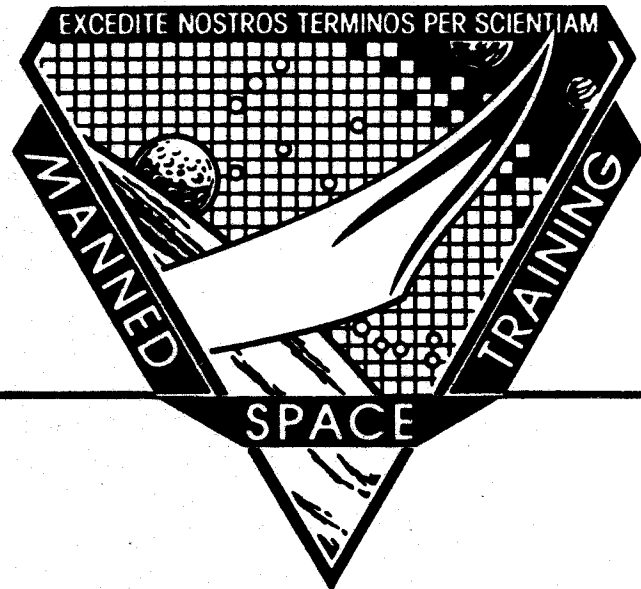


MCC Trajectory and Orbit Determination Software Workbook

MCC OD SOFT 2102

Mission Operations Directorate

April 1991



NASA

National Aeronautics and
Space Administration

Lyndon B. Johnson Space Center
Houston, Texas

211

Contents

Section		Page
1	<u>INTRODUCTION</u>	1-1
2	<u>TRAJECTORY PREDICTION PROCESSORS OVERVIEW</u>	2-1
2.1	MODELS USED FOR TRAJECTORY PREDICTION IN THE MISSION	
	CONTROL CENTER TRAJECTORY SOFTWARE	2-1
2.1.1	<u>Central Force Model</u>	2-1
2.1.2	<u>Sun/Moon Perturbations Model</u>	2-1
2.1.3	<u>Geopotential Model</u>	2-3
2.1.4	<u>Solar Radiation Pressure</u>	2-3
2.1.5	<u>Atmospheric Density Model</u>	2-3
2.1.6	<u>Drag Model</u>	2-3
2.1.7	<u>Vent Model</u>	2-3
2.2	ROUTINES THAT SUPPLY INFORMATION TO THE ENCKE	
	EFFECTIVE FORCES ROUTINE	2-4
2.2.1	<u>ENCKE Effective Forces Routine (ECENEF)</u>	2-4
2.2.2	<u>Body Attitude Determination Routine (ECENBY)</u>	2-4
2.2.3	<u>Vehicle Weight and Configuration Routine (ECENWT)</u>	2-4
2.2.4	<u>Low-Level Thrust Determination Routine (ECENVT)</u>	2-6
2.2.5	<u>Fourth-Order Runge-Kutta Routine (ECENER)</u>	2-12
2.2.6	<u>Atmosphere Modeling Routines</u>	2-13
2.2.7	<u>Gravitational Perturbation Routines</u>	2-15
2.3	PROCESSORS SUPPLYING INFORMATION TO THE NUMERICAL	
	INTEGRATION PREPROCESSOR (EMPREPRO)	2-17
2.3.1	<u>Attitude Timeline User Service</u>	2-17
2.3.2	<u>Vent Timeline User Service</u>	2-22
2.3.3	<u>Weight Gain/Loss User Service Routine (ELWGTA2)</u>	2-25
2.4	FREE-FLIGHT NUMERICAL INTEGRATOR (ECENCKE)	2-28
2.4.1	<u>System and Trajectory Parameters Used in the Process</u>	2-30
2.4.2	<u>Supporting Routines</u>	2-36
2.5	POWERED-FLIGHT NUMERICAL INTEGRATOR (ECPFNI)	2-53
2.5.1	<u>System and Other Parameters Used</u>	2-54
2.5.2	<u>Supporting Routines</u>	2-62
2.6	NUMERICAL INTEGRATION SUPERVISOR (EMSMISS)	2-66
2.6.1	<u>Description of Interfaces</u>	2-67
2.7	ANALYTIC EPHEMERIS GENERATOR (ECPAEG)	2-67
2.7.1	<u>System and Other Parameters Used in the Process</u>	2-69
2.7.2	<u>Supporting Routines</u>	2-71
2.8	ROUTINES PROVIDING ROTATION OF COORDINATES	2-73
2.8.1	<u>Coordinate System Transformation Processor</u>	2-73
2.8.2	<u>Rotation, Nutation, and Precession Matrix Load Supervisor (ECLOAD)</u>	2-75

Section		Page
2.8.3	<u>Rotation, Nutation, and Precession Matrix Generator (ECLONP)</u>	2-75
2.8.4	<u>Universal Timekeeping Processor (UVTIME)</u>	2-77
3	<u>ORBIT DETERMINATION PROCESSOR/MODULES</u>	3-1
3.1	MANUAL ENTRY DEVICE DECODER	3-1
3.2	PUSHBUTTON INDICATOR CONTROL PROCESSOR	3-3
3.3	LOW-SPEED INPUT PROCESSOR	3-4
3.4	VEHICLE BIAS TABLE MAINTENANCE PROCESSOR	3-9
3.5	VEHICLE DATA TABLE MAINTENANCE PROCESSOR	3-10
3.5.1	<u>Assign a Batch Number</u>	3-10
3.5.2	<u>Store a Batch</u>	3-10
3.5.3	<u>Assign a Transmitter</u>	3-11
3.5.4	<u>Exclude/Include for Batch-to-Batch/Superbatch</u>	3-11
3.5.5	<u>Delete Batches</u>	3-11
3.5.6	<u>Split Around Maneuvers</u>	3-11
3.5.7	<u>Split Around a Specified Time</u>	3-11
3.5.8	<u>Copy Batches</u>	3-11
3.5.9	<u>Change the Link Identification on a Copied Batch</u>	3-12
3.5.10	<u>Apply a Time Bias</u>	3-12
3.5.11	<u>Fetch a Batch for Differential Correction</u>	3-12
3.6	DIFFERENTIAL CORRECTION EXECUTIVE	3-13
3.7	VECTOR TABLE MAINTENANCE AND CONTROL PROCESSOR	3-14
3.7.1	<u>The VAT</u>	3-14
3.7.2	<u>VTMC Services</u>	3-15
3.8	COVARIANCE STORAGE TABLE MAINTENANCE PROCESSOR	3-16
3.8.1	<u>Functions of the CSTM</u>	3-16
3.9	STATE TRANSITION MATRIX MODULE	3-17
3.10	RESIDUAL AND STATISTICS PROCESSOR	3-18
3.11	A PRIORI SETUP PROCESSOR	3-19
3.12	CONVERGENCE PROCESSOR	3-21
3.13	DATA EDITING PROCESSOR	3-23
3.14	DIFFERENTIAL CORRECTION MODULE	3-23
3.15	OBSERVATION COMPUTATION MODULE	3-27
3.16	MEASUREMENT PARTIAL DERIVATIVE MODULE	3-29
3.17	BB PROCESSOR FLOW	3-30
4	<u>MISCELLANEOUS TRAJECTORY PROCESSORS</u>	4-1
4.1	PREDICTED SATELLITE ACQUISITION DISPLAY PROCESSOR	4-1
4.1.1	<u>Predicted Site Processor</u>	4-1
4.1.2	<u>Special Stations Contacts Generator</u>	4-1
4.1.3	<u>Generalized Stations Contacts Generator</u>	4-1
4.1.4	<u>Horizon Crossing</u>	4-2

Section		Page
4.1.5	<u>Terrain Masking</u>	4-3
4.2	SATELLITE ACQUISITION PROCESSOR	4-3
4.2.1	<u>Supporting Routines</u>	4-5
4.3	STATION CONTACTS GENERATOR (EMGSTGEN)	4-5
4.3.1	<u>Supporting Routines</u>	4-6
4.4	DELTA-TIME PROCESSOR	4-8
4.5	BVENT/LVENT PROCESSOR (B/L VENT)	4-11
4.6	CHECKOUT MONITOR DISPLAY PROCESSOR (EMDCHECK)	4-13
4.7	RELAY GEOMETRY DISPLAY PROCESSOR	4-15
4.8	VECTOR COMPARE PROCESSOR	4-16
4.8.1	<u>Input Queues</u>	4-17
4.8.2	<u>Vector Fetch</u>	4-17
4.8.3	<u>Vector Integration</u>	4-18
 Appendix		
A	<u>PROCESSOR LOCATION WITHIN CCS TRAJECTORY OPERATIONS DESIGN SPECIFICATIONS DOCUMENT</u>	A-1
B	<u>BIBLIOGRAPHY</u>	B-1
C	<u>ACRONYMS AND ABBREVIATIONS</u>	C-1
D	<u>STUDY QUESTIONS</u>	D-1
	NON-CRITICAL QUESTIONS	D-1
	CRITICAL QUESTIONS	D-3
	COMPLEX QUESTIONS	D-5
E	<u>ANSWERS TO STUDY QUESTIONS</u>	E-1
	ANSWERS TO NON-CRITICAL QUESTIONS	E-1
	ANSWERS TO CRITICAL QUESTIONS	E-4
	ANSWERS TO COMPLEX QUESTIONS	E-6

FIGURES

Figures		Page
2-1	Models used for trajectory prediction in the MCC software	2-2
2-2	ENCKE effective forces routine interface	2-5
2-3	Numerical integration processor interface	2-18
2-4	Attitude timeline/computation processor data flow	2-22
2-5	Vent timeline maintenance and control processor data flow	2-24
2-6	Weight gain/loss and cargo bay door table maintenance and control processor data flow	2-26
2-7	Free-flight numerical integrator (ECENCKE) interface	2-37
2-8	Difference table	2-48
2-9	Powered-flight numerical integrator (ECPFNI)	2-63
2-10	Numerical integration supervisor (EMSMISS) interface	2-68
2-11	Analytic ephemeris generator (ECPAEG)	2-71
2-12	Coordinate system transformation processor data flow	2-74
2-13	Routines providing rotation of coordinates	2-76
2-14	UVTIME data flow	2-78
3-1	Orbit determination processor/module interface	3-2
3-2	Example BB processor flow	3-31
4-1	PSAD processor	4-2
4-2	Satellite acquisition processor (EMPSAP)	4-4
4-3	Station contacts generator (EMGSTGEN)	4-7
4-4	DTP data flow	4-9
4-5	BVENT/LVENT display processor data flow	4-12
4-6	Checkout monitor display processor (EMDCHECK)	4-14
4-7	Relay geometry processor data flow	4-16
4-8	Vector compare processor data flow	4-17

SECTION 1

INTRODUCTION

The purpose of this workbook is to familiarize the reader with the major routines located within the Mission Control Center (MCC) ground navigation software that are responsible for orbit trajectory determination.

The information is presented in three parts. The first part consists of a description of the routines that are involved in trajectory prediction processing. The second part contains a description of the processors/modules used in orbit determination (OD). An example of OD which is used during real-time mission support is batch-to-batch (BB) processing. The third part consists of a description of routines that are not directly involved in the orbit trajectory determination process, but they have been included in this document because they are very important during ground navigation mission support/console operation.

This workbook does not include a breakdown of the software to the code level but consists of a description of each routine and how it interacts with other routines to complete its task. For a more indepth study of MCC ground navigation software, the reader should study reference (1) in appendix B of this document.

As with most workbooks, a list of study questions has been placed at the end for the benefit of the reader. The reader should be prepared to answer all these questions before meeting with his or her training monitor for workbook signoff.

Since most of the information in this workbook is based on Command and Control System (CCS) Trajectory Operations Design Specifications, an index has been provided for reference in appendix A.

SECTION 2

TRAJECTORY PREDICTION PROCESSORS OVERVIEW

This section consists of an overview of the modules used in the trajectory prediction process (fig. 2-1), followed by a description of the routines that supply this information.

After reading this section, you should know the names of the models used for trajectory prediction in the MCC trajectory software and what they are used for.

2.1 MODELS USED FOR TRAJECTORY PREDICTION IN THE MISSION CONTROL CENTER TRAJECTORY SOFTWARE

Since you have already read TRJ INT METH 2102, you are familiar with the different ways of integrating the equations of motion. This section will give you an overview of the forces used to compute the total acceleration of the vehicle in question.

Refer to figure 2-1 while reading the supporting paragraphs. This information is the core of section 2.

2.1.1 Central Force Model

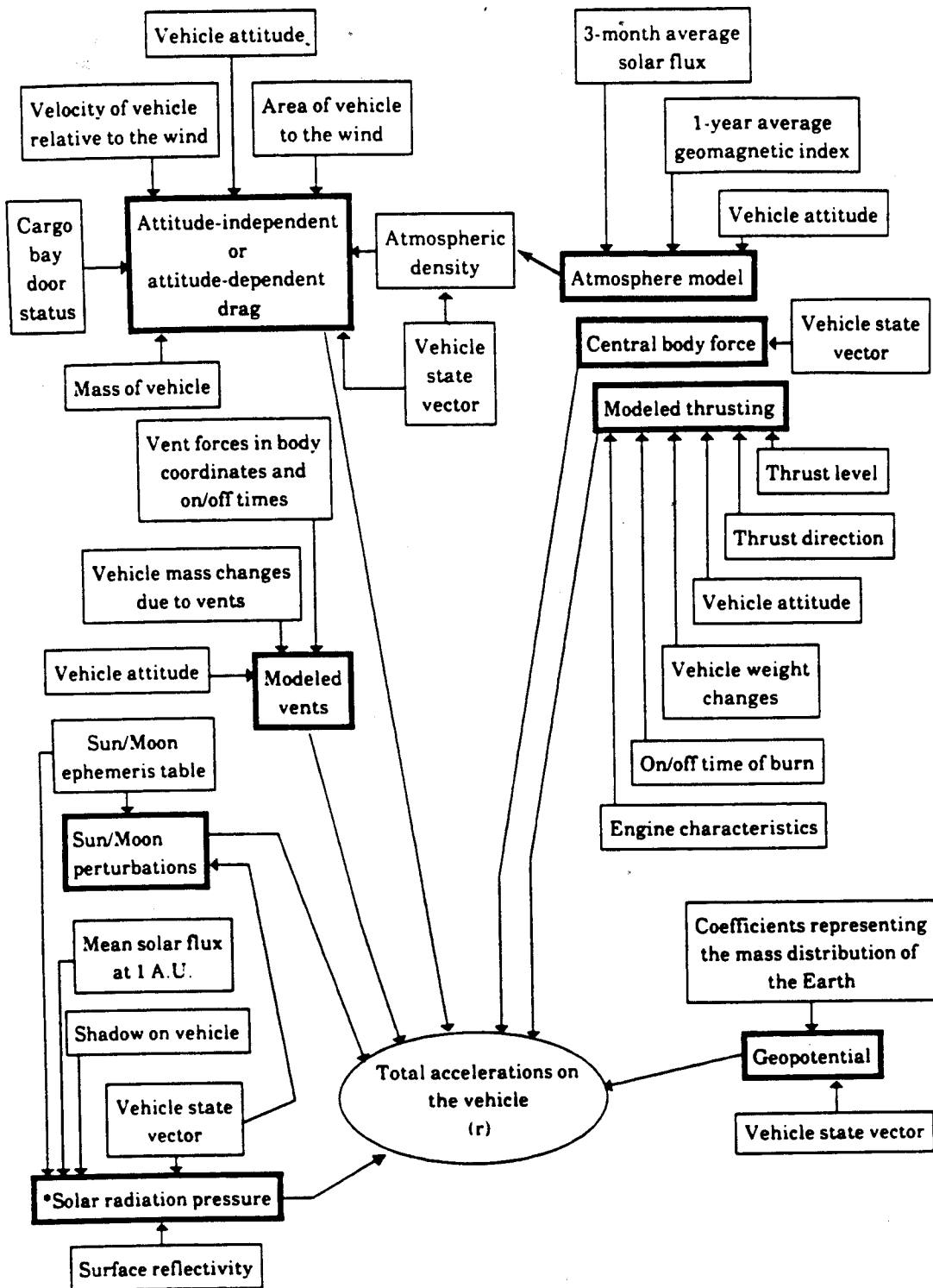
The central force in the two-body equation is given by; $\vec{r} = \frac{-\mu \vec{r}}{r^3}$

2.1.2 Sun/Moon Perturbations Model

The gravitational accelerations caused by the Sun and Moon are incorporated.

2.1.3 Geopotential Model

Since the Earth is not a perfect sphere, a spacecraft will experience variable gravitational accelerations as it passes over the different mass distributions of the Earth. The geopotential model simulates the gravitational accelerations caused by the irregular shape of the Earth.



*Solar radiation pressure will be modeled for the TDRS satellites only.
 Figure 2-1. - Models used for trajectory prediction in the MCC software.

2.1.4 Solar Radiation Pressure

Using the tracking and data relay satellite (TDRS) system to track the orbiter requires a highly accurate estimate of the location of the satellites. To predict the trajectory of the TDRS, the accelerations on the TDRS due to solar radiation pressure are modeled. The solar radiation pressure model is not in the MCC trajectory software as of the publishing date of this workbook. The planned implementation date for the solar radiation pressure model is October 1991.

2.1.5 Atmospheric Density Model

It is important to have a good model of the density of the atmosphere because the density of the atmosphere is used by the drag model. Atmospheric density is a function of altitude, vehicle position, Sun position, average solar flux, and average geomagnetic index.

2.1.6 Drag Model

Drag is computed as a function of the cross-sectional area of the vehicle relative to the wind, the density of the atmosphere at the altitude where the vehicle is located, the drag coefficient, the mass of the vehicle, and the velocity of the vehicle relative to the wind.

The vehicle attitude relative to the wind plays a major role in the computation of drag. The drag can be computed using changing attitude information or using a constant average attitude.

Since the mass of the vehicle normally changes throughout the flight, this information is also incorporated into the drag equation.

2.1.7 Vent Model

Although perturbations due to the attitude control system and known vents such as water dumps are not modeled in the software, the capability does exist to input values in vehicle body coordinates to compensate for these accelerations. The vehicle attitude, vehicle mass changes, vent forces, and on/off times are used.

2.2 ROUTINES THAT SUPPLY INFORMATION TO THE ENCKE EFFECTIVE FORCES ROUTINE

2.2.1 ENCKE Effective Forces Routine (ECENEF)

ECENEF is a subroutine located within the free-flight numerical integrator (ECENCKE) that is used to calculate the total accelerations upon a vehicle in the gravitational field of three attracting bodies (Earth, Moon, and Sun) and is also subject to other perturbing forces such as venting, drag, and geopotential.

ECENEF takes information acquired from other routines and uses it to calculate the total accelerations upon the vehicle (fig. 2-2). This information is then used by the Encke numerical integration routine within the free-flight numerical integrator. A detailed description of ECENEF is in section 2.4.2.2.

2.2.2 Body Attitude Determination Routine (ECENBY)

ECENBY accesses the body attitude table (BTAB) and eigenaxis table (ETAB) to compute a rotation matrix to the Mean of 1950 (M50) coordinate system. Each entry in the BTAB contains a time, mode indicator, 3×3 body matrix, and a corresponding entry in the ETAB, if any. Each entry in the ETAB contains a three-element eigenvector and a rotational rate value. Only body attitude types are listed in the BTAB. The types listed are as follows:

- a. Inertial attitude hold (IH)
- b. Inertial barbecue (BBQ), also known as rotor (ROTR)
- c. Local vertical/local horizontal (LVLH)
- d. Solar inertial (SI)

Only the BBQ mode will have a corresponding ETAB entry to describe its rotation. For each call to ECENBY, a rotation matrix will be computed and given to ECENEF.

2.2.3 Vehicle Weight and Configuration Routine (ECENWT)

The purpose of this subroutine is to look up the vehicle weight for low-level thrust and drag acceleration computations and the vehicle configuration flag used to specify constants for attitude-dependent drag forces. ECENWT searches the weight table (WTAB) until a weight entry corresponding to the current time is located. If current time is prior to the first WTAB entry, then a

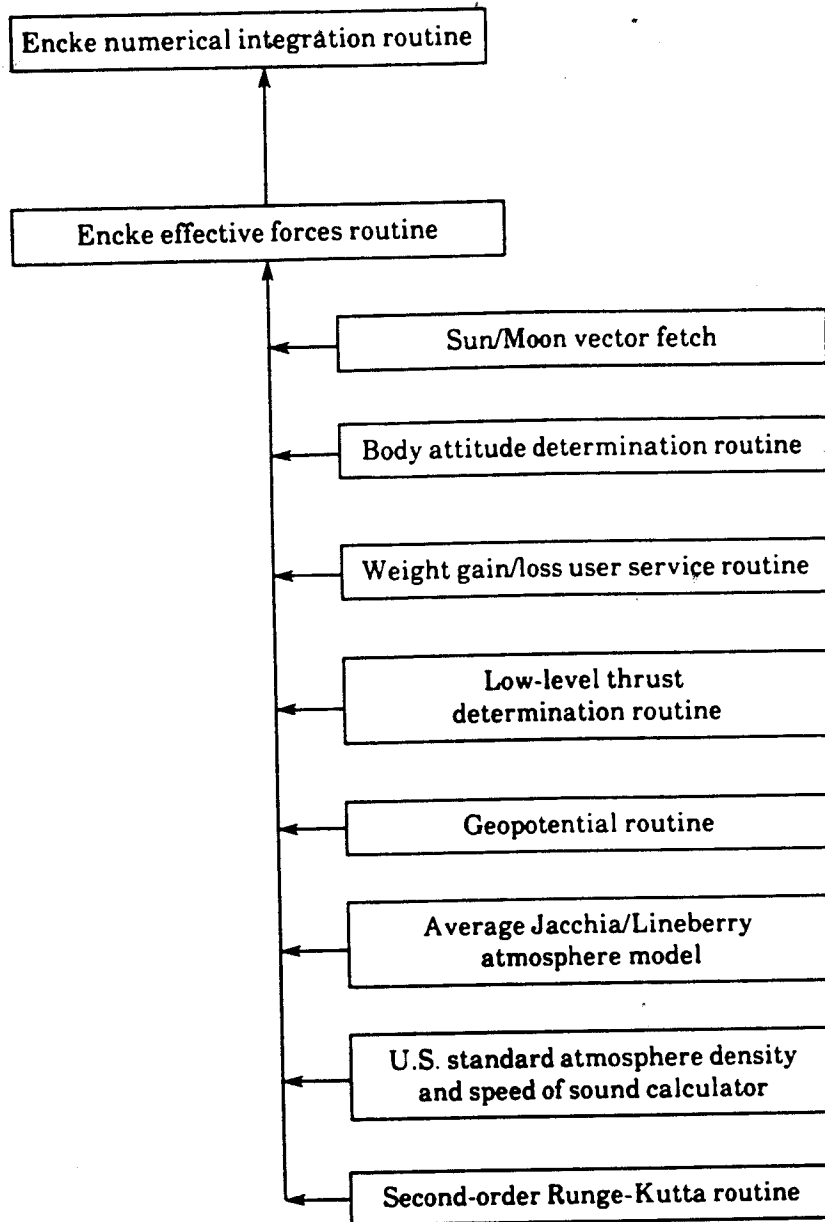


Figure 2-2.- ENCKE effective forces routine interface.

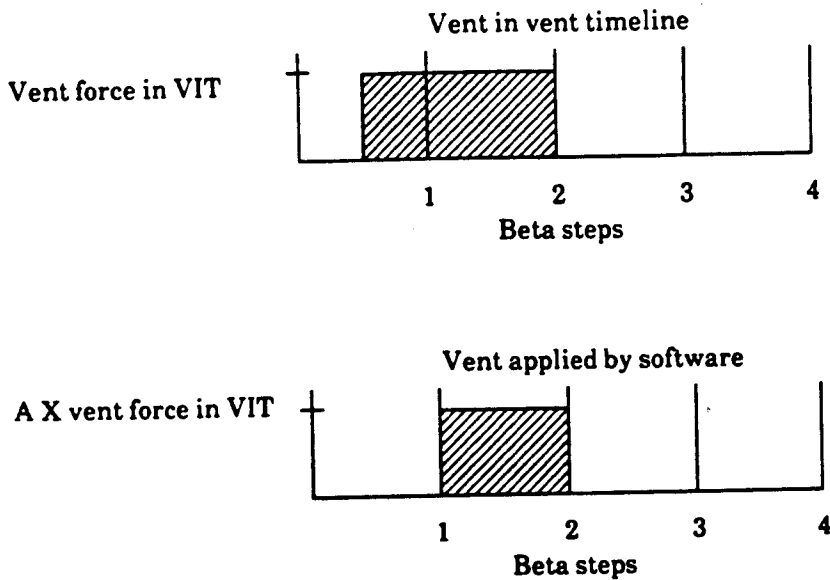
zero weight is returned. If current time is beyond the last value in the WTAB, then the weight and cargo bay door flag for the last entry is returned.

2.2.4 Low-Level Thrust Determination Routine (ECENVT)

ECENVT is called by the effective forces routine and is used to sum all the vent forces for each integration step. These forces are converted into a vent force vector in the M50 coordinate frame.

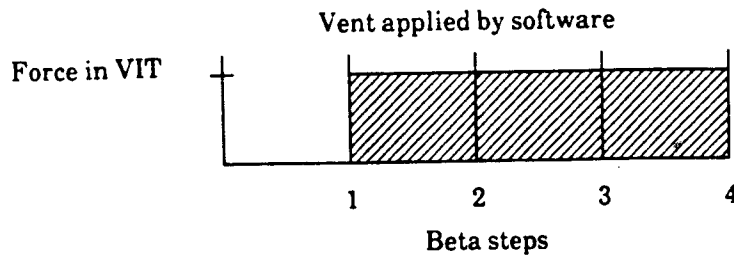
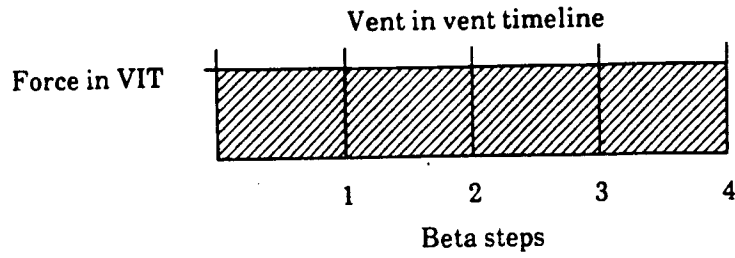
Because the integration steps are asynchronous with the vent force data, the ECENVT determines how vent forces, contained within the integration step interval, are applied across the integration step. Vent forces in the vent table (VTAB) that fall within the integration step interval are applied according to the following logic.

Case 1



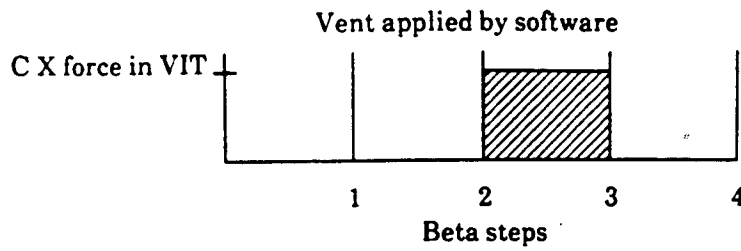
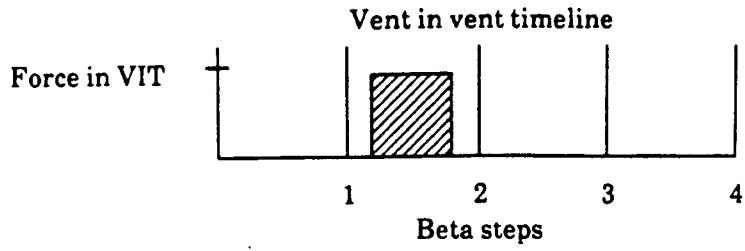
$$A = \frac{(\text{Time of } \beta\text{-step 2}) - (\text{Vent start time})}{(\text{Time of } \beta\text{-step 2}) - (\text{Time of } \beta\text{-step 1})}$$

Case 2



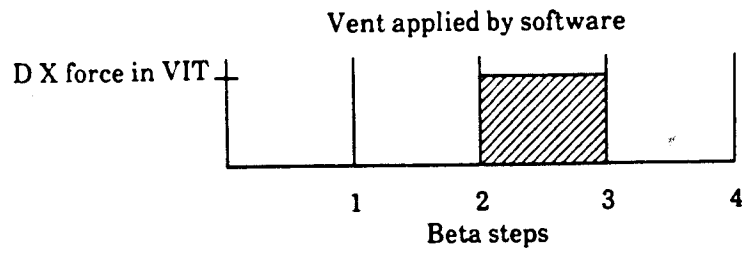
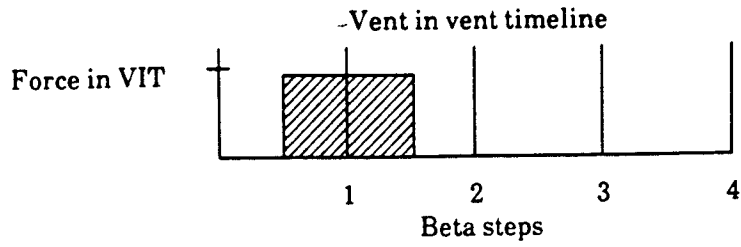
If a vent is on for the entire previous beta step and is on for subsequent beta steps, the force as entered in the vent initialization table (VIT) will be applied for all beta steps that fall within the vent interval.

Case 3



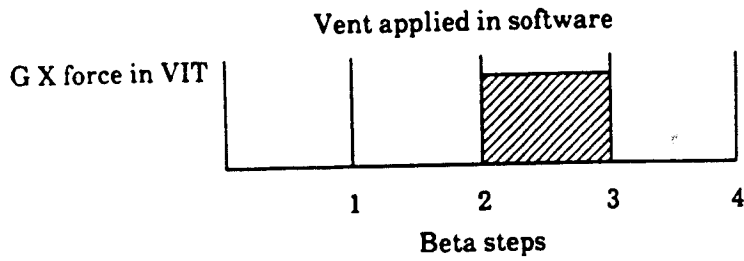
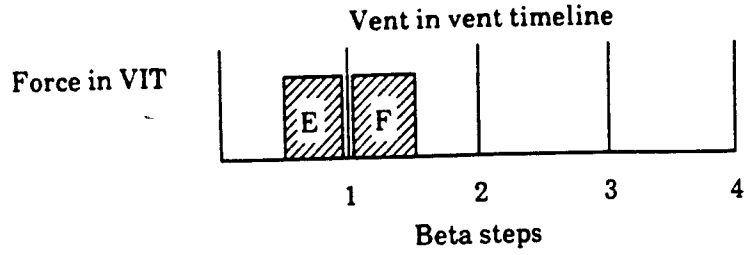
$$C = \frac{(\text{Vent stop time}) - (\text{Vent start time})}{(\text{Time of } \beta - \text{step 2}) - (\text{Time of } \beta - \text{step 1})}$$

Case 4



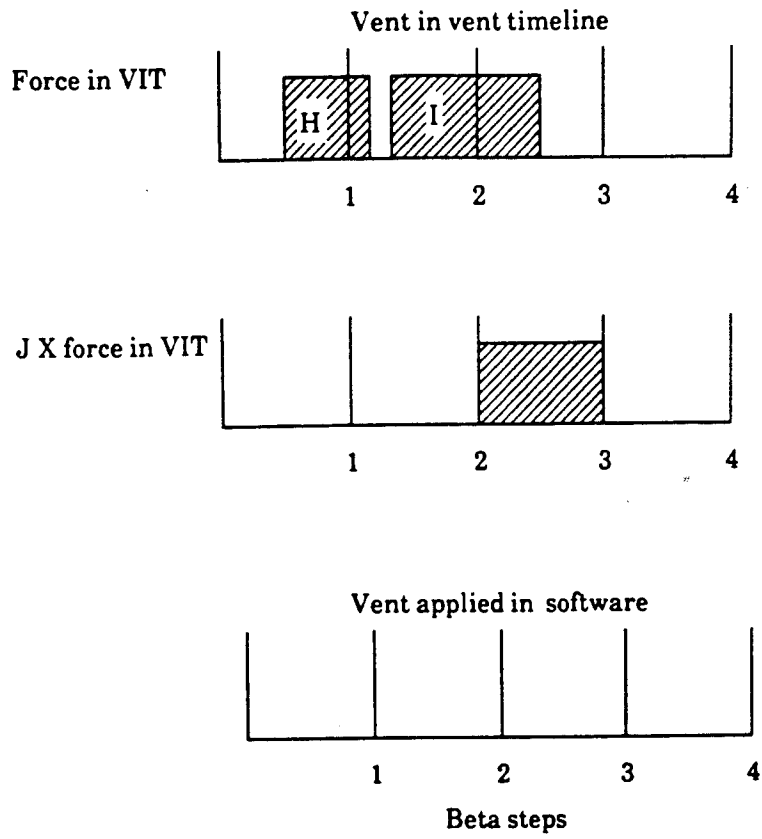
$$D = \frac{(\text{Vent stop time}) - (\text{Time of } \beta - \text{step 1})}{(\text{Time of } \beta - \text{step 2}) - (\text{Time of } \beta - \text{step 1})}$$

Case 5



$$G = \frac{(\text{Vent F start time}) - (\text{Vent F stop time})}{(\text{Time } \beta - \text{step 2}) - (\text{Time of } \beta - \text{step 1})}$$

Case 6



$$J = \frac{(\text{Time of } \beta - \text{step } 2) - (\text{Vent I start time})}{(\text{Time of } \beta - \text{step } 2) - (\text{Time of } \beta - \text{step } 1)}$$

2.2.5 Fourth-Order Runge-Kutta Routine (ECENER)

This routine uses the Nystrom version of the fourth-order Runge-Kutta formulas to determine the new position and velocity vectors at $t = t_o + h$, given the initial state vector (t_o, R_o, V_o) and the integration step size (h) .

To determine the new position and velocity vectors at $t = t_o + h$, given the initial state vector (t_o, R_o, V_o) and the integration step size (h) , this routine uses the fourth-order Runge-Kutta formulas to take four evaluations of the second derivative.

These four derivatives are calculated as shown below:

$$\bar{f}_1 = f\left(t_m, \bar{y}_m, \bar{y}'_m\right) \quad (1)$$

$$\bar{f}_2 = f\left(t_m + \frac{h}{2}, \bar{y}_m + \frac{h\bar{y}'_m}{2} + \frac{h^2\bar{f}_1}{8}, \bar{y}'_m + \frac{h\bar{f}_1}{2}\right) \quad (2)$$

$$\bar{f}_3 = f\left(t_m + \frac{h}{2}, \bar{y}_m + \frac{h\bar{y}'_m}{2} + \frac{h^2\bar{f}_1}{8}, \bar{y}'_m + \frac{h\bar{f}_2}{2}\right) \quad (3)$$

$$\bar{f}_4 = f\left(t_m + h, \bar{y}_m + h\bar{y}'_m + \frac{h^2\bar{f}_3}{2}, \bar{y}'_m + h\bar{f}_3\right) \quad (4)$$

where

t_m = value of the independent variable of integration at the beginning of the step

\bar{y}_m = current position vector

\bar{y}'_m = current velocity vector

h = integration step size

From these four samples of the acceleration, the new position and velocity vectors at $t = t_m + h$ may be evaluated by:

$$\bar{y}_{m+1} = \bar{y}_m + h\bar{y}'_m + \frac{h^2(f_1 + f_2 + f_3)}{6} \quad (5)$$

$$\bar{y}'_{m+1} = \bar{y}'_m + \frac{h(\bar{f}_1 + 2\bar{f}_2 + 2\bar{f}_3 + \bar{f}_4)}{6} \quad (6)$$

This routine participates in the averaging of vent accelerations by summing the saved vent acceleration vector from ECENEF into the averaged vent acceleration vector (all in common area VENTC).

2.2.6 Atmosphere Modeling Routines

There are two atmospheric density models used in the MCC ground navigation program. The computations are based on the height at the requested time. At altitudes from 91.4 km (300,000 ft) to 125 km (67 n. mi.), the 1962 U.S. standard atmosphere model is used; from 125 km to 700 km, the average Jacchia/Lineberry atmosphere is used. Above 700 km, no atmosphere is assumed and drag is set to zero. These are used by the free-flight numerical integrator to model the atmospheric density when integrating a state vector over a data arc interval.

The two atmosphere models mentioned above are actually two separate processors in the mission operations computer (MOC) and are described below.

2.2.6.1 Average Jacchia/Lineberry Atmosphere Model

The average Jacchia is a dynamic model accounting for periodic variations due to diurnal, annual, and seasonal effects and for periodic effects due to solar flux and geomagnetic index. The Lineberry approximation divides the atmosphere into altitude bands and expresses the log of the density as a series in temperature.

The Jacchia/Lineberry atmosphere model is divided into four routines: the Jacchia executive routine, Jacchia initialization and solar activity routines, and altitude band routine. Each of these is described below.

The Jacchia executive routine is the overall driver and contains most of the mathematical processing. It calls the Jacchia initialization routine to compute slowly varying parameters and the altitude band routine to calculate the altitude band. The Jacchia executive routine also adds the effect of hydrogen to the atmospheric density for altitudes above 500 km (270 n. mi.) to 700 km.

The Jacchia initialization routine computes slowly varying parameters such as the Sun's right ascension and declination, functions of the solar flux and geomagnetic index, and functions of the time of year. This routine calls the solar activity routine to provide values for the solar flux and geomagnetic index.

The solar activity routine interpolates the current values of the solar flux and geomagnetic index from the Jacchia constants table.

The altitude band routine determines the current altitude band of the vehicle, since the Jacchia/Lineberry formulation uses curve fits to the tables of Jacchia over different altitude bands.

a. Inputs

- Vehicle data array, which contains time, altitude, and either a Cartesian position vector or right ascension, declination in true-of-epoch inertial (TEI) coordinates
- Sun position vector
- Flag to indicate input of either a Cartesian position vector or right ascension, declination

b. Outputs

Atmospheric density in kilograms per meters cubed

Both solar flux and geomagnetic activity have a delayed effect on atmospheric density. The Jacchia/Lineberry atmosphere model in the MCC ground navigation program requires these values.

The values requiring manual input are as follows:

FB10.7 - 90-day running average solar flux

Kpbar - 12-month running average geomagnetic index

Solar flux values are delayed by 1 day. For example, the 3-month solar flux value for December 21, 1988, would be used as the input for December 22, 1988.

Predicted and historical solar flux and geomagnetic values from 1983 through 1992 are stored at 3-month intervals in the MCC ground navigation software. The predicted and historical values are used for simulations. For actual flights, this information is replaced by current values as part of the launch minus 24 hour inputs via the Jacchia table maintenance processor.

The MOC Lineberry approximation to the Jacchia atmosphere model does not account for the daily solar activity. Drag accelerations computed by the MOC software can be multiplied by a scalar K-factor to theoretically compensate for these mismodelings. The drag K-factor is the ratio of the

average density computed with the full Jacchia atmosphere model, using the percent change in daily solar flux (F10.7) and the 90-day running average solar flux (FB10.7), to the average density computed using the Jacchia/Lineberry approximation using only FB10.7.

The drag K-factor tables can be used for a variety of functions to compensate for atmospheric mismodelings due to the Lineberry approximation and variations in daily solar flux. These tables are available on console and can be used as input to account for higher or lower than average daily solar activity. This value is input via manual entry device (MED) and is controlled by the flight dynamics officer (FDO) and entered by Dynamics.

2.2.6.2 U.S. Standard 1962 Atmospheric Density and Speed of Sound Calculator (EM1962)

EM1962 uses the U.S. standard atmosphere model to perform its calculations. It computes the density of the atmosphere in kilograms per cubic meter and the speed of sound in Earth radii per hour for any given altitude from sea level to 700 km. For altitudes above 700 km it returns a density value of zero. The atmosphere is divided into sections, with the first section being from sea level to 90 km. The sections are further divided by altitude into layers, and the density is computed as a function of density and temperature at the base of the altitude layer, temperature and gradient within the layer, and precomputed interpolation coefficient. The speed of sound is computed as a function of temperature at the specific altitude. The ground navigation software requires only density calculation from this processor at altitudes from 300,000 feet to 125 km (67 n. mi.).

a. Inputs

- Altitude above the oblate Earth
- System parameter MCC MCU (meters/Earth radii)

b. Outputs

Density of atmosphere for input altitude/speed of sound

2.2.7 Gravitational Perturbation Routines

2.2.7.1 Geopotential Routine (ECENMP)

Since the Earth is not an idealized homogeneous sphere, the geopotential must be employed to account for the gravitational effects of the aspherical Earth on an orbiting vehicle. A geopotential model adds mass concentrations to a spherical Earth to generate an accurate gravitational representation of the Earth. The accuracy of the model rests on the number, size, and location of the mass concentrations added. The concentrations are added in three different fashions. The most significant concentrations are the zonal harmonics, J-terms, which add mass rings encircling the

Earth at a specified latitude. The most prominent of these is the J2 term, which models the Earth's equatorial bulge. The second type of concentrations is sectorial harmonics. These terms add bulges in sections along a longitudinal band completely encircling the Earth. The primary sectorial term accounts for ellipticity of the Earth's equator. Tesseral harmonics are the final concentrations added. Mass concentrations are added locally in curved rectilinear surfaces bounded by lines of constant latitude and longitude. The effect of the tesseral harmonics is not as great as those of the zonal and sectorial harmonics. The gravitational potential of the Earth is then expressed as a truncated series utilizing the coefficients. The model currently used in the geopotential routine (ECENMP) is the Goddard Earth Model version 10 (GEM 10) 7×7 model. This information is used by the Encke effective forces routine to numerically integrate a state vector over a data arc.

.2.2.7.2 Sun/Moon Vector Fetch (ELEFEM)

The Sun/Moon ephemeris table consists of 102 blocks of data. Each block contains a timetag, in hours; one Sun vector; and eight Moon vectors. The block timetags are separated by 96 hours. The timetag associated with each block is in hours from the base time (January zero of the MOC initialization year). The date indicated on the trajectory initialization MED (P80) dictates the initial timetag. In all cases, the first data block is 12 to 16 days before the MED date (this allows for interpolation at the beginning of the ephemeris), and the last timetag is 396 to 400 days from the MED date. With the standard ephemeris build for a flight load, this procedure allows operations into the rollover year for approximately 20 to 30 days. If operations longer than this should be required, it would become necessary to build the Sun/Moon ephemeris table with a later P80 MED start date.

ELEFEM uses Greenwich mean time (GMT) to search the Sun/Moon ephemeris table for Sun and Moon vectors with that timetag. If no vectors exist for the specified time, an interpolation is performed to generate a Moon vector and a barycenter vector. The barycenter vector is converted into a Sun vector. The results are then returned to the user.

1. Inputs

- Specified GMT in hours
- User option
- Sun/Moon ephemeris

2. Outputs

- Requested Sun/Moon vectors
- Error indicators

2.3 PROCESSORS SUPPLYING INFORMATION TO THE NUMERICAL INTEGRATION PREPROCESSOR (EMPREPRO)

The numerical integration preprocessor (EMPREPRO) supervises the generation of specific input data for the free-flight or powered-flight numerical integrator. The integrator options, mission plan table (MPT) and caller's input list, are interrogated. Depending upon their settings, EMPREPRO retrieves data from any or all of the following tables:

- a. Attitude timeline (ATL) table
- b. Reentry attitude timeline (RATL) table
- c. Vent timeline (VTL) table
- d. Reentry vent timeline (RVTL) table
- e. Cargo bay door status (CBDS) table
- f. Weight gain/loss (WGL) table

In order to retrieve this information, EMPREPRO interfaces with the following processors (fig. 2-3):

- a. ATL user service
- b. VTL user service
- c. Weight and vehicle configuration user service

2.3.1 Attitude Timeline User Service

The attitude timeline user service (ATUS) computes and provides orbiter orientation data based on the requested time or time interval.

Given the EMPREPRO parameter list, the ATUS constructs the BTAB and ETAB tables, which are used to compute a rotation matrix from body coordinates to the M50 coordinate system. Each entry in the BTAB contains a time, mode indicator, 3×3 body matrix, and a pointer to a corresponding entry in the ETAB, if any. The number of entries in BTAB are stored in BNUM. Each entry in the ETAB contains a three-element eigenvector and a rotational rate value. There are only four body attitudes allowed in the BTAB. The types are as follows:

- a. IH
- b. Inertial BBQ or ROTR

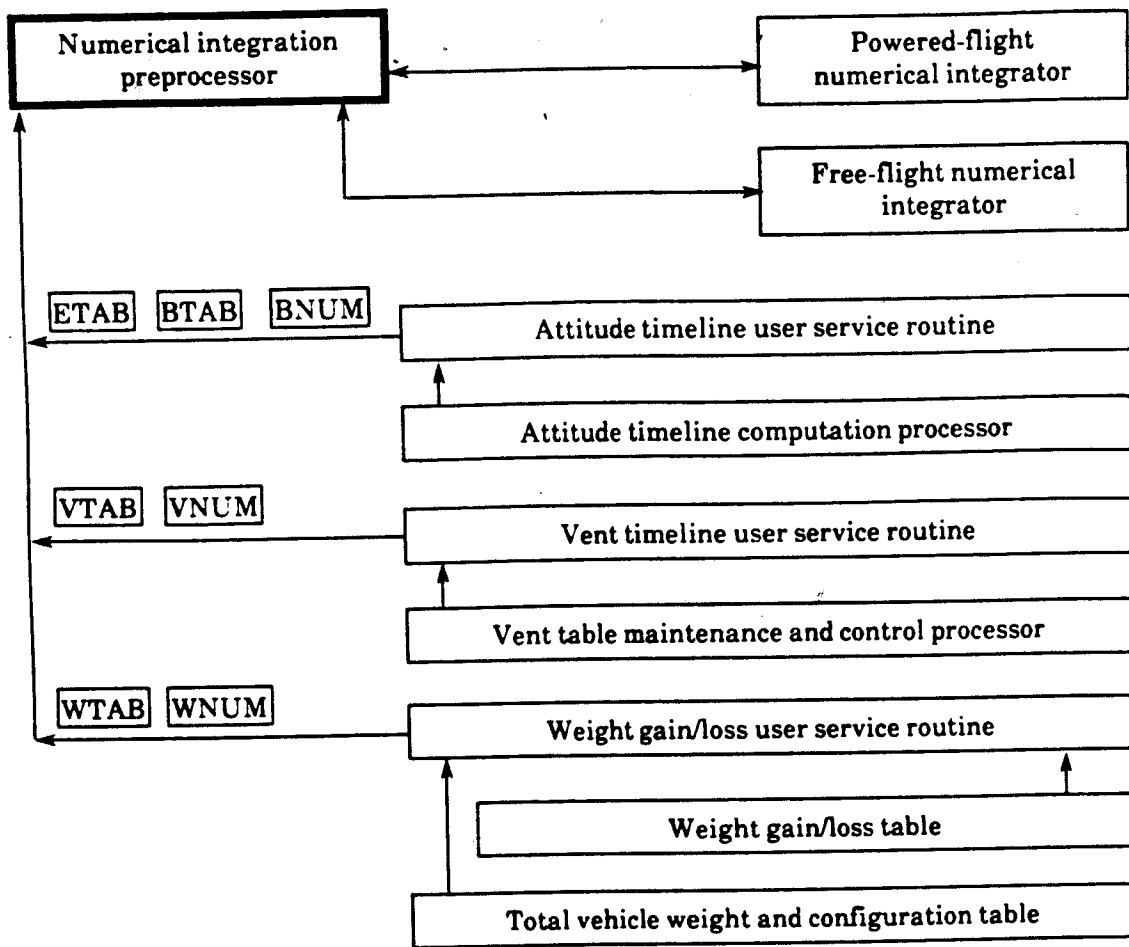


Figure 2-3.- Numerical integration processor interface.

c. LVLH

d. SI

Only the ROTR mode will have a corresponding ETAB entry to describe its rotation.

Given the numerical integration preprocessor parameter list, the ATUS constructs the BTAB and ETAB output tables using appropriate information from the requested flight ATL or RATL.

Given a time, position, velocity vector, and a working area, the entry of the ATL at the time of interest is stored in the working area with an updated version of its body or rotation matrix and the time of the next entry in the ATL.

a. Inputs

- MED parameters
- Entry updates
- Reference stable member matrix (REFSMMAT)/relative matrix (RELMAT) identification (ID) changes
- Time changes
- Data tables
- ATL
- RATL
- REFSMMAT/RELMAT locker

b. Outputs

- ATL
- RATL
- To EMPREPRO: time, attitude mode, computed matrix, eigenaxis, rotation rate.

2.3.1.1 Attitude Timeline/Computation Processor

The attitude timeline/computation processor (ATCP) (see fig. 2-4) supplies orbiter orientation data based on predefined orbiter ATLs within the ATL and the RATL. These include the ability to

a. Maintain the ATL and the RATL which is done by

1. Adding, modifying, or deleting single entries
2. Modifying times or REFSMMAT/RELMAT IDs for multiple entries
3. Computing a rotation or body matrix for each entry

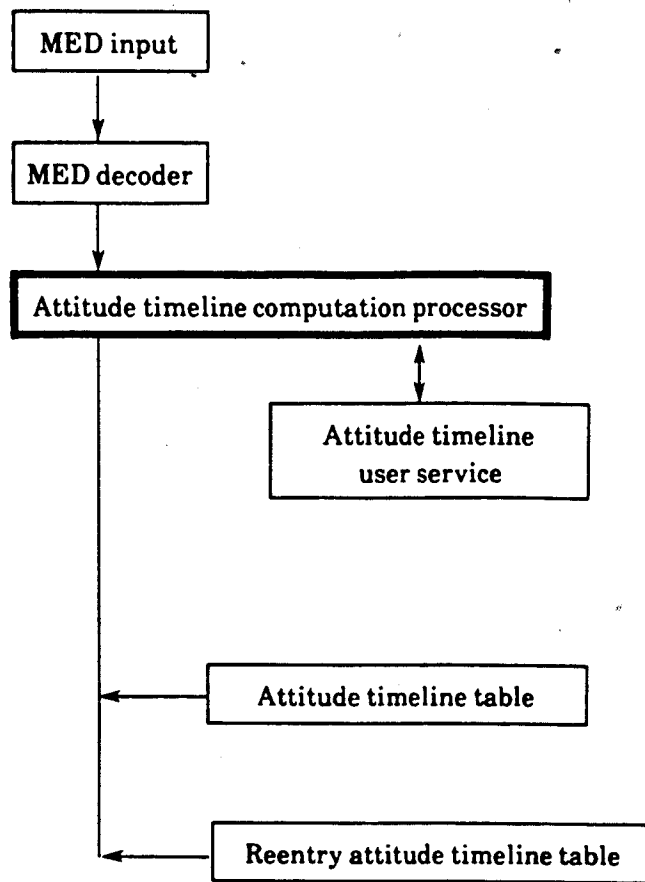


Figure 2-4.- Attitude timeline/computation processor data flow.

- b. Compute and provide orbiter orientation data based on the ATL tables at a user-requested time or from a time interval

The ATL table is initialized by premission processing. The ATPC performs maintenance functions required to provide the best estimate of the orbiter orientation. The ATL and the RATL contain a header and 50 entries for the RATL, 500 for the ATL. Each entry consists of

- Entry number
- Attitude mode
- Time of attitude change
- REFSMMAT/RELMAT ID
- Computed body matrix or rotation matrix
- Eigenaxis
- Rotation rate
- Attitudes
- Entry purpose

Both the ATL and the RATL will be initialized premission. The ATL is initialized via a D64 element created by the attitude and pointing flight controllers.

The capability exists to add, modify, or delete entries in the ATL or RATL. These functions are executed by the attitude and pointing flight controller via MED.

If EMPREPRO requests orbiter orientation data during a specific time interval (type 1 call), the following information is required as input to the ATCP:

- a. GMT to begin orientation data
- b. GMT to end orientation data
- c. Deorbit time of ignition (DTIG)
- d. GMT of major mode 3 (entry)
- e. Body matrix of major mode 3 - nine-element body matrix
- f. GMT of major mode 4 (on-orbit)
- g. Attitude of major mode 4 - pitch, yaw, roll LVLH attitude

Upon a type 1 call, the DTIG is checked. If the time of ignition (TIG) is nonzero, two entries are built from the major mode input data. The only computation required for major mode 4 is a rotation matrix using the LVLH attitude and an identity matrix. Entries are gathered from the ATL and the RATL. In the case of DTIG being set, major mode 303 (post-deorbit coast) and major mode 304 (early phase of entry) entries are sorted in, where appropriate. While the data is gathered, two tables are constructed.

The first table, named BTAB, will contain

- a. GMT of attitude change
- b. Attitude mode
- c. Body matrix or rotation matrix

The second table, named ETAB, will contain

- a. Unit eigenvector
- b. Angular rate of rotation

If EMPREPRO requests orbiter orientation data at a specific time (type 2 call), the processor searches the ATL to find an entry whose time is less than the time of interest. Once the proper entry time is found, the attitude mode is checked. If the time attitude mode is IH, the entire entry contents are returned. For the rest of the attitude modes the current body matrix is computed. The entry contents, with the current body matrix substituted, will be returned to the calling software via the ATUS routine.

a. Inputs

- MED parameters
- Entry updates
- REFSMMAT/RELMAT ID changes
- Time changes
- Data tables
- ATL
- RATL
- REFSMMAT/RELMAT locker

b. Outputs

- ATL
- RATL
- To ATUS: time, attitude mode, computed matrix, eigenaxis, rotation rate.

2.3.2 Vent Timeline User Service

The vent timeline user service (VTUS) receives request via EMPREPRO or flight controller MED.

Flight controller MED requests (see fig. 2-5) consist of adding, deleting, and modifying entries in the VTL or RVTL. This type of information is passed to the vector table maintenance and control (VTMC).

Two types of EMPREPRO requests exist, a request for VTL information and a request for RVTL information. Based on the type of call, VTUS pulls information from the appropriate table and places it in a temporary work area containing low-level thrust data (VTAB), the number of entries in VTAB is stored in VNUM. The addresses of VTAB and VNUM are returned to EMPREPRO.

a. Inputs

- MED inputs
- VTL table
- RVTL table
- Deorbit parameters
- VIT

b. Outputs

Low-level thrust table (VTAB)

2.3.2.1 Vent Table Maintenance and Control Processor

The maintenance function consists of adding, deleting, and modifying entries in the following tables:

a. Orbit VTL table

A table that is used during the timeframe from beginning of postinsertion to approximately deorbit down to 300,000 ft. Up to 200 VTL entries are stored in the following format:

- Vent ID
- Entry number
- Vent start time
- Vent stop time

b. RVTL table

A table covering the time from approximately 6 hours prior to entry interface to the time of landing. Up to 50 RVTL entries are stored in the following format:

- Entry number
- Vent ID
- Vent start delta time
- Vent stop delta time

c. Vent initialization table

A table defining the X, Y, Z forces of each vent in orbiter body axis. Up to 50 VIT entries are stored in the following format:

- Vent ID
- FX - force (pounds)
- FY - force (pounds)
- FZ - force (pounds)

The VITs are initialized premission and can be updated during a mission via flight controller MED entry (figure 2-5).

Inputs

- MED inputs
- VTL table
- RVTL table
- Deorbit parameters
- VIT

Outputs

- Updated VTL table
- Updated RVTL table
- Updated VIT

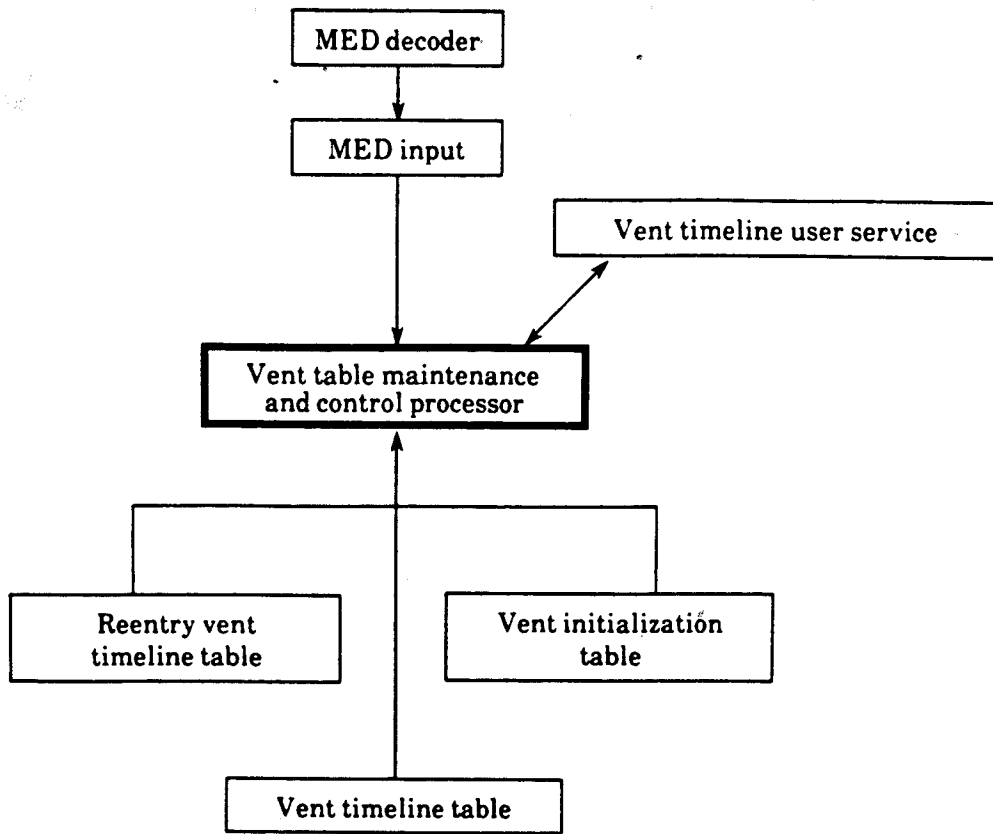


Figure 2-5.- Vent timeline maintenance and control processor data flow.

2.3.3 Weight Gain/Loss User Service Routine (ELWG TAB2)

ELWG TAB2 constructs a table of total vehicle weight and configuration WTAB for a given time or at any instant for a requested interval (fig. 2-6). ELWG TAB2 also stores the number of entries within WTAB in WNUM.

Delta weights and the vehicle configuration status are retrieved from the requested orbiter WGL table and cargo bay door status table, respectively. The initial vehicle weight is either passed in the input/output (I/O) parameter list or fetched from the indicated MPT. In addition, the K-factor and the vehicle area for constant or variable area drag may be returned.

The three options available for returning the vehicle weight for a given weight and configuration are as follows:

Option 1 - Return the total vehicle weight for a given time. Consider the selected WGL table.

Option 2 - Return the vehicle weight and configuration table for a requested interval. The initial vehicle weight may or may not be passed by the calling routine. Consider the selected WGL table and the CBDS table.

Option 3 - Return the total vehicle weight and configuration table for a requested interval. The initial vehicle weight may or may not be passed in the I/O list. Consider the CBDS table only.

If the initial vehicle weight is not passed, ELWG TAB2 searches the selected MPT for the last maneuver prior to request time. If the request time equals the maneuver initiate time or falls inside the burn time of a maneuver, the end weight of the previous maneuver is taken as the initial vehicle weight. If there are no maneuvers prior to the request time, the initial vehicle weight is fetched from the MPT header.

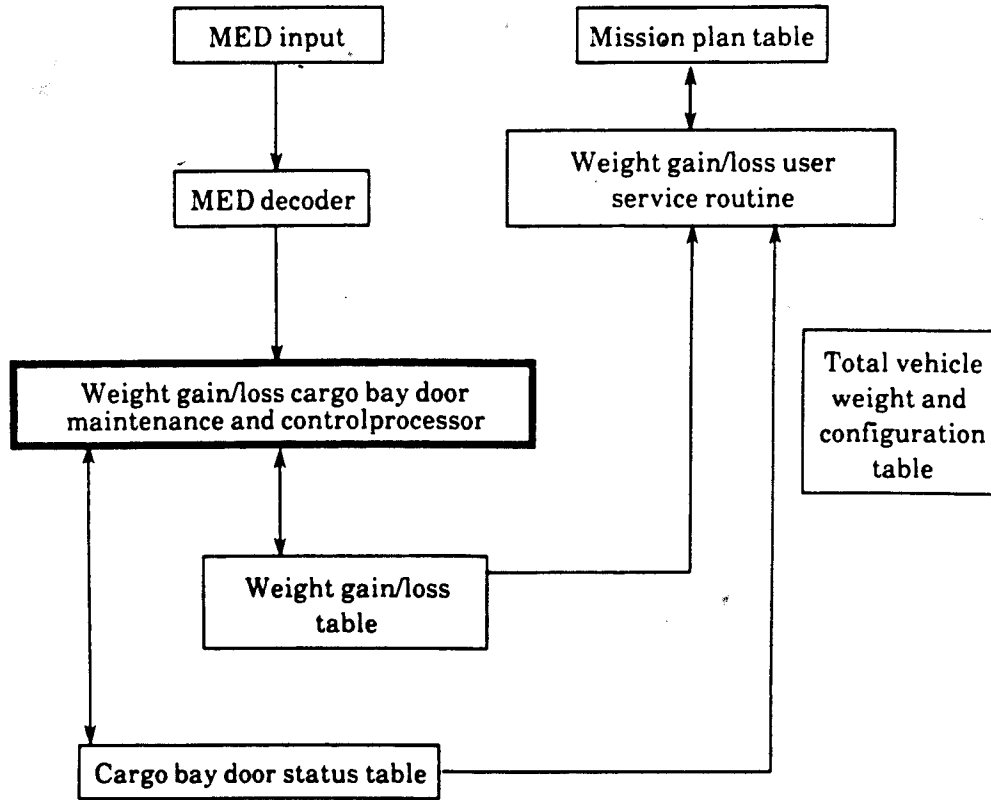


Figure 2-6.- Weight gain/loss and cargo bay door table maintenance and control processor data flow.

The time of the selected vehicle weight is located in the WGL table. All delta weights between the initial time and T-left are added to the initial vehicle weight. The status of the cargo bay door is retrieved from the CBDS table for T-left.

The area and K-factor are stored in the I/O list if they are requested.

For option 1, the vehicle weight is stored in the I/O parameter list and the WTAB is not constructed. For option 2, the WGL and CBDS tables are interrogated for changes occurring between the start and stop request times. All changes are stored in successive order in the WTAB. Processing continues until the end time of the request is reached. For option 3, a WTAB for a constant vehicle weight is constructed. The CBDS table is interrogated for changes in the bay door status. Any change generates an entry in the WTAB. Processing continues until the request end time is reached.

a. Inputs

- Total vehicle weight
- GMT of bay doors assumed closed
- Area and K-factor return indicator
- Start time of request
- Stop time of request

b. Outputs

- Total vehicle weight
- K-factor and area (if requested)
- Return code
- Number of WTAB entries
- Address of WTAB

2.3.3.1 Weight Gain/Loss and Cargo Bay Door Maintenance and Control Processor (EMWGTBL)

EMWGTBL maintains a history of the cargo bay door status and significant weight changes affected by weights lost or gained from deploying or retrieving payloads. The control function generates and stores weight configuration data (WTAB) for use by the free-flight numerical integrator.

A WGL table of 75 entries is associated with each of the orbiter ephemerides.

Table maintenance is initialized via MED (fig. 2-6) to add, modify, or delete weight entries.

Add option - When the add option is selected, EMWGTBL examines the table fill indicator and determines whether a new weight entry may be added to the table. If the table is filled the user is notified and the processing is terminated. Otherwise, the entry is positioned in the table in

chronological order by the GMT associated with the delta weight change. Existing weight entries are not overlaid but are shifted down to allow for the insertion of the new entry.

Delete/modify option - For delete or modify option, EMWGTBL searches the appropriate WGL table using the manually entered GMT (time of weight entry) as the keyword. If a match is not found, the user is notified and the processing is terminated. The delete option is honored if the entry is found, and all remaining entries are shifted up to prevent accessing errors.

Total weight option - The WGL user service routine (ELWG TAB) is called to compute the total vehicle weight for the requested MED time. The delta weight is calculated, and the functions as stated under the add option are then performed.

Cargo bay door maintenance - The cargo bay door table maintenance function is executed via MED to add an entry to or delete an entry from the table.

a. Inputs

- MED parameters
- WGL table
- CBDS table
- MPT

b. Outputs

- WGL table
- CBDS table

2.4 FREE-FLIGHT NUMERICAL INTEGRATOR (ECENCKE)

ECENCKE is a sixth-order prediction-correction numerical integrator used for free-flight trajectories. It is capable of using either the Encke scheme of integration, with time or beta as the independent variable of integration, or the Cowell mode of integration. However, the Cowell mode is not used in the ground-based space systems. ECENCKE is used for integration during a trajectory update and for any free-flight integration that may be required.

Integration of the second-order equations of motion is done by an Adams-Moulton-Bashforth sixth-order prediction and correction with finite differences. Starting values are obtained by Nystrom's version of the fourth-order Runge-Kutta equations.

Encke's method is used to integrate noncentral body forces and solve analytically for the motion caused by the spherical primary body. In the Encke mode the independent variable of integration may be time or a single function of the eccentric anomaly referred to as beta. The second derivative includes gravitational accelerations, atmospheric drag, and venting.

The integrator will use

- Gravitational acceleration models assuming point mass distributions of the Sun and Moon and a nonspherical mass distribution of the Earth
- Vent (low-level thrust) model
- Attitude-dependent and independent atmospheric drag models employing the average Jacchia/Lineberry atmosphere model and the 1962 standard atmosphere model
- The central body defined as the Earth at which the center of the inertial reference system is located

The methods available for performing the stepwise numerical integration of the three second-order differential equations of motion are a sixth-order backward difference multistep predictor-corrector procedure and a one-step Runge-Kutta method, as modified by Nystrom, equivalent to a fourth-order Taylor series expansion. The predictor-corrector method uses the Runge-Kutta method as a starting procedure in all cases to build the needed difference tables. The first seven integration steps of any restart caused by rectification, step size change, and/or reference body switch are taken and, using the Runge-Kutta method, are calculated as one-fourth the predictor-corrector method's step size. The step sizes for the predictor-corrector are obtained as a function of the radial distance of the spacecraft from the central body or, as they are commonly called, concentric rings, and their values have been chosen conservatively to enable the predictor-corrector formulas to be applied only once per step.

The β mode of the Encke integration scheme was introduced for a variety of reasons. The use of a consistent set of two-body motion modules by all phases of the mission planning program has been convenient. Reasons more specific to the integration procedure require examination of some of the definitions, and the first such examination reveals that the behavior of β enables us to use a more nearly uniform integration step size for the entire Earth-Moon trajectory without sacrificing accuracy. This results in a time savings by eliminating the need for frequent rebuilding of the difference tables associated with the predictor-corrector procedure. A more important factor is the elimination of the need for solving Kepler's equation by an iterative procedure. Since the two-body position and velocity vectors are the foundation of the Encke scheme, any reduction in the complexity of their solution is normally worthwhile. The β -step for the shuttle is approximately equal to 0.9 minutes. The β -step for TDRS is approximately equal to 5.5 minutes.

An ephemeris may be generated for any integration request. Ephemeris output control is provided by allowing input of an approximate delta time between ephemeris vectors and a maximum number of vectors to be stored.

a. Inputs

- Integration mode indicator
- Force model indicators
- Termination and time limit controls
- Step size and storage controls
- Vehicle state vector
- Weight and door status table - WTAB
- Body attitude tables - BTAB and ETAB
- Vent force table - VTAB
- System and trajectory parameters

b. Outputs

- Vehicle state vector
- Vehicle weight
- Ephemeris
- Termination indicator
- Error code

2.4.1 System and Trajectory Parameters Used in the Process

ECENCKE uses system parameters loaded into the MCC trajectory software via the flight software build/reconfiguration process.

ECENCKE also uses parameters computed by other routines. The names of the other parameters are known by the calling processor. Each parameter contains the address of the information needed by the calling processor. The parameters specified as system parameters are available for all applications within the MCC software.

2.4.1.1 System Parameters Used by ECENCKE

Name	Units	Description
MCGECC		Eccentricity limit
MCGSMA		Semimajor axis limit
MDCMAT		C-matrix used in Earth potential
MDSMAT		S-matrix used in Earth potential
MGZONE		Number of zonal terms in Earth potential
MCZFES	hr	Upper integration limit for fail safe
MGSECT		Number of sectoral terms in Earth potential
MDJCOE		J harmonics used in Earth potential
MGTESS		Number of tesseral harmonics in Earth potential
MCMTPI		2 n
MCZMIS	hr	Maximum integration limit for a low ecc. orbit
MCAKOA	Er	Kickout altitude for the Earth
MGREPH		Indicator for GO/NO-GO integration
MCGEER	Er	Radius of Earth for use in recursive potential model
MCEMUU	Er^3/hr^2	Mu of the Earth
MCGMUM	Er^3/hr^2	Mu of the Earth
MCERTS	radians/hr	Rotational rate of the Earth
MCEABS		(Earth's polar radius/Earth's equatorial radius)
MCCRAM	$\frac{M^3 * lb}{kg * hr^2 * Er}$	Conversion factor used in drag computation

System parameters

Name	Units	Description
MCADRG		Constant drag coefficient
MDVDRG		Variable drag coefficients
MCMHPI		$1/2 * \text{PI}$
MCMPPIR		PI
MCMTPI		$2 * \text{PI}$
MCEAGG	Er/hr ²	Acceleration of Earth gravity
MC300K	Er	300k ft
MCCFCU		ft/Er conversion
MDTALT	Er	Threshold altitude for low-altitude drag model
MDADRG		Low-altitude drag model constants
MDDRGB		Low-altitude drag model constants
MDCDRG		Low-altitude drag model constants
MDDRG		Low-altitude drag model constants
MDPDRG		Low-altitude drag model constants

2.4.1.2 Other Parameters Used by ECENCKE

Input parameters

Name	Description
DIN	(All items are double precision) GMT of the input vector (hr) Input position vector (E_r) Input velocity vector (E_r/hr) Minimum Δt of integration (hr) Maximum Δt of integration (hr) Minimum ephemeris storage interval Step-size control multiplier Desired ending value of stopping parameter at the Earth K-factor Drag indicator Venting indicator Maximum time mode step size (hr)

Other output parameters

Name	Description
DBLOUT	(Double precision) GMT of output vector Position vector returned by Encke Velocity vector returned by Encke Acceleration vector returned by Encke Weight at output time
EARTH	These arrays in which Encke will store the respective ephemerides as requested by the user who is responsible for allowing enough space in his program so that Encke will not overflow it in storing the maximum number of vectors IIN(7). The ephemeris stored by Encke will be in the format $[T_1, R_1, V_1, T_2, R_2, V_2 \dots]$. If no ephemeris is desired by the user, Earth may not be supplied
IOUT	(Fullword fixed point) Reference indicator Termination indicator Number of points in Earth ephemeris Error code
PDTAB	Address for output drag partial derivative table.

2.4.2 Supporting Routines

The following is a description of the routines that supply information to ECENCKE. Refer to figure 2-7 when reading this section.

2.4.2.1 Integration Termination Control (ECENEE)

ECENEE edits the results of the previous integration step and sets up the appropriate course of action. These are termination control, integration step size selection.

a. Termination control

ECENEE has the ability to terminate the integration process after a maximum delta time of integration or on any of the following cutoff conditions:

- (1) A specified radial distance from the center of the Earth.
- (2) A specified altitude over the oblate Earth.
- (3) A specified flightpath angle with respect to the Earth.
- (4) An ascending node with respect to the Earth.
- (5) A specified body-fixed longitude with respect to the Earth.
- (6) A specified body-fixed geocentric latitude with respect to the Earth.
- (7) Termination indicator (system parameter MGREPH) is set. This value is usually set to 10 days but can be changed in the launch minus 1 day inputs via the FO2 MED, if necessary.

An iteration procedure is required before any of the first six cutoff conditions can be satisfied. The particular parameter desired by the user is calculated around each integration step and saved with its associated vector. Once the desired solution has been bounded (i.e., the last integration step overshoot the solution), ECENEE goes back to the previously saved state vector, forces a rectification at that point, and calculates the approximate step size needed to reach the desired solution. This calculation is done by the cord or straightline approximation method. After this integration step has been taken, the selected parameter is recalculated and the bounding test is once again applied. This process is repeated until the residual (the difference between the actual parameter and the desired parameter) is less than 1.0×10^{-10} or if the calculated step size is less than 1.0×10^{-12} . During this iteration procedure no reference switching or ephemeris storage is allowed.

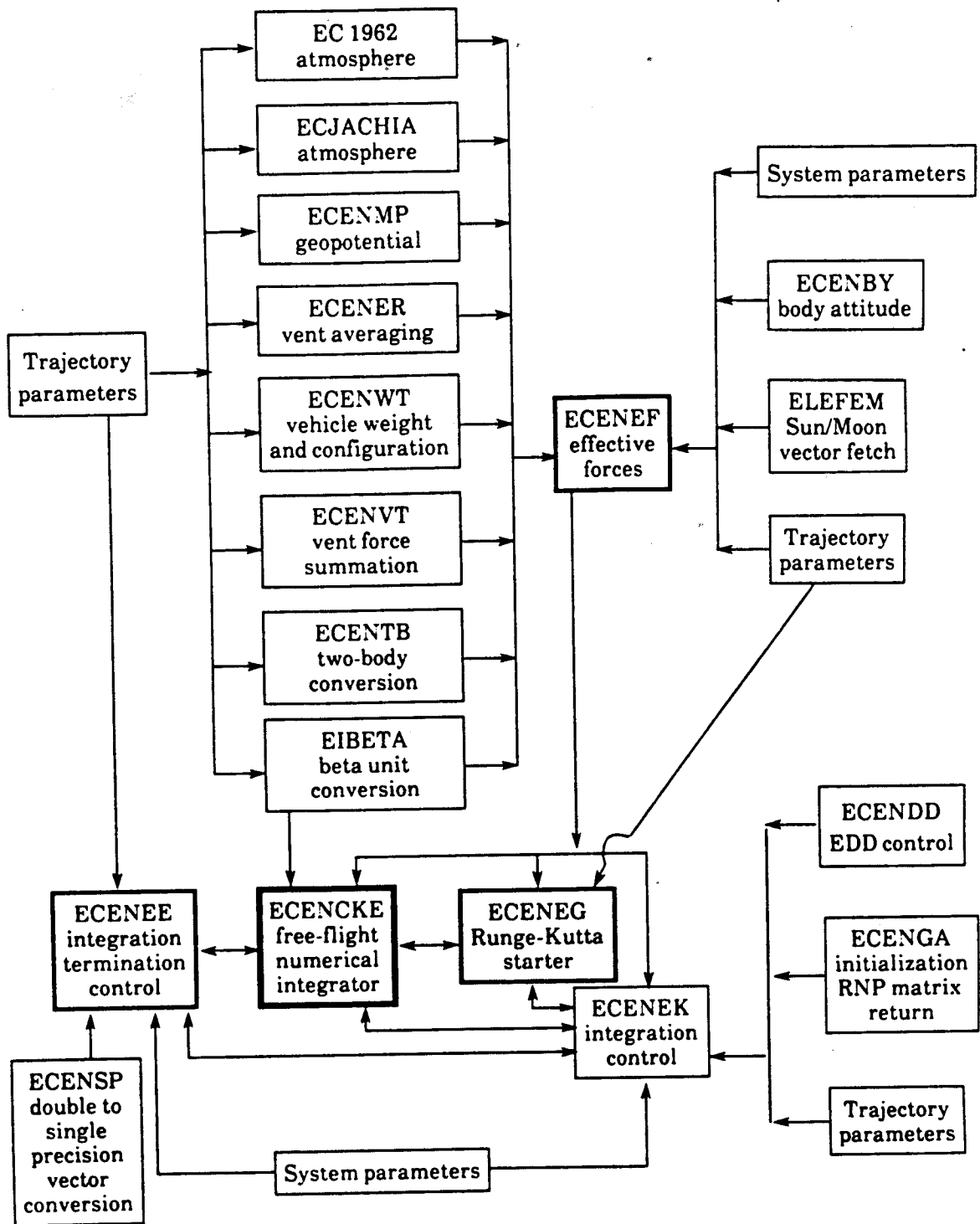


Figure 2-7.- Free-flight numerical integrator (ECENCKE) interface.

In the case of stopping on a maximum time of integration time, no iteration technique is required. Once the maximum time of integration has been overstepped, ECENEE switches into the Encke-time integration mode and steps back exactly to the requested integration stop time.

Vent acceleration processing is initialized at the beginning of the integration back to the stopping condition by setting up a vent sample interval prior to the integration interval of length in time equal to one-half of the time interval corresponding to one Runge-Kutta step size.

No cutoff conditions will be monitored until the minimum delta time of integration has been satisfied. This allows added flexibility to the user who knows how his trajectory behaves. For example, if he wants to stop the integration process at perigee, he can be assured of not terminating on the zero path angle at apogee by specifying a minimum delta time of integration before the flightpath angle will be monitored.

Upon entry, ECENEE checks system parameter MGREPH. If MGREPH is not zero, ECENEE sets an error flag and immediately terminates integration; otherwise, processing proceeds.

b. Integration step size selection

When the independent variable of integration is time, the integration step size must decrease as the vehicle nears the central body where the gravitational effects become larger. Stored in this routine is an array of numbers for each central body, representing radial distances from the corresponding central force field with a table of step size values, each of which corresponds to an interval in the array of radial distances. To ensure a degree of accuracy, nominal step sizes are chosen after each integration step by noting the value of the position magnitude and selecting the corresponding step size for the next step. If this requires a change in step size, rebuilding the difference table (a restart) becomes necessary and a rectification will occur. When β is the independent variable, the above procedure is not necessary, since an increment in β near the central body corresponds to a smaller time interval than the same increment of β farther from the central force field. Therefore, a change in the β step size will not occur.

c. Rectification

When the Encke integration scheme is being used, the process of redefining the base point is referred to as rectification. Rectification occurs automatically after a reference body switch or when the integration step size changes. A rectification is desired when the perturbative position or velocity vectors become greater than a percentage of the true position or velocity vectors. Therefore, rectification will also occur when

$$\frac{|\bar{\xi}|^2}{r^2} \geq \xi^2 \text{ or } \frac{|\dot{\bar{\xi}}|^2}{v^2} \geq \epsilon^2 \quad (1)$$

where $\bar{\xi}$ = the position due to perturbative forces other than the central body
 r = the true position magnitude
 v = the true velocity vector magnitude
 ϵ = the defined epsilon or percentage

If, after either of these tests, a rectification is desired, the base point is redefined by

$$\begin{aligned}\bar{R}_{base} &= \bar{R}_{2-body} + \bar{\xi} \\ \dot{\bar{R}}_{base} &= \dot{\bar{R}}_{2-body} + \dot{\bar{\xi}}\end{aligned}$$

and the vectors $\bar{\xi}$ and $\dot{\bar{\xi}}$ are zeroed.

Vent acceleration processing is also initialized by setting up a vent sample interval prior to the integration interval of length in time equal to one-half of the time interval corresponding to one Runge-Kutta step size.

d. Ephemeris storage

Program ECENEE stores an ephemeris at a variable time step in the format T, \bar{R}, \bar{V} . Each integration step is stored unless the user specifies a minimum output step. In this case, a special algorithm is used which was developed for storing a variable step ephemeris at a multiple of the natural variable step size, subject to a desired minimum interval.

This algorithm, developed by H. L. Normal of IBM, states that when the time required to reach the next minimum output point is less than one-half of the time span of the last integration step, the vector at the end of the last integration step should be stored. The minimum step size specified by the user is not regarded as an absolute minimum but is used as a guide in determining the output step. The algorithm prevents excessive use of space by not storing at too small an interval, guards against too large an interval with resulting interpolation errors, and gives an almost constant step size within the time domain where a certain multiple "n" is used. The discontinuity points where "n" changes are given by

$$\Delta t = n\Delta t_i = \frac{2\Delta t_{min}}{2n + 1} \quad (2)$$

where $n = 1, 2, \dots$. For $\Delta t_{min} = 3$ minutes, the ephemeris would be stored as follows:

$$\begin{aligned}2 \text{ minutes} \leq \Delta t_i < \infty & \quad \Delta t = \Delta t_i \\ 1.2 \text{ minutes} \leq \Delta t_i < 2 \text{ minutes} & \quad \Delta t = 2\Delta t_i \\ 51 \text{ seconds} \leq \Delta t_i < 1.2 \text{ minutes} & \quad \Delta t = 3\Delta t_i\end{aligned} \quad (3)$$

When the maximum number of allowable vectors has been stored, the integration process is terminated. The user may also specify whether a single or double precision ephemeris is desired. If a single precision ephemeris is desired, ECENEE calls ECENSP to output a rounded single precision ephemeris (vectors only single precision) into the user's output area.

2.4.2.2 Effective Forces Routine (ECENEF)

This routine evaluates the total accelerations upon a vehicle in the gravitational field of three attracting bodies (Earth, Moon, and Sun), with the vehicle also subject to other perturbing forces such as venting, drag, and oblateness. The equations of motion of a vehicle in this gravitational field are given by

$$\ddot{\bar{R}}_v = - \sum_{i=1}^3 \mu_i \frac{\bar{R}_{vi}}{r_{vi}^3} + \bar{F}_0 + \bar{F}_1 + \bar{F}_2 \quad (4)$$

where $\ddot{\bar{R}}_v$ = the total accelerations on the vehicle

μ_i = the gravitational constant of the *i*th body

\bar{R}_{vi} = the position vector of the vehicle with respect to the *i*th attracting body

r_{vi} = the magnitude of R_{vi}

\bar{F}_0 = the acceleration due to the oblateness of the central body

\bar{F}_1 = the acceleration due to drag

\bar{F}_2 = the acceleration due to venting

These equations are put into an observable form by referring them to a reference body *c* (Earth in shuttle). The equations of motion of the reference body are

$$\ddot{\bar{R}}_c = - \sum_{\substack{i=1 \\ i \neq c}}^3 \mu_i \frac{\bar{R}_{ci}}{r_{ci}^3} \quad (5)$$

where $\ddot{\bar{R}}_c$ = the total accelerations on the reference body

\bar{R}_{ci} = the position vector of the reference body with respect to the *i*th body

r_{ci} = the magnitude of R_{ci}

Subtraction of equation (5) from equation (4) results in the equation of the vehicle with respect to the reference body c.

$$\ddot{\bar{R}}_{cv} = -\mu_c \frac{\bar{R}_{vc}}{r_{vc}^3} - \sum_{\substack{i=1 \\ i \neq c}}^3 \mu_i \left[\frac{\bar{R}_{vi}}{r_{vi}^3} - \frac{\bar{R}_{ci}}{r_{ci}^3} \right] + \bar{F}_0 + \bar{F}_1 + \bar{F}_2 \quad (6)$$

where $\ddot{\bar{R}}_{cv}$ = the total accelerations on the vehicle with respect to the reference body

μ_c = the gravitational constant of the reference body

\bar{R}_{vc} = the position vector of the vehicle with respect to the reference body

r_{vc} = the magnitude of \bar{R}_{vc}

The integrator processor has the capability of numerical integration by the Cowell and Encke methods. In the Cowell scheme, equation (6) is integrated directly. The Encke method takes advantage of the fact that the motion of a vehicle normally deviates only slightly from two-body motion. Therefore, instead of integrating the total acceleration, only the deviations of the actual motion from that of the reference conic are integrated. The Encke form of equation (6) can be written as

$$\ddot{\bar{R}} = \ddot{\bar{R}}_{2-body} + \ddot{\xi} \quad (7)$$

Where

$$\ddot{\xi} = \mu_c \left(\frac{\bar{R}_{vc}}{r_{vc}^3} - \frac{\bar{R}_{2-body}}{r_{2-body}^3} \right) - \sum_{\substack{i=1 \\ i \neq c}}^3 \mu_i \left(\frac{\bar{R}_{vi}}{r_{vi}^3} - \frac{\bar{R}_{ci}}{r_{ci}^3} \right) + \bar{F}_0 + \bar{F}_1 + \bar{F}_2 \quad (8)$$

The first term of equation (8) is frequently referred to as the Encke term. The solution R_{2-body} is obtained by solving Kepler's equation in a form attributed to Herrick, described in the writeup of subroutine ECENTB.

The perturbative accelerations caused by the attraction of the two secondary bodies expressed in equation (8) are of the form

$$\ddot{\bar{R}}_i = \sum_{\substack{i=1 \\ i \neq c}}^3 \mu_i \left(\frac{\bar{R}_{vi}}{r_{vi}^3} \right) - \frac{\bar{R}_{ci}}{r_{ci}^3} \quad (9)$$

For example, in Earth reference, the force due to the Moon's attraction is expressed as

$$\ddot{\bar{R}}_m = -\mu_m \left(\frac{\bar{R}_{vm}}{r_{vm}^3} - \frac{\bar{R}_{em}}{r_{em}^3} \right) \quad (10)$$

where $\ddot{\bar{R}}_m$ = the accelerations on the vehicle caused by the Moon while in Earth reference

\bar{R}_{vm} = the position vector of the vehicle with respect to the Moon

r_{vm} = the magnitude of R_{vm}

\bar{R}_{em} = the position of the Earth with respect to the Moon

r_{em} = the magnitude of R_{em}

μ_m = the gravitational constant of the Moon

Equation (10) may create numerical problems when the vehicle is very near the central body because of the loss of accuracy in subtracting nearly equal terms. The same problem is presented in calculating the Encke term by equation (8). An expression based on the binomial expansion of the difference of the reciprocals of cubes improves the situation.

$$\frac{\bar{R}}{r^3} - \frac{\bar{R}_o}{r_o^3} = \frac{1}{r_o^3} \left[\bar{R} \left(\frac{L^3 + 3L^2 + 3L}{(1+L)^{3/2} + 1} \right) + \Delta \bar{R} \right] \quad (11)$$

where

$$\Delta \bar{R} = \bar{R} - \bar{R}_o \quad (12)$$

$$L = \frac{-2 \bar{R}_o \cdot \Delta \bar{R} - \Delta \bar{R} \cdot \Delta \bar{R}}{r_o} \quad (13)$$

$$\frac{r_o^3}{r^3} + 1 = (1+L)^{3/2} + 1 \quad (14)$$

The equations used to account for the oblateness potential of the Earth are of common form. The oblateness effects of all noncentral bodies (Sun and Moon) are ignored by the program. To minimize expensive calls to ELEFEM for Sun/Moon vectors, ECENEF performs linear interpolations on Sun/Moon vectors (with a magnitude correction) over 4-hour timespans. When the time progresses

beyond the current 4-hour interval, ECENEF calls ELEFEM (see section 2.2.6.2) and sets up the next 4-hour interval.

The form of the equation for the oblateness potential dictates that it be evaluated in a body-fixed reference frame (geographic). The inertial reference frame of integration used by the program is based on the mean equinox of 1950 which enables direct use of ephemeris tape data. Details about the potential acceleration computation can be found in the writeup of program ECENMP (section 2.2.7.1).

In the computation of the drag acceleration, the Jacchia atmosphere is being used. (See writeup of ECJACHIA, section 2.2.6.1). The standard 1962 atmosphere (see section 2.2.6.2) is used below 125 km. The drag equations in component form are

$$\ddot{X}_d = -\frac{1}{2} C_d K p (A/W) |\bar{V}| * \left(\dot{X} - (\bar{\Omega} \times \bar{r})_X \right) \quad (15)$$

$$\ddot{Y}_d = -\frac{1}{2} C_d K p (A/W) |\bar{V}| * \left(\dot{Y} - (\bar{\Omega} \times \bar{r})_Y \right) \quad (16)$$

$$\ddot{Z}_d = -\frac{1}{2} C_d K p (A/W) |\bar{V}| * \left(\dot{Z} - (\bar{\Omega} \times \bar{r})_Z \right) \quad (17)$$

$$\bar{V} = \left(\dot{X} - (\bar{\Omega} \times \bar{r})_X \right), \left(\dot{Y} - (\bar{\Omega} \times \bar{r})_Y \right), \left(\dot{Z} - (\bar{\Omega} \times \bar{r})_Z \right) \quad (18)$$

where \ddot{X}_d = x component of drag acceleration

\ddot{Y}_d = y component of drag acceleration

\ddot{Z}_d = z component of drag acceleration

C_d = drag coefficient

K = K-factor

A = frontal surface area of vehicle

W = weight of vehicle

p = density of atmosphere at given altitude (Z)

$\bar{\Omega}$ = Earth's rotational velocity vector (not necessarily along Z-axis)

$|\bar{V}|$ = speed of vehicle with respect to the rotating Earth

$(\bar{\Omega} \times \bar{r})_X$ = X component of $\bar{\Omega} \times \bar{r}$, etc.

For attitude-independent drag, C_d and A are constant, and K is used to apply drag corrections that are usually empirically determined during mission operations. For the case of attitude-dependent shuttle drag calculations, an attitude-dependent drag coefficient is computed from the expression

$$C_d(\alpha, \beta) = \left(C_{df} + C_{dn} |\sin \alpha|^n \right) (1 - |\sin \beta|) + C_{ds} (|\sin \beta|) + C_{da} |\sin 2\beta \sin \alpha|$$

where α and β are the attack and sideslip angles calculated in vehicle body coordinates by means of the equations.

```

•  $\frac{T}{V_b} = \frac{T}{V}^{T(ROMAT)}$ 
IF  $\left( (V_{bx}^2 + V_{bz}^2) \text{EQ. } 0 \right)$  THEN
     $\alpha = 0$ 
ELSE  $\alpha = \cos^{-1} \left( V_{bx} / (V_{bx}^2 + V_{bz}^2)^{\frac{1}{2}} \right)$ 
IF  $(V_{bz} \text{LT. } 0)$  THEN
     $\alpha = -\alpha$ 
ENDIF
ENDIF
NOTE:  $-\pi < \alpha \leq \pi$ 

```

where $\left. \begin{matrix} C_{da} \\ C_{df} \\ C_{dn} \\ C_{ds} \\ n \end{matrix} \right\}$ constants for attitude-dependent drag coefficient expression

ROMAT - 3×3 matrix defining body to inertial coordinate (M50) rotation (computed by ECENBY)

In the case of attitude-dependent drag, a special correction is made to the coefficient of drag for low altitude. Specifically, if the oblate altitude is less than 600k feet and the payload bay doors are closed, then the coefficient of drag C_D described above is corrected as shown below. The constants appearing in the following equations are all system or trajectory parameters.

- Call ECLOAD.

(b) If not SS1, go to step 11.

10. RNP matrix load supervisor (ECLOAD)

Action: ASP request for an RNP matrix.

Reaction: An RNP matrix is supplied whose timetag is within 3 hours of the timetag of the input vector.

11. A priori setup processor (APS)

Action: CSTM supplies requested covariance.

Reaction: ASP applies K-gammas, if requested, then computes the inverse of the input covariance matrix.

The ASP also makes sure that the correct TDRS ephemeris is available. It does this by setting a pointer to the BB-control-table-specified external TDRS ephemeris or by calling the numerical integration supervisor to generate an internal TDRS ephemeris spanning the TDRS tracking data arc interval based on the BB-control-table-specified TDRS vector. When the work of the ASP is complete the DCE is notified.

12. Differential correction executive (DCE)

Action: ASP tells DCE that data gathering is complete.

Reaction: DCE sets iteration count to zero and calls CP.

13. Convergence processor (CP)

Action: CP is invoked by the DCE to begin differential correction process.

Reaction: CP checks for solution convergence. If the solution has converged, go to step 2b; if not, go to step 14.

14. Convergence processor (CP)

Action: Solution has not converged.

Reaction: CP increments iteration count by one and computes the current state correction and covariance matrix by going through steps 15 to 22.

15. Differential correction module (DCM)

Action: Request for a differential correction from the CP.

Reaction: The DCM checks the type of request. There are two types of requests. The first type is a request to generate a vehicle ephemeris and compute residuals based on the current solution state. This is performed after iterations are complete and before a new differential correction, if BB editing has been requested. For this type of request move to step 23.

The second type of request is to compute an update to the current solution state, using an input external ephemeris or an internally generated ephemeris. For this type go to step 16.

16. Differential correction module (DCM)

Action: Request for an update to the current solution state.

Reaction: If an internal ephemeris is to be used, call the numerical integration supervisor (EMSMISS) to generate a vehicle ephemeris of position and velocity vectors covering the batch solution tracking data arc based on the input state vector.

The vehicle characteristics (weight area and constant area drag multiplier) of the external ephemeris assigned to the link will be used in the propagation of the input vector.

If ephemeris zero is assigned to the link, then the input vector will be propagated using the vehicle characteristics (weight, area and constant area drag multiplier) specified in the BB control table for the link.

If an external ephemeris is to be used, vectors from the specified ephemeris within the batch solution tracking data arc timeframe are used.

The batch to be processed is read by the DCM. Batch-dependent OCM inputs and observation-dependent inputs for each data frame are assembled and given to the OCM.

17. Observation computation module (OCM)

Action: The OCM is invoked by the DCM for a DC, the BRP display processor for a batch residual plot, and the residual summary display (RSD) processor for a vector residual summary request.

Reaction: The OCM computes signal path parameters, a computed observation, and corrects the observation for refraction, if specified by the station characteristics, based on the input ephemeris.

The OCM then computes the residuals, actual tracking data measurements minus OCM-generated measurements computed from the input ephemeris. The DCM is notified when processing is complete.

18. Differential correction module (DCM)

Action: Invoked when OCM is finished.

Reaction: The DCM calls the MPDM to compute the measurement partial derivatives matrix, which consists of the partials of the OCM-computed observations with respect to the ephemeris generated, based on the current solution vector.

19. Measurement partial derivative module (MPDM)

Action: Request from DCM for partial derivative matrix.

Reactions: The MPDM computes the following:

- Partial derivatives of computed observations with respect to vehicle position (M50 coordinates)
- Partial derivatives of computed observations with respect to vehicle velocity (MSO coordinates)

The derivatives are then assembled into the 1×6 matrix.

The DCM is notified when completed.

20. Differential correction module (DCM)

Action: MPDM notifies that the 1×6 matrix is available.

Reaction: The DCM calls the STMM to compute a state transition matrix.

21. State transition matrix module (STMM)

Action: The DCM requests a state transition matrix.

Reaction: The STMM computes the position and velocity state transition matrix from anchor time to request time. The DCM is notified when computations are completed.

22. Differential correction module (DCM)

Action: The STMM notifies the DCM that processing is complete.

Reaction: The DCM increments totals for the DC process, computes an updated covariance matrix, and computes the solution state correction. The DCM then supervises the computation and accumulation of residuals described in step 23.

23. Differential correction module (DCM)

Action: Residual request from CP.

Reaction: The DCM calls EMSMISS to generate a vehicle ephemeris across DC data arc.

24. Numerical integration supervisor (EMSMISS)

Action: DCM request for vehicle ephemeris.

Reaction: EMSMISS generates a vehicle ephemeris across the DC data arc interval. The DCM is notified when processing complete.

25. Residuals and statistics processor (RSP)

Action: The DCM calls the RSP to compute and accumulate residuals when notified that the ephemeris generation is complete.

Reaction: The ASP computes residuals for each observation in the batch, accumulates a running total of residual sums and observation counts, and calculates the averages and standard deviations, if required. This information is then given to the CP. Go to step 13.

26. Differential corrections executive (DCE)

Action: DCE invoked by CP.

Reaction: The DCE checks to see if a BB auto edit request is on. If so, then the DCE calls the DEP (step 27). If not, the DCE updates the current solution table. Go to step 1.

27. Data editing processor (DEP)

Action: DEP is invoked by the DCE.

Reaction: The DEP copies the current batch into a buffer and tags all the points as included, determines outlying points based on the residuals, tags all points appropriately in

the buffer copy, copies the edited batch over the original, and keeps track of the number of edited points for each observation type. When editing is complete, the DCE is notified. .

28. Differential corrections executive (DCE)

Action: The DCE receives notice that editing is complete.

Reaction: The DCE checks to see if any edits were made. If edits were made, then go to step 12. If edits were made, then set indicator.

SECTION 4

MISCELLANEOUS TRAJECTORY PROCESSORS

In sections 2 and 3 you learned about those routines involved in the trajectory prediction and orbit determination processes.

This section covers those processors which are not directly involved in the orbit trajectory determination process but have been included in the document because they are very important during mission support/console operations. Each processor is described by a data flowchart.

4.1 PREDICTED SATELLITE ACQUISITION DISPLAY PROCESSOR

Predicted satellite acquisition display (PSAD) processing is initiated only by MED request. The specified spacecraft ephemeris, PSAD table ID (orbiter or payload), request time, station characteristics, and minimum elevation angle are used to generate station contacts. The terrain masking is applied to selected contacts and then all contacts are stored chronologically in the PSAD save table. All information in the save table is available for MCC console display.

The PSAD processor (fig. 4-1) interfaces with the following generators and/or modules.

4.1.1 Predicted Site Processor

This module controls the generation of data for the PSAD.

4.1.2 Special Stations Contacts Generator

This module provides an interface between the predicted site processor and the generalized stations contacts generator.

4.1.3 Generalized Stations Contacts Generator

This module provides an interface with the computational modules to compute the horizon crossing data and apply terrain masking to generate contact. Generated contact data is sorted chronologically on acquisition of signal (AOS) times and returned to the requested module.

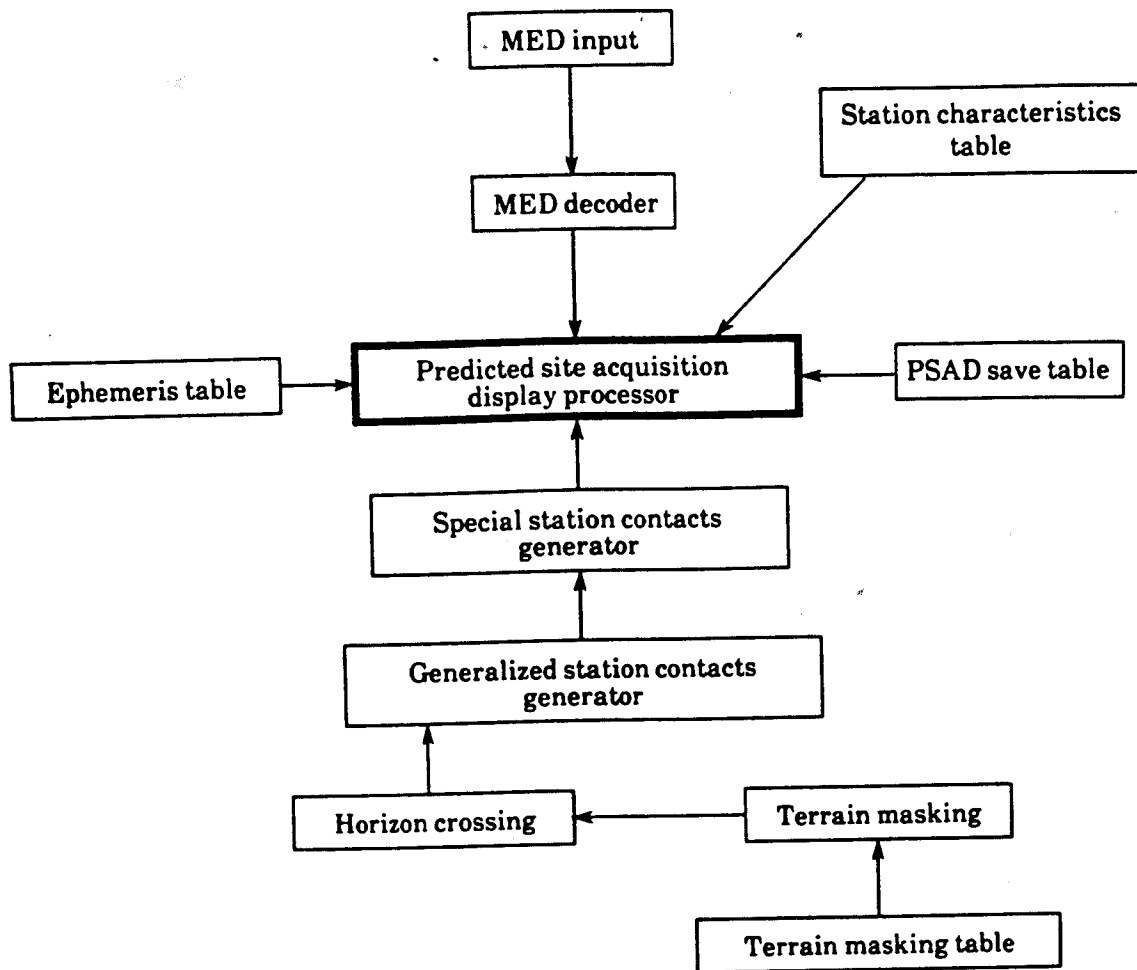


Figure 4-1.- PSAD processor.

4.1.4 Horizon Crossing

This module determines the number of times a ground site "contacts" a specified spacecraft during a given timespan. The data computed for each contact are

- Ascending horizon crossing time
- Descending horizon crossing time
- Maximum elevation above horizon plane or point of closest approach
- Time of maximum elevation

4.1.5 Terrain Masking

This module performs further processing to adjust contact data for antenna obstructions caused by the local station terrain. A table of unique azimuth and elevation points, ordered by increasing azimuth values from 0° to 360°, describing the local terrain, provides the basis for the terrain masking computations.

a. Inputs

- MED parameters
- Specified spacecraft ephemeris
- Station characteristics

b. Outputs

PSAD table

4.2 SATELLITE ACQUISITION PROCESSOR

The satellite acquisition processor (SAP) provides orbiter-to-TDRS rise/set times and AOS/loss of signal (LOS) blockage time of the Ku-band-to-TDRS LOS blocked by the orbiter body. Reference figure 4-2.

The SAP can generate up to 18 orbiter-to-TDRS contacts. For each contact, up to seven AOS/LOS blockage determinations may be computed for each of two Ku-band antennas. The SAP may be queued upon a MED input or upon the completion of an ephemeris update for any of the orbiter or two TDRS ephemerides specified. Upon being queued by either method, the common celestial acquisition routine is EKLINKed to compute rise/set times. Then the blockage data is computed with the orbiter orientation determined through the routine's ATL user service and matrix computations. All data is subsequently stored in either the SAP current table or the SAP planning table; this is determined by the trajectory update queue or the miscellaneous data comp and storage queue, respectively.

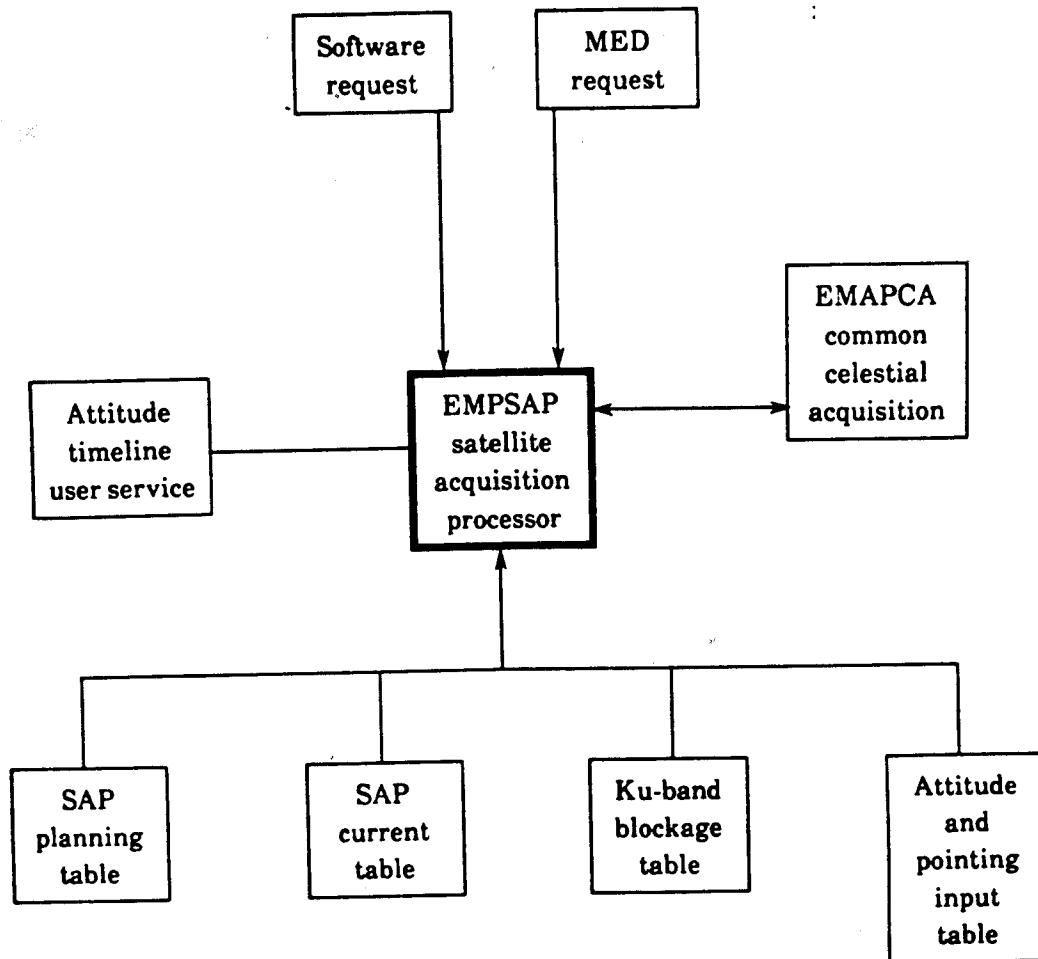


Figure 4-2.- Satellite acquisition processor (EMPSAP).

a. Inputs

- Ephemeris tables
- Ku-band composite blockage table
- Ku-band curve table
- Attitude and pointing input table

- SAP display control MED
- Requested Ku-band blockage MED
- SAP generation

b. Outputs

- Data tables
- SAP output table
- SAP current or SAP planning EDD

4.2.1 Supporting Routines

4.2.1.1 Common Celestial Acquisition Processor

Upon request the celestial acquisition processor (CAP) obtains celestial target definitions from the celestial target table. This information is required for inertial hold attitudes such as the solar inertial hold. The target IDs used are based on a block of target IDs within the star group table or up to five target IDs manually specified. CAP then queues the common celestial acquisition routine, optionally specifying an elevation constraint. The common celestial acquisition routine computes the indicated number of acquisition times, based on the specified celestial targets. Upon generating up to 20 acquisitions, CAP stores the acquisition times and target vectors in the CAP output table, EZAKCELA.

4.2.1.2 Attitude Timeline User Service

Refer to section 2.3.1.

4.2.1.3 Ku-Band Blockage Table (EZAZCBLK)

Contains Ku-band antenna blockage data. This information is generated by the SAP blockage computation processor.

4.3 STATION CONTACTS GENERATOR (EMGSTGEN)

Next station contacts processing is queued upon request or after a trajectory update. The specified orbiter or payload ephemeris, elevation angle, and station characteristics are used to generate up to 24 contacts. Terrain masking is applied to selected contacts and then all contacts are stored chronologically in the station contacts table (EZSTACTX) corresponding to the specified ephemeris. Keyhold constraints are computed and stored with their associated contacts.

EMGSTGEN controls next station contacts (NSCs) and entry predicted site acquisition data (EPSAD) processing. As such, its products are the station contacts tables, the acquisition contacts tables, and the EPSAD table. Its primary responsibilities are

- a. Retrieve selected spacecraft ephemeris (NSC, EPSAD)
- b. Retrieve TDRS ephemeris (EPSAD)
- c. Retrieve station characteristics (NSC, EPSAD)
- d. Retrieve TACAN characteristics (EPSAD)
- e. Queue generalized station contacts generator (EMGENGEN) to interface with computational modules (NSC, EPSAD)
- f. Queue keyhold constraints generation (NSC)
- g. Queue common celestial acquisition to generate TDRS contacts
- h. Inputs
 - MED parameters
 - Current time
 - Specified spacecraft ephemeris
 - TDRS ephemerides associated with current satellite acquisition table (SAT)
 - Station characteristics
- i. Outputs
 - Station contacts table
 - Acquisition contacts table
 - EPSAD table

4.3.1 Supporting Routines

Refer to figure 4-3 as you read this section.

4.3.1.1 Generalized Station Contacts Generator (EMGENGEN)

This module provides an interface with the computational modules to compute horizon crossing data and apply terrain masking to generated contacts. Generated contact data is sorted chronologically on AOS times and returned to the requesting module.

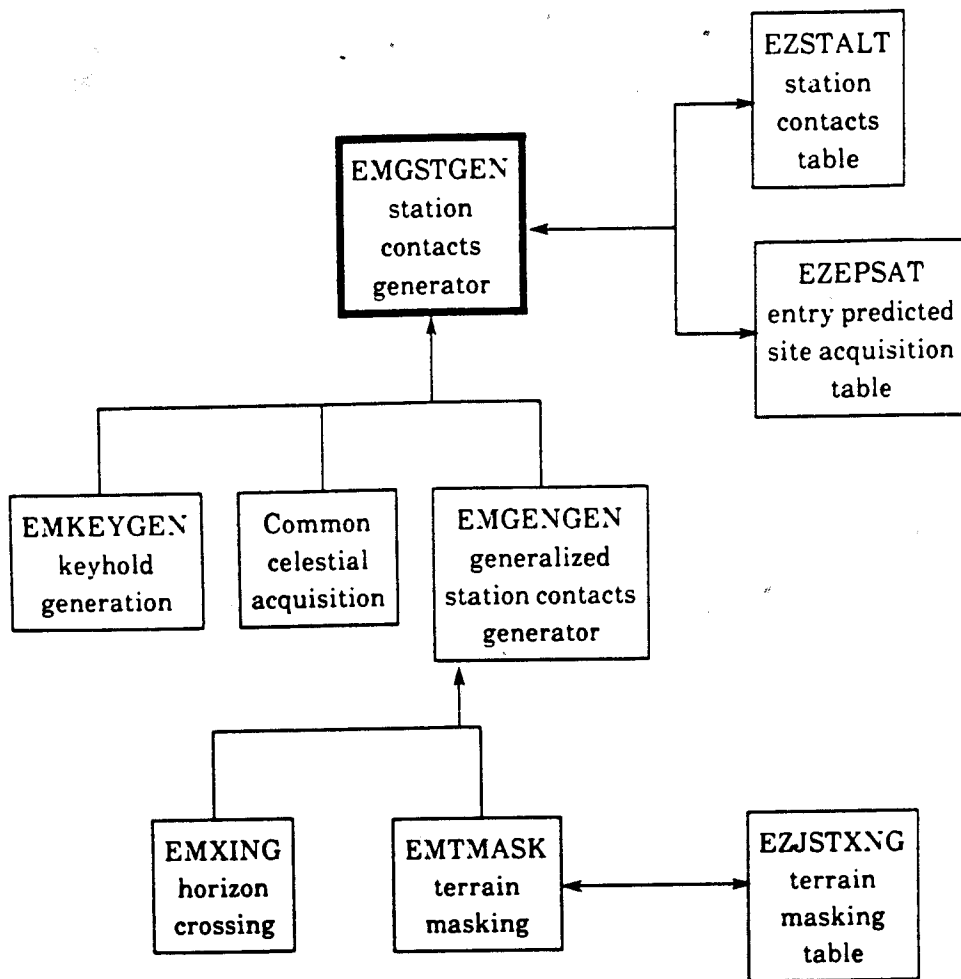


Figure 4-3.- Station contacts generator (EMGSTGEN).

4.3.1.2 Horizon Crossing (EMXING)

This module determines the number of times a ground site "contacts" a specified spacecraft during a given time span. The data computed for each contact are

- Ascending horizon crossing time (AHCT or GMT AOS)
- Descending horizon crossing time (DHCT or GMT LOS)
- Maximum elevation above horizon plane or point of closest approach (EMAX)
- Time of EMAX (GMT EMAX)

EMXING expects an angle of elevation relative to the horizon plane which is used in the **AHCT** and **DHCT** computations.

4.3.1.3 Terrain Masking (EMTMASK)

This module performs further processing to adjust ground tracking station contact data for antenna obstructions caused by the local station terrain. A table of unique azimuth and elevation points, ordered by increasing azimuth values from 0° to 360°, describing the local terrain, provides the basis for the terrain masking computations.

4.3.1.4 Keyhold Generation (EMKEYGEN)

This module determines the keyhold constraints for all S-band radar contacts in the station contacts tables. The keyhold data computed for each such contact are

- Time of entry into keyhold; i.e., time of keyhole LOS (GMT KEYLOS)
- Time of exit from keyhold; i.e., time of keyhole AOS (GMT KEYAOS)

EMGSTGEN (fig. 4-3) interfaces with the following routines.

4.4 DELTA-TIME PROCESSOR

The **DTP** is used to compute a delta time to be applied to a state vector timetag in order to minimize instantaneous downtrack position error.

The **DTP** (fig. 4-4) is queued initially via **MED** by specifying the link, anchor vector, threshold time, and station for which data is to be processed. An **EDD** is illuminated when the **DTP** is enabled and remains on until the **DTP** is inhibited.

The **DTP** is able to access both open and closed batches of radar data. The batches must meet the following criteria:

- The data must be from the station specified on the initialization **MED**.
- The anchor time of the batch must be greater than or equal to the threshold time specified.
- If more than one batch meets the first two criteria, the one with the earliest anchor time is chosen.

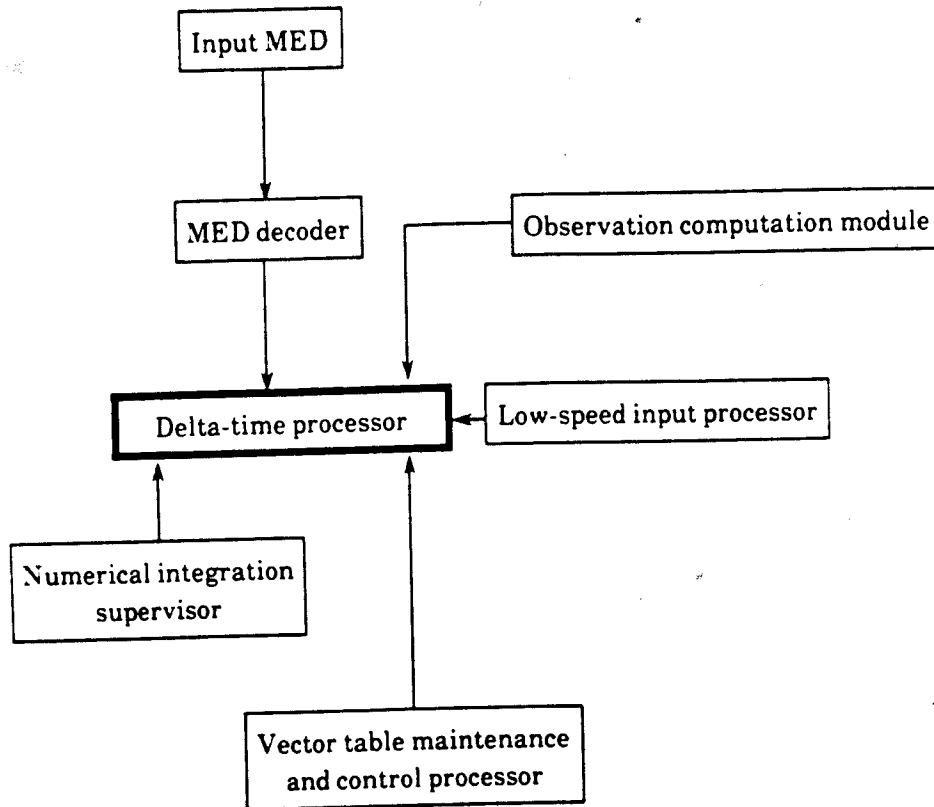


Figure 4-4.- DTP data flow.

When a closed batch is fetched by the DTP, it processes all the data frames in the batch (up to 50 will be displayed on the tables). If the DTP fetches an open batch, it processes all the saved data frames, then it waits for a queue from the LSIP to process new incoming radar data. If the batch ID cannot be found, the DTP remains enabled and waits for an LSIP queue to start processing data as soon as the LSIP receives the data and sends it to the DTP. The DTP is automatically inhibited when a batch closes.

The vector specified on the initialization MED is propagated by the free-flight numerical integrator (ECENCKE) to a time just prior to the anchor time of the batch being processed. The OCM is then called to compute range and Doppler observations based on the ECENCKE-generated ephemeris.

The partials of downtrack error with respect to the range and Doppler observations from the Nth data frame are computed as follows:

$$\left(\frac{\partial DT}{\partial R} \right)_N = \frac{V_N (\Delta TP)}{R_{2N} - R_{1N}}$$

$$\left(\frac{\partial DT}{\partial D} \right)_N = \frac{V_N (\Delta TP)}{D_{2N} - D_{1N}}$$

where $\left(\frac{\partial DT}{\partial R} \right)_N$ is the partial of downtrack position with respect to the range observation for the Nth

data frame, $\left(\frac{\partial DT}{\partial D} \right)_N$ is a partial of downtrack position with respect to the Doppler observation for the Nth data frame, V_N is the spacecraft velocity at the time of the Nth data frame computed based on the external ephemeris 1, ΔTP is the timetag adjustment applied to the state vector used to create the ECENCKE-generated internal perturbed ephemeris, and R_{1N} , R_{2N} , D_{1N} , and D_{2N} are the observations computed by the OCM. R_{2N} and D_{2N} are from the internal perturbed ephemeris.

The internal perturbed ephemeris contains a vector propagated to a time that is equal to Δt + the timetag of a vector in the nominal ephemeris. An example of the nominal and perturbed ephemerides vector timetags is shown below.

Nominal ephemeris

Perturbed ephemeris

T_0
 T_1
 T_2
 T_3
 T_4
 \vdots
 \vdots
 \vdots
 T_n

$T_0 + \Delta TP$
 $T_1 + \Delta TP$
 $T_2 + \Delta TP$
 $T_3 + \Delta TP$
 $T_4 + \Delta TP$
 \vdots
 \vdots
 \vdots
 $T_n + \Delta TP$

Two estimates of the delta-T (Δt), based on range and Doppler observations, respectively, are computed for the Nth data frame as follows:

$$\Delta t_{RN} = \left(\frac{\partial DT}{\partial R} \right)_N \left(\frac{R_{0N} - R_{1N}}{V_N} \right)$$

$$\Delta t_{DN} = \left(\frac{\partial DT}{\partial D} \right)_N \left(\frac{D_{0N} - D_{1N}}{V_N} \right)$$

Where Δt_{RN} is the value of Δt based on the range observation, Δt_{DN} is the value of Δt based on the Doppler observation, R_{0N} is the observed range, and D_{0N} is the observed Doppler.

Prior to termination, the DTP determines whether additional data frames are available for processing without recalling batch fetch software. If no additional data frames are available, the DTP returns control to the calling software.

a. Inputs

- Flight controller MED - MED parameters
- LSIP queues - Incoming radar data observations

b. Outputs

Outputs from the DTP are three displays (DTP1, DTP2, and DTP3) and EDD. DTP1 and DTP2 display the range and Doppler residuals, partials, and delta time computed by the DTP for each data frame. DTP3 displays a plot of elapsed time from anchor time versus the computed delta time.

4.5 BVENT/LVENT PROCESSOR (B/L VENT)

The BVENT and LVENT processor provides the processing required to build displayable tables of

- The net vent forces in the orbiter body axis (BVENT processing)
- The net VTL vent forces in the UVW coordinate system (LVENT processing)

B/L VENT (fig. 4-5) is invoked via MED request. The MED inputs an ephemeris file number and a time interval and optionally specifies overrides to the selected ephemeris anchor vector and/or integrator options to be used for a temporary ephemeris generation. There are two primary functional units:

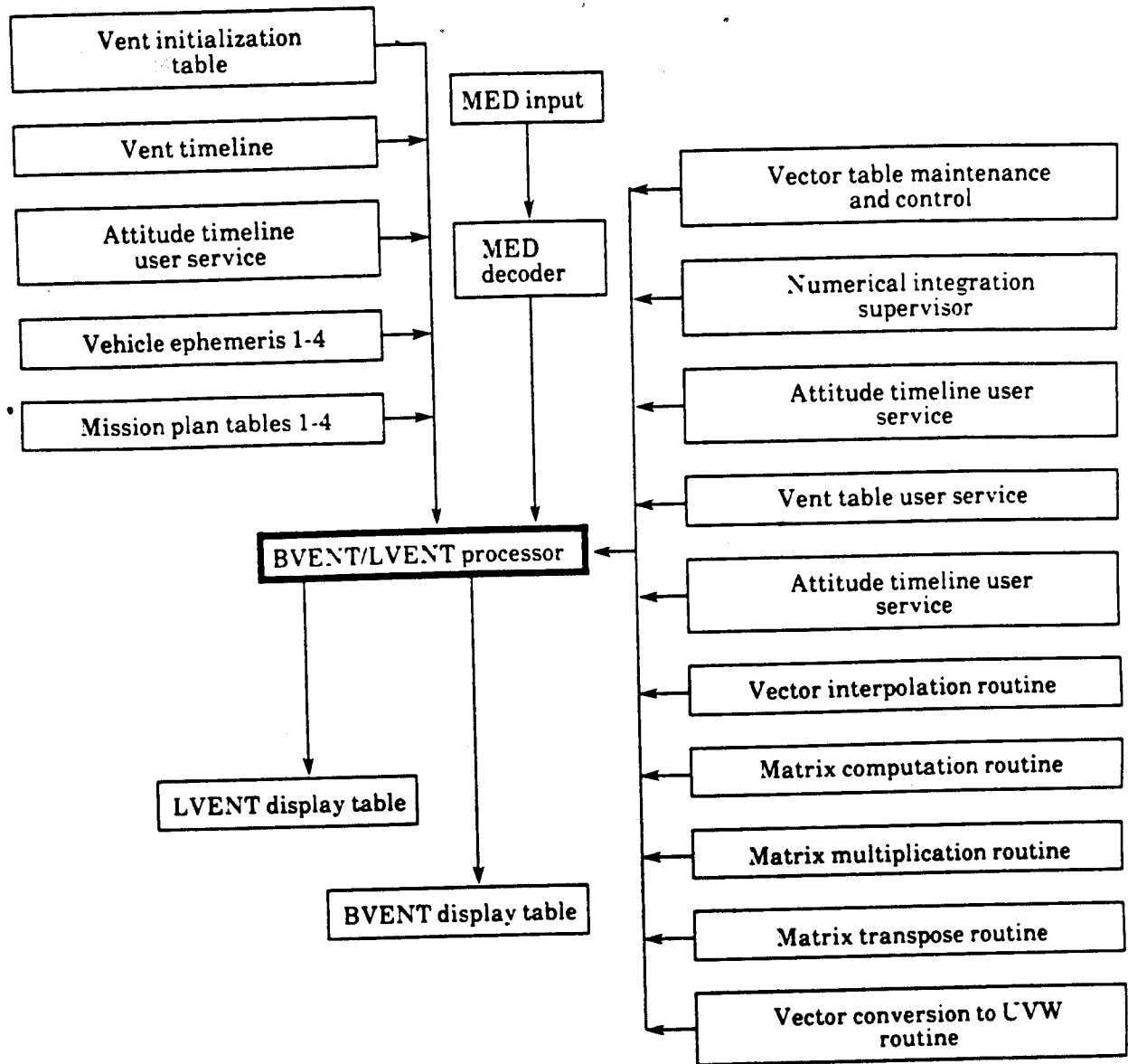


Figure 4-5.- BVENT/LVENT display processor data flow.

- a. BVENT processing - The BVENT function determines a maximum of 36 net VTL venting force intervals over the MED input timespan. A net venting force interval is a period of time during which the net venting thrust, in orbiter body coordinates, is constant and no active vent changes occur. The start time, the net venting forces, the number of active vents, and as many as nine active vent IDs are stored for each net venting force interval. The vent table user service routine is used to provide a low-level summation table, VTAB, to the BVENT function.

b. LVENT processing - If no vector or integrator options overrides are input on the MED, the specified profile supplies ephemeris data for LVENT processing. Otherwise, a temporary generated ephemeris, dependent upon the anchor vector and/or integrator options overrides, is used. Free-flight numerical integrator (ECENCKE) handles the ephemeris generation. The attitude timeline user service routine provides a body matrix table, BTAB, and an eigenvector table, ETAB, for the time interval. To plot display data, the interval is uniformly subdivided into 51 points. For each time point, the LVENT function will

- (1) Determine a vehicle state vector at the time point using the extended interpolation routine ELVCTR
- (2) Determine a body-to-M50 transformation matrix, based upon the vehicle position and attitude (using the matrix computations routine, ELMTXC, and the matrix transpose routine, ELRRMT)
- (3) Using the VTAB, find the net venting force vector at the time point
- (4) Compute the M50 venting force vector (using the matrix multiply routine, ELMMXD)
- (5) Transform the M50 vector to UVW coordinates (using vector conversion routine, ELVUVW).

c. Inputs

MED inputs

- Ephemeris ID
- Start time
- Stop time
- State vector override
- Integrator option override

d. Outputs

Updated BVENT and LVENT tables

4.6 CHECKOUT MONITOR DISPLAY PROCESSOR (EMDCHECK)

EMDCHECK consists of the processing required to obtain a vector to be used by the shuttle common display system (SCDS) computational display routines. The processing is queued (figure 4-6) upon manual requests. Inputs are manual inputs, maneuver tables, ephemeris tables, and the vector administration table. Output is a vector in the desired coordinate system (M50 or true of date rotating (TDR)) indicated in the manual request.

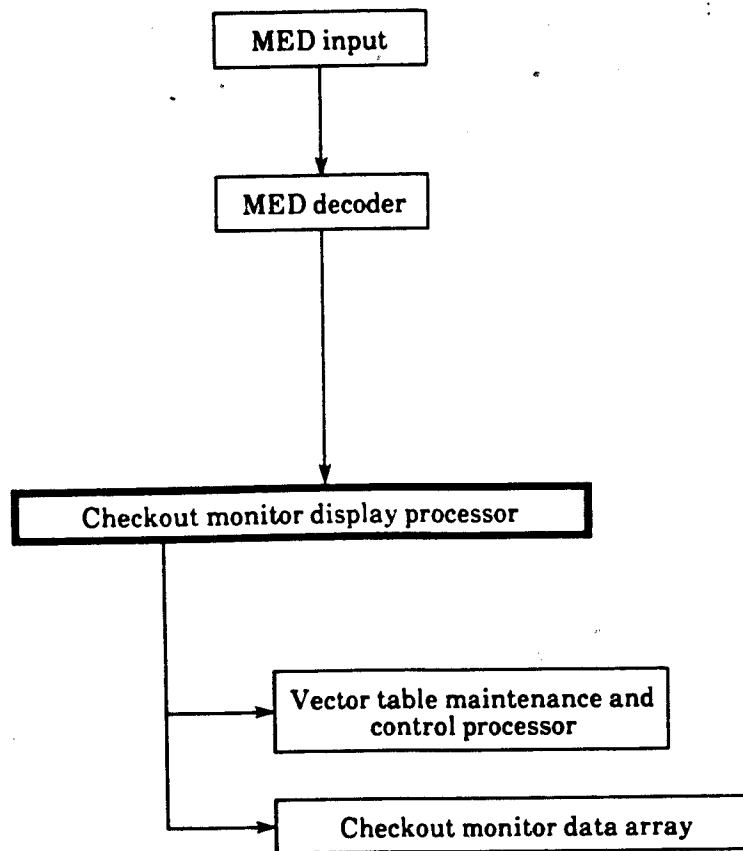


Figure 4-6.- Checkout monitor display processor (EMDCHECK).

To obtain a vector, EMDCHECK fetches a vector from the specified ephemeris table, interpolates against the specified ephemeris, using an eight-order interpolation routine, to arrive at a given time between the stored ephemeris vectors, or it retrieves an initiate or burnout vector from the specified maneuver table or retrieves any valid vector.

When a requested event is specified (e.g., a particular longitude or flightpath angle, etc.), then the vector is integrated to the requested stopping point. The retrieved and/or integrated vector is stored in the processor output table for SCDS access.

a. Inputs

MED parameters

- Ephemeris ID or link number
- Request indicator

- Threshold time
- Vector ID
- Integration options

b. Outputs

A processor table containing the desired vector and input request parameters

4.7 RELAY GEOMETRY DISPLAY PROCESSOR

In response to a MED request (fig. 4-7), the relay geometry display processor (RGDP) builds a table that may be accessed by TDR to output a plot of the target vehicle ground track for a specified interval. Points are obtained that identify the position of each TDRS at the center of the plotting interval. Additionally, times and angles of ground track closest approach between the target vehicle and each TDRS are computed and stored for display.

The RGDP is invoked by a MED that identifies a target vehicle and a timespan of interest. This timespan (display interval) is from one to three orbits long. The target vehicle ephemeris is obtained and used to build a table of 30 state vectors per orbit. The vectors are timetagged such that they are evenly spaced across the display interval. A state vector is obtained for each available TDRS timetagged at the center of the display interval.

The ground track times and angles of closest approach are then calculated between the target vehicle and each available TDRS.

a. Inputs

Relay geometry MED parameters.

b. Outputs

All outputs will be contained in the relay geometry table and the relay geometry ephemeris table.

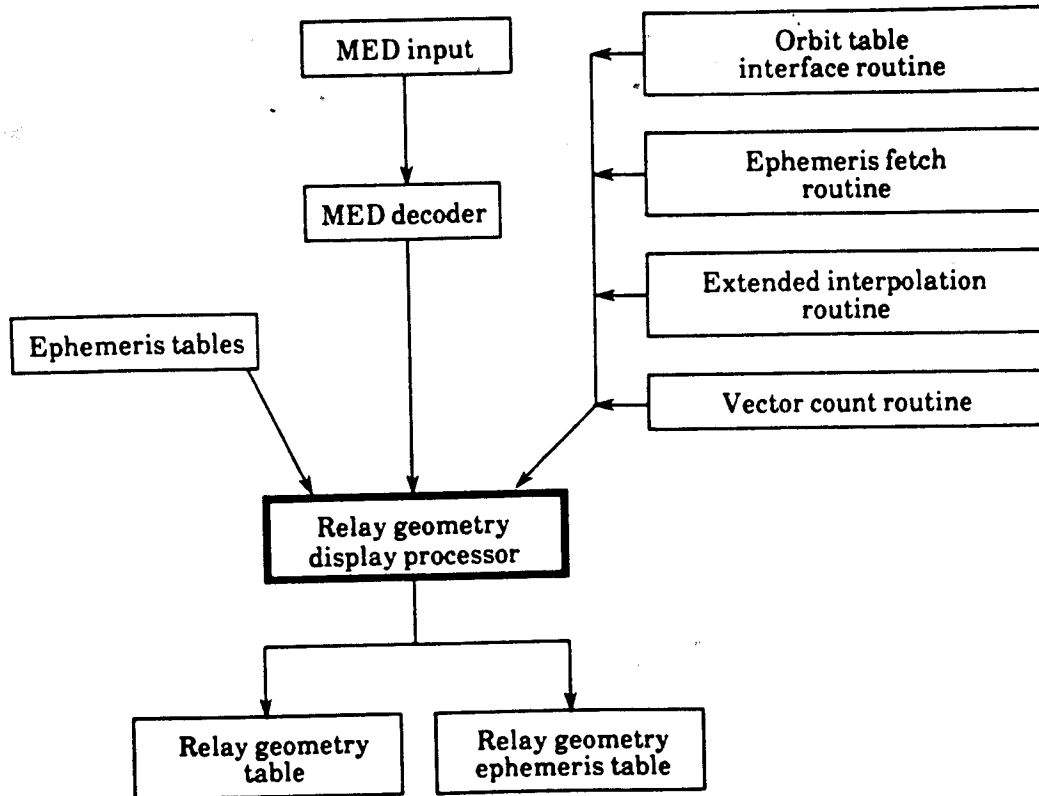


Figure 4-7.-- Relay geometry processor data flow.

4.8 VECTOR COMPARE PROCESSOR

The vector compare processor (VCP) contains the logic necessary for the comparison of vectors specified by MED and the display of the resultant parameters.

The VCP may be used as an aid in the evaluation of vectors from several different sources. It is invoked via MED entry. Then vectors to be compared are collected and integrated to the appropriate cutoff conditions.

4.8.1 Input Queues

The vector process is queued by MED or DRK. (See fig. 4-8.)

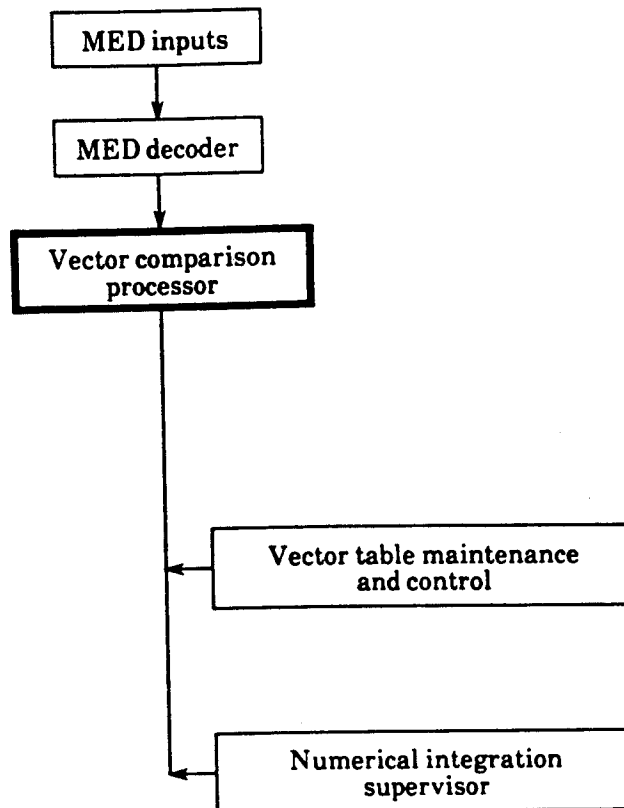


Figure 4-8.- Vector compare processor data flow.

The vector comparison MED specifies the DC link ID, base vector ID, up to three vector compare IDs, the integration cutoff option to be used, cutoff value, a threshold time, integrator option, display coordinate system reference, and display units. If parameters are omitted, the appropriate default values are used. The inputs are temporarily stored and are used only for that particular MED entry.

4.8.2 Vector Fetch

The first step is to locate each of the vectors specified. Any vector acceptable to the vector control program is valid. The vector ID is checked to see if it represents an ephemeris vector. If it does, its timetag is either the input time or the current time.

4.8.3 Vector Integration

After all the vectors have been located, they must be integrated to the appropriate cutoff conditions. The paragraphs below give a general outline of the integration process.

DC solution vectors, external ephemeris vectors, VAT slot vectors, and telemetry vectors are made available to the vector compare processor via the VTMC processor.

By using the appropriate S44 MED input options, the user has the capability to assign various integration modeling options. The vehicle characteristics (weight, area, and KCON) and modeling options of an external ephemeris can be used by specifying the link that the ephemeris is assigned to in the S44 MED.

If ephemeris zero is assigned to the link specified in the S44 MED, then the vehicle characteristics specified in the BB control table for the specified link will be used, and the integrator option will automatically be set to constant area drag.

The base vector is integrated to the specified cutoff condition, using the input integration options as specified by the user using the S44 MED. The associated time becomes T (cutoff) for the other vectors. They are integrated to T (cutoff), using the TIME option.

The various stopping options are listed below.

4.8.3.1 Time

- TIME - Phase elapsed time (PET) or GMT
- FPA - Cutoff on a flightpath angle
- ALT - Terminate on a specified altitude
- AN - Terminate on next upcoming ascending node
- LON - Cause integration to stop at a certain longitude
- RAD - Terminate integration upon a radial distance from the Earth

For all options except TIME, if a threshold time is input, the integrator integrates the base vector to that time before checking for the cutoff condition. If that time is defaulted, the integrator immediately begins checking for the cutoff condition.

If the integrator option requests maneuvers, checking for the cutoff condition does not occur during maneuvers. This would result in the integrator's not reaching the specified cutoff value if it fell during the time of the maneuver.

The logic for this option has two major divisions. The first is referenced when a VAT vector is the base vector; the second is used when an ephemeris vector is the base.

If a VAT vector is the base, it is fetched by the VTMC processor and integrated to the input time. When input time is defaulted, the base vector is displayed at vector time. The other vectors are integrated/interpolated to the time of the base vector.

If an ephemeris vector is the base, it is an interpolated vector at the specified time. If input time is defaulted, interpolation is performed for a vector at current time. As with the VAT vector, the other vectors are integrated/interpolated to the time of the base vector.

a. Inputs

MED parameters

- DC link ID
- Base vector ID
- Stopping option
- Option value
- Coordinate system reference
- Units
- Threshold time
- Integrator options

b. Outputs

Updated vector compare data array

APPENDIX A
PROCESSOR LOCATION WITHIN CCS TRAJECTORY OPERATIONS DESIGN
SPECIFICATIONS DOCUMENT

<u>Processor</u>	<u>Page</u>
A Priori Setup Processor	3-19
Analytic Ephemeris Generator	2-67
Attitude Timeline	2-17
Attitude Timeline User Service	2-17
BVENT/LVENT	4-11
Checkout Monitor	4-13
Convergence Processor	3-21
Vector Table Maintenance and Control	3-14
Covariance Storage Table Maintenance	3-16
Data Editing Processor	3-23
Delta Time Processor	4-8
Differential Correction Executive	3-13
Differential Correction Module	3-23
Effective Forces Routine	2-4
ET-UTC	2-77
Free-Flight Numerical Integration	2-28
Geopotential Routine	2-1
Jacchia Atmosphere Model	2-13
Low Speed Input Processor	3-4
Measurement Partial Derivative Module	3-29
MED Decoder	3-1
Numerical Integration Supervisor	2-66
Observation Computation Module	3-27
PBI Control	3-3
Powered-Flight Numerical Integrator	2-53
Predicted Satellite Acquisition Processor	4-1
Relay Geometry Processor	4-15
Residuals and Statistics	3-18
RNP Matrix Generator	2-75
Satellite Acquisition Processor	4-3
State Transition Matrix Module	3-17
Sun/Moon Vector Fetch	2-16
U.S. Standard 1962 Atmosphere Model	2-15
Vector Compare	4-16
Vector Maintenance and Control	3-14
Vehicle Data Table Maintenance	3-10
Weight Gain/Loss User Service	2-25
Weight Gain/Loss User Service	2.2.3E1WG TAB2

APPENDIX B
BIBLIOGRAPHY

- (1) **Ground Based Space Systems**
Command and Control Systems (CCS)
Trajectory Operations (OPS)
Design Specifications
June 24, 1987
JSC-18286

- (2) **OPS MCC**
Ground Navigation Program
Orbit Determination Processing
Formulation Requirements
Mission Planning and Analysis Division
October 1983
JSC-14266

- (3) **Mission Control Center**
Level B and C Requirements for Shuttle
Revision 11
Change 1
Ground Data Systems Division
July 23, 1984
JSC-11028

- (4) **On-Orbit Ground Navigation Console Handbook**
Mission Operations Directorate
Flight Design and Dynamics Division
June 1, 1988
JSC-20768

- (5) **C. Propst, J. Lacey, "Ground Navigation Leap Second/Year Rollover Investigation," STSOC**
Transmittal Form FDD-NAV-620-90-009, March 13, 1990.

APPENDIX C
ACRONYMS AND ABBREVIATIONS

AEG	analytic ephemeris generator
AHCT	ascending horizon crossing time
AOA	abort once around
AOS	acquisition of signal
APU	auxiliary power unit
ASP	a priori setup processor
ATCP	attitude timeline/computation processor
ATL	attitude timeline
ATUS	attitude timeline user service
B.O.	burnout
BB	batch to batch
BBQ	barbeque
BI	burn initiate
BRP	batch residual plot
BTAB	body attitude table
c.g.	center of gravity
CAP	celestial acquisition processor
CBDS	cargo bay door status
CCS	command and control system
CP	convergence processor
CST	covariance storage table
CSTM	covariance storage table maintenance
CSTP	coordinate system transformation processor
DC	differential correction
DCE	differential correction executive
DCM	differential correction module
DEP	data editing processor
DHCT	descending horizon crossing time
DOD	Department of Defense
DRK	display request keyboard
DTIG	deorbit time of ignition
DTP	delta-time processor
EDD	event display driver
ENU	east, north, up (coordinate system)
EOT	end of transmission

EPSAD	entry predicted site acquisition data
ET	ephemeris time
ETAB	eigenaxis table
FDO	flight dynamics officer
FFNI	free-flight numerical integrator
GHA	Greenwich hour angle
GMT	Greenwich mean time
I/O	input/output
ICV	intercenter vector
ID	identification
IH	inertial hold
JSC	Johnson Space Center
KSA	K-band single access
L/L	launch and landing
LOS	loss of signal
LSIP	low-speed input processor
LVLH	local vertical/local horizontal
MA	multi-access
M50	Mean of 1950
MCC	Mission Control Center
MCU	master control unit
MED	manual entry device
MOC	mission operations computer
MPDM	measurement partial derivative module
MPT	mission plan table
MWS	mini workstation
NASA	National Aeronautics and Space Administration
NCIC	network communications interface common
NP	nutaton precession
NSC	next station contacts
OCM	observation computation module
OD	orbit determination
OMS	orbital maneuvering system
OPS	operations

PBI	pushbutton indicator
PEG	powered explicit guidance
PET	phase elapsed time
PFNI	powered-flight numerical integrator
PSAD	predicted satellite acquisition display
RATL	reentry attitude timeline
RCS	reaction control system
REFSMMAT	reference stable member matrix
RELMAT	relative matrix
RGDP	relay geometry display processor
RNP	rotation, nutation, and precession
ROTR	rotor attitude hold
RSD	residual summary display
RSP	residuals and statistics processor
RT	real time
RTS	remote tracking station
RVTL	reentry vent timeline
SAP	satellite acquisition processor
SAT	satellite acquisition table
SB	superbatch
SCDS	shuttle common display system
SCT	station characteristics table
SDP	shuttle data processor
SGLS	space ground link system
SI	solar inertial
SIC	support identification code
SIM	simulation
SMEK	simulated manual entry keyboard
SSA	S-band single access
STDN	space tracking and data network
STM	state transition matrix
STMM	state transition matrix module
SWCO	software checkout
TDR	trajectory data retrieval
TDRS	tracking and data relay satellite
TEI	true of epoch inertial (coordinate system)
TIG	time of ignition

USB	unified S-band
UTC	universal time corrected
UVW	U, V, W coordinate system
VAT	vector administration table
VBT	vehicle bias table
VBTM	vehicle bias table maintenance
VCP	vector compare processor
VCT	vehicle characteristics table
VDT	vehicle data table
VDTI	vehicle data table index
VDTM	vehicle data table maintenance
VIT	vent initialization table
VTAB	vent table
VTL	vent timeline
VTMC	vector table maintenance and control
VTUS	vent timeline user service
WGL	weight gain/loss
WTAB	weight table

APPENDIX D
STUDY QUESTIONS

NONCRITICAL QUESTIONS

1. List and give a brief description of the major force models used for the trajectory prediction process (see fig. 2-1).
2. What are internal and external ephemerides?
3. How many external ephemerides are there in the trajectory application? Briefly, describe how they are used.
4. Briefly describe the concept of ET-UTC and how it is used in the trajectory applications.
5. Briefly describe the major functions of the LSIP.
6. Provide a brief description of the solar flux and geomagnetic index values and how they are used within the trajectory application.
7. Briefly describe the function of the VDT and the number of batches that may be stored.
8. What does the MED decoder do?
9. List and give a brief description of the three types of integrators used in the trajectory application.
10. Differentiate between open and closed tracking data batches.
11. Briefly describe the major functions of the vector compare processor and how it is used to help the navigator assess state vector quality; also define and briefly describe the available stopping options.
12. Define and discuss the overall role of the residual. What displays and processors use this information?
13. Briefly describe the concept of weighted least squares processing.
14. List the major routines involved in OD processing.
15. What processor is used by the navigator to evaluate residuals.
16. What three routines compute shuttle tracking station acquisition times?

17. Define terrain masking and list the routines that use this information.
18. Briefly describe the concept that the delta-time processor is based upon.

CRITICAL QUESTIONS

1. What atmosphere models are used by the ground navigation software and what are their operational altitudes?
2. What is the purpose of the ENCKE effective forces routine?
3. What is the default maximum integration time allowed in days? Can this value be changed? If so, how?
4. What are the five major functions of the LSIP?
5. Describe the major functions of the numerical integration preprocessor and list the information it supplies to the numerical integrators.
6. Describe the major function of the a priori setup processor. How is the a priori setup processor invoked?
7. What information does the predicted satellite acquisition display (PSAD) processor need to generate the PSAT display?
8. What is the purpose of the checkout monitor display processor? Briefly describe the coordinate systems and frame of reference available for display.
9. If the "M" integrator option is used, will the vector compare logic check for the integration stop value if it falls during a maneuver?
10. Define and briefly describe the use of the RNP matrix table and briefly describe how it is used in the trajectory application. How long is the table? What happens if an RNP matrix request lies outside the table bounds?
11. Will a vehicle bias table entry be applied to an open batch when a delta-time processing or a batch residual plot request is made?
12. Will a bias be applied to a batch that initially had a bias applied to it if it is copied or split?
13. What processor assigns batch numbers? What is the maximum batch number that can be assigned? What happens if a batch needs to be assigned a number that exceeds the upper limit?
14. What are the functions of the covariance matrices stored in the following locations: BCOV, S1COV, S2COV, SS2, and SS3? What are the last four generally used for during real-time operations?

15. What is the BVENT/LVENT processor used for? What does it compute for display?
16. Describe the sequence of events and the major processors used for BB orbit determination.
17. Does the delta-time processor operate on open and closed batches? Based on a predefined threshold seek time, describe the sequence of events within the DTP logic.

COMPLEX QUESTIONS

1. Briefly describe the two types of integration step sizes used in the free-flight numerical integrator. Which is used the majority of the time? What does the beta series computation routine do?
2. How is the Sun/Moon ephemeris table initialized, how long is the table, and what is it used for?
3. What processors or modules compute residual statistics?
4. How does the observation computation module compute observations? Describe how the light time algorithm work.
5. What processors or modules compute residuals?
6. How does the OCM correct for refraction?
7. What is the purpose of the powered-flight numerical integrator? Give examples of how it is used in MCC operations. What atmosphere model does it use?
8. What routine is used for integration startup? Describe its major tasks.
9. What routine controls the numerical integration process?
10. Give the name and function of each part of the equation below.

$$\Delta x = (A^T W A + k^n \Gamma^{-1}) (A^T W \Delta y + k^n \Gamma^{-1} \Delta \bar{x})$$

What role does this play in the weighted least squares concept?

11. What is the purpose of the delta-time processor? For what type of error does it compute a correction?
12. Describe how the DTP computes a correction using range data. When during the pass does the range data compute the most correct delta-time value? Why? Answer the same question for Doppler.
13. How are vents occurring on or near an integration step handled by the free-flight numerical integrator?

APPENDIX E
ANSWERS TO STUDY QUESTIONS

ANSWERS TO NON-CRITICAL QUESTIONS

- p. 2-1 1. The major force models used for the trajectory prediction process are

Central force Model $\vec{r} = \frac{-\mu \vec{r}}{|\vec{r}|^3}$

Sun/Moon perturbations model

Gravitational accelerations caused by the Sun and Moon

Geopotential model

Best attempt to simulate gravitational accelerations caused by the irregular shape of the Earth

p. 2-3

Solar radiation pressure

Solar radiation pressure on the TDRS tracking satellite is modeled.

Atmospheric density model

Used by the drag model, the MCC uses the Jacchia/Lineberry model. It is a function of altitude, average solar flux, and the average geomagnetic index.

Vent model

A vehicle used to model known vents such as water dumps.

p. 2-66

2. External ephemerides make vectors available to the user at any time, and their characteristics can be seen on the Trajectory Profile Status display.

Internal ephemeris is generated by a request from a calling routine and is transparent to the user.

p. 2-66

3. There are four shuttle and three TDRS external ephemerides in the trajectory application. External ephemerides 1 through 4 are used by the flight dynamics officer (FDO) for trajectory prediction and maneuver planning.

Ground navigation uses external ephemerides 1 through 4 to assign vehicle characteristics to a specified link. The TDRS ephemerides are usually used in the DC process and are assigned by MED prior to processing.

p. 2-77

4. ET-UTC stands for ephemeris time minus universal time coordinated. It is a time correction value applied to the Sun/Moon ephemeris. The Sun/Moon ephemeris is in ET and all the trajectory software is in UTC, so the Sun/Moon

timetags are converted from ET to UTC by the application of the ET-UTC value. The ET-UTC value is also used to compute the GHA which is used in RNP matrix computations.

- p. 3-4 5. The LSIP receives tracking data and ensures the validity, integrity, and consistency of the data before allowing the orbit determination routines to act upon the data.
- p. 2-14 6. Predicted and historical solar flux geomagnetic index values are stored at 3-month intervals in the trajectory software. For flights, this information is replaced with actual values in the launch - 1 day inputs. Solar flux values are delayed by 1 day; for example, the solar flux value for 12-21-88 would be used for 12-22-88.
- p. 3-10 7. The VDT is the storage area for tracking data batches. It holds 100 batches.
- p. 3-1 8. The MED decoder. The MED decoder interprets flight controller MED inputs.
- p. 3-1 9. The three types of integrators used in the trajectory applications are
- p. 2-28 ECENCKE (free-flight) - Used for integration during a trajectory update and for any free flight interval.
- p. 2-53 ECPFNI (powered-flight) - Numerical integration of state vectors through modeled maneuvers.
- p. 2-67 ECPAEG (analytic ephemeris generator) - Propagates a vector with or without drag, using the Lineberry formulations.
- p. 3-7 10. Open batch - A buffer of valid low speed messages of like characteristics for which data is still being received.
- p. 3-7 Closed batch - Contains at least three valid measurement sets and is placed in the vehicle data table.
- p. 4-16 11. The VC contains the logic necessary for the comparison of vectors specified by MED and the display of the resultant parameters. The VCP may be used as an aid in the evaluation of vectors from several different sources. The vectors to be displayed or compared are collected and integrated to the appropriate cutoff conditions. The available stopping options are
- TIME - Phase elapsed time (PET) or GMT
 - FPA - Cutoff on a flightpath angle
 - ALT - Terminate on a specified altitude

- AN - Terminate on next upcoming ascending node
- LON - Causes integration to stop at a certain longitude
- RAD - Terminate integration upon a radial distance from Earth.

Section 3.0 12. The residual is used in the equation below as Δy (observed minus computed values). Residual statistics are used to evaluate the quality of the solution. The batch residual plot is used to evaluate solution and tracking data quality. Processors using residuals are

- RCP
- OCM
- DCM
- CP

Displays containing residual information are

- Batch Residual Plot
- BB Output
- SB Output
- Batch Residual Summary

Section 3.14 13. The weighted least squares concept.

- A Propagated vector and SS4 COV to anchor time of batch
- B Propagated vector across the time span of batch
- C Convert R, V components to X, Y, R, D format
- D Find differential corrections through multiple iterations of

$$\Delta x = \left(A^T W A + k^n \Gamma^{-1} \right) \left(A^T W \Delta y + k^n \Gamma^{-1} \Delta \bar{x} \right)$$

until convergence criteria are met. Use covariance downweighting to give tracking data more or less influence on solution.

Section 3.0 14. The major routines involved in OD processing are

- | | |
|---------------|-----------------------|
| MED decoder | DEP |
| PBI processor | DCM |
| LSIP | OCM |
| VBTM | MPDM |
| VDTM | Trajectory prediction |
| DCE | ESMISS |
| VTMC | ECENCKE |
| CSTP | ECPFNI |

STMM
RSP
CP

- Section 3.10 15. The residual and statistics processor is used by the navigator to evaluate residuals.
- p. 4-1 16. The three routines that compute shuttle tracking station acquisition times are
- a. Satellite acquisition processor
 - b. Stations contacts generator
 - c. Predicted site processor
- p. 4-6 17. Terrain masking is antenna obstructions information caused by the local terrain. It is used by the PSAD processor and the station contacts generator (EMGSTGEN).
- p. 4-8 18. The delta-time processor is used to minimize instantaneous downtrack position error. The downtrack error is corrected by the application of a delta-time (Δt) applied to the timetag of the shuttle state vector.

ANSWERS TO CRITICAL QUESTIONS

- p. 2-13 1. The average Jacchia/Lineberry and the U.S. standard 1962 model are used in the trajectory software. Their operational altitudes are as follows:

<u>Model</u>	<u>Altitude</u>
Average Jacchia/Lineberry	300,000 ft to 700 km
U. S. standard 1962	sea level to 700 km

The trajectory software uses the U. S. 1962 model from 300,000 ft to 67 n.mi. From 67 n.mi. to 700 km the average Jacchia/Lineberry model is used.

- p. 2-4 2. The ENCKE effective forces routine is used to calculate the total accelerations upon a vehicle in the gravitational field of three attracting bodies (Earth, Sun, Moon) and is also subject to other perturbing forces such as venting, drag, and geopotential.
- p. 2-36 3. The default value for the maximum integration time is 10 days. It can be changed via the FO2 MED in the L - 1 day inputs, if necessary.
- p. 3-5 4. The five major functions of the LSIP are to
- Monitor interface and data flow status
 - Format error detection
 - Measure set validation/status
 - State conversion
 - Data correction
- p. 2-17 5. The major function of the numerical integration preprocessor is to supervise the generation of specific input data for the free-flight or powered-flight numerical integrator. The preprocessor interfaces with the ATL, VTL, and weight and vehicle configurations user services.
- The EMPREPRO uses ATL, VTL, RATL, RVTL, CBDS, and WGL information.
- p. 3-19 6. The ASP is used to set up data in the DC common buffer which is required by the DCE. The ASP is invoked once at the beginning of each DC solution computation.
- p. 4-1 7. To generate the PSAT display, the PSAD processor needs the MED request, specified spacecraft ephemeris, PSAD table ID, request time, station characteristics, minimum elevation angle, and terrain masking.

- p. 4-13 8. The purpose of the checkout monitor display processor is to retrieve, integrate, and display a vector in M50 or TDR coordinates.
- p. 4-19 9. No, if the "M" integrator option is used, the vector compare logic will not check for the integration stop value if it falls during a maneuver.
- Section 2.8 10. The RNP matrix table has 40 RNP blocks. The initial block is 4 days before the anchor time of the ephemeris. The blocks are separated by 6 hours. RNP matrices are used for rotation coordinates. If an RNP matrix is not in the table, the RNP matrix load supervisor (ECLOAD) is called to generate a matrix within 3 hours of the vector to be rotated.
- p. 3-9 11. Yes, a vehicle bias table entry will be applied to an open batch when a delta-times processing or a batch residual plot request is made.
- p. 3-9 12. Yes, a bias will be applied to a batch that initially had a bias applied to it if it is copied or split.
- p. 3-10 13. The VDTM, assigns batch numbers. The maximum batch number that can be assigned is 999. If a batch needs a number and batch numbers 1 through 999 have already been assigned then the number assigned to the batch will be 001.
- p. 3-16 14. The functions of the covariance matrices stored in BCOV, S1COV, S2COV, SS2, and SS3 are as follows:
- a. BCOV - 3×3 submatrix containing vent accelerations covariance matrix to be used in the BB nine-element DC.
 - b. S1COV, S2COV - 3×3 submatrix containing vent accelerations covariance matrix to be used in SB DC, including vent elements and covariance elements for drag, two-way/three-way relay Doppler, and hybrid relay Doppler biases.
 - c. SS2, SS3 - Diagonal elements needed to build a 6×6 covariance matrix with respect to position and velocity elements.
- p. 4-11 15. The BVENT/LVENT processor is used to build displayable tables of
- a. The net vent forces in the orbiter body axis (BVENT processing)
 - b. The net VTL vent forces in the UVW coordinate system (LVENT processing)
- Section 3 16. See figure 3-2, BB flow.

17. Yes, it operates on both open and closed batches.

The sequence of events within the DTP logic are

- a. The data must be from the station specified on the initialization MED.
- b. The anchor time of the batch must be greater than or equal to the threshold time specified.
- c. If more than one batch meets the first two criteria, the one with the earliest anchor time is chosen.

When a closed batch is fetched by the DTP, it processes all the data frames in the batch (up to 50 will be displayed on the tables). If the DTP fetches an open batch, it processes all the saved data frames, then it waits for a queue from the LSIP to process new incoming radar data. If the batch ID cannot be found, the DTP remains enabled and waits for an LSIP queue to start processing data as soon as the LSIP receives the data and sends it to the DTP. The DTP is automatically inhibited when a batch closes.

The vector specified on the initialization MED is propagated by (ECENCKE) to a time just prior to the anchor time of the batch being processed. The OCM is then called to compute range and Doppler observations based on the ECENCKE-generated ephemeris.

The partials of downtrack error with respect to the range and Doppler observations from the data frame are computed.

Two estimates of the delta-T(Δt), based on range and Doppler observations, respectively, are computed for the data frame as follows:

Prior to termination, the DTP determines whether additional data frames are available for processing without recalling batch fetch software. If no additional data frames are available, the DTP returns control to the calling software.

ANSWERS TO COMPLEX QUESTIONS

- p. 2-38 1. The free-flight numerical integrator ECENCKE uses two types of integration step sizes: beta, a function of the eccentric anomaly, and time. The beta mode is used to integrate the majority of the time. The time mode is used when the desired step time falls in between beta step sizes. In this case the beta mode is used to get the integration near the cutoff time, then the software uses the time mode to get to the desired integration stop time.
- p. 2-16 2. The Sun/Moon ephemeris table is initialized by the P80 MED. The table is 400 days long. It is used to model perturbation to the trajectory caused by the gravitational effects of the Sun/Moon.
- p. 3-18 3. The RSP computes residual statistics.
- p. 3-27 4. To compute observations, the OCM converts the position and velocity values computed by ECENCKE over the time interval of the batch to angles, range, and Doppler measurements corresponding with the timetags of the actual tracking data values using station characteristics stored in the software. (See the tracking systems measurement models workbook for a discussion of the light time algorithm.)
- Sections 3.10 and 3.15 5. The OCM and the RSP compute residuals.
- p. 3-28 6. The OCM uses mean monthly refraction multiplier values to correct for atmospheric refraction errors, if necessary.
- Section 2.5 7. The powered-flight numerical integrator uses the U.S. 1962 standard atmosphere model.
- The purpose of the PFNI is to numerically integrate state vectors through modeled maneuvers.
- The flight dynamics officer uses the PFNI to determine accurate models trajectory perturbations caused by OMS maneuvers.
- Ground navigation also uses the PFNI while processing tracking data batches that contain maneuvers.
- p. 2-47 8. The Runge-Kutta starter routine (ECENEG) is used for integration startup. See steps 1 through 4 on pages 2-47 through 2-49.

p. 2-66 9. The numerical integration supervisor (EMSMISS) controls the numerical integration process.

p. 3-25 10. The following are the name and function of each part of the equation

$$\Delta x = \left(A^T W A + k^n \Gamma^{-1} \right) \left(A^T W \Delta y + k^n \Gamma^{-1} \Delta \bar{x} \right)$$

Δx = Best of differential correction to the a priori position and velocity state vector estimate

A = Matrix containing the partial derivatives of the data observations with respect to the Cartesian position and velocity components

W = Diagonal matrix of observation weights

Γ^{-1} = Covariance from the last batch to batch differential correction, propagated to the time of the current batch

k^n = Covariance multiplier

Δx = The previous (a priori) differential correction to the position and velocity state vector.

The multiplier k^n is applied to the a priori covariance to control the amount by which previous history constrains the current solution. The number of times the k value is applied, n is controlled by the flight controller during processing. The multiplier can be applied to the entire covariance matrix (an (XYZ) $k\Gamma$). The in-plane or out-of-plane elements of the covariance can be selectively downweighted through a transformation of the covariance from Cartesian to UVW (radial, along-track, and crosstrack) coordinates (a (UVW) $k\Gamma$). An in-plane (UV) $k\Gamma$ allows the current data to change the in-plane elements of the state vector while constraining out-of-plane changes. A (W) $k\Gamma$ affects only the out-of-plane covariance elements. A (UV) $k\Gamma$ followed by a (W) $k\Gamma$ is completely equivalent to an (XYZ) $k\Gamma$. TDRS BTB processing often employs in-plane covariance downweighting in an attempt to compensate for the weakness of Doppler-only TDRS data in orbital plane determination following trajectory perturbations.

p. 4-8 11. The DTP is used to compute a delta-time to be applied to a state vector timetag in order to minimize instantaneous downtrack position error.

12. The DTP MED specified vector is propagated by ECENCKE to just prior to the anchor time of the batch being processed. The OCM computes observations based on the ECENCKE-generated ephemeris.

The partials of downtrack error with respect to the range and Doppler observations from the Nth data frame are computed as follows:

$$\left(\frac{\partial DT}{\partial R}\right)_N = \frac{V_N(\Delta TP)}{R_{2N} - R_{1N}}$$

$$\left(\frac{\partial DT}{\partial D}\right)_N = \frac{V_N(\Delta TP)}{D_{2N} - R_{1N}}$$

When $\left(\frac{\partial DT}{\partial R}\right)_N$ is the partial of downtrack position with respect to the range observation for the Nth data frame

$\left(\frac{\partial DT}{\partial D}\right)$ = Downtrack position with respect to the Doppler observation for the Nth data frame

V_N = Spacecraft velocity of the time of the Nth data frame computed based on ephemeris one

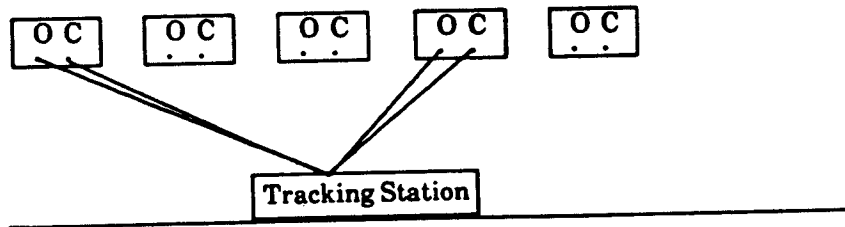
ΔTP = The timetag adjustment applied to the state vector used to create the internal perturbed ephemeris

$$\Delta t_{RN} = \left(\frac{\partial DT}{\partial R}\right)_N \left(\frac{D_{ON} - D_{1N}}{V_N}\right)$$

$$\Delta_{DN} = \left(\frac{\partial DT}{\partial R}\right)_N \left(\frac{D_{ON} - D_{1N}}{V_N}\right)$$

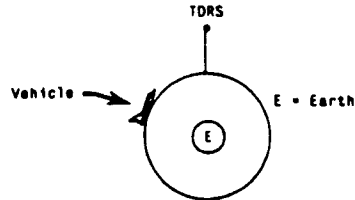
Downtrack Error Using Range Data

Using range data the computed delta-time value is most correct at the beginning and end of the batch. See the figure below:

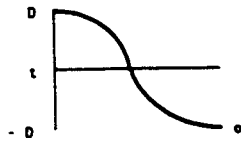


This is the case because the angle between the observed and the computed measurements is smallest at the beginning and end of the pass. The smaller the angle becomes, the closer the range residual is to the downtrack residual.

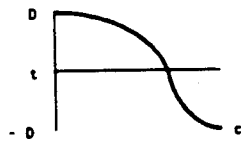
Downtrack Error using Doppler Data



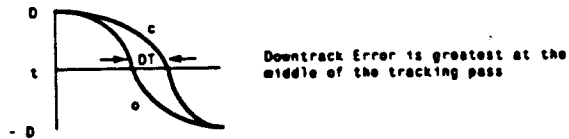
Actual Doppler measurement (a)



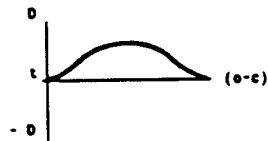
Computed Doppler measurement (c) with downtrack error



Overlay of Actual and Computed Measurements



Batch Residual Plot of Doppler (actual minus computed) with Downtrack Error



20757-001

p. 2-6

13. This is the case because vents occurring on or near an integration step are handled by the free-flight numerical integrator according to the logic on pages 2-6 through 2-11.