JSC-16395

Scheduling Algorithm for Mission Planning and Logistics Evaluation (SAMPLE)

Volume II
Mission Payloads Subsystem Description

February 1980

(NASA-TM-81115) SCHEDULING ALGORITHM FOR MISSION PLANNING AND LOGISTICS EVALUATION (SAMPLE). VOLUME 2: MISSION PAYLOADS SUBSYSTEM DESCRIPTION (NASA) 182 p HC A09/MF A01 CSCL 22A G3/16

N80-20413

Unclas 28326

NASA

National Aeronautics and Space Administration

Lyndon B. Johnson Space Center
Houston, Texas 77058

SCHEDULING ALGORITHM FOR MISSION PLANNING AND LOGISTICS EVALUATION (SAMPLE)

VOLUME II

MISSION PAYLOADS SUBSYSTEM DESCRIPTION

Prepared by

Edwin Dupnick
Utilization Planning Office
Shuttle Payload Integration and
Development Office

and

Diana Wiggins
Lockheed Electronics Company, Inc.
Aerospace Systems Division

FOREWORD

These reports document the eighth baseline version (SA8) of the Scheduling Algorithm for Mission Planning and Logistics Evaluation (SAMPLE). Volume I is the Users' Guide for SAMPLE, Volume II documents the Mission Payloads (MPLS) subsystem, the primary computational portion of SAMPLE, and Volume III discusses the GREEDY algorithm, the technique used to solve a set covering problem and determine a traffic model.

CONTENTS

Section	Page

TABLES

Table	Page

FIGURES

Figure	Page

DEFINITIONS

Term	Definition
ETR launch	A launch performed from the Eastern Test Range
Feasible combination	A collection of payloads that meet certain system constraints (e.g., Shuttle weight-to-orbit capability)
Flight schedule	A traffic model to which Shuttle resources are assigned
Load factor	The maximum value of two ratios found by comparing actual weight-to-orbit of a mission to its theoretical capability and the total cargo weight at landing versus the maximum allowed
Mission type	The Orbiter assignment for a payload. The payloads type either accompany the Shuttle to orbit (deployment), from orbit (retrieval), or both (in the case of attached and service payloads).
MPLS	The Mission Payloads Subsystem
OMS fuel	The fuel required for the Orbital Maneuvering System
Payload	One or more integrated experiment packages and associated third stage(s), if any, or simply a third stage itself
Payload margin	The minimum value of two calculations found by subtracting the total Shuttle weight at insertion from its theoretical weight-to-orbit capability, and the total cargo weight at landing subtracted from the maximum allowed
Payload type	The mission type for a payload
RCS fuel	The fuel required for the Reaction Control System
Traffic model	A subset of the list of feasible combinations such that there are no redundant loads which can cover all the payloads. (The same payload is not assigned to two or more distinct flights.)
TSV	Third stage vehicle
Up/down payload	A payload that is deployed and retrieved in the same year
WTR launch	A launch performed from the Western Test Range

THE PAYLOAD DISCIPLINE CODES

Symbol	Experiment
AS	Astronomy
OP	Earth and Ocean Physics; same as EP
CN	Command and Navigation
ST	Space Technology
LS	Life Sciences
EO	Earth Observation
PL	Planetary
SS	(undefined)
SP	Space Processing
PH	Physics
LU	Lunar
OA	Office of Applications
AP	Atmospheric
SO	Solar Physics
HE	High Energy Astropysics
EP	Earth and Ocean Physics
NN	Non-NASA

1. INTRODUCTION

The Scheduling Algorithm for Mission Planning and Logistics Evaluation (SAMPLE) is an interactive computer program for automatically generating traffic models for the Space Transportation System (STS). The SAMPLE is composed of three major subsystems: the Mission Payloads (MPLS) program and the Set Covering Program (SCP). The MPLS program determines a set of payload combinations which satisfy various STS constraints, such as: the maximum weight-to-orbit capability, cargo bay capacity, Reaction Control System (RCS) and Orbital Maneuvering System (OMS) fuel capacities, etc. The SCP forms a subset (traffic model) of the feasible payload combinations from MPLS such that a minimum number of Shuttle flights will transport all the specified payloads without redundancies.

The SAMPLE was written in FORTRAN V and was designed to execute on the UNIVAC 1100 series computers using the EXEC 8 operating system. The program was written to be used primarily in a demand (interactive) mode, but it may also be run in the batch mode.

This document describes the Mission Payloads Subsystem (MPLS) program and includes a general and technical program description as well as subroutine documentation and program functional flow.

The MPLS, a subsystem of the Scheduling Algorithm for Mission Planning and Logistics Evaluation (SAMPLE), was designed to generate a list of feasible combinations (LFC) from a payload model for a given calendar year. The Set Covering Algorithm (SCA), another subsystem of SAMPLE, uses the LFC to determine an optimum traffic model.

This technique used by the MPLS to determine the validity of a combination is based on payload sequence dependent and independent constraint tests. The independent constraints are performed first to eliminate those missions which fail due to simple parameters tests; the dependent constraints are tested to determine the feasibility of the combination with respect to delta velocity (ΔV) requirements. The specific order of the tests tends to minimize the computer time required to examine the data.

Since the MPLS began to evolve in 1973, the logic has been modified wherever necessary to incorporate new requirements and capability. The changes significantly decrease program run time and increase the number of feasible solutions found.

2. PROGRAM DESCRIPTION

2.1 GENERAL DESCRIPTION

The MPLS generates and evaluates payload combinations from an input payload model for a specific year. The feasibility of each combination is evaluated by flight sequence tests, both dependent and independent. The flight sequence is a series of maneuvers designed to achieve the orbital requirements of each payload in the combination; its validity is contingent upon the accumulative impact of the payload characteristics compared to the system constraints. A combination is infeasible when a constraint or set of constraints have been violated.

2.1.1 Flight Sequence Independent Tests

The flight sequence independent tests are controlled by subroutine CPTEST. Tests are made for payload redundancy, mission type, discipline mix, weight, length, Reaction Control System (RCS) fuel, third stage vehicle (TSV) cargo capacity, and miscellaneous constraints. Each constraint is a gross check of a combination against performance limits of the Orbiter.

2.1.1.1 Payload Redundancy

Each payload combination is examined for redundant payloads, rejecting those combinations with duplicate payload names. Payloads are exempted from this test with the use of the flag IRPT (see Payload Model data).

2.1.1.2 Mission Type

Each combination's mission type is compared to an internal list, rejecting those that do not match. <u>This test is optional.</u>

2.1.1.3 Discipline Mix

The combinations discipline mix is compared to an allowable list, rejecting those that do not match. <u>This test is optional.</u>

2.1.1.4 Weight

The total chargeable weight in the Orbiter's cargo bay is compared to a limit of 65,000 lb at launch and 32,000 lb at landing. The chargeable payload weight is composed of the individual payloads (with TSV's if included) flown on the mission. A combination that exceeds the limit is rejected.

2.1.1.5 Length

The total length of the payloads at launch and landing is compared to a limit of 60 feet. A combination that exceeds the limits is rejected.

2.1.1.6 RCS Fuel

The RCS fuel is a function of the number of rendezvous. If more than 9000 lb of fuel is required, the combination is rejected.

2.1.1.7 TSV Cargo Capacity

The total payload plus TSV length and weight at launch and landing are compared against system limits. If a TSV cannot be found which meets the combination's requirements, it is rejected.

2.1.1.8 Miscellaneous Constraints

The total number of payloads on a TSV must be less than four. Only one dedicated TSV payload is permitted per combination.

2.1.2 Flight Sequence Dependent Tests

The flight sequence dependent tests are controlled by subroutine SDTL. These tests consider the ΔV fuel usage by using a form of the ideal rocket equation as well as payload weight and length as a function of its center-of-gravity (CG). The fuel requirements of the TSV are not known initially, so the Orbiter and TSV are considered separately. The total TSV configuration is then used as an Orbiter payload.

2.1.2.1 Orbiter Requirements

The Orbiter has been designed to satisfy the requirements of nominal (low altitude) payloads; high altitude payloads are serviced by the use of a TSV. The TSV is considered to be an Orbiter payload deployed at the first orbit or at an altitude of 150 miles (whichever is less) and retrieved at 20 miles above the last Orbiter orbit. The Orbiter requirements are determined by computing the orbit-to-orbit ΔV's, the TSV requirements, and the ΔV fuel used for deorbit. The fuel requirements may exceed the maximum Orbital Maneuvering System (OMS) main tank fuel capacity, so the OMS capability may be extended by the use of up to three OMS kits. The maximum number of allowable OMS kits can be specified by the user. The use of OMS kits reduces the cargo bay capacity, as the dry weight of the kits at landing must be included with payload down weight, assuming that all OMS fuel is exhausted. Furthermore, the OMS kits physically intrude into the cargo bay envelope.

2.1.2.2 TSV Requirements

A combination requires a TSV if at least one payload has a desired orbit of 700 miles or more. The type of TSVs available are:

a. Expendable - Used to deploy a payload but not retrieved

b. Reusable - Used to deploy a payload and retrieved

In making an assignment, the first TSV meeting all requirements is used, then the TSV requirements are included in the Orbiter's payload sequence. If the Orbiter's capability is exceeded, the next TSV is examined. If all available TSVs fail to meet the requirements of the payload sequence, the combination is rejected.

2.2 TECHNICAL DESCRIPTION

2.2.1 Analysis

The solutions generated by the MPLS tend to compute the minimum ΔV for a particular mission without regard to payload type; the marginal cases may be rejected as only the initial sequence is examined. (An exhaustive numeration technique has been inhibited in the current program version.) The ΔV computations are performed by the following techniques.

2.2.1.1 Hohmann Transfers

The Hohmann transfer algorithm is used to compute the ΔV required to change orbits in all instances except deorbit. The initial (insertion) transfer is made from an elliptical orbit; the other transfers are made to and from circular orbits.

2.2.1.2 Empirical Equation

The deorbit is simulated by using an empirical formula which computes the deorbit ΔV from the inclination and altitude of the last parking orbit so as to satisfy reentry conditions.

Fuel usage is computed for both the Orbiter and TSV by a form of the ideal rocket equations, incorporating ΔV requirements.

The MPLS assumes a 20 ft/sec ΔV from the OMS for any rendezvous maneuver; hence, the best possible phasing required to accomplish the rendezvous is considered. This optimistic philosophy may be justified because of the built-in program penalties:

a. An OMS fuel reserve of 1660 lb is carried from insertion to landing. This reserve requires an additional 295 lb of fuel for a mission requiring a total ΔV of 1600 ft/sec.

b. An RCS fuel reserve of 3400 lb is carried from insertion to landing. This reserve requires an additional 580 lb of fuel for a mission requiring a total ΔV of 1600 ft/sec.

2.2.2 Method of Solution

The Orbiter/TSV mission is composed of three distinct segments:

a. Insertion of the Orbiter into initial parking orbit and deployment of the TSV

b. Orbiter/TSV maneuvers performed in earth orbit

c. Orbiter reentry

The function of the trajectory calculations in the MPLS is to compute the ΔV requirements for each segment. In general, the ΔV is computed using a Hohmann transfer algorithm at the insertion to parking orbit and maneuver phase; the reentry ΔV is computed using an empirical equation (ref. 1). The specific algorithms for computing the ΔV are:

a. Segment 1 - Entry point DVIPK of subroutine DLTAV is used to compute the ΔV required to transfer from insertion ellipse to the initial circular parking orbit. The following heuristic is used

$$\Delta V = 3.35(H - 100) + 200 \quad \text{(fps)}$$

where H is the parking orbit altitude

b. Segment 2 - Functions DLTAV and entry point DVEL compute the ΔV between two circular parking orbits. Computations are performed with conic equations.

c. Segment 3 - Entry point DVDORB of subroutine DLTAV computes the deorbit ΔV as a function of the altitude and inclination of the last parking orbit.

The following heuristics are used depending on the last parking orbit altitude

$$\begin{aligned} \Delta V &= 1.25H + 132 && H > 145 \\ &= .638H + 222 && 110 < H < 145 \\ &= 291 && H < 100 \end{aligned}$$

where H is the circular parking orbit altitude prior to deorbit.

The mass and length history computation requires discrete weight, length, and velocity changes at each maneuver. The basic approach considers the total payload down (deorbit) and then computes the fuel requirement backwards to insertion. The equation used is the ideal rocket equation

$$W_S = W_E \cdot e^{\Delta V / g I_{sp}}$$

where

W_S = Weight at the start of the maneuver

W_E = Weight at the end of the manuever

g = Acceleration due to gravity

I_{sp} = The specific impulse for the Orbiter OMS engines

The total weight at landing is

$$W_L = W_V + W_{RCS} + W_{OMS} + W_{KITS} + PL_D$$

where

W_V = Vehicle dry weight

W_{RCS} = Weight of the RCS fuel reserve

W_{OMS} = Weight of the OMS reserve fuel

W_{KITS} = OMS kit dry weight

PL_D = Weight of the payload(s) carried down

Once the weight of the vehicle is determined at insertion, the total weight of the OMS fuel is computed as

$$W_{FUEL} = W_I - W_V - W_{UP} - W_{TRCS}$$

where

W_{UP} = Weight of the Orbiter payloads at insertion, including the TSV, its fuel, and payloads

W_{TRCS} = Total weight of the RCS fuel used

W_I = Weight of the vehicle at insertion

Assuming that the OMS fuel is within the maximum 58350 (integral OMS tank + 3 kits) lb allowed, the total weight of the vehicle must be readjusted to allow for OMS kits, if they are required. A kit is needed if the OMS fuel required exceeds 23340 lb (Table I) and the additional fuel used to carry the dry kit is computed as

$$W_{DRY} = W_{KITS} \; e^{T_{DV}/gI_{sp}} - 1$$

and

$$W_{UP} = W_{KITS} + W_{DRY}$$

where T_{DV} = total Orbiter delta velocity required. The down weight is then adjusted as

$$W_{DOWN} = W_{KITS} + W_{DRY} + \sum_{i=1}^{n} W_{PLD_i}$$

where W_{PLD_i} = the weight of the ith payload

The length history is maintained by the summation of the discrete length changes for each payload activity. The length of any OMS kits will reduce the length available for the payloads.

Both the total insertion weight of the Orbiter and the total length requirements for the payloads must be reverified. The total insertion weight is compared to the total weight-to-orbit capability of the Orbiter. The insertion weight-to-orbit is computed empirically as a function of the initial payload orbit inclination.

When all other constraints have been met, then it is necessary to determine if the cargo CG is acceptable. This test is optional. The basic form of the equation used to compute CG is

$$CG = \frac{W_1 \quad d + C_1 \quad + \sum_{L=2}^{N} \quad d + \ell_i + C_i}{\sum_{L=1}^{N} W_i}$$

where

W_i = Weight of the ith payload in the cargo bay

d = Distance from the front wall to the first payload

ℓ_i = Total length of the payloads and reserved gaps between payloads

C_i = the CG of the ith payload in the cargo bay

The CG is tested against the cargo minimum and maximum length constraints to determine mission feasibility. Logic exists within the CG processor for rearranging the payloads in an attempt to satisfy CG constraints.

TABLE I.- SHUTTLE CHARACTERISTICS

DESCRIPTION	LENGTH (FT)	DRY WEIGHT (LB)	PROPELLANT CAPACITY (LB)	Δ V (FT/SEC)
MAIN OMS TANK			23340	
1 OMS KIT	9.42	4100	12500	
2 OMS KITS	9.42	5000	25000	
3 OMS KITS	9.42	6100	37500	
RCS MAIN TANK			3974	
RCS TANK 2			3400	
RCS SPILLOVER			2000	
RESERVE OMS AT LANDING	(DISPERSIONS + RESIDUALS)		1660	
OMS RENDEZVOUS REQUIREMENTS		UNIQUE FOR EACH PAYLOAD		20
RCS RENDEZVOUS REQUIREMENTS		1800		20

ORBITER DATA

	LENGTH (FT)	WEIGHT (LB)	
CARGO BAY (CAPABILITY)	60	65000 32000	(MAXIMUM UP) (MAXIMUM DOWN)
OV102 DRY WEIGHT		178860	
OV099 DRY WEIGHT		173500	
OV103 DRY WEIGHT		175000	
SPECIFIC IMPULSE		313.2 SEC	
MINIMUM OMS LOAD FOR ABORT:			
ETR LAUNCH		13500	
WTR LAUNCH		18700	

3. PROGRAM USAGE

3.1 INPUT DESCRIPTION

Input to the MPLS is a subset of the input required for the SAMPLE. Information in this section supplements Volume I, SAMPLE User's Guide, as applied to the MPLS. Details of the TSV and payload data sets are provided. All input is represented as data card images, regardless of whether they originate on cards or line images from a demand terminal.

The first user input, an executive request, identifies the payload model; the input format of the payload model is described in detail in table II. The remaining inputs specify user options. The user options are input, starting in column 1, as responses to prompts. Each prompt may be explained in detail by entering a zero and a carriage return. A sample job stream is given in section 5.4.

3.2 OUTPUT DESCRIPTION

3.2.1 Normal Output

The output of the MPLS can be classified as input data and trajectory data. The initial output consists of approximately three pages of information pertaining to the payload model and is optional. The next set of output pertains to the relationship of the payloads to their mission type, discipline mix code, the TSV, and the number of flights per year.

The trajectory section prints the following parameters for feasible combinations:

Flight number

Launch site

Payload identification number/name

Orbiter sequence

Inclination

Total weight up/down

Up/down length

TSV name

TSV sequence

Altitude

Payload type

Orbiter and TSV ΔV

Number of OMS kits

Load factor

Payload margin

Percentage used of the first OMS kit

The total number of combinations generated is printed at the end of the trajectory section, as well as the number of feasible and infeasible combinations. The Feasible Mission File is also output; refer to table III.

3.2.2 Abnormal Output

Subroutine ERRPRT is referenced by the following diagnostic messages.

Diagnostic Message	Subroutine
DOWN WEIGHT CONSTRAINT VIOLATED	ERRPRT
MISSION TYPE NOT ALLOWED	ERRPRT
NO FEASIBLE SEQUENCE FOUND	ERRPRT
NO TUGS SATISFY LENGTH AND WEIGHT CONSTRAINTS	ERRPRT
NUMBER OF PAYLOADS ON A TUG GREATER THAN 3	ERRPRT
PAYLOAD iiii CAN ONLY BE ON A DEDICATED TUG	ERRPRT
PAYLOAD DISCIPLINE MIX TYPE NOT ALLOWED iiiiii	ERRPRT
THE RCSWT IS GREATER THAN THE CAPACITY FOR THIS CASE	ERRPRT
TOTAL LENGTH GREATER THAN BAY LENGTH, DOWN TOTAL LENGTH = rrrrr.rr	ERRPRT
TOTAL LENGTH GREATER THAN BAY LENGTH, UP TOTAL LENGTH = rrrrr.rr	ERRPRT
UP WEIGHT CONSTRAINT VIOLATED	ERRPRT
INCLINATION RANGE GREATER THAN .5	ERRPRT

3.3 PROGRAM UNITS

English units of measure are used for the MPLS. Units given for altitudes, inclinations, ΔV, specific impulse, weight, and length are nautical miles (n.mi.), degrees (deg), feet/second (ft/s), seconds (sec), pounds (lb), and feet (ft).

TABLE II.- PAYLOAD MODEL CARDS

The first set of cards in the payload model identifies the solid-propellant Interim Upper Stage (IUS) data. The second set of cards pertains to the Liquid-Propellant Upper Stage (LUS) data, and the remaining cards identify individual payload characteristics.

Card	Word	Symbol	Type	Units	Format	Column	Description
1	1	N	I	-	Free	-	Number of unique stages in the model
2-L ($L \leq 15$)	1	TISP	R	sec	F10.0	1-10	Specific impulse of the stage
	2	TTSVWT	R	lb	F10.0	11-20	Total weight of the stage
	3	FUEL	R	lb	F10.0	21-30	Total fuel available for the stage
	4	TSVLN	R	ft	F10.0	31-40	Length of the stage
L+1	1	NTSVS	I	-	Free	-	Number of IUS vehicles to be used ($NTSVS \leq 10$)
L+2	1	N	I	-	Free	-	Number of stages on the IUS ($1 \leq N \leq S$)
	2	$NUNQS(NSTVS_1 1)$	I	-	Free	-	First stage identified by the first set of cards
	3	$NUNQS(NSTVS_1 2)$	I	-	Free	-	Second stage number
	N+1	$NUNQS(NSTVS_1 N)$	I	-	Free	-	Last stage number
	N+2	YRAVAL	I	-	Free	-	A two-digit number which represents the year of availability of this vehicle

TABLE II.- CONTINUED

The second set of cards pertains to the LUS vehicles. The order in which the data are input is the order each LUS is considered. For simplicity, the next card in the sequence is denoted as "k".

Card	Word	Symbol	Type	Units	Format	Column	Description
k	1	N1	I	-	Free	-	Number of LUS vehicles to be input
k+1	1	TUGLN	R	ft	F10.3	1-10	Length of the LUS
	2	TUGWT	R	lb	F10.3	11-20	Weight of the LUS
to	3	TUGCAP	R	lb	F10.3	21-30	Capacity of the LUS
	4	TUGISP	R	sec	F10.3	31-40	Specific impulse of the LUS
k+N1	5	TUGTYP	I	-	I2	44-45	The LUS type =1, expendable =3, reusable
	6	YRAVAL	I	-	I2	47-48	First year available for the LUS
k+N1+1	1	NUMPL	I	-	Free field	-	Number of payloads in the model
	2	MKS	I	-	Free	-	A flag specifying the internal units of the payload model =1, the units are in mks =2, the units are in fps

TABLE II.- CONTINUED

The rest of the cards are identified in sets of three and identify individual payload characteristics.

Card	Word	Symbol	Type	Units	Format	Column	Description
1	1	NUMB	A	-	2A6	4-15	Payload alphanumeric identification label
	2	NDISP	A	-	2A6	16-27	Payload discipline
	3	NAME	A	-	6A6	28-63	Payload description
	4	LEN	R	ft	F5.0	64-68	Total payload length, including the pallet and/or lab
	5	WT	R	lb	F6.0	69-74	Total weight of the payload at lift-off, including the pallet and/or lab, if applicable
	6	WT1	R	lb	F6.0	75-80	Total weight of the payload at landing, including the pallet and/or lab, if applicable
2	1	DIAM	R	ft	F4.1	4-7	Diameter of the payload
	2	HA	R	n.mi.	F9.0	8-16	Desired circular altitude
	3	INCL	R	deg	F5.1	17-21	Desired inclination
	4	C3	R	ft^2/s^2	F5.0	22-26	C3 energy, this number will be multiplied by 100,000
	5	PMT	I	-	I2	27-28	Payload mission type flag =1, attached =2, servicing =3, deploy =4, retrieved

TABLE II.- CONTINUED

Card	Word	Symbol	Type	Units	Format	Column	Description
	6	FLTPYR	I	-	13I3	29-67	Flight frequency for 1979 to 1991. Each word contains a flag and denotes the number of times a payload goes up and/or down in a given year. The word is entered as XYZ, where X=1 The up and down trips for this payload can be combined on a flight. X=2 The up and down trips for this payload cannot be combined on a flight. X is ignored if the mission type is 1 or 2, or if the payload is always deployed or retrieved. In these situations, X is set to zero (or blank). If X is nonzero, Y is the number of deployments ($Y \leq 9$) and Z is the number of retrievals. If X is zero, YZ is the number of deploys, retrieves, sorties, or services.
	7	IRPT	I	-	I3	68-70	A flag which indicates the repeat conditions of a payload

TABLE II.- CONTINUED

Card	Word	Symbol	Type	Units	Format	Column	Description
							=0 Payloads to be repeated in a given year cannot be flown on the same flight. =1 Payloads flown can be repeated in a given year on the same flight.
	8	PLDUR	R	hr	F3.1	71-73	Desired time on-orbit
	9	OPTIME	R	hr	F3.1	74-76	Nominal duration of payload operation/day of time onboard the orbiter
	10	IFREQ	I	-	I3	77-79	Number of hrs/day the payload is operated while onboard the orbiter
	11	MODE	I	-	I1	80	Preferred delivery mode (attached payloads only) =1 lab =2 pallet =3 lab and pallet
3	1	RCS	R	lb	F6.1	4-9	Reaction Control System (RCS) fuel requirements based on individual payload requirements

TABLE II.- CONCLUDED

Card	Word	Symbol	Type	Units	Format	Column	Description
	2	OXEPS	R	lb	F6.1	10-15	Electrical Power System (EPS) O_2 requirements based on individual payload demands
	3	HEPS	R	lb	F5.1	16-20	EPS H_2 requirements based on individual payload demands
	4	CGPOS	R	ft	F4.1	21-24	Distance of the payload center of gravity from the front end of the payload
	5	FTSV	R	-	I1	25	A flag when set nonzero forces the use of a TSV

4. EXECUTION CHARACTERISTICS

4.1 RESTRICTIONS

The MPLS has the following limitations which apply to the analysis of any year under investigation:

a. The maximum number of payloads allowed on a flight is six.

b. The maximum number of single payloads which may be investigated is 200.

c. A limit of 6200 feasible missions is allowed for a specific number of payloads in a combination.

d. The maximum amount of data which may be written onto a mass storage file before the program terminates is 50 positions (3200 tracks).

e. The numbers of payloads per combination is restricted to three when both the mission type and discipline constraints are activated.

Further restrictions are specified in the subprogram documentation.

4.2 VALIDITY

Validation of the MPLS has been accomplished primarily by the comparison of results obtained from other programs and by hand calculations. Reference 2 gives a detailed explanation of the validation performed.

5. REFERENCE INFORMATION

5.1 DETAILED FLOWCHART

Figure 5-1 illustrates the subprogram interaction for the MPLS. Figure 5-2 illustrates the flow of the MPLS executive logic.

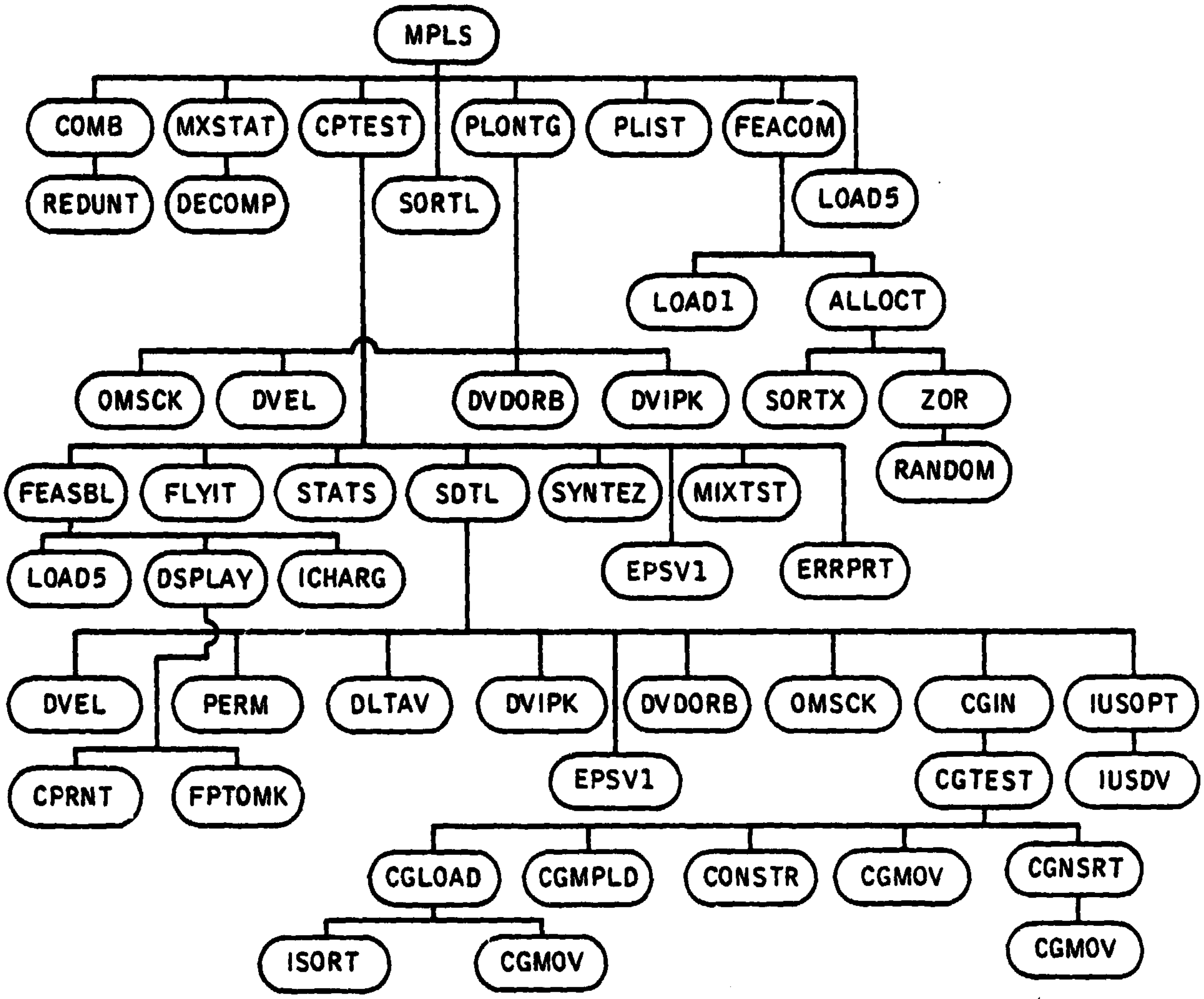


Figure 5-1.- MPLS subprogram interaction.

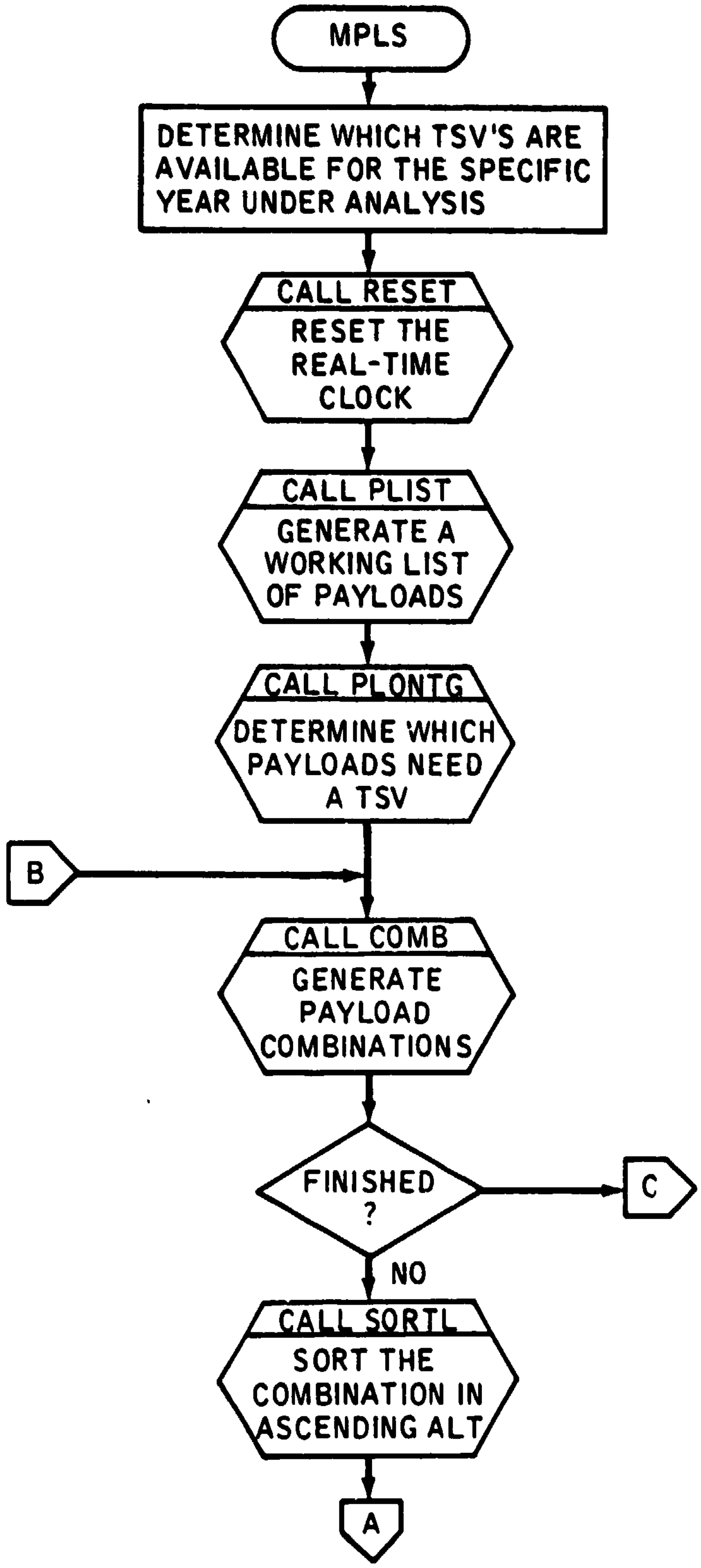


Figure 5-2.- MPLS functional flow.

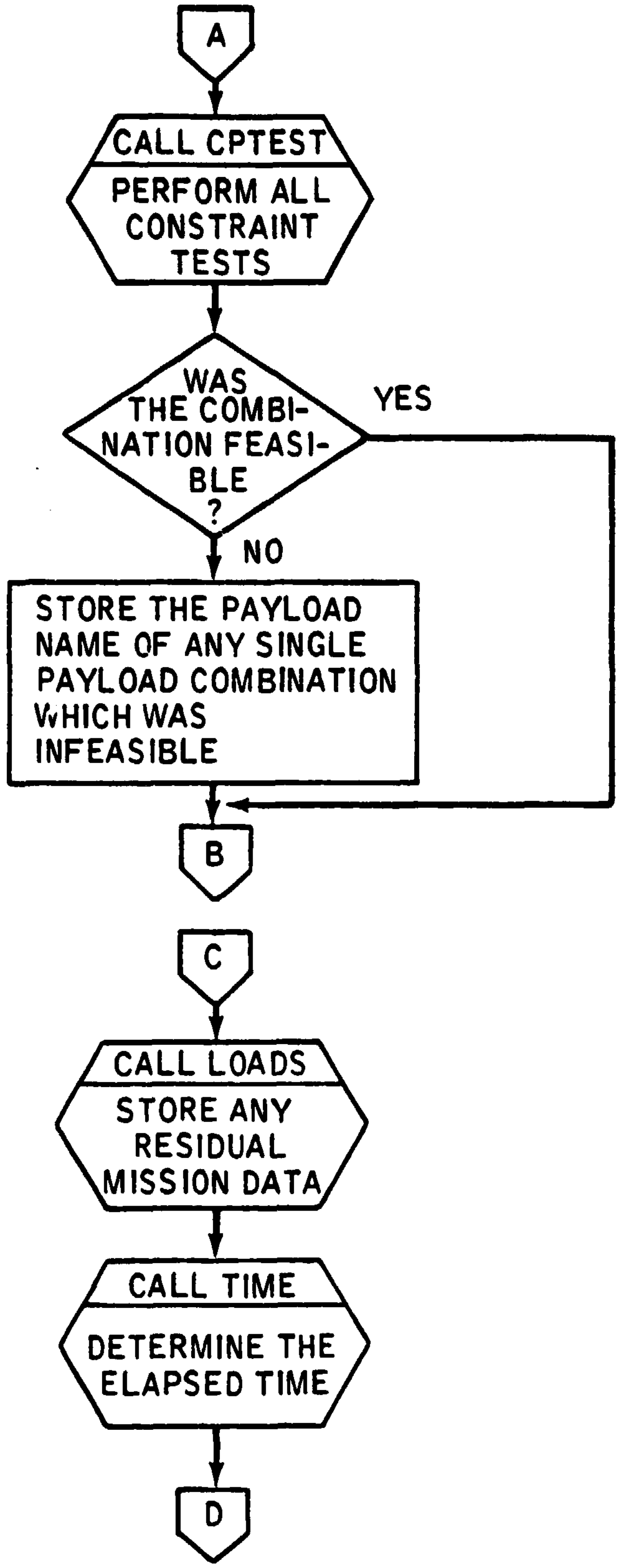


Figure 5-2.- Continued.

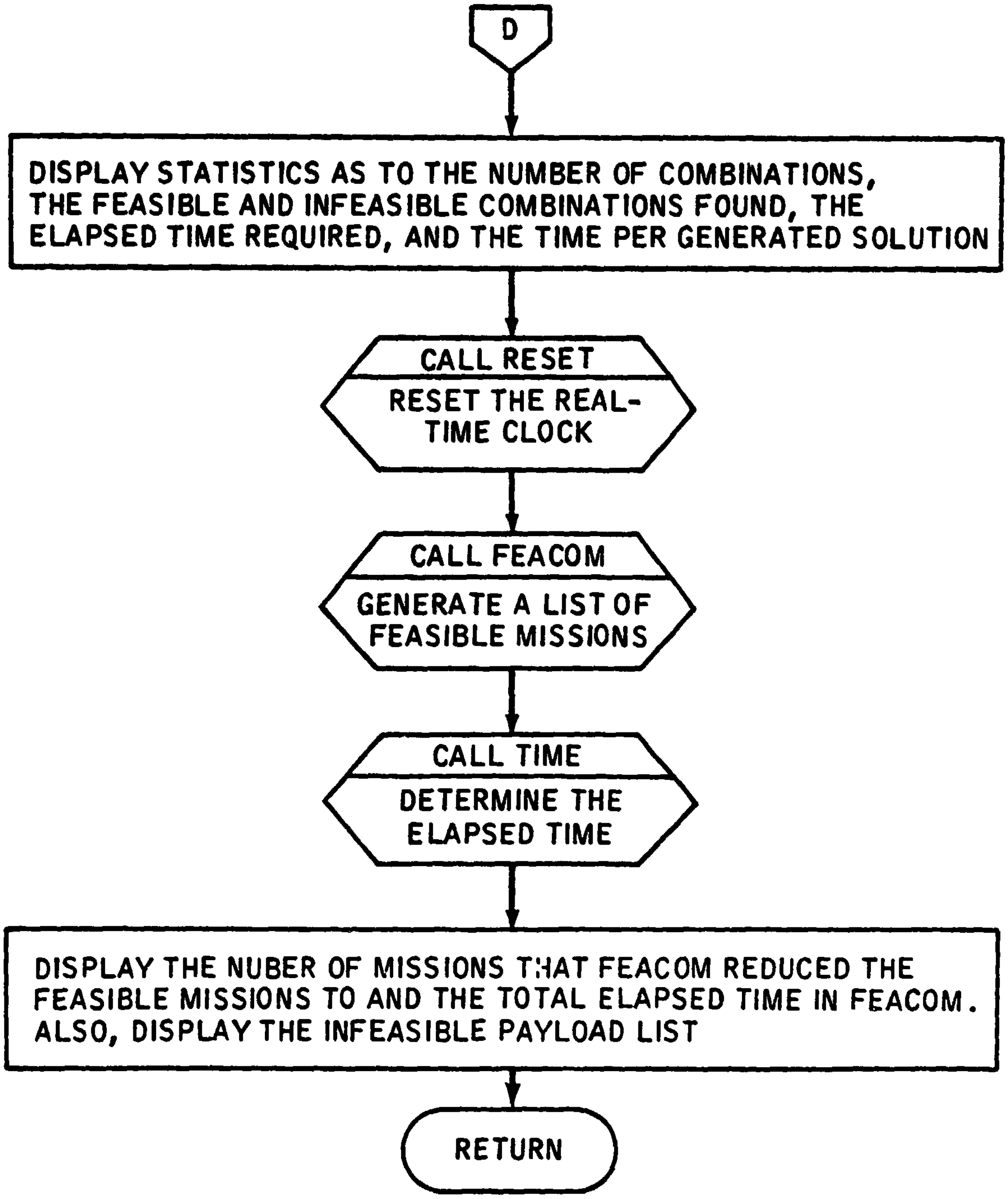


Figure 5-2.- Concluded.

5.2 VARIABLES IN LABELED COMMON

- COMMON block name: C1

Description: Labeled COMMON C1 transmits the payload model data to various subprograms of MPLS.

Storage required: 8201

Location	Name	Dimension	Type	Description
1-400	NUMB	200x2	A	Hollerith payload identification names
401-800	NDISP	200x2	A	Hollerith payload disciplines
801-2000	NAME	200x6	A	Hollerith array describing the payload
2001-2200	LEN	200	R	Array containing the total payload length including the pallet and/or lab
2201-2400	WT	200	R	Array containing the total weight of the payload at lift-off
2401-2600	WT1	200	R	Array containing the total weight of payload at landing
2601-2800	DIAM	200	R	Payload diameter
2801-3000	HA	200	R	Desired circular altitude
3001-3200	INCL	200	R	Desired orbital inclination
3201-3400	C3	200	R	C3 energy
3401-3600	PMT	200	I	Payload mission type flag: =1 attached =2 servicing =3 deploy =4 retrieve
3601-6200	FLTPYR	200x13	I	Array of the flight frequencies; each of the 13 words represents the data for an entire year starting at 1979. Each word contains a flag and denotes the number of times a payload goes up and/or down in a given year. The word is entered as XYZ where

Location	Name	Dimension	Type	Description
				X = 1 The up and down trips for this payload can be combined on a flight. X = 2 The up and down trips for this payload cannot be combined on a flight. X is ignored if the mission type is always deployed or always retrieved. In these situations, X is set to zero. If X is nonzero, Y is the number of deployments ($Y \leq 9$) and Z is the number of retrievals. If X is zero, YZ is the number of deploys, retrieves, sorties, or services.
6201-6400	IRPT	200	I	A flag which indicates the repeat conditions of a payload =0 Payloads to be repeated in a given year cannot be flown on the same flight. =1 Payloads to be repeated in a given year can be flown on the same flight.
6401-6600	PLDUR	200	R	Desired time on orbit
6601-6800	OPTIME	200	R	Nominal duration of payload operation per day of time onboard the orbiter
6801-7000	IFREQ	200	I	Number of times per day the payload is operated while onboard the orbiter
7001-7200	MODE	200	I	Preferred delivery mode for attached payloads: = 1, lab = 2, pallet = 3, lab and pallet
7201-7400	RCS	200	R	RCS fuel requirements based on individual payload requirements
7401-7600	OXEPS	200	R	EPS O_2 requirements based on individual payload demands
7601-7800	HEPS	200	R	EPS H_2 requirements based on individual payload demands

Location	Name	Dimension	Type	Description
7801-8000	CGPOS	200	R	Distance of payload center of gravity from the front end of the payload
8001-8200	FTSV	200	I	A flag when set nonzero forces the use of a TSV
8201	NUMPL	1	I	Number of payloads in the model

- COMMON block name: C2

Description: Labeled COMMON C2 transmits the EPS data to various subprograms of MPLS.

Storage required: 46

Location	Name	Dimension	Type	Description
1	EPSWT	1	R	The total EPS weight carried to insertion
2	TEPSRV	1	R	The total EPS required for payloads retrieved by the TSV and loaded onto the orbiter
3	TEPSDP	1	R	The total EPS required for payload deployed by the TSV
4	W	1	R	The weight of the orbiter excluding fuel, payloads, and consumables
5-13	DVORB	9	R	The ΔV requirements for each maneuver in the mission
14-22	DWORB	9	R	The discrete weight changes corresponding to DVORB
23-31	ORBMR	9	R	An array which is computed as $e^{\Delta V/G \cdot I_{sp}}$ where ΔV corresponds to DVORB and the terms $g \cdot I_{sp}$ denote the gravity term and specific impulse
32	NORBP3	1	I	The number of orbiter payloads plus 3
34	EPSKIT	1	I	The number of EPS kits used
35	EPSDRY	1	R	The total EPS dry tank weight
36	EDRY	1	R	The weight of one dry EPS kit
37	EO2RZ	1	R	The EPS O_2 deadweight requirements
38	EH2RZ	1	R	The EPS H_2 deadweight requirements
39	EPNKT	1	I	The number of EPS kits initially loaded and not charged to the cargo payload weight

Location	Name	Dimension	Type	Description
40	EPCRW	1	R	The crew and deadweight requirements not chargeable to the orbiter's payload weight
41	WTO2	1	R	The weight of one EPS O_2 kit, not including the tank
42	WTH2	1	R	The weight of one EPS H_2 kit, not including the tank
43	O2SL	1	R	The O_2, crew, and deadweight requirements not charged to the payload weight
44	H2SL	1	R	The H_2 deadweight requirements not charged to the payload weight
45	EPSO2	1	R	The O_2 EPS deadweight requirements charged to the payload weight
46	EPSH2	1	R	The H_2 EPS deadweight requirements charged to the payload weight

- COMMON block name: C3

Description: Labeled COMMON C3 transmits a debug print flag to various MPLS routines and loading information to the CG routines.

Storage required: 11

Location	Name	Dimension	Type	Description
1	IPRNT	1	I	A flag when set nonzero causes debug information to be printed from MPLS routines
2-7	ISORB	6	I	An array which contains the orbiter payload sequence being flown. This array is used as an index into ICORB.
8-10	ISTUG	6	I	An array which contains the TSV payload sequence being flown. This array is used as an index into ICTUG.
11	INCG	1	I	CG TEST FLAG: INCG = 0: Do not test cargo cg. = 1: Test cargo cg.

- COMMON block name: C4

Description: Labeled COMMON C4 conveys detailed information about a specific permuted flight sequence of a payload combination to various MPLS routines.

Storage required: 110

Location	Name	Dimension	Type	Description
1	IEND	1	I	
2	ITUG	1	I	
3	ITER	1	I	
4	TINCI	1	I	
5	IFLAG	1	I	
6	DVTF1	1	R	
7	JEND	1	I	
8	JTUG	1	I	
9	WEDGE	1	R	
10	YALT	1	R	
11	EOMR3	1	R	
12	WRCS	1	R	
13	EODV3	1	R	
14	DVTI1	1	R	
15	RODV2	1	R	
16	ROMR2	1	R	
17	RODV3	1	R	
18	ROMR3	1	R	
19	DVTI2	1	R	
20	POINC1	1	R	
21	POINC2	1	R	
22	DVTF2	1	R	

Location	Name	Dimension	Type	Description
23	GMSMAX	1	R	
24	JFLAG	1	R	
25-33	RDVORB	9	R	
35-43	DVORB	9	R	
45-53	ORBMR	9	R	
55-63	DWTUG	9	R	
65-74	TUGMR	10	R	
75-84	DVTUG	10	R	
85-93	DWORB	9	R	
96	YINC	1	R	
97	ITRY	1	I	
98	XINC	1	R	
99	XALT	1	R	
100	XL	1	R	
101	TALTI	1	R	
102	TWU	1	R	
103	TL	1	R	
104	TW	1	R	
105	CTUG	1	R	
106	TINCF	1	R	
107	TALTF	1	R	
108	TSVRCS	1	R	

- COMMON block name: C5

  Description: Labeled COMMON C5 transmits a working list of the payload model data to various subprograms of MPLS.

  Storage required: 903

Location	Name	Dimension	Type	Description
1-100	PAYNO	100	I	List of payloads which fly during the year under analysis
101-300	PLREP	200	I	Information for duplicate payloads where even indices indicate the number of duplicates of the payload number identified by the odd indices
301	NUMRPT	1	I	Total number of repeated payloads found for the year under analysis
302	NPLNYR	1	I	Number of payloads which fly during the year under analysis
303-502	IDOWN	200	I	Identification of payloads which are both deployed and retrieved in the same year
503	NUPDN	1	I	Number of up/down payloads found
504-703	IREP	200	I	A working storage area used to keep track of the repeated payloads as they are output
704-803	PMT1	100	I	Payload mission types defined as: = 1 attached = 2 servicing = 3 deploy = 4 retrieve = -3 an up/down payload; deployed = -4 an up/down payload; retrieved
804-903	NEEDTG	100	I	Array used to indicate whether a payload in the PAYNO list requires a TSV as: = 0, no TSV required = 1, a TSV is required

• COMMON block name: C6

Description: Labeled COMMON C6 transmits the TSV data from the payload model to various subprograms of MPLS.

Storage required: 106

Location	Name	Dimension	Type	Description
1-15	TUGLN	15	R	Array containing the lengths of the TSV's
16-30	TUGWT	15	R	Array containing the dry weight of each TSV
31-45	TUGISP	15	R	Specific impulse of each of the TSV's main engines
46-60	TUGCAP	15	R	Maximum amount of propellant for each TSV
61-75	YRAVAL	15	I	First year available for each of the TSV's
76-90	STAGE3	15	I	Array used to identify which TSV's are available for the year under analysis; a nonzero work in the array specifies that the ith TSV is available
91-105	TUGTYP	15	I	The TSV type = 1, expendable LUS = 2, expendable LUS = 3, reusable LUS
106	NTUGN	1	I	The number of TSV's used for the analysis

- COMMON block name: C7

Description: Labeled COMMON C7 transmits data pertaining to the feasible missions. Other parameters identifying the payload sequence and the number of single mission payloads that have been rejected are also transmitted.

Storage required: 111

Location	Name	Dimension	Type	Description
1	NCC	1	I	Number or combinations generated
2	M	1	I	Number of payloads in the IA array
3-8	IA	6	I	Initial payload sequence, used to index into the PAYNO array
9	NFS	1	I	Number of single payload missions rejected
10-109	NFLS	100	I	An array containing the names of the single payload missions rejected
110	NOPL	1	I	An option which causes repeated payloads to be flown on the same mission; ignored if zero
111	KOPT	1	I	A flag, set nonzero, used to inhibit permutations of a payload sequence

- COMMON block name: C8

Description: Labeled COMMON C8 transmits information pertaining to the orbiter/TSV payloads carried up and down.

Storage required: 50

Location	Name	Dimension	Type	Description
1	OPWU	1	R	Orbiter payload weight carried to insertion
2	TPWU	1	R	TSV payload weight carried to insertion
3	OPWD	1	R	Orbiter payload weight at landing
4	TPWD	1	R	TSV payload weight at landing
5	OPLU	1	R	Orbiter payload length at launch
6	TPLU	1	R	TSV payload length at launch
7	OPLD	1	R	Orbiter payload length at landing
8	TPLD	1	R	TSV payload length at landing
9	NORBPL	1	I	Number of payloads on the orbiter
10	NTUGPL	1	I	Number of payloads on the TSV
11-16	ICORB	6	I	Identification number for each orbiter payload
17-22	ICTUG	6	I	Identification number for each TSV payload
23-28	IEORB	6	I	An array containing the numerical mission type for each orbiter payload
29-34	IETUG	6	I	An array containing the numerical mission type for each TSV payload
35	IREUSE	1	I	A flag set to indicate that dedicated TSV's are required
36	NREUSE	1	I	Number of reusable TSV's that are available
37	NTUGS	1	I	Number of TSV's which meet the requirements of the mission

Location	Name	Dimension	Type	Description
38-47	LTUGS	10	I	Indices of available TSV's which meet the requirements of the mission
48	RCSWT	1	R	Total RCS fuel used in pounds
49	TOTPLU	1	R	Total orbiter and TSV payload weight at launch, not including the OMS kits or TSV dry weight
50	TOTPLD	1	R	Total orbiter and TSV payload weight at landing, not including the OMS kits or TSV dry weight

- COMMON block name: C9

  Description: Labeled COMMON C9 transmits the majority of user options to the MPLS.

  Storage required: 54

Location	Name	Dimension	Type	Description
1	YFAIL	1	I	A diagnostic message flag, when set nonzero, causes the reason a mission is infeasible to be printed
2	MM	1	I	Number of feasible missions found for the year under analysis
3-47	NMCLD	45	I	Cumulative total of occurrences for each mission type over all years processed
48	MLIST	1	I	A flag, when set nonzero, compares the payload combinations to the allowable mission types. Unmatched combinations are declared infeasible.
49	NN	1	I	Number of mission type parameters in NMCLD
50	FEASOP	1	I	A flag, when set nonzero, causes the printout of data associated with feasible combinations
51	COSTOP	1	I	Not used
52	STATOP	1	I	A flag, when set nonzero, causes the mission type occurrence statistics for feasible combinations to be generated
53	MIXDIS	1	I	A flag, when set nonzero, causes the statistics of the mission types to be printed
54	NOTAB	1	I	A flag, when set nonzero, causes SCA occurrence table to be printed

• COMMON block name: C10

Description: Labeled COMMON C10 transmits the orbiter/TSV data to various routines after it has been evaluated by subroutine. SDTL.

Storage required: 25

Location	Name	Dimension	Type	Description
1	POALT	1	R	Altitude of the initial parking orbit
2	POINC	1	R	Inclination of the initial parking orbit
3	ORBDV	1	R	Orbiter total ΔV for the mission
4	TUGDV	1	R	TSV total ΔV for the mission
5	OMSWT	1	R	Weight of the orbiter propellant needed to form the sequence
6	TUGOMS	1	R	Weight of the TSV propellant needed to form the sequence
7	TTWU	1	R	Total TSV weight at launch
8	TWD	1	R	TSV weight at landing
9	TLU	1	R	TSV length at launch
10	TLD	1	R	TSV length at landing
11	XLMAX	1	R	Length of the cargo bay used for the orbiter payloads
12-17	IFSORB	6	R	Final sequence of the orbiter payloads in the combination
18-20	IFSTUG	3	I	Final sequence of the TSV payloads
21	IFTUG	1	I	Index of the TSV used to fly this mission
22	DOALT	1	R	Altitude of the orbit at which the TSV is deployed
23	DOINC	1	R	Inclination of the orbit in which the TSV is deployed

Location	Name	Dimension	Type	Description
24	ROALT	1	R	Altitude of the TSV being retrieved
25	ROINC	1	R	Inclination of the orbit in which the TSV is to be retrieved

- COMMON block name: C11

Description: COMMON C11 Transmits the resultant output of subroutines SDTL and OMSCK to various other modules for output.

Storage required: 17

Location	Name	Dimension	Type	Description
1	PLPOMS	1	R	Theoretical maximum weight the Shuttle can carry to the initial parking orbit
2	WT2ORB	1	R	Maximum payload weight the Shuttle can carry to the initial parking orbit
3	TOTWU	1	R	Total cargo weight at launch; includes payloads, OMS kits, etc.
4	TOTWD	1	R	Total cargo weight at landing; includes payloads, OMS kits, etc.
5	PWMARU	1	R	Additional payload weight the orbiter can carry up
6	PWMARD	1	R	Additional payload weight the orbiter can carry down
7	PWMARG	1	R	Additional payload weight the orbiter can carry
8	TOTLU	1	R	Total length of the cargo at launch
9	TOTLD	1	R	Total length of the cargo at landing
10	CLMAX	1	R	Maximum length used by any permutation of the sequence
11	TOTLMX	1	R	Greatest length referenced, either TOTLU, TOTLD, or CLMAX
12	PLMARG	1	R	Additional payload cargo length the orbiter can use for this flight
13	OMSTNK	1	R	Total weight of the OMS fuel used by the orbiter
14	OMSKIT	1	R	Weight of the OMS fuel carried in the kits

Location	Name	Dimension	Type	Description
15	WTKITS	1	R	Weight of the dry OMS kits used
16	XLKITS	1	R	Length of the stacked OMS kits used
17	NOKITS	1	R	Number of OMS kits used

- COMMON block name: C12

Description: Labeled COMMON C12 transmits the mission type code related data to various subroutines.

Storage required: 55

Location	Name	Dimension	Type	Description
1-9	MSSTYP	9	I	Symbols for the nine mission types considered
10-54	MSCLCD	45	I	List of the allowable mission types for all combinations
55	NMTYP	1	I	Number of allowable mission types in MSCLCD

- COMMON block name: C13

Description: Labeled COMMON C13 transmits IUS TSV data from the payload model to various subprograms of MPLS.

Storage required: 181

Location	Name	Dimension	Type	Description
1-15	TTWT	15	R	An array containing the total weight of the IUS's being used
16-30	TISP	15	R	The specific impulse of each stage
31-45	NTSVA	15	I	The number of stages used on each IUS vehicle
46-60	FUEL	15	R	The total fuel available for each stage
61-75	TTSVWT	15	R	The total weight of each stage
76-150	NUNQS	15x5	I	An array which contains the stage numbers used for an IUS. This array is used to index into TISP, FUEL, TSVLN, and TTSVWT.
151	NTSVS	1	I	The number of IUS vehicles to be used (NTSVS $\leq$ 10)
152-166	TSVLN	15	R	The length of each stage
167-181	TLS	15	R	The total length of each IUS being used

- COMMON block name: C14

  Description: Labeled COMMON C14 transmits information pertaining to the payload combination to various subroutines in MPLS.

  Storage required: 30

Location	Name	Dimension	Type	Description
1-6	IB	6	I	Payload sequence as formed from the ICTUG and ICORB array in labeled COMMON C8
7-12	IC	6	I	Payload identification numbers in the combination; the indices are pointers into each array of COM-MCN C1
13-18	ID	6	I	Discipline mix code; the numbers are pointers into the DISPNM array of COMMON C30
19-24	IE	6	I	Payload mission type; the array is stored as a function of the IC array pointing into the PMT1 array of COMMON C5
25-30	IG	6	A	Alphanumeric payload mission type; the array is stored as a function of the absolute value of the IE array as it points into the MSSTYP array of COMMON C12

- COMMON block name: C15

  Description: Labeled COMMON C15 retains two words used to identify the launch window and whether the WTR is available.

  Storage required: 2

Location	Name	Dimension	Type	Description
1	NWTR	1	I	A flag which indicates, if nonzero, that the WTR is available; otherwise, an ETR launch is assumed
2	NYAV	1	I	An integer used to denote the year of availability for WTR launches

• COMMON block name: C17

Description: Labeled COMMON C17 transmits information to the statistical subprograms.

Storage required: 1440

Location	Name	Dimension	Type	Description
1-675	MCT	45x15	I	An array which contains the unique mission type codes for all years investigated
676-1350	MCTNO	45x15	I	The accumulative mission class codes for all years under investigation
1351-1395	LOC	45	I	An array of accumulative mission class codes
1396-1440	JORD	45	I	A working array used to store the particular class code for the year under investigation

- COMMON block name: C25

Description: Labeled COMMON C25 retains the majority of the system constraints used by the program.

Storage required: 20

Location	Name	Dimension	Type	Description
1	UPLBS	1	R	Maximum payload chargeable weight with which the orbiter can launch
2	DNLBS	1	R	Maximum payload chargeable weight with which the orbiter can land
3	SBOWT	1	R	Orbiter dry weight
4	OMSINT	1	R	Weight of the OMS fuel in the main tank
5	SPI	1	R	Specific impulse of the orbiter's OMS engines
6	BAYLN	1	R	Length of the orbiter cargo bay
7-9	FTKIT	3	R	Length of the stacked OMS kits; the first word represents one kit, the second is the accumulative length of two kits, and the third is the accumulative length of three kits
10-12	WTKIT	3	R	Weight of the stacked OMS kits; the accumulative weights of one, two, and three kits
13	RESOMS	1	R	Amount of reserve OMS fuel carried for contingencies
14	REDZDV	1	R	The ΔV required for rendezvous maneuvers
15	RCSRDZ	1	R	Weight of the RCS fuel used for each rendezvous
16	GRAV	1	R	Acceleration due to gravity
17	CORB	1	R	Acceleration of gravity times the specific impulse of the OMS engines
18	RCSCAP	1	R	Total fuel capacity of the RCS system

Location	Name	Dimension	Type	Description
19	RESRCS	1	R	Reserve RCS fuel
20	MAXKTS	1	I	MAX number of OMS kits allowed

- COMMON block name: C30

Description: Labeled COMMON C30 transmits information pertaining to the payload type and discipline to various statistics routines.

Storage required: 223

Location	Name	Dimension	Type	Description
1-20	DISPNM	20	A	A list of two-character alphanumeric discipline names allowed
21-120	MIXLST	100	I	The allowed discipline mix codes as pertain to the payload sequence
121-220	MIXCNT	100	I	Cumulative count of the payload discipline mix codes
221	MIXCHK	1	I	A flag, when set nonzero, causes the discipline mix constraint to be applied to a payload sequence
222	MNIX	1	I	Current number of mission types
223	NODIS	1	I	Number of discipline mix combinations considered

- COMMON block name: C33

Description: Labeled COMMON C33 retains information pertaining to the number of combinations generated.

Storage required: 7

Location	Name	Dimension	Type	Description
1-6	KCOMB	6	I	An array which indicates the starting numbers of the mission payload sets
7	NUM	1	I	The maximum number of payloads generated in a combination

- COMMON block name: C34

  Description: Labeled COMMON C34 transmits information within the combination generator routines to track the data as it is generated.

  Storage required: 12403

Location	Name	Dimension	Type	Description
1-6200	SETIN	6200	I	Reference set of payload combinations generated
6201-12400	NXCOMB	6200	I	Current set of payload combination generated
12401	M2	1	I	Number of combinations in SETIN
12402	K	1	I	Index of the current payload combination being generated
12403	LAST	1	I	First nonzero combination in SETIN, set to 1

- COMMON block name: C39

  Description: Labeled COMMON C39 transmits the optional print information.

  Storage required: 15

Location	Name	Dimension	Type	Description
1	MKS	1	I	A flag specifying the printed output units =1, the units are mks =2, the units are fps
2-14	LFRK	13	I	Used to print error statistics
15	NPLDS	1	I	

5.3 SUBPROGRAM DOCUMENTATION

Individual subprogram documentation is given in alphabetical order on the following pages. Functions RANDOM and ZOR and subroutine EXP are available from the MSC*LOCALIB on EXEC 8. Their documentation is found in reference 3.

SUBROUTINE ALLOCT

IDENTIFICATION

Name (Title)	- ALLOCT (Mission Allocation Routine)
Author, Date	- J. Williams, August 1975
Machine Identification	- UNIVAC 1110
Source Language	- FORTRAN V

PURPOSE

Subroutine ALLOCT generates a list of random numbers in ascending order for a specific interval defined in COMMON.

USAGE

- Calling Sequence
  CALL ALLOCAT (M, LIST, IOPT)
  Arguments:

Parameter name	In/Out	Dimension	Type	Description
M	Out	1	I	Number of words stored in LIST
LIST	Out	1	I	A list of random numbers generated in ascending order for a particular interval
IOPT	In	1	I	An initialization flag used as =1, initialization =2, causes a list of random numbers to be generated as a function of the intervals specified in COMMON C33

- Labeled COMMON used: C33

METHOD

Subroutine ALLOCT is used to generate a list of random numbers within an interval. The routine solves the problem of reducing the number of feasible missions found by the MPLS for use by the SCA. The list is then sorted into ascending order and checked for redundant numbers.

RESTRICTIONS

- Operational

  Function ZOR and subroutine SORTX are required.

ROUTINES CALLED

SORTX

CALLED BY

FEACOM

SUBROUTINE ANTONC

IDENTIFICATION

Name (Title)	- ANTONC
Author, Date	- E. Dupnick, January 1975
Machine Identification	- UNIVAC 1110
Source Language	- FORTRAN V

PURPOSE

Subroutine ANTONC converts an alphanumeric payload discipline code to an equivalent numeric code.

USAGE

- Calling Sequence
  CALL ANTONC

- Labeled COMMON used: C1, C33

RESTRICTIONS

The first two characters of the NUMB parameter of the payload model are assumed to be a discipline code for each payload. Twenty distinct codes can be stored as scanned from the payload model. More than 20 codes causes the folowing message to be printed:

WARNING: MORE THAN 20 PAYLOAD DISCIPLINES HAVE BEEN IDENTIFIED

ROUTINES CALLED

CALLED BY

MAIN

SUBROUTINE CGIN

IDENTIFICATION

Name (Title) - CGIN (CG Initialization Routing)
Author, Date - J. Williams, April 1976
Machine Identification - UNIVAC 1110
Source Language - FORTRAN V

PURPOSE

SUBROUTINE CGIN provides an interface between subroutine SDTL and the CG model.

USAGE

- Calling Sequence

CALL CGIN (LMODE, KTUGS, $)
Arguments:

Parameter name	In/Out	Dimension	Type	Description
LMODE	In	I	J	A dummy flag which is set to unity to ensure a proper routine interface from SDTL
KTUGS	In	1	I	Not used
$N	-	-		The statement in the calling program to which control is transferred if an error occurs

- Labeled COMMON used: C1, C3, C6, C7, C8, C10, C11, C12, C25

METHOD

Subroutine CGIN is used to initialize the arguments for a call to subroutine CGTEST. If it has been determined by CGTEST that the mission failed, then an error return is made.

ROUTINES CALLED

CGTEST

CALLED BY

STDL

SUBROUTINE CGLOAD

IDENTIFICATION

Name (Title)	- CGLOAD (Initial Loading of Payloads in Cargo Bay for CG Model)
Author, Date	- E. H. Perrenot, February 1976
Machine Identification	- UNIVAC 1110
Source Language	- FORTRAN V

PURPOSE

Subroutine CGLOAD creates the initial loading sequence of payloads in the Shuttle cargo bay prior to the mission events.

USAGE

- Calling Sequence
  CALL CGLOAD (NPLDS, ITNL, ISEQ, NTUGPL, ITUG, IPMT, PLDWT, NOMSKT, NSEQ)
  Arguments:

Parameter name	In/Out	Dimension	Type	Description
NPLDS	In	1	I	The number of payloads involved in the candidate mission
ITNL	In	1	I	=0, no roundtrip payloads in ISEQ requiring a tunnel =n (≠), payload number n in ISEQ requires a tunnel
ISEQ	In	Dimensioned in calling program	I	The identification numbers of the payloads involved in the mission (in the order flown)
NTUGPL	In	1	I	The number of payloads in ISEQ that require a third stage
ITUG	In	Dimensioned in calling program	I	The identification numbers of the payloads requiring a third stage

Parameter name	In/Out	Dimension	Type	Description
IPMT	In	Dimensioned in calling program	A	Array containing payload mission types for the payloads in ISEQ: A = attached, S = service, D = deploy, R = retrieve
PLDWT	In	Dimensioned in calling program	R	The weights of the payloads in the payload model plus adapter weights, if required
NOMSKT	In	1	I	The number of OMS kits on board for this mission
NSEQ	Out	Dimensioned in calling program	I	The sequence of payloads as loaded (front to back of the cargo bay)

- Labeled COMMON used: None

METHOD

Subroutine CGLOAD examines all payloads on the candidate mission with regard to payload mission type. If the payload is to be retrieved, whether by the Shuttle or by a third stage, it is not considered for loading. All payloads are ordered in the cargo bay from front to rear by increasing weight. The exceptions are that a third stage and its payloads are loaded in the rear and that other payloads are grouped into deploys and round trips, the heavier group loaded in back of the other. In the case of the third stage, it is loaded to the rear of the bay with its payloads stacked immediately in front in the reverse order of deployment. A flag is set in the eighth word of the arrray NSEQ if OMS kits are present. These kits will be loaded against the rear wall of the cargo bay.

Example:

Flight sequence: (1) deploy third stage (TS) with payloads A and B (B to be deployed first), (2) retrieve C, (3) round trip D, (4) deploy E, and (5) deploy F.

Weights - D - 1500 lb
E - 700 lb
F - 600 lb

Order in cargo bay, front to rear: F, E, D, B, A, TS

RESTRICTIONS

- Operational

  Subroutine ISORT is required.

ROUTINES CALLED

CGMOV, ISORT

CALLED BY

CGTEST

SUBROUTINE CGMOV

IDENTIFICATION

Name (Title)	- CGMOV (Shift Payloads in Array)
Author, Date	- E. H. Perrenot, February 1976
Machine Identification	- UNIVAC 1110
Source Language	- FORTRAN V

PURPOSE

Subroutine CGMOV shifts elements in an array when an element is being inserted in the array.

USAGE

- Calling Sequence
  CALL CGMOV (IBUF, IDIM, I, J, K)
  Arguments:

Parameter name	In/Out	Dimension	Type	Description
IBUF	In/Out	Dimensioned in calling program	I	Array
IDIM	In	1	I	Dimension of IBUF
I	In	1	I	The word in IBUF where the shift is to begin
J	In	1	I	The number of places (words) to shift the array
K	In	1	I	≤ 0, shift to the left; 0, to the right

- Labeled COMMON used: None

Method

Subroutine CGMOV uses another array to store words from IBUF instead of actually shifting them in IBUF itself. It tests for zeroed words and does not include them in the shift. Thus, if the calling arguments specify a two-word shift to the right, the first word indicated in IBUF is indeed moved two words to the right, but the shifting process will be terminated if two zeroed words are encountered.

ROUTINES CALLED

CALLED BY

CGLOAD, CGNSRT, CGTEST

SUBROUTINE CGMPLD

IDENTIFICATION

Name (Title)	- CGMPLD (Multipayload Center-of-Gravity Computation)
Author, Date	- E. H. Perrenot, February 1976
Machine Identification	- UNIVAC 1110
Source Language	- FORTRAN V

PURPOSE

Subroutine CGMPLD computes the CG for a group of payloads in the Shuttle cargo bay.

USAGE

- Calling Sequence
  CALL CGMPLD (IPLD, PLDWT, PLDLEN, PLDCG, TUGWT, TUGLEN, F, OMSWT, OMSLEN, D, NSEQXT, TWT, WTXLEN, CG)
  Arguments:

Parameter name	In/Out	Dimension	Type	Description
IPLD	In/Out	Dimensioned in calling program	I	Array containing identification numbers of payloads
PLDWT	In	Dimensioned in calling program	R	The weights of the payloads in the payload model plus adapter weights, if required
PLDLEN	In	Dimensioned in calling program	R	Array of payload lengths, corresponding to PLDWT
PLDCG	In	Dimensioned in calling program	R	Array of payload CG's corresponding to PLDWT
TUGWT	In	1	R	The dry weight of third stage,if used
TUGLEN	In	1	R	The length of the third stage
F	In	1	R	Weight of the fuel on board the third stage

Parameter name	In/Out	Dimension	Type	Description
OMSWT	In	1	R	Total weight of OMS kits, if used
OMSLEN	In	1	R	Length of OMS kits
D	In	1	R	The distance (in feet) from the front of the cargo bay to the first payload in the cargo bay, a function of CG constraints in previous mission phases (events)
NSEQXT	In/Out	Dimensioned in calling program	I	Array containing the amount of available space in the cargo bay created by previous payload deployments
TWT	Out	1	R	Total weight of payloads in the cargo bay (plus third stage and fuel, if applicable)
WTXLEN	Out	1	R	For all payloads in the cargo bay, a summation of the following (for each payload): the sum of payload CG and total distance from the front of the bay multiplied by payload weight, used in computing CG of retrieval payloads loaded in the rear of the cargo bay
CG	Out	1	R	The CG for the group of payloads in IPLD, expressed in feet from the front of the cargo bay

- Labeled COMMON used: None

METHOD

- Model

Subroutine CGMPLD calculates the distance from the front of the cargo bay to the front of each payload in the bay. This distance includes the sum of the lengths of payloads in front of it as well as the extra space left by deployed payloads. This length is used in the basic CG equation:

$$CG = \frac{W_1(d + C_1) + \sum_{i=2}^{n} W_i(d + L_i + C_i)}{\sum_{i=i}^{n} W_i}$$

where

W_i = Weight of ith payload in the cargo bay
d = Distance from the front wall to the first payload
C_i = The CG of ith payload in the cargo bay
L_i = Total length of the payloads and reserved gaps between payloads

ROUTINES CALLED

CALLED BY

CGTEST

SUBROUTINE CGNSRT

IDENTIFICATION

Name (Title)	- CGNSRT (Inserts Payload into Gap in Cargo Bay)
Author, Date	- E. H. Perrenot, February 1976
Machine Identification	- UNIVAC 1110
Source Language	- FORTRAN V

PURPOSE

Subroutine CGNSRT checks the cargo bay for an available area in which to insert a retrieval payload after it has been determined by subroutine CGTEST that the payload will not fit in the front or rear of the bay.

USAGE

- Calling Sequence
  CALL CGNSRT (NP, ITNL, PLEN, TOTLEN, PWT, TWT, NSEQ, NSEQXT, MIND1, MAXD1, NPINB, $a, $b)
  Arguments:

Parameter name	In/Out	Dimension	Type	Description
NP	In	1	I	Identification number of payload to be inserted
ITNL	In	1	I	=0, no roundtrip payloads in ISEQ requiring a tunnel =n (>0), payload number n in ISEQ requires a tunnel
PLEN	In/Out	1	R	Length of the payload
TOTLEN	In/Out	1	R	Total length of the payloads in the cargo bay
PWT	In	1	R	Weight of the payload
TWT	In/Out	1	R	Total weight of the payloads in the cargo bay
NSEQ	In/Out	Dimensioned in calling program	I	The sequence of payloads in the cargo bay (front to back

Parameter name	In/Out	Dimension	Type	Description
NSEQXT	In/Out	Dimensioned in calling program	I	Array containing the amount of available space in the cargo created by previous payload deployments
MIND1	In/Out	1	R	The minimum distance from the front of the cargo bay to the first payload
MAXD1	In/Out	1	R	The maximum distance from the front of the cargo bay to the first payload
NPINB	In/Out	1	R	The number of payloads in the cargo bay
$a				Returns to a statement numbered a in the calling program if the payload cannot be loaded
$b				Returns to a statement numbered b in the calling program if it can be loaded

- Labeled COMMON used: None

METHOD

Subroutine CGNSRT searches the array of available space (in the form of "gaps" in the payload bay) from the rear of the cargo bay to the front to determine if the candidate retrieval payload will fit in such a location. If such a space is found, the payload number will replace the flag (-5) in the array NSEQ and the length of the payload is considered to be, for CG purposes, that of the gap in the cargo bay that it occupies. This is necessary because the other payloads in the bay cannot be moved about and "squeezed" against the new addition. Actually, the payload is loaded in the forwardmost part of the gap it occupies. When the payload is loaded, the total payload length and weight in the cargo bay is updated, as well as the number of payloads in the bay.

RESTRICTIONS

- Operational
  Subroutine CGMOV is required.

ROUTINES CALLED

CGMOV

CALLED BY

CGTEST

SUBROUTINE CGTEST

IDENTIFICATION

Name (Title) - CGTEST (Center-of-Gravity Test)
Author, Date - E. H. Perrenot, October 1976
Machine Identification - UNIVAC 1110
Source Language - FORTRAN V

PURPOSE

Subroutine CGTEST checks a given combination of payloads to assure that center-of-gravity requirements are satisfied.

USAGE

- Calling Sequence
  CALL CGTEST (ARM, WT, N, IFLAG)
  Arguments:

Parameter name	In/Out	Dimension	Type	Description
ARM	Out	Dimensioned in calling program	R	Distance in feet from the front wall of the cargo bay to the CG of the ith payload (payload group if configuration is side-by-side), i=1 through 6, front to rear
WT	In	Dimensioned in calling program	R	Weight of the ith payload (payload group), front to rear
N	In	1	I	Number of payloads (payload groups) in the cargo bay
IFLAG	Out	1	I	CG status flag: -1 indicates the CG of the payload combination is to the front of the most forward allowable CG (fails); 0 indicates that the combination meets CG requirements

Parameter name	In/Out	Dimension	Type	Description
				1 indicates the CG of the payload combination is to the rear of the allowable CG envelope (fails)

METHOD

Subroutine CGTEST determines whether a given payload combination in a specific configuration meets CG requirements within the Shuttle cargo bay. First, the CG of the group of payloads in the bay is computed in the following manner:

$$CG = \frac{\sum_{i=1}^{n} A_i W_i}{\sum_{i=1}^{n} W_i}$$

where

A_i = ARM_i (see calling arguments)
W_i = WT_i (see calling arguments)
n = Number of payloads in the cargo bay

Subroutine CONSTR is then referenced to determine if the computed CG is within the allowable CG envelope. (An empty cargo bay automatically meets CG requirements.)

- Labeled COMMON used: C3

ASSUMPTIONS

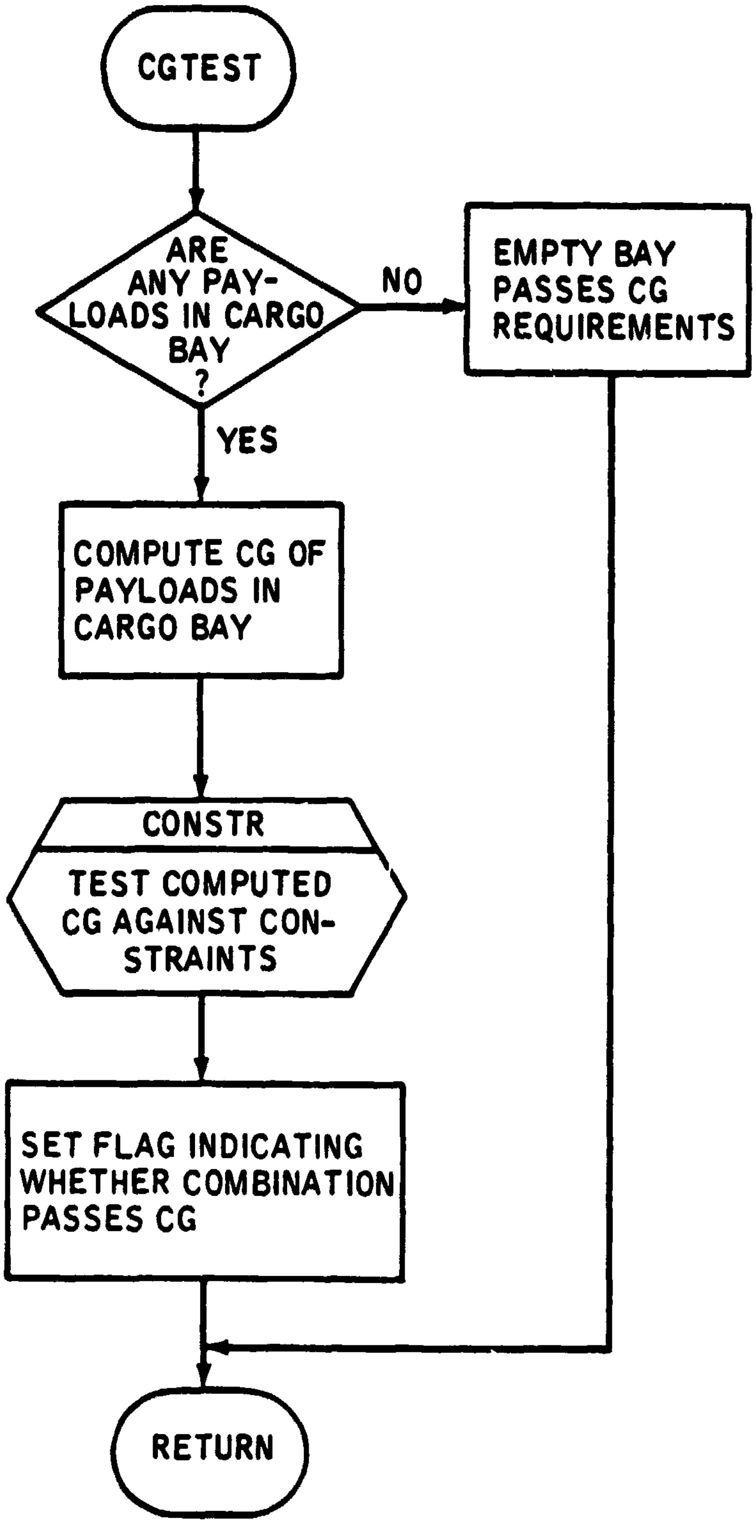
1. Payloads are initially loaded with increasing weight toward rear of cargo bay.
2. OMS kits are placed at rear of cargo bay.
3. A lab payload is loaded in front of any other payload(s).
4. Attached payloads are loaded in front of deploy payloads.
5. Center-of-gravity distances are measured from front of payload.

ROUTINES CALLED

CGLOAD, CGMOV, CGMPLD, CGNSRT, CONSTR

CALLED BY

CGIN


CGTEST
ARE ANY PAY-LOADS IN CARGO BAY ?
NO
EMPTY BAY PASSES CG REQUIREMENTS
YES
COMPUTE CG OF PAYLOADS IN CARGO BAY
CONSTR
TEST COMPUTED CG AGAINST CON-STRAINTS
SET FLAG INDICATING WHETHER COMBINATION PASSES CG
RETURN

SUBROUTINE COMB

IDENTIFICATION

Name (Title) - COMB (Combination Generator)
Author, Date - J. M. Williams, August 1975
Machine Identification - UNIVAC 1110
Source Language - FORTRAN V

PURPOSE

Subroutine COMB generates the possible combinations for MPLS; the feasible combinations are kept and used to generate the remaining combinations.

USAGE

- Calling Sequence
  CALL COMB ($N, NPLNYR, MAXPL, IPASS, IMM, K1, IJ)
  Arguments:

Parameter name	In/Out	Dimension	Type	Description
$N	In	-	-	Statement number in the calling program to which control is transferred when all the combinations have been generated
NPLNYR	In	1	I	Number of single mission payloads
MAXPL	In	1	I	Maximum number of payloads allowed on a combination
IPASS	In/Out	1	I	A flag, when set to zero, causes the single mission payloads to be generated; if nonzero, the remaining combinations will be generated
IMM	Out	1	I	Number of payloads in the current combinations
K1	In/Out	1	I	Index number of the current combination; if the combination is rejected, this number is decremented by 1

Parameter name	In/Out	Dimension	Type	Description
IJ	Out	6	I	The payload combination. The array is used as a pointer into COMMON C1. The IJ array is stored in the IC array of COMMON C14

- Labeled COMMON used: C7, C34

METHOD

Subroutine COMB generates the combination of payloads for MPLS evaluation by using the results of previous combinations. The combinations are generated in sets; each set is generated from the previous set. This scheme allows only those combinations to be generated as a result of successful feasible combinations. For example:

Initial single payloads	Doubles	Triples
1	12*	235
2	13*	
3	14	
4	15*	
5	23	
	24*	
	25	
	34*	
	35*	
	45*	

*Indicates a rejected combination of payloads.

RESTRICTIONS

- Operational
  A maximum of 4000 combinations may be generated for each set with M payloads.

ROUTINES CALLED

REDUNT

CALLED BY

MPLS

SUBROUTINE CONSTR

IDENTIFICATION

Name (Title)	- CONSTR (Center-of-Gravity Constraint)
Author, Date	- E. H. Perrenot, February 1976
Machine Identification	- UNIVAC 1110
Source Language	- FORTRAN V

PURPOSE

Subroutine CONSTR supplies minimum and maximum allowable centers of gravity for a given total payload weight.

USAGE

- Calling Sequence
  CALL CONSTR (TWT, CMIN, CMAX)
  Arguments:

Parameter name	In/Out	Dimension	Type	Description
TWT	In	1	R	Total weight of payloads in the cargo bay (plus third stage and fuel, if applicable)
CMIN	Out	1	R	Minimum CG constraint (in feet from the front of the cargo bay)
CMAX	Out	1	R	Maximum CG constraint

METHOD

CONSTR constructs a table of weights from 500 to 65,000 lb in 1500-lb increments. A payload combination weight is then checked against this table and the two weight values, which are found on either side of the reference weight, are used to locate the minimum and maximum CG constraints for each weight. The minimum and maximum CG constraints are then calculated for the reference weight by linear interpolation.

ROUTINES CALLED

None

CALLED BY

CGTEST, READIN

Reserved page

SUBROUTINE CPRNT

IDENTIFICATION

Name (Title)	- CPRNT (Detailed Print Routine)
Author, Date	- J. Williams, March 1976
Machine Identification	- UNIVAC 1110
Source Language	- FORTRAN V

PURPOSE

Subroutine CPRNT is a special purpose print routine used to display detailed information pertaining to feasible combinations.

USAGE

- Calling Sequence
  CALL CPRNT

- Labeled COMMON used: C2, C8, C10, C11, C25

METHOD

Subroutine CPRNT displays information pertaining to the EPS, the missions ΔV requirements, and discrete weight changes.

ROUTINES CALLED

None

CALLED BY

DISPLY

SUBROUTINE CPTEST

IDENTIFICATION

Name (Title)	- CPTEST (Combination Preliminary Testing Routine)
Author, Date	- J. Williams, July 1975
Machine Identification	- UNIVAC 1110
Source Language	- FORTRAN V

PURPOSE

Subroutine CPTEST performs the flight sequence independent tests for a given payload combination.

USAGE

- Calling Sequence
  CALL CPTEST ($N, IYEAR)
  Arguments:

Parameter name	In/Out	Dimension	Type	Description
$N	-	-	-	The statement in the calling program to which control is transferred if an error is indicated
IYEAR	In	-	I	The year that the combination of payloads is to be flown

- Labeled COMMON used: C1, C2, C5, C6, C7, C8, C9, C10, C12, C14, C25, C30, C39

METHOD

Subroutine CPTEST performs flight sequence independent tests and initializes parameters for use in statistical analysis and flight dependent tests. The method is as follows:

1. Payload combination conformity constraint tests are made. The first constraint test is a function of redundant payloads on the same flight; the remaining conformity tests are a function of mission type and the payload discipline mix.

2. Orbiter conformity constraint tests are made to determine if each combination is within orbiter limits. The orbiter limits are a

function of the cargo bay length, the maximum weight allowed on the TSV, the number of TSV payloads, and number of payloads which require a dedicated TSV.

3. Subroutine SDTL is called to perform the flight sequence dependent tests.

4. Subroutines STATS and MIXTST are called to tabulate data for statistical analysis of payload mission types and discipline mix.

RESTRICTIONS

- Operational
  K6 = 1 - Mission Type
  = 2 - Total Wt up
  = 3 - Total Wt down
  = 4 - Need Dedicated Tug
  = 5 - NTUGS 73
  = 6 - No Tugs meet Wt & length
  = 7 - Total length up
  = 8 - Total length down
  = 9 - Discipline Mix
  = 10 - Seq. Test
  = 11 - RCS Capacity
  = 12 - Redundant Payload
  = 13 - INCL Range > $.5^{\circ}$

ROUTINES CALLED

DECOMP, ERRPRT, FEASBL, FLYIT, MIXTST, SDTL STATS

CALLED BY

MPLS

SUBROUTINE DECOMP

IDENTIFICATION

Name (Title)	- DECOMP (Decompose) SYNTEZ (Compose)
Author, Date	- J. Williams, August 1975
Machine Identification	- UNIVAC 1110
Source Language	- FORTRAN V

PURPOSE

Subroutine DECOMP is used to separate each digit of a multidigit integer number into a list of single digit integer words; entry point SYNTEZ is used to generate a single integer word from an input list.

USAGE

- Calling Sequence
  CALL DECOMP (N, M, NOYES, NARRAY)
  CALL SYNTEZ (N, M, NOYES, NARRAY)
  Arguments:

Parameter name	In/Out	Dimension	Type	Description
N	In/Out	1	I	Number of words in NARRAY
M	In	1	I	Number of digits each word (NARRAY) will occupy in NOYES
NOYES	In/Out	1	I	An integer formed from the integer list NARRAY
NARRAY	In/Out	6	I	A list of integers which range in value from 1 to 9

METHOD

Subroutine DECOMP uses the MOD function to separate each integer digit from the word NOYES. Entry point SYNTEZ forms the integer word from a list by first sorting the integers into descending order.

RESTRICTIONS

- Operational
  The largest integer that can be decoded is nine digits.

ROUTINES CALLED

SORT

CALLED BY

CPTEST, INLIST, MIXTST, PRTLST

SUBROUTINE DISPLY

IDENTIFICATION

Name (Title)	- DISPLY (Display Routine)
Author, Date	- J. Williams, August 1975
Machine Identification	- UNIVAC 1110
Source Language	- FORTRAN V

PURPOSE

Subroutine DISPLY prints the payload characteristics and associated information for a feasible mission.

USAGE

- Calling Sequence
  CALL DISPLY (MM, M, IC, IB, NAME1, LAUNCH,
  * NTUGPL, TUG, ITUG, NOKITS, PWMARG, PCTUSE, FLOAD,
  * POINC, POALT, ALT, XINC, TOTLU, TOTLD, TOTWU,
  * TOTWD, TUGDV, ORBDV, NOYES, IDENT)
  Arguments:

Parameter name	In/Out	Dimension	Type	Description
MM	In	1	I	Flight number
M	In	1	I	Numbers of payloads on the flight
IC	In	6	I	Payload numbers which represent the combination
IB	In	6	I	Payload numbers ordered with respect to TSV and orbiter events
NAME1	In	6	I	Numerical mission type
LAUNCH	In	1	I	An integer set to 1 or 2 to indicate an ETR or WTR launch
NTUGPL	In	1	I	Number of TSV payloads
TUG	In	1	I	TSV number used in this mission
ITUG	In	6	I	Payload numbers of the TSV payloads

Parameter name	In/Out	Dimension	Type	Description
NOKITS	In	1	I	Number of OMS kits used
PWMARG	In	1	R	Additional payload weight the orbiter could carry on this flight
PCTUSE	In	1	R	Percentage of the first OMS kit used
FLOAD	In	1	R	Load factor
POINC	In	1	R	Inclination of the first orbit
POALT	In	1	R	Altitude of the first orbit
ALT	In	1	R	Altitude of each payload in the combination
XINC	In	1	R	Inclination of each payload in the combination
TOTLU	In	1	R	Total payload length at launch
TOTLD	In	1	R	Total payload length at landing
TOTWU	In	1	R	Total payload weight at launch
TOTWD	In	1	R	Total payload weight at landing
TUGDV	In	1	R	Total TSV ΔV
ORBDV	In	1	R	Total orbiter ΔV
NOYES	In	1	I	Not used
IDENT	In	12	A	A list of two-word payload names in the combination

METHOD

Subroutine DISPLY prints a table of data related to each feasible combination; the subroutine contains no special computations.

RESTRICTIONS

- Operational

ROUTINES CALLED

CPRNT

CALLED BY

FEASBL, FLTSUM, FNDFLT

FUNCTION DLTAV

IDENTIFICATION

Name (Title)	-	DLTAV (ΔV function 1) DVEL (ΔV function 2) DVIPK (ΔV from insertion to the parking orbit) DVDORB (deorbit ΔV to landing)
Author, Date	-	J. Williams, August 1975
Machine Identification	-	UNIVAC 1110
Source Language	-	FORTRAN V

PURPOSE

FUNCTION DLTAV and its three entry points are used to compute the ΔV requirements for the MPLS. DLTAV computes the ΔV between orbits for the ith and jth payloads. DVEL computes the ΔV between orbits for two payloads by specifying their altitudes and inclinations. DVIPK computes the ΔV from insertion to the first parking orbit and DVDORB computes the deorbit ΔV to landing.

USAGE

- Calling Sequence

```
X = DLTAV (I,J)
X = DVEL (E,B,C,D)
X = DVIPK (HH)
X = DVDORB (H,XIN)
```

Arguments:

Parameter name	In/Out	Dimension	Type	Description
I	In	1	I	Index of the ith payload
J	In	1	I	Index of the jth payload
E	In	1	R	Orbital altitude of the ith payload
B	In	1	R	Orbital altitude of the jth payload
C	In	1	R	Inclination of the ith payload

D	In	1	R	Inclination of the jth payload
HH	In	1	R	Altitude of the first parking orbit
H	In	1	R	Altitude of the last parking orbit before landing
XIN	In	1	R	Inclination of the last parking orbit before landing

- Labeled COMMON used: C1

METHOD

Function DLTAV and its three entry points calculate the ΔV requirements for the MPLS. The functions use a Hohmann transfer algorithm at the insertion and orbital phases of the mission; the deorbit ΔV is computed using an empirical equation.

1. ENTRY DVIPK

$$a = (r_{p_i} + r_{a_i})/2$$

where

a = Semimajor axis

r_{p_i} = Perigee radius at insertion

r_{a_i} = Apogee radius at insertion

$$V_{p_i} = \sqrt{\frac{\mu r_{a_i}}{a_1 r_{p_i}}}$$

where

μ = Gravitational constant

V_{p_i} = Perigee velocity of the insertion orbit

$$a_1 = (r_{p_i} + r_{t_e})/2$$

where

a_1 = The semimajor axis of the transfer ellipse

r_{t_e} = The apogee radius of the transfer ellipse

$$V_p = \sqrt{\frac{\mu\, r_{t_e}}{a_1\, r_{p_i}}}$$

where V_p = The perigee velocity of the transfer ellipse

$$V_a = \sqrt{\frac{\mu\, r_{p_i}}{a_1\, r_{t_e}}}$$

where V_a = The apogee velocity of the transfer ellipse

$$V_c = \sqrt{\mu / R_{t_e}}$$

where V_c = The circular velocity of the transfer ellipse

and $$\Delta V_1 = \sqrt{(V_p - V_{p_i}) + (V_c - V_a)}$$

where ΔV_1 = The delta velocity required to transfer from insertion to the initial parking orbit

2. Function DLTAV and entry point DVEL

These two functions perform identical tasks and differ only in their calling arguments. Function DLTAV computes the ΔV to transfer from the ith payload orbit to the jth payload orbit. DVEL permits direct entry of orbital altitudes and inclinations; the transfer goes from the E to B altitude.

$$a = (r_{a_1} + r_{a_2})/2$$

where

r_{a_1} = The apogee radius of the initial orbit (E)

r_{a_2} = The apogee radius of the final orbit (B)

$$\Delta i = i_1 - i_2$$

where

i_1 = The inclination of the initial orbit

i_2 = The inclination of the final orbit

If the altitude of the initial orbit is greater than the final orbit, then

$$\Delta i_a = \Delta i$$

and $$\Delta i_b = 0$$

where

Δi_a = The change in inclination at the first impulse point

Δi_b = The change in inclination at the second impulse point

If the altitude of the initial orbit is less than or equal to the final orbit, then

$$\Delta i_a = 0$$

and $$\Delta i_b = \Delta i$$

Then $$V_{c1} = \mu/r_{a1}$$

and $$V_{e1} = \frac{\mu\, r_{a2}}{a\, r_{a1}}$$

C-2

where

V_{c_1} = The circular velocity at the first impulse point

V_{e_1} = The elliptical velocity at the first impulse point

The delta velocity of the first impulse is computed as

$$\Delta V_1 = \sqrt{(V_{c1} - V_{e1})^2 + 4V_{c1} V_{e1}\left[\sin\left(\frac{i_1}{2}\right)\right]^2}$$

The circular and elliptical velocity of the second impulse point is computed as

$$V_{c2} = \frac{\mu}{r_{a2}}$$

$$V_{e2} = \frac{\mu\, r_{a1}}{a\, r_{a2}}$$

where

V_{c_2} = The circular velocity of the second impulse point

V_{e_2} = The elliptical velocity of the second impulse point

The total velocity change required for the transfer is computed as

$$\Delta V = \Delta V_1 + \sqrt{(V_{c2} - V_{e2})^2 + 4V_{c2} V_{e2}\left[\sin\left(\frac{i_2}{2}\right)\right]^2}$$

3. Entry DVDORB computes the ΔV required for the deorbit maneuver. The ΔV is computed as a function of inclination and altitude.

For altitudes (H) at the last orbit less than 140 n.mi.

$$\Delta V_1 = -0.05 \cdot H + 256.0$$

For altitudes greater than 140 but less than 457 n.mi.

$$\Delta V_1 = 1.4 \cdot H + 52.0$$

For altitudes greater than 457 n.mi.

$$\Delta V_1 = 1.443 \cdot H + 32.0$$

If the inclination of the last orbit is greater than or equal to 28.5^0, the ΔV as computed is output. Assuming the inclination is less than 28.5^0, then

$$\Delta V = \Delta V_1 + 2V_c \sin \frac{C_1 - i}{2C_2}$$

where

C_1 = 28.5^0

C_2 = 57.29578, a conversion factor used to convert degrees to radians

V_c = The circular velocity at the altitude of the last orbit

The V_c parameter is computed as

$$V_c = \frac{\mu_e}{R + H \cdot C_3}$$

where

$_e$ = 1.40765392E16, gravitational constant in ft^3/sec^2

R = The radius of the earth

C_3 = 6076.11548, a conversion factor used to convert n.mi. to feet

RESTRICTIONS

- Analytical

  1. All orbital ΔV's are computed for circular orbits

  2. All plane change maneuvers are performed at the higher altitude orbit

ROUTINES CALLED

None

CALLED BY

ORBITR, PLONTG, TSV

SUBROUTINE ERRPRT

IDENTIFICATION

Name (Title) - ERRPRT (Error Printing Routine)
Author, Date - J. Williams, August 1975
Machine Identification - UNIVAC 1110
Source Language - FORTRAN V

PURPOSE

Subroutine ERRPRT is used to display the diagnostic messages associated with infeasible missions generated by MPLS.

USAGE

- Calling Sequence

CALL ERRPRT (KERR,MIXCOD)

Arguments:

Parameter name	In/Out	Dimension	Type	Description
KERR	In	1	I	The error number to indicate the type of constraint that was violated
MIXCOD	In	1	I	The discipline mix code

- Labeled COMMON used: C1, C7, C8, C11, C14

Block name	Input	Output
C1	3201-3400	
C7	2	
C8	10,17-22,49-50	
C11	Not used	
C14	7-18,25-30	

METHOD

Subroutine ERRPRT is a logic routine that prints a diagnostic as a function of an error number. Refer to section 3.2.2 for the diagnostic messages.

RESTRICTIONS

- Operational

  Only 12 diagnostic messages are available.

ROUTINES CALLED

None

CALLED BY

CPTEST

SUBROUTINE FEACOM

IDENTIFICATION

Name (Title) - FEACOM (Feasible Combination Routine)
Author, Date - J. M. Williams, August 1975
Machine Identification - UNIVAC 1110
Source Language - FORTRAN V

PURPOSE

Subroutine FEACOM generates a data file of feasible missions by randomly selecting missions from the feasible mission file. The reduced data is used by the SCA to form traffic models.

USAGE

- Calling Sequence

  CALL FEACOM (MM, FEASOP, NP)

  ARGUMENTS:

Parameter name	In/Out	Dimension	Type	Description
MM	In/Out	1	I	The number of feasible missions; MM is the number generated by the MPLS when input; when output, it represents the number of missions kept
FEASOP	-	1	I	Not used
NP	-	1	I	Not used

- Labeled COMMON used: C33

METHOD

Subroutine FEACOM generates a reduced list of missions from the feasible mission file for use in the SCA. The missions to be retained are selected by subroutine ALLOCT, which generates a list of missions for a specific interval. The intervals used to select the mission numbers are obtained from COMMON C33,in the array KCOMB.

RESTRICTIONS

- Operational

  FEACOM will limit the number of feasible missions to 2000.

ROUTINES CALLED

ALLOCT

CALLED BY

TABLE

SUBROUTINE FEASBL

IDENTIFICATION

Name (Title) FEASBL (Feasible Mission Output Routine)
Author, Date - J. Williams, August 1975
Machine Identification - UNIVAC 1110
Source Language - FORTRAN V

PURPOSE

SUBROUTINE FEASBL is used to output data related to a feasible mission both on mass storage and the printer.

USAGE

- Calling Sequence
  CALL FEASBL (INCR, ICNT, IYEAR, IEE)
  Arguments:

Parameter name	In/Out	Dimension	Type	Description
INCR	In/Out	1	I	A flag used to count the number of missions written to a mass storage file, set to zero before each call
ICNT	In	1	I	A flag used to indicate if repeated payloads are not on this mission; ignored if zero
IYEAR	In	1	I	The year that the combination of payloads is to be flown
IEE	In	1	I	The payload mission types for the combination of payloads

- Labeled COMMON used: C1, C5, C7, C8, C9, C10, C11, C12, C14, C15, C25, C33, C39

METHOD

Subroutine FEASBL is used to set up data related to a feasible mission for purposes of output. The following procedure is used for any feasible combinations.

1. The load factor for the orbiter is computed as the maximum of the ratio of total weight up versus the weight to orbit capability and the total weight down versus the maximum down weight allowed.

2. Function ICHARG is referenced to initialize the cost array.

3. The up/down payloads are coded for print.

4. The combination is scrutinized for repeated payloads.

RESTRICTIONS

- Operational

ROUTINES CALLED

DISPLY, FPTOMK, ICHARG

CALLED BY

CPTEST

SUBROUTINE FLYIT

IDENTIFICATION

Name (Title) - FLYIT (Compatible Payloads Routine)
Author, Date - J. Williams, August 1975
Machine Identification - UNIVAC 1110
Source Language - FORTRAN V

PURPOSE

Subroutine FLYIT determines if repeated payloads can fly on the same mission.

USAGE

- Calling Sequence
  CALL FLYIT (ICNT, ICM1, ICM2, $N, IYEAR, ICM3)
  Arguments:

Parameter name	In/Out	Dimension	Type	Description
ICNT	In	1	I	Number of unique repeated payloads found
ICM1	In	6	I	Index of the payload found
ICM2	In	6	I	Number of payloads which have been duplicated for each duplicated payload
$N	-	-		The statement number to which control is passed if an error occurs
IYEAR	In	1	I	A two-digit number which represents the year under analysis
ICM3	In	6x6	I	An array of payload mission types. Each row in the matrix represents a set of mission types for a specific redundant payload

- Labeled COMMON used: C1, C5

METHOD

Subroutine FLYIT compares the compatibility of all redundant payloads in a combination to the mission flight parameter, FLTPYP, of the payload model. The following procedure is used for any specific payload:

1. The flight frequency parameter is decoded into a flag and two parameters which represent the number of up/down payloads flown this year. The flag is used to indicate whether up/down payloads can fly together.

2. If the number of payloads of the same name exceeds three, the combination is rejected.

3. If the number of redundant payloads of the same name exceeds that allowed by the payload model, the combination is rejected.

4. If both up and down payloads of the same name are on the combinations, the flag parameter must be checked to determine if the flight is allowed.

ROUTINES CALLED

None

CALLED BY

CPTEST

SUBROUTINE FPTOMK

IDENTIFICATION

Name (Title) - FPTOMK (Foot-Pound-Second System to Meter-Kilogram-Second System)
Author, Date - H. Chang, March 1976
Machine Identification - UNIVAC 1110
Source Language - FORTRAN V

PURPOSE

Subroutine FPTOMK transforms the measurement of output data of MPLS from the English to the metric system.

USAGE

- Calling Sequence

CALL FPTOMK (POALT, ALT, TOTLN, TOTLN1, TOTWTU, TOTWTD, PLMARG, CURDV, TOTDV)

Arguments:

Parameter name	In/Out	Dimension	Type	Description
POALT	In/Out	1	R	Altitude of the initial parking orbit
ALT	In/Out	6	R	Orbital altitude of each payload
TOTLN	In/Out	1	R	Total length up
TOTLN1	In/Out	1	R	Total length down
TOTWTU	In/Out	1	R	Total weight up
TOTWTD	In/Out	1	R	Total weight down
PLMARG	In/Out	1	R	Additional payload weight the Shuttle can carry on this flight
CURDV	In/Out	1	R	Total Shuttle ΔV used
TOTDV	In/Out	1	R	Total TSV ΔV used

METHOD

SUBROUTINE FPTOMK changes the value of those variables in the calling arguments from the foot-pound-second system to the meter-kilogram-second system by multiplying by conversion constants.

ROUTINES CALLED

None

CALLED BY

FEASBL

FUNCTION ICHARG

IDENTIFICATION

Name (Title) - ICHARG (Cost Coefficient Routine)
Author, Date - J. Williams, August 1975
Machine Identification - UNIVAC 1110
Source Language - FORTRAN V

PURPOSE

Function ICHARG computes a selected cost coefficient for a specific feasible payload combination.

USAGE

- Calling Sequence

  J(I) = ICHARG (I, FLOAD)

  Arguments:

Parameter name	In/Out	Dimension	Type	Description
I	In	1	I	Index of the cost coefficient being computed
FLOAD	In	1	R	Load factor, the ratio of the total orbiter weight up to its capability, or the down weight to its capability

- Labeled COMMON used: C10, C11, C25

METHOD

Function ICHARG computes 8 of 12 integer cost coefficients of a mission for use in the SCA. The technique used allows the initialization or computation of a coefficient based on the input argument I. The function has the following meanings for I:

Value of I	Set value of each flight equal to
1:	Unity
2:	Maximum of weight load factor up or down
3:	On-orbit OMS propellant required
4:	Minimum of unused weight capability up or down
5:	Maximum of length load factor up or down

6:	Cargo weight up
7:	Cargo length up
8:	Maximum of weight load factor up or down, or length load factor up or down
9:	Product of priority of constituent payloads
10:	Sum of sharability of constituent payloads
11:	Charge factor (unavailable)
12:	Unallocated

Values for I = 9, 10 are computed in routine TABLE.

SUBROUTINE ISORT

IDENTIFICATION

Name (Title) - ISORT (Sort Array and Rearrange Additional Array(s))
Author, Date - E. H. Perrenot, November 1975
Machine Identification - UNIVAC 1110
Source Language - FORTRAN V

PURPOSE

Subroutine ISORT sorts an array in either ascending or descending order and rearranges up to three arrays in the same sequence.

USAGE

- Calling Sequence
  CALL ISORT (A, N, B, C, D, K, SWITCH)
  Arguments:

Parameter name	In/Out	Dimension	Type	Description
A	In/Out	N	R	Array to be sorted
N	In	1	I	The number of words in array A (and B, C, and/or D, if used)
B C D	In/Out	N	R	Arrays to be rearranged
K	In	1	I	0 = sort A only 1 = rearrange B 2 = rearrange B and C 3 = rearrange B, C, and D
SWITCH	In	1	I	> 0, sort will be in ascending order < 0, in descending order

METHOD

Subroutine ISORT uses a binary search technique to reorder array A. If the options for K are exercised, B, C, and/or D will be rearranged in the same sequence as A. When a test is made between two elements of the A array to determine which is larger, a switch is made depending upon whether the sort is in ascending or descending order. If a switch is made, the same respective elements in B, C, and/or D are switched.

SUBROUTINE IUSDV

IDENTIFICATION

Name (Title) - IUSDV (ISU ΔV Routine)
INSTG (Initialize IUS Vehicles)
Author, Date - J. Williams, April 1976
Machine Identification - UNIVAC 1110
Source Language - FORTRAN V

PURPOSE

Subroutine IUSDV is used to compute the ΔV for a specific IUS which was predefined by entry point INSTG.

USAGE

- Calling Sequence
  CALL IUSDV (L, WT, TDV, V)
  CALL INSTG (IOUT)
  Arguments:

Parameter name	In/Out	Dimension	Type	Description
L	In	1	I	The index of the IUS being used
WT	In	1	R	The payload weight carried on the IUS
TDV	Out	1	R	The total IUS ΔV required
V	Out	Dimensioned in calling program	R	An array of ΔV's computed for each stage of the IUS
IOUT	Out	1	I	The number of IUS vehicles which may be used

- Data In/Out

  Labeled COMMON (refer to the labeled COMMON description section):

Block name	Input	Output
C6	91-105	1-46,61-75
C13	1-166	167-181

METHOD

Entry point INSTG computes the weight and length of an IUS vehicle defined by the payload model; refer to table II. The computation is performed by a summation of all IUS stage weights and lengths defined for a specific IUS.

Subroutine IUSDV uses the total loaded weight of the IUS to compute the ΔV requirements for each stage. The equation used to compute the ΔV requirement is a form of the ideal rocket equation and is

$$\Delta V = g \cdot I_{sp} \quad \ell n \; W_I \; W_F$$

where

g = Acceleration due to gravity
I_{sp} = The specific impulse for the orbiter OMS engines
ℓn = Natural logarithm
W_I = Vehicle weight before the burn
W_F = Vehicle weight after the burn

The fuel requirement for each stage is known; therefore, the problem solution is simplified.

SUBROUTINE IUSOPT

IDENTIFICATION

Name (Title) - IUSOPT (IUS Option Routine)
Author, Date - J. Williams, April 1976
Machine Identification - UNIVAC 1110
Source Language - FORTRAN V

PURPOSE

Subroutine IUSOPT determines feasibility of a mission for a specific IUS vehicle, given the weight of the payload to be carried.

USAGE

- Calling Sequence

CALL IUSOPT (HA1, HA2, XINC1, XINC2, PYLDWT, TW, ISN, TL, TUGDV, $)

Arguments:

Parameter name	In/Out	Dimension	Type	Description
HA1	In	1	R	The altitude of the circular orbit at which the IUS and its payloads are deployed from the Shuttle
HA2	In	1	R	The altitude of the payload final circular orbit. If HA1 < 0 then HA2 is the C3 energy required rather than altitude
XINC1	In	1	R	The inclination of the IUS deployment orbit
XINC2	In	1	R	The desired inclination
PYLDWT	In	1	R	Weight of the payload
TW	Out	1	R	Weight of the IUS
ISN	In	1	I	Index of the IUS used
TL	Out	1	R	Length of the IUS
TUGDV	Out	1	R	ΔV required by the IUS

$N - A statement number in the calling program to which control is transferred when an error occurs

- Data In/Out

Labeled COMMON (refer to the labeled COMMON description section):

Block name	Input	Output
C3	1	
C13	1-15,167-181	

METHOD

Subroutine IUSOPT verifies whether a payload(s) can be flown on a specific IUS. The logic optionally allows the use of C3 energy cases or permits the user to specify altitudes and inclinations. The method is as follows:

1. Subroutine IUSDV is called to compute the IUS ΔV available.

2. The Hohmann transfer ΔV is computed.

3. The ΔV from each stage of the IUS is summed until it exceeds the transfer ΔV, or until there are no more stages. If the IUS stage ΔV is less than the transfer ΔV, the case fails.

4. Next, the transfer ΔV for the second burn is computed by readjusting the plane change.

5. If the remaining IUS stages are within 10 ft/sec of the transfer ΔV, the case is converged. If not, the initial plane change is modified and the program transfers to step 2. The problem is considered infeasible if 200 iterations have been exceeded.

The basic form of the Hohmann equations used is given in the documentation of FUNCTION DLTAV.

RESTRICTIONS

- Operational

Subroutine IUSDV is required.

SUBROUTINE LIQUID

IDENTIFICATION

Name (Title) - LIQUID
Author, Date - Frank Roth and Jack Williams, June 1976
Machine Identification - UNIVAC 1110 EXEC 8
Source Language - FORTRAN V

PURPOSE

The purpose of LIQUID is to compute the ΔV and delta weight (DW) associated with the liquid IUS's which are needed for the Sequence Dependent Test Logic (SDTL) subroutine.

USAGE

- Calling Sequence

  CALL LIQUID ($N)

  Argument:

Parameter name	In/Out	Dimension	Description
$N	-	-	The statement in the calling program to which control is transferred if the IUS does not meet its weight and/or ΔV requirements

- Data In/Out

  Labeled COMMON (Refer to labeled COMMON description section):

Block name	In	Out
C2	2-3,40	4-32
C6	1-60,91-105	
C8	1-47	
C10	1-6	2,7,11-25
C25	3,9,10-13	

METHOD

The ΔV's and DW's for the liquid stage are calculated. They are added to their respective totals and tested to determine if they are within restrictions. If requirements are met, the exit is made via the normal return. If not, the error exit ($N) is taken.

RESTRICTIONS

- Operational

  Subroutine EXP is required.

SUBROUTINE LORBWT

IDENTIFICATION

Name (Title) - LORBWT
Author, Date - Frank Roth and Jack Williams, June 1976
Machine Identification - UNIVAC 1110 EXEC 8
Source Language - FORTRAN V

PURPOSE

The purpose of LORBWT is to compute the launch and deorbit weights of the orbiter.

USAGE

- Calling Sequence

  CALL LORBWT ($N1, $N2)

  Arguments:

Parameter name	In/Out	Dimension	Type	Description
$N1	-	-	-	The statement number in the calling program to which control will pass if the weight is excessive
$N2	-	-	-	The statement number in the calling program to which control will pass if ΔV requirements are not met

- Data In/Out

  Labeled COMMON (Refer to labeled COMMON description section):

Block name	In	Out
C2	2-3,40	4-32
C3	1	2-10
C6	1-60,91-105	
C8	1-47	
C10	1-6	2,7,11-25
C11	7,12	
C25	3,9,10-13	

METHOD

The weight of the vehicle, including payloads but not fuel, and the ΔV required to change orbits, based on each payload are calculated. The weight of OMS fuel is computed and tested against orbiter capacity. If this test is passed, the number of OMS kits required is computed. A new ΔV is calculated and the iteration is continued until enough fuel can be carried to produce the required ΔV. If more than three kits are required, the program exits through RETURN2 and control goes to $N2 in the calling program. RETURN1 is taken when the weight limit is exceeded. Control goes to $N1 in the calling program. The normal return is taken when both criteria are met.

RESTRICTIONS

- Operational

  Subroutines OMSCK and EXP are required.

SUBROUTINE MIXTST

IDENTIFICATION

Name (Title)	- MIXTST (Discipline Mix Testing Routine)
	- MXSTAT (Display Discipline Frequency Routine)
Author, Date	- J. Williams, August 1975
Machine Identification	- UNIVAC 1110
Source Language	- FORTRAN V

PURPOSE

Subroutine MIXTST is used to generate a list of up to 100 payload disciplines; entry point MXSTAT is used to display the frequency of occurrence of each discipline mix.

USAGE

- Calling Sequence

```
CALL MIXTST ($N, KK, MIXCOD)
CALL MXSTAT
```

Arguments

Parameter name	In/Out	Dimension	Type	Description
$N	-	-	-	The statement number in the calling program to which control is passed if an error occurs
KK	In	1	I	A flag, when set nonzero causes the discipline mix to be checked against a list in COMMON
MIXCOD	In	1	I	The discipline mix to be verified or stored

- Data In/Out

Labeled COMMON (refer to the labeled COMMON description section):

Block name	Input	Output
C30	1-221	21-220

METHOD

Subroutine MIXTST cumulates numeric payload discipline mix codes (PDMC) for use by CPTEST as a constraint. The information pertaining to the PDMC is displayed from MXSTAT from subroutine MPLS after the analysis of a particular year has been completed.

RESTRICTIONS

- Operational

  Subroutine DECOMP is required.

SUBROUTINE MPLS

IDENTIFICATION

Name (Title) - MPLS (The MPLS Executive Routine)
Author, Date - J. Williams, August 1975
Machine Identification - UNIVAC 1110
Source Language - FORTRAN V

PURPOSE

Subroutine MPLS is the executive routine to control the initialization of payload data, to generate combinations referencing the payload numbers, and to cause each combination to be tested for feasibility.

USAGE

- Calling Sequence

CALL MPLS (IYEAR, MAXLP)

Arguments:

Parameter name	In/Out	Dimension	Type	Description
IYEAR	In	1	I	The last two digits of the year under analysis
MAXLP	In	1	I	The maximum number of payloads allowed on a combination

- Data In/Out
Labeled COMMON (refer to the labeled COMMON description section):

Block Name	Input	Output
C1	3401-3600	
C5	1-100,202	
C6	61-90	
C7	1-109	
C9	2	
C10	(Not used)	
C11	(Not used)	
C15	2	

METHOD

Subroutine MPLS is the executive for the MPLS. It uses the following procedure:

1. The number of TSV's which are available for use are determined by examining their year of availability.

2. Subroutine PLIST is referenced to generate a working list of payloads.

3. Subroutine PLONTG is referenced to determine which payloads in the list require TSV's.

4. Subroutine COMB is called to generate a unique payload combination.

5. Subroutine SORTL is referenced to arrange the payload combination in order of ascending altitude.

6. A call to subroutine CPTEST is made to perform the sequence dependent and independent test.

7. Once all the combinations have been generated, subroutine FEACOM is called to generate a list of missions.

8. If required, the number of feasible combinations is reduced to 500 or less.

Subroutines STATS and MIXTST are called to output tables related to the discipline mix and mission types.

RESTRICTIONS

- Operational

  Subroutines PLIST, PLONTG, COMB, SORTL, CPTEST, TIME, FEACOM, STATS, and MIXTST are referenced.

SUBROUTINE OMSCK

IDENTIFICATION

Name (Title) - OMSCK (Checks Required Number of OMS Kits)
Author, Date - J. Eggleston, August 1976
Machine Identification - UNIVAC 1110 EXEC 8
Source Language - FORTRAN V

PURPOSE

The purpose of OMSCK is to determine the number of OMS fuel kits required to execute a particular mission and their associated weight.

USAGE

- Calling Sequence

CALL OMSCK (POINC, ORBDV, OMSWT, NOMSKT)

Arguments:

Parameter name	In/Out	Dimension	Type	Description
POINC	In	1	R	Parking orbit inclination
ORBDV	In	1	R	Total orbiter ΔV for the mission
OMSWT	Out	1	R	Total weight of OMS fuel required
NOMSKT	Out	1	I	Number of OMS fuel kits required

- Labeled COMMON (refer to labeled COMMON description section):

Block name	Input	Output
C32	1-19	

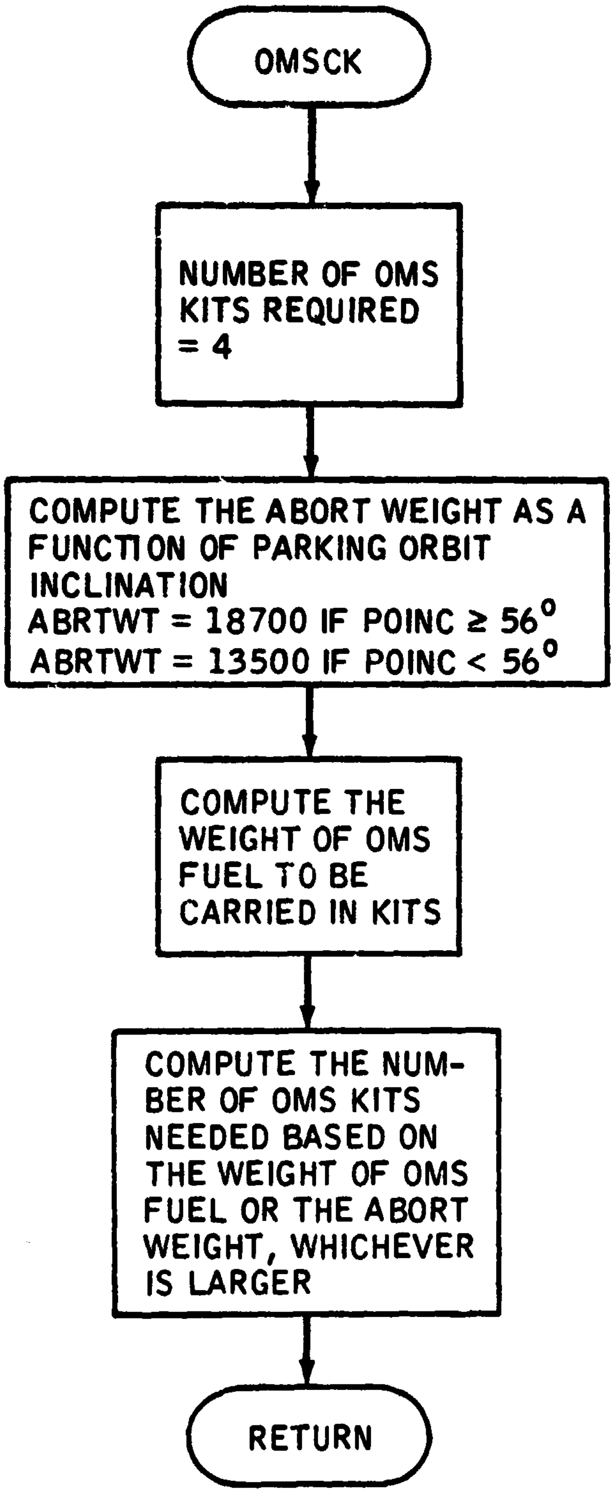
METHOD

The number of OMS fuel kits required is initially set to four. The weight of the OMS fuel needed and the number of kits are calculated as a function of the initial parking orbit and total ΔV. Both are returned as output.

RESTRICTIONS

- Operational

  Subroutine EXP is required.


OMSCK
NUMBER OF OMS KITS REQUIRED = 4
COMPUTE THE ABORT WEIGHT AS A FUNCTION OF PARKING ORBIT INCLINATION
ABRTWT = 18700 IF POINC ≥ 56°
ABRTWT = 13500 IF POINC < 56°
COMPUTE THE WEIGHT OF OMS FUEL TO BE CARRIED IN KITS
COMPUTE THE NUMBER OF OMS KITS NEEDED BASED ON THE WEIGHT OF OMS FUEL OR THE ABORT WEIGHT, WHICHEVER IS LARGER
RETURN

SUBROUTINE ORBITR

IDENTIFICATION

Name (Title) - ORBITR (Orbit Calculations)
Author, Date - F. Roth and J. Williams, June 1976
Machine Identification - UNIVAC 1110 EXEC 8
Source Language - FORTRAN V

PURPOSE

The purpose of ORBITR is to compute the ΔV requirements for deorbit.

USAGE

o Calling Sequence

CALL ORBITR (TFLAG, OFLAG, IPM, $)

Arguments:

Parameter name	In/Out	Dimension	Type	Description
TFLAG	In	1	Logical	= .TRUE. There are TSV payloads = .FALSE. No TSV payloads
OFLAG	In	1	Logical	= .TRUE. There are orbiter payloads = .FALSE. No orbiter payloads
IPM	In	1	I	Payload permutation
$	-	-	-	Statement to which control is transferred if payload length exceeds XLMAX

• Data In/Out

Labeled COMMON (Refer to labled COMMON description section):

Block name	In	Out
C1	2001-2600	
	2801-3600	
	7201-7800	
C2	2-3,40	4-32
C3	1	2-10
C8	1-47	
C10	1-6	2,7,11-25
C25	3,9,10-13	

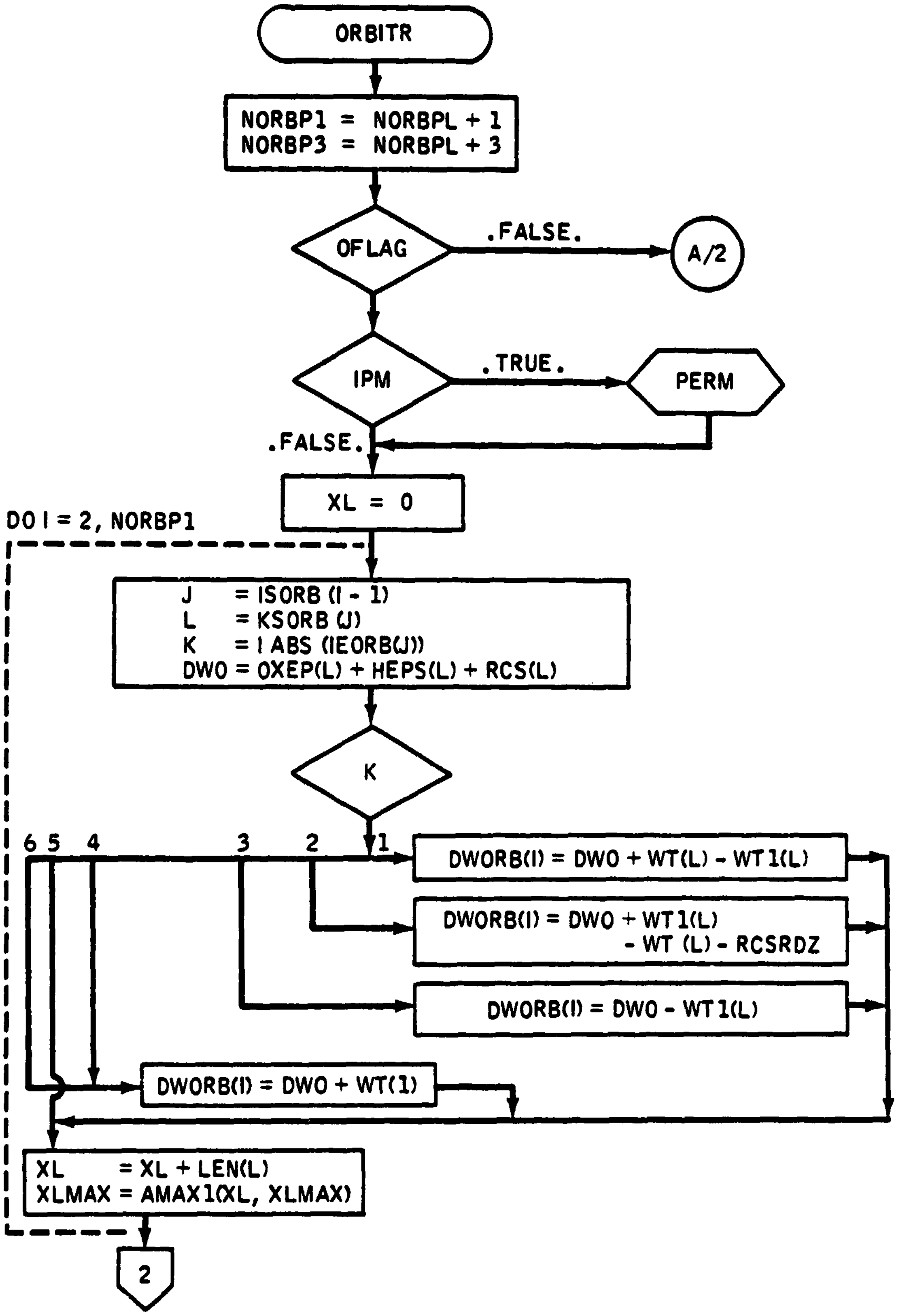
METHOD

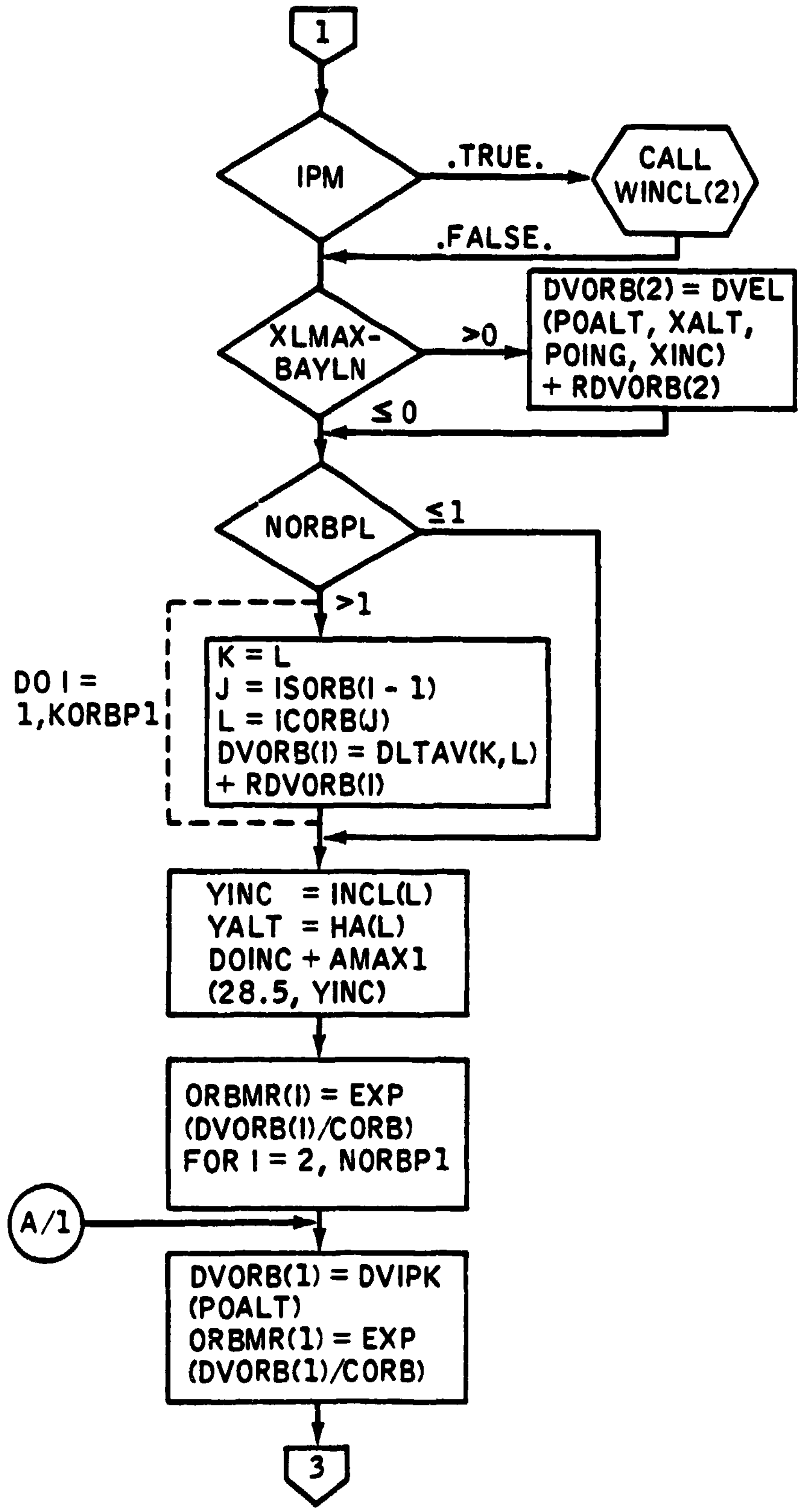
The deorbit weight is computed by starting with the final down weight and computing the weight of expendables used to perform the most recently enacted maneuver. Stepping back through each maneuver to launch, the total launch weight is calculated along with the ΔV requirements. Fuel load is altered as required and the process repeated until fuel requirements are met or the orbiter limits exceeded.

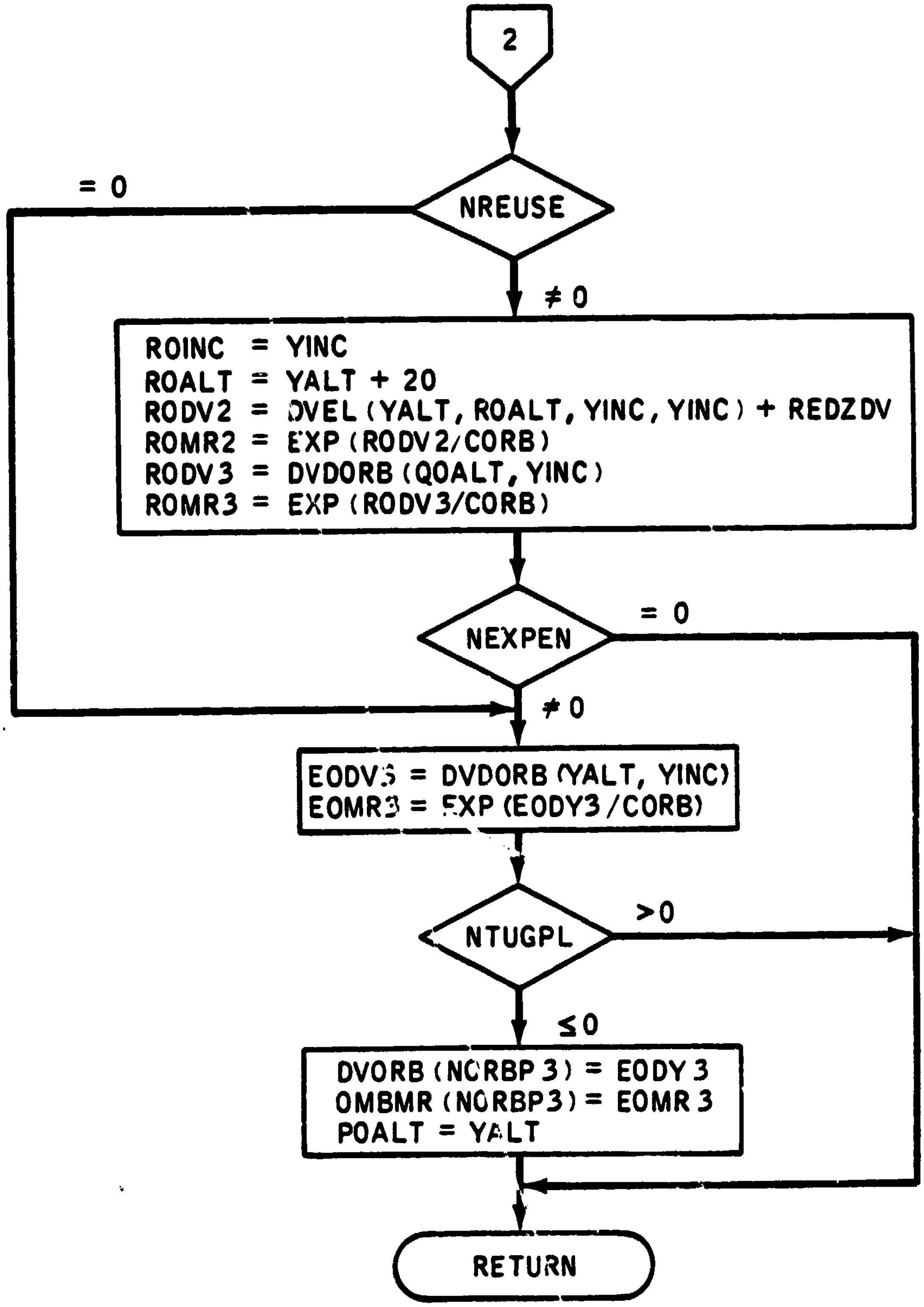
RESTRICTIONS

- Operational

  Subroutines PERM, DLTAV, and WINCL are required.


ORBITR
NORBP1 = NORBPL + 1
NORBP3 = NORBPL + 3
OFLAG
.FALSE.
A/2
IPM
.TRUE.
PERM
.FALSE.
XL = 0
DO I = 2, NORBP1
J = ISORB (I - 1)
L = KSORB (J)
K = I ABS (IEORB(J))
DWO = OXEP(L) + HEPS(L) + RCS(L)
K
6
5
4
3
2
1
DWORB(I) = DWO + WT(L) - WT1(L)
DWORB(I) = DWO + WT1(L) - WT (L) - RCSRDZ
DWORB(I) = DWO - WT1(L)
DWORB(I) = DWO + WT(1)
XL = XL + LEN(L)
XLMAX = AMAX1(XL, XLMAX)
2


1
IPM
.TRUE.
CALL WINCL(2)
.FALSE.
XLMAX-BAYLN
>0
DVORB(2) = DVEL (POALT, XALT, POING, XINC) + RDVORB(2)
≤ 0
NORBPL
≤1
>1
DO I = 1,KORBP1
K = L
J = ISORB(I - 1)
L = ICORB(J)
DVORB(I) = DLTAV(K, L) + RDVORB(I)
YINC = INCL(L)
YALT = HA(L)
DOINC + AMAX1 (28.5, YINC)
ORBMR(I) = EXP (DVORB(I)/CORB) FOR I = 2, NORBP1
A/1
DVORB(1) = DVIPK (POALT)
ORBMR(1) = EXP (DVORB(1)/CORB)
3


2
NREUSE
= 0
≠ 0
ROINC = YINC
ROALT = YALT + 20
RODV2 = DVEL (YALT, ROALT, YINC, YINC) + REDZDV
ROMR2 = EXP (RODV2/CORB)
RODV3 = DVDORB (QOALT, YINC)
ROMR3 = EXP (RODV3/CORB)
NEXPEN
= 0
≠ 0
EODV3 = DVDORB (YALT, YINC)
EOMR3 = EXP (EODY3/CORB)
NTUGPL
>0
≤0
DVORB (NORBP3) = EODY3
OMBMR (NORBP3) = EOMR3
POALT = YALT
RETURN

SUBROUTINE PERM

IDENTIFICATION

Name (Title) - PERM (Permutation Generator)
Author, Date - J. Williams, August 1975
Machine Identification - UNIVAC 1110
Source Language - FORTRAN V

PURPOSE

Subroutine PERM generates a permutation of a set of N variables taken N at a time. One permutation is generated for each call.

USAGE

- Calling Sequence

  CALL PERM (N, IEND, IPERM, ICNT)

  Arguments:

Parameter name	In/Out	Dimension	Type	Description
N	In	1	I	Number of variables in the permutation
IEND	In/Out	1	I	An initialization flag = 0, initialize the permutation generator ≠0, generate the next permutation
IPERM	Out	Dimensioned in calling program	I	A list which contains the new permutation
ICNT	Out	Dimensioned in calling program	I	An array of counters used to generate a permutation

METHOD

Each permutation is formed by a cyclic permutation of all or part of a previous permutation. The logic is structured such that a single call to PERM generates a permutation.

SUBROUTINE PLDSCN

IDENTIFICATION

Name (Title) - PLDSCN (Scans Payloads in Combination)
Author, Date - E. H. Perrenot, October 1976
Machine Identification - UNIVAC 1110
Source Language - FORTRAN V

PURPOSE

Subroutine PLDSCN scans payloads within a combination in the order of their operations and groups them into mission events.

USAGE

- Calling Sequence

CALL PLDSCN (NPL, IPTR, TDO, EVNT, NEVNT)

Arguments:

Parameter name	In/Out	Dimension	Type	Description
NPL	In	1	I	Number of payloads in the combination
IPTR	In	Dimensioned in calling program	I	Array containing pointers to payload data in array TDO (reflects payload sequence within the combination)
TDO	In	6x5	R	Array of payload data
EVNT	Out	6x4	R	Array of data concerning events
NEVNT	Out	1	I	Number of events

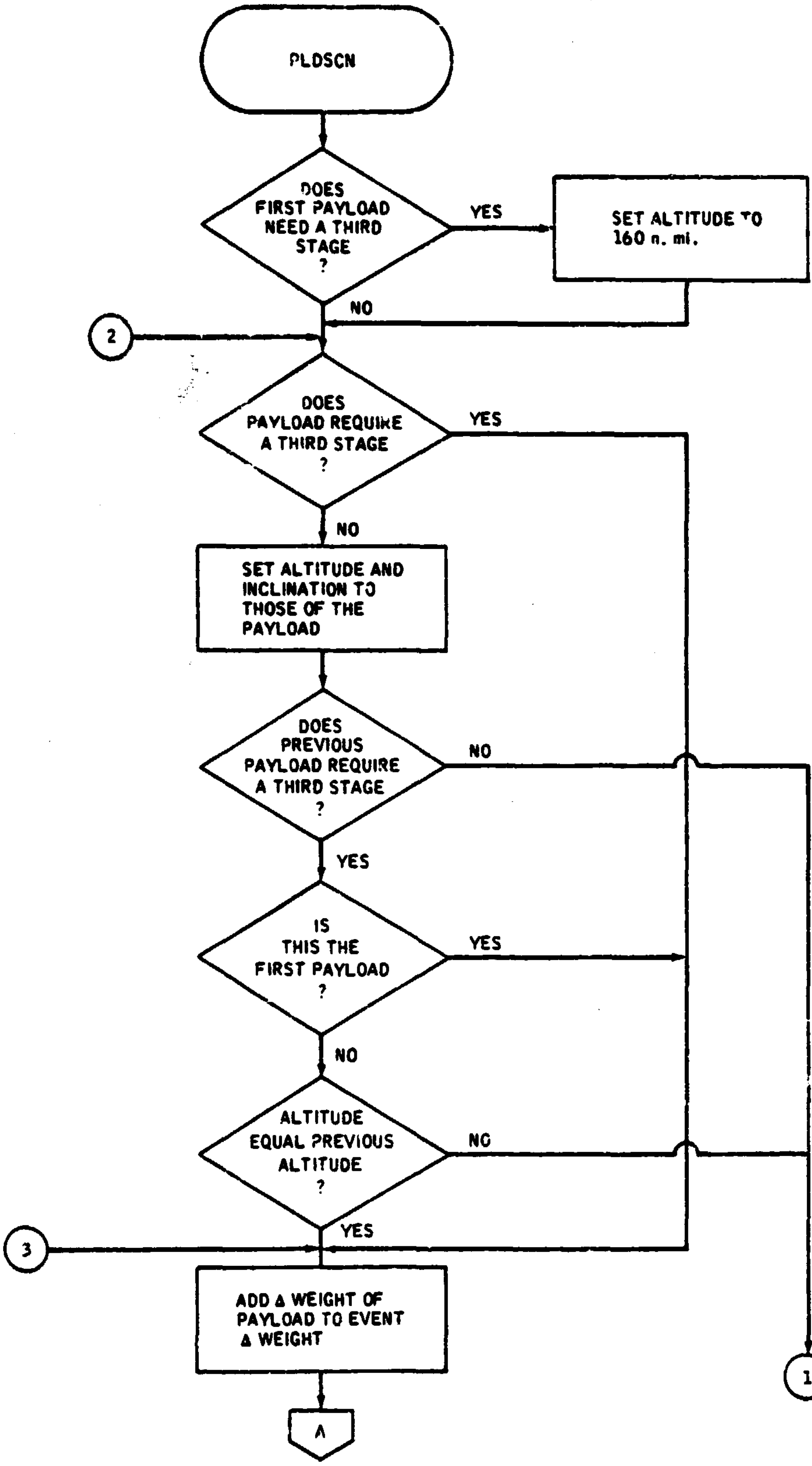
METHOD

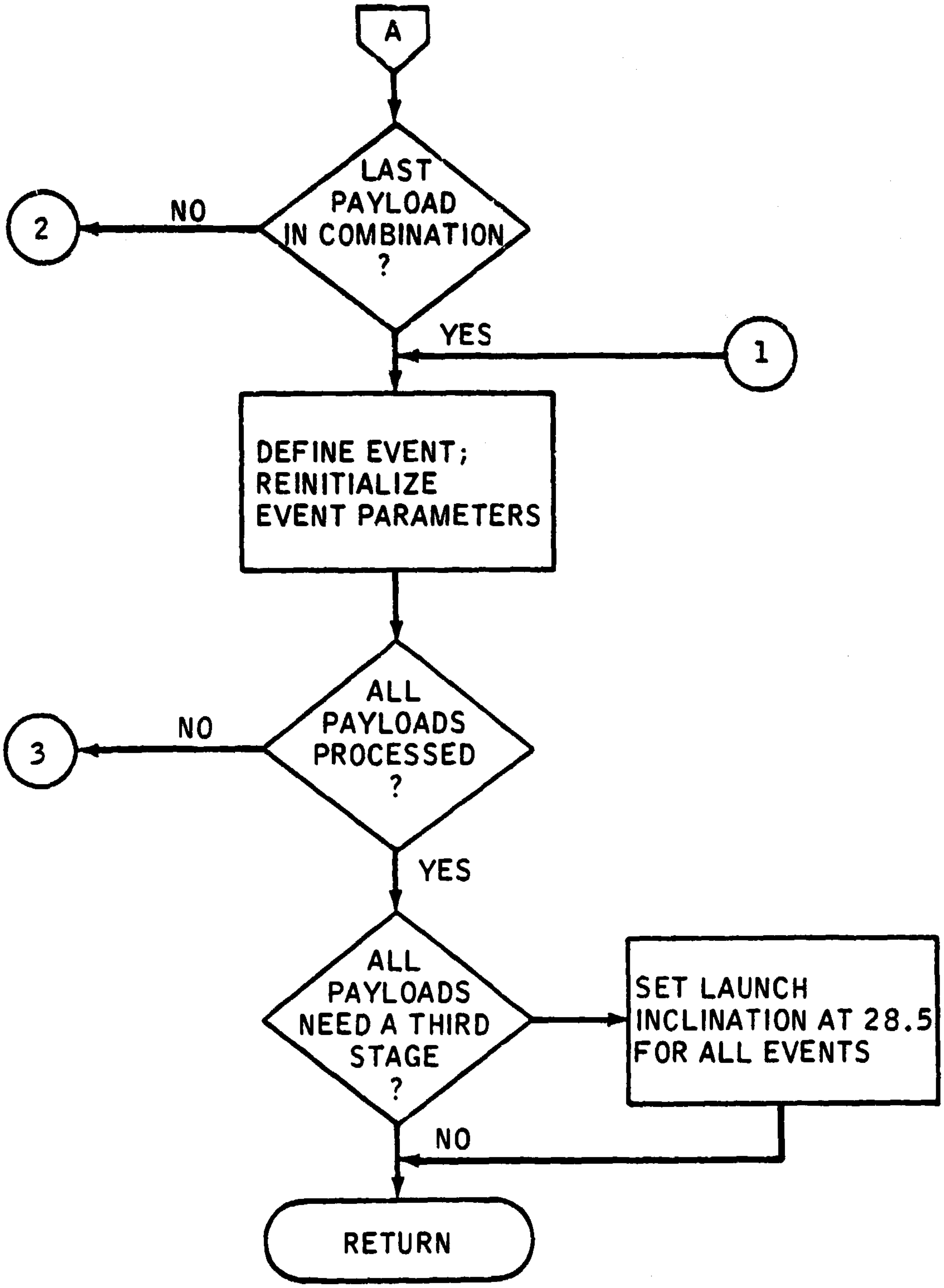
Subroutine PLDSCN groups payloads in a combination into events. It accomplishes this task by examining payload characteristics such as altitude and need for a third stage. In defining events, the following guidelines are used:

1. All payloads requiring a third stage and preceding the first non-third-stage payload are deployed at an altitude of 160 n. mi. if the first non-third-stage payload requires an altitude of greater than 160 n. mi.

2. If third-stage payloads precede a non-third-stage payload with an altitude of less than 160 n. mi., they are deployed at the altitude of the non-third-stage payload.

3. Excepting the above cases, no third-stage payload(s) ever comprises an event by itself; it is deployed at the altitude of the preceding non-third-stage payload.


PLDSCN
DOES FIRST PAYLOAD NEED A THIRD STAGE ?
YES
SET ALTITUDE TO 160 n. mi.
NO
2
DOES PAYLOAD REQUIRE A THIRD STAGE ?
YES
NO
SET ALTITUDE AND INCLINATION TO THOSE OF THE PAYLOAD
DOES PREVIOUS PAYLOAD REQUIRE A THIRD STAGE ?
NO
YES
IS THIS THE FIRST PAYLOAD ?
YES
NO
ALTITUDE EQUAL PREVIOUS ALTITUDE ?
NO
YES
3
ADD Δ WEIGHT OF PAYLOAD TO EVENT Δ WEIGHT
1
A


A
LAST PAYLOAD IN COMBINATION ?
NO
2
YES
1
DEFINE EVENT; REINITIALIZE EVENT PARAMETERS
ALL PAYLOADS PROCESSED ?
NO
3
YES
ALL PAYLOADS NEED A THIRD STAGE ?
SET LAUNCH INCLINATION AT 28.5 FOR ALL EVENTS
NO
RETURN

SUBROUTINE PLIST

IDENTIFICATION

Name (Title) - PLIST (Payload List Routine)
Author, Date - J. Williams, August 1975
Machine Identification - UNIVAC 1110
Source Language - FORTRAN V

PURPOSE

Subroutine PLIST searches the payload model and selects a list which can be flown in a particular year.

USAGE

- Calling Sequence

  CALL PLIST (IYEAR)

  Argument:

Parameter name	In/Out	Dimension	Type	Description
IYEAR	In	1	I	The last two digits of the year under analysis

- Data In/Out

  Labeled COMMON (refer to the labeled COMMON description section):

Block name	Input	Output
C1	3401-6200,8201	
C5		1-363
C9	50	

METHOD

Subroutine PLIST forms a working list of payloads from the payload model by eliminating those payloads which do not fly in the year under analysis. In addition, payloads that are repeated or those which are up/down payloads are identified. The technique used is based on the definition of the variable FLTPYR which is described in detail in section 5.2.

A distinction is made between nominal payloads and up/down payloads by using negative numbers to represent the mission type.

SUBROUTINE PLONTG

IDENTIFICATION

Name (Title) - PLONTG (Payload on TSV Routine)
Author, Date - J. Williams, August 1975
Machine Identification - UNIVAC 1110
Source Language - FORTRAN V

PURPOSE

Subroutine PLONTG determines the TSV requirements for payloads to be flown in a particular year.

USAGE

- Calling Sequence

  CALL PLONTG

- Data In/Out

  Labeled COMMON (refer to labeled COMMON description section):

Block name	Input	Output
C1	2001-2600,2801-3400, 8001-8200	
C5	1-100,202,264-363	364-463
C8		1,3,5,7
C9	50	
C10		1-3,4
C11		7
C25	10-11,13,15	

METHOD

Subroutine PLONTG is used to determine which payloads in the working list require a TSV. The following procedure is used to determine if a TSV is required:

1. If the dedicated TSV velocity parameter associated with that payload is nonzero, a TSV is required.

2. If the payload's orbit exceeds 700 n.mi., a TSV is required.

3. If the payload when flown alone requires more than three OMS kits or has a payload margin less than zero, a TSV is required.

If all the conditions are met, then a 1 is stored in the NEEDTG array.

RESTRICTIONS

- Operational

  Functions DVIPK, DVEL, DVDORB, and subroutine OMSCK are required.

SUBROUTINE REDUNT

IDENTIFICATION

Name (Title) - REDUNT (Redundancy Check Routine)
Author, Date - J. Williams, August 1975
Machine Identification - UNIVAC 1110
Source Language - FORTRAN V

PURPOSE

Subroutine REDUNT is used to eliminate payload combinations that have an infeasible subset.

USAGE

- Calling Sequence

  CALL REDUNT ($N)

  Argument:

Parameter name	In/Out	Dimension	Type	Description
$N	-	-	-	The statement number in the calling program to which control is transferred if an error occurs

- Data In/Out

  Labeled COMMON (Refer to the labeled COMMON description section):

Block name	Input	Output
C34	All	

METHOD

Subroutine REDUNT evaluates a candidate payload combination by comparing a subset of the combination to the previous generated set in the feasible mission file. The technique eliminates the second payload from the combination and compares it to a previous generated set, for example:

Candidate combination	Temporary set	Feasible missions previous set
ABCDE	ABCE	ABCD ABCE ACDE ACDF ADFG BCDE

Since the combination was generated by adding the last payload, E, then the subset ABCE must exist before the combination can be successful. This method avoids complex testing of a combination which has an infeasible subset.

SUBROUTINE SDTL

IDENTIFICATION

Name (Title) - SDTL (Sequence Dependent Test Logic)
Author, Date - Frank Roth and Jack Williams, June 1976
Machine Identification - UNIVAC 1110 EXEC 8
Source Language - FORTRAN V

PURPOSE

The purpose of SDTL is to perform the sequence-dependent test logic for choosing payload permutations and third stage vehicles. It is an executive routine that calls other modeling routines such as LORBWT, CGIN, and TSV as they are needed.

USAGE

- Calling Sequence

CALL SDTL ($N)

Argument:

Parameter name	In/Out	Dimension	Type	Description
$N	-	-	-	The statement in the calling program to which control is transferred if the combination is infeasible

- Data In/Out

Labeled COMMON (refer to the labeled COMMON description section):

Block name	Input	Output
C3	1	2-10
C6	1-60,91-105	
C8	1-47	
C10	1-6	2,7,11-25
C25	3,9,10-13	

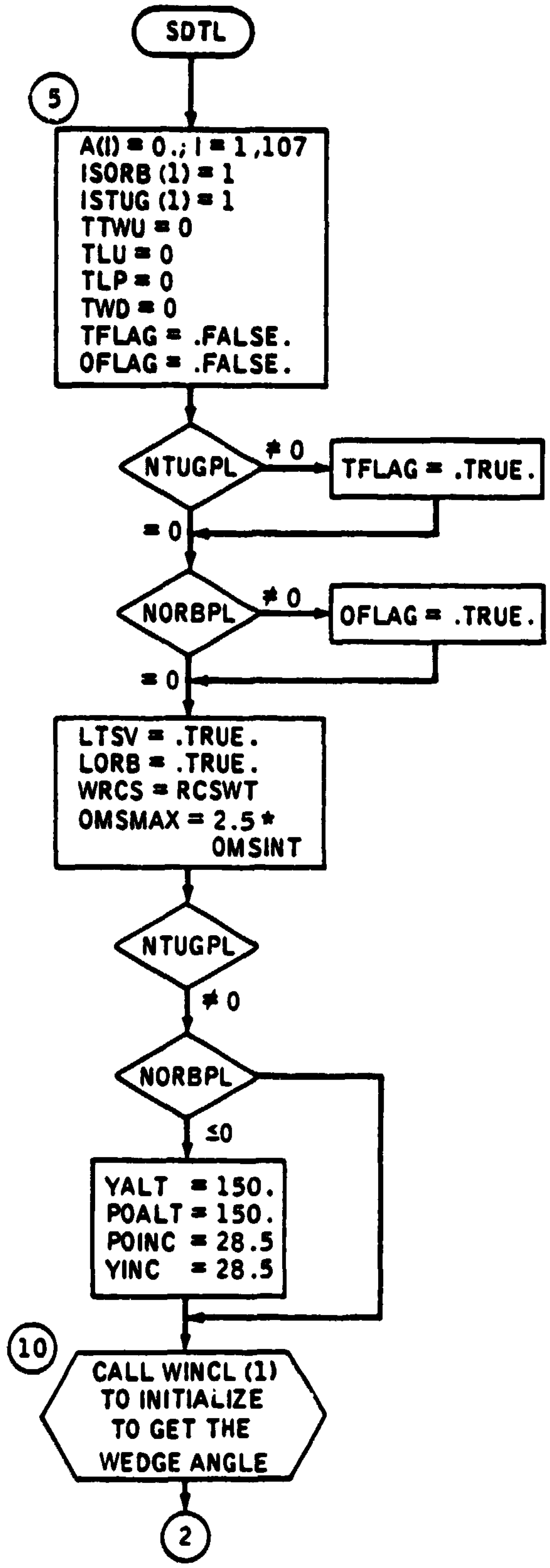
METHOD

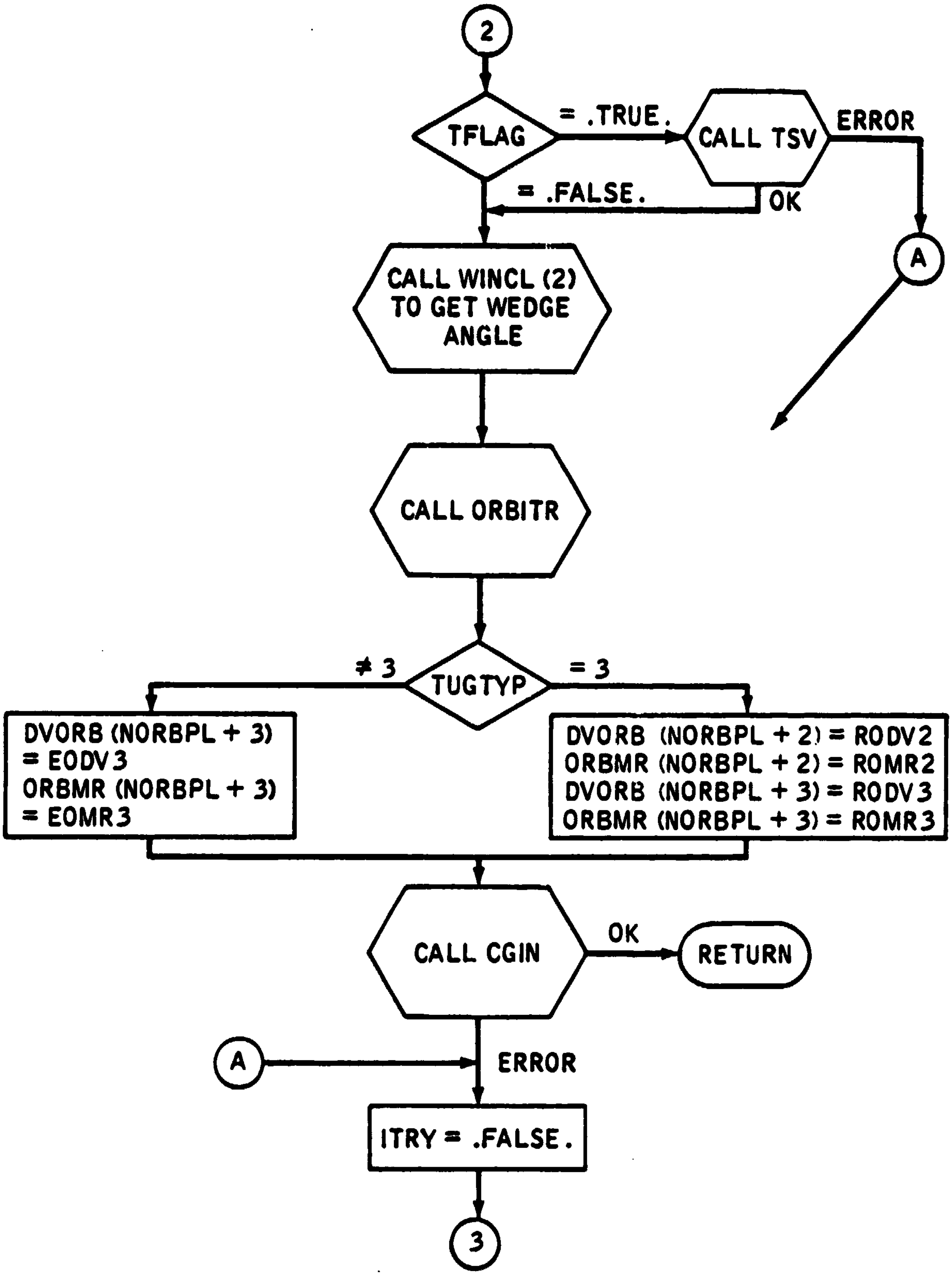
Subroutine SDTL verifies that a payload combination is feasible. It examines each permutation within the combination until it finds one that will fly with an available third stage vehicle (TSV). If no suitable TSV can be found, the next permutation is obtained and the process repeated. If the payload cannot fly on any available TSV, control goes to $N. If constraints allow the payload to fly, the normal exit is taken from the subroutine. The various constraints are tested by the modeling routines.

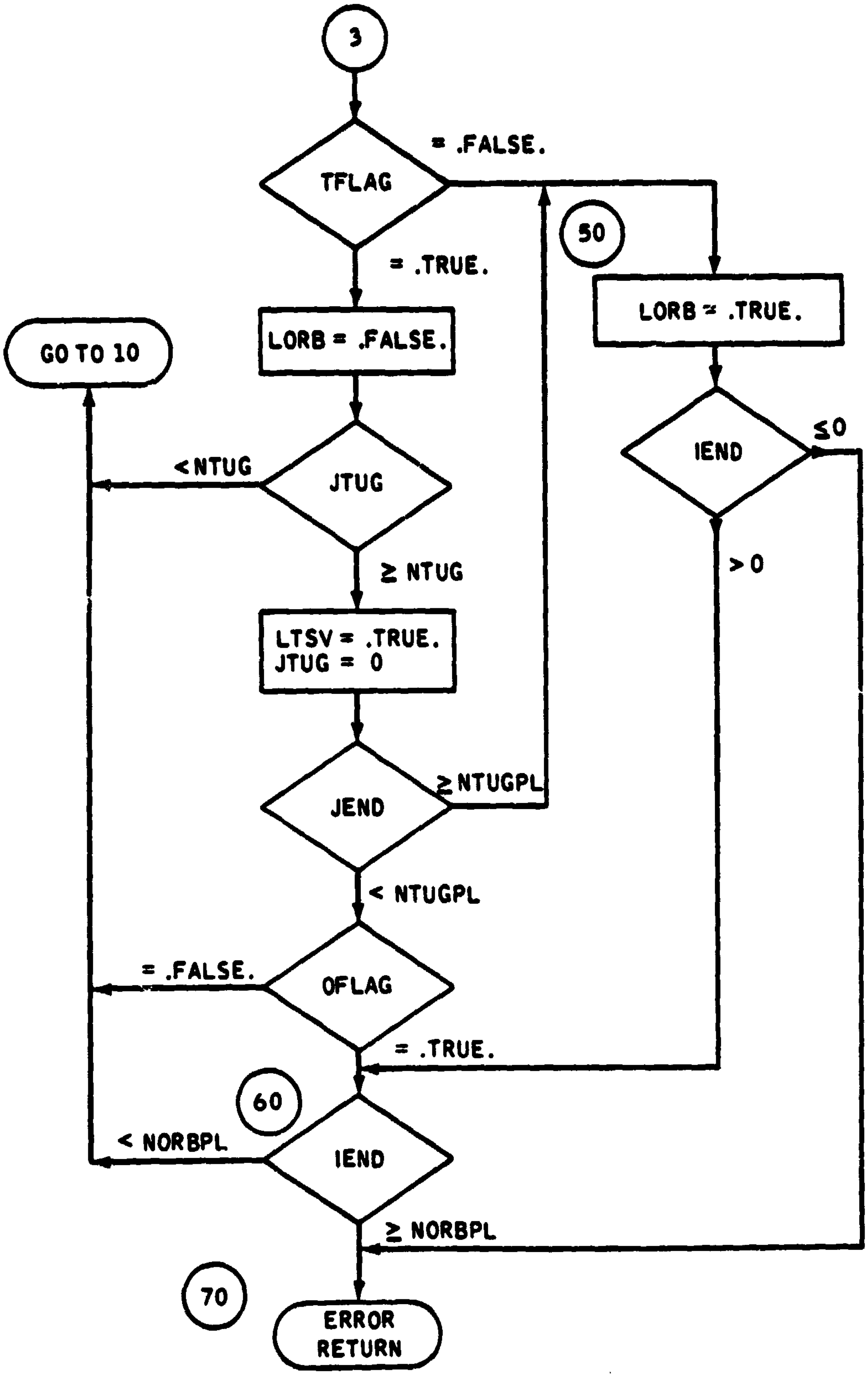
RESTRICTIONS

- Operational

  Subroutines LORBWT, CGIN, and TSV are required.


SDTL
5
A(I) = 0.; I = 1,107
ISORB (1) = 1
ISTUG (1) = 1
TTWU = 0
TLU = 0
TLP = 0
TWD = 0
TFLAG = .FALSE.
OFLAG = .FALSE.
NTUGPL
≠ 0
TFLAG = .TRUE.
= 0
NORBPL
≠ 0
OFLAG = .TRUE.
= 0
LTSV = .TRUE.
LORB = .TRUE.
WRCS = RCSWT
OMSMAX = 2.5 * OMSINT
NTUGPL
≠ 0
NORBPL
≤0
YALT = 150.
POALT = 150.
POINC = 28.5
YINC = 28.5
10
CALL WINCL (1) TO INITIALIZE TO GET THE WEDGE ANGLE
2


2
TFLAG
= .TRUE.
CALL TSV
ERROR
= .FALSE.
OK
A
CALL WINCL (2) TO GET WEDGE ANGLE
CALL ORBITR
≠ 3
TUGTYP
= 3
DVORB (NORBPL + 3) = EODV3
ORBMR (NORBPL + 3) = EOMR3
DVORB (NORBPL + 2) = RODV2
ORBMR (NORBPL + 2) = ROMR2
DVORB (NORBPL + 3) = RODV3
ORBMR (NORBPL + 3) = ROMR3
CALL CGIN
OK
RETURN
A
ERROR
ITRY = .FALSE.
3


3
TFLAG
= .FALSE.
= .TRUE.
50
LORB = .TRUE.
GO TO 10
LORB = .FALSE.
IEND
≤0
< NTUG
JTUG
≥ NTUG
> 0
LTSV = .TRUE.
JTUG = 0
JEND
≥ NTUGPL
< NTUGPL
= .FALSE.
OFLAG
= .TRUE.
60
< NORBPL
IEND
≥ NORBPL
70
ERROR
RETURN

SUBROUTINE SOLID

IDENTIFICATION

Name (Title)	- SOLID
Author, Date	- Frank Roth and Jack Williams, June 1976
Machine Identification	- UNIVAC 1110 EXEC 8
Source Language	- FORTRAN V

PURPOSE

The purpose of subroutine SOLID is to compute that portion of the total ΔV and total ΔV resulting from the use of a particular solid fuel TSV.

USAGE

- Calling Sequence
  CALL SOLID ($N)
  Argument:

Parameter name	In/Out	Dimension	Type	Description
$N	-	-	-	The statement in the calling program to which control will pass if this TSV cannot be used

- Data In/Out

  Labeled COMMON (refer to the labeled COMMON description section):

Block name	Input	Output
C1	2001-2600,2801-3600 7201-7800	
C2	2-3,40	4-32
C3	1	2-10
C8	1-47	
C10	1-6	2,7,11-25

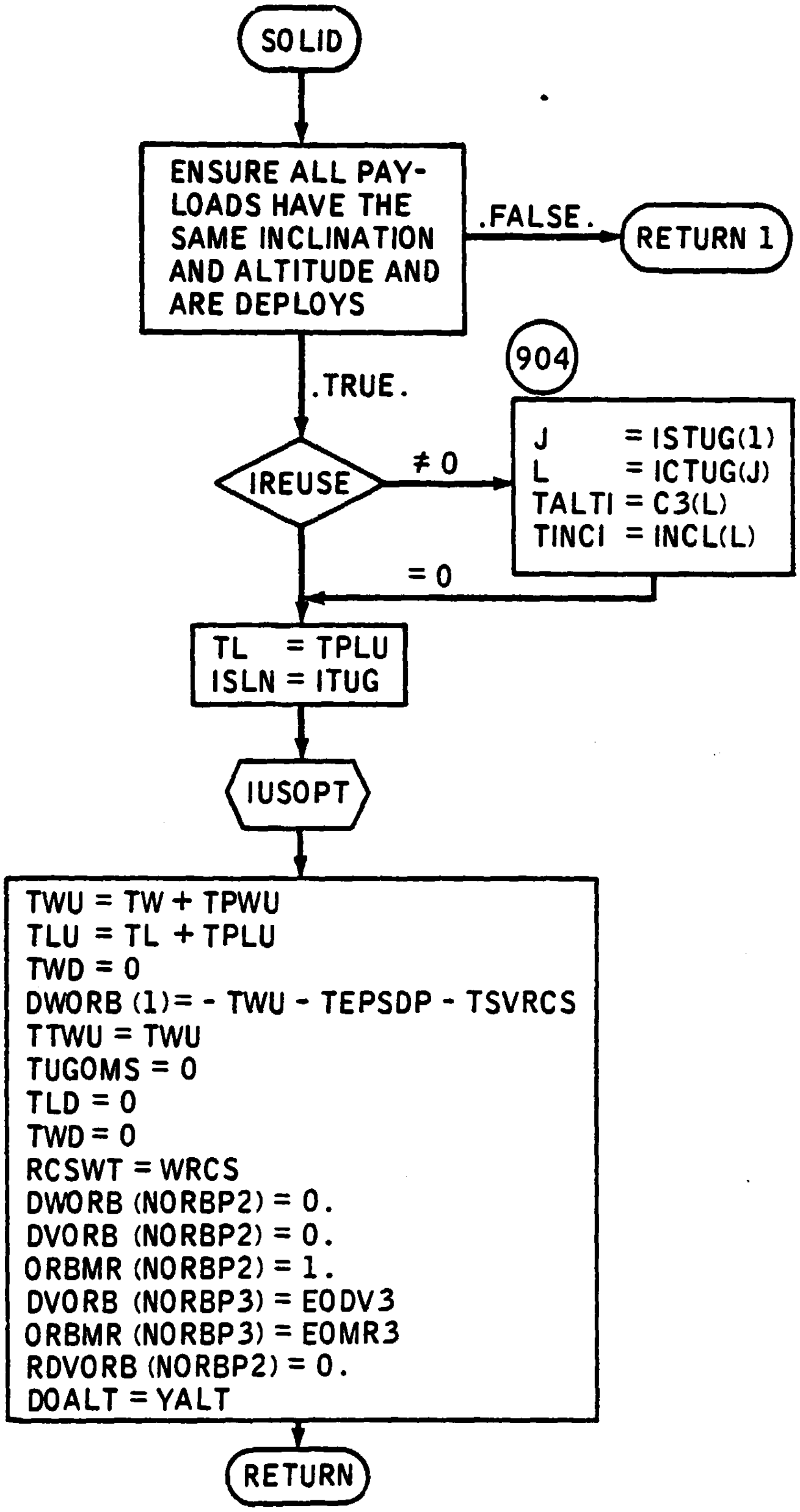
METHOD

The program ensures that all the payloads have the same altitude and inclination and are deploys. It computes total length and weight up and down by summing over all the payloads to determine if the payloads can fly on this TSV. If all criteria are passed, it computes the ΔV and ΔW for this rocket for this portion of the orbit. If the TSV fails to meet one of the criteria, the subroutine exits to $N.

RESTRICTIONS

- Operational

  Subroutine IUSOPT is required.


SOLID
ENSURE ALL PAY-LOADS HAVE THE SAME INCLINATION AND ALTITUDE AND ARE DEPLOYS
.FALSE.
RETURN 1
.TRUE.
904
IREUSE
≠ 0
J = ISTUG(1)
L = ICTUG(J)
TALTI = C3(L)
TINCI = INCL(L)
= 0
TL = TPLU
ISLN = ITUG
IUSOPT
TWU = TW + TPWU
TLU = TL + TPLU
TWD = 0
DWORB (1)= - TWU - TEPSDP - TSVRCS
TTWU = TWU
TUGOMS = 0
TLD = 0
TWD = 0
RCSWT = WRCS
DWORB (NORBP2) = 0.
DVORB (NORBP2) = 0.
ORBMR (NORBP2) = 1.
DVORB (NORBP3) = EODV3
ORBMR (NORBP3) = EOMR3
RDVORB (NORBP2) = 0.
DOALT = YALT
RETURN

SUBROUTINE SORT

IDENTIFICATION

Name (Title) - SORT (Descending Sort Module)
Author, Date - J. Williams, August 1975
Machine Identification - UNIVAC 1110
Source Language - FORTRAN V

PURPOSE

Subroutine SORT sorts a list of N integers into descending order.

USAGE

- Calling Sequence

  CALL SORT (N, X, Y)

  Arguments:

Parameter name	In/Out	Dimension	Type	Description
N	In	1	I	Number of variables to be sorted
X	In	Dimensioned in calling routine	I	Listed of integers to be sorted
Y	Out	Dimensioned in calling routine	I	List of integers sorted in descending order

METHOD

Subroutine SORT arranges a list in descending order by searching the list to find the maximum number in the array; the tests are repeated N times.

SUBROUTINE SORTL

IDENTIFICATION

Name (Title) - SORTL (Ascending Order Sort)
Author, Date - J. Williams, August 1975
Machine Identification - UNIVAC 1110
Source Language - FORTRAN V

PURPOSE

Subroutine SORTL sorts the HA array into ascending order and rearranges the IA array accordingly.

USAGE

- Calling Sequence

  CALL SORTL (M, IJ, HA, IA)

  Arguments:

Parameter name	In/Out	Dimension	Type	Description
M	In	1	I	Number of variables to be sorted
IJ	In	Dimensioned in calling program	I	An array which is stored into IA
HA	In/Out	Dimensioned in calling program	I	An array sorted into ascending order
IA	Out	Dimensioned in calling program	I	The IJ array arranged in an order corresponding to the sorted HA array

METHOD

Subroutine SORTL arranges the HA array into ascending order by incrementing through the list to determine if it was less than the previous minimum value. If the ith word is greater than the previous word, the word values are switched and a counter is decremented. The procedure continues until all the words have been tested. Since each element of the IA array corresponds to the HA array, it is arranged in an order corresponding to the sorted HA array.

SUBROUTINE SORTX

IDENTIFICATION

Name (Title) - SORTX (Sort Ascending Order)
Author, Date - J. Williams, August 1975
Machine Identification - UNIVAC 1110
Source Language - FORTRAN V

PURPOSE

Subroutine SORTX sorts a list of M integers into ascending order.

USAGE

- Calling Sequence

  CALL SORTX (M, IA)

  Arguments:

Parameter name	In/Out	Dimension	Type	Description
M	In	1	I	Number of variables to be sorted
IA	In/Out	Dimensioned in calling program	I	Array to be sorted into ascending order

METHOD

Subroutine SORTX arranges the IA array into ascending order by incrementing through the list to determine if it was less than the previous minimum value. If the ith word was greater than the previous word, the word values are switched and a counter is decremented. The procedure continues until all the words have been tested.

SUBROUTINE STATS

IDENTIFICATION

Name (Title)	- STATS (Mission Type Status Routine)
Author, Date	- J. Williams, August 1975
Machine Identification	- UNIVAC 1110
Source Language	- FORTRAN V

PURPOSE

Subroutine STATS uses the mission type list to reject payload combinations as a function of the mission types.

USAGE

- Calling Sequence

  CALL STATS ($N, IB, M, INCR, MATCH, ISTART, IEND, IALT)

  Arguments:

Parameter name	In/Out	Dimension	Type	Description
$N	In	-	I	Statement number in the calling program to which control is transferred if an error occurs
IB	In	6	I	Mission type numbers of the payload sequence being evaluated
M	In	1	I	Number of elements in IB
INCR	In	1	I	Number of missions generated with the same sequence; the repeated missions stored
MATCH	Out	1	I	A flag which denotes if nonzero that the mission types in IB correspond to the mission class codes in COMMON C12
ISTART	In	1	I	First nonzero word in the mission type list

Parameter name	In/Out	Dimension	Type	Description
IEND	In	1	I	Last nonzero word in the mission type list
IALT	In	1	I	Not used

- Data In/Out

  Labeled COMMON (refer to the labeled COMMON description section):

Block name	Input	Output
C9	3-49	
C12	1-54	

METHOD

Subroutine STATS uses an internal allowable list of mission-type codes to constrain candidate missions. If a candidate mission-type's code does not correspond to the MSCLCD array in COMMON, it is rejected; otherwise, the mission is accepted.

SUBROUTINE TSV

IDENTIFICATION

Name (Title) - TSV (Third Stage Vehicles)
Author, Date - F. Roth and J. Williams, June 1976
Machine Identification - UNIVAC 1110, EXEC 8
Source Language - FORTRAN V

PURPOSE

The purpose of TSV is to compute all the data for ΔV and ΔW which is independent of fuel consumption. TSV calls the appropriate subroutine (SOLID or LIQUID) to complete data requirements.

USAGE

- Calling Sequence

CALL TSV (TFLAG, LTSV, $N)

Arguments:

Parameter name	In/Out	Dimension	Type	Description
TFLAG	In	1	Logical	No longer used
LTSV	In	1	Logical	.TRUE. = There is a TSV for this payload .FALSE.= No TSV
$N	-	-	-	The statement in the calling program to which control is passed if an abnormal exit is made from subroutine SOLID or LIQUID

- Data In/Out

Labeled COMMON (refer to the labeled COMMON description section):

Block name	Input	Output
C1	2001-2600 2801-3600 7201-7800	
C2	2,3,40	4-32
C3	1	2-10
C6	1-60,91-105	

Block name	Input	Output
C10	1-6	2,7,11-25
C32	1-3	

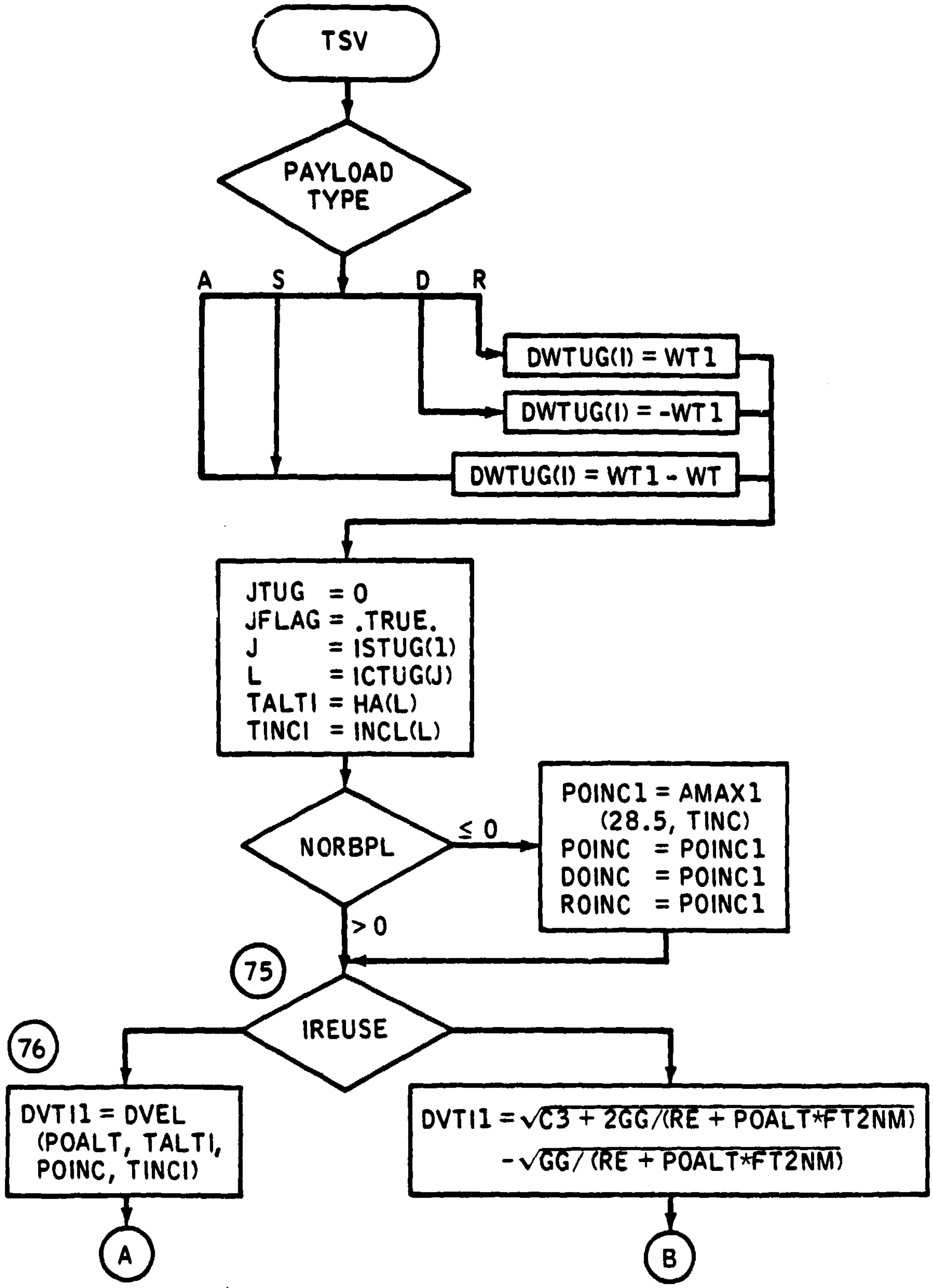
METHOD

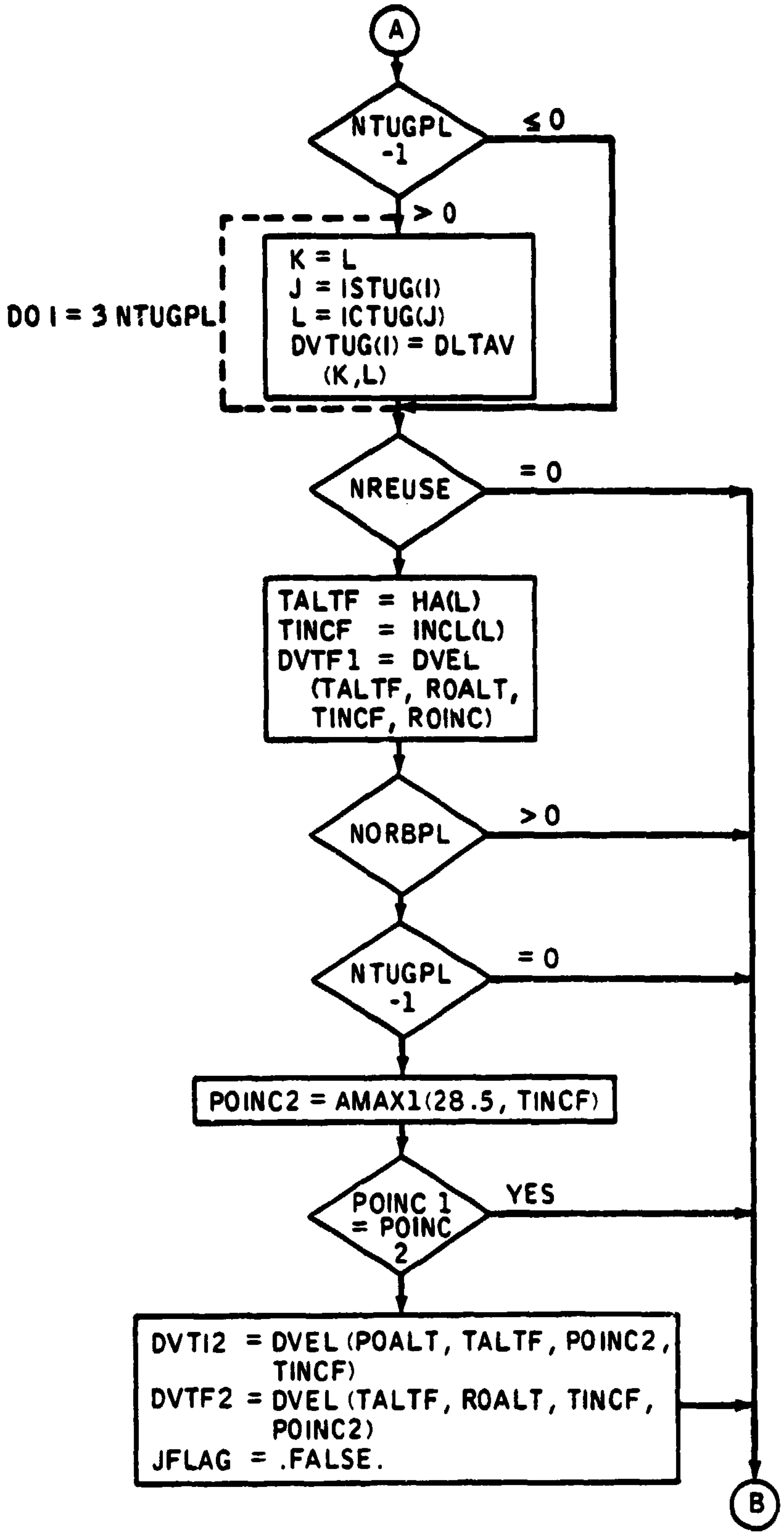
If LTSV = .FALSE. the program returns with no action. If LTSV = .TRUE. the total down weight of the payloads is calculated and the ΔV computed for the required altitude and inclination. Subroutine SOLID or LIQUID is called to retrieve the associated ΔV and ΔW.

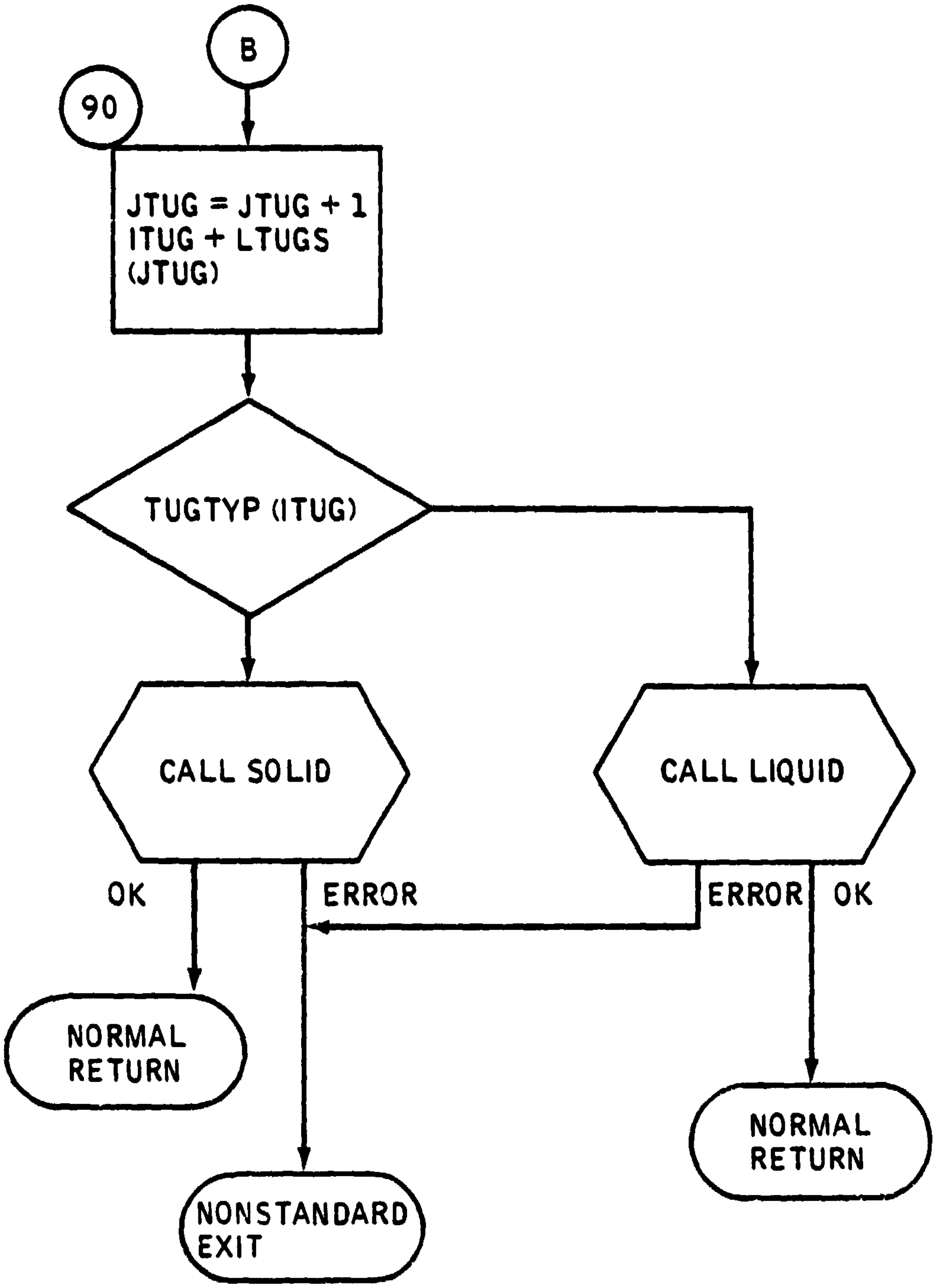
RESTRICTIONS

- Operational

  Subroutines SOLID, LIQUID, PERM, and DLTAV are required.


TSV
PAYLOAD TYPE
A
S
D
R
DWTUG(I) = WT1
DWTUG(I) = -WT1
DWTUG(I) = WT1 - WT
JTUG = 0
JFLAG = .TRUE.
J = ISTUG(1)
L = ICTUG(J)
TALTI = HA(L)
TINCI = INCL(L)
NORBPL
≤ 0
> 0
POINC1 = AMAX1 (28.5, TINC)
POINC = POINC1
DOINC = POINC1
ROINC = POINC1
75
IREUSE
76
DVTI1 = DVEL (POALT, TALTI, POINC, TINCI)
DVTI1 = √C3 + 2GG/(RE + POALT*FT2NM) - √GG/ (RE + POALT*FT2NM)
A
B


A
NTUGPL -1
≤ 0
> 0
DO I = 3 NTUGPL
K = L
J = ISTUG(I)
L = ICTUG(J)
DVTUG(I) = DLTAV (K,L)
NREUSE
= 0
TALTF = HA(L)
TINCF = INCL(L)
DVTF1 = DVEL (TALTF, ROALT, TINCF, ROINC)
NORBPL
> 0
NTUGPL -1
= 0
POINC2 = AMAX1(28.5, TINCF)
POINC 1 = POINC 2
YES
DVTI2 = DVEL (POALT, TALTF, POINC2, TINCF)
DVTF2 = DVEL (TALTF, ROALT, TINCF, POINC2)
JFLAG = .FALSE.
B


B
90
JTUG = JTUG + 1
ITUG + LTUGS
(JTUG)
TUGTYP (ITUG)
CALL SOLID
CALL LIQUID
OK
ERROR
ERROR
OK
NORMAL RETURN
NONSTANDARD EXIT
NORMAL RETURN

SUBROUTINE WINCL

IDENTIFICATION

Name (Title)	- WINCL (Wedge Angle Inclination)
Author, Date	- F. Roth and J. Williams, June 1976
Machine Identification	- UNIVAC 1110 EXEC 8
Source Language	- FORTRAN V

PURPOSE

The purpose of WINCL is to compute the wedge angle between two orbital planes.

USAGE

- Calling Sequence

  CALL WINCL (IOPT)

  Argument:

Parameter name	In/Out	Dimension	Type	Description
IOPT	In	1	I	Initialization flag 1 = initialize 2 = compute angle between orbital planes

- Data In/Out

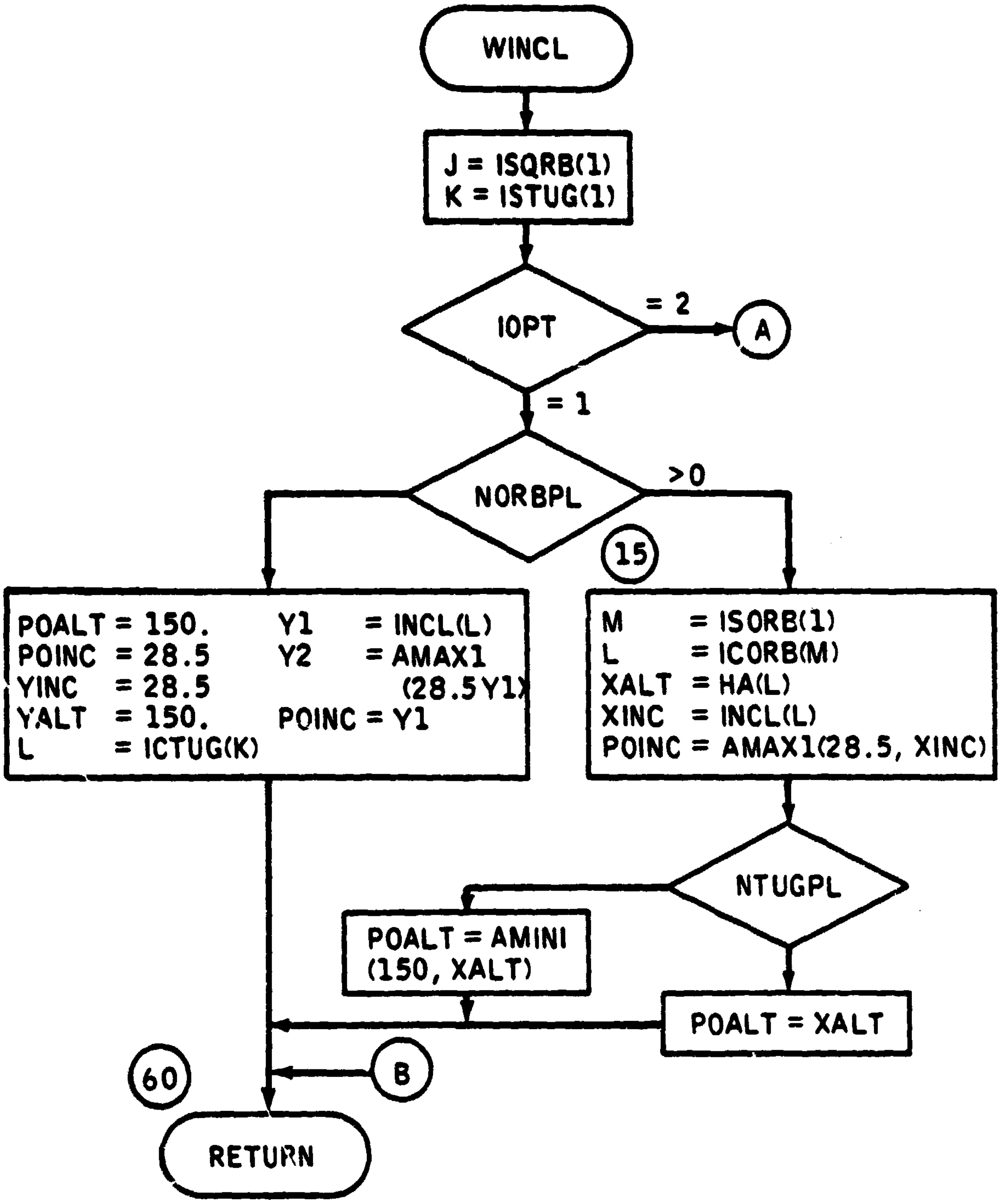
Block name	In	Out
C1	2001-2600 2801-3600 7201-7800	
C2	4,9	
C3	1-8	
C6	1-60 91-105	
C8	9-36	
C10		1,2

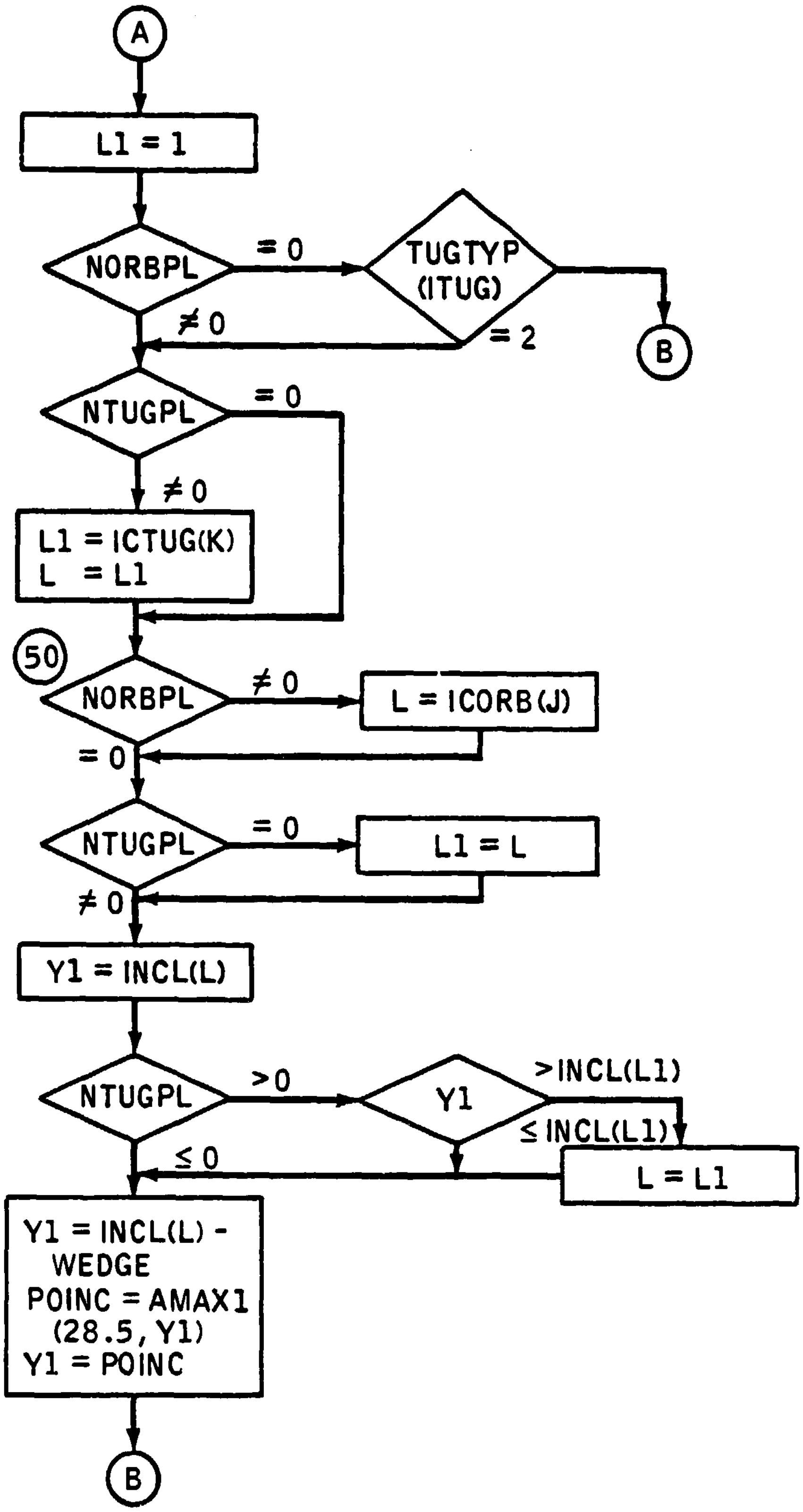
METHOD

Initially, the altitude and inclination of the first orbit are saved. On succeeding passes, the angular difference between the previous and current orbits is calculated. The total orbital change is restricted so that the final orbit is not allowed to go below 28.5^0 inclination.

RESTRICTIONS

None.


WINCL
J = ISQRB(1)
K = ISTUG(1)
IOPT
= 2
A
= 1
NORBPL
>0
15
POALT = 150.
POINC = 28.5
YINC = 28.5
YALT = 150.
L = ICTUG(K)
Y1 = INCL(L)
Y2 = AMAX1 (28.5Y1)
POINC = Y1
M = ISORB(1)
L = ICORB(M)
XALT = HA(L)
XINC = INCL(L)
POINC = AMAX1(28.5, XINC)
NTUGPL
POALT = AMINI (150, XALT)
POALT = XALT
60
B
RETURN


A
L1 = 1
NORBPL
= 0
TUGTYP (ITUG)
≠ 0
= 2
B
NTUGPL
= 0
≠ 0
L1 = ICTUG(K)
L = L1
50
NORBPL
≠ 0
L = ICORB(J)
= 0
NTUGPL
= 0
L1 = L
≠ 0
Y1 = INCL(L)
NTUGPL
> 0
Y1
>INCL(L1)
≤ INCL(L1)
≤ 0
L = L1
Y1 = INCL(L) - WEDGE
POINC = AMAX1 (28.5, Y1)
Y1 = POINC
B

SUBROUTINE WTTEST

IDENTIFICATION

Name (Title) - WTTEST (Weight Test)
Author, Date - John Eggleston, August 1976
Machine Identification - UNIVAC 1110
Source Language - FORTRAN V

PURPOSE

Subroutine WTTEST checks a set of payloads to ensure that they do not exceed the orbiter take-off and landing capabilities.

USAGE

- Calling Sequence

CALL WTTEST (NPL, NDWNPL, IFLY, NEPS, NCREW, LWTU, LWTD)

Arguments:

Parameter name	In/Out	Dimension	Type	Description
NPL	In	1	I	Number of payloads in the combination
NDWNPL	In	1	I	Number of retrieve payloads in the combination
IFLY	In	6	I	Payload ID's in a combination
NEPS	In	1	I	Number of EPS kits for this combination
NCREW	In	1	I	Number of crewmen assigned to the combination
LWTU	Out	4	L	'Weight-up' pass/fail flag for 0-3 OMS kits
LWTD	Out	4	L	'Weight-down' pass/fail flag for 0-3 OMS kits

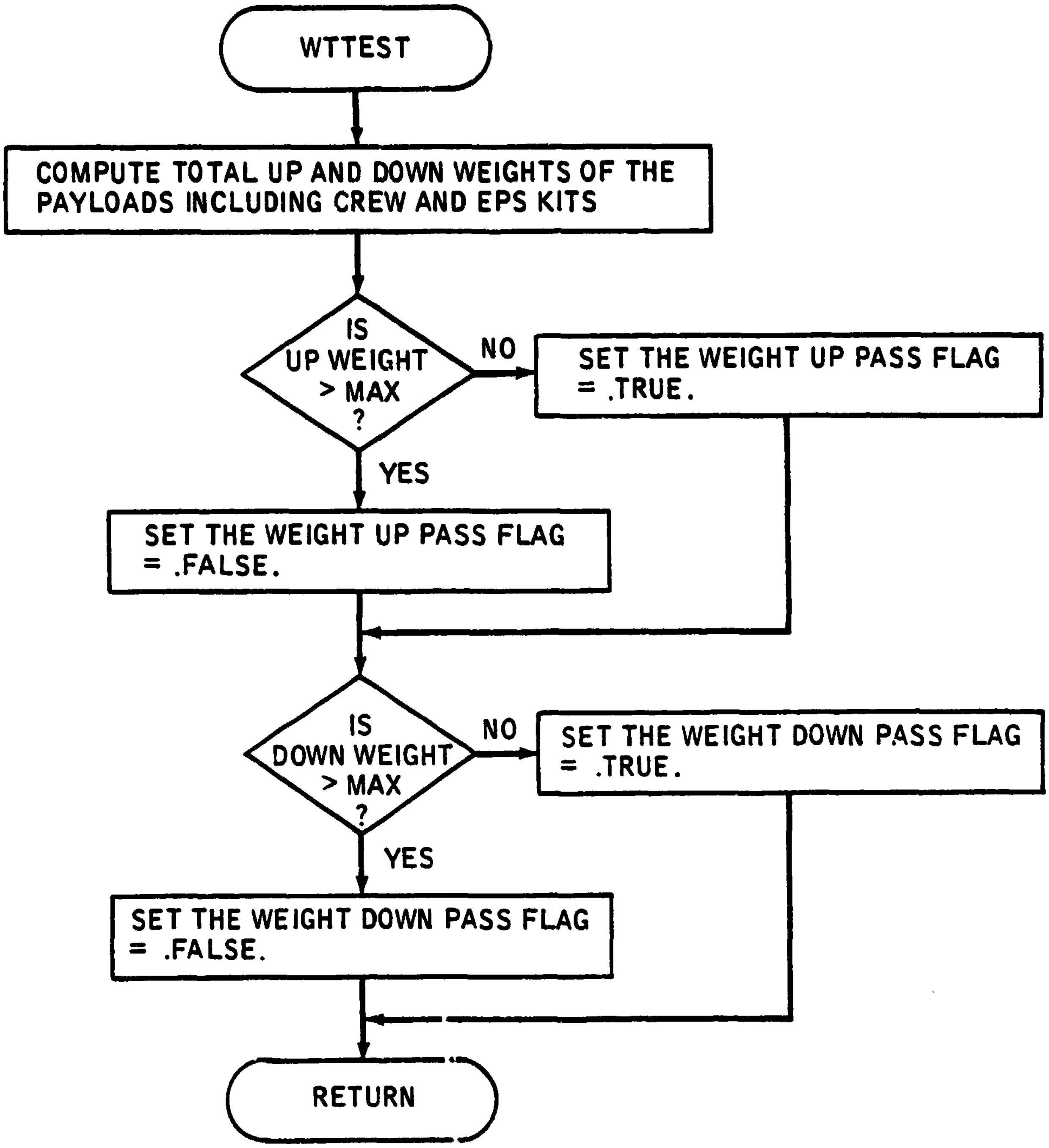
- Data In/Out

Labeled COMMON (refer to the labeled COMMON description section):

Block name	Input	Output
C1	2201-2600	
C32	6-7	
	9-16	

METHOD

WTTEST sums the weight of full EPS kits, crewpersons, and payloads at launch, adds the weight of zero to three full OMS kits, and checks the launch weight constraints. The same is done for landing, except the OMS and the EPS kits are empty.


WTTEST
COMPUTE TOTAL UP AND DOWN WEIGHTS OF THE PAYLOADS INCLUDING CREW AND EPS KITS
IS UP WEIGHT > MAX ?
NO
SET THE WEIGHT UP PASS FLAG = .TRUE.
YES
SET THE WEIGHT UP PASS FLAG = .FALSE.
IS DOWN WEIGHT > MAX ?
NO
SET THE WEIGHT DOWN PASS FLAG = .TRUE.
YES
SET THE WEIGHT DOWN PASS FLAG = .FALSE.
RETURN

5.4 SAMPLE INPUT/OUTPUT

5.4.1 Sample Input

The job stream given below indicates the operations necessary to execute the MPLS. To gain a clean understanding of the input, consult Volume 1- Sample User's Guide.

Card image	Description
@RUN JWLXIA	Run card
@USE SAMPLE., FM3-L71194*SAMPLE	Specifies an internal file name for an external file name
@XQT SAMPLE.SAMPLE	Starts execution
2	Selects the performance vehicle
@ADD SAMPLE.DATAIS	Adds the payload model to the run
1	Selects the display option
7	Requests all displays
2	Allows the selection of an analysis type
1	Selects the MPLS only option
80	Specifies the year of analysis (1980)
10	No data base options
5	Terminate
@FIN	Sign off the system

5.4.2 Sample Output

The printed output generated by the input data described in section 5.4.1 follows.

XXXXXXXXXXXXXXXXXXXXXXXX MISSION MODEL DISPLAY XXXXXXXXXXXXXXXXXXXXXXXX

NO.	PAYLOAD DISCIPLINE	PAYLOAD ID	NAME
1	AG	ADV RELATVTY	ADVANCED RELATIVITY
2	AE	SMM	SOLAR MAXIMUM MISSION
3	AE	SMM RETRIEVE	SOLAR MAXIMUM MISSION RETRIEVE
4	AA	AMPTE	ACTIVE MAGNETOSPHERIC PART TRCR EX
5	AB	SPACE TELE	SPACE TELESCOPE
6	AB	ST RETRIEVE	SPACE TELESCOPE RETRIEVE
7	AB	ST REVISIT	SPACE TELESCOPE REVISIT
8	AC	GAMMA RAY OB	GAMMA RAY OBSERVATORY
9	AC	GRO RETRIEVE	GAMMA RAY OBSERVATORY RETRIEVE
10	AC	GRO REVISIT	GAMMA RAY REVISIT
11	AB	X-RAY OB	X-RAY OBSERVATORY
12	AB	XRO RETRIEVE	X-RAY OBSERVATORY RETRIEVE
13	AB	XRO REVISIT	X-RAY OBSERVATORY REVISIT
14	AA	EUVE	EXTREME ULTRAVIOLET EXPLORER
15	AC	CRO	COSMIC RAY OBSERVATORY
16	AC	CRO RETRIEVE	COSMIC RAY OBSERVATORY RETRIEVE
17	AC	CRO REVISIT	COSMIC RAY OBSERVATORY REVISIT
18	AB	LSO	LARGE SOLAR OBSERVATORY
19	AG	OPEN	ORIGIN OF PLASMAS IN EARTHS NEIGHBD
20	AF	VLBI-A	VERY LONG BASELINE INTERFEROMETER-A
21	AF	VLBI-B	VERY LONG BASELINE INTERFEROMETER-B
22	AH	GALILEO	GALILEO
23	AJ	SOLAR POLAR	SOLAR POLAR
24	AK	VOIR	VENUS ORBITING IMAGING RADAR
25	AK	HALLEY-C FLY	HALLEY COMET FLYBY

73:)

26	AJ	SOLAR PROBE	SOLAR PROBE
27	AH	ASTEROID RDZ	ASTEROID RENDEZVOUS
28	AK	MARS RET 1	MARS SAMPLE RETURN 1
29	AK	MARS RET 2	MARS SAMPLE RETURN 2
30	AK	MARS RET 3	MARS SAMPLE RETURN 3
31	AP	ERBSS	EARTH RADIATION BUDGET SATELLITE SYS
32	AP	ERBSS RET	EARTH RADIATION BUDGET SAT. SYS RETR
33	AQ	LTNG MAPPER	LIGHTNING MAPPER
34	AQ	REG H2O QM	REGIONAL WATER QUALITY MONITOR
35	AQ	STORMS OBS	SEVERE STORMS OBSERVATION SYSTEM
36	AP	SEOS	SYNCHRONOUS ENVIRONMENT OBS SATELLIT
37	AR	HALOGEN OCC	HALLOGEN OCCULTATION
38	AP	COASTAL ZONE	COASTAL ZONE MONITOR
39	AV	WIDE BAND	WIDE BAND
40	AV	ADV MBEAM AR	ADVANCED MULTIBEAM ARRAY
41	AW	RURAL COMM	RURAL COMMUNICATIONS
42	AV	MOBIL COMM	MOBIL COMMUNICATIONS
43	AW	SERCH & RESC	SEARCH AND RESCUE
44	AL	LDEF	LONG DURATION EXPOSURE FACICLTY
45	AL	LDEF RETR	LONG DURATION EXPOSURE FACILITY RETR
46	AM	SLERV	SHUTTLE LAUNCHED ENTRY RESEARCH VEHI
47	AM	SSST	SPACE STRUCTURE SYSTEM TECHNOLOGY EX
48	AN	25KW PWR MOD	25 KW POWER MODULE
49	AM	SCI &APPL MO	SCIENCE AND APPLICATIONS MODULES
50	AM	MAR EXP CARR	MATERIALS EXPERIMENT CARRIER
51	AM	MAT MODULES	MATERIALS MODULES
52	AN	LG SPACE STR	LARGE SPACE STRUCTURES

128:)

NO.	DIAM	HEPS	OXEPS	HA	PLDUR	OPTIME	C3
1	7.	0.	0.	281.	1.	0.	.0000
2	15.	0.	0.	160.	1.	0.	.0000
3	15.	0.	0.	380.	1.	0.	.0000
4	5.	0.	0.	160.	1.	0.	.0000
5	14.	0.	0.	270.	1.	0.	.0000
6	14.	0.	0.	270.	1.	0.	.0000
7	15.	0.	0.	270.	1.	0.	.0000
8	14.	0.	0.	160.	1.	0.	.0000
9	14.	0.	0.	160.	1.	0.	.0000
10	15.	0.	0.	250.	1.	0.	.0000
11	15.	0.	0.	160.	1.	0.	.0000
12	15.	0.	0.	160.	1.	0.	.0000
13	15.	0.	0.	280.	1.	0.	.0000
14	3.	0.	0.	186.	1.	0.	.0000
15	15.	0.	0.	160.	1.	0.	.0000
16	15.	0.	0.	160.	1.	0.	.0000
17	15.	0.	0.	250.	1.	0.	.0000
18	15.	0.	0.	160.	1.	0.	.0000
19	5.	0.	0.	160.	1.	0.	.0000
20	8.	0.	0.	160.	1.	0.	.0000
21	8.	0.	0.	160.	1.	0.	.0000
22	12.	0.	0.	160.	1.	0.	.0000
23	8.	0.	0.	160.	1.	0.	.0000
24	10.	0.	0.	160.	1.	0.	.0000
25	15.	0.	0.	160.	1.	0.	.0000
26	15.	0.	0.	[illegible]	1.	3.	.0000
27	15.	0.	0.	160.	1.	0.	.0000
28	15.	0.	0.	160.	1.	0.	.0000
29	15.	0.	0.	160.	1.	0.	.0000
30	15.	0.	0.	160.	1.	0.	.0000
31	10.	0.	0.	160.	1.	0.	.0000
32	10.	0.	0.	160.	1.	0.	.0000
33	5.	0.	0.	160.	1.	0.	.0000
34	5.	0.	0.	160.	1.	0.	.0000
35	5.	0.	0.	160.	1.	0.	.0000
36	10.	0.	0.	160.	1.	0.	.0000
37	3.	0.	0.	160.	1.	0.	.0000
38	15.	0.	0.	160.	1.	0.	.0000
39	10.	0.	0.	160.	1.	0.	.0000
40	10.	0.	0.	160.	1.	0.	.0000
41	4.	0.	0.	160.	1.	0.	.0000
42	8.	0.	0.	160.	1.	0.	.0000
43	4.	0.	0.	160.	1.	0.	.0000
44	15.	0.	0.	225.	1.	0.	.0000
45	15.	0.	0.	225.	1.	0.	.0000
46	8.	0.	0.	160.	1.	0.	.0000
47	15.	0.	0.	160.	1.	0.	.0000
48	15.	0.	0.	250.	1.	0.	.0000
49	15.	0.	0.	250.	1.	0.	.0000
50	15.	0.	0.	250.	1.	0.	.0000
51	15.	0.	0.	250.	1.	0.	.0000
52	15.	0.	0.	250.	1.	0.	.0000
53	15.	0.	0.	250.	1.	0.	.0000

4021>

NO.	INCL	RCS	CG	LAUNCH LENGTH, FT.	LAUNCH WT. INCL. ADAPTER	ADAPTER WT., LB.	PMT
1	90.0	.0	6.3	11.8	1633.6	1433.0	D
2	28.5	.0	10.8	21.7	11700.0	4500.0	D
3	28.5	.0	10.8	21.7	4500.0	11700.0	R
4	28.5	.0	5.6	11.3	6578.0	2250.0	D
5	28.5	.0	21.5	43.0	23531.3	20944.0	D
6	28.5	.0	21.5	43.0	.0	20944.0	R
7	28.5	.0	5.0	10.0	5000.0	5000.0	S
8	28.5	.0	16.8	33.6	25700.0	4000.0	D
9	28.5	.0	16.8	33.6	4000.0	19400.0	R
10	28.5	.0	5.0	10.0	5000.0	5000.0	S
11	28.5	.0	25.3	50.6	31700.0	4000.0	R
12	28.5	.0	25.3	50.6	4000.0	25400.0	R
13	28.5	.0	5.0	10.0	5000.0	5000.0	S
14	28.5	.0	1.5	3.0	604.2	530.0	D
15	28.5	.0	17.3	34.6	25700.0	4000.0	D
16	28.5	.0	17.3	34.6	4000.0	19400.0	R
17	28.5	.0	5.0	10.0	5000.0	5000.0	S
18	28.5	.0	29.8	59.6	32700.0	4000.0	D
19	28.5	.0	5.6	11.3	7618.0	2250.0	D
20	28.5	.0	13.8	27.5	54728.0	6924.0	D
21	28.5	.0	13.8	27.6	54728.0	6924.0	D
22	28.5	.0	22.3	44.7	65000.0	6591.0	D
23	28.5	.0	18.5	37.0	62607.0	6591.0	D
24	28.5	.0	20.5	41.0	61033.0	6924.0	D
25	28.5	.0	24.0	48.0	65000.0	6591.0	D
26	28.5	.0	24.0	48.0	62657.0	6591.0	D
27	28.5	.0	24.0	48.0	65000.0	6591.0	D
28	28.5	.0	24.0	48.0	65000.0	6591.0	D
29	28.5	.0	24.0	48.0	65000.0	6591.0	D
30	28.5	.0	24.0	48.0	65000.0	6591.0	D
31	56.0	.0	9.8	19.7	10000.0	4500.0	D
32	56.0	.0	9.8	19.7	4500.0	9833.0	R
33	28.5	.0	5.6	11.3	7318.0	2250.0	D
34	28.5	.0	5.6	11.3	7318.0	2250.0	D
35	28.5	.0	5.6	11.3	7318.0	2250.0	D
36	28.5	.0	11.4	22.8	15275.0	3500.0	D
37	56.0	.0	3.7	7.5	627.0	550.0	D
38	28.5	.0	11.2	22.5	16160.0	3800.0	D
39	28.5	.0	13.7	27.5	16360.0	3800.0	D
40	28.5	.0	13.7	27.5	16360.0	3800.0	D
41	28.5	.0	5.4	10.8	7473.0	2250.0	D
42	28.5	.0	13.7	27.5	16396.0	3800.0	D
43	28.5	.0	5.4	10.8	7468.0	2250.0	D
44	28.5	.0	15.0	30.0	22500.0	20000.0	D
45	28.5	.0	15.0	30.0	.0	20000.0	R
46	28.5	.0	5.0	10.0	5700.0	5000.0	D
47	28.5	.0	5.0	10.0	9120.0	8000.0	D
48	28.5	.0	30.0	60.0	30147.5	27000.0	D
49	28.5	.0	5.0	10.0	16950.0	15000.0	D
50	28.5	.0	5.0	10.0	11400.0	10000.0	D
51	28.5	.0	5.0	10.8	13620.0	12000.0	D

586!>

FLIGHTS PER YEAR

NO.	1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993
1	0	0	0	0	0	0	0	0	0	0	2	0	0
2	0	0	0	1	0	0	1	0	1	0	1	1	0
3	0	1	0	0	0	1	0	1	0	1	0	0	0
4	0	1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	1	0	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0	1	0	0	0	0
7	0	0	0	0	1	1	1	0	0	0	1	0	0
8	0	0	0	0	1	0	0	1	0	0	1	0	0
9	0	0	0	0	0	0	1	0	0	1	0	0	0
10	0	0	0	0	0	0	0	0	1	0	0	1	0
11	0	0	0	0	0	0	1	0	0	0	0	1	0
12	0	0	0	0	0	0	0	0	0	0	1	0	0
13	0	0	0	0	0	0	0	0	1	1	0	0	0
14	0	0	0	0	1	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	1	0	0	1	1	0	0
16	0	0	0	0	0	0	0	0	0	1	0	0	0
17	0	0	0	0	0	0	0	1	1	0	0	1	0
18	0	0	0	0	0	0	0	0	0	1	0	0	0
19	0	0	0	0	1	1	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	1	0	0	0	0	0
21	0	0	0	0	0	0	0	0	1	0	0	0	0
22	0	1	0	0	0	0	0	0	0	0	0	0	0
23	0	0	1	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	1	0	0	0	0	0	0	0	0
25	0	0	0	0	1	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	1	0	0	0	0	0	0
27	0	0	0	0	0	0	0	1	0	0	0	0	0
28	0	0	0	0	0	0	0	1	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	1	0	0	0
30	0	0	0	0	0	0	0	0	0	1	0	0	0
31	0	1	0	0	0	1	0	0	0	1	0	0	0
32	0	0	0	0	1	0	0	0	1	0	0	0	0
33	0	0	0	0	1	0	0	0	0	0	0	0	0
34	0	0	0	0	0	1	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	1	0	0	0	0	0
36	0	0	0	0	0	0	0	0	1	0	0	1	0
37	0	3	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	1	0	0	0	1	0	0	0
39	0	0	0	1	0	0	0	1	0	0	0	1	0
40	0	1	0	0	0	1	0	0	0	1	0	0	0
41	0	0	0	0	0	0	0	0	0	0	1	0	0
42	0	0	0	0	0	0	0	0	0	1	0	0	0
43	0	1	1	1	0	0	0	0	0	0	0	0	0
44	0	1	1	0	1	1	0	1	1	0	1	0	0
45	0	0	1	1	1	0	1	1	0	1	0	1	0
46	0	0	0	0	1	1	1	0	0	0	0	0	0
47	0	0	0	0	0	1	1	0	0	0	0	0	0
48	0	0	0	1	0	0	0	0	0	0	0	0	0
49	0	0	0	1	1	1	1	1	1	1	1	1	0
50	0	0	0	1	1	0	0	1	0	0	1	0	0
51	0	0	0	0	0	1	2	1	2	2	1	2	0

732:>

ORIGINAL PAGE
OF POOR QUALITY

```
 SEQUENCE NO.,PAYLOAD ID,PMT
   1  57 3      2  59 3      3  61 3      4  62 3      5  63 3      6  64 3
   7  80 3      8  82 3      9  92 1

DUPLICATED PAYLOADS
 PAYLOAD ID,TIMES DUPLICATED
  61  2       63  2       80  2
 SEQ,PAYLOAD NO,TUG
      1 57  0         2 59  0         3 61  0         4 62  0         5 63  0
      6 64  0         7 80  0         8 82  0         9 92  0

FLT. NO.     1     LAUNCH SITE: ETR

 PAYLOADS:    GOES
                 57
 SHUTTLE SEQUENCE      57-D
 ALTITUDE               160.
 INCLINATION            28.5
 TOTAL LENGTH UP:  22.     TOTAL WEIGHT UP: 13994.0
 PAYLOAD MARGIN: 32000.    LOAD FACTOR:  .21529
 SHUTTLE DELTAV:   581.

FLT. NO.     2     LAUNCH SITE: ETR

 PAYLOADS:    INTELSAT V
                 59
 SHUTTLE SEQUENCE      59-D
 ALTITUDE               160
 INCLINATION            28.5
 TOTAL LENGTH UP:  32.     TOTAL WEIGHT UP: 16343.0
 PAYLOAD MARGIN: 32000.    LOAD FACTOR:  .25143
 SHUTTLE DELTAV:   581.

FLT. NO.     3     LAUNCH SITE: ETR

 PAYLOADS:    TDRSS/WESTAR
               61001
 SHUTTLE SEQUENCE      61-D
 ALTITUDE               160.
 INCLINATION            28.5
 TOTAL LENGTH UP:  35.     TOTAL WEIGHT UP: 57533.0
 PAYLOAD MARGIN:  7367.    LOAD FACTOR:  .88666
 SHUTTLE DELTAV:   581.

FLT. NO.     4     LAUNCH SITE: ETR

 PAYLOADS:    TDRSS/WESTAR
               61002
 SHUTTLE SEQUENCE      61-D
 ALTITUDE               160.
 INCLINATION            28.5
 TOTAL LENGTH UP:  35.     TOTAL WEIGHT UP: 57633.0
 PAYLOAD MARGIN:  7367.    LOAD FACTOR:  .88666
 SHUTTLE DELTAV:   581.

899:>
```

```
FLT. NO.    5     LAUNCH SITE: ETR

 PAYLOADS:    RCA
                  62
 SHUTTLE SEQUENCE      62-D
 ALTITUDE               160.
 INCLINATION            28.5
 TOTAL LENGTH UP:  18.      TOTAL WEIGHT UP:  9375.0
 PAYLOAD MARGIN: 32000.     LOAD FACTOR:   .14423
 SHUTTLE DELTAV:    581.

FLT. NO.    6     LAUNCH SITE: ETR

 PAYLOADS:    SBS
               63001
 SHUTTLE SEQUENCE      63-D
 ALTITUDE               160.
 INCLINATION            28.5
 TOTAL LENGTH UP:  21.      TOTAL WEIGHT UP:  9375.0
 PAYLOAD MARGIN: 32000.     LOAD FACTOR:   .14423
 SHUTTLE DELTAV:    581.

FLT. NO.    7     LAUNCH SITE: ETR

 PAYLOADS:    SBS
               63002
 SHUTTLE SEQUENCE      63-D
 ALTITUDE               160.
 INCLINATION            28.5
 TOTAL LENGTH UP:  21.      TOTAL WEIGHT UP:  9375.0
 PAYLOAD MARGIN: 32000.     LOAD FACTOR:   .14423
 SHUTTLE DELTAV:    581.

FLT. NO.    8     LAUNCH SITE: ETR

 PAYLOADS:    SYNCOM 1V
                  64
 SHUTTLE SEQUENCE      64-D
 ALTITUDE               160.
 INCLINATION            28.5
 TOTAL LENGTH UP:  12.      TOTAL WEIGHT UP: 13844.2
 PAYLOAD MARGIN: 32000.     LOAD FACTOR:   .21299
 SHUTTLE DELTAV:    581.

FLT. NO.    9     LAUNCH SITE: ETR

 PAYLOADS:    TELESAT
               80001
 SHUTTLE SEQUENCE      80-D
 ALTITUDE               160.
 INCLINATION            28.5
 TOTAL LENGTH UP:  18.      TOTAL WEIGHT UP:  8521.0
 PAYLOAD MARGIN: 32000.     LOAD FACTOR:   .13109

 SHUTTLE DELTAV:    581.

954:>
```

FLT. NO.   1   LAUNCH SITE: ETR

PAYLOADS: GOES
                 57
SHUTTLE SEQUENCE   57-D
ALTITUDE   160.
INCLINATION   28.5
TOTAL LENGTH UP: 22.   TOTAL WEIGHT UP: 13984.0
PAYLOAD MARGIN: 32000.   LOAD FACTOR: .81529
SHUTTLE DELTAV: 581.

FLT. NO.   3   LAUNCH SITE: ETR

PAYLOADS: TDRSS/WESTAR
                 61001
SHUTTLE SEQUENCE   61-D
ALTITUDE   160.
INCLINATION   28.5
TOTAL LENGTH UP: 35.   TOTAL WEIGHT UP: 57633.0
PAYLOAD MARGIN: 7367.   LOAD FACTOR: .88666
SHUTTLE DELTAV: 581.

FLT. NO.   4   LAUNCH SITE: ETR

PAYLOADS: TDRSS/WESTAR
                 61002
SHUTTLE SEQUENCE   61-D
ALTITUDE   160.
INCLINATION   28.5
TOTAL LENGTH UP: 35.   TOTAL WEIGHT UP: 57633.0
PAYLOAD MARGIN: 7367.   LOAD FACTOR: .88666
SHUTTLE DELTAV: 581.

FLT. NO.   12   LAUNCH SITE: ETR

PAYLOADS: PLA SL1 LM+P
                 92
SHUTTLE SEQUENCE   92-A
ALTITUDE   135.
INCLINATION   57.0
TOTAL LENGTH UP: 60.   TOTAL LENGTH DOWN: 60.
TOTAL WEIGHT UP: 35000.0   TOTAL WEIGHT DOWN: 32000.0
PAYLOAD MARGIN: 0.   LOAD FACTOR: 1.00000
SHUTTLE DELTAV: 466.

FLT. NO.   20   LAUNCH SITE: ETR

PAYLOADS: INTELSAT V   SBS
                 59   63002
SHUTTLE SEQUENCE   59-D   63-D
ALTITUDE   160.   160.
INCLINATION   28.5   28.5
TOTAL LENGTH UP: 53.   TOTAL WEIGHT UP: 25718.0
PAYLOAD MARGIN: 32000.   LOAD FACTOR: .39568
1704?)

```
FLT. NO.   40     LAUNCH SITE: ETR

 PAYLOADS:   RCA              SBS              TELESAT
                   62    63001     80001
 SHUTTLE SEQUENCE      62-D        63-D        80-D
 ALTITUDE              160.         160.        160.
 INCLINATION           28.5        28.5        28.5
 TOTAL LENGTH UP:  57.     TOTAL WEIGHT UP: 27271.0
 PAYLOAD MARGIN: 38000.    LOAD FACTOR:  .41955
 SHUTTLE DELTAV:   581.

FLT. NO.   48     LAUNCH SITE: ETR

 PAYLOADS:   SYNCOM 1V       TELESAT          INSAT-INDIA
                   64    80002         82
 SHUTTLE SEQUENCE      64-D        80-D        82-D
 ALTITUDE              160.         160.        160.
 INCLINATION           28.5        28.5        28.5
 TOTAL LENGTH UP:  51.     TOTAL WEIGHT UP: 31740.2
 PAYLOAD MARGIN: 32000.    LOAD FACTOR:  .48831
 SHUTTLE DELTAV:   581.

INPUT OPTION :
 STATISTICS FOR CURRENT FLIGHT SCHEDULE

       AVERAGE NUMBER OF PAYLOADS PER FLIGHT = 1.71
       TOTAL NUMBER OF TUGS REQUIRED =    0
       TOTAL NUMBER OF INITIAL OMS KITS REQUIRED =     0
       TOTAL NUMBER OF SECOND AND THIRD OMS KITS REQUIRED =     0

DO YOU WANT ANOTHER SCHEDULE ?
    1: YES.
    0: NO .

SELECT AN OPTION: ( 5 TO TERMINATE )
 RUN FINISHED NORMALLY
@BRKPT PRINTS
EOF:1743 SCAN:37
@:>
```

XXXXXXXXXXXXXXXXXXXX STATISTICAL ANALYSIS FOR 1981 XXXXXXXXXXXXXXXXXXXX

TOTAL NUMBER OF COMBINATIONS GENERATED: 134
NUMBER OF FEASIBLE COMBINATIONS: 48
NUMBER OF INFEASIBLE COMBINATIONS: 86

TOTAL ELAPSED TIME: 854
(ALL TIMES ARE IN MILLISECONDS)
AVERAGE TIME PER FEASIBLE COMBINATION: 17
AVERAGE TIME PER GENERATED COMBINATION: 6

DO YOU WANT TO STORE THE FEASIBLE COMBINATION DATA?
0: NO
1: YES

E R R O R   S T A T I S T I C S

0 FAILED : MISSION TYPE NOT ALLOWED
6 FAILED : UP WEIGHT CONSTRAINT
0 FAILED : DOWN WEIGHT CONSTRAINT
0 FAILED : NEEDED DEDICATED TUG
0 FAILED : NUMBER OF TUG PAYLOADS > 3
0 FAILED : LENGTH & WEIGHT CONSTRAINT
36 FAILED : UP LENGTH > BAY LENGTH
0 FAILED : DOWN LENGTH > BAY LENGTH
0 FAILED : DISCIPLINE MIX
0 FAILED : SEQUENCE DEPENDENT TESTS
0 FAILED : RCS WEIGHT > CAPACITY
44 FAILED : REDUNDANT PAYLOAD
0 FAILED : INCLINATION RANGE > .5

SPECIFY VALUE INDEX OF FLIGHTS TO BE USED IN TRAFFIC MODEL SELECTION

1981 OCCURRENCE TABLE

	PAYLOAD	FEASIBLE COMBINATIONS											
1)	57	1	13	14	15	16	17	18	34	35	36	37	38
2)	59	2	13	19	20	21	22	23					
3)	61001	3											
4)	61002	4											
5)	62	5	14	19	24	25	26	27	34	35	39	40	41
		42	43	44									
6)	63001	6	15	24	29	36	40	45	47				
7)	63002	7	20	28	30	39	41	46					
8)	64	8	16	21	25	28	31	32	34	36	37	38	39
		42	43	45	46	48							
9)	80001	9	17	26	31	35	40	44	47				
10)	80002	10	22	29	33	37	42	45	48				
11)	82	11	18	23	27	30	32	33	38	41	43	44	46
		47	48										
12)	92	12											

SPECIFY SOLUTION STRATEGY FOR TRAFFIC MODEL SOLUTION
TRAFFIC MODEL CONTAINS THE FOLLOWING 7 MISSIONS
1 3 4 12 20 40 48
THE SELECTED TRAFFIC MODEL VALUE IS 7
DO YOU WISH TO SEE INFORMATION ON THESE MISSIONS?
1642:>

6. REFERENCES

1. Babb, G. R.: Space Shuttle Performance Capabilities. JSC IN 71-FM-350, Sept. 1971.

2. Gonzales, L.: Validation of the Mission Payloads (MPLS) of the Scheduling Algorithm for Mission Planning and Evaluation (SAMPLE). JSC, FM33 (75-74), May 1975.

3. IDSD Procedures Manual - MSC, Part 20 (Revision 0). MSC, Oct. 1973.

4. Proposed Design of a Center of Gravity and Geometric Fit Constraint Check in SAMPLE. JSC Memorandum No. FM34 (75-122), Aug. 20, 1975.