that have SEPS specified on the orbit data card. The flight is first tried on the upper stage without the SEPS, and, if the upper stage alone can make the flight to orbit, then the SEPS will be used only for phasing. The payload queue could continue to grow due to the waiting feature of the remainder of the program. In that case, if an upper stage vehicle could not perform the flight, the flight would be made using the upper stage and the SEPS.

The standard upper stage delivery option comprises the major portion of the code in the routine. This option includes the following features:

a. The number of service units is determined from the number of modules on the flight and the permissible number of modules in a service unit.

b. The volumetric check is made.

c. The data for the orbit is obtained from the definition area.

d. The weights for deployment, service, and retrieval are determined from the loading queue.

e. The vehicle data is passed stage by stage to the FORTRAN code via the FORTRAN linkage subroutine LINKT.

f. The subroutine PASER is called to determine the delivery sequence of the payloads in the loading queue.

g. The ordered loading queue of payloads is changed to a series of orbital legs, each carrying a weight and requiring a ΔV from the vehicle.

h. The FORTRAN subroutine PRFORM is called to determine the amount of propellant remaining at the end of the flight composed of orbital legs.

i. If the amount of propellant remaining is positive, the data pertinent to the flight are saved in the loading queue and the orbit queue.

3.1.4.25 Subroutine QDMP

The function of this subroutine is to remove duplicate payloads from the loading queue and to inhibit modules from entering the queue when their satellite is in the queue. It is called whenever a new payload is about to be put into the loading queue. The subroutine QDMP is called with the parameters IS, IM, and ILL where IS is the pointer to the specific satellite in an orbit position, IM is the pointer to the new element MODSY (Appendix A) identifying the specific module or if 0, the item is a new satellite, and ILL is a flag with two values: =0 implies new payload can be put into the loading queue and $\neq 0$ implies new payload cannot be put into the loading queue.

The following rules apply to comparing the new payload against the entire portion of the loading queue pertaining to the orbit of the new payload.

a. If the new payload is a Sortie mission, the routine discontinues processing and returns.

b. If the new payload is to be retrieved, and there are several satellites still in the orbit position, the routine returns.

c. If the orbit queue is empty, the routine returns.

d. If the new payload is identical to a payload in the queue, the previous payload is removed from the queue by the subroutine DROPQ.

e. If the new payload is a satellite, and a module is destined for delivery to the same orbit position as the satellite, the module is removed from the loading queue by the subroutine DROPQ.

f. If the new payload is a module, and a satellite is destined for delivery to the same orbit position as the module, the flag ILL is set to 1, and the routine returns.

3.1.4.26 Subroutine QUAD

This subroutine is used to force angles into the region from 0-360 degrees. The subroutine QUAD is called with the parameter A where A is the input angle to be put into the region 0-360 degrees. If A is negative, A is increased by 360 degrees until A is positive. If A is greater than 360 degrees, then A is decreased by 360 degrees until A is less than 360 degrees.

3.1.4.27 Subroutine REDUN

This subroutine will determine the redundant nature of a given module on a specific satellite. The subroutine REDUN is called with the parameters IS and IM where IS is an index to the satellite at an orbital position and IM is a pointer to the module on a satellite.

The variable DELTA is accessible via the definition section (Appendix B). The subroutine will locate the module, determine if it is in a redundant group, and set DELTA as follows:

a. 0 Failure of this module will render the satellite inoperative.

b. 3000 Failure in a redundant group with satellite still operative.

c. -3000 Last permissible failure in a redundant group; one more failure and DELTA will be 0.

Any module so tagged with a positive value is referred to as a FREEBIE. Such modules are entered at the end of the loading queue and are permitted to fly on mandatory flights only as excess baggage. In no way are they permitted to flush the loading queue when they are put into the queue. FREEBIES are ranked by the value 3000, plus the time on the simulation clock, plus the satellite priority, plus 1000 times the number of FREEBIES remaining in the subsystem and have not failed.

Use of the third entry in the subsystem can cause a FREEBIE to assume the value of -3000. This will cause a FREEBIE to be entered as a normal payload into the loading queue. However, the satellite status is unchanged by the module failure and the module is not replaced on a mandatory basis.

3.1.4.28 Subroutine SAVER

This subroutine is used to create satellite replacements automatically. It is called when the satellite becomes operational because an entire satellite is put in orbit, and it is called when an NRU fails. The subroutine SAVER is called with the parameters T2 and IS where T2 is the moment of termination of the satellite and IS is the index to the satellite in orbit. The subroutine generates the events NWSAT and RETRI as necessary for the policy variable POLDN (Appendix B).

3.1.4.29 Subroutine SHIP

This subroutine receives payloads to be launched from events that have just occurred. The subroutine controls the loading algorithm and makes the decision of when to launch. The subroutine SHIP is called with the parameters IS and IM where IS is the index to the satellite in orbit table and IM is the address pointer to the module on the satellite (if not 0).

The payload is entered into the loading queue via calls to subroutines QDMP and PAYLQ. The entire queue is scheduled for launch (subroutine MARKQ), and the flight is attempted (subroutine PROP). If there is usable propellant remaining at the end of the flight, the subroutine returns. Otherwise, the record of the previous good launch is checked, and, if present, that set of payloads is launched (subroutine ISSUE). If the record is not present, the subroutine will construct a series of payload groups beginning with the first payload alone, save the record (if good), add the next payload from the first-in first-out queue to the group and iterate until the payload group exceeds the launch capability. The preceding group of payloads can then be launched. The subroutine includes logic to upgrade the launch vehicle to expendable if a single payload requires the increased performance. The loading sequence is tried again, and usually the entire loading queue can be flown. The subroutine would then return leaving the record of a good launch on an expended vehicle.

The subroutine can also be entered with IS equals 0 for the occurrence of a mandatory launch forced by a payload in the loading queue or with IS less than 0 for the occurrence of a vehicle becoming available and the sensing of an impending launch being delayed by no vehicle available for the launch. If there is space available on a mandatory launch, modules destined for redundant systems can be carried. If a payload is too heavy to be carried on an expendable vehicle, the subroutine deletes the payload from the queue and enters a message in the time history.

3.1.4.30 Subroutine STATUS

This subroutine is called when events occur for modules or satellites and performs three functions:

a. The status (active, inactive, or permanently disabled) is determined for the module, its satellite, and its system.

b. The statistics are gathered for availability and outage intervals for the satellites and the system.

c. The routine prints the lines in the two chronological time histories generated on the first Monte Carlo cycle.

The subroutine STATUS (IS, IM, IST) is called with the parameters IS, IM, and IST where IS is the index to the satellite in orbit table, IM is the pointer to the module on the satellite (if it is not 0, and IST varies from 1-10, depending on which event is occurring.

3.1.4.31 Subroutine TEREV

This subroutine is intended to print a summary of the number of times each event occurred on the average. Additionally, it was to print other run parameters and statistical quantities as yet to be defined. The routine currently does nothing.

3.1.4.32 Subroutine TERMD

This subroutine prints the statistical summary at the end of the run showing the failures, warnings, and replacements on all satellites for all modules.

3.1.4.33 Subroutine TERSY

This subroutine prints the system summary at the end of the run for the statistics for module replacement and associated load factors for each module on its individual satellite in orbit (the satellite in orbit is displayed by replacement of whole satellites with associated load factors and percentage availability and outage intervals).

3.1.4.34 Subroutine TERV1

This subroutine prints the flight summary at the end of the first Monte Carlo cycle and the end of the run for Shuttles, upper stages, and SEPS for each year followed by a totals line and the count of upper stages expended in the run (if not 0).

3.1.4.35 Subroutine TERV2

This subroutine prints the weight summary at the end of the first Monte Carlo cycle and at the end of the run for the average launch weight delivered to each orbit by Shuttles, upper stages, and SEPS.

3.1.4.36 Subroutine WEIBUL

This subroutine is called whenever a module is switched on or to an active state. The purpose of the subroutine is to predict the random occurrence of a failure and/or warning of the module at some time in the future. The subroutine WEIBUL is called with the parameters AW, BW, TW, AF, BF, and TF where AW is the Weibul parameter α for warning, BW is the Weibul parameter β for warning, TW is the predicted time interval to the occurrence of the warning, AF is the Weibul parameter α for failure, BF is the Weibul parameter β for failure, and TF is the predicted time interval to the occurrence of the failure.

The following alternatives are covered by routine WEIBUL:

a. If AW is 0 for the module, TW is then 0, implying no warning can ever occur for this module.

b. If AF is 0 for the module, TF is then 0, implying no failures can ever occur for this module.

c. If AW is 0 and AF is not 0, TF is defined by

$$TF = - AF \cdot \left(\log_e R_n\right)^{1/BF}$$

where R_n is the next sample from the random number generator.

d. If both AW and AF are not 0, then TW is defined by

$$TW = - AW \cdot \left(\log_e R_n\right)^{1/BW}$$

and TF is defined by

$$TF = - AW \cdot \left(\log_e R_n \cdot e\left(\frac{TW}{AF}\right)^{BF}\right)^{1/BF}$$

This form assures TF > TW.

3.2 ELEMENTS OF FORTRAN IV

3.2.1 Linkage Subroutines List

The following linkage subroutines exist for the sole purpose of passing data between SIMSCRIPT subroutines and FORTRAN subroutines. The FORTRAN subroutines then communicate the data via labeled common. This is done to expedite the integration of the subroutines written in the two languages.

3.2.1.1 Subroutine CON

This subroutine is used to convert the NRU or redundant entries for each module on each satellite. The subroutine CON is called with the following parameters:

a. I is the four-character entry after the module on the data card.

b. K is the returned value of the converted entry as follows:

   1. 0 is blank or 0(SRU).
   2. 1-15 is 1-15 (redundant SRU).
   3. 100 is anything else (NRU).

3.2.1.2 Subroutine CONEC

The subroutine CONEC is called with the following parameters:

a. NS is the number of stages on the vehicle.

b. NVEH is not used.

c. ISEPS is the SEPS flag, and non-zero means the SEPS is to be used.

3.2.1.3 Subroutine LDSEP

The subroutine LDSEP is called with the following parameters:

a. A is the structural weight.

b. B is the solar electric propulsion efficiency.

c. C is the solar cell power for thrust.

d. D is the I_{sp}.

e. H is the performance reserve factor.

f. I is the vehicle expend flag.

g. F is the total thrust time in days.

h. G is the total propellant onboard the SEPS.

3.2.1.4 Subroutine LINKT

The subroutine LINKT is called with the following parameters:

a. I is the number of the stage.

b. A is the I_{sp} of the stage.

c. B is the inert weight of the stage.

d. C is not used.

e. D is the maximum propellant in the stage.

f. E is the expend flag, and a non-zero value means expend.

g. F is the solid flag, and a non-zero value means a solid stage.

h. G is the gross weight of the stage including payload.

3.2.1.5 Subroutine SEPSV

The subroutine SEPSV is called with the following parameters:

a. N is the number of service legs.

b. PER is not used.

c. VS is not used.

d. DT is the array of differences in phase angles for each service position.

e. PAY is the corresponding array of payload weights to be transported across each phase angle.

3.2.1.6 Subroutine TPHAS

The subroutine TPHAS is called with the following parameters:

a. A is the array of the time of flight of the SEPS during a sequence of phasing maneuvers.

b. N is the number of maneuvers.

3.2.1.7 Subroutine TWOBR

The subroutine TWOBR is called with the following parameters:

a. DV is the total ΔV for the transfer.

b. DV1 is the ΔV for the first burn of two.

3.2.1.8 Subroutine GETFR

The purpose of this routine is to provide a linkage between SIMSCRIPT and the FORTRAN binary read capability. The routine is called with the following parameters:

a. FR is a four-word array to be filed in the set FRS.

b. LL + 9 is the number of the tape to be read.

c. IK is a flag (non-zero for end of data).

3.2.1.9 Subroutine PUTFR

The purpose of this routine is to provide a linkage between SIMSCRIPT and the FORTRAN binary write capability. The routine is called with the following parameters:

a. FR is a four-word array to be filed in the set FRS.

b. LL + 9 is the number of the tape to be used.

c. IK is a flag (non-zero for end of data).

3.2.2 Vehicle Performance Subroutines List

The subroutines in the following paragraphs are coded in FORTRAN and provide various vehicle performance computations.

3.2.2.1 Subroutine PRFORM

This subroutine determines which performance subroutine is to be called. The subroutine PRFORM is called with the following parameters:

a. DVLEG is the name of the array containing the ΔV for each leg in the order of deployment.

b. PLEG is the name of the array containing the payload weight for each leg.

c. NLEG is the integer number of legs to be simulated.

d. WPER is the minimum percentage value of the residual propellant or the excess capability in gross weight.

e. NEXIT is the type of mission exit (1-9) from the subroutine SEPX; i. e., whether the mission is possible or not.

f. ERFLG = 0 (do not erase the previous maneuver).
= 1 (erase the previous maneuver).

g. NT is the number of Tugs used in the SEPS mode.

The other data required to compute performance are passed via the linking subroutines.

3.2.2.2 Subroutines SEPX, SEPIM, TUGCP, CRY01, INTORB, SEPDV, PLUPD, and FAZS

This is an integrated group of subroutines designed to determine the performance capability of a specified Space Tug/Solar Electric Propulsion Stage (SEPS) combination for any number of deployment/servicing/retrieval missions to synchronous equatorial orbit.

In normal usage, a call is made to the executive subroutine SEPX containing information on the mission to be flown (e. g., weight to be deployed, weight to be retrieved, and servicing maneuvers to be performed in synchronous orbit). The program determines the optimal (minimum SEPS fuel) intermediate orbit at which changeover should occur based upon the characteristics of the Tug and SEPS vehicles which are stored in a common area. Program outputs include intermediate orbit altitude and inclination, SEPS ascent and descent times, and SEPS fuel remaining at the end of the mission.

The subroutine SEPX is called with the following parameters:

a. MPLA is the weight of payload to be deployed (lb)

b. MPLB is the weight of payload to be retrieved (lb)

c. ERFLG = 0 (do not erase the previous maneuver).
= 1 (erase the previous maneuver).

d. NEXIT is the type of mission exit (1-9) from the programs; i.e., whether the mission is possible or not.

The following variables are passed via labeled common.

a. SEPS data:

1. MS is the mass of the SEPS (lb).
2. P is the solar cell power (watts).
3. E is the solar electric propulsion efficiency.
4. SISP is the SEPS I_{sp} (sec).
5. TSEP is the thrusting time available on a fully fueled SEPS (days).
6. MPT is the mass of propellant on board (lb).
7. TLEFT is the corresponding thrusting time (days).
8. RSEP is the performance reserve factor for the SEPS.
9. SG is the gravity constant (ft/sec^2).

b. Tug data:

1. TWS is the Tug structural weight (lb).
2. TWPA is the allowable Tug propellant weight (lb).
3. TISP is the Tug effective I_{sp} (sec).
4. TG is the gravity constant (ft/sec^2).
5. TWGA is the gross weight allowable (lb).
6. TR is the performance reserve factor for the Tug.

c. Returned data:

1. NTUGS is the number of Tug flights required to perform the mission and return the expended SEPS, if required.
2. TLEFT is the SEPS life remaining after accomplishing the mission (days).
3. MPT is the SEPS fuel remaining after accomplishing the mission (lb).
4. HGO is the optimal changeover orbit altitude (nmi).

5. ICOS is the optimal changeover orbit inclination (deg).

6. TU is the time required by the SEPS for ascent maneuvers (days).

7. TD is the time required by the SEPS for descent maneuvers (days).

The simulation of the Space Tug/SEPS operation is performed using the following set of subroutines:

3.2.2.2.1 SEPX

This is the executive subroutine. In it, the decisions are made as to when a new SEPS must be launched or retrieved and whether the given deploy and retrieve payloads commands can be handled in a single Tug/SEPS encounter or whether additional Tug flights are required to support the mission. This logic has been kept separate from the SEPS performance equations to facilitate logic changes incorporating mission modes other than the ground-based mode which is presently baselined.

3.2.2.2.2 SEPIM

In this subroutine, the performance of the Tug/SEPS combination is computed by calls to more detailed performance subroutines. The delivery, servicing, and retrieval of payloads are accomplished, and the fuel and time remaining on the SEPS are decremented accordingly.

3.2.2.2.3 TUGCP

This subroutine calls the appropriate Space Tug configuration to determine the Tug capability. Presently only a single-stage cryogenic Tug performance model (CRY01) is available.

3.2.2.2.4 CYR01

In this routine, the ΔV capability of a single-stage cryogenic Space Tug carrying the required deploy and retrieve payloads is computed. The Tug is initially either filled with propellant or filled to the gross weight limit of the Shuttle. Impulsive ΔV's are assumed with a specified performance reserve to allow for finite burn effects and propellant margin.

3.2.2.2.5 INTORB

From the ΔV which can be supplied by the TUG, this subroutine computes the optimal (minimum SEPS fuel required) changeover orbit altitude and inclination. This is done by a table look-up procedure, since the optimization study for a synchronous equatorial mission orbit has been previously accomplished in a manner that is dependent only upon the ΔV which can be supplied by the Tug.

3.2.2.2.6 SEPDV

Given the optimal changeover orbit calculated by subroutine INTORB, SEPDV determines the ΔV (and corresponding mass ratio) which must be expended by the SEPS. Edelbaum's simplified equations for low-thrust vehicles are used in calculating the ΔV required.* Again, a performance reserve factor is provided.

3.2.2.2.7 PLUPD

With the mass ratio from subroutine SEPDV, this subroutine decrements both the propellant on board and the SEPS remaining lifetime for each payload carried up or down by the SEPS.

3.2.2.2.8 FAZS

The subroutine performs the on-orbit servicing maneuvers using the SEPS and decrements the fuel and lifetime of the SEPS accordingly.

3.2.2.3 Subroutine SSHOT

This subroutine computes the propellant necessary to deploy payloads by the "slingshot" method. The subroutine can handle

*Edelbaum, T. N., "Propulsion Requirements for Controllable Satellites," ARS Journal, pp. 1079-1089 (August 1961).

up to 3 liquid (variable burn) stages and 10 legs. The "slingshot" method refers to a type of deployment in which the lower stages boost the upper stage into the service orbit. The stages may be recoverable or not and only the upper stage does the servicing. A leg is the service loop, that part of the trajectory between two service/deploy events.

The subroutine SSHOT is called with the following parameters:

a. DVLEG is the name of the array containing the ΔV for each leg in the order of deployment.

b. PLEG is the name of the array containing the payload weights for each leg.

c. NLEG is the integer number of legs to be simulated.

The following variables are passed via labeled common.

a. WS is the name of the array for the structure weights for each stage (lb).

b. WPA is the name of the array for the allowable propellant weight for each stage (lb).

c. EISP is the name of the array for the effective I_{sp} for each stage (sec).

d. NSTG is the integer number of stages to be processed.

e. FEAS(1) is the ratio of the weight of the propellant necessary to the weight of the propellant available.

f. FEAS(2) is the ratio of the gross weight necessary to the gross weight allowable.

Given the propellant allowable, the structure weights, and the I_{sp} for each stage and the ΔV required for the payloads for each leg, the subroutine computes the amount of fuel necessary to perform the mission. The weight of the propellant is then divided by the weight allowable and the ratio is returned to the user as an index of feasibility.

3.2.2.4 Subroutine SSLQD

This subroutine computes the propellant necessary to deploy payloads with a single liquid stage. There may be multiple deliveries (legs) required.

The subroutine SSLQD is called with the following parameters:

a. DVLEG is the name of the array containing the ΔV for each leg in the order of deployment.

b. PLEG is the name of the array containing the payload weight for each leg.

c. NLEG is the integer number of legs to be simulated.

The following variables are passed via labeled common:

a. WS is the name of the array for the structure weight for the stage (lb).

b. WPS is the name of the array for the allowable propellant weight for the stage (lb).

c. EISP is the name of the array for the effective I_{sp} for the stage (lb).

d. G is the gravity constant (ft/sec^2).

e. WGA is the allowable gross weight (lb).

f. FEAS(1) is the ratio of the weight of the propellant necessary to the weight of the propellant available.

g. FEAS(2) is the ratio of the gross weight necessary to the gross weight allowable.

Given the weight allowable, the structure weights, the I_{sp} of the stages and the ΔV required for each payload, the subroutine computes the required propellant and then forms the ratio of the total weight to the allowable weight. This ratio is returned as an index of feasibility.

3.2.2.5 Subroutine TRNKC

This subroutine computes the propellant necessary for the liquid burn stage with the restriction that the solid stage(s) is totally consumed. The subroutine will handle either one or two kicks to boost a single payload into its orbit.

The subroutine TRNKC is called with the following parameters:

a. DVLEG(1) is the name of the array for the ΔV requirement for a low-altitude burn.

b. DVLEG(2) is the name of the array for the ΔV requirement for a high-altitude burn.

c. PLEG(1) is the name of the array containing the weight of the payload to be deployed.

d. NLEG is set equal to 2.

The following variables are passed via labeled common.

a. G is the gravity constant (ft/sec^2).

b. WGA is the allowable gross weight (lb).

c. REUSE is the name of the array containing the flags denoting whether the stage is reusable or expendable. 0 = expendable and non-zero = reusable.

d. FEAS(1) is the ratio of the weight of the propellant necessary to the weight of the propellant available.

e. FEAS(2) is the ratio of the gross weight necessary to the gross weight allowable.

f. WS is the name of the array for the structure weights for each stage (lb).

g. WPA is the name of the array for the allowable propellant weight for each stage (lb).

h. EISP is the name of the array for the effective I_{sp} for each stage (sec).

i. NSTG is the integer number of stages to be processed.

Given the allowable propellant weight, the structural weights, and the effective I_{sp} for each of the stages, the subroutine computes the solid stages(s) capabilities. Then, if needed, it determines the remaining requirement for the liquid first stage. It is assumed that the solid stage(s) is totally expended, but if it is not expended, the residuals are burned off by yaw steering.

APPENDIX A: DESCRIPTION OF QUEUES

A queue is defined as a waiting line. In the LOVES Computer Program, there are seven queues in operation.

EVENT	-	Events which will occur in the future
FRS	-	Summary of activity for each satellite
ORBQ	-	A multi-dimensional loading queue containing all subqueues each of which pertains to a particular orbit
MDS	-	Modules initially comprising each satellite
MES	-	Input mission equipment changes
MOD	-	Modules currently comprising each satellite
NEWS	-	Input satellite launch schedule

Each queue contains elements consisting of groups of computer words. For example, all events require four words of memory. The words within a group can be subdivided so that the data can be packed. LOVES uses 1/4 word minimum for some integer variables, 1/3 word for some larger integer variables, 1/2 word for integers that could be addresses, and full words for floating point variables.

In the following pages of this appendix, each queue is described in detail.

A.1 THE EVENT QUEUE

The EVENT queue is automatically handled by the SIMSCRIPT system as to which is the next event to occur. The programmer sets up the events, and, when they occur, he destroys each event. The events used in the LOVES Computer Program were described in detail in the Programmers Manual section of this document and are listed below:

External: BEGIN

Internal:					
ARRIV	LAUNC	QWAIT	REFVE	SATDN	WARN
BACK	NEWME	REFMO	REMOV	START	
FAIL	NWSAT	REFSA	RETRI	TERM	

PRECEDING PAGE BLANK NOT FILMED

Each event is four words in size. The first two words of each event are reserved for SIMSCRIPT bookkeeping. The third word has two definitions, depending on the event.

TIME1

or

PSAT	PMOD

a. TIME1 - is a time-data word.

b. PSAT - is a 1/2-word pointer to a satellite position in the orbit.

c. PMOD - is a 1/2-word address pointing to a module on the satellite.

The fourth word has three definitions, depending on the event. Each uses the full word:

a. TIME2 - is the second time-date word.

b. VNAME - is the name of a vehicle.

c. TIMEA - is the time of arrival of a satellite in orbit. It is used to inform a module failure event that the original satellite is gone, and, therefore, the event really cannot occur.

A.2 THE QUEUE FRS

This queue retains a record of everything that occurs with respect to modules and satellites. No vehicle events are in the queue. The queue is built during the first Monte Carlo cycle and destroyed at the end of the cycle. The elements of the queue are written to ten scratch tapes on disc (to reduce memory requirements). The first 1/10th of the satellite positions are assigned to tape 10, the second 1/10th to tape 11, etc. At the end of the first Monte Carlo cycle, tape 10 is read back into memory and entered into the queue for ranking. After the history contained on the first tape has been printed, the queue is destroyed and the memory reused for the queue data on the next tape. The queue is ranked by satellite position in orbit, and, due to SIMSCRIPT filing equal elements at the end of the queue, the queue is kept in the correct time sequence. A simple scan of the queue produces the chronological time history of each satellite position. If TRIG2 is non-zero, the queue is not built, the report will not be printed, and the memory requirement for the program is reduced. The structure of an item in the queue is:

<table>
<tr><td colspan="2">SFRS</td><td colspan="2">PFRS</td></tr>
<tr><td colspan="4">TIMEF</td></tr>
<tr><td>SATNO</td><td>ST</td><td>SATSY</td><td>NPS</td></tr>
<tr><td colspan="2">MODNO</td><td colspan="2">NOSTA</td></tr>
</table>

The item is four words in size and called FR. The definition of the names in the item are:

a. SFRS - is a 1/2-word address pointing to the next item in the queue.

b. PFRS - is a 1/2-word address pointing to the previous item in the queue.

c. TIMEF - is the time at which an event occurred.

d. SATNO - is a 1/4-word pointer to the satellite orbit position involved in the event.

e. ST - is a 1/4-word for the status of the satellite after the event.

f. SATSY - is a 1/4-word for the status of the system after the event.

g. NPS - is a 1/4-word for the number of satellites at the orbit position after the event.

h. MODNO - is a 1/2-word address pointing to the module on the satellite, if it is non-zero.

i. NOSTA - is a 1/2-word for the event identifier.

A.3 THE QUEUE ORBQ

This queue contains all payloads available for deployment and/or retrieval at a given instant of time. The queue is ranked in ascending order based on the sum of the simulation time the payload enetered the queue, the priority of the satellite for which the payload is destined, and the parameter DELTA (associated with FREEBIES). The structure of an item in the queue is:

<table>
<tr><td colspan="2">SORBQ</td><td colspan="2">PORBQ</td></tr>
<tr><td colspan="2">IMOD</td><td>ISAT</td><td>IRT</td></tr>
<tr><td colspan="4">ANGLE</td></tr>
<tr><td colspan="4">PAYWT</td></tr>
<tr><td colspan="4">PAYLN</td></tr>
<tr><td colspan="4">GOTIM</td></tr>
<tr><td colspan="4">LQTIM</td></tr>
<tr><td colspan="2">MLEV</td><td colspan="2">$\bar{C}$ITEM</td></tr>
</table>

The item is eight words in size and called PAYLD. The definition of the names in the item are:

a. SORBQ - is a 1/2-word address pointing to the item in the queue.

b. PORBQ - is a 1/2-word address pointing to the previous item in the queue.

c. IMOD - is a 1/2-word address pointing to the module (if non-zero).

d. ISAT - is a 1/4-word index to a satellite in the orbit table.

e. IRT - is a 1/4-word flag. If it is 0, the payload is for deployment; if it is non-zero, the payload is for retrieval.

f. ANGLE - is the angular position in the orbit of a satellite.

g. PAYWT - is the weight of the payload.

h. PAYLN - is the length of the payload.

i. GOTIM - is the time interval between launch and arrival in the orbit of the payload.

j. LQTIM - is the time the payload enters the loading queue.

k. MLEV - is a 1/2-word address pointing to the waiting mandatory event in the event queue, or zero if no such event was set up.

l. CITEM - is a 1/2-word address pointing to the next payload in the loading queue assigned to the next flight. This address is ued to maintain the order of flight sequence for ease of interpretation in the printed reports.

A.4 THE QUEUE MDS

This queue contains the list of modules comprised in the original satellite. Therefore, there is one of these queues attached to each satellite. Each queue is first-in first-out, so that the printouts match the input data. The structure of an item in the queue is:

<table>
<tr><td>SMDS</td><td colspan="2">NOMOD</td></tr>
<tr><td colspan="2"></td><td>NRU</td></tr>
<tr><td colspan="3"></td></tr>
<tr><td colspan="3"></td></tr>
</table>

The item is four words in size and called MDSAT. The definition of the names in the item are:

a. SMDS - is the 1/2-word address pointing to the next item in the queue.

b. NOMOD - is the 1/2-word pointer to the module in the module table.

c. NRU - is a 1/4-word flag for the NRU, SRU, and redundant module.

A.5 THE QUEUE MES

This queue retains the schedule of mission equipment changes or upgrades as supplied by the input data deck. The queue is scanned at the beginning of each Monte Carlo cycle, and, for each item in the queue, an event NEWME is put into the event queue. The queue is last-in first-out to minimize the core requirement. The structure of an item in the queue is:

SMES	
MEDT	
PSAT	PMOD

The item is four words in size and called MESET. The definition of the names in the item are:

a. SMES - is a 1/2-word address pointing to the next item in the queue.

b. MEDT - is the date at which the mission equipment change is to occur.

c. PSAT - is a 1/2-word pointer to the satellite position in orbit.

d. PMOD - is a 1/2-word address of the module on a satellite to be upgraded.

A.6 THE QUEUE MOD

This queue contains the list of modules comprised in the current version of the satellite located at its orbital position. Therefore, there is one of these queues attached to each satellite in orbit. Each queue is first-in first-out. The structure of an item in the queue is:

<table>
<tr><td colspan="3">SMOD</td><td colspan="3">NOMOD</td></tr>
<tr><td colspan="3">EFAIL</td><td colspan="2">NUM</td><td>NRU</td></tr>
<tr><td>MAXNU</td><td colspan="2">MINNU</td><td colspan="2">MSTAT</td><td>SUMNU</td></tr>
<tr><td colspan="2">LOADF</td><td colspan="2">SUMLF</td><td colspan="2">MAXLF</td></tr>
<tr><td colspan="2">MINLF</td><td colspan="2">EDO</td><td colspan="2"></td></tr>
<tr><td colspan="3">EWARN</td><td colspan="3"></td></tr>
<tr><td colspan="6"></td></tr>
<tr><td colspan="6"></td></tr>
</table>

The item is eight words in size and called MODSY. The definition of the names in the item are:

a. SMOD - is a 1/2-word address pointing to the next item in the queue.

b. NOMOD - is a 1/2-word pointer to the module in the module table.

c. EFAIL - is a 1/2-word address pointing to a module failure event, if it is non-zero.

d. NUM - is a 1/4-word for the number of times the module alone was launched during a Monte Carlo cycle.

e. NRU - is a 1/4-word flag for the NRU, SRU, and redundant module.

f. MAXNU - is a 1/4-word for the maximum value of NUM over all cycles.

g. MINNU - is a 1/4-word for the minimum value of NUM over all cycles.

h. MSTAT - is a 1/4-word for the active/inactive status flag of the module.

i. SUMNU - is a 1/4-word for the sum of NUM over all Monte Carlo cycles.

j. LOADF - is a 1/3-word for the sum of the load factors changed to this module over one Monte Carlo cycle.

k. SUMLF - is a 1/3-word for the sum of LOADF over all cycles.

l. MAXLF - is a 1/3-word for the maximum value of LOADF over all cycles.

m. MINLF - is a 1/3-word for the minimum value of LOADF over all cycles.

n. EDO - is a 1/3-word flag for a warned or failed module in a group of redundant modules.

o. EWARN - is a 1/2-word address pointing to a module warning event, if it is non-zero.

A.7 THE QUEUE NEWS

This queue retains the launch schedule of satellites as supplied by the input data deck. The queue is scanned at the beginning of each Monte Carlo cycle, and, for each item in the queue, an event NWSAT is put into the event queue. This queue is LIFO (last-in first-out) to minimize the core requirement. The structure of an item in the queue is:

<table>
<tr><td>SNEWS</td><td>SCHSY</td></tr>
<tr><td colspan="2">SCHDT</td></tr>
</table>

The item is two words in size and called NEW. The definition of the names in the item are:

a. SNEWS - is a 1/2-word address pointing to the next item in the queue.

b. SCHSY - is a 1/2-word pointer to the satellite in the orbit table identifying which satellite is to be launched.

c. SCHDT - is the scheduled launch date of a satellite.

Because this queue is LIFO, satellites become available for launch in reverse order of the input data.

C.2

APPENDIX B: DEFINITION OF VARIABLES

This appendix provides a description of all the permanent variables entered in the Definition Section preceding the SIMSCRIPT Program Code. These variables are accessible to all SIMSCRIPT subroutines and events. The list is important, as it identifies the entries in the initialization section of the data deck. Items with numbers 60, 80, 85, 100, 120, 130, 140, 150, 180, 200, 230, and 270 are especially of interest, since they are used to dimension the vector-type variables immediately following them. The relationship can be seen by referring to the beginning of the sample printout in Appendix C.

GLOBAL VARIABLES AND CONSTANTS

ARRAY NO.	NAME	DESCRIPTION
1	UP	A six-letter constant used to print the status of an operating element.
2	DOWN	A six-letter constant used to print the status of a temporarily inoperative element
3	OUT	A six-letter constant used to print the status of a permanently disabled element
4	SHUT	A six-letter constant used to print the name of the Shuttle
5	TUG	A six-letter constant used to print the name of the upper stage
6	SEPS	A six-letter constant used to print the name of the SEPS vehicle.
7	BLANK	A six-letter constant containing blanks
8	G	Gravity constant.
9	TIMEB	Time to begin collection of statistics in years.
10	TIMES	Time to terminate collection of statistics in years.
11	TIMEC	Monte Carlo random number control variable. If = 0, program uses random numbers. If 0<TIMEC<1., the program uses TIMEC as the only random number sample.
12	PDOWN	Global policy for response to failure of an NRU on a satellite: =0, no response to failure. ≠0, satellite will be retrieved and/or deployed as per satellite policy of 0, 2, 3, 4.
13	NINSU	Maximum number of modules carried by a service unit.
14	WTSU	Weight of empty service unit.
15	LENSU	Length of service unit.

ARRAY NO.	NAME	DESCRIPTION
16	NMD	Number of modules assigned to current flight.
17	SU	Number of service units assigned to current flight.
18	WAIT1	The time interval in days after the termination of a satellite, shall the replacement satellite be put in loading queue (+ or -).
19	WAIT2	The time interval in days after the replacement satellite is scheduled, shall the retrieval of the satellite be scheduled into the loading queue (+ or -).
20	WSATU	The maximum wait for a satellite launch after a satellite enters the loading queue if the satellite to be replaced is operational.
21	WSATN	The maximum wait for a satellite launch after a satellite enters the loading queue if the satellite to be replaced is inoperative.
22	WMODU	The maximum wait for a module launch after a module enters the loading queue if the module to be replaced is operational.
23	WMODN	The maximum wait for a module launch after a module enters the loading queue if the module to be replaced is inoperative.
24	TRIG	The current count of the number of Monte Carlo cycles performed.
25	TRIG2	A flag; if zero, collect and print chronological history for each satellite.
26	TRIGS	Number of Monte Carlo cycles to be simulated.
27	EXVEH	Expend vehicle flag: $=0$, vehicle is reusable. $\neq 0$, vehicle is expendable.
28	PREFT	Pad refurb time in days.
29	TREFT	Upper stage refurb time in days.
30	SREFT	Shuttle refurb time in days.

ARRAY NO.	NAME	DESCRIPTION
31	PADT	Number of days delay between decision to fly and actual launch (integration, move to pad, preflight check).
32	DAYS	Number of days the upper stage can remain in orbit.
33	DV	Delta velocity required to get from parking orbit to final orbit.
34	FISP	I_{sp} for the upper stage
35	WD	Inert structural weight of upper stage.
36	WPNU	Weight of unused propellant of upper stage remaining at the end of maneuvers.
37	WCONS	Maximum weight of upper stage plus payloads (derived from either upper stage or Shuttle considerations).
38	IORB	Points to current orbit
39	NQ	Number of payloads going to the IORB orbit.
40	RA	Radius of apogee of final orbit.
41	VCO	Circular velocity of final orbit.
42	RO	Radius of the Earth.
43	P1	Period of the final orbit in hours.
44	RTFLG	Retrieve payload flag: =0, no retrieve. ≠0, retrieve.
45	PALEN	Maximum total length of multiple payloads on any flight.
46	FLYT	Total flight time of current flight.
47	WAIT3	The delay time between an NRU failure and the availability of a replacement satellite.
48	TLIMS	The maximum time of flight for a SEPS mission after disconnect from the Tug to the final payload operation.
49	TRIN	The performance reserve (0.01 for 1% margin).

SETS AND QUEUES

ARRAY NO.	NAME	DESCRIPTION
50	FNEWS	Pointer to the first item in the set NEWS.
51	FFRS	Pointer to the first element in the set FRS.
52	LFRS	Pointer to the last element in the set FRS.
53	FMES	Pointer to the first item in the set MES.

GLOBAL VARIABLES AND CONSTANTS

ARRAY NO.	NAME	DESCRIPTION
54	IQ	Flag; if non-zero, payload was put into loading queue.
55	SEPFT	SEPS refurb time in days.
56	MODB	The value of EXMOD before the year TIMEG.
57	EXMOD	Expendable model flag.
58	DELTA	Indication of redundancy: 0 implies group is out of service. +3000 implies group is still in service. -3000 implies one more failure and group is out of service.
59	EXTUG	A flag to expend the upper stage because payload is too heavy for reusable stage.

ORBIT TABLE

ARRAY NO	NAME	DESCRIPTION
60	NORBS	Number of orbits the simulation can deploy to and service.
61	ORBID	Name of orbit.
62	ORBDV	Delta V from parking orbit to final orbit.
63	ORBPD	Final orbit period in hours.
64	ORBRA	Apogee radius of final orbit in feet.
65	ORBVC	Equivalent circular orbit velocity in feet/second.
66	ORBTM	Total flight time for vehicle to leave launch pad and return to ground.
67	RQUP	Name of required upper stage or blank.
68	RQSEP	Name of required SEPS vehicle.
69	RQSUT	Name of required Shuttle vehicle.
70	PQUE	Pointer to first payload in loading queue assigned to current flight.
71	NL	Number of payloads currently assigned to a launch to each orbit.
72	ANMD	Number of modules on current flight to the orbit.
73	W	Percentage propellant remaining on this flight If negative, the payload group is too heavy or long.
74	NMDFL	Space available.
75	FORBQ	Pointer to the first payload in the loading queue.
76	LORBQ	Pointer to the last payload in the loading queue.
77	DV1	Delta V used in first burn of two-burn sequence (trans-kick).
78	EXORB	Flag used to indicate the upper stage serving this orbit is being used in an expendable or reusable mode.

GLOBAL CONSTANTS

ARRAY NO.	NAME	DESCRIPTION
79	MODS	The value of EXMOD after the year TIMEG.

PAYLOAD TABLE

ARRAY NO.	NAME	DESCRIPTION
80	IL	Maximum number of payloads that can be on any vehicle.
81	FLTIM	Flight time for each payload on current flight from launch to final orbit position.
82	ILOAD	Pointers to the loading queue for the payloads in the current flight.
83	PANGL	Phase angles between the payloads in the current flight

GLOBAL CONSTANTS

ARRAY NO.	NAME	DESCRIPTION
84	TIMEG	The year the program is to shift from an expendable model to a reusable model.

VEHICLE TABLE

ARRAY NO.	NAME	DESCRIPTION
85	NVEH	Number of different types of vehicles (or stages) to be used in the simulation.
86	NAMEV	Name of vehicle.
87	DAYSV	Number of days the vehicle can remain aloft.
88	ISPV	I_{sp} of vehicle or stage.
89	WDV	Inert structural weight of vehicle or stage.
90	WPNUV	Weight of unused propellant remaining in stage at end of maneuvers.
91	WCONV	Maximum weight of vehicle.
92	REFTV	Stage refurb time in days.
93	EXPV	Expend stage flag; if not zero, stage is to be expended.
94	PAYLV	Maximum total length of payload to be placed on this vehicle.
95	IDV	Space vehicle available.
96	NSTAG	Number of stages comprised in this vehicle
97	SOLID	Solid stage flag. Stage is solid if flag non-zero

GLOBAL CONSTANTS

ARRAY NO.	NAME	DESCRIPTION
98	PSERV	Service policy: 0 = All modules and SUs returned. 1 = No modules or SUs returned. 2 = Only SUs returned (modules left at last orbital position visited).
99	WAIT4	The number of days delay in checking out a module or satellite from the time of need until the time it can be put into the loading queue.

VEHICLE FLEET STATISTICS

ARRAY NO.	NAME	DESCRIPTION
100	NYEAR	Number of years for accumulation of vehicle statistics.
101	SUTFY	An array used to collect the number of Shuttle flights in each year.
102	SUM90	An array used to collect the number of Tug flights in each year.
103	MAX90	An array used to keep the maximum values of SUTFY over all Monte Carlo cycles.
104	MIN90	An array used to keep the minimum values of SUTFY over all Monte Carlo cycles.
105	TUGFY	An array used to collect the number of Tug flights in each year.
106	SUM39	An array used to keep the total number of Tug flights in each year for all Monte Carlo cycles.
107	MAX39	An array used to keep the maximum values of TUGFY over all Monte Carlo cycles.
108	MIN39	An array used to keep the minimum values of TUGFY over all Monte Carlo cycles.
109	SEPFY	An array used to collect the number of SEPS flights in each year.
110	SUM86	An array used to keep the total of all SEPS flights in each year for all Monte Carlo cycles.
111	MAX86	An array used to keep the maximum values of SEPFY over all Monte Carlo cycles.
112	MIN86	An array used to keep the minimum values of SEPFY over all Monte Carlo cycles.

GLOBAL CONSTANTS

ARRAY NO.	NAME	DESCRIPTION
113	LIMIT	A parameter used to control the number of FREEBIES in the loading queue (4000. implies none, 5000. implies the last FREBIE in the subsystem, etc.)

LAUNCH PAD TABLE

ARRAY NO.	NAME	DESCRIPTION
114	NPAD	The number of launch pads available.
115	VPAD	The list of launch pads showing which is available or not.
116	IPAD	The number of the launch pad selected for the next flight.

GLOBAL CONSTANTS

ARRAY NO.	NAME	DESCRIPTION
117	NOTUG	Flag used to inhibit the shift to an expendable Tug if a reusable Tug cannot do the delivery.
118	TUP	The upwards flight time for the SEPS going from the Tug to the first orbital position.
119	TDOWN	The downwards flight time for the SEPS coming down to meet the Tug for payload interchange.

VEHICLE FLEET STATISTICS

ARRAY NO.	NAME	DESCRIPTION
120	NSHUT	Number of Shuttles in fleet.
121	VSHUT	Shuttle availability array.
122	ISHUT	Pointer to currently available Shuttle.
123	MFSUT	Minimum number of Shuttle total flights in any Monte Carlo cycle.
124	IFSUT	Total number of Shuttle flights over all Monte Carlo cycles.
125	NFSUT	Maximum number of total Shuttle flights in any Monte Carlo cycle.

GLOBAL CONSTANTS

ARRAY NO.	NAME	DESCRIPTION
126	WUSEP	The weight of a SEPS complete with propellant ready to be delivered to orbit by the Tug.
127	WDNSP	The weight of an empty SEPS ready to be returned to the Earth's surface by the Tug.
128	LSEP	The length of the SEPS in feet.
129	QUIT	Flag (normally zero) used to have the run continue in spite of heavy payloads.

VEHICLE FLEET STATISTICS

ARRAY NO.	NAME	DESCRIPTION
130	NTUG	Number of Tugs in fleet.
131	VTUG	Tug availability array.
132	ITUG	Pointer to currently available Tug.
133	NTFLT	Minimum number of Tug total flights in any Monte Carlo cycle.
134	ITFLT	Total number of Tug flights over all Monte Carlo cycles.
135	MTFLT	Maximum number of total Tug flights in any Monte Carlo cycle.
140	NSEPS	Number of SEPS in fleet.
141	VSEPS	SEPS availability array.
142	ISEPS	Pointer to currently available SEPS.
143	NFSEP	Minimum number of SEPS total flights in any Monte Carlo cycle.
144	IFSEP	Total number of SEPS flights over all Monte Carlo cycles.
145	MFSEP	Maximum number of SEPS flights in any Monte Carlo cycle.
136 - 139	--	Spares

GLOBAL CONSTANTS

ARRAY NO.	NAME	DESCRIPTION
146	AVSEP	The time at which the SEPS last became available for use in orbit (delivery sequence completed).
147	SWDN	The payload weight remaining at the end of the previous SEPS flight. It will be retrieved on the next SEPS flight (not on a Tug).
148	SLDN	The length of the payload described in the previous item.
149	--	Spare

MODULAR TABLE

ARRAY NO.	NAME	DESCRIPTION
150	MITAB	Maximum number of modules
151	MNAME	Name of module.
152	ALPF	Weibul parameter α for failure
153	BETAF	Weibul parameter β for failure.
154	TTFMD	Truncation time for failure.
155	ALPW	Weibul parameter α for warning
156	BETAW	Weibul parameter β for warning
157	TTWMD	Truncation time for warning.
158	MODWT	Module weight in pounds.
159	MDVOL	Module volume in cubic feet.
160	MCLAS	Module classification
161	MDCNT	Count of modules replaced during a Monte Carlo cycle.
162	S121	Sum of MDCNT over run - used to find average.

ARRAY NO.	NAME	DESCRIPTION
163	X121	Maximum value of MDCNT.
164	N121	Minimum value of MDCNT.
165	NOWAR	Count of module warnings during a Monte Carlo cycle.
166	S125	Sum of NOWAR over run - used to find average.
167	X125	Maximum value of NOWAR.
168	N125	Minimum value of NOWAR.
169	NOFAL	Count of module failures during a Monte Carlo cycle.
170	S129	Sum of NOFAL over run - used to find average.
171	X129	Maximum value of NOFAL.
172	N129	Minimum value of NOFAL.
173 - 178	--	Spares

GLOBAL CONSTANTS

ARRAY NO.	NAME	DESCRIPTION
179	CHEM	Flag reserved for chemical scooter feature (not implemented).

SATELLITE TABLE

ARRAY NO.	NAME	DESCRIPTION
180	SITAB	Maximum number of satellites.
181	SNAME	Satellite name.
182	SWT	Satellite weight in pounds.
183	SVOL	Satellite volume in cubic feet.
184	PRIOR	The priority of the satellite (1-6).
185	INCL	The orbit inclination.
186	ORBIT	Orbit apogee-perigee key.
187	TTSAT	Satellite termination time in years.
188	EXWT	Weight of satellite if the satellite is to be used in expendable mode (if not input, the program will use the normal input weight).
189	EXSAT	Flag to indicate expendable satellite.
190	NRSAT	Flag for non-replaceable satellite.
191	NMODS	Number of modules for this satellite.
192	POLDN	Policy indication of action to be taken at the end of satellite life.
193	SORTE	If not zero, this is the number of days duration of a Sortie flight.
194	FMDS	Pointer to first module in module set belonging to this satellite.
195	LMDS	Pointer to the last module in module set belonging to this satellite.
196	--	Spare

SEPS STATUS TABLE

ARRAY NO.	NAME	DESCRIPTION
197	NEXIT	SEPS status flag returned from the SEPS performance routine.
198	LEXIT	Previous value of NEXIT.
199	MSEP	Erase flag used in the SEPS performance routine.

SATELLITE SYSTEM TABLE

ARRAY NO.	NAME	DESCRIPTION
200	STSTB	Maximum number of satellite systems.
201	SYNAM	System name.
202	TTSYS	System termination time.
203	PTTSY	Premature termination time.
204	NSAT	Number of satellites in system.
205	FSAT	Index to first satellite in array SYORB belonging to this system.
206	LSAT	Index to the last satellite in the array SYORB belonging to this system.
207	STAT	The system status.
208	NFUP	Number of active satellites required to have system active.
209	TGOSY	Time at which system is to be terminated.
210	SYLF	Total weight factors or flights in support of this system (used for averaging).
211	XSYLF	Maximum value of SYLF in any one Monte Carlo cycle.
212	NSYLF	Minimum value of SYLF in any one Monte Carlo cycle.
213	BEGSY	Time at which system was initialized.
214	HALSY	Time at which system was terminated.

ARRAY NO.	NAME	DESCRIPTION
215	TLASY	If positive, the time that the system became active. If negative, inactive system.
216	SDTSY	Sum of all the time intervals the system is up.
217	PERSY	Total of percentages of system availability over all Monte Carlo cycles.
218	X200	Maximum percentage of system availability in any one Monte Carlo cycle.
219	N200	Minimum percentage of system availability in any one Monte Carlo cycle.
220	DNTSY	Total of all delay time intervals incurred while system was being restored to service.
221	C208	Count of the number of times the system was delayed in being restored to service.
222	X208	Maximum delay interval incurred while system was being restored to service.
223	N208	Minimum delay interval incurred while system was being restored to service.
224 - 229	--	Spares

SATELLITES IN ORBIT TABLE

ARRAY NO.	NAME	DESCRIPTION
230	SYORB	Maximum number of satellites in orbit
231	ITSAT	Index for this satellite in satellite array
232	ITSYS	Index for the system in the system array to which this satellite belongs.
233	SSTAT	Satellite status.
234	PHASE	Phase angle of this satellite.
235	ATIME	Time satellite arrives in orbit Used to block failures and warnings of earlier satellites

ARRAY NO.	NAME	DESCRIPTION
236	DTIME	Spare variable.
237	MARKS	Pointer to event SATDN for this satellite.
238	MARKU	Pointer to event NWSAT for this satellite (set up by SAVER).
239	MARKD	Pointer to event RETRI for this satellite (set up by SAVER).
240	LFSAT	Sum of weight factors for this satellite over a Monte Carlo cycle.
241	SUMSL	Total of LFSAT over all cycles (used for averaging).
242	MAXSL	Maximum value of LFSAT.
243	MINSL	Minimum value of LFSAT.
244	TGO	Time when satellite is to be terminated.
245	BEGST	Time at which satellite is initialized.
246	HALST	Time at which satellite is terminated.
247	TLAST	If positive, the time that the satellite became active. If negative, inactive satellite.
248	SDTST	Sum of all the time intervals the satellite is up in a Monte Carlo cycle.
249	PERST	Total of percentages of satellite availability over all Monte Carlo cycles.
250	X216	Maximum percentage of satellite availability in any one Monte Carlo cycle.
251	N216	Minimum percentage of satellite availability in any one Monte Carlo cycle.
252	DNTST	Total of all delay time intervals incurred while satellite was being restored to service.
253	C223	Count of the number of times the satellite was delayed in being restored to service.
254	X223	Maximum delay interval incurred while satellite was being restored to service.
255	N223	Minimum delay interval incurred while satellite was being restored to service.

ARRAY NO.	NAME	DESCRIPTION
256	SATLF	Sum of weight factors for deploy-only mode (no service) for this satellite over a Monte Carlo cycle.
257	S227	Total of SATLF over all cycles (used for averaging).
258	X227	Maximum value of SATLF.
259	N227	Minimum value of SATLF.
260	NPOS	Number of satellites currently at each satellite position in orbit.
261	NDEP	Number of satellites deployed to each satellite position in orbit.
262	FMOD	Pointer to first module in module set belonging to this satellite in orbit.
263	LMOD	Pointer to last module in module set belonging to this satellite in orbit.
264	XSAT	A flag on a satellite to show it was launched as expendable. (This flag is important in a run in which the model shifts from expendable to reusable.)
265 - 269	--	Spares

VEHICLE AVAILABILITY STATISTICS

ARRAY NO.	NAME	DESCRIPTION
270	NVS	Number of vehicle types.
271	CVA	Number of times in a Monte Carlo cycle a vehicle type became unavailable.
272	TCVA	Total of CVA over all cycles (for averaging)
273	XCVA	Maximum value of CVA.
274	MCVA	Minimum value of CVA.
275	VDATE	Unavailability interval for vehicle type: 0, vehicle type available -, negative sum of dates vehicle type became unavailable. +, interval of unavailability
276	VTD	Total of VDATE over all occurrences
277	XTD	Maximum value of VDATE.
278	MTD	Minimum value of VDATE.
279	--	Spare

UPPERMOST STAGE STATISTICS

ARRAY NO.	NAME	DESCRIPTION
280	CTUG	Total number of Tugs used as uppermost stage on each orbit.
281	WTUG	Total weight of payload delivered on Tugs, etc
282	CSHUT	Total number of Shuttles, etc.
283	WSHUT	Total weight of Shuttles, etc.
284	CSEPS	Total number of SEPS, etc.
285	WSEPS	Total weight payload SEPS, etc

ORBIT TABLE (Continued)

ARRAY NO.	NAME	DESCRIPTION
286	NPAD1	Pointer to VPAD list for first launch pad usable to this orbit.
287	NPAD2	Pointer to VPAD list for last launch pad usable to this orbit (This allows simulation of pads assigned to ETR and WTR).

DOWN FLIGHT STATISTICS

ARRAY NO.	NAME	DESCRIPTION
288	CDTUG	Total number of down flights by the Tug.
289	WDTUG	Total down weight carried by the Tug.
290	CDSUT	Total number of down flights by the Shuttle.
291	WDSUT	Total down weight carried by the Shuttle.

ARRAY NO.	NAME	DESCRIPTION
292	CDSEP	Total number of down flights by the SEPS.
293	WDSEP	Total down weight carried by the SEPS.

GLOBAL CONSTANTS

ARRAY NO.	NAME	DESCRIPTION
297	FREE	Flag used to request the word FREEBIE be printed for a module.
298	SCOOT	Flag used to restrict the SEPS to a scooter mode.
299	IEVQW	Event counter for event QWAIT.
300	IEVST	Event counter for event START.
301	IEVTE	Event counter for event TERM.
302	IEVNW	Event counter for event NWSAT.
303	IEVWA	Event counter for event WARN.
304	IEVFA	Event counter for event FAIL.
305	IEVLA	Event counter for event LAUNC.
306	IEVAR	Event counter for event ARRIV.
307	IEVVE	Event counter for event REFVE.
308	IEVMO	Event counter for event REFMO.
309	IEVBE	Event counter for event BACK.
310	IEVSA	Event counter for event REFSA.
311	IEVOV	Event counter for event REMOV.
312	IEVRI	Event counter for event RETRI.
313	IEVDN	Event counter for event SATDN.
314	IEVME	Event counter for event NEWME.
315	NSUUP	Number of SUs deployed.
316	NSUDN	Number of SUs retrieved.
317	MODOF	Spare.

ARRAY NO.	NAME	DESCRIPTION
318	SATOF	Spare.
319	MIXED	Spare.
320	FWAIT	Spare.
321	FWGT	Spare.

APPENDIX C: SAMPLE PRINTOUT

This appendix contains a small sample run which can be used to verify the basic operations of the program. There are 20 pages of computer output for a very small data case. This sample can be executed in 3 seconds on the CDC 7600 computer at The Aerospace Corporation, El Segundo, California, and in 12 seconds on the UNIVAC 1108 computer at the NASA Computer Facility, Slidell, Louisiana.

The appendix consists of the following parts:

a.	Pages 2-4	Listing of the entire input data deck.
b.	Pages 5-6	SIMSCRIPT print of the initialization data deck.
c.	Pages 7-8	Program print of the input data cards.
d.	Page 9	Summary of unused input data.
e.	Pages 10-12	Chronological time history of the first Monte Carlo cycle of the run.
f.	Pages 13-14	Chronological time history of each satellite position. This is a rearrangement of most of the information on pages 10-12.
g.	Pages 15-21	Statistical summaries of the 25 Monte Carlo cycles in this run. The summaries are for vehicles, orbits, systems, satellites, and modules.

```
*DECK FRONT
    1     7 0 R                                          7(A6)
OK     OUT    NG      SHUT   TUG    SEPS
     8        0 R                                        32.17
     9   11   0  Z
    12        0 R                                         1       PDOWN(NRU)
    13        0 R                                        16       NO MOD IN SU
    14        0 R                                       400.      WT SU
    15        0 R                                        8.       LNTH SU
    16   17   0  Z
    18        0 R                                       -90.      WAIT 1
    19        0 R                                         0.      WAIT 2
    20        0 R                                        90.      WAIT SAT IF UP
    21        0 R                                        90.      WAIT SAT IF ON
    22        0 R                                        90.      WAIT MOD IF UP
    23        0 R                                        90.      WAIT MOD IF ON
    24   25   0  Z
    26        0 R                                        25       NO MONTE CARLO
    27        0  Z
    28        0 R                                        60.      SEPS REFURB T
    29        0 R                                        15.      TUG  REFURB T
    30        0 R                                        15.      SHTL REFURB T
    31        0 R                                         0.      PL VEH MATE T
    32   41   0  Z
    42        0 R                                        3443.9336
    43   46   0  Z
    47        0 R                                        0.250    WAIT 3
    48   59   0  Z
    60        0 R                                        43       NO ORBITS
    61   78   1  Z    43   60
    79        0  Z
    80        0 R                                        20       NO PL ON VEH
    81   63   1  Z    20   80
    84        0  Z
    85        0 R                                        20       NO OF DIFF VEH
    86   97   1  Z    20   85
    98   99   0  Z
   100        0 R                                        30       NO YRS IN RUN
   101  112   1  Z    30  100
   113  119   0  Z
   120        0 R                                        20       NO OF SHUTLS
   121        1  Z    20  120
   122  129   0  Z
   130        0 R                                        10       NO OF TUGS
   131        1  Z    10  130
   132  139   0  Z
   140        0 R                                        10       NO OF SEPS
   141        1  Z    10  140
   142  149   0  Z
   150        0 R                                        19       NO OF MODS
   151  172   1  Z    19  150
   173  179   0  Z
   180        0 R                                         5       NO OF SATS
   181  195   1  Z     5  180
   196  199   0  Z
   200        0 R                                         4       NO OF SYSTMS
```

```
 201  223 1   Z       4 200
 224  229 0   Z
 230      0  R
 231  263 1   Z      15 230                                                    15          NO SAT-POS
 264  269 0   Z
 270      0  R
 271  278 1   Z       3 270                                                    3           NO TYPES VEH
 279      0   Z
 280  285 1   Z     43   60
                                                   (ONE BLANK CARD)
   1               1979.00.00 1984.00.00
*DECK  LS
   3
TUG          7.0   464.0 5223.0   950.063538.0   14.0      0.0     60.0
SHTL                                  65000.0                      60.0
SHTL1                                 63538.0                      60.0
   3
SNC/0 13882.0      24.022763.010100.0TUG              SHTL1
1-1/28     0.0      0.0      0.0      0.0             SHTL1
SORT                                                  SHTL
  19             NUMBER OF MODULES
AVCS2        14.80 1.0   10.     94.
TTC5          5.10 1.02   5.    119.
DP1          18.93 1.00  10.    103.
EPS1A        21.51 0.99   4.    110.
EPS2         21.43 0.99   5.    287.
EPS3         99.99 1.00   5.    112.
AST1C1       13.48 1.0    5.    376.
AVCS3        14.80 1.0   10.    241.
AVCS5         6.42 1.15  10.    123.
AVCS7        14.35 1.02   7.    112.
GN1           6.17 1.00  10.    173.
GN2           7.13 1.00  10.     94.
TTC1         14.67 1.10  10.    112.
NN011        26.95 1.0   10.    377.
NN012        18.36 1.0   10.    548.
NAST1C       99.99 1.0   10.    653.
NNND1        28.01 1.0   10.   1828.
MODULE        0.0  0.0  999.       .
MOD1               1.0                                                    0.250 5.0
  5         NUMBER OF SATELLITES
AST1A        1373.13.        0.     0.1-1/28                                               1   20.
2
             MOD1       X
AST10        2491. 3.        0.     0.SNC/0                                               13   20.
2
             NAST1C     XAVCS3      AVCS5      AVCS7      AVCS7      AVCS7      AVCS7
             GN1        GN2         TTC5       DP1        EPS1A      AST1C1
NN01A        5772. 8.        0.     0.SNC/0                                               20   20.
2
             NNND1      XAVCS2      AVCS5      AVCS7      AVCS7      AVCS7      AVCS7
             GN2        TTC1        DP1        EPS2       EPS2       EPS2       EPS2
             EPS2       EPS2        EPS3       EPS3       NN011      NN012
NN01B        5772. 8.        0.     0.SNC/0                                               20   20.
2
             NNND1      XAVCS2      AVCS5      AVCS7      AVCS7      AVCS7      AVCS7
             GN2        TTC1        DP1        EPS2       EPS2       EPS2       EPS2
```

```
         EPS2
SORT7    10000.60.0     EPS2       EPS3       EPS3       NND11      NND12
D   7.                             SORT                                         1
         MODULE    X
  4    NUMBER OF SYSTEMS
AST1A    1      120.0AST1A
AST1D    2      2 7.0AST1D        -80.       AST1D      -80.
NND1AB   4      420.0NND1A        -40.       NND1B      -180.      NND1A      -40.0
                     NND1B        -180
SORT7    1      112.0SORT7
1AST1D   1983.09    2AST1D        1983.09
1NND1AB  1980.33    2NND1AB       1980.33    3NND1AB    1980.33    4NND1AB    1980.49
1SORT7   1982.09
```

(ONE BLANK CARD)
(ONE BLANK CARD)

```
INITIALIZATION CARDS
          1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
EQUIP,5=COMPILE
EQUIF,6=TAPE6
1L   L   285     30    12          5     5     6    55
    1    7 0 R                                       7(A6)
OK     OUT    NG     SHUT  TUG   SEPS
    8      0 R                                       32.17
    9   11 0  Z
   12      0 R                                        1        PDOWN(NRU)
   13      0 R                                       16        NO MOD IN SU
   14      0 R                                      400.       WT SU
   15      0 R                                        8.       LNTH SU
   16   17 0  Z
   18      0 R                                      -90.       WAIT 1
   19      0 R                                        0.       WAIT 2
   20      0 R                                       90.       WAIT SAT IF UP
   21      0 R                                       90.       WAIT SAT IF DN
   22      0 R                                       90.       WAIT MOD IF UP
   23      0 R                                       90.       WAIT MOD IF DN
   24   25 0  Z
   26      0 R                                       25        NO MONTE CARLO
   27      0  Z
   28      0 R                                       60.       SEPS REFURB T
   29      0 R                                       15.       TUG  REFURB T
   30      0 R                                       15.       SHTL REFURB T
   31      0 R                                        0.       PL VEH MATE T
   32   41 0  Z
   42      0 R                                       3443.9336
   43   46 0  Z
   47      0 R                                        0.250    WAIT 3
   48   59 0  Z
   60      0 R                                       43        NO ORBITS
   61   78 1  Z   43   60
   79      0  Z
   80      0 R                                       20        NO PL ON VEH
   81   83 1  Z   20   80
   84      0  Z
   85      0 R                                        20       NO OF DIFF VEH
   86   97 1  Z   20   85
   98   99 0  Z
  100      0 R                                        30       NO YRS IN RUN
  101  112 1  Z   30  100
  113  119 0  Z
  120      0 R                                        20       NO OF SHUTLS
  121      1  Z   20  120
  122  129 0  Z
  130      0 R                                        10       NO OF TUGS
  131      1  Z   10  130
  132  139 0  Z
  140      0 R                                        10       NO OF SEPS
  141      1  Z   10  140
  142  149 0  Z
  150      0 R                                        19       NO OF MODS
```

```
INITIALIZATION CARDS
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890
 151 172 1  Z   19 150
 173 179 0  Z
 180     0 R                                            5         NO OF SATS
 181 195 1  Z    5 180
 196 199 0  Z
 200     0 R                                            4         NO OF SYSTMS
 201 223 1  Z    4 200
 224 229 0  Z
 230     0 R                                            15        NO SAT-POS
 231 263 1  Z   15 230
 264 269 0  Z
 270     0 R                                            3         NO TYPES VEH
 271 278 1  Z    3 270
 279     0  Z
 280 285 1  Z   43  60

INITIALIZATION TERMINATED WITHOUT ERROR.EXECUTION CONTINUED
```

INPUT DATA

3 VEHICLES INPUT

NAME	DAYS	ISP	WDV	WPNU	WCONV	REFT	EXP	LENGTH	NSATGE	SOLID	ID	
TUG	7.0		464.0	5223.0	950.0	63538.0		14.0	0.0	60.0	0	0
SHTL	0.0		0.0	0.0	0.0	65000.C		0.0	0.0	60.0	0	0
SHTL1	0.0		0.0	0.0	0.0	63538.0		0.0	0.0	60.0	0	0

3 ORBITS INPUT

NAME	DV	PERIOD	RA	VC	UPPER	SEPS	SHUTTLE	
SNG/0	13882.0	24.0	22763.0		10100.0	TUG	SHTL1	0.0
1-1/28	0.0	0.0	0.0		0.0		SHTL1	0.0
SORT	0.0	0.P	0.0		0.0		SHTL	0.0

19 MODULES INPUT

NAME	ALPHA E	BETA E	TT	ALPHA W	BETA W	WT	VOL	CLASS
AVCS2	14.80	1.00	10.00	0.00	1.00	94.00	0.00	
TTC5	5.10	1.02	5.00	0.00	1.00	119.00	0.00	
OP1	18.93	1.00	10.00	0.00	1.00	103.00	0.00	
EPS1A	21.51	0.99	4.00	0.00	1.00	110.00	0.00	
EPS2	21.43	0.99	5.00	0.00	1.00	287.00	0.00	
EPS3	99.99	1.00	5.00	0.00	1.00	112.00	0.00	
AST1C1	13.48	1.00	5.00	0.00	1.00	376.00	0.00	
AVCS3	14.80	1.00	10.00	0.00	1.00	241.00	0.00	
AVCS5	6.42	1.15	10.00	0.00	1.00	123.00	0.00	
AVCS7	14.35	1.02	7.00	0.00	1.00	112.00	0.00	
GN1	6.17	1.00	10.00	0.00	1.00	173.00	0.00	
GN2	7.13	1.00	10.00	0.00	1.00	94.00	0.00	
TTC1	14.67	1.10	10.00	0.00	1.00	112.00	0.00	
NND11	26.95	1.00	10.00	0.00	1.00	377.00	0.00	
NND12	18.36	1.00	10.00	0.00	1.00	548.00	0.00	
NAST1C	99.99	1.00	10.00	0.00	1.00	653.00	0.00	
NNND1	28.01	1.00	10.00	0.00	1.00	1828.00	0.00	
MODULE	0.00	0.00	999.00	0.00	1.00	0.00	0.00	
MOD1	0.00	1.00	0.00	0.00	1.00	0.00	0.00	

5 SATELLITES INPUT

NAME	WT	VOL	PRIOR	INCLINATICN	ORBIT	MODULES		SAT TT	POLICY	
AST1A	1373.	13.	0.	0.	1-1/28	1		20.	2	0.
	MOD1	X								
AST1D	2491.	3.	0.	0.	SNC/0	13		20.	2	0.
	NAST1C	X	AVCS3	AVCS5	AVCS7		AVCS7	AVCS7		AVCS7
	GN1		GN2	TTC5	OP1		EPS1A	AST1C1		
NND1A	5772.	8.	0.	0.	SNC/0	20		20.	2	0.
	NNND1	X	AVCS2	AVCS5	AVCS7		AVCS7	AVCS7		AVCS7
	GN2		TTC1	OP1	EPS2		EPS2	EPS2		EPS2
	EPS2		EPS2	EPS3	EPS3		NND11	NND12		
NND1B	5772.	8.	0.	0.	SNC/0	20		20.	2	0.
	NNND1	X	AVCS2	AVCS5	AVCS7		AVCS7	AVCS7		AVCS7
	GN2		TTC1	OP1	EPS2		EPS2	EPS2		EPS2
	EPS2		EPS2	EPS3	EPS3		NND11	NND12		
SORT7	10000.	60.	0.	0.	SORT	1		10.	0	7.
	MODULE	X								

4SYSTEMS INPUT

NAME	NUP	NTOT	SYS TT	SAT	PHASE	SAT	PHASE	SAT	PHASE
AST1A	1	1	20.00	AST1A	0.0		0.0		0.0
AST1D	2	2	7.00	AST10	-80.0	AST10	-80.0		0.0

```
NNO1AB      4     4     20.00     NNO1A     -40.0     NNO1B     -180.0     NNO1A     -40.0
                                  NNO1B    -180.0                  0.0                 0.0
SORT7       1     1     12.00     SORT7       0.0                  0.0                 0.0
    SCHEDULES INPUT
    1   AST1D        1983.09000 2  AST1D     1983.09000 0                0.00000 0                0.00000
    1   NNO1AB       1980.33000 2  NNO1AB    1980.33000 3  NNO1AB    1980.33000 4  NNO1AB    1980.49000
    1   SORT7        1982.09000 0               0.00000 0                0.00000 0                0.00000
    0                   0.00000 0               0.00000 0                0.00000 0                0.00000
ME UPGRADE SCHEDULES INPUT
                      0                 0       0. 0. 0
```

```
     SYNOPSIS OF INPUT
UNUSED SYSTEM - AST1A
PROBLEM USED   8 SATELLITE/SYSTEM POSITIONS OUT OF AVAILABLE  15
PROBLEM USED   3 SYSTEMS OUT OF AVAILABLE   4
  UNUSED SATELLITE - AST1A
PROBLEM USED   4 SATELLITES OUT OF AVAILABLE   5
  UNUSED MODULE - MOD1
PROBLEM USED  18 MODULES OUT OF AVAILABLE  19
```

```
                                  CHRONOLOGICAL TIME HISTORY OF BASE CYCLE
TIME                SYSTEM        STATUS     SATELLITE STATUS    MODULE      STATUS      VEHICLE    STATUS
      1.  0.  0     START SIMULATION

 1980.  3.28       NND1AB   3 OUT           NND1A      AVAILABLE
 1980.  3.28       NND1AB   2 OUT           NND1B      AVAILABLE
--LAUNCH NOW -- SHUTTLE NO.   20 TUG NO.   10 SEPS NO.   0 WEIGHT =    5772. LENGTH =          8.0-----
 1980.  3.28       NND1AB   3 OUT           NND1A      LAUNCHED
------------------------------------------------------------------------------------------------------
 1980.  3.28       NND1AB   1 OUT           NND1A      AVAILABLE
--LAUNCH NOW -- SHUTTLE NO.   19 TUG NO.    9 SEPS NO.   0 WEIGHT =    5772. LENGTH =          8.0-----
 1980.  3.28       NND1AB   2 OUT           NND1B      LAUNCHED
------------------------------------------------------------------------------------------------------
 1980.  3.28       NND1AB   3 OUT           NND1A      OK
 1980.  3.28       NND1AB   2 OUT           NND1B      OK
 1980.  4.14                                                                      SHUT    20 REFURBISHED
 1980.  4.14                                                                      TUG     10 REFURBISHED
 1980.  4.14                                                                      SHUT    19 REFURBISHED
 1980.  4.14                                                                      TUG      9 REFURBISHED
 1980.  5.26       NND1AB   4 OUT           NND1B      AVAILABLE
--LAUNCH NOW -- SHUTTLE NO.   20 TUG NO.   10 SEPS NO.   0 WEIGHT =    5772. LENGTH =          8.0-----
 1980.  5.26       NND1AB   1 OUT           NND1A      LAUNCHED
------------------------------------------------------------------------------------------------------
 1980.  5.26       NND1AB   1 OUT           NND1A      OK
 1980.  6.11                                                                      SHUT    20 REFURBISHED
 1980.  6.11                                                                      TUG     10 REFURBISHED
 1980.  7.16       NND1AB   1 OUT           NND1A      OUT         TTC1      OUT
 1980.  7.22       NND1AB   3 OUT           NND1A      OUT         TTC1      OUT
--LAUNCH NOW -- SHUTTLE NO.   20 TUG NO.   10 SEPS NO.   0 WEIGHT =    6396. LENGTH =         16.0-----
 1980.  8.26       NND1AB   4 OUT           NND1B      LAUNCHED
 1980.  8.26       NND1AB   1 OUT           NND1A      OUT         TTC1      LAUNCHED
 1980.  8.26       NND1AB   3 OUT           NND1A      OUT         TTC1      LAUNCHED
------------------------------------------------------------------------------------------------------
 1980.  8.26       NND1AB   4 OUT           NND1B      OK
 1980.  8.26       NND1AB   1 OUT           NND1A      OK          TTC1      OK
 1980.  8.26       NND1AB   3 OK            NND1A      OK          TTC1      OK
 1980.  9.11                                                                      SHUT    20 REFURBISHED
 1980.  9.11                                                                      TUG     10 REFURBISHED
 1981.  0.22       NND1AB   4 OUT           NND1B      OUT         NND12     OUT
 1981.  2.  7      NND1AB   3 OUT           NND1A      OUT         AVCS5     OUT
 1981.  3.  1      NND1AB   2 OUT           NND1B      OUT         AVCS7     OUT
--LAUNCH NOW -- SHUTTLE NO.   20 TUG NO.   10 SEPS NO.   0 WEIGHT =    1183. LENGTH =          8.0------
 1981.  3.22       NND1AB   4 OUT           NND1B      OUT         NND12     LAUNCHED
 1981.  3.22       NND1AB   2 OUT           NND1B      OUT         AVCS7     LAUNCHED
 1981.  3.22       NND1AB   3 OUT           NND1A      OUT         AVCS5     LAUNCHED
------------------------------------------------------------------------------------------------------
 1981.  3.22       NND1AB   4 OUT           NND1B      OK          NND12     OK
 1981.  3.22       NND1AB   2 OUT           NND1B      OK          AVCS7     OK
 1981.  3.22       NND1AB   3 OK            NND1A      OK          AVCS5     OK
 1981.  4.  7                                                                     SHUT    20 REFURBISHED
 1981.  4.  7                                                                     TUG     10 REFURBISHED
 1981.  5.  0      NND1AB   2 OUT           NND1B      OUT         AVCS7     OUT
 1981.  5.  1      NND1AB   4 OUT           NND1B      OUT         AVCS7     OUT
 1981.  5.  9      NND1AB   4 OUT           NND1B      OUT         OP1       OUT
 1981.  7.  5      NND1AB   4 OUT           NND1B      OUT         AVCS5     OUT
--LAUNCH NOW -- SHUTTLE NO.   20 TUG NO.   10 SEPS NO.   0 WEIGHT =     850. LENGTH =          8.0------
```

ORIGINAL PAGE IS OF POOR QUALITY

```
 1981. 8. 0    NND1AB   2 OUT           NND1B      OUT         AVCS7     LAUNCHED
 1981. 8. 0    NND1AB   4 OUT           NND1B      OUT         AVCS7     LAUNCHED
 1981. 8. 0    NND1AB   4 OUT           NND1B      OUT         DP1       LAUNCHED
 1981. 8. 0    NND1AB   4 OUT           NND1B      OUT         AVCS5     LAUNCHED
--------------------------------------------------------------------------------------------------
 1981. 8. 0    NND1AB   2 OUT           NND1B      OK          AVCS7     OK
 1981. 8. 0    NND1AB   4 OUT           NND1B      OUT         AVCS7     OK
 1981. 8. 0    NND1AB   4 OUT           NND1B      OUT         DP1       OK
 1981. 8. 0    NND1AB   4 OK            NND1B      OK          AVCS5     OK
 1981. 8.15                                                                       SHUT    20 REFURBISHED
 1981. 8.15                                                                       TUG     10 REFURBISHED
 1981.11.20    NND1AB   1 OUT           NND1A      OUT         NND12     OUT
 1982. 1. 2    SORT7    1 OUT           SORT7      AVAILABLE
--LAUNCH NOW -- SHUTTLE NO.   20 TUG NO.     0 SEPS NO.     0 WEIGHT =   10000. LENGTH =        60.0-------
 1982. 1. 2    SORT7    1 OUT           SORT7      LAUNCHED
--------------------------------------------------------------------------------------------------
 1982. 1. 2    SORT7    1 OK            SORT7      OK
 1982. 1. 9    SORT7    1 NG            SORT7      REMOVED
 1982. 1.24                                                                       SHUT    20 REFURBISHED
 1982. 2.16    NND1AB   4 OUT           NND1B      OUT         EPS2      OUT
--LAUNCH NOW -- SHUTTLE NO.   20 TUG NO.    10 SEPS NO.     0 WEIGHT =    1235. LENGTH =         8.0-------
 1982. 2.20    NND1AB   4 OUT           NND1B      OUT         EPS2      LAUNCHED
 1982. 2.20    NND1AB   1 OUT           NND1A      OUT         NND12     LAUNCHED
--------------------------------------------------------------------------------------------------
 1982. 2.20    NND1AB   4 OUT           NND1B      OK          EPS2      OK
 1982. 2.20    NND1AB   1 OK            NND1A      OK          NND12     OK
 1982. 3. 5                                                                       SHUT    20 REFURBISHED
 1982. 3. 5                                                                       TUG     10 REFURBISHED
 1982. 3.17    NND1AB   3 OUT           NND1A      OUT         AVCS7     OUT
 1982. 6.13    NND1AB   2 OUT           NND1B      OUT         EPS2      OUT
 1982. 6.13    NND1AB   2 OUT           NND1B      OUT         EPS2      OUT
--LAUNCH NOW -- SHUTTLE NO.   20 TUG NO.    10 SEPS NO.     0 WEIGHT =    1086. LENGTH =         8.0-------
 1982. 6.17    NND1AB   2 OUT           NND1B      OUT         EPS2      LAUNCHED
 1982. 6.17    NND1AB   2 OUT           NND1B      OUT         EPS2      LAUNCHED
 1982. 6.17    NND1AB   3 OUT           NND1A      OUT         AVCS7     LAUNCHED
--------------------------------------------------------------------------------------------------
 1982. 6.17    NND1AB   2 OUT           NND1B      OUT         EPS2      OK
 1982. 6.17    NND1AB   2 OUT           NND1B      OK          EPS2      OK
 1982. 6.17    NND1AB   3 OK            NND1A      OK          AVCS7     OK
 1982. 7. 3                                                                       SHUT    20 REFURBISHED
 1982. 7. 3                                                                       TUG     10 REFURBISHED
 1982. 9. 6    NND1AB   4 OUT           NND1B      OUT         AVCS2     OUT
--LAUNCH NOW -- SHUTTLE NO.   20 TUG NO.    10 SEPS NO.     0 WEIGHT =     494. LENGTH =         8.0-------
 1983. 0. 6    NND1AB   4 OUT           NND1B      OUT         AVCS2     LAUNCHED
--------------------------------------------------------------------------------------------------
 1983. 0. 6    NND1AB   4 OK            NND1B      OK          AVCS2     OK
 1983. 0.21                                                                       SHUT    20 REFURBISHED
 1983. 0.21                                                                       TUG     10 REFURBISHED
 1983. 1. 2    AST1D    2 OUT           AST1D      AVAILABLE
 1983. 1. 2    AST1D    1 OUT           AST1D      AVAILABLE
 1983. 2.27    NND1AB   2 OUT           NND1B      OUT         AVCS2     OUT
--LAUNCH NOW -- SHUTTLE NO.   20 TUG NO.    10 SEPS NO.     0 WEIGHT =    4982. LENGTH =         6.0-------
 1983. 2.27    AST1D    2 OUT           AST1D      LAUNCHED
 1983. 2.27    AST1D    1 OUT           AST1D      LAUNCHED
--------------------------------------------------------------------------------------------------
 1983. 2.27    AST1D    2 OUT           AST1D      OK
 1983. 2.27    AST1D    1 OK            AST1D      OK
```

```
1983. 3.12
1983. 3.12                                                                       SHUT   20 REFURBISHED
1983. 3.22  NND1AB   4 OUT        NND1B      OUT      EPS2       OUT             TUG    10 REFURBISHED
1983. 5. 6  NND1AB   3 OUT        NND1A      OUT      AVCS7      OUT
--LAUNCH NOW -- SHUTTLE NO.   20 TUG NO.   10 SEPS NO.    0 WEIGHT =      781.  LENGTH =        8.0-------
1983. 5. 6  NND1AB   2 OUT        NND1B      OUT      AVCS2      LAUNCHED
1983. 5. 6  NND1AB   4 OUT        NND1B      OUT      EPS2       LAUNCHED
---------------------------------------------------------------------------------------------------------
1983. 5. 6  NND1AB   2 OUT        NND1B      OK       AVCS2      OK
1983. 5. 6  NND1AB   4 OUT        NND1B      OK       EPS2       OK
1983. 5.21                                                                       SHUT   20 REFURBISHED
1983. 5.21                                                                       TUG    10 REFURBISHED
1983. 6. 9  NND1AB   2 OUT        NND1B      OUT      EPS2       OUT
--LAUNCH NOW -- SHUTTLE NO.   20 TUG NO.   10 SEPS NO.    0 WEIGHT =      799.  LENGTH =        8.0-------
1983. 8. 6  NND1AB   2 OUT        NND1B      OUT      EPS2       LAUNCHED
1983. 8. 6  NND1AB   3 OUT        NND1A      OUT      AVCS7      LAUNCHED
---------------------------------------------------------------------------------------------------------
1983. 8. 6  NND1AB   2 OUT        NND1B      OK       EPS2       OK
1983. 8. 6  NND1AB   3 OK         NND1A      OK       AVCS7      OK
1983. 8.19  AST1D    1 OUT        AST1D      OUT      TTC5       OUT
1983. 8.21                                                                       SHUT   20 REFURBISHED
1983. 8.21                                                                       TUG    10 REFURBISHED
1983. 9.27  NND1AB   4 OUT        NND1B      OUT      AVCS7      OUT
1983.10. 7  NND1AB   3 OUT        NND1A      OUT      AVCS5      OUT
--LAUNCH NOW -- SHUTTLE NO.   20 TUG NO.   10 SEPS NO.    0 WEIGHT =      519.  LENGTH =        8.0--------
1983.11.19  AST1D    1 OUT        AST1D      OUT      TTC5       LAUNCHED
---------------------------------------------------------------------------------------------------------
1983.11.19  AST1D    1 OK         AST1D      OK       TTC5       OK
1984. 0. 0  NND1AB   3 OUT        NND1A      NG
1984. 0. 0  NND1AB   2 OUT        NND1B      NG
1984. 0. 0  NND1AB   1 OUT        NND1A      NG
1984. 0. 0  NND1AB   4 NG         NND1B      NG
1984. 0. 0  AST1D    2 OUT        AST1D      NG
1984. 0. 0  AST1D    1 NG         AST1D      NG
1984. 0. 4                                                                       SHUT   20 REFURBISHED
1984. 0. 4                                                                       TUG    10 REFURBISHED

3000. 0. 0  TERMINATF SIMULATION
```

CHRONOLOGICAL TIME HISTORY OF SATELLITE POSITION IN ORBIT

TIME	SYSTEM	STATUS	SATELLITE	STATUS	MODULE	STATUS
1983. 1. 2	AST1D	1 OUT	AST1D	AVAILABLE		
1983. 2.27	AST1D	1 OUT	AST1D	LAUNCHED		
1983. 2.27	AST1D	1 OK	AST1D	OK		
1983. 8.19	AST1D	1 OUT	AST1D	OUT	TTC5	OUT
1983.11.19	AST1D	1 OUT	AST1D	OUT	TTC5	LAUNCHED
1983.11.19	AST1D	1 OK	AST1D	OK	TTC5	OK
1984. 0. 0	AST1D	1 NG	AST1D	NG		

CHRONOLOGICAL TIME HISTORY OF SATELLITE POSITION IN ORBIT

TIME	SYSTEM	STATUS	SATELLITE	STATUS	MODULE	STATUS
1983. 1. 2	AST1D	2 OUT	AST1D	AVAILABLE		
1983. 2.27	AST1D	2 OUT	AST1D	LAUNCHED		
1983. 2.27	AST1D	2 OUT	AST1D	OK		
1984. 0. 0	AST1D	2 OUT	AST1D	NG		

CHRONOLOGICAL TIME HISTORY OF SATELLITE POSITION IN ORBIT

TIME	SYSTEM	STATUS	SATELLITE	STATUS	MODULE	STATUS
1980. 3.28	NND1AB	1 OUT	NND1A	AVAILABLE		
1980. 5.26	NND1AB	1 OUT	NND1A	LAUNCHED		
1980. 5.26	NND1AB	1 OUT	NND1A	OK		
1980. 7.16	NND1AB	1 OUT	NND1A	OUT	TTC1	OUT
1980. 8.26	NND1AB	1 OUT	NND1A	OUT	TTC1	LAUNCHED
1980. 8.26	NND1AB	1 OUT	NND1A	OK	TTC1	OK
1981.11.20	NND1AB	1 OUT	NND1A	OUT	NND12	OUT
1982. 2.20	NND1AB	1 OUT	NND1A	OUT	NND12	LAUNCHED
1982. 2.20	NND1AB	1 OK	NND1A	OK	NND12	OK
1984. 0. 0	NND1AB	1 OUT	NND1A	NG		

CHRONOLOGICAL TIME HISTORY OF SATELLITE POSITION IN ORBIT

TIME	SYSTEM	STATUS	SATELLITE	STATUS	MODULE	STATUS
1980. 3.28	NND1AB	2 OUT	NND1B	AVAILABLE		
1980. 3.28	NND1AB	2 OUT	NND1B	LAUNCHED		
1980. 3.28	NND1AB	2 OUT	NND1B	OK		
1981. 3. 1	NND1AB	2 OUT	NND1B	OUT	AVCS7	OUT
1981. 3.22	NND1AB	2 OUT	NND1B	OUT	AVCS7	LAUNCHED
1981. 3.22	NND1AB	2 OUT	NND1B	OK	AVCS7	OK
1981. 5. 8	NND1AB	2 OUT	NND1B	OUT	AVCS7	OUT
1981. 8. 8	NND1AB	2 OUT	NND1B	OUT	AVCS7	LAUNCHED
1981. 8. 8	NND1AB	2 OUT	NND1B	OK	AVCS7	OK
1982. 6.13	NND1AB	2 OUT	NND1B	OUT	EPS2	OUT
1982. 6.13	NND1AB	2 OUT	NND1B	OUT	EPS2	OUT
1982. 6.17	NND1AB	2 OUT	NND1B	OUT	EPS2	LAUNCHED
1982. 6.17	NND1AB	2 OUT	NND1B	OUT	EPS2	LAUNCHED
1982. 6.17	NND1AB	2 OUT	NND1B	OUT	EPS2	OK
1982. 6.17	NND1AB	2 OUT	NND1B	OK	EPS2	OK
1983. 2.27	NND1AB	2 OUT	NND1B	OUT	AVCS2	OUT
1983. 5. 6	NND1AB	2 OUT	NND1B	OUT	AVCS2	LAUNCHED
1983. 5. 6	NND1AB	2 OUT	NND1B	OK	AVCS2	OK
1983. 6. 9	NND1AB	2 OUT	NND1B	OUT	EPS2	OUT
1983. 8. 6	NND1AB	2 OUT	NND1B	OUT	EPS2	LAUNCHED
1983. 8. 6	NND1AB	2 OUT	NND1B	OK	EPS2	OK
1984. 0. 0	NND1AB	2 OUT	NND1B	NG		

CHRONOLOGICAL TIME HISTORY OF SATELLITE POSITION IN ORBIT

TIME	SYSTEM	STATUS	SATELLITE	STATUS	MODULE	STATUS
1980. 3.28	NND1AB	3 OUT	NND1A	AVAILABLE		
1980. 3.28	NND1AB	3 OUT	NND1A	LAUNCHED		
1980. 3.28	NND1AB	3 OUT	NND1A	OK		
1980. 7.22	NND1AB	3 OUT	NND1A	OUT	TTC1	OUT
1980. 8.26	NND1AB	3 OUT	NND1A	OUT	TTC1	LAUNCHED
1980. 8.26	NND1AB	3 OK	NND1A	OK	TTC1	OK
1981. 2. 7	NND1AB	3 OUT	NND1A	OUT	AVCS5	OUT
1981. 3.22	NND1AB	3 OUT	NND1A	OUT	AVCS5	LAUNCHED
1981. 3.22	NND1AB	3 OK	NND1A	OK	AVCS5	OK
1982. 3.17	NND1AB	3 OUT	NND1A	OUT	AVCS7	OUT
1982. 6.17	NND1AB	3 OUT	NND1A	OUT	AVCS7	LAUNCHED
1982. 6.17	NND1AB	3 OK	NND1A	OK	AVCS7	OK
1983. 5. 6	NND1AB	3 OUT	NND1A	OUT	AVCS7	OUT
1983. 8. 6	NND1AB	3 OUT	NND1A	OUT	AVCS7	LAUNCHED
1983. 8. 6	NND1AB	3 OK	NND1A	OK	AVCS7	OK
1983.10. 7	NND1AB	3 OUT	NND1A	OUT	AVCS5	OUT
1984. 0. 0	NND1AB	3 OUT	NND1A	NG		

CHRONOLOGICAL TIME HISTORY OF SATELLITE POSITION IN ORBIT

TIME	SYSTEM	STATUS	SATELLITE	STATUS	MODULE	STATUS
1980. 5.26	NND1AB	4 OUT	NND1B	AVAILABLE		
1980. 8.26	NND1AB	4 OUT	NND1B	LAUNCHED		
1980. 8.26	NND1AB	4 OUT	NND1B	OK		
1981. 0.22	NND1AB	4 OUT	NND1B	OUT	NND12	OUT
1981. 3.22	NND1AB	4 OUT	NND1B	OUT	NND12	LAUNCHED
1981. 3.22	NND1AB	4 OUT	NND1B	OK	NND12	OK
1981. 5. 1	NND1AB	4 OUT	NND1B	OUT	AVCS7	OUT
1981. 5. 9	NND1AB	4 OUT	NND1B	OUT	DP1	OUT
1981. 7. 5	NND1AB	4 OUT	NND1B	OUT	AVCS5	OUT
1981. 8. 0	NND1AB	4 OUT	NND1B	OUT	AVCS7	LAUNCHED
1981. 8. 0	NND1AB	4 OUT	NND1B	OUT	DP1	LAUNCHED
1981. 8. 0	NND1AB	4 OUT	NND1B	OUT	AVCS5	LAUNCHED
1981. 8. 0	NND1AB	4 OUT	NND1B	OUT	AVCS7	OK
1981. 8. 0	NND1AB	4 OUT	NND1B	OUT	DP1	OK
1981. 8. 0	NND1AB	4 OK	NND1B	OK	AVCS5	OK
1982. 2.16	NND1AB	4 OUT	NND1B	OUT	EPS2	OUT
1982. 2.20	NND1AB	4 OUT	NND1B	OUT	EPS2	LAUNCHED
1982. 2.20	NND1AB	4 OUT	NND1B	OK	EPS2	OK
1982. 9. 6	NND1AB	4 OUT	NND1B	OUT	AVCS2	OUT
1983. 0. 6	NND1AB	4 OUT	NND1B	OUT	AVCS2	LAUNCHED
1983. 0. 6	NND1AB	4 OK	NND1B	OK	AVCS2	OK
1983. 3.22	NND1AB	4 OUT	NND1B	OUT	EPS2	OUT
1983. 5. 6	NND1AB	4 OUT	NND1B	OUT	EPS2	LAUNCHED
1983. 5. 6	NND1AE	4 OUT	NND1B	OK	EPS2	OK
1983. 9.27	NND1AB	4 OUT	NND1B	OUT	AVCS7	OUT
1984. 0. 0	NND1AB	4 NG	NND1B	NG		

CHRONOLOGICAL TIME HISTORY OF SATELLITE POSITION IN ORBIT

TIME	SYSTEM	STATUS	SATELLITE	STATUS	MODULE	STATUS
1982. 1. 2	SORT7	1 OUT	SORT7	AVAILABLE		
1982. 1. 2	SORT7	1 OUT	SORT7	LAUNCHED		
1982. 1. 2	SORT7	1 OK	SORT7	OK		
1982. 1. 9	SORT7	1 NG	SORT7	REMOVED		

STATISTICAL SUMMARY FOR 25 MONTE CARLO CYCLES

FLIGHT SUMMARY

	SHUTTLE			TUG			SEPS		
YEAR	MIN	AVG	MAX	MIN	AVG	MAX	MIN	AVG	MAX
1980	4	4.2	5	4	4.2	5	0	0.0	0
1981	1	2.2	3	1	2.2	3	0	0.0	0
1982	2	3.2	5	1	2.2	4	0	0.0	0
1983	1	3.0	5	1	3.0	5	0	0.0	0
PROGRAM	11	12.7	16	10	11.7	15	0	0.0	0
PERCENT	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
DELAY	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

ORBIT TRAFFIC SUMMARY

ORBIT	AVERAGE FLIGHTS SHUTTLE	TUG	SEPS	AVERAGE UP WEIGHT SHUTTLE	TUG	SEPS	SHUTTLE ONLY LOAD FACTOR
SNC/0	0.0	11.7	0.0	0.	3179.	0.	0.00
1-1/28	0.0	0.0	0.0	0.	0.	0.	0.00
SORT	1.0	0.0	0.0	10000.	0.	0.	0.16

STATISTICS FOR SYSTEM - AST1D

MODULE	MIN		AVG	MAX	MIN FLT	AVG FLT	MAX FLT
NAST1C100							
AVCS3	0	0	0.2	2	0.00	0.06	0.80
AVCS5	0						
AVCS7	0						
AVCS7	0						
AVCS7	0						
AVCS7	0						
GN1	0	0	0.2	1	0.00	0.07	0.54
GN2	0	0	0.0	1	0.00	0.02	0.38
TTC5	0	0	0.0	1	0.00	0.04	1.00
OP1	0						
EPS1A	0	0	0.1	1	0.00	0.03	0.39
AST1C1	0	0	0.1	1	0.00	0.05	0.65
SATELLITE							
AST1D			1.0		0.40	0.49	0.76
EQ SAT			1.05				
SATELLITE TOTAL FLIGHTS					0.44	0.77	1.78
PERCENT SATELLITE AVAIL.					26.55	85.05	100.00
DELAY INTERVAL TO RESTORE					2.99	61.93	90.00

MODULE	MIN		AVG	MAX	MIN FLT	AVG FLT	MAX FLT
NAST1C100							
AVCS3	0	0	0.0	1	0.00	0.02	0.60
AVCS5	0	0	0.1	1	0.00	0.02	0.30
AVCS7	0	0	0.0	1	0.00	0.01	0.25
AVCS7	0						
AVCS7	0						
AVCS7	0						
GN1	0	0	0.2	1	0.00	0.05	0.39
GN2	0	0	0.2	1	0.00	0.05	0.40
TTC5	0	0	0.1	1	0.00	0.01	0.26
OP1	0	0	0.1	1	0.00	0.03	0.49
EPS1A	0	0	0.0	1	0.00	0.02	0.39
AST1C1	0	0	0.0	1	0.00	0.04	1.00
SATELLITE							
AST1D			1.0		0.30	0.47	0.50
EQ SAT			1.06				
SATELLITE TOTAL FLIGHTS					0.30	0.72	1.70
PERCENT SATELLITE AVAIL.					35.89	84.37	100.00
DELAY INTERVAL TO RESTORE					6.17	58.60	93.34
SYSTEM TOTAL FLIGHTS					0.88	1.49	2.20
PERCENT SYSTEM AVAILABLE					24.68	71.23	100.00
DELAY INTERVAL TO RESTORE					2.99	62.48	95.97

ORIGINAL PAGE IS OF POOR QUALITY

STATISTICS FCR SYSTEM - NNO1AB

MODULE	MIN		AVG	MAX	MIN FLT	AVG FLT	MAX FLT
NNND1	100						
AVCS2	0	0	0.1	1	0.00	0.03	0.49
AVCS5	0	0	0.6	2	0.00	0.32	2.00
AVCS7	0	0	0.2	1	0.00	0.13	1.00
AVCS7	0	0	0.3	2	0.00	0.10	0.99
AVCS7	0	0	0.2	1	0.00	0.04	0.49
AVCS7	0	0	0.1	1	0.00	0.03	0.50
GN2	0	0	0.4	2	0.00	0.17	2.00
TTC1	0	0	0.2	1	0.00	0.09	0.51
OP1	0	0	0.2	1	0.00	0.05	0.51
EPS2	0	0	0.2	2	0.00	0.12	1.21
EPS2	0	0	0.1	1	0.00	0.05	0.61
EPS2	0	0	0.2	1	0.00	0.11	1.00
EPS2	0	0	0.1	1	0.00	0.02	0.32
EPS2	0	0	0.3	1	0.00	0.14	1.00
EPS2	0	0	0.0	1	0.00	0.02	0.39
EPS3	0						
EPS3	0						
NND11	0	0	0.2	2	0.00	0.12	1.00
NND12	0	0	0.2	1	0.00	0.12	0.72
SATELLITE							
NND1A			1.2		0.89	1.18	3.00
EQ SAT			1.39				
SATELLITE TOTAL FLIGHTS					1.00	2.83	5.64
PERCENT SATELLITE AVAIL.					44.95	78.05	94.02
DELAY INTERVAL TO RESTORE					6.24	77.77	95.86

MODULE	MIN		AVG	MAX	MIN FLT	AVG FLT	MAX FLT
NNND1	100						
AVCS2	0	0	0.3	1	0.00	0.07	0.38
AVCS5	0	0	0.3	2	0.00	0.11	1.51
AVCS7	0	0	0.3	2	0.00	0.09	1.00
AVCS7	0	0	0.1	2	0.00	0.07	1.00
AVCS7	0	0	0.2	1	0.00	0.04	0.27
AVCS7	0	0	0.4	2	0.00	0.10	1.00
GN2	0	0	0.3	2	0.00	0.15	1.48
TTC1	0	0	0.2	1	0.00	0.05	0.29
OP1	0	0	0.2	2	0.00	0.05	0.87
EPS2	0	0	0.2	1	0.00	0.09	1.00
EPS2	0	0	0.3	1	0.00	0.12	0.61
EPS2	0						
EPS2	0	0	0.3	2	0.00	0.14	1.00
EPS2	0	0	0.2	1	0.00	0.06	0.61
EPS2	0	0	0.2	1	0.00	0.05	0.50
EPS3	0	0	0.0	1	0.00	0.00	0.08
EPS3	0						
NND11	0	0	0.2	1	0.00	0.06	0.65

NND12	0	0	0.1	1	0.00	0.04	0.70
SATELLITE							
NND1B			1.1		1.00	1.07	1.92
EQ SAT			1.26				
SATELLITE TOTAL FLIGHTS					1.08	2.37	4.23
PERCENT SATELLITE AVAIL.					68.34	84.77	98.97
DELAY INTERVAL TO RESTORE					0.22	61.03	96.53

MODULE	MIN		AVG	MAX	MIN FLT	AVG FLT	MAX FLT
NNN01	100						
AVCS2	0	0	0.1	1	0.00	0.05	1.00
AVCS5	0	0	0.5	2	0.00	0.23	1.03
AVCS7	0	0	0.2	2	0.00	0.09	1.00
AVCS7	0	0	0.2	1	0.00	0.07	0.50
AVCS7	0	0	0.3	1	0.00	0.08	0.50
AVCS7	0	0	0.3	1	0.00	0.09	0.50
GN2	0	0	0.3	2	0.00	0.15	1.31
TTC1	0	0	0.2	1	0.00	0.04	0.34
OP1	0	0	0.1	1	0.00	0.03	0.51
EPS2	0	0	0.3	2	0.00	0.09	0.50
EPS2	0	0	0.1	1	0.00	0.06	1.00
EPS2	0	0	0.1	1	0.00	0.08	1.00
EPS2	0	0	0.2	1	0.00	0.08	0.61
EPS2	0	0	0.2	2	0.00	0.13	1.24
EPS2	0	0	0.2	1	0.00	0.10	0.61
EPS3	0						
EPS3	0						
NND11	0	0	0.1	1	0.00	0.06	1.00
NND12	0	0	0.2	1	0.00	0.09	0.71
SATELLITE							
NND1A			1.1		1.00	1.10	1.92
EQ SAT			1.31				
SATELLITE TOTAL FLIGHTS					1.00	2.65	4.59
PERCENT SATELLITE AVAIL.					64.36	82.77	100.00
DELAY INTERVAL TO RESTORE					7.78	70.49	95.86

MODULE	MIN		AVG	MAX	MIN FLT	AVG FLT	MAX FLT
NNN01	100						
AVCS2	0	0	0.2	1	0.00	0.12	1.00
AVCS5	0	0	0.3	2	0.00	0.10	1.00
AVCS7	0	0	0.4	2	0.00	0.12	0.56
AVCS7	0	0	0.2	1	0.00	0.09	1.00
AVCS7	0	0	0.2	1	0.00	0.06	0.51
AVCS7	0	0	0.1	1	0.00	0.01	0.14
GN2	0	0	0.4	2	0.00	0.10	0.76
TTC1	0	0	0.1	1	0.00	0.06	1.00
OP1	0	0	0.2	1	0.00	0.06	0.49
EPS2	0	0	0.1	1	0.00	0.02	0.37
EPS2	0	0	0.2	1	0.00	0.10	1.00
EPS2	0	0	0.2	1	0.00	0.12	1.00
EPS2	0	0	0.2	1	0.00	0.07	1.00

EPS2	0	0	0.2	1	0.00	0.07	0.61
EPS2	0	0	0.2	1	0.00	0.06	0.62
EPS3	0						
EPS3	0						
NN011	0	0	0.0	1	0.00	0.02	0.54
NN012	0	0	0.1	1	0.00	0.07	0.71
SATELLITE							
NN018			1.2		0.88	1.09	2.00
EQ SAT			1.33				
SATELLITE TOTAL FLIGHTS					0.88	2.34	4.25
PERCENT SATELLITE AVAIL.					57.76	84.46	100.00
DELAY INTERVAL TO RESTORE					3.93	67.56	96.64
SYSTEM TOTAL FLIGHTS					8.46	10.19	13.10
PERCENT SYSTEM AVAILABLE					18.67	41.25	54.73
DELAY INTERVAL TO RESTORE					12.41	100.66	229.06

STATISTICS FOR SYSTEM - SORT7

SATELLITE				
SORT7	1.0	1.00	1.00	1.00
SYSTEM TOTAL FLIGHTS		1.00	1.00	1.00
PERCENT SYSTEM AVAILABLE		100.00	100.00	100.00
DELAY INTERVAL TO RESTORE		0.00	0.00	0.00

TOTAL SATELLITES 6.6

EQU. SATELLITES 7.39

MODULE SUMMARY

NAME	WARN MIN	WARN AVR	WARN MAX	FAIL MIN	FAIL AVR	FAIL MAX	REPLACE MIN	REPLACE AVR	REPLACE MAX
AVCS2	0	0.0	0	0	0.8	3	0	0.7	3
TTC5	0	0.0	0	0	0.2	2	0	0.1	1
DP1	0	0.0	0	0	0.7	3	0	0.7	3
EPS1A	0	0.0	0	0	0.2	1	0	0.1	1
EPS2	0	0.0	0	0	4.2	7	0	4.0	7
EPS3	0	0.0	0	0	0.0	1	0	0.0	1
AST1C1	0	0.0	0	0	0.1	1	0	0.1	1
AVCS3	0	0.0	0	0	0.2	2	0	0.2	2
AVCS5	0	0.0	0	0	1.9	4	0	1.7	4
AVCS7	0	0.0	0	2	3.9	8	1	3.5	8
GN1	0	0.0	0	0	0.4	2	0	0.3	1
GN2	0	0.0	0	0	1.9	5	0	1.6	5
TTC1	0	0.0	0	0	0.8	2	0	0.8	2
NND11	0	0.0	0	0	0.6	3	0	0.6	3
NND12	0	0.0	0	0	0.6	2	0	0.6	2
NASTIC									
NNND01	0	0.0	0	0	0.5	2	0	0.0	0
MODULE									

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX D: CLARIFICATION OF INTERNAL MECHANISMS

This appendix has been added to the original document to clarify some of the internal mechanisms of the program. In particular, emphasis is placed on what types of modes of operation can be invoked by a slight change in the input data. The following is a list of options described in this appendix:

- Quit on Heavy Payload
- Scooter Mode
- Switch from Reusable Tug to Expendable
- Expendable Model Flag with Transition
- Service Policies
- Timing Parameters
- Launch Pads
- Satellite Priority
- Mandatory Launch Priority
- Satellite Terminition Time
- FREEBIES
- SEPS Modes of Operation
- TDOWN
- Scheduling Delays
- Mandatory Waits
- Not Mixing SEPS and Tug Flights
- Automatic Count of Modules on Satellites

D.1 QUIT ON HEAVY PAYLOAD

The quit on heavy payload option has been implemented with two alternatives. The option is controlled by the variable QUIT (Array No. 129) in the initialization table. It has an integer value of 0 or 1.

D.1.1 QUIT = 0

This is the normal mode of operation of the program in which the program is permitted to proceed on an error as far as is deemed feasible. Normally, the program stops on an input error at the completion of processing all data. Also, the program stops at the end of a Monte Carlo cycle, if excessively heavy payloads were encountered during vehicle performance computations. This stopping action is intended to operate the program for a maximum of information at a minimum of cost.

D.1.2 QUIT = 1

In this mode, the user has deliberately specified that he wishes to proceed, even though there have been errors. Currently the error flag for excessively heavy payloads is the only error that is blocked by this value of QUIT. However, the error message is still output, and the payload is dropped from the loading queue.

D.2 SCOOTER MODE

The scooter mode of operation has two alternatives: 1) the Tug and SEPS fly in a normal fashion (the Tug and SEPS meet at an intermediate altitude). 2) the SEPS is restricted to operations at sync eq. The option is controlled by the variable SCOOT (Array No. 298) in the initialization table. It has an integer value of 0 or 1.

D.2.1 SCOOT = 0

This is the normal mode referred to above, and described in detail elsewhere.

D.2.2 SCOOT = 1

In this mode, the SEPS can only be delivered to sync eq orbit with other payloads. The SEPS operates within the limits of the orbit until the fuel is nearly depleted. Then the SEPS can fly down to an intermediate altitude to meet the Tug with or without some payloads that are being retrieved. The SEPS would then be replaced by another SEPS delivered to sync eq.

D.3 SWITCH FROM REUSABLE TUG TO EXPENDABLE

In certain applications, situations arise in which a normal vehicle (upper stage) cannot deploy a payload in a reusable mode because the payload weight exceeds the capability of the vehicle in this mode. The program will automatically shift the vehicle into an expendable mode and attempt to perform the mission. This is done only when there is a single payload at the beginning of the loading queue. If the vehicle upgrade is successful, the program will attempt to load as many payloads on the flight as possible. If the prime vehicle is the SEPS and it cannot perform the flight, the program will shift to the reusable Tug and then to the expendable Tug, if necessary. The shifting process can be inhibited by setting the flag NOTUG (Array No. 117) to 1 in the initialization data.

D.4 EXPENDABLE MODEL FLAG WITH TRANSITION

The normal usage of the program is to simulate various modes of servicing reusable satellites. There is a flag called EXMOD (Array No. 57), which if set non-zero will have the effect of flying an expendable model. No modules are replaced and if a module does fail, the satellite will be replaced as if an NRU failure had occurred. As a part of this feature, several other variables and options were implemented. The satellite deployment weight can be different between the two modes, therefore, the weight of an expendable satellite can be input as a separate input. If the weight is zero, the program will use the reusable weight in both places. The satellite variable XSAT was added to mark the satellite as once having been launched, it had the properties of an expendable satellite. Three other variables were introduced and are defined in the initialization table. They are MODB (Array No. 56), MODS (Array No. 79) and TIMEG (Array No. 84). The functions is: prior to TIMEG, EXMOD is set to MODB; after TIMEG, EXMOD is set to MODS. This permits the simulation of a transition from expendable satellites to reusable ones. The XSAT tag is required to maintain the proper reaction to failures during the transition phase.

D.5 SERVICE POLICIES

There are three forms of service policies available to the user. There is PDOWN (Array No. 12) which controls the response to the failure of an NRU on a satellite. This policy is global and affects all satellites. If PDOWN is zero, there is no response to an NRU failure. If not zero, then the response is in accordance with the individual satellite policies which are defined as:

- 0 no reaction
- 2 deploy a new satellite
- 3 deploy and retrieve the satellite
- 4 retrieve the satellite

Depending on the policy value, the appropriate payload may be entered into the loading gueue. The final service policy is the global policy variable PSERV (Array No. 98). This variable controls the retrieval of modules from orbit (strongly affects flight rates) and has the following values:

- 0 all modules and SUs are retrieved
- 1 nothing is retrieved
- 2 only SUs are retrieved

A wide variety of conditions can be simulated through the use of these policies.

D.6 TIMING PARAMETERS

There are three variables available for control of the timing of the simulation. They are: TIMEB (input on the event card), TIMES (also on the event card), and TIMEC (Array No. 11). TIMEB is the time at which the simulation starts; no payloads enter the loading queue before this time. TIMES is the termination time of the simulation; all systems are turned off at this time and no modules can enter the loading queue in the three months prior to this time. The variable TIMEC is used to control the random number generator. A value of zero permits random events to occur. A non-zero value yields that value as the output of the random process, thereby making it a predictable sequence of events. This is valuable in verifying the program as it is passed from machine to machine, for not all random number generators yield the same sequence. A practical application is to input a value of 0.85 in which all modules fail at the 15 percent point of their life (useful in studying extremals).

D.7 LAUNCH PADS

There is an array named VPAD which can be defined as 4 items in size. This would be interpreted as: there are 4 launch pads available in the simulation. If the user does nothing, the 4 pads are automatically available to each and every orbit. However, by the use of inputs on the orbit data cards, the user may split the pads into 3 located at ETR and used to fly to certain orbits; the remaining one can be assigned as WTR to fly to certain other orbits. All Shuttles and Tugs are assumed to be available on demand at either location.

D.8 SATELLITE PRIORITY

The satellite data card has an entry for satellite priority. This parameter affects the placement of a payload in the loading queue, as the lower the priority, the closer to the beginning of the queue the payload will be placed. The range of values is typically 1-6 with 6 being the programmed-in default value. An assignment of a low priority is recommended for payloads that can cause a shift to an expended Tug. This will have the effect of putting the payload at the head of the queue and prevent a flushing action from taking place (the previous flight will be flown if the current payload cannot be added to the flight). This application will yield a more effective usage of the Tug.

D.9 MANDATORY LAUNCH PRIORITY

When payloads are placed in the loading queue, they are given a rank of TIME, plus priority, plus DELTA and perhaps a mandatory launch event will be scheduled at a later time (DELTA =0). When the mandatory launch event occurs, the loading queue is scanned for the payload, and if still there, the payload is reranked with a value of the priority alone. This operation quarantees that the mandatory launch requirement is fulfilled by having the payload included on the next available flight. Previously set up flights are destroyed by this process.

D.10 SATELLITE TERMINATION TIME

The input on each satellite, for the satellite termination time, is used to turn off the satellite x years after it becomes operational. The satellite is turned off at the earlier of; satellite termination time, system termination time, or the end of the simulation (TIMES). When the satellite is turned off, the satellite policy is checked to determine the need for retrieval and/or deployment of the satellite.

D.11 FREEBIES

Satellites consist of modules that can be further classed as NRUs, single string, subsystems with redundancy, and single FREEBIES. As discussed before, NRUs are repaired only by replacing the entire satellite. Single string modules are replaceable, but the satellite is out of service until the module is replaced. Redundant subsystems consist of several modules with the stipulation of n active elements out of a possible m elements in the subsystem. If the subsystem consists of at least three elements, the priority of replacement can be controlled by entering a number in the third module slot to define how many non-essential failures can be tolerated. For example, a subsystem of 5 modules with the requirement of 2 active modules for an active system and 1 non-essential failure, implies the first two failures are FREEBIES (noncritical replacements) and that the third failure would be placed in the loading queue as a normally failed module, but without an accompanying mandatory launch event (because of the 1 non-essential failure). Certain single string modules (scientific experiments just along for the ride) can be classed as FREEBIES if subsystem size of 1 is entered in the data. A length of 1 is implied for all blank entries, but a specific entry transforms the module into a FREEBIE. The number of FREEBIES in the loading queue can sometimes be excessive leading to the conclusion that there is more weight to be delivered to orbit than is really necessary. The means of limiting the number of FREEBIES is via the vaiable LIMIT (Array No. 113). FREEBIES whose rank is in excess of the value of LIMIT are not put into the loading queue. A value of 4000 prevents any FREEBIES; a value of 5000 admits only the most critical (DELTA = 3000 + 1000 times the number of FREEBIES in the subsystem that have not yet failed).

D.12 SEPS MODES OF OPERATION

The SEPS is unique in that its mission profile begins with the delivery to sync eq. of the SEPS and a payload aboard the Tug. Thereafter the SEPS comes down to some intermediate altitude to hand off retrieved payloads or modules to the Tug, and to receive from the Tug payloads to be deployed by the SEPS through an ascent phase and any necessary phasing manuevers.

The routine SEPIM is called with a group of payloads and with the previous state of the SEPS stored in SEPIM, a trial flight is attempted. SEPIM returns a flag NEXIT which gives the results of the attempted flight. The permissible values are:

NEXIT	Interpretation
1	new SEPS at min altitude with payload
2	new SEPS at sync eq. with payload
3	no good
4	no good
5	ok - SEPS down to meet Tug
6	ok - SEPS and Tug meet at sync eq
7	no good
8	retrieve SEPS without retrieval payload
9	retrieve SEPS without retrieval payload
10	retrieve SEPS with retrieval payload

Should the driving routine PROP2 decide to reject that SEPS flight to attempt another flight, the SEPS erase flag MSEP is used to restore the SEPS to the condition prior to the flight.

D.13 TDOWN

The instant the SEPS completes its mission and becomes available for the next mission, is the time the SEPS is considered as beginning the descent manuever to meet the Tug at some intermediate altitude. This is done even though the flight may not be known for several months. This premature departure can result in considerable savings in operational capability of the SEPS.

D.14 SCHEDULING DELAYS

Certain scheduling delay effects can be studied by variation of the parameters WAIT1 (Array No. 18), WAIT2 (Array No. 19), WAIT3 (Array No. 47), and WAIT4 (Array No. 99). When a satellite is put into the loading queue, the satellite policy is checked and satellite deployments are scheduled WAIT1 after termination. Satellite retrievals are scheduled WAIT2 after the deployment. By appropriate selection of values, one can simulate ground refurbishment, preference of retrieval over deployment, and advance deployment to prevent satellite outage.

When an NRU fails, a new satellite is required to be launched as a replacement (if the satellite policy so requests). The earliest time the satellite can become available is WAIT3 after the NRU failure. WAIT3 is a global variable, affecting all satellites.

The parameter WAIT4 is used as the delay time for the event QWAIT after a module failure, or as a warning for the activities of removal of module from a shelf and reconditioning it for usage.

D.15 MANDATORY WAITS

The delay times for the mandatory events are given by the four parameters: WSATU (Array No. 20), WSATN (Array No. 21), WMODU (Array No. 22), and WMODN (Array No. 23).

D.16 NOT MIXING OF SEPS AND TUG FLIGHTS

The program keeps the traffic for the SEPS separate from that of the Tug if and when a Tug flight is flown with the need of the SEPS.

Also, no SEPS down traffic can be retrieved by a Tug alone; that is, the SEPS was assigned to handle the payload and so it shall. The program keeps no record of where the SEPS flight terminates (the longitude is not noted) and thus, the phasing manuever required to get from the last position of the previous mission to the first position of the next mission is not performed.

D.17 AUTOMATIC COUNT OF MODULES ON SATELLITES

This is a new feature implemented in the routine that reads in the satellite data. The feature consists of three items: the user need not supply the module count on the satellite card, the last card of the module list contains the word Last in the first module position (columns 11-14), and the user need not fill in all seven entries on a card. The last item is particularily handy with subsystems where the subsystem is all the data on a card, and a single card can contain only an NRU, etc. Revisions to the module list are very easily accommodated with this technique. Of course, the input routine accepts the data in the rigid format described in the earlier portion of the document.

AEROSPACE CORPORATION

INTERNAL DISTRIBUTION LIST

(REFERENCE: COMPANY PRACTICE 7-21-1)

REPORT TITLE

Manned Systems Utilization Analysis (Study 2.1) Final Report
Volume IV: Program Manual and Users Guide for the LOVES Computer Code

REPORT NO.	PUBLICATION DATE	SECURITY CLASSIFICATION
ATR-76(7361)-1, Vol IV, formerly ATR-74(7341)-6)	1 September 1975	Unclassified

(NOTE: FOR OFF-SITE PERSONNEL, SHOW LOCATION SYMBOL, e g. JOHN Q PUBLIC/VAFB)

N. R. Campbell	(1)
T. J. Lang	(1)
L. T. Stricker	(1)
S. T. Wray	(3)
R. R. Wolfe	(2)

APPROVED BY Robert R Wolfe DATE 8/31/75

SHEET______ OF ______

AEROSPACE FORM 2394 REV 7-66

THE AEROSPACE CORPORATION

EXTERNAL DISTRIBUTION LIST

(REFERENCE: COMPANY PRACTICE 7-21-1)

REPORT TITLE
Manned Systems Utilization Analysis (Study 2.1) Final Report
Volume IV: Program Manual and Users Guide for the LOVES Computer Code

REPORT NO.	PUBLICATION DATE	SECURITY CLASSIFICATION
ATR-76(7361)-1, Vol IV, formerly ATR-74(7341)-6)	1 September 1975	Unclassified

MILITARY AND GOVERNMENT OFFICES	ASSOCIATE CONTRACTORS AND OTHERS

(NOTE SHOW FULL MAILING ADDRESS, INCLUDE ZIP CODE, MILITARY OFFICE SYMBOL, AND "ATTENTION" LINE)

NASA Scientific & Technical
Information Facility (3)
P. O. Box 33
College Park, Maryland 20740

NASA - Headquarters
Washington, D.C. 20546

Fred Roberts, Code MTS (50)

New Technology Rep., Code KT (1)

NASA
Mr. Duncan Collins
P. O. Box 92960
Worldway Postal Center
Los Angeles, CA 90000
Building 120, Rm 1406B

☐ THIS DISTRIBUTION LIST PERMITS ONLY PRIMARY DISTRIBUTION OF THIS TOR

AFR 80-45 DISTRIBUTION STATEMENT X'D BELOW APPLIES

☐ NO DISTRIBUTION STATEMENT
(Classified documents only)

☐ A APPROVED FOR PUBLIC RELEASE, DISTRIBUTION UNLIMITED

☐ B DISTRIBUTION LIMITED TO U S GOV'T AGENCIES ONLY, ______ (Reason)

______ (Date statement applied). OTHER REQUESTS FOR THIS DOCUMENT MUST BE REFERRED TO ______ (Controlling DOD office)

APPROVED BY Robert R Wolfe (FOR THE AEROSPACE CORPORATION) DATE 8/31/75

APPROVED BY ______ (FOR COGNIZANT AF OFFICE) ______ (SYMBOL) DATE ______

IF LIST COMPRISES TWO OR MORE SHEETS, COMPLETE THIS SIGNATURE BLOCK ON LAST SHEET ONLY

SHEET ____ OF ____

AEROSPACE FORM 2380 REV 7-75