SAND--88-3158

DE89 009706

SAND88-3158
Unlimited Release
Printed March 1, 1989

User's Manual for the Trajectory Simulation and Analysis Program (TSAP)

David E. Outka
Aerospace Projects Division
Sandia National Laboratories
Albuquerque, NM 87185

Abstract

The Trajectory Simulation and Analysis Program (TSAP) provides a generalized package for studying point-mass trajectory problems. TSAP expands upon the input and output features of the code PMAST, from which it is derived. All important features, such as trajectory segmentation, iterative parameter searches and surveys, user defined output values, and target vehicle simulation, have been maintained and often improved. Differences include the ways in which vehicle orientation can be defined, the free format table input option, the coordinate systems for initial condition specification, and the capability to simulate vehicle launch from zero velocity. This report documents the mathematical basis of the trajectory simulation and explicitly defines the input and output variables. General input guidelines and examples are also presented.

---

* This work performed at Sandia National Laboratories supported by the U. S. Dept. of Energy Contract No. DE–AC04–76DP00789.

Acknowledgments

Several individuals have contributed to providing the TSAP user with both this document and the code itself, and the author expresses his appreciation for their support. First and foremost among these is D. E. Salguero (9144), who continues to support and update the code PMAST from which TSAP was derived. Guidance was also obtained from J. L. McDowell (9144), from whose code PMAST was originally derived. Notes from both of these individuals helped the author understand the workings of software they had written. R. W. Greene (1555), R. A. Lafarge (1551), and W. A. Millard (1555) were helpful in defining some of the terminology and notational conventions used in other trajectory codes found at Sandia and in the industry as a whole. S. A. Kerr (9144) has exercised early versions of the code heavily and provided many constructive comments and suggestions, as have other intrepid early users, including D. L. Keese (1555) and R. J. Weir (1551).

Contents

List of Figures

List of Tables

Nomenclature

notational conventions

$\hat{u}$	a caret ($\hat{}$) denotes a unit vector
$\vec{v}$	an arrow ($\vec{}$) denotes a vector of arbitrary magnitude
$\| \vec{v} \|$	denotes the magnitude of a vector $\vec{v}$
$\frac{^{\alpha}d\vec{v}}{dt}$	denotes the first time derivative of a vector $\vec{v}$ in a reference frame α
$^{\alpha}\vec{\omega}^{\beta}$	denotes the angular velocity of a reference frame β relative to a reference frame α
$(\dot{\ })$	first time derivative of ( )
$(\ddot{\ })$	second time derivative of ( )

symbols

a	freestream speed of sound
$^{I}\vec{a}$	acceleration of the vehicle in inertial space
$^{\oplus}\vec{a}$	acceleration of the vehicle relative to an earth fixed reference frame
$\frac{^{I}d(\ )}{dt}$	first time derivative of ( ) relative to an inertial reference frame
$\frac{^{\oplus}d(\ )}{dt}$	first time derivative of ( ) relative to an earth fixed reference frame
e	eccentricity of the reference ellipsoid
f	flatness parameter of the reference ellipsoid
$\vec{F}$	net force on the vehicle
m	mass of the vehicle
$\vec{r}$	position vector from the earth's center to the vehicle
$R_{\oplus}$	equatorial radius of the reference ellipsoid (earth)
R_p	polar radius of the reference ellipsoid
$^{\oplus}\vec{V}$	velocity of the vehicle relative to an earth fixed reference frame

${}^{I}\vec{V}$	velocity of the vehicle in inertial space
${}^{w}\vec{V}$	velocity of the vehicle relative to the surrounding air mass (identical to ${}^{\oplus}\vec{V}$ if no winds are present)
t	time
$\hat{x}_b, \hat{y}_b, \hat{z}_b$	orthogonal unit vectors aligned with the vehicle, $\hat{x}_b$ parallel to the longitudinal axis, $\hat{y}_b$ out the right wing, $\hat{z}_b$ in the vehicle "down" direction
$\hat{x}_\oplus, \hat{y}_\oplus, \hat{z}_\oplus$	orthogonal unit vectors aligned with the X, Y, and Z axes, repsectively, of the ECFC reference frame; if located at the earth's center, $\hat{x}_\oplus$ and $\hat{y}_\oplus$ are in the equatorial plane, $\hat{x}_\oplus$ toward the greenwich meridian, $\hat{y}_\oplus$ toward the 90° E meridian, and $\hat{z}_\oplus$ toward the north pole.
$\hat{x}_{gc}, \hat{y}_{gc}, \hat{z}_{gc}$	orthogonal unit vectors aligned with the local geocentric horizon axis system; $\hat{x}_{gc}$ and $\hat{y}_{gc}$ perpendicular to $\vec{r}$, $\hat{x}_{gc}$ directed locally north, $\hat{y}_{gc}$ directed locally east, $\hat{z}_{gc}$ directed opposite to $\vec{r}$ (the local geocentric down unit vector)
$\hat{x}_{gd}, \hat{y}_{gd}, \hat{z}_{gd}$	orthogonal unit vectors aligned with the local geodetic horizon axis system; $\hat{x}_{gd}$ and $\hat{y}_{gd}$ tangent to the reference ellipsoid, $\hat{x}_{gd}$ directed locally north, $\hat{y}_{gd}$ directed locally east ($\hat{y}_{gd} \equiv \hat{y}_{gc}$), $\hat{z}_{gd}$ directed down, perpendicular to the reference ellipsoid
$\hat{x}_I, \hat{y}_I, \hat{z}_I$	unit vectors of fixed orientation in inertial space, $\hat{z}_I$ parallel to $\hat{z}_\oplus$, $\hat{x}_\oplus$ and $\hat{y}_\oplus$ rotated by the angle $\Omega = \Omega_0 + {}^{I}\vec{\omega}^{\oplus}\,(t - t_{epoch})$ from $\hat{x}_I$ and $\hat{y}_I$, respectively
$\hat{x}_p, \hat{y}_p, \hat{z}_p$	unit vectors corresponding to the inertial platform axis system
$\hat{x}_w, \hat{z}_w, \hat{y}_w$	orthogonal unit vectors, $\hat{x}_w$ aligned with ${}^{w}\vec{V}$, $\hat{z}_w$ rotated about $\hat{y}_b$ from $\hat{z}_b$ by negative α
α	angle of attack
α_r	azimuth from radar station north to vehicle
α_T	total angle of attack
β	sideslip angle
β_E	Euler sideslip angle
γ	vertical flight path angle
δ	latitude
ε_r	elevation from radar station horizontal to vehicle
Θ	pitch angle

λ	longitude
μ	bank angle
ρ	atmospheric mass density
Φ	roll angle
ϕ_w	windward meridian
ψ	horizontal flight path angle (heading)
Ψ	yaw angle
${}^{I}\vec{\omega}^{\oplus}$	angular velocity of the ECFC reference frame relative to an inertial reference frame
Ω	rotation angle between ECIC and ECFC coordinate systems

superscripts and subscripts

0	initial value
b	vehicle (body)
E	Euler
f	final value
gc	geocentric
gd	geodetic
I	inertial
p	polar
T	total
$\mathcal{V}$	denotes a reference frame fixed to the vehicle mass center, ${}^{\oplus}\vec{V}$, and a perpendicular local horizontal vector
τ	target
$\oplus$	earth fixed, equatorial, or ECFC
∞	freestream

Preface

The development history of TSAP goes back to at least 1984 when J. L. McDowell created the code PMARV for the study of maneuvering reentry vehicle trajectories. In 1985 this grew into the code PMAST after D. E. Salguero made large contributions in code format, the input/output capabilities, and overall flexibility of the code.[1] The author's involvement began by introducing multi-parameter trajectory optimization to the code in 1986.[2] Some of the customary methods of defining rocket and exoatmospheric vehicle attitude were not represented in PMAST so in 1987 work began on revising the code toward this end. The equations of motion were reformulated to facilitate this process and to remove singularities. Other changes were made at user suggestion until the mathematical formulation of the code had significantly changed. At this point it was decided the name should be changed, and TSAP came into being. (The intended pronunciation of TSAP is "zäp," following the example of a variant spelling of the title of a Russian monarch.)

Externally, the changes made to TSAP relative to PMAST have been minimized, the deviations required (more or less) in order to achieve specific ends. For example the angle combinations with which vehicle attitude is specified via the FLY data blocks must conform to what can in TSAP be considered physically meaningful. The associated iterative guidance policies have also been affected somewhat. Other input file incompatibilities exist in the RADAR and TARGET data blocks, but otherwise many PMAST input files can be run on TSAP without modification. A few output variable names and definitions have changed as have some default parameter values; but in the main, the trajectories should match rather well.

Internally, the overall structure of TSAP is patterned after that of PMAST with only the details varying. Both programs begin by processing the input files, checking for obvious errors, and preparing for the trajectory computation. Path integration and output routines follow. Even parts of this manual follow closely that of the PMAST reference manual. What differs is the formulation of the equations within the subroutines.

One philosophy to which TSAP is meant to adhere is to minimize the surprises found in the resulting output. The techniques used to accomplish this include having the program check the input for obvious errors, using consistent input conventions and default parameter values, and by providing documentation sufficient for describing both the input and output data. It is hoped that this document will fulfil the latter goal.

1 TSAP Overview

The purpose of the Trajectory Simulation and Analysis Program (TSAP) is to provide a generalized package for studying point mass trajectory problems about the earth. By intent this includes straight forward methods with which to describe vehicle characteristics and define trajectory conditions, compute the resulting trajectory, and then deliver output in several usable forms. Although the point mass (three degree-of-freedom) equations of motion employed for the flight path calculations are incapable of modeling body dynamics, the flexible way in which body attitude can be defined along with the capability to define aerodynamic and propulsion models of arbitrary complexity enables TSAP to reliably simulate trajectories for any vehicle having predictable body dynamics. Many aircraft, rocket boosters, spacecraft, and re-entry vehicles are often in this category.

A given mission profile might require a vehicle to assume several configurations and guidance strategies. Assuming changes of this sort constitute discrete events, the trajectory can be subdivided into segments. The flight of a sounding rocket, for example, might consist of a launch segment, an ascent during thrust segment, a coasting segment, an experiment deployment segment, and so forth. As many as 99 such segments may be defined for a TSAP simulation. A large number of output quantities are available, including those describing the trajectory in relation to several standard coordinate systems, to a radar station, and to a target. From these the user may formulate still others. Several methods are available with which to systematically vary vehicle and guidance scheme parameters to explore vehicle performance boundaries, evaluate guidance schemes, and meet trajectory constraints. The purpose of this document is to describe TSAP in terms of usage and to define the input and output quantities in a mathematical sense.

TSAP is a derivative of the code PMAST and is specifically designed for compatibility with input files orginally for use with PMAST.[1] Conversely, since some symbol names have been changed and some input formats generalized, TSAP input files may or may not run on PMAST and some output quantities will vary. The initial impetus for creating TSAP came from the need to resolve problems in defining vehicle attitude. This led to some major changes in the internal mathematics; for instance, the equations of motion were written in a different axis system. However, the input and output techniques used by the two codes are largely equivalent, and situations for which this is not the case will be noted.

The experience with both PMAST and the early versions of TSAP has been that new versions of the software are required to accomodate changing needs and user requests. This document corresponds to TSAP version 89.001. The current intention is to update this manual concurrently with future code revisions, republishing as the changes increase in number and significance. Between publications, dated update pages will be distributed to users upon request.

1.1 Input Files

The primary input is made up of two files. The *table file,* described in Section 3, consists of data describing the vehicle aerodynamic and propulsive characteristics, while the *problem file,* described in Section 4, consists of information pertaining to the trajectory problem of interest. Splitting the input in this way allows an aerodynamic and propulsion data base for a particular vehicle to be generated and maintained in a single file, separate from mission specific data for the various flight scenarios which might be under consideration. It also helps break the input into more manageable portions, which may reduce the chance of using erroneous data.

Optionally, additional files may be used to plot data from external sources on standard TSAP plots. The use of such files is requested in the problem input file. Refer to Section 4.19 for more information on this optional feature.

1.2 Output Files

Several output formats are available, including various degrees of input echoing, tabular listings in block or column format, and plots. The standard printout file contains a copy of the problem input file, summary information of the table file contents, and optional information including further problem file echoing, trajectory computation statistics, and tabular trajectory data (Section 4.18). Tabular trajectory data can also be directed to separate files. Plots (Section 4.19) are produced using the DISSPLA graphics package and delivered in meta-file format for reproduction on a variety of output devices.[3]

1.3 Running TSAP

The source code for TSAP has been developed and maintained on a VAX/VMS machine. Included in the source code are a few machine-dependent lines which make the software compatible with either a VMS machine or a Cray operating under CTSS. (Primarily, such lines are involved with input/output operations and do not affect the trajectory calculations. The exception is that the floating point variables on the VAX are double precision to make them roughly equivalent to those on the Cray, where they are single precision.) The CTSS lines, in comment form in the source code, can be made to replace the corresponding VMS lines by using an automated procedure for creating a CTSS-compatible version of the source code. The procedure continues by shipping the result to the Cray for compilation, after which the executable is stored on the Integrated File System using the MASS utility. This permits one basic version of the source code, maintained on one machine, to be used to create executable code for two or more incompatible machines.

Procedures exist for running TSAP under both VMS and CTSS. At Sandia National

Laboratories in Albuquerque (SNLA) the interactive VMS procedure TSAPRUN is available to run TSAP interactively or in batch mode on a VAX or in batch mode on CTSS via PROD and the SENDCTSS facility. This utility is available on the Department 1550 AEROVAX as

```
SAV07::U1:[DEOUTKA.UTIL]TSAPRUN.COM
```

When copying this procedure to another site be sure to include the similarly named file with the .FOR extension. Both files will have to be modified so as to search the appropriate directory for executable files, and the .FOR file compiled so that the .COM file can execute the result. The VMS executable version of TSAP is available as

```
SAV07::U1:[DEOUTKA.TSAP]TSAP.EXE
```

To run TSAP interactively on CTSS, the executable image on IFS should be made a local file via a command such as

```
get tsap.x : /e00031650/tsap/ctss.exe
```

By making the local files *tables* and *problem* be the associated input files TSAP can be run with the command

```
tsap.x / 10.0 1.0
```

using time limit and priority (here 10.0 and 1.0) as appropriate. At a minimum the files *status* and *print* will result as output, with other files appearing as named in the problem file for tabular listings. When plots are requested the file *pltfile* becomes the plot meta-file. Note that none of the output files may exist prior to running TSAP on CTSS or an error condition will result since the existing files can not be overwritten.

TSAP begins execution by making copies of the primary input files, freeing them for other uses. The problem input file is copied both to a scratch file and to the standard printout file. Next the table input file is read, the data preprocessed and stored on a scratch file. If all table sets appear to be of proper format and syntax the data for the first problem (up to the first EXECUTE line) from the problem file is similarly checked. The flight path simulation then proceeds based on the verified problem data.

During the trajectory calculations output is minimized. Status messages are printed which mark the progression of the simulation through the trajectory segments. If the block style output was requested, this data is printed at the appropriate intervals to the standard printout file. Otherwise the trajectory data requested as output is saved until the entire flight path has been completed, after which the summary tables and plots are created.

Once all paths requested by the current problem file data have been computed, the next set of data (if any) from the problem file is processed. This procedure continues until the problem file data has been exhausted.

2 Mathematical Formulation

The general mathematical formulation is presented in this section, consisting of definitions and derivations. First, the reference frames and coordinate systems to which the input and output quantities are referenced will be defined. Formulation of the equations of motion is next followed by dynamic terms which can be derived from them. Finally, contributions to the net force vector acting on the vehicle will be considered.

The notation used includes vectors of arbitrary magnitude (denoted with an arrow, such as $\vec{r}$) and unit vectors (denoted with a caret, such as $\hat{x}_b$). A left superscript on a vector indicates the reference frame dependence which arises from vector differentiation in time; for example, ${}^{\oplus}\vec{V}$ is the velocity vector relative to an earth fixed frame. (The importance of specification of reference frame is made clear in Section 2.2.) A right superscript may be used to indicate vector applicability (for example, ${}^{\oplus}\vec{V}^{T}$ is the target velocity) but is frequently left off since the point mass derivations deal primarily with one vehicle at a time. Vector magnitude is formally denoted with double bars but is also denoted by omission of the arrow (hence, ${}^{\oplus}V = \|{}^{\oplus}\vec{V}\|$). Direct reference to standard output symbol names is made where appropriate.

Often the derivatives of scalars and unit vectors are denoted with a dot above the the symbol (such as $\dot{h}$). Similarly, the second derivatives may be denoted with a double dot (such as $\ddot{h}$). In the case of unit vectors the frame of reference relative to differentiation should be clear from the context. Dot notation is employed in Section 2.2.1, for example.

For notational convenience unit vectors are frequently paired as dyadics, such as $\hat{x}_{gd}\hat{y}_{gd}$. Operations on dyadics produce what might be considered the expected results; for example, $\hat{x}_b \cdot \hat{x}_{gd}\hat{y}_{gd} = (\hat{x}_b \cdot \hat{x}_{gd})\hat{y}_{gd}$ and $\hat{x}_{gd}\hat{y}_{gd} \cdot \hat{x}_b = (\hat{y}_{gd} \cdot \hat{x}_b)\hat{x}_{gd}$. Section 2.1.10 contains several example uses of dyadic notation.[5]

Some of the equations presented here have denominators which could potentially go to zero, creating numerical overflow. Protection from such errors is provided by condition testing. If one of the input quantities is involved, the interpretation of such an anomolous value is probably somewhat arbitrary anyway so TSAP will proceed using assumed results. The equations of motion are purposely written in non-singular form to avoid numerical problems so the computed trajectory should be reasonable. Output variables will be assigned reasonable values when conditions of undefined value occur. The two conditions which will cause computation to cease is a vehicle at the center of the earth (such that $r = 0$) and a vehicle of zero mass (such that $m = 0$).

2.1 Coordinate Systems

When deriving equations describing the motion of one object relative to others, it is necessary to define one or more reference frames in which the terms of the equations may

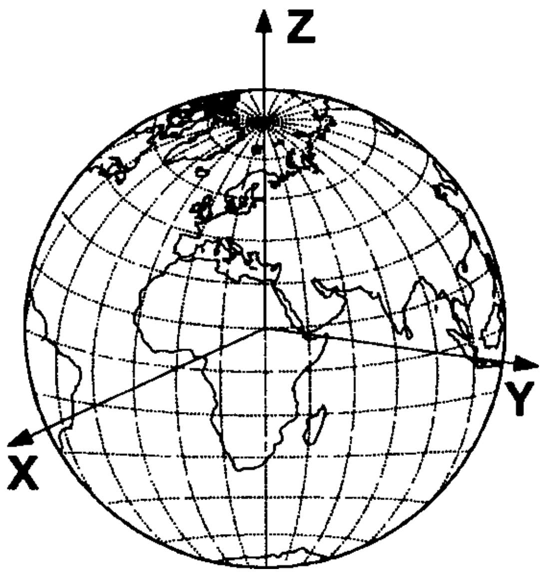

Figure 1: The ECFC Coordinate System Axes

be defined. When dynamics (vis-a-vis kinematics) are involved, one must also know the relationship of the reference frames to an inertial (or Newtonian) frame. This section will define several reference frames for use when deriving equations in later sections.

Associated with each reference frame may be one or more coordinate systems; methods in which to describe terms such as position and velocity. A position near the earth can be described in terms of x, y, and z components or as an altitude, longitude and latitude. Which method is chosen may be a matter of convention or of convenience. A general purpose code should have several such choices available. The coordinate systems used in TSAP are defined in this section.

2.1.1 Earth Centered and Fixed Cartesian (ECFC)

The ECFC coordinate system is used to express the integrated state vector (made up of the position and velocity vectors) as well as many other vectors. The equations of motion are written in ECFC terms. Hence, it is the primary coordinate system within TSAP. The origin is at the center of the earth, and the axes are fixed such that the $X_\oplus$ and $Y_\oplus$ axes point through the zero (Greenwich) and ninety degree meridians, respectively, in the equatorial plane. The $Z_\oplus$ axis, orthogonal to these, points out through the north pole and is assumed to be the axis of the earth's spin (see Figure 1.) The unit vectors $\hat{x}_\oplus$, $\hat{y}_\oplus$, and $\hat{z}_\oplus$ are parallel to these axes.

Position, velocity, and acceleration in ECFC terms are referenced to the component values of the vectors in each coordinate direction. The associated output variables are

XECF	$\vec{r}\cdot\hat{x}_\oplus$, that is, the component of the position vector (from the earth's center to the vehicle's mass center) in the ECFC X-axis direction
YECF	$\vec{r}\cdot\hat{y}_\oplus$
ZECF	$\vec{r}\cdot\hat{z}_\oplus$
XECFD	${}^\oplus\vec{V}\cdot\hat{x}_\oplus$, that is, the component of the earth relative velocity vector in the ECFC X-axis direction
YECFD	${}^\oplus\vec{V}\cdot\hat{y}_\oplus$
ZECFD	${}^\oplus\vec{V}\cdot\hat{z}_\oplus$
XECFDD	${}^\oplus\vec{a}\cdot\hat{x}_\oplus$, that is, the component of the earth relative acceleration vector in the ECFC X-axis direction
YECFDD	${}^\oplus\vec{a}\cdot\hat{y}_\oplus$
ZECFDD	${}^\oplus\vec{a}\cdot\hat{z}_\oplus$

2.1.2 Earth Centered, Inertially Fixed Cartesian (ECIC)

In TSAP, the center of the earth is assumed not to accelerate in an inertial (Newtonian) frame of reference. Hence, an inertial reference frame can be defined with origin at this point. The Z axis of the ECIC coordinate system is chosen as coincident with the earth's spin axis and $\hat{z}_\oplus$ and, hence, is fixed with respect to the earth. The ECFC X and Y axes are related to their ECIC counterparts by a dextral rotation about the Z axis of angle $\Omega = \Omega_0 + {}^I\omega^\oplus(t - t_{epoch})$, where ${}^I\omega^\oplus = {}^I\vec{\omega}^\oplus\cdot\hat{z}_\oplus$ (by assumption, $\|{}^I\vec{\omega}^\oplus\| = {}^I\vec{\omega}^\oplus\cdot\hat{z}_\oplus$). Output variables are available for the component values of the inertial position and velocity vectors in each ECIC coordinate direction as follows.

XECI	$\vec{r}\cdot\hat{x}_I = \vec{r}\cdot(\hat{x}_\oplus\cos\Omega - \hat{y}_\oplus\sin\Omega)$
YECI	$\vec{r}\cdot\hat{y}_I = \vec{r}\cdot(\hat{x}_\oplus\sin\Omega + \hat{y}_\oplus\cos\Omega)$
ZECI	$\vec{r}\cdot\hat{z}_I = \vec{r}\cdot\hat{z}_\oplus$
XECID	${}^I\vec{V}\cdot\hat{x}_I = {}^\oplus\vec{V}\cdot(\hat{x}_\oplus\cos\Omega - \hat{y}_\oplus\sin\Omega) - {}^I\omega^\oplus\vec{r}\cdot(\hat{x}_\oplus\sin\Omega + \hat{y}_\oplus\cos\Omega)$
YECID	${}^I\vec{V}\cdot\hat{y}_I = {}^\oplus\vec{V}\cdot(\hat{x}_\oplus\sin\Omega + \hat{y}_\oplus\cos\Omega) + {}^I\omega^\oplus\vec{r}\cdot(\hat{x}_\oplus\cos\Omega - \hat{y}_\oplus\sin\Omega)$
ZECID	${}^I\vec{V}\cdot\hat{z}_I = {}^\oplus\vec{V}\cdot\hat{z}_\oplus$
XECIDD	${}^I\vec{a}\cdot\hat{x}_I = \vec{F}\cdot(\hat{x}_\oplus\cos\Omega - \hat{y}_\oplus\sin\Omega)/m$
YECIDD	${}^I\vec{a}\cdot\hat{y}_I = \vec{F}\cdot(\hat{x}_\oplus\sin\Omega + \hat{y}_\oplus\cos\Omega)/m$
ZECIDD	${}^I\vec{a}\cdot\hat{z}_I = \vec{F}\cdot\hat{z}_\oplus/m$

2.1.3 Spherical Earth Centered and Fixed Geocentric (Geocentric)

The geocentric axis system is similar to the ECFC system except that position and velocity vectors are expressed in terms of their magnitudes and orientation angles. The position and velocity variables associated with this system are defined below. Both the symbols and output variable names are given. Potential problems with singularities in these definitions are avoided using simple condition testing and arbitrarily assigned alternate results.

r	RCM	position radius magnitude, $\| \vec{r} \|$
λ	LONG	longitude; the dextral rotation about the ECFC Z axis from the ECFC X axis to the projection of $\vec{r}$ onto the equatorial plane, i.e., the angle between $\hat{x}_\oplus$ and $\vec{r}\cdot(\hat{x}_\oplus\hat{x}_\oplus + \hat{y}_\oplus\hat{y}_\oplus)$; may be written $\tan\lambda = \vec{r}\cdot\hat{y}_\oplus/\vec{r}\cdot\hat{x}_\oplus$
δ_{gc}	LATGC	geocentric latitude; the angle between $\vec{r}$ and the equatorial plane, positive north; may be written $\sin\delta_{gc} = \vec{r}\cdot\hat{z}_\oplus/r$
${}^\oplus V$	VEL	earth relative velocity magnitude, $\| {}^\oplus\vec{V} \|$
γ_{gc}	GAMGC	geocentric vertical flight path angle; the angle between the velocity vector and the local geocentric horizontal plane, or 90° minus the angle between $\vec{r}$ and ${}^\oplus\vec{V}$; may be written $\sin\gamma_{gc} = -{}^\oplus\vec{V}\cdot\hat{z}_{gc}$
ψ_{gc}	PSIGC	geocentric horizontal flight path angle; the angle between the projection of the velocity vector onto the local geocentric horizontal plane and the local north vector $\hat{x}_{gc}$, positive east; may be written $\tan\psi_{gc} = {}^\oplus\vec{V}\cdot\hat{y}_{gc}/{}^\oplus\vec{V}\cdot\hat{x}_{gc}$

2.1.4 Spherical Earth Centered and Fixed Geodetic (Geodetic)

The geodetic axis system is similar to the geocentric system except that position and velocity vector orientations are related to the surface of the reference ellipsoid. This is the system in most common use. The ellipsoid (hypothetical earth shape) is an oblate spheroid with the major axes in the equatorial plane of the earth. One quadrant of an ellipsoid cross section containing the polar axis and $\vec{r}$ is shown in Figure 2. The shape of the ellipsoid is defined by two parameters. One is the equatorial earth radius $R_\oplus$. The other is one of three redundant terms R_p (the polar radius), e (the ellipsoid eccentricity), or f (the ellipsoid flattening factor). These parameters are related by

$$R_p = R_\oplus\sqrt{1-e^2} = R_\oplus(1-f) \tag{1}$$

Note that e must not exceed unity. It follows that R_p must not exceed $R_\oplus$ and that f must not exceed unity. In TSAP, the values describing earth shape normally conform to the WGS 84 model but may be reset as desired (see Section 4.6).[6,7]

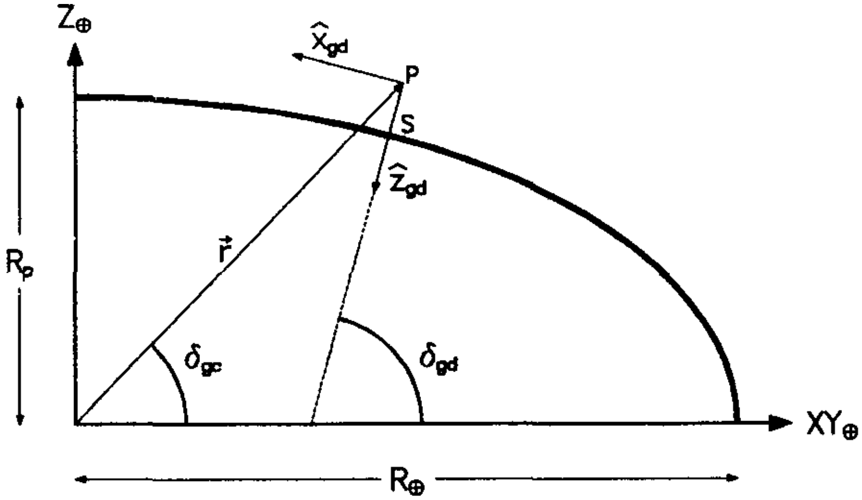

Figure 2: Surface of the Reference Ellipsoid

The position and velocity variables associated with the geodetic coordinate system are as follows.

h	ALT	altitude; the distance between the vehicle and the surface of the reference ellipsoid
λ	LONG	longitude (defined as for the geocentric system)
δ_{gd}	LATGD	geodetic latitude; the angle between $\hat{z}_{gd}$ and the equatorial plane, positive north, where $\hat{z}_{gd}$ is normal to the ellipsoid and directed toward the earth's equatorial plane from the vehicle mass center
${}^{\oplus}V$	VEL	$\lVert {}^{\oplus}\vec{V} \rVert$ (as for the geocentric system)
γ_{gd}	GAMGD	geodetic vertical flight path angle; the angle between the velocity vector and the local geodetic horizontal plane, or the angle between $\hat{z}_{gd}$ and ${}^{\oplus}\vec{V}$ minus 90°; may be written $\sin\gamma_{gd} = -{}^{\oplus}\vec{V}\cdot\hat{z}_{gd}$
ψ_{gd}	PSIGD	geodetic horizontal flight path angle; the angle between the projection of the velocity vector onto the local geodetic horizontal plane and the local north vector $\hat{x}_{gd}$, positive east; may be written $\tan\psi_{gd} = {}^{\oplus}\vec{V}\cdot\hat{y}_{gd}/{}^{\oplus}\vec{V}\cdot\hat{x}_{gd}$

The relationship between the ECFC and geodetic position terms is described by

$$\vec{r}\cdot\hat{x}_{\oplus} = \tilde{x}\cos\lambda \tag{2}$$

$$\vec{r}\cdot\hat{y}_{\oplus} = \tilde{x}\sin\lambda \tag{3}$$

$$\vec{r}\cdot\hat{z}_{\oplus} = \left[\frac{R_{\oplus}(1-e^2)}{\sqrt{1-e^2\sin^2\delta_{gd}}} + h\right]\sin\delta_{gd} \tag{4}$$

$$\tilde{x} = \left[\frac{R_\oplus}{\sqrt{1 - e^2 \sin^2 \delta_{gd}}} + h \right] \cos \delta_{gd} \tag{5}$$

Computing the ECFC components of $\vec{r}$ from the geodetic position terms is straight forward, but the converse is not true: an iterative solution is required to obtain h and δ_{gd} from ECFC components due to the transcendental nature of δ_{gd} in these equations. TSAP uses the iterative procedure described below which is similar to that described in reference 8.

The $Z_\oplus$ axis intercept of the line through P parallel to $\hat{z}_{gd}$ can be shown (see Equations 4 and 5) to be given by

$$z_i = -\frac{R_\oplus e^2 \sin \delta_{gd}}{\sqrt{1 - e^2 \sin^2 \delta_{gd}}} \tag{6}$$

If the earth is assumed to be nearly spherical (hence $e^2 \ll 1$) then $\sin \delta_{gd} \approx \sin \delta_{gc} = \vec{r} \cdot \hat{z}_\oplus / r$. (When $r \ll R_\oplus$ this is a poor but still adaquate assumption.) Therefore an approximation to z_i is

$$z_i = -R_\oplus e^2 \sin \delta_{gc} \tag{7}$$

The following sequence of equations is then used repeatedly to refine this estimate, terminating when the relative difference between successive estimates is less than 10^{-8}. Typical applications of the algorithm achieve accuracy in geodetic position within 0.01 ft in three or less iterations (in all cases no more than 25 iterations are permitted). Note that $\sin \delta_{gd}$ is given directly, and h can be determined from Equation 9.

$$z_d = \vec{r} \cdot \hat{z}_\oplus - z_i \tag{8}$$

$$N + h = \sqrt{\tilde{x}^2 + z_d^2} \tag{9}$$

$$\sin \delta_{gd} = \frac{z_d}{N + h} \tag{10}$$

$$N = \frac{R_\oplus}{\sqrt{1 - e^2 \sin^2 \delta_{gd}}} \tag{11}$$

$$z_i = -N e^2 \sin \delta_{gd} \tag{12}$$

2.1.5 Local Geocentric Horizon

This system is defined primarily for ease of computing the geocentric flight path and the vehicle attitude Euler angles. The origin is located at $\vec{r}$ relative to the earth's center (i.e., at the vehicle mass center). The X and Y axes are perpendicular to $\vec{r}$ and directed north and east, repectively, while the Z axis is directed opposite to $\vec{r}$. The unit vectors $\hat{x}_{gc}$, $\hat{y}_{gc}$, and $\hat{z}_{gc}$ are parallel to these axes and may be expressed as

$$\hat{x}_{gc} = -\sin \delta_{gc} (\cos \lambda \hat{x}_\oplus + \sin \lambda \hat{y}_\oplus) + \cos \delta_{gc} \hat{z}_\oplus \tag{13}$$

$$\hat{y}_{gc} = -\sin \lambda \hat{x}_\oplus + \cos \lambda \hat{y}_\oplus \tag{14}$$

$$\hat{z}_{gc} = -\cos \delta_{gc} (\cos \lambda \hat{x}_\oplus + \sin \lambda \hat{y}_\oplus) - \sin \delta_{gc} \hat{z}_\oplus \tag{15}$$

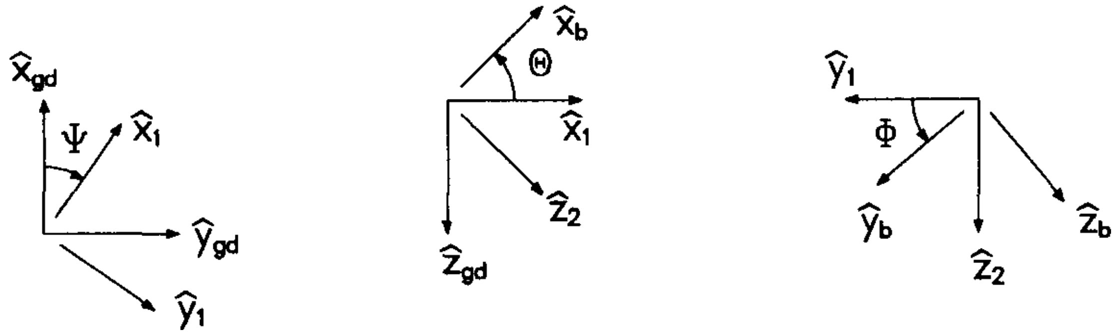

Figure 3: The Geodetic Yaw, Pitch and Roll Angles

2.1.6 Local Geodetic Horizon

The local geodetic horizon coordinate system is used for computing the geodetic flight path and the vehicle attitude Euler angles. The origin is located at the vehicle mass center. The Z axis is directed downward, normal to the ellipsoid. The X axis is perpendicular to the Z axis, pointed north and in the $\vec{r}\ \hat{z}_\oplus$ plane. The Y axis is perpendicular to these, parallel to the equatorial plane and directed east. The unit vectors $\hat{x}_{gd}$, $\hat{y}_{gd}$ and $\hat{z}_{gd}$ are parallel to these axes, as shown in Figure 2. Note that the origins of the local geocentric and geodetic coordinate systems are identical as are the $\hat{y}_{gc}$ and $\hat{y}_{gd}$ unit vectors. Moreover, $\hat{x}_{gc}$ and $\hat{z}_{gc}$ are rotated about $\hat{y}_{gc}$ by an angle $\delta_{gd} - \delta_{gc}$ from their geodetic counterparts. The local geodetic unit vectors may be expressed by replacing the *gc* subscripts in the equations in Section 2.1.5 for *gd* subscripts.

2.1.7 Inertial Platform Cartesian

The inertially fixed system for referencing YAWI, PITCHI, and ROLLI is known as the inertial platform cartesian coordinate system. Unless reset by the user (see Section 4.11), it is aligned with the local geodetic horizon system corresponding to the initial conditions. The unit vectors $\hat{x}_p$, $\hat{y}_p$, and $\hat{z}_p$ are parallel to the platform axes.

2.1.8 Body Fixed Cartesian

Aerodynamic and propulsive forces are generally defined in relation to the body fixed (vehicle) axis system, where the X axis is normally the longitudinal axis, the Y axis is (for an aircraft) toward the right wing, the Z axis is considered *down,* and the origin is at the mass center. The body coordinate system can be related to the local geocentric, local geodetic, and inertial platform coordinate systems through the (three sets of) vehicle attitude Euler angles yaw (Ψ), pitch (Θ), and roll (Φ). The successive rotation sequence required to go from the local geodetic system to the body system is shown in Figure 3. Note that the yaw and pitch angles are analogous to the horizontal and vertical flight path angles, respectively, where the longitudinal vehicle axis replaces the velocity vector. The unit vectors corresponding to the vehicle axis are $\hat{x}_b$, $\hat{y}_b$,and $\hat{z}_b$. They may be related to the local geodetic unit vectors by the expressions below (or to the local geocentric or inertial platfrom unit vectors through similar equations) .

$$\hat{x}_b = \cos\Theta_{gd}\cos\Psi_{gd}\hat{x}_{gd} + \cos\Theta_{gd}\sin\Psi_{gd}\hat{y}_{gd} - \sin\Theta_{gd}\hat{z}_{gd} \tag{16}$$

$$\begin{aligned}\hat{y}_b = & (\sin\Phi_{gd}\sin\Theta_{gd}\cos\Psi_{gd} - \cos\Phi_{gd}\sin\Psi_{gd})\hat{x}_{gd} \\ & (\sin\Phi_{gd}\sin\Theta_{gd}\sin\Psi_{gd} + \cos\Phi_{gd}\cos\Psi_{gd})\hat{y}_{gd} \\ & + \sin\Phi_{gd}\cos\Theta_{gd}\hat{z}_{gd}\end{aligned} \tag{17}$$

$$\begin{aligned}\hat{z}_b = & (\cos\Phi_{gd}\sin\Theta_{gd}\cos\Psi_{gd} + \sin\Phi_{gd}\sin\Psi_{gd})\hat{x}_{gd} \\ & (\cos\Phi_{gd}\sin\Theta_{gd}\sin\Psi_{gd} - \sin\Phi_{gd}\cos\Psi_{gd})\hat{y}_{gd} \\ & + \cos\Phi_{gd}\cos\Theta_{gd}\hat{z}_{gd}\end{aligned} \tag{18}$$

2.1.9 Velocity Vector Cartesian

Two intermediate coordinate systems are defined in the process of arriving at the wind axis system from each of the two local horizon systems. The X axis of both is aligned with the wind corrected velocity vector ${}^{w}\vec{V}$ (the vehicle velocity ${}^{\oplus}\vec{V}$ minus the earth relative wind velocity), and the origins are at the vehicle mass center. The Z axis is perpendicular to the X axis and, given this constraint, directed toward the earth's center in the X $\vec{r}$ plane. The Y axis is in the local (geocentric or geodetic) horizontal plane, completing the dextral system.

2.1.10 Wind Axis Cartesian

The wind axis system is used to relate the aerodynamic angles to the body fixed axis system. The X axis of the system, aligned with ${}^{w}\vec{V}$, is identical to that of the two velocity vector cartesian coordinate systems. The Y and Z axes are rotated by the bank angle from their velocity vector system counterparts (μ_{gc} for the geocentric velocity vector system and μ_{gd} for the geodetic). Figure 4 illustrates this rotation. The unit vectors $\hat{x}_w$, $\hat{y}_w$, and $\hat{z}_w$ are parallel to the corresponding wind axes.

The wind axes may be related to the body axes through the aerodynamic angles. Five aerodynamic angles are defined, used in pairs to describe the vehicle orientation in the wind system. For example, a negative (sinistral) rotation of the body axes about $\hat{y}_b$ by the angle of attack α (so that the resulting Z axis is that of the wind axis system), followed by

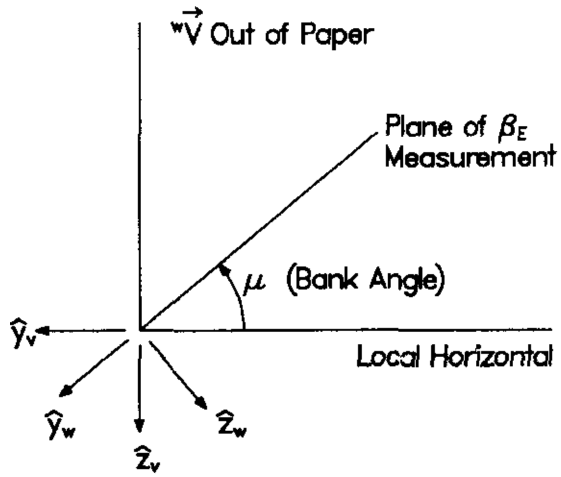

Figure 4: The Definition of Bank Angle

a positive (dextral) rotation about $\hat{z}_w$ by β_E, the Euler sideslip angle, achieves alignment with the wind axes. These angles are illustrated in Figure 5. Other angles which relate the body and wind coordinate systems are the sideslip (β), total angle of attack (α_T), and the windward meridian (ϕ_w) angles. β is defined as the angle between $\hat{x}_b$ and the projection of $^w\vec{V}$ onto the $\hat{x}_b$ $\hat{y}_b$ plane. This definition of sideslip is commonly used for axisymmetric vehicles such as reentry vehicles and artillery shells, while β_E is typically used with aircraft applications. The total angle of attack and the windward meridian are also used for axisymmetric vehicles. The complete list of aerodynamic angles are defined in terms of wind and body unit vectors below.

$$\sin\alpha = \frac{\hat{x}_w \cdot \hat{z}_b}{\| \hat{x}_w \cdot (\hat{x}_b\hat{x}_b + \hat{z}_b\hat{z}_b) \|} \tag{19}$$

$$\cos\alpha = \frac{\hat{x}_w \cdot \hat{x}_b}{\| \hat{x}_w \cdot (\hat{x}_b\hat{x}_b + \hat{z}_b\hat{z}_b) \|} \tag{20}$$

$$\sin\beta_E = \hat{x}_w \cdot \hat{y}_b \tag{21}$$

$$\cos\beta_E = \| \hat{x}_w \cdot (\hat{x}_b\hat{x}_b + \hat{z}_b\hat{z}_b) \| \tag{22}$$

$$\sin\beta = \frac{\hat{x}_w \cdot \hat{y}_b}{\| \hat{x}_w \cdot (\hat{x}_b\hat{x}_b + \hat{y}_b\hat{y}_b) \|} \tag{23}$$

$$\cos\beta = \frac{\hat{x}_w \cdot \hat{x}_b}{\| \hat{x}_w \cdot (\hat{x}_b\hat{x}_b + \hat{y}_b\hat{y}_b) \|} \tag{24}$$

$$\sin\alpha_T = \| \hat{x}_w \cdot (\hat{y}_b\hat{y}_b + \hat{z}_b\hat{z}_b) \| \tag{25}$$

$$\cos\alpha_T = \hat{x}_w \cdot \hat{x}_b \tag{26}$$

$$\sin\phi_w = \hat{x}_w \cdot \hat{y}_b \tag{27}$$

$$\cos\phi_w = \hat{x}_w \cdot \hat{z}_b \tag{28}$$

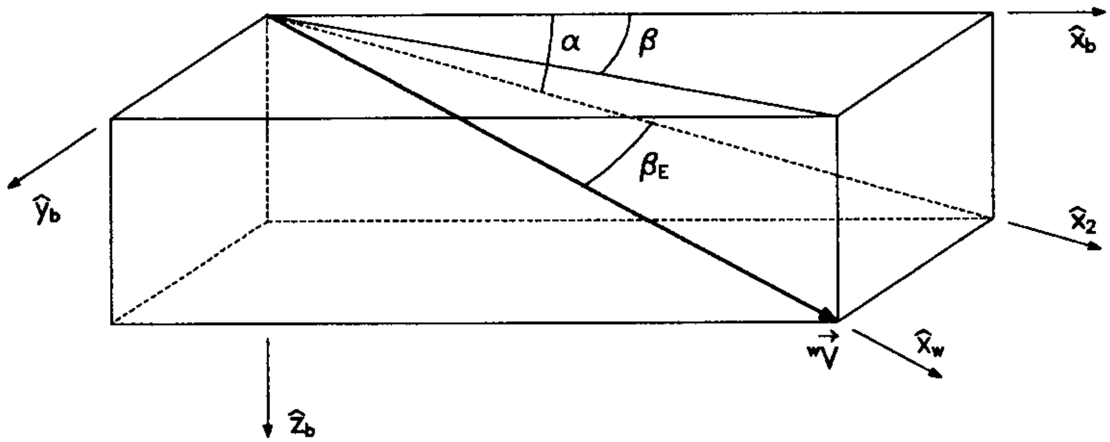

Figure 5: Relationship of the Body Axis to the Relative Wind Vector

When computing the Euler sideslip angle from known unit vectors, β_E is bounded by $\pm 90°$. This need not be the case when determining the unit vectors from known angles since any values of α and β_E constitute a physically realizable combination. The angle of attack and the sideslip angle β do not form such a robust combination. Specifically, $\cos\alpha$ and $\cos\beta$ must have the same sign to be physically meaningful. The two sideslip angles are related by the expressions

$$\sin\beta_E = \frac{\sin\beta\cos\alpha}{\sqrt{\cos^2\beta + \sin^2\beta\cos^2\alpha}}\frac{\cos\alpha}{|\cos\alpha|} \tag{29}$$

$$\cos\beta_E = \frac{|\cos\beta|}{\sqrt{\cos^2\beta + \sin^2\beta\cos^2\alpha}} \tag{30}$$

2.1.11 Vector Transformations Between Axis Systems

Vector transformations between axis systems has traditionally been performed through the use of transformation matrices. The explicit use of transformation matrices has been avoided in TSAP in favor of unit vector operations. The equivalence of both methods is straight forward, as will be demonstrated.

The ECFC axis system is used as a basis for expressing the unit vectors corresponding to the various axis systems. This common basis is also used for expressing most other vectors and for performing vector operations. Transforming vectors from the ECFC basis to other coordinate systems is through three dot product operations with the corresponding unit vectors. For example, the specific loading vector ${}^I\vec{a}_s$ (which may be written as ${}^{\oplus}({}^I\vec{a}_s)$ to explicitly denote its expression in the ECFC basis) may be transformed to the body fixed system as follows.

$$ {}^{b}({}^{I}\vec{a}_s) = \begin{bmatrix} \hat{x}_b \cdot {}^{I}\vec{a}_s \\ \hat{y}_b \cdot {}^{I}\vec{a}_s \\ \hat{z}_b \cdot {}^{I}\vec{a}_s \end{bmatrix} \quad (31) $$

$$ = \begin{bmatrix} \hat{x}_b{}^T \\ \hat{y}_b{}^T \\ \hat{z}_b{}^T \end{bmatrix} {}^{I}\vec{a}_s \quad (32) $$

$$ = {}^{b}A^{\oplus}\, {}^{I}\vec{a}_s \quad (33) $$

Note that the transformation matrix ${}^{b}A^{\oplus}$ consists of the unit vectors $\hat{x}_b$, $\hat{y}_b$, and $\hat{z}_b$, or conversely, that the unit vectors comprise a decomposition of the transformation matrix. Moreover, vector transformations from the body frame to the ECFC basis can also be considered as either a matrix operation or as vector dot products. For example, thrust ${}^{b}\vec{T}$ (expressed in the body frame) may be transformed to the ECFC frame as follows.

$$ {}^{\oplus}\vec{T} = {}^{\oplus}A^{b}\, {}^{b}\vec{T} \quad (34) $$

$$ = ({}^{b}A^{\oplus})^{-1}\, {}^{b}\vec{T} \quad (35) $$

$$ = \begin{bmatrix} \hat{x}_b{}^T \\ \hat{y}_b{}^T \\ \hat{z}_b{}^T \end{bmatrix}^T {}^{b}\vec{T} \quad (36) $$

$$ = \begin{bmatrix} \hat{x}_b & \hat{y}_b & \hat{z}_b \end{bmatrix} {}^{b}\vec{T} \quad (37) $$

Transformation Equations 31 and 37 are representative of those found in TSAP. Although these expressions seem more cumbersome than their matrix equivalents when written as above, they are easily coded into FORTRAN and permit reference to the unit vectors for other types of operations.

Finally, it is useful to note that in an ordered triple of unit vectors of the sort corresponding to an axis system (such as $\hat{x}_{\oplus}$, $\hat{y}_{\oplus}$, and $\hat{z}_{\oplus}$) any one of the vectors may be determined from the other two. For example, if $\hat{y}_{\oplus}$ and $\hat{z}_{\oplus}$ are known, $\hat{x}_{\oplus}$ is given as $\hat{x}_{\oplus} = \hat{y}_{\oplus} \times \hat{z}_{\oplus}$.

2.2 Equations of Motion and Associated Dynamic Terms

Assume that at time t_0 the initial position $\vec{r}$ and velocity ${}^{\oplus}\vec{V}$ are known. It is desired to propagate these vectors in time. Begin with the Newton relationship

$$\vec{F} = m\,{}^{I}\vec{a} \tag{38}$$

An expression for ${}^{I}\vec{a}$ is found by differentiating $\vec{r}$ twice, that is

$$\begin{aligned}
{}^{I}\vec{a} &= \frac{{}^{I}d^2\vec{r}}{dt^2} & (39)\\
&= \frac{{}^{I}d}{dt}\left(\frac{{}^{\oplus}d\vec{r}}{dt} + {}^{I}\vec{\omega}^{\oplus} \times \vec{r}\right) & (40)\\
&= \left[\frac{{}^{\oplus}d^2\vec{r}}{dt^2} + {}^{I}\vec{\omega}^{\oplus} \times \frac{{}^{\oplus}d\vec{r}}{dt}\right] + \left[\frac{{}^{\oplus}d}{dt}\left({}^{I}\vec{\omega}^{\oplus} \times \vec{r}\right) + {}^{I}\vec{\omega}^{\oplus} \times \left({}^{I}\vec{\omega}^{\oplus} \times \vec{r}\right)\right] & (41)\\
&= \left[{}^{\oplus}\vec{a} + {}^{I}\vec{\omega}^{\oplus} \times {}^{\oplus}\vec{V}\right] + \left[\frac{{}^{\oplus}d\,{}^{I}\vec{\omega}^{\oplus}}{dt} \times \vec{r} + {}^{I}\vec{\omega}^{\oplus} \times \frac{{}^{\oplus}d\vec{r}}{dt} + {}^{I}\vec{\omega}^{\oplus} \times \left({}^{I}\vec{\omega}^{\oplus} \times \vec{r}\right)\right] & (42)
\end{aligned}$$

Assuming that $\vec{F}$ and m are available and that ${}^{I}\vec{\omega}^{\oplus}$ is constant, Equations 38 and 42 may be combined to yield acceleration in an earth fixed frame of reference

$${}^{\oplus}\vec{a} = \frac{\vec{F}}{m} - 2\,{}^{I}\vec{\omega}^{\oplus} \times {}^{\oplus}\vec{V} - {}^{I}\vec{\omega}^{\oplus} \times \left({}^{I}\vec{\omega}^{\oplus} \times \vec{r}\right) \tag{43}$$

Equation 43 represents a second order differential equation of motion which, for purposes of numerical integration, may be rewritten as a system of first order equations. A state $\vec{S}$ to be propagated can be defined as

$$\vec{S}(t) = \begin{bmatrix} \vec{r}(t) \\ {}^{\oplus}\vec{V}(t) \end{bmatrix} \tag{44}$$

Then the desired equations of motion can be written in integral form as

$$\begin{aligned}
\vec{S}(t_f) &= \int_{t_0}^{t_f} \frac{{}^{\oplus}d\vec{S}(t)}{dt}\,dt + \vec{S}(t_0) & (45)\\
&= \int_{t_0}^{t_f} \begin{bmatrix} {}^{\oplus}\vec{V}(t) \\ {}^{\oplus}\vec{a}(t) \end{bmatrix} dt + \vec{S}(t_0) & (46)
\end{aligned}$$

Expressed in ECFC components, $\vec{S}$ is propagated in time (TIME) to describe the vehicle position and velocity. Other components of the integrated state vector are the pathlength, range, target state, and integral user defined variables. Pathlength (PATHLENG) is simply the time integral of the velocity magnitude ${}^{\oplus}V$. Range is the length of the projection of the flight path onto the reference ellipsoid (explicitly defined in Section 2.2.2). A target may be defined and its state propagated using Equation 46 and a simplified force or acceleration computation (see Section 4.27). User defined variables are discussed in Section 4.4.

TSAP computes a number of the terms used in spherical forms of the equations of motion for use as output variables. Singularities encountered in these equations are avoided through condition checking. Listed below are derivations for some of the geodetic terms. The geocentric term derivations are either similar or somewhat simpler so will not be included.

2.2.1 Longitude and Latitude Rates

The rates at which the longitude and geodetic latitude are changing is useful in describing the motion of an object relative to the earth's surface. The longitude rate $\dot{\lambda}$ (LONGD) arises from the definition of longitude as follows.

$$\lambda = \tan^{-1}\left(\frac{\vec{r}\cdot\hat{y}_\oplus}{\vec{r}\cdot\hat{x}_\oplus}\right) \tag{47}$$

hence,

$$\dot{\lambda} = \frac{{}^\oplus\vec{V}\cdot\hat{y}_\oplus\vec{r}\cdot\hat{x}_\oplus - {}^\oplus\vec{V}\cdot\hat{x}_\oplus\vec{r}\cdot\hat{y}_\oplus}{(\vec{r}\cdot\hat{y}_\oplus)^2+(\vec{r}\cdot\hat{x}_\oplus)^2} \tag{48}$$

$$= \frac{{}^\oplus\vec{V}\cdot(-\sin\lambda\hat{x}_\oplus+\cos\lambda\hat{y}_\oplus)}{\sqrt{(\vec{r}\cdot\hat{y}_\oplus)^2+(\vec{r}\cdot\hat{x}_\oplus)^2}} \tag{49}$$

The geocentric latitude rate (LATGCD) may be expressed as

$$\dot{\delta}_{gc} = \frac{{}^\oplus\vec{V}\cdot\hat{x}_{gc}}{r} \tag{50}$$

One derivation for the geodetic latitude rate $\dot{\delta}_{gd}$ (LATGDD) begins by multiplying Equation 5 by $(1-e^2)\tan\delta_{gd}$ and subtracting Equation 4 to get

$$\tilde{x}(1-e^2)\tan\delta_{gd} - z = -he^2\sin\delta_{gd} \tag{51}$$

where $z=\vec{r}\cdot\hat{z}_\oplus$. Differentiating with respect to time yields

$$\dot{\tilde{x}}(1-e^2)\tan\delta_{gd} + \dot{\delta}_{gd}\tilde{x}(1-e^2)\sec^2\delta_{gd} - \dot{z} = -\dot{h}e^2\sin\delta_{gd} - \dot{\delta}_{gd}he^2\cos\delta_{gd} \tag{52}$$

from which the expression for $\dot{\delta}_{gd}$ is found as

$$\dot{\delta}_{gd} = \frac{\left\{\dot{z}\cos\delta_{gd} - \left[\dot{\tilde{x}}(1-e^2)+\dot{h}e^2\cos\delta_{gd}\right]\sin\delta_{gd}\right\}\cos\delta_{gd}}{\tilde{x}(1-e^2)+he^2\cos^3\delta_{gd}} \tag{53}$$

where $\dot{\tilde{x}} = {}^\oplus\vec{V}\cdot(\cos\lambda\hat{x}_\oplus+\sin\lambda\hat{y}_\oplus)$, $\dot{z} = {}^\oplus\vec{V}\cdot\hat{z}_\oplus$, and $\dot{h} = -{}^\oplus\vec{V}\cdot\hat{z}_{gd}$ (ALTD). The singularity which occurs for $\tilde{x}=\cos\delta_{gd}=0$ (or when the altitude h is negative with an appropriate magnitude) is avoided by setting $\dot{\delta}_{gd} = {}^\oplus\vec{V}\cdot\hat{x}_{gc}/r$ as the numerical error condition is approached.

A related term is the rate at which $\hat{z}_{gd}$ changes orientation. Expressed as shown below, $\dot{\hat{z}}_{gd}$ is used to compute vertical acceleration $\ddot{h}$ (ALTDD, also shown) and ground speed (Section 2.2.2).

$$\begin{aligned} \dot{\hat{z}}_{gd} &= \frac{^{\oplus}d}{dt}\{-\cos\delta_{gd}(\cos\lambda\hat{x}_{\oplus} + \sin\lambda\hat{y}_{\oplus}) - \sin\delta_{gd}\hat{z}_{\oplus}\} && (54) \\ &= [\sin\delta_{gd}(\cos\lambda\hat{x}_{\oplus} + \sin\lambda\hat{y}_{\oplus}) - \cos\delta_{gd}\hat{z}_{\oplus}]\,\dot{\delta}_{gd} + \cos\delta_{gd}(\sin\lambda\hat{x}_{\oplus} + \cos\lambda\hat{y}_{\oplus})\dot{\lambda} && (55) \end{aligned}$$

$$\ddot{h} = -^{\oplus}\vec{a}\cdot\hat{z}_{gd} - {}^{\oplus}\vec{V}\cdot\dot{\hat{z}}_{gd} \tag{56}$$

2.2.2 Ground Speed and Range

As mentioned previously, range (RANGE) is defined as the length of the projection of the path of motion onto the surface of the earth's reference ellipsoid. Just as pathlength is the integral of the earth referenced velocity magnitude, range can be similarly described by ground speed, the velocity magnitude of the point on the reference ellipsoid nearest the vehicle. Note that RANGE is the length of the trace of the vehicle path over the ellipsoid surface, where as DWNRNG (Section 4.5) is the surface distance beginning at a reference point in the direction of a specified azimuth. An expression for ground speed (VGROUND) can be derived as follows.

Referring to Figure 2, let P be the center of mass of the vehicle and S be the point on the ellipsoid over which the vehicle is moving. (A line parallel to $\hat{z}_{gd}$ which passes through P will also intersect S, the point on the ellipsoid nearest the vehicle.) An expression for the ground velocity ${}^{\oplus}\vec{V}^{S}$ may be written as

$${}^{\oplus}\vec{V}^{S} = {}^{\oplus}\vec{V}^{P} + {}^{P}\vec{V}^{S} \tag{57}$$

where the vehicle velocity ${}^{\oplus}\vec{V}$ has been written as ${}^{\oplus}\vec{V}^{P}$ for clarity and ${}^{P}\vec{V}^{S}$ is the velocity of S with respect to P. The latter term may be expressed as

$$\begin{aligned} {}^{P}\vec{V}^{S} &= \frac{^{\oplus}d(h\hat{z}_{gd})}{dt} && (58) \\ &= \dot{h}\hat{z}_{gd} + h\dot{\hat{z}}_{gd} && (59) \end{aligned}$$

where $\dot{\hat{z}}_{gd}$ is given in Equation 55. Since ${}^{\oplus}\vec{V}^{P} = {}^{\oplus}\vec{V}^{P}\cdot(\hat{x}_{gd}\hat{x}_{gd} + \hat{y}_{gd}\hat{y}_{gd} + \hat{z}_{gd}\hat{z}_{gd})$ and $\dot{h} = -{}^{\oplus}\vec{V}^{P}\cdot\hat{z}_{gd}$ (ALTD), Equation 57 may be rewritten as

$$\begin{aligned} {}^{\oplus}\vec{V}^{S} &= {}^{\oplus}\vec{V}^{P}\cdot(\hat{x}_{gd}\hat{x}_{gd} + \hat{y}_{gd}\hat{y}_{gd}) + h\dot{\hat{z}}_{gd} && (60) \\ &= {}^{\oplus}V^{P}\cos\gamma_{gd}(\cos\psi_{gd}\hat{x}_{gd} + \sin\psi_{gd}\hat{y}_{gd}) + h\dot{\hat{z}}_{gd} && (61) \end{aligned}$$

Ground speed is the magnitude of ${}^{\oplus}\vec{V}^{S}$ and range the integral of ground speed in time.

2.2.3 Flight Path Angle Rates

For the derivation of the flight path angle rates (GAMGDD and PSIGDD), three orthogonal unit vectors are introduced. (The geodetic terms are derived here, but the geocentric equations for GAMGCD and PSIGCD are analogous.)

$$\hat{x}_v = {}^{\oplus}\vec{V}/{}^{\oplus}V \tag{62}$$
$$= \cos\gamma_{gd}\cos\psi_{gd}\hat{x}_{gd} + \cos\gamma_{gd}\sin\psi_{gd}\hat{y}_{gd} - \sin\gamma_{gd}\hat{z}_{gd} \tag{63}$$

$$\hat{y}_v = \hat{z}_{gd}\times\hat{x}_v / \parallel \hat{z}_{gd}\times\hat{x}_v \parallel \tag{64}$$
$$= -\sin\psi_{gd}\hat{x}_{gd} + \cos\psi_{gd}\hat{y}_{gd} \tag{65}$$

$$\hat{z}_v = \hat{x}_v\times\hat{y}_v \tag{66}$$
$$= \sin\gamma_{gd}\cos\psi_{gd}\hat{x}_{gd} + \sin\gamma_{gd}\sin\psi_{gd}\hat{y}_{gd} + \cos\gamma_{gd}\hat{z}_{gd} \tag{67}$$

Assume these unit vectors are fixed in a reference frame $\mathcal{V}$ which is fixed with respect to the vehicle mass center and the velocity vector while maintaining $\hat{y}_v$ in the horizontal plane. The angular velocity of $\mathcal{V}$ may be expressed as a sum of components.

$${}^{\oplus}\vec{\omega}^{\mathcal{V}} = {}^{\oplus}\vec{\omega}^{gd} + {}^{gd}\vec{\omega}^{\mathcal{V}} \tag{68}$$

where

$${}^{\oplus}\vec{\omega}^{gd} = \dot{\lambda}\hat{z}_{\oplus} - \dot{\delta}_{gd}\hat{y}_{gd} \tag{69}$$
$${}^{gd}\vec{\omega}^{\mathcal{V}} = \dot{\psi}_{gd}\hat{z}_{gd} + \dot{\gamma}_{gd}\hat{y}_v \tag{70}$$

The earth fixed acceleration may be written as

$${}^{\oplus}\vec{a} = \frac{{}^{\mathcal{V}}d{}^{\oplus}\vec{V}}{dt} + {}^{\oplus}\vec{\omega}^{\mathcal{V}}\times{}^{\oplus}\vec{V} \tag{71}$$

where

$$\frac{{}^{\mathcal{V}}d{}^{\oplus}\vec{V}}{dt} = \frac{d\parallel {}^{\oplus}\vec{V}\parallel}{dt}\hat{x}_v \tag{72}$$
$$= \frac{{}^{\oplus}\vec{a}\cdot{}^{\oplus}\vec{V}}{{}^{\oplus}V}\hat{x}_v \tag{73}$$
$$= {}^{\oplus}\vec{a}\cdot\hat{x}_v\hat{x}_v \tag{74}$$

and

$${}^{\oplus}\vec{\omega}^{\mathcal{V}}\times{}^{\oplus}\vec{V} = {}^{\oplus}V({}^{\oplus}\vec{\omega}^{\mathcal{V}}\cdot\hat{z}_v\hat{y}_v - {}^{\oplus}\vec{\omega}^{\mathcal{V}}\cdot\hat{y}_v\hat{z}_v) \tag{75}$$

hence,

$${}^{\oplus}\vec{a} = {}^{\oplus}\vec{a}\cdot\hat{x}_v\hat{x}_v + {}^{\oplus}V{}^{\oplus}\vec{\omega}^{\mathcal{V}}\cdot\hat{z}_v\hat{y}_v - {}^{\oplus}V{}^{\oplus}\vec{\omega}^{V}\cdot\hat{y}_v\hat{z}_v \tag{76}$$

and

$${}^{\oplus}\vec{\omega}^{\mathcal{V}}\cdot\hat{y}_v = -\frac{{}^{\oplus}\vec{a}}{{}^{\oplus}V}\cdot\hat{z}_v \tag{77}$$
$${}^{\oplus}\vec{\omega}^{\mathcal{V}}\cdot\hat{z}_v = \frac{{}^{\oplus}\vec{a}}{{}^{\oplus}V}\cdot\hat{y}_v \tag{78}$$

From Equations 68 through 70

$$
\begin{aligned}
{}^{\oplus}\vec{\omega}^{\mathcal{V}} \cdot \hat{y}_v &= (\dot{\lambda}\hat{z}_{\oplus} - \dot{\delta}_{gd}\hat{y}_{gd}) \cdot \hat{y}_v + \dot{\gamma}_{gd} && (79)\\
&= \dot{\lambda}\hat{y}_v \cdot \hat{z}_{\oplus} - \dot{\delta}_{gd}\cos\psi_{gd} + \dot{\gamma}_{gd} && (80)
\end{aligned}
$$

and

$$
\begin{aligned}
{}^{\oplus}\vec{\omega}^{\mathcal{V}} \cdot \hat{z}_v &= (\dot{\lambda}\hat{z}_{\oplus} - \dot{\delta}_{gd}\hat{y}_{gd} - \dot{\psi}_{gd}\hat{z}_{gd}) \cdot \hat{z}_v && (81)\\
&= \dot{\lambda}\hat{z}_v \cdot \hat{z}_{\oplus} - \dot{\delta}_{gd}\sin\gamma_{gd}\sin\psi_{gd} + \dot{\psi}_{gd}\cos\gamma_{gd} && (82)
\end{aligned}
$$

So the flight path angle rates may be expressed as

$$
\begin{aligned}
\dot{\gamma}_{gd} &= -\frac{{}^{\oplus}\vec{a}}{{}^{\oplus}V} \cdot \hat{z}_v - \dot{\lambda}\hat{y}_v \cdot \hat{z}_{\oplus} + \dot{\delta}_{gd}\cos\psi_{gd} && (83)\\
\dot{\psi}_{gd} &= \sec\gamma_{gd}\left[\frac{{}^{\oplus}\vec{a}}{{}^{\oplus}V} \cdot \hat{y}_v - \dot{\lambda}\hat{z}_v \cdot \hat{z}_{\oplus} + \dot{\delta}_{gd}\sin\gamma_{gd}\sin\psi_{gd}\right] && (84)
\end{aligned}
$$

2.2.4 Specific Loading

Specific Loading may be defined as the inertial acceleration of the vehicle less that due to gravity, and is sometimes useful for defining structural requirements. If $\vec{g}_I$ is the component of acceleration due to gravity, the specific loading vector is simply $\vec{a}_{sp} = {}^{I}\vec{a} - \vec{g}_I$. Equivalently, the specific loading vector is the sum of the aerodynamic and propulsive forces on the vehicle divided by the vehicle mass. Total loading (NTOTAL) is the magnitude of $\vec{a}_{sp}$ and the component loadings NX, NY, and NZ are as given below.

$$
\begin{aligned}
N_x &= ({}^{I}\vec{a} - \vec{g}_I) \cdot \hat{x}_b && (85)\\
N_y &= ({}^{I}\vec{a} - \vec{g}_I) \cdot \hat{y}_b && (86)\\
N_z &= ({}^{I}\vec{a} - \vec{g}_I) \cdot \hat{z}_b && (87)
\end{aligned}
$$

2.3 Contributions to the Inertial Acceleration

As seen from Equation 38, the inertial acceleration consists of the net force vector $\vec{F}$ divided by the mass. The components of $\vec{F}$ come from aerodynamics, propulsion, and gravity.

2.3.1 Aerodynamic Force

The aerodynamic force calculations are based upon coefficients provided through the table file. Three sets of coefficients may be defined in TSAP. One is C_L, C_D, and C_S, the lift, drag, and side force coefficients (CL, CD, and CS). These relate the aerodynamic force vector $\vec{F}_a$ to the wind axis system through the equation

$$
\vec{F}_a = -(C_D\hat{x}_w - C_S\hat{y}_w + C_L\hat{z}_w)qS_{ref} \qquad (88)
$$

where q is the dynamic pressure and S_{ref} is the aerodynamic reference area. Similarly, C_x, C_y, and C_z (CX, CY, and CZ) relate the aerodynamic force to the vehicle axis.

$$\vec{F}_a = (C_x\hat{x}_b + C_y\hat{y}_b + C_z\hat{z}_b)qS_{ref} \tag{89}$$

The coefficients C_A and C_N (CA and CN), sometimes used for axisymmetric vehicles, relate to a combination of the wind and body axis systems. To avoid a numerical singularity, the normal force contribution is asssumed to be zero when $\sin\alpha_T = 0$.

$$\vec{F}_a = -\left[C_A\hat{x}_b + C_N\frac{\hat{x}_w\cdot(\hat{y}_b\hat{y}_b+\hat{z}_b\hat{z}_b)}{\|\hat{x}_w\cdot(\hat{y}_b\hat{y}_b+\hat{z}_b\hat{z}_b)\|}\right]qS_{ref} \tag{90}$$

2.3.2 Propulsion Force

The magnitude of the propulsion force is defined in the table input file by the user. The direction of the force vector is along $\hat{x}_b$ by default, but this may be changed in the problem input file as described in Section 4.20. Since the equations of motion are for a point mass, the thrust is considered to act through the mass center of the vehicle.

2.3.3 Gravitation Model

The gravitation model used in TSAP is based on a potential function of the form

$$V = \frac{GM}{r}\left\{1+\sum_{n=2}^{\infty}\sum_{m=0}^{n}\left[\frac{R_\oplus}{r}\right]^n P_{n,m}(\sin\delta_{gc})\left[C_{n,m}\cos(m\lambda)+S_{n,m}\sin(m\lambda)\right]\right\} \tag{91}$$

Here, the $P_{n,m}(\sin\delta_{gc})$ terms are Legendre functions of the form

$$P_{n,m}(\sin\delta_{gc}) = \cos^m\delta_{gc}\frac{d^m}{d(\sin\delta_{gc})^m}\left\{\frac{1}{2^n n!}\frac{d^n}{d(\sin\delta_{gc})^n}\left[\left(sin^2\delta_{gc}-1\right)^n\right]\right\} \tag{92}$$

associated with the zonal ($n > m = 0$), sectorial ($n = m > 0$), and tesseral ($n > m > 0$) harmonics, while the GM, $C_{n,m}$, and $S_{n,m}$ terms are constants. However, in TSAP the degree n is truncated to (at most) four and the order m to zero. Hence, the first three zonal harmonic coefficients may be non-zero but the sectorial and tesseral harmonics are zero. The corresponding simplified gravitational potential function may be written as follows, with $J_n = -C_{n,0}$.

$$V = \frac{GM}{r}\left\{1-\sum_{n=2}^{4}\left[\frac{R_\oplus}{r}\right]^n J_n P_{n,0}(\sin\delta_{gc})\right\} \tag{93}$$

The gravity model constants used in TSAP are discussed in Section 4.6 and references 6 and 7.

2.4 Unit System Conventions

The English system of units is used for standard input and output values. For the most part this means force in pounds, length in feet, angles in degrees, and time in seconds. The units used in specific instances are noted in Sections 3 and 4 for input and Appendix B for output. The user defined output variables, described in Sections 4.4 and 4.25, may be used to convert a limited number of output quantites to other unit systems.

3 The Table Input File

The fidelity with which TSAP can simulate a vehicle trajectory is partially dependent upon the validity of the aerodynamic and propulsion properties ascribed to the vehicle. Aerodynamic properties are defined in terms of force coefficents (e.g., lift and drag coefficents) and a reference area while propulsion data consists of thrust magnitude, direction, and mass flow rate. Data of this sort is grouped into a labeled *table set* and placed into a *table input file*. As many as 99 table sets of each type, each representing different flight conditions or vehicle configurations, may be placed in the table file and referenced as needed during trajectory integration. Definition of the terms can be almost arbitrarily complex although no more than 20,000 words may be used to store the table sets in use at any one time. (The standard printout lists the amount of storage required for each table set.)

In addition, a heating table set is available. It may be thought of as an auxiliary data processor with which up to three variables computed using trajectory data may be defined. As for the other table sets, heating variables may be defined by complex relationships. Although primarily for output purposes, these variables may also be used to influence the trajectory integration.

Once included in the table input file, the selection of a table set for use during trajectory calculations is specifed in the problem file (see Sections 4.1, 4.9, and 4.20). If no such specification is made or if a table file is not provided, the corresponding data elements are assumed to be zero.

All PMAST table sets may be used with TSAP, and hence, table sets may be input in the fixed format defined in the PMAST users manual[1]. TSAP is also capable of reading free format table sets of similar structure which relax the column dependency rules. Table sets may even mix the two input styles although some individual lines with several parts require the exclusive use of one input style or the other. The remainder of this section will describe the use of free format table sets.

The first 80 characters of each line in the table file are used for data input; information beyond column 80 is ignored. Blank lines and lines containing an asterisk (*) in column one are regarded as comment lines and are ignored. All other lines are parsed into data items with maximum length of 20 characters. (Hence, numeric inputs should not exceed 20 characters in length.) Comma, equal sign, and space characters on the original lines serve to delimit the items. (As in the problem file, the relative value signs $<$ and $>$ are also interpreted as delimiters. They should not be used in the table file, however, as their special function is not available.) The data items are processed by the table file input processor according to the procedures outlined in this section.

3.1 Table Set Identification Labels

Within the table file the beginning of a table set is indicated by enclosing the first item on a line in parentheses. Such an item becomes the table set name or identification label and may be up to nine characters long. This label will be used in the problem file to specify use of the table set. Hence no other table sets in the table file should have the same name.

The first item on the next line of a table set specifies the table type via one of the keywords AERO, PROP, or HEATING. Any data to the right of the type keyword (up to 50 characters in length) becomes the table set title. Upon execution of TSAP the table set name and title are printed after the table has been successfully read.

The end of a table set is indicated by either the beginning of a new table set (that is, a line which begins with an item in parentheses) or the end of the table file. However, the sections within a table set must be properly terminated (with an END statement) or an error condition will result.

3.2 Sections

Each type of table set can define a finite number of functions, which in turn define procedures for computing given values. For example, a propulsion table set can be used to define a thrust function, a mass flow rate function, or both. The procedures for computing each are described in distinct parts of the table set known as sections. Each section is named by a specific keyword; each table set type has its own name keywords. These are listed in Table 1. At each integration time step the thrust function defined in the THRUST section of the active propulsion table set will be called and used to compute a value for thrust. Similarly, the functions defined in other sections will be used to compute their associated values.

A section begins on a line which has the keyword SECTION as the first item followed by the section name keyword. Several optional parameters, which are dependent on the section named, may follow. Within a table set the order in which the sections are entered is not important. If a section is left out its corresponding value will be set to zero when the table set is called.

Any one of the three distinct sets of aerodynamic coefficients defined in Section 2.3.1 may be defined in an AERO table set. The set used is specified on the CA/CD section line with one of the optional parameters CA (for the default C_A C_N set), CD (for the C_D C_L C_S set), or CX (for the C_X C_Y C_Z set). The aerodynamic reference area and reference length may also be set on this line by matching values to the keywords SREF and LREF. The following example section line initiates the section CA/CD, specifies that the coefficients C_X, C_Y, and C_Z will be used, sets S_{ref} to 4.37 ft^2, and sets L_{ref} to 17.84 ft.

```
SECTION  CA/CD   CX   SREF=4.37   LREF=17.84
```

Table Type Keyword	Section Name Keywords	Functional Description
AERO	CA/CD	C_A or C_D or C_X
	CN/CL	C_N or C_L or C_Z
	CS	– C_S or C_Y
	CM	C_M C_M C_M
PROP	THRUST	Thrust Force (lb)
	WTDT	Fuel Flow Rate (lb_m/sec or slug/sec)
HEATING	QWALL1	Heating Rate 1
	QWALL2	Heating Rate 2
	QWALL3	Heating Rate 3

Table 1: Table Set Section Name Keywords and Descriptions

As noted in Section 4.1, values for the reference area and length specified in the problem file will override those specified in the table file. The C_M aerodynamic section is not used for integration of the point mass equations of motion but may be useful as an output variable in some applications.

In the propulsion table set the user may specify the units in which mass flow rate is given. To do so the appropriate keyword LB/SEC (default) or SLUG/SEC should appear on the WTDT section line. Since propulsion devices usually expel mass, a positive fuel flow rate value resulting from the WTDT section corresponds to a decreasing vehicle mass. Hence the value of the WTDT table section will be of opposite sign to the WTDT output variable.

3.3 Terms

Whereas each table file is composed of tables sets of various types and each table set is composed of sections which define the functions for computing certain values, so each section is composed of subordinate units known as *terms*. Each term defines an operation to be performed on the current section value. The complete set of terms in a section define the function for computing the output section value. A term consists of at least one line of input data containing a mathematical operation. Additional lines may be required if the term includes data input by the user as a scalar value or a tabulated function. Some terms, such as END, consist of only the mathematical operation.

Each section of a table set may be thought of as a function subroutine for calculating the quantity specified on the section line. The section has an associated *section value* which is zero at the beginning of the calculations and may be modified by each successive

ADD	Add the term value to the section value
SUB	Subtract the term value from the section value
MULT	Multiply the term value by the section value
DIV	Divide the section value by the term value
IDIV	Divide the term value by the section value
EXP	Take the section value to the term value power
IEXP	Take the term value to the section value power
MAX	Limit the section value to less than the term value
MIN	Limit the section value to greater than the term value
ABS	Take the absolute value of the section value
NEG	Negate the section value
SQR	Square the section value
SQRT	Take the square root of the section value
LN	Take the natural log of the section value
LOG	Take the base 10 log of the section value
E	Take e to the power of the section value
SIN	Take the sine of the section value (in degrees)
COS	Take the cosine of the section value (in degrees)
TAN	Take the tangent of the section value (in degrees)
ASIN	Take the arcsine of the section value (in degrees)
ACOS	Take the arccosine of the section value (in degrees)
ATAN	Take the arctangent of the section value (in degrees)
ZERO	Set the section value to zero
STOR	Save the term value and leave the section value unchanged
CSTO	Save the section value under the temporary variable name, then set the section value to zero
LT	If the current section value is less than the given variable, jump to a given label
GT	If the current section value is greater than the given variable, jump to a given label
JMP	Jump to a given label
LABL	Define a label
END	End the calculations for this section

Table 2: Math Operations Available within the Table Input File

term. The calculations are terminated by the END term (the mandatory final line of each section) and the resulting section value is assigned to the quantity of interest. A number of the standard output variables are available for reference by individual term lines, and further data used to modify the section value may be input by the user in the form of scalar values or tabular data. Each section may consist of an (almost) arbitrarily large number of terms, which may be combined in an arbitrarily complex manner.

The mathematical operations listed in Table 2 are available for use in the table input file. The first group, from ADD through MIN, operate on the section value and a *term value*. Terms employing these mathematical operations must provide additional information to define the term value. The next group, from ABS through ATAN, operate only on the section value; therefore the term consists of only the mathematical operation. The STOR and CSTO operations provide storage of temporary calculations (STOR requires a term value and CSTO does not). The operations LT, GT, JMP, and LABL provide some branching capability; the LT and GT operations require a term value while the JMP and LABL operations do not. Finally the END operation is used to terminate the section.

A term value may be a constant, a trajectory value computed by the program (such as altitude or Mach number) and referenced by its standard output variable name, a stored value (from STOR or CSTO), a segment constant value from the problem input file (from the CONSTANTS segment data block), or a value interpolated from a table. The way in which each is entered is specified below.

3.3.1 Constant Term Values

A term value may be specified as a constant by putting a numerical value to the left of the math operation. An optional *term label,* used for reference by a LT, GT, or JMP operation, may be placed left of both the numerical value and the math operation. The following example of this format may be interpreted as ***add 35.64 to the section value,*** with ADD35 the term label.

```
ADD35   35.64   ADD
```

Numeric values may be given in decimal format (as above) or in exponential format (as in the next example).

For consistency with previous table file format versions, constant term values may also be input on two lines, the first consisting of the (optional) term label and the math operation and the second of the constant value. An example of such a term follows.

```
X1          MULT
      1.0E-6
```

Here, the section value will be multiplied by 1.0×10^{-6}. Note that (for the free format style of TSAP table files) the order of the items on each line is important but their exact placement is not (provided each line is less than 80 characters long).

3.3.2 Trajectory Variable Term Values

A limited number (66, listed in Tables 3 and 4) of the standard output variables are available for use as term values or as independent variables for tabulated data. For example, the term

```
MACH        ADD
```

adds the current value of Mach number to the section value. Each time the section value is computed, the current value of Mach number will be used for the term value. Term labels may not be used for this type of term.

ALFA	angle of attack (deg)
ALFAT	total angle of attack (deg)
ALT	geodetic altitude (ft)
ALTD	geodetic altitude rate (ft/sec)
BANKGC	bank angle relative to local geocentric (deg)
BANKGD	bank angle relative to local geodetic (deg)
BETA	sideslip angle (deg)
BETAE	Euler sideslip angle (deg)
DYNPRS	dynamic pressure (lb/ft^2)
GAMGC	local geocentric vertical flight path angle (deg)
GAMGD	local geodetic vertical flight path angle (deg)
LATGC	geocentric latitude (deg)
LATGCD	rate of change of geocentric latitude (deg/sec)
LATGD	geodetic latitude (deg)
LATGDD	rate of change of geodetic latitude (deg/sec)
LONG	longitude (deg)
LONGD	rate of change of longitude (deg/sec)
LREF	aerodynamic reference length
MACH	Mach number
MASS	vehicle mass (slugs)
PATHLENG	trajectory pathlength (ft)
PATHLMAR	trajectory pathlength since last mark (ft)
PATHLSEG	trajectory pathlength for current segment (ft)
PHI	windward meridian (deg)
PITCHGC	pitch Euler angle with respect to the geocentric horizon (deg)
PITCHGD	pitch Euler angle with respect to the geodetic horizon (deg)
PITCHI	pitch Euler angle with respect to the inertial platform horizon (deg)
PRESS	freestream atmospheric pressure (lb/ft^2)
PSIGC	geocentric horizontal flight path angle (deg)
PSIGD	geodetic horizontal flight path angle (deg)
RANGE	length of the flight path projected onto the earth's ellipsoid (nm)
RCM	radial distance from earth's center to the vehicle (ft)
RCMD	rate of change of RCM (ft/sec)
REYPFT	Reynolds number per foot (ft^{-1})
RHO	freestream atmospheric density ($slugs/ft^3$)
RNGMARK	range since last mark (nm)
RNGSEG	range during current segment (nm)
ROLLGC	geocentric Euler roll angle (deg)
ROLLGD	geodetic Euler roll angle (deg)
ROLLI	inertial platform Euler roll angle (deg)
SEGMENT	current segment number
SNDSPD	freestream speed of sound (ft/sec)
SREF	aerodynamic reference area (ft^2)

Table 3: First 43 Standard Output Variables Recognized in the Table Input File

TEMP	freestream atmospheric temperature (deg Rankine)
TIME	path time (sec)
TMARK	time since last mark (sec)
TSEG	time of current segment (sec)
TTARG	target mission time (sec)
VAIR	vehicle speed with respect to the air mass (ft/sec)
VEL	earth relative velocity magnitude (ft/sec)
VGROUND	ground speed (ft/sec)
VISC	freestream kinematic viscosity (ft^2/sec)
WINDD	downward component of wind velocity (ft/sec)
WINDE	eastward component of wind velocity (ft/sec)
WINDN	northward component of wind velocity (ft/sec)
WT	vehicle mass (weight) (lb_m)
XCG	center of gravity position in vehicle
XECF	ECFC position X component (ft)
XECFD	ECFC velocity X component (ft/sec)
YAWGC	geocentric pitch Euler angle (deg)
YAWGD	geodetic pitch Euler angle (deg)
YAWI	inertial platform pitch Euler angle (deg)
YECF	ECFC position Y component (ft)
YECFD	ECFC velocity Y component (ft/sec)
ZECF	ECFC position Z component (ft)
ZECFD	ECFC velocity Z component (ft/sec)

Table 4: Last 23 Standard Output Variables Recognized in the Table Input File

3.3.3 Stored Term Values

Temporary values can be computed and saved in sections with the STOR and CSTO operations. These operations use the term label as a variable name to identify the stored quantity. In subsequent terms this variable name can be used to refer to the stored value; thus, the stored value can be used as a term value or as an independent variable for tabulated data.

In the following example the correction required to estimate thrust when vacuum thrust data is available is computed and stored. The correction can then be added to the vacuum thrust to give the actual thrust.

```
PRESS      SUB
ANOZL  4.2763711    MULT
BPCOR      CSTO
         .
         .  (Additional terms which compute vacuum thrust)
         .
BPCOR      ADD
```

The first term subtracts the freestream pressure from the section value (which is zero initially). The next term multiplies the result by 4.2763711 (the nozzle area). The results are stored and may be referred to later by the name BPCOR. The CSTO operation is useful for temporary calculations because it both stores the current section value and then clears it or sets it to zero. Storage variables must have been defined with a STOR or CSTO operation before they can be referenced.

3.3.4 Segment Constant Values

If a term variable name is encountered in the table file that is not one of the standard output variables listed in Tables 3 and 4, is not associated with a constant or an interpolation table, and has not been defined in a STOR or CSTO term, it is assumed to be a segment constant value input by the user via the CONSTANTS segment data block in the problem input file (see Section 4.3). Segment constant values permit the user to change the way section values are computed in each segment of a flight path while using a single table, and permit some parameters of the table file to be set in the problem file.

For example, the terms

```
FLAG1      LT  LABL1
                .
                . (Additional terms)
                .
LABL1    3.141592654  ADD
```

use a segment constant value (called FLAG1) in a branching operation. The first term can be interpreted as *if the value of* FLAG1 *is less than the section value, then jump to the term with a label of* LABL1; *otherwise, execute the next term.* The name FLAG1 has not been stored previously by a STOR or CSTO operation so it is assumed the user will provide its value in a CONSTANTS segment data block in the problem input file (see Section 4.3). For example, the problem file line

```
CONSTANTS  FLAG1=2
```

sets the value of FLAG1 to 2.0.

3.3.5 Interpolated Term Values

Tabulated data is frequently used for defining term values. Data tables which employ as many as five independent variables to define the dependent variable may be used. To do so the independent variable names (which must be trajectory, stored, or segment constant values) are listed following the mathematical operation of a term line. The data which makes up the tabulation follows on subsequent lines. The entire table definition is considered a single term. All interpolation of tabular data is linear.

One-dimensional Tables

An example of a simple one-dimensional table which uses Mach number as the independent variable is

```
TABL1   ADD    MACH
    8
0.0     2.0     4.0     6.0     8.0     10.0   12.0    14.0
0.0002  0.0003  0.0003  0.0005  0.0008  0.0015 0.0025  0.0060
```

The table independent variable (or x variable) name, in this case MACH, is input following the math operation on the term definition line. The trajectory variables listed in Tables 3 and 4, storage variables, or segment constant values can be used as independent variable names in tables. The next line contains an integer giving the number of tabulated points, in this case 8. There is no size limit on each table although the overall limit of 20,000 words on the amount of storage required for all table sets used concurrently in a problem applies.

The eight Mach number values are given next, followed on the next line by the dependent variable (or y variable) values (or term values). It is important that the number of values given correspond to the integer value given above defining the number of points, that no more than 20 values be given on any one line, and that the independent value lines are distinct from the dependent value lines. Moreover, values of the independent variable (here, MACH) must be strictly increasing. Discontinuities in the data can be simulated by very small changes in the value of the independent variable. A table employing this technique is shown below.

```
TABL1   SUB     ALT
  11
0.      10000.  20000.  30000.   40000.   50000.
50001.  60000.  70000.  100000.  200000.
0.0011  0.0013  0.0017  0.0022   0.0029   0.0038
0.0052  0.0053  0.0055  0.0060   0.0071
```

This table is a function of altitude and has a discontinuity at 50,000 ft where the two different values have been input corresponding to 50,000 ft and 50,001 ft.

Two-dimensional Tables

Tables of two independent variables may be defined using either of two methods, depending on the available tabulated values as shown in Figure 6. When a square matrix of values is available (that is, for each combination of independent variable values there exists a dependent value), the table may be input using a special format. An example of this format is given in Figure 7. The first independent variable, MACH in the example, follows the math operation on the term definition line and is referred to as the *x variable.* Following the x variable is the name of the second independent variable, ALT, referred to as the

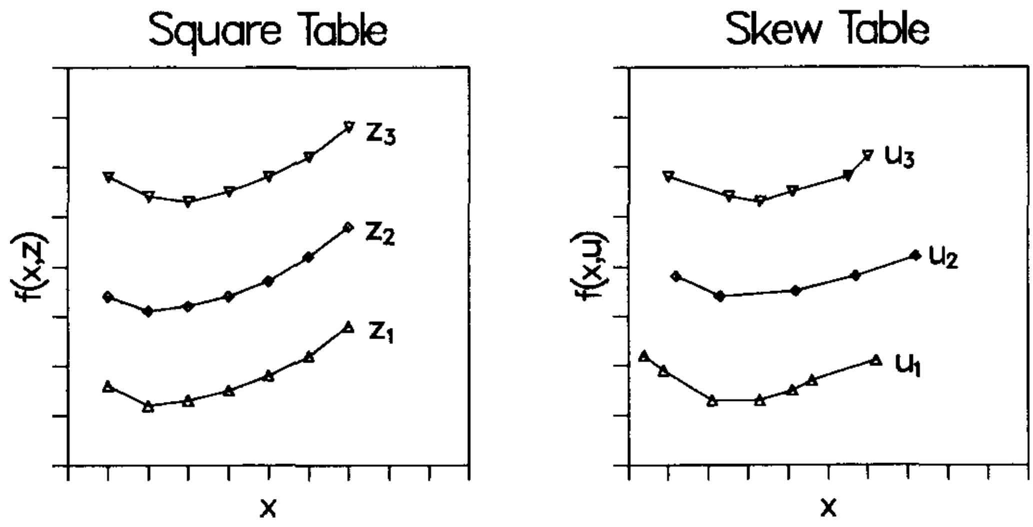

Figure 6: Both *square* and *skew* two-dimensional tables may be input.

```
TABL2       ADD         MACH        ALT
   9     4      (No. of Mach's and Altitudes)
* Mach numbers
2.          4.          6.          8.          10.         12.
14.         16.         20.
* Altitudes
0.          50000.      100000.     200000.
* Data for altitude=0
0.25        0.30        0.40        0.50        0.55        0.58
0.54        0.45        0.30
* Data for altitude=50000
0.27        0.32        0.43        0.54        0.59        0.62
0.57        0.48        0.39
* Data for altitude=100000
0.29        0.34        0.45        0.57        0.64        0.69
0.65        0.61        0.43
* Data for altitude=200000
0.32        0.37        0.56        0.64        0.70        0.77
0.73        0.65        0.51
```

Figure 7: A Two-Dimensional Table with Square Data

z variable. For the general table the x and z variables may be any trajectory variable, storage variable, or segment constant value.

The second line of Figure 7 contains the number of Mach numbers (or x values) followed by the number of altitudes (or z values). This is followed by lines giving the Mach number values (or x values) followed by lines containing the altitude values (or z values). Both sets

```
TABL2      ADD   MACH   ALT   SKEW
    4       (No. of Altitudes)
0.0         (Altitude = 0 ft)
    9       (No. of Mach numbers at Alt=0)
2.         4.         6.         8.         10.        12.
14.        16.        20.
0.25       0.30       0.40       0.50       0.55       0.58
0.54       0.45       0.30
50000.0     (Altitude = 50,000 ft)
    9
2.         4.         6.         8.         10.        12.
14.        16.        20.
0.27       0.32       0.43       0.54       0.59       0.62
0.57       0.48       0.39
100000.0    (Altitude = 100,000 ft)
    7
6.         8.         10.        12.        14.        16.
20.
0.45       0.57       0.64       0.69       0.65       0.61
0.43
200000.0    (Altitude = 200,000 ft)
    5
10.        12.        14.        16.        20.
0.70       0.77       0.73       0.65       0.51
```

Figure 8: A Two-Dimensional Table with Skew Independent Variable Data

of independent variable values must be strictly increasing. The remaining lines contain the tabulated term values (or y values). Data corresponding to each z variable must begin on a new line.

Another method is used to input a two-dimensional table when a square matrix of data is not available. A table of this sort is indicated by introducing the keyword SKEW anywhere following the math operation on the term definition line. In skew tables the second independent variable is known as the *u variable.* The example table in Figure 8 defines MACH as the x variable and ALT as the u variable. Note that data is available at different Mach numbers for each altitude: the table is skew.

The second line contains the number of u values or altitudes, in this case 4. This is followed by a line with the first altitude value, 0.0. The next line contains the number of Mach numbers for the first altitude, 9. The Mach number values are listed in ascending order on the next two lines followed by the tabulated dependent values or term values.

The second altitude value, 50000.0, is input next along with lines giving the number of Mach numbers at this altitude, the Mach number values, and the term values. This block of data is followed by the third altitude value and its data, and so on. It should be noted that there are only seven Mach number values for 100,000 ft altitude and five Mach number values for 200,000 ft altitude. Thus in a skew table different sets of Mach numbers (x values) can be given at each altitude (u value).

Multi-dimensional Tables

Tables of higher dimensions may be used by adding further independent parameters (the *v variable* and the *w variable*) as required. The z variable may be used if a square matrix of independent data pairs is available in at least two dimensions, in which case tables of up to five dimensions (five independent variables) may be used. Otherwise, the keyword SKEW must appear, and tables may not exceed four dimensions.

Tabulated values for these variables are input similarly to the skew table shown in the previous section. An example of a three-dimensional table that allows all of the independent values to be skewed is defined with an x variable, u variable, and v variable in

```
TABL3    ADD   MACH    SKEW    ALT    TYPE
    2       (Number of types)
0.0         (Type=0 - term value always zero)
*
    2       (Number of altitudes for Type=0)
0.0         (1st altitude for Type=0)
*
    2       (Number of Mach's at Alt=0, Type=0)
0.0         20.0        (Mach numbers)
0.0         0.0         (Term values)
*
200000.     (2nd altitude for Type=0)
    2       (Number of Mach's at Alt=200000, Type=0)
0.0         20.0        (Mach numbers)
0.0         0.0         (Term values)
*
*   All through for Type=0...now enter Type=1 data
*
1.0         (Type=1 - term value non zero)
    2       (Number of altitudes for Type=1)
0.0         (1st altitude for Type=1)
*
    5       (Number of Mach's at Alt=0, Type=1)
2.0         6.0         10.0        15.0        20.0
13000.      13500.      14500.      14000.      12000.
*
200000.     (2nd altitude for Type=1)
    4       (Number of Mach's at Alt=200000, Type=1)
7.0         12.0        16.0        22.0
15000.      16000.      17500.      17000.
```

Figure 9: A Three-Dimensional Table in which All Independent Data Sets are Skew

Figure 9. The presence of the SKEW keyword indicates that the z variable will not be used and that the innermost independent variable (the x variable) can be skewed. The table is a function of MACH, ALT, and TYPE where TYPE is a segment constant value controlled in the problem input file. When type is equal to zero, the term value is zero; and when type is equal to one, the term value is non zero.

The tabulated values are input starting with the right most independent variable, in this case TYPE. The number of TYPE's, 2, is input on the first line after the term definition line. This is followed by a line with the first value of TYPE (the values must be strictly increasing). Proceeding to the next independent variable, one enters the number of altitudes for TYPE=0, in this example 2, and follows with the first altitude value, 0. The next line contains the number of MACH's for TYPE=0 and ALT=0. Since this is the innermost independent variable, the Mach number values are input in a group followed by the term values.

At this point all data has been input for TYPE=0 and ALT=0 so the next ALT value is input, 200000, followed by its MACH's and term values. Since there is only data at two altitudes for TYPE=0, this section of the table is complete. The data for TYPE=1 is entered next in the same manner except that there are additional Mach numbers and term values at each altitude.

When entering multi-dimensional tables, such as the one shown above, one must ensure that *the indicated number of values corresponds to the actual number of values input.* In other words, if the number of altitudes has been given as 3, then three altitude values must be given. When modifying existing tables it is easy to add data (for example another altitude) but neglect to change the number indicating how many altitude values to expect. When reading the table file, TSAP may not even notice this sort of error or, if it does, may not catch it until it reaches the beginning of the next term. Hence, multi-dimensional tables should be checked carefully for accuracy.

The four-dimensional table presented in Figure 10 illustrates the difference when a *z variable* is present (i.e. the SKEW keyword does not appear) and the two innermost independent variables are square. This table is a function of four independent variables: angle-of-attack, Mach number, altitude, and TYPE. Again, the tabulated values are input starting with the right most independent variable, TYPE. The number of TYPE's, 2, is input on the first line followed by the first type value, 0.0, on the next line. The next variable is altitude so the number of altitudes and the first altitude value, corresponding to TYPE=0.0, are input on the next two lines.

Both an x variable (ALFA) and a z variable (MACH) have been given so they are loaded as a square table; that is, a term value is input at each angle-of-attack/Mach number combination. The number of angle-of-attack values and the number of Mach numbers is input next. The set of angle-of-attack values follows, begining on a new line, after which the Mach number values appear, also on a new line. Finally, the tabulated term values are input. The first set of term values corresponds to the first Mach number (a value must

```
TABL3      ADD        ALFA       MACH       ALT        TYPE
    2      (Number of types)
0.0        (Type=0)
*
    2      (Number of altitudes for Type=0)
0.0        (1st altitude for Type=0)
*
    4    4    (Number of Alfa's and Mach's at Alt=0, Type=0)
0.0        4.0        8.0        12.0     (Alfa's)
2.0        6.0        10.0       14.0     (Mach numbers)
0.0033     0.0040     0.0051     0.0062   (Values for Mach=2)
0.0034     0.0042     0.0053     0.0066   (Values for Mach=6)
0.0032     0.0041     0.0052     0.0064   (Values for Mach=10)
0.0030     0.0039     0.0048     0.0062   (Values for Mach=14)
*
100000.    (2nd altitude for Type=0)
    4    3    (Number of Alfa's and Mach's at Alt=100000)
0.0        4.0        10.0       15.0     (Alfa's)
8.0        12.0       16.0                (Mach numbers)
0.0040     0.0060     0.0072     0.0091   (Values for Mach=8)
0.0042     0.0062     0.0075     0.0095   (Values for Mach=12)
0.0045     0.0066     0.0079     0.0108   (Values for Mach=16)
*
* --------------------------------------------------------
*
2.0        (Type=2)
    2      (Number of altitudes for Type=2)
20000.     (1st altitude for Type=2)
    3    4       (Mach/Alfa table for Type=2, Alt=20000)
0.0        10.0       20.0
4.0        8.0        12.0       16.0
0.0020     0.0022     0.0024
0.0021     0.0024     0.0028
0.0022     0.0026     0.0033
0.0024     0.0029     0.0038
*
150000.    (2nd altitude for Type=2)
    5    4       (Mach/Alfa table for Type=2, Alt=150000)
0.0        4.0        8.0        12.0       16.0
8.0        12.0       16.0       20.0
0.0030     0.0033     0.0037     0.0045     0.0056
0.0032     0.0035     0.0039     0.0047     0.0058
0.0028     0.0031     0.0033     0.0041     0.0049
0.0025     0.0029     0.0031     0.0038     0.0045
```

Figure 10: A Four-Dimensional Table with Square Data

be given for each angle-of-attack). The next set corresponds to the second Mach number, and so on. Each such set may extend over several lines so long as each begins on a new line.

As shown with the two examples in Figures 9 and 10 the general pattern is to enter the first set of tabulated values starting with the right most independent variable and working left. Values for the independent variables must be input in ascending (strictly increasing) order, and the correct number of values must be specified.

Table Extrapolation Option

By default tabulated data defining the term value will be extrapolated. Since linear interpolation is used, the values are extrapolated linearly also. The ***no extrapolation*** option causes the term value to remain within the bounds of the tabulated values, as shown in Figure 11. Extrapolation can be set on or off via the keywords EXTRAP (default) or NOEXTRAP, the latter of which prevents extrapolation. If used, the extrapolation keywords must appear to the left of the math operation on the term definition line, as shown below.

```
TABL1   NOEXTRAP  ADD    MACH
     8
0.0        2.0       4.0       6.0       8.0       10.0
12.0       14.0
0.0002     0.0003    0.0003    0.0005    0.0008    0.0015
0.0025     0.0060
```

For this example term values would be interpolated normally for Mach numbers between 0.0 and 14.0. For Mach numbers greater than 14 the term value would be fixed at 0.0060;

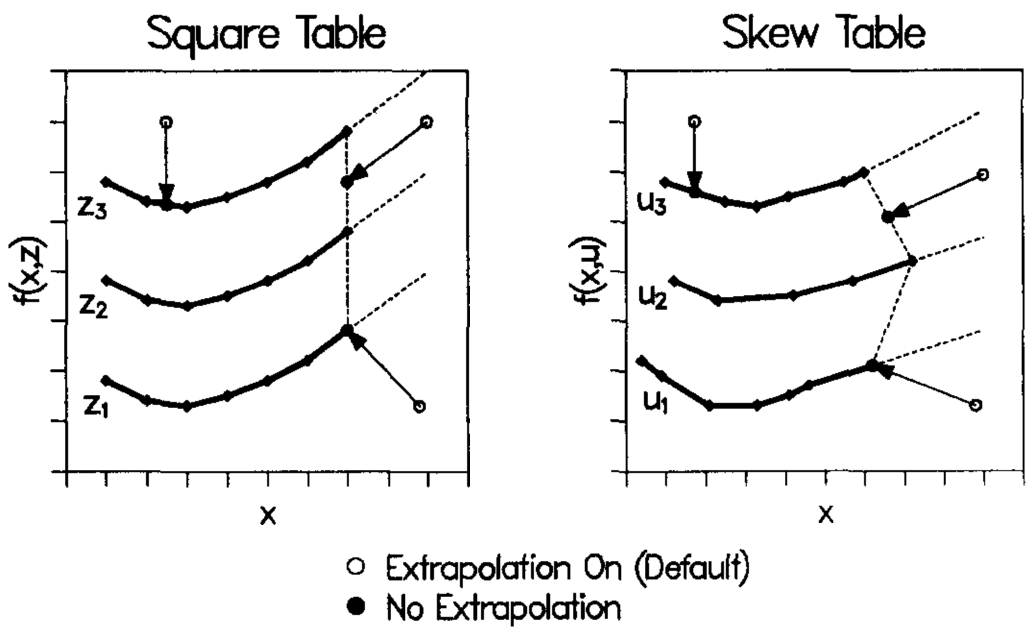

Figure 11: An available term definition line keyword eliminates extrapolation.

and for Mach numbers less than 0.0 the term value would be fixed at 0.0002.

3.3.6 Storage Variables

Storage variables, created with the STOR and CSTO operations, provide some unique capabilities for defining functions. As an example, a term value can be stored with the STOR operation as shown below.

```
ST1       STOR      ALFA
   8
0.0       2.0       4.0       6.0       8.0       10.0
12.0      14.0
-0.0021   -0.0015   -0.0006   0.0004    0.0011    0.0018
0.0023    0.0035
```

This creates a new variable in the section symbol table called ST1 that has a value interpolated from the tabulated data. The STOR operation does not change the current section value. The new variable ST1 can be used in subsequent terms as a term value or as an independent variable in a table. For example,

```
ST1    ADD
```

or

```
X1     SUB    ST1
 5
-0.0050   -0.0025   0.0       0.0025    0.0050
-5.0      -2.5      0.0       2.5       5.0
```

In the first example the value of ST1 is added to the current section value, and in the second example the value of ST1 is used as an independent variable to interpolate a term value that is subtracted from the section value.

The CSTO (clear and store) operation performs a similar function in that it creates a storage variable; however, it uses the current section value for the storage variable value and it sets the section value back to zero. The term

```
SAV1      CSTO
```

creates a new variable called SAV1 with a value equal to the current section value. Then it zeros the section value. The variable SAV1 can be used as a term value or as an independent variable as shown in the previous example with ST1.

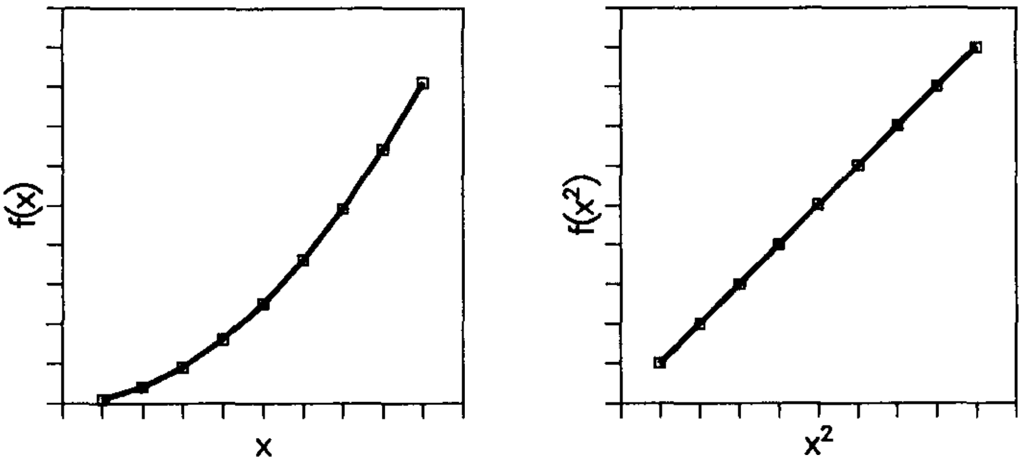

Figure 12: Storage variables can be used to simulate higher order interpolation.

Linear interpolation is always used to obtain the term values from tabulated data; however, storage variables can be used to simulate higher order interpolation. For example, in Figure 12 the original function with x as the independent variable is approximately 2nd order; however, it can be reduced to a nearly linear function if x is replaced with x^2. If x is Mach number, this can be represented in a section containing the following terms.

```
*
*  1st define a storage variable as
*  MACH**2
*
MACH      ADD
MACH      MULT
M2        CSTO
*
*  Now use this as the independent
*  variable for interpolation
*
F1        ADD       M2
    5
4.0       9.0       16.0      25.0      36.0
0.324     0.542     0.856     1.575     4.236
```

This procedure results in 2nd order interpolation for the term value F1.

3.3.7 Branching

Mathematical operations are provided that give a limited IF-THEN type of branching between terms. Branching significantly increases the computer time required to evaluate a section value so it should be used as a last resort. Instead of using branching, one can often add an independent variable to act as a flag: if its value is zero, the table returns a zero term value; and if its value is non-zero, the table returns a non-zero term value.

The LT and GT operations provide IF-THEN branching, and the JMP operation provides a direct transfer (no IF test). All of these operations require a transfer term label to be input left of the math operation on the term definition line. For example,

```
IF1   2.0   LT   TRM2
  .
  . (Other terms)
  .
TRM2        ADD         ALT
    3
0.0         50000.      100000.
0.3452      3.3450      6.8921
```

causes transfer to the term TRM2 if the term value 2.0 is less than the section value. If the transfer condition is not satisfied, the next term in the section will be executed.

A term is identified with the name given left of the math operation on the term definition line unless this name is a trajectory variable name, a storage variable name, a segment constant value name, or an extrapolation keyword. In the example shown above, the first term has a label of IF1 and the second term has a label of TRM2. If the term value is a trajectory variable, a storage variable, or a segment constant value, a term label cannot be assigned. Instead, an additional term may be added employing the LABL operation. The LABL operation acts strictly as a term label; it does not use a term value, and performs no operations on the section value. The example shown below is similar to the previous one except that transfer is made to a LABL term.

```
IF1   2.0   LT   TRM2
  .
  . (Other terms)
  .
TRM2        LABL
MACH        ADD
PRESS       MULT
  .
  . (etc.)
  .
```

Since the term MACH uses a trajectory variable as its term value it cannot have a label. Thus a LABL term has been inserted in front of it to act as a term label. Obviously this is only required when branching to a specific term in the section that cannot be labeled.

Although only simple IF tests are provided, they can be used to create IF-THEN-ELSE structures as demonstrated below.

```
C1  50000.  ADD
ALT  GT  LAB1
*
* Execute the next set of terms
* If altitude < 50,000 feet
ZERO
DX          ADD         MACH
    4
2.0         4.0         6.0         10.0
0.003       0.002       0.004       0.006
     JMP LAB2
*
* Execute the next set of terms
* If altitude > 50,000 feet
LAB1        LABL
ZERO
DX2         ADD         MACH
    4
8.0         12.0        16.0        20.0
0.005       0.006       0.008       0.010
*
LAB2        LABL
```

Assuming a section value of zero upon entry into this set of term lines, the first term sets the section value to 50,000 so that the next term can test on it. The IF test can be interpreted as *if altitude is greater than 50,000 feet, then go to label* LAB1. Consequently if the altitude is less than or equal to 50,000 the DX table is interpolated, otherwise, the DX2 table is interpolated. The JMP operation following the DX term transfers to LAB2 so that both sequences of logic arrive at the same point.

3.4 Example Tables

The following sections present examples of an aerodynamic table set and a propulsion table set. These table sets were designed for a ballistic launch and reentry trajectory where separate aerodynamic data was required for the launch and reentry portions of the flight path.

3.4.1 Aerodynamic Table Set

Figure 13 shows an example of an aerodynamic table set where only the body axial force coefficient has been defined. All of the other aerodynamic coefficients will be set to zero.

```
(EXAMPLE1)
AERO       AN EXAMPLE OF AN AERO TABLE SET
SECTION    CA/CD    CA      SREF=2.0
CAB        ADD      MACH    SKEW    BOOST
    2
* NO BOOSTER (BOOST=0)
0.0
    2
0.0        20.0
0.0        0.0
* BOOSTER (BOOST=1)
1.0
    5
0.0        0.8        1.2        4.0        30.0
1.234      1.234      2.423      1.862      1.435
CARV       ADD        MACH       ALT        RV
    2
* NO REENTRY DRAG (RV=0)
0.0
    2    2
0.0        30.0
0.0        1000000.
0.0        0.0
0.0        0.0
* REENTRY DRAG (RV=1)
1.0
    6    7
2.0        4.0        8.0        12.0       16.0       20.0
0.0        50000.     75000.     100000.    150000.    200000.
300000.
.35715     .23613     .24038     .18639     .15653     .14215
.35922     .22929     .25265     .19849     .16826     .14372
.28380     .25726     .25458     .18544     .17819     .15490
.30638     .26995     .26034     .19553     .18523     .13054
.32345     .28557     .27569     .19063     .19930     .15397
.33759     .24956     .26011     .19503     .21136     .17350
.44974     .30401     .30099     .28137     .24975     .16448
           END
```

Figure 13: An example AERO table set which employs segment constant values.

It only has one section, CA/CD, and the section parameter CA (as opposed to CD or CX) indicates which type of data it contains. The reference area is 2.0 square feet; no reference length is given.

In this example separate sets of aerodynamic data are available for the launch and reentry vehicle configurations. One way to change aerodynamic data in different mission segments is to set up separate table sets for each configuration and to designate which table set to use on each segment. Another approach, shown here, is to make use of segment constant values as independent variables. In Figure 13 the segment constant values BOOST and RV are used in the terms to control what table is used in the various parts of the trajectory. Values for BOOST and RV must be set in the CONSTANTS segment data block in the problem input file for each flight path segment (see Section 4.3).

During boost phase, the segment constant values would be BOOST=1 and RV=0. From the tabulated data given in the terms, these values would cause the first term to interpolate a table for the booster drag and the second term to return zero. During reentry the values would be BOOST=0 and RV=1 causing the first term to return zero and the second term to interpolate the reentry C_A.

3.4.2 Propulsion Table Set

The propulsion table set shown in Figure 14 defines the thrust and fuel flow for a three-stage rocket booster system. The first section defines the vacuum thrust as a function of flight path time and subtracts the back pressure from the nozzle. The first three terms compute the back pressure as the nozzle area times the atmospheric pressure and save it as a storage variable called BKPRS. The CSTO operation first saves the current section value then zeros it. The next term sets the section value to the vacuum thrust after which the stored back pressure is subtracted from the vacuum thrust to get the actual thrust at altitude.

The fuel flow section illustrates a simple case where the section consists of only one term with a single-dimensional table. As indicated by the keyword on the section line, the fuel flow values are given in units of lb/sec. The table is given as a function of the flight path time and accounts for the discontinuities due to staging. Note that both the thrust and fuel flow are zero for path times greater than 300.01 seconds.

```
(EXAMPLE2)
PROP       AN EXAMPLE PROPULSION TABLE SET
SECTION    THRUST
*
* COMPUTE BACK PRESSURE = NOZZEL EXIT AREA * PRESSURE
*
ANOZ       ADD        TIME
   12
0.0        125.00     125.01     126.00     126.01     127.00
127.01     290.00     290.01     300.00     300.01     10000.
11.2       11.2       0.35       0.35       0.35       0.35
3.40       3.40       0.35       0.35       0.0        0.0
PRESS      MULT
BKPRS      CSTO
*
* LOOKUP THRUST AND SUBTRACT BACK PRESSURE
*
T          ADD        TIME
   12
0.0        125.00     125.01     126.00     126.01     127.00
127.01     290.00     290.01     300.00     300.01     10000.
220000.    220000.    3000.      3000.      3000.      3000.
30000.     30000.     3000.      3000.      0.0        0.0
BKPRS      SUB
           END
*
* FUEL FLOW SECTION
*
SECTION    WTDT       LB/SEC
WTDT       ADD        TIME
   12
0.0        125.00     125.01     126.00     126.01     127.00
127.01     290.00     290.01     300.00     300.01     10000.
710.0      710.0      10.0       10.0       10.0       10.0
100.0      100.0      10.0       10.0       0.0        0.0
           END
```

Figure 14: An example PROP table set demonstrates the use of storage variables.

4 The Problem Input File

The problem input file defines the conditions under which the trajectory is flown (vis-a-vis the table input file which describes physical characteristics of the vehicle in one or more configurations). It is made up of data blocks consisting of keywords and data in free (column independent) format. The keywords and data are separated by delimiters which consist of spaces, commas, and equal signs in any number and combination. The only limitation is that each line of the input file must not exceed 80 characters in length: any entries beyond column 80 will be ignored. Two special delimiters are available for indicating relative value, those being the less than (<) and greater than (>) signs. A single relative value sign may be combined with the standard delimiters but has meaning only in particular data blocks.

A hierarchy of keywords is used to break the associative rules into a manageable structure. On top of the hierarchy are the keywords representing data block names. Once such a name is encountered, the input file processor recognizes other keywords defined in a particular way for that data block. Such subordinate keywords may be complete by themselves or may be combined with a value in a symbol/value pair. In some instances a particular sequence of keywords and values may be expected. This section defines the keywords, the way they may be combined, and how such combinations will be interpreted and used in the trajectory calculations.

Data blocks are listed sequentially in the problem input file. The order in which they appear is sometimes important when the definitions of one rely on those of another, but otherwise they may be arranged as desired. Multiple versions of a data block are often permitted; for some the last use will supersede earlier ones while for others additional data will be conveyed to the program. A data block begins wherever its name appears as the first (non delimiter) entry on a line and ends where another begins. The names of the data blocks are listed in Table 5.

The DUMP data block is available for consistency with PMAST but is not otherwise supported in TSAP. The EGS-DBF and EGS-PDF data blocks can be used to create output files which the Engineering Graphics System created by David Salguero can use.

AERO	EARTH	INITIAL	PLOT	SUMMARIZE
ATMOS	EGS-DBF	LIMITS	PROP	SURVEY
DEFINE	EGS-PDF	NOPRINT	RADAR	TARGET
DUMP	EXECUTE	OPTIMIZE	SEARCH	TITLE
DWN/CRS	HEATING	OUTPUT	SEGMENT	WINDS

Table 5: Data Block Names

AERO	DELTA	INCREMENT	INTEG	LAUNCH	RESET
CONSTANTS	FLY	INERTIAL	HEATING	PROP	WHEN

Table 6: Segment Data Block Names

Since EGS availability is limited, however, these data blocks will not be considered further in this document.

The SEGMENT data block is special in that it consists of subordinate data blocks known as Segment data blocks which are grouped to define each segment. These are named in Table 6. Note that AERO, HEATING, and PROP are also standard data block names. Which type of data block is applicable is taken from the context of where the name appears in the input file.

The DELTA segment data block defined for PMAST is still available in TSAP. Its function has been superseded, however, by the RESET and INCREMENT segment data blocks and will not be documented in this manual. See the PMAST User's Manual for more information on this data block.[1]

With the exceptions noted, descriptions of all data blocks and segment data blocks are arranged alphabetically and presented in this section.

4.1 AERO Data Block and Segment Data Block

The AERO data block and segment data block provide aerodynamic table set names to be used for retrieving aerodynamic coefficients. Both are optional: if absent, the aerodynamic coefficients are set to zero. When TSAP initializes a trajectory path, it searches the table file for the table set requested in the AERO data block and loads it into the program. Similarly, as each segment is initialized any table sets indicated by AERO segment data blocks are loaded overriding those loaded previously.

The format for both blocks is identical. Two positional parameters are required following the AERO keyword. The first parameter is the word TABLE, and the second parameter is the aerodynamic table set name. The example shown below indicates that an aerodynamic tables set identified with STAGE-1 is to be used. An error message will result if such a table set does not exist in the table input file.

```
AERO  TABLE  STAGE-1
```

The AERO data block may also provide a limited amount of data concerning the vehicle geometry and center of gravity location which may be required for aerodynamic force calculations. Doing so overrides any such values given in the table set. Table 7 lists the keywords available to input this data.

As noted in Section 2.3.1, the aerodynamic reference area S_{ref} is used in computing the aerodynamic forces and the aerodynamic reference length L_{ref} is used to compute the Reynolds number. The center of gravity is given as a fraction of the vehicle length from the nose of the vehicle. For point mass simulations the aerodynamic data is often input for zero moment (trimmed) conditions. Consequently, the aerodynamic coefficients may be functions of the center of gravity since the vehicle trims differently for each c.g. location. (Of course, all such relationships are input via the table input file so the user has control over their formulation.)

The angle DPHIW locates the reference windward meridian relative to the vehicle y-axis $\hat{y}_b$, where DPHIW is measured clockwise as shown in Figure 15. The windward ray and hence the windward meridian correspond to the vector ${}^w\vec{V}_{yz} = {}^w\vec{V} \cdot (\hat{y}_b\hat{y}_b + \hat{z}_b\hat{z}_b)$. The windward meridian is located at an angle ϕ_w (PHIW) from the reference windward

Keyword	Units	Description
CG		center of gravity location
DPHIW	deg	reference windward meridian
LREF	ft	aerodynamic reference length
SREF	ft^2	aerodynamic reference area

Table 7: Aero Data Block Keywords

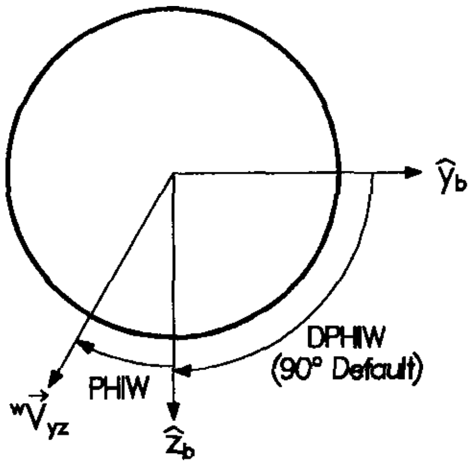

Figure 15: Definition of the Reference Windward Meridian

meridian, or equivalently, at an angle ϕ_{wba} (PHIWBA) from $\hat{y}_b$. The default value of DPHIW is 90 degrees, which aligns the reference meridian with $\hat{z}_b$ (the vehicle z-axis). With this convention when β_E is zero α and α_T are measured in the same plane and have identical absolute value.

An example of the data block which includes these terms is

```
AERO  TABLE  CONFIG1
  SREF=4.23   LREF=15.23   CG=0.70   DPHIW=-90.0
```

which specifies an aerodynamic table set called CONFIG1, a reference area of 4.23 ft^2, a reference length of 15.23 ft, a center of gravity location at 70 percent of the vehicle length, and a reference windward meridian 180° from the default meridian.

4.2 ATMOS Data Block

The ATMOS data block may be used to indicate that atmospheres other than the default 1976 U.S. standard atmosphere are to be modeled. Although the atmosphere models extend to 1000 km, at high altitude (30 km for 75°N, 80 km for 60°N January, and 120 km for all others) all models conform to the 1976 standard. Below sea level the atmospheric density grows exponentially. The Tonopah atmospheres are based on limited meteorological sounding data. Details of the other atmosphere models can be found in references 9 through 13, and reference 14 describes a method of tabulating data from the atmosphere models. The form of the data block is

```
ATMOS  Index
```

where index takes on one of the values 0 through 22 to specify the associated atmosphere listed in Table 8.

The user supplied atmosphere (type 21) requires a table defining the atmosphere parameters as a function of altitude (Figure 16). The table independent variable, ALT, must be listed first and its values (in units of feet) strictly increasing. The remaining variables can be listed in any order as long as the column headings agree with the tabulated values. The temperature is input in degrees Rankin, the pressure in pounds per square foot, the density in slugs per cubic foot, the speed of sound in feet per second, and the kinematic viscosity in ft^2/sec. A maximum of 325 tabulated values can be input. Three interpolation methods are available for the atmosphere table (Table 9). The default method is INTERP-3. The interpolation method is input as the second positional parameter on the ATMOS line, as shown in Figure 16.

Index	Description	Index	Description
0	no atmoshpere	12	75 deg north January
1	1976 U.S. standard	13	75 deg north January (cold)
2	15 deg north annual	14	75 deg north January (warm)
3	30 deg north January	15	75 deg north July
4	30 deg north July	16	Tonopah winter
5	45 deg north January	17	Tonopah spring
6	45 deg north July	18	Tonopah summer
7	45 deg north spring/fall	19	Tonopah fall
8	60 deg north January	20	Kwajalein mean annual
9	60 deg north January (cold)	21	user supplied atmosphere data
10	60 deg north January (warm)	22	site measured atmosphere data
11	60 deg north July		

Table 8: Atmosphere Model Index Descriptions

```
ATMOS   21    INTERP-1
    ALT        TEMP       PRESS       RHO            SNDSPD      VISC
      0        508.0      1897.      .0021751        1104.      .000156
  10000        472.3      1294.      .0015961        1065.      .000200
  20000        436.7      857.3      .0011435        1024.      .000261
  30000        401.2      548.6      .0007965        981.4      .000347
  40000        390.0      340.6      .0005087        967.6      .000510
  50000        390.0      211.1      .0003152        967.6      .000822
  60000        390.0      130.9      .0001955        967.6      .001325
  70000        393.9       81.3      .0001203        972.4      .002301
```

Figure 16: User supplied data used to define an atmosphere.

INTERP-1	linear interpolation
INTERP-2	lagrange polynomial interpolation[15]
INTERP-3	circular interpolation[16]

Table 9: Interpolation Method Keywords

The site measured atmosphere model (type 22) requires a similar table except that only pressure and density are given as functions of altitude (see Figure 17). An exponential function is used to interpolate pressure and density values from the table. A maximum of 15 values can be input. The units in this table are the same as for the user supplied atmosphere.

```
ATMOS   22
    ALT        PRESS      RHO
      0        1897.     .0021751
  20000        857.3     .0011435
  40000        340.6     .0005087
  60000        130.9     .0001955
  80000        50.9      .0000742
```

Figure 17: Example use of site measured data to define an atmosphere.

4.3 CONSTANTS Segment Data Block

The CONSTANTS segment data block is required only if segment constant values (Section 3.3.4) have been used in the aerodynamic, propulsion, or heating table sets active during the segment. This data block allows the user to set the the segment constant values to control the table look up procedure. A warning message will be issued for segment constant values which are referenced in the active table sets but are not set in a CONSTANTS segment data block, and the values will be set to zero.

All of the segment constant values are set with symbol/value pairs where the symbol is the value name. If, for example, the values named CHUTE and SPDBRK have been used in the aerodynamic table set their numeric values must be given with a data block such as

```
CONSTANTS  CHUTE=1  SPDBRK=60
```

which sets the value of CHUTE to 1.0 and SPDBRK to 60. These values can be changed in different segments to control the results from the aerodynamic table.

4.4 DEFINE Data Block

As can be seen from Appendix B a large number of standard output variables are available. Even so, it is impossible to provide all quantities which may be of interest. However, the DEFINE data block can be used to create ***user defined*** output variables which are combinations of the standard variables. A variety of mathematical operations and some conditional logic functions are available to define the new variables. The user may define the required variable itself or its first derivative and initial value and permit TSAP to perform the required integration. Once defined, a user defined variable may be referenced just like a standard output variable: printed in summary tables, plotted, used in a WHEN segment data block, or even referenced in a subsequent DEFINE data block. As many as six such variables may be created per problem, one for each DEFINE data block.

A simple example of a DEFINE data block which defines an output variable for altitude in kilometers is shown below:

```
DEFINE  ALT-KM
   ADD  ALT
   MULT 0.0003048
```

This example illustrates the basic syntax of the data block. The positional parameter which follows the DEFINE keyword, in this case ALT-KM, is the name of the new output variable. The name must be eight characters or less and unique, that is, not one of the standard output variables or a previously created user defined variable. The subsequent lines in the example data block define operations for computing the new variable and might be considered analogous to a simple programming language. In the example each such line contains two positional parameters, the first representing a mathematical operation and the second representing an output variable name or a constant. At the beginning of the computation, the value of the new variable is set to zero. Each subsequent mathematical operation, processed in the order input, can alter this value. A maximum of 20 mathematical operations can be used to define each output variable.

In the example above, the first mathematical operation is to add the value of altitude (ALT) to the initial value (zero). The second mathematical operation is to multiply by the constant 0.0003048 to convert from feet to kilometers. When this operation is complete, the output variable will contain the altitude converted to kilometers. The calculations are made and the value of the user defined variables updated for each integration step.

The math operations available beyond the two used in the example are shown in Table 10 (where *variable* refers to the input variable name or constant and *value* refers to the current value of the user defined output variable). The mathematical operations ADD through MIN operate on both the input variable and the current value so an input variable or constant must be given. The mathematical operations ABS through ZERO operate only on the current value; variable names or constants right of such operations are ignored. CSTO can be used to store a temporary result for use elsewhere in the definition.

ADD	Add the variable to the value
SUB	Subtract the variable from the value
MULT	Multiply the value by the variable
DIV	Divide the value by the variable
IDIV	Divide the variable by the value
EXP	Take the value to the variable power
IEXP	Take the variable to the value power
MAX	Limit the value to less than the variable
MIN	Limit the value to greater than the variable
ABS	Take the absolute value of the value
NEG	Negate the value
SQR	Square the value
SQRT	Take the square root of the value
LN	Take the natural log of the value
LOG	Take the base 10 log of the value
E	Take e to the power of the value
SIN	Take the sine of the value (in degrees)
COS	Take the cosine of the value (in degrees)
TAN	Take the tangent of the value (in degrees)
ASIN	Take the arcsine of the value (in degrees)
ACOS	Take the arccosine of the value (in degrees)
ATAN	Take the arctangent of the value (in degrees)
ZERO	Set the value to zero
CSTO	Save the value under the temporary variable name, then set the value to zero
LT	If the current value is less than the given variable, jump to a given label
GT	If the current value is greater than the given variable, jump to a given label
JMP	Jump to a given label
LABL	Define a label
END	End the calculations for this user defined variable

Table 10: Math Operations Available for the DEFINE Data Block

The remaining functions are associated with conditional logic and branching. Labels are strings of four or less characters placed before any math operation. (Data block names and math operations may not be used as labels.) The following equivalent examples illustrate the syntax of some of the branching operations.

```
DEFINE     XVAL
        ADD   ALT
        LT 100000.  GOTO  LB1
        ZERO
        ADD   DYNPRS
        JMP   LB2
   LB1  ZERO
   LB2  LABL
        END
```

```
DEFINE     XVAL
        ADD   ALT
        LT 100000.  GOTO  LB1
        ZERO
        ADD   DYNPRS
        END
   LB1  ZERO
```

The example variable XVAL is zero unless altitude is greater than 100000 ft, in which case XVAL is the dynamic pressure. Note that END as the final operation of the first example is superfluous since the beginning of a new data block would have accomplished the same function. The second example performs the same function with fewer operations; the END statement terminates calculations in the primary branch, as does the end of the data block in the other branch.

In the integral form of the DEFINE data block the math operations combine to define the first derivative of the resulting output variable. This form is specified by following the variable name on the first line of the data block with the keyword INTEGRAL. Following this the additional keyword, IC may be matched with the initial value of the variable (default is zero). An example of this form of the data block is shown below.

```
DEFINE     INTLOAD   INTEGRAL IC=1.5
           ADD  NTOTAL
           LT   2.3     GOTO BLOW
           SUB  2.3
           END
      BLOW ZERO
```

Here, total specific loadings in excess of 2.3 g will be integrated and made available via the variable INTLOAD. The initial value of the variable is 1.5 g-sec.

In addition to the standard output variables listed in Appendix B, DEFINE data blocks may reference variables previously defined in other DEFINE and SUMMARIZE data blocks. However, if an integral user defined variable is referenced by another integral user defined variable, the first derivative value will be passed to the referencing variable.

4.5 DWN/CRS Data Block

At times the vehicle position over the ellipsoid surface relative to a given reference point and azimuth in terms of downrange and crossrange is of interest. The standard output variables DWNRNG and CRSRNG are computed when the DWN/CRS data block is present in the problem file. The reference position and azimuth may be specified by pairing values to the keywords defined in Table 11. An example use of the DWN/CRS data block is as follows.

```
DWN/CRS
   LONG=-121.   LATGD=33.   PSIGD=195.0
```

Keyword	Units	Description
LONG	deg	longitude of reference point
LATGD	deg	geodetic latitude of reference point
PSIGD	deg	geodetic reference azimuth

Table 11: DWN/CRS Data Block Keywords

When not set explicitly with the keywords LONG, LATGD, and PSIGD, the reference position and azimuth values are identical to the geodetic values of the initial conditions specified via the INITIAL data block. The standard output variables NORTH and EAST are equivalent to DWNRNG and CRSRNG for an azimuth of zero. The reference point is also used in the computation of TRGNORTH and TRGEAST, which describe the target position in a manner corresponding to the NORTH and EAST variables.

The somewhat complex DWNRNG and CRSRNG calculations are skipped when the DWN/CRS data block is not present in a problem. The distinction between the standard output variables RANGE, DWNRNG, and CRSRNG is significant. As noted in Section 2.2.2, RANGE is the length of the trace of the vehicle path over the surface of the reference ellipsoid. By doubling back on itself or extending for several earth orbits, such a trace could conceivably exceed the circumference of the earth. In contrast DWNRNG and CRSRNG each define the length of a geodesic, which is the shortest arc over an ellipsoid surface between points on that surface. In this case three points are involved: the reference point R, the point S on the ellipsoid surface closest to the vehicle, and an intermediate point Q which lies on the arc beginning at the reference point and extending in the reference azimuth direction for the distance DWNRNG. The point Q is further defined as the point for which the azimuth back to point R varies from that of the azimuth to point S by 90°. CRSRNG is the length of the arc over the ellipsoid surface from S to Q.

4.6 EARTH Data Block

The EARTH data block may be used to alter the default earth shape and gravity models. The input symbols are described in Table 12 along with their default values. Earth shape parameters entered through this data block will be checked for consistency. In particular, REQTR must be greater than zero and only one of FLAT, ECC, or RPOLAR need be specified (since they are redundant). Furthermore, FLAT and ECC must be between zero and one, and RPOLAR must not be greater than REQTR (see Section 2.1.4). Either set of values shown in Table 12 may be activated via the keywords WGS-84 (default) or WGS-72, as appropriate.[6,7] For both earth model standards the published values of REQTR, GM (in metric units), FLAT and the zonal harmonic coefficients are assumed exact and used to compute the English unit equivalents and the related terms to the maximum number of digits. An example use of the data block which specifies the WGS-72 set of values follows.

```
EARTH  WGS-72
```

Individual values of the set can be changed independently. An example use of the data block in which the earth is modeled as non-rotating and spherical is shown below. Parameters not specified will conform to the default WGS-84 values. Refer to Section 2.3.3 for an explanation of the gravity terms: note that REQTR ($R_\oplus$), GM, J2, J3, and J4 are included in the gravity model.

```
EARTH  ECC=0.0  OMEGA=0.0   J2=0.0   J3=0.0   J4=0.0
```

Symbol	Units	Description	Default WGS-84 Value	WGS-72 Value
REQTR	ft	equatorial earth radius	20925646.3255	20925639.7638
RPOLAR	ft	polar earth radius	20855486.5953	20855480.7087
ECC		earth first eccentricity	0.0818191908426	0.0818188106627
FLAT		earth flatness parameter	1/298.257223563	1/298.26
OMEGA	rad/sec	earth rotation rate	7.292115×10^{-5}	$7.292115147 \times 10^{-5}$
G		lbm to slug conversion	32.174	32.174
GM	ft^3/sec^2	earth gravity constant	$1.40764438125 \times 10^{16}$	$1.40764544069 \times 10^{16}$
J2		degree 2 zonal harmonic	$1.08262998905 \times 10^{-3}$	$1.08261579002 \times 10^{-3}$
J3		degree 3 zonal harmonic	$-2.532153068 \times 10^{-6}$	$-2.538810043 \times 10^{-6}$
J4		degree 4 zonal harmonic	$-1.610987610 \times 10^{-6}$	$-1.655970000 \times 10^{-6}$

Table 12: EARTH Data Block Keywords and Default Values

4.7 EXECUTE Data Block

The EXECUTE data block signals the end of problem file data for a given path and may also indicate the segment with which execution is to begin. For example, the line

```
EXECUTE  5
```

will cause path integration to begin with segment five. If no number is given, the first segment is assumed to be number one. Surveys of this value are permitted. Doing so in conjuction with the RESET or INCREMENT Segment data blocks will permit a series of starting conditions to be surveyed. Data blocks given after an EXECUTE data block but prior to the next problem or END lines will modify the current problem. An EXECUTE data block following such blocks will cause the modified problem to be executed.

4.8 FLY Segment Data Block

FLY segment data blocks are used to describe the vehicle attitude during each segment. The specification of attitude angles can be done directly or indirectly; for the latter case, an iterative search is required. The body attitude may be described directly in relation to the surrounding air mass with aerodynamic angles or through use of the conventional Euler yaw, pitch, roll rotational sequence relative to inertially fixed, local geocentric, or local geodetic reference systems. The vehicle attitude angles which may be specified directly are defined in Table 13 and in Sections 2.1.8 and 2.1.10.

A set of three angles from a consistent set are required to fully describe the vehicle attitude. The angles defined in Table 13 can be combined to form the nine sets in Table 14. Normally, each segment includes three FLY data blocks to account for each angle in a set. Only one is required; however, the remaining angles of the set will default to zero. The guidance policies chosen by the user are checked to ensure they represent a consistent set. In addition, it should be noted that ALFA and BETA can be incorrectly chosen such that a physically unrealizable combination is requested. Specifically, ALFA and BETA can be considered valid only when $\cos\alpha\cos\beta \geq 0.0$. Hence, situations which require robust angle

Keyword	Symbol	Description	Angle Between		
ALFA	α	angle of attack	$\hat{x}_b$	and	${}^w\vec{V}\cdot(\hat{x}_b\hat{x}_b+\hat{z}_b\hat{z}_b)$
ALFAT	α_T	total angle of attack	$\hat{x}_b$	and	${}^w\vec{V}$
BANKGC	μ_{gc}	geocentric bank angle	$\hat{y}_w$	and	${}^w\vec{V}\times\hat{z}_{gc}\ (-)$
BANKGD	μ_{gc}	geodetic bank angle	$\hat{y}_w$	and	${}^w\vec{V}\times\hat{z}_{gd}\ (-)$
BETA	β	sideslip angle	$\hat{x}_b$	and	${}^w\vec{V}\cdot(\hat{x}_b\hat{x}_b+\hat{y}_b\hat{y}_b)$
BETAE	β_E	Euler sideslip angle	${}^w\vec{V}$	and	${}^w\vec{V}\cdot(\hat{x}_b\hat{x}_b+\hat{z}_b\hat{z}_b)$
PHI	ϕ_w	windward meridian	$\hat{y}_b$	and	${}^w\vec{V}\cdot(\hat{y}_b\hat{y}_b+\hat{z}_b\hat{z}_b)$
PITCHGC	Θ_{gc}	vehicle geocentric pitch angle	$\hat{x}_b$	and	$\hat{x}_b\cdot(\hat{x}_{gc}\hat{x}_{gc}+\hat{y}_{gc}\hat{y}_{gc})$
PITCHGD	Θ_{gd}	vehicle geodetic pitch angle	$\hat{x}_b$	and	$\hat{x}_b\cdot(\hat{x}_{gd}\hat{x}_{gd}+\hat{y}_{gd}\hat{y}_{gd})$
PITCHI	Θ_I	vehicle pitch angle w.r.t. the inertial platform axis system	$\hat{x}_b$	and	$\hat{x}_b\cdot(\hat{x}_p\hat{x}_p+\hat{y}_p\hat{y}_p)$
ROLLGC	Φ_{gc}	vehicle geocentric roll angle	$\hat{y}_b$	and	$\hat{x}_b\cdot(\hat{x}_{gc}\hat{y}_{gc}-\hat{y}_{gc}\hat{x}_{gc})$
ROLLGD	Φ_{gd}	vehicle geodetic roll angle	$\hat{y}_b$	and	$\hat{x}_b\cdot(\hat{x}_{gd}\hat{y}_{gd}-\hat{y}_{gd}\hat{x}_{gd})$
ROLLI	Φ_I	vehicle roll angle w.r.t. the inertial platform axis system	$\hat{y}_b$	and	$\hat{x}_b\cdot(\hat{x}_p\hat{y}_p-\hat{y}_p\hat{x}_p)$
YAWGC	Ψ_{gc}	vehicle geocentric yaw angle	$\hat{x}_{gc}$	and	$\hat{x}_b\cdot(\hat{x}_{gc}\hat{x}_{gc}+\hat{y}_{gc}\hat{y}_{gc})$
YAWGD	Ψ_{gd}	vehicle geodetic yaw angle	$\hat{x}_{gd}$	and	$\hat{x}_b\cdot(\hat{x}_{gd}\hat{x}_{gd}+\hat{y}_{gd}\hat{y}_{gd})$
YAWI	Ψ_I	vehicle yaw angle w.r.t. the inertial platform axis system	$\hat{x}_p$	and	$\hat{x}_b\cdot(\hat{x}_p\hat{x}_p+\hat{y}_p\hat{y}_p)$

Table 13: FLY Segment Data Block Vehicle Attitude Angle Keyword Definitions

SET NUMBER	VEHICLE ATTITUDE ANGLES		
1	ALFA	BETAE	BANKGC
2	ALFA	BETA	BANKGC
3	ALFAT	PHI	BANKGC
4	ALFA	BETAE	BANKGD
5	ALFA	BETA	BANKGD
6	ALFAT	PHI	BANKGD
7	YAWGC	PITCHGC	ROLLGC
8	YAWGD	PITCHGD	ROLLGD
9	YAWI	PITCHI	ROLLI

Table 14: Consistent Sets of Vehicle Attitude Angles

definitions should not specify BETA. (BETAE does not share this difficulty.) Note also that although the definition in Section 2.1.10 specifies that the total angle of attack always has a positive value, a negative value for ALFAT specified in the FLY segment data block will be interpreted to mean the windward meridian is 180° from that indicated by the associated PHI.

The angular rate (deg/sec) of these angles may be specified by appending the characters DT to the end of any of the above names. For example, an angle of attack rate could be specified with the line

```
FLY  ALFADT = 0.5
```

Of the three angles required to specify vehicle attitude, not all need be specified directly. Instead, several guidance policies are available which iteratively solve for the body attitude required to meet requested conditions. Normally, one or two of the attitude angles are specified directly as described previously. The remaining FLY data blocks in the segment (maximum of three) may indicate a desired flight condition such as level flight or constant rate of heading change. The *free* attitude angles (those not specified directly) are then varied in an attempt to satisfy the requested flight conditions. The angles which will be varied are those which will complete the lowest numbered guidance policy set from Table 14. The flight conditions which may be requested are given in Table 15.

One known performance degradation of TSAP relative to PMAST involves the robustness of the iterative guidance policies. The more general way in which vehicle attitude is specified required changes in the iteration scheme. In most instances the number of iterations required for convergence has been reduced, but if the requested flight conditions are significantly different from the actual conditions, convergence problems may develop. Hence, convergence of the control angle iterator to achieve the requested guidance policy condition is by no means certain. For example, the user must ensure the free control angle(s) can be expected to provide the control appropriate to the requested guidance scheme(s). Moreover, successful use of iterative guidance policies may require prudent

Keyword	Description
ALT	fly at the given altitude
ALTD	fly at the given altitude rate of change
CL	fly at the given lift coefficient
CS	fly at the given aerodynamic side force coefficient
GAMGC	fly at the given geocentric vertical flight path angle
GAMGCD	fly at the given geocentric vertical flight path angle rate
GAMGD	fly at the given geodetic vertical flight path angle
GAMGDD	fly at the given geodetic vertical flight path angle rate
(L/D)MAX	fly at the maximum lift to drag ratio
NY	fly at the given normal acceleration in the $\hat{y}_b$ direction
NZ	fly at the given normal acceleration in the $\hat{z}_b$ direction
PSIGC	fly at the given geocentric horizontal flight path angle
PSIGCD	fly at the given geocentric horizontal flight path angle rate
PSIGD	fly at the given geodetic horizontal flight path angle
PSIGDD	fly at the given geodetic horizontal flight path angle rate

Table 15: Iterative Guidance Policy Keyword Definitions

use of the LIMITS data block (Section 4.15). This can prevent the iterative scheme from straying too far from *reasonable* control angles. When no LIMITS data block is defined, segments using iterative controls will have a default limit on ALFAT of 45°. Future development of TSAP may include a more robust iteration scheme for the guidance policies. When this occurs, the methods used will also be documented.

The format of the FLY segment data block when requesting an iterative guidance policy is identical to that for specifying attitude angles directly. The appropriate keywords from the above list are selected and paired with the desired value. (The (L/D)MAX policy, however, needs no value.) In the example which follows the free attitude angles ALFA and BANKGC will be varied to fly at the maximum lift to drag ratio while at a heading rate of 0.21°/sec. Such a scheme might be typical for a bank to turn vehicle.

```
SEGMENT  28  MAINTAIN CONSTANT HEADING RATE, USE MAX L/D
  INTEG  DT=0.1  DTPRNT=1.0  DTPLOT=0.5
  FLY   BETAE  = 0
  FLY   PSIGDD = 0.21
  FLY   (L/D)MAX
  WHEN  ALT    = 10000   GO TO 81
```

ALT	geodetic altitude (ft)
ALTD	geodetic altitude rate (ft/sec)
GAMGC	local geocentric vertical flight path angle (deg)
GAMGD	local geodetic vertical flight path angle (deg)
LATGC	geocentric latitude (deg)
LATGCD	rate of change of geocentric latitude (deg/sec)
LATGD	geodetic latitude (deg)
LATGDD	rate of change of geodetic latitude (deg/sec)
LONG	longitude (deg)
LONGD	rate of change of longitude (deg/sec)
LREF	aerodynamic reference length (ft)
MASS	vehicle mass (slugs)
PATHLENG	trajectory pathlength (ft)
PATHLMAR	trajectory pathlength since last mark (ft)
PATHLSEG	trajectory pathlength for current segment (ft)
PRESS	freestream atmospheric pressure (lb/ft^2)
PSIGC	geocentric horizontal flight path angle (deg)
PSIGD	geodetic horizontal flight path angle (deg)
RANGE	length of the flight path projected onto the ellipsoid (nm)
RCM	radial distance from earth's center to the vehicle (ft)
RCMD	rate of change of RCM (ft/sec)
RHO	freestream atmospheric density ($slugs/ft^3$)
RNGMARK	range since last mark (nm)
RNGSEG	range during current segment (nm)
SEGMENT	current segment number
SNDSPD	freestream speed of sound (ft/sec)
SREF	aerodynamic reference area (ft^2)
TEMP	freestream atmospheric temperature (deg Rankine)
TIME	path time (sec)
TMARK	time since last mark (sec)
TSEG	time of current segment (sec)
TTARG	target mission time (sec)
VEL	earth relative velocity magnitude (ft/sec)
VGROUND	ground speed (ft/sec)
VISC	freestream kinematic viscosity (ft^2/sec)
WT	vehicle mass (weight) (lb_m)
XCG	center of gravity position in vehicle
XECF	ECFC position X component (ft)
XECFD	ECFC velocity X component (ft/sec)
YECF	ECFC position Y component (ft)
YECFD	ECFC velocity Y component (ft/sec)
ZECF	ECFC position Z component (ft)
ZECFD	ECFC velocity Z component (ft/sec)

Table 16: Type 1 Independent Variables: Available for FLY, PROP, and WIND Tables

DYNPRS	dynamic pressure (lb/ft^2)
MACH	Mach number
REYPFT	Reynolds number per foot (ft^{-1})
VAIR	vehicle speed with respect to the air mass (ft/sec)
WINDD	downward component of wind velocity (ft/sec)
WINDE	eastward component of wind velocity (ft/sec)
WINDN	northward component of wind velocity (ft/sec)

Table 17: Type 2 Independent Variables: Available for FLY and PROP Tables

The value associated with the attitude angles or iterative guidance policies may be interpolated from a table if a constant value is not appropriate. An example of the FLY data block in tabular form follows.

```
FLY   PITCHGD  VRS  ALT  INTERP-2
        90.0           0.0
        90.0        1000.0
        70.0       10000.0
        30.0       90000.0
        20.0      400000.0
```

The guidance policy name follows the data block name as before, but the keyword VRS indicates that the data block is in tabular form and an independent variable name will follow. The independent variable may be any of those listed in Tables 16 and 17, a subset of the standard output variables which are recognized in the table input file. Up to 25 pairs of corresponding tabular data is placed on the lines which follow, one pair per line. Any of the three interpolation methods listed in Table 9 may be selected by placing the corresponding keyword as the last parameter on the first line (INTERP-3 is the default).

An asterisk (*) may be used in the FLY segment data block wherever a value might otherwise be found. Upon execution, the associated parameter will be assigned the value it had at the end of the previous segment. In the example which follows, angle of attack is cycled from its value at segment initiation to 10°, to −10°, and back to its original value. Meanwhile, the Euler type sideslip angle will be adjusted to try to maintain the current heading rate.

```
SEGMENT 81   MAINTAIN TURN WHILE "JINKING"
  INTEG  DT=0.1  DTPRNT=1.0  DTPLOT=0.5
  FLY   ALFA  VRS  TIME
          *          *
         10        100
        -10        105
          *        110
  FLY   PSIGDD = *
  FLY   BANKGD = 0
  WHEN  TIME = 110  GO TO 27
```

4.9 HEATING Data Block and Segment Data Block

The HEATING data block and segment data block may be used to indicate which heating table set is to be used in computing the heating output variables QWALL-1, QWALL-2, and QWALL-3. Both are optional. If absent, the heating output variables are set to zero. When TSAP initializes a trajectory path, it searches the table file for the table set requested in the HEATING data block and loads it into the program. Similarly, as each segment is initialized any table sets indicated by HEATING segment data blocks are loaded overriding those loaded previously.

Although created for defining heating calculations, HEATING data blocks and the associated table sets might be more appropriately thought of as an additional method for the user to define special output variables. That is, since the computation of QWALL-1, QWALL-2, and QWALL-3 is completely defined by the user, the interpretation of the resulting values is also up to the user. The advantage of using the heating table approach is that the number of available operations is much greater than with either the DEFINE or SUMMARIZE data blocks.

The format for both blocks is identical. Two positional parameters are required following the HEATING keyword. The first parameter is the word TABLE, and the second parameter is the table set name. To specify that the heating table set identified with (NOSETIP) in the table file is to be used, the following data block could be entered.

```
HEATING  TABLE  NOSETIP
```

The corresponding heating file might be as follows.

```
(NOSETIP)
HEATING             APPROXIMATE NOSETIP HEATING
SECTION   QWALL-1
*
*     EMPIRICAL FUNCTION WHICH APPROXIMATES NOSE TIP HEATING
*     ON A GIVEN REENTRY VEHCILE
*
      RHO          ADD
      7124.5147    MULT
                   SQRT
      17600.       MULT
      F1      CSTO
      VEL          ADD
      26000.       DIV
      3.15         EXP
      F1           MULT
              END
```

4.10 INCREMENT Segment Data Block

The INCREMENT segment data block can be used to alter values of the time variables and the integrated state vector at the beginning of any segment. Unlike the RESET segment data block, which sets the specified variable to the given value, INCREMENT adds the given value to the current value of the variable indicated. If both data blocks are present in a segment, the reset function will be performed first followed by the increment function. Together, these data blocks replace the DELTA segment data block used in PMAST (although the DELTA block may still be used; see reference 1 for its format).

The RESET and INCREMENT data blocks change the integrated state vector by referencing the output variable names which describe it. Any of the redundant variable

Vehicle Position State Variables			
Geocentric	Geodetic	ECFC	ECIC
RCM	ALT	XECF	XECI
LONG	LONG	YECF	YECI
LATGC	LATGD	ZECF	ZECI
Vehicle Velocity State Variables			
Geocentric	Geodetic	ECFC	ECIC
VEL	VEL	XECFD	XECID
GAMGC	GAMGD	YECFD	YECID
PSIGC	PSIGD	ZECFD	ZECID
Target Position State Variables			
Geocentric	Geodetic	ECFC	
TRGRCM	TRGALT	TRGX	
TRGLONG	TRGLONG	TRGY	
TRGLATGC	TRGLATGD	TRGZ	
Target Velocity State Variables			
Geocentric	Geodetic	ECFC	
TRGVEL	TRGVEL	TRGXD	
TRGGAMGC	TRGGAMGD	TRGYD	
TRGPSIGC	TRGPSIGD	TRGZD	
Vehicle Mass Variables			
	MASS	WT	
Independent Variables (Any Combination Permitted)			
PATHLENG	RANGE	TIME	TTARG
PATHLSEG	RNGSEG	TSEG	
PATHLMAR	RNGMARK	TMARK	FUEL

Table 18: Variables Which may be Changed via the RESET and INCREMENT Blocks

names for position or velocity (in either ECFC, ECIC, Geocentric, or Geodetic coordinate systems) may be used. Within a segment, however, the selection of names for either block must be consistent (though the RESET names need not be consistent with the INCREMENT names). For example, if the position variable ALT is to be changed, LONG and LATGD may also be listed but ZECF may not. Table 18 lists the valid variable names for use in the RESET and INCREMENT data blocks with consistent names listed in columns for each variable type. An example use of the RESET and INCREMENT segment data blocks follows. Note the sort of parameter combinations which are allowed.

```
SEGMENT 17   ADJUST VEHICLE STATE
  RESET        ALT=1.E6    WT=7545.3   TMARK=0  FUEL=0.0
               XECFD=15000 YECFD=14000 ZECFD=4300
  INCREMENT    LATGC=SURV-1   VEL=SURV-2
  PROP  TABLE STAGE_3
  .
  .  (etc.)
  .
```

The time output variable TSEG is normally set to zero at the beginning of each segment, and hence becomes the time since segment initiation. Similarly the pathlength variable PATHLSEG and the range variable RNGSEG are set to zero and become the respective values over the current segment. Alternately, the user may set these values as desired using the RESET and INCREMENT data blocks. Note that use of the INCREMENT block will sum the associated values with zero (or with the value from the RESET block also entered for the current segment).

The variable TMARK is analgous to TSEG but is reset only with the RESET and INCREMENT data blocks. Until reset TMARK counts time from zero at problem initiation, while PATHLMAR and RNGMARK do the same for pathlength and range. If TMARK is reset in a segment with a RESET or INCREMENT block PATHLMAR and RNGMARK are zeroed automatically, then updated if also included in the segment's RESET or INCREMENT blocks. The mark variables can be useful to keep track of of time, pathlength, and range relative to intermediate values.

The term FUEL may be expressed $w_f = -\int \dot{W} dt$ and is normally zero at problem initiation. It may be reset directly, but altering MASS or WT with a RESET or an INCREMENT data block will not alter its value.

4.11 INERTIAL Segment Data Block

The INERTIAL PLATFORM ALIGNMENT segment data block (the PLATFORM and ALIGNMENT keywords are optional) may be used to realign the inertial platform axis system (which remains fixed in inertial space and to which YAWI, PITCHI, and ROLLI are referenced) with any of several specified axis systems defined at the end of the previous segment. In addition, the platform axis may be aligned with any axis system which can be defined by a latitude/longitude pair or by two orthogonal ECFC vectors representing east and down directions.

By default, the inertial platform is aligned with the local geodetic horizon (north, east, down) axis system corresponding to the latitude and longitude specified in the INITIAL data block. This alignment can be superseded at the beginning of any segment by including the INERTIAL segment data block with one of the keywords specified in Table 19. Note, however, that since body attitude information is not available until the first segment is executed, neither the BODY nor WIND options should be used in the first segment to be executed. In the following example the platform is aligned with the local geocentric axis system at the beginning of Segment 1.

```
SEGMENT 1   PITCHOVER AFTER PLATFORM ALIGNMENT
  INTEG  DT=0.2         DTPRNT=1.0
  INERTIAL PLATFORM ALIGNMENT  GEOCENTRIC
  FLY    PITCHIDT = 0.5
  FLY    YAWI    = 247.9
  FLY    ROLLI   =   0.0
  .
  .  (etc.)
  .
```

The platform will be aligned with the vehicle body axis (Section 2.1.8) which existed at the end of the previous segment if the keyword BODY is given. This can be useful when comparing TSAP results with those of other analysis techniques, or when subsequent small

Keyword	Description: realign platform axis to the
BODY	body axis
ECFC	specified east and down vectors
GEOCENTRIC	local geocentric horizon axis
GEODETIC	local geodetic horizon axis (default)
VELOCITY	$\hat{x}_I$ parallel to $^{\oplus}\vec{V}$, $\hat{z}_I$ coplanar with $\hat{z}_{gd}$
WIND	wind axis
LONG LAT	local horizon system at specified coordinates
ATTIME	current conditions indicated by above keywords, but as though they existed at the time specified (default is at current time)

Table 19: Primary Keywords Available within the INERTIAL Segment Data Block

attitude changes are to be related to the current attitude. An example use of the segment data block in this form is

```
INERTIAL  BODY
```

Similarly, the platform may be aligned with the wind axis (Section 2.1.10) which existed at the end of the previous segment by giving the keyword WIND. The alternate keyword VELOCITY aligns $\hat{x}_p$ with the current ${}^{\oplus}\vec{V}$ and makes the orthogonal $\hat{z}_p$ coplanar to ${}^{\oplus}\vec{V}$ and $\hat{z}_{gd}$ such that $\hat{z}_p \cdot \hat{z}_{gd} \geq 0$. As for all other uses of the INERTIAL segment data block, the optional time tag ATTIME may be used to indicate that the given alignment (defined relative to the earth) occurred at the time given (rather than the current time). In the following example the inertial platform is aligned as though the current position and velocity existed at the specified time.

```
INERTIAL  VELOCITY  ATTIME 31.41592654
```

Platform alignment is not restricted to correspond to current vehicle conditions. Latitude and longitude may be specified so as to align the inertial platform with arbitrary local conditions. Independently, the mission relative time at which the platform is aligned may be varied from the current time by specifying a value for ATTIME. In the example which follows, the platform is aligned with the local horizon system (geodetic or geocentric alignment, consistent with the latitude value given) which exists at the given point on the earth's surface when mission time is 100 seconds.

```
INERTIAL PLATFORM   LAT=28.8  LONG=-81.     ATTIME=100.0
```

If ECFC is specified, the inertial platform will be aligned with local north, east, and down at zero longitude, zero latitude unless perpendicular east and/or down vectors are provided. The ECFC component values of these vectors may be specified with the keywords EASTX, EASTY, EASTZ, DOWNX, DOWNY, and DOWNZ. (If only the east vector is specified, it will be assumed the corresponding equatorial down vector will apply. If only the down vector is given, it is assumed the corresponding horizontal east vector is appropriate.) The following example illustrates specification of these vectors.

```
INERTIAL  ECFC   EASTX=1.3  EASTY=3.4  EASTZ=0.0
                 DOWNX=0.0  DOWNY=0.0  DOWNZ=-1.
```

4.12 INITIAL Data Block

The INITIAL data block is used to specify the initial values of that portion of the integrated state vector associated with the vehicle; that is, to specify the initial state (position and velocity) and weight of the vehicle, the initial time, pathlength, range, and the angular rotation Ω_0 of the ECFC coordinate system from the ECIC coordinate system. The default geodetic form for state input can be replaced with other forms by following INITIAL with the keyword GEOCENTRIC, ECFC, or ECIC. The available keywords which may be matched with values (all of which default to zero) are listed in Table 20. Only one of the redundant terms WT or MASS and of VEL or MACH may be specified. In addition, the state keywords used must be consistent with the coordinate system they are associated with. An example use of the data block using geodetic state parameters is

```
INITIAL
  ALT = 100000.     LONG = -92.1096       LAT = 43.03
  VEL = 15000.      GAMA = -5.876         PSI = 70.1
  WT  = 1500        TIME = 103.97      PATHLE = 419458.22
```

To indicate an initial state in terms of Earth Centered and Fixed Cartesian (ECFC)

Keyword	Units	Description	Coordinate System
ALT	ft	altitude	Geodetic
GAMA	deg	vertical flight path angle	Geodetic or Geocentric
LONG	deg	longitude	Geodetic or Geocentric
LAT	deg	latitude	Geodetic or Geocentric
MACH		Mach number	Geodetic or Geocentric
MASS	slugs	mass	
OMEGA0	deg	angle between ECIC and ECFC axis systems when mission time is TEPOCH	
PATHLE	ft	pathlength	
PSI	deg	horizontal flight path angle	Geodetic or Geocentric
RANGE	nm	range	
RCM	ft	distance to center of the earth	Geocentric
TEPOCH	sec	time at which OMEGA0 applies	
TIME	sec	mission time	
VEL	ft/sec	velocity magnitude	Geodetic or Geocentric
WT	lb	mass	
X	ft	position X component	ECFC or ECIC
XDT	ft/sec	velocity X component	ECFC or ECIC
Y	ft	position Y component	ECFC or ECIC
YDT	ft/sec	velocity Y component	ECFC or ECIC
Z	ft	position Z component	ECFC or ECIC
ZDT	ft/sec	velocity Z component	ECFC or ECIC

Table 20: Keywords for the INITIAL Data Block

coordinates or Earth Centered, Inertially fixed Cartesian (ECIC) coordinates, the keywords X, Y, and Z for the position and XDT, YDT, and ZDT for the velocity must be matched with the appropriate values. An example data block of this form might be

```
INITIAL  ECIC
  X=19099345.2   Y=9974312.7      Z=6489105.4
  XDT=-173.978   YDT=10327.524    ZDT=26425.991
  TIME=769.35    TEPOCH=-1045.2   OMEGA0=25.3
  WT=1053.9
```

Note use of the term TEPOCH which specifies a mission relative time at which the ECFC coordinate system is rotated by the angle OMEGA0 from the ECIC coordinate system. Hence, the angular difference between these systems can be expressed

$$\Omega = \Omega_0 + \omega_\oplus(t - t_{epoch}) \tag{94}$$

4.13 INTEG Segment Data Block

The INTEG segment data block may be used to define the size of the integration time step and the interval at which data is output. TSAP uses a fixed step Runge-Kutta fourth order integrator so integration accuracy is dependent upon the user setting an appropriate step size for each segment. Backwards integration is possible by setting the time step DT to a negative value. For maximum accuracy, DT should be at least as small (in absolute value) as the time frame for significant force changes (for example, the time intervals between data points in the propulsion table). Alternately, a large value of DT will reduce computation time. Note, however, that if DTPLOT or DTPRNT is smaller than DT, the integration time step will effectively be the smaller value.

The keywords to which values may be paired are in Table 21. The tabular output logic is such that a data point will be printed at each multiple of DTPRNT and at segment boundaries. Plot data is also saved at each print interval so specifing DTPLOT larger than DTPRNT is meaningless. An example use of the INTEG segment data block follows.

```
INTEG DT=0.01   DTPRNT=5.0  DTPLOT=0.25
```

Keyword	Units	Description	Default
DT	sec	integrator time step	0.05
DTPLOT	sec	maximum interval between data points for PLOT data	DTPRNT
DTGUID	sec	time parameter for iterative guidance schemes	1.0
DTPRNT	sec	maximum interval between tabular output data points	greater of 1.0 or |DT|

Table 21: INTEG Data Block Keywords

4.14 LAUNCH Segment Data Block

A LAUNCH segment data block is available for the simulation of a vehicle launch either from a pad or a rail. When this option is active, acceleration is confined to the positive direction of the vehicle longitudinal axis $\hat{x}_b$ specified by the yaw and pitch of the FLY segment data blocks. (Note: one of the yaw, pitch, roll angle sets *must* be specified directly in the FLY segment data blocks associated with a segment employing the LAUNCH segment data block.) Forces normal to this direction contribute to the frictional force of magnitude $F_f = \mu \parallel \vec{N} \parallel$, where μ is the coefficient of static or sliding friction, as appropriate. A rail launch is simulated using the two example segment data blocks which follow. Splitting the launch phase in this manner will more accurately capture the time of first motion, thus reducing integration error. The keywords for the data block are defined in Table 22.

```
SEGMENT  1   IGNITION TO FIRST MOVEMENT
  INTEG DT=0.01    DTPRNT=0.10
  LAUNCH     CFSTAT=0.15    CFSLID=0.1
  FLY    PITCHGD   =  86.0
  FLY    YAWGD     =  00.0
  FLY    ROLLGD    =  00.0
  WHEN   VEL       >  00.001       GO TO 2
*
SEGMENT  2   TO END OF LAUNCH RAIL
  INTEG DT=0.01    DTPRNT=0.10
  LAUNCH     CFSTAT=0.15    CFSLID=0.1
  FLY    PITCHGD   =  86.0
  FLY    YAWGD     =  00.0
  FLY    ROLLGD    =  00.0
  WHEN  PATHLENG   =  19.958    GO TO 5
```

It should be noted that acceleration, not velocity, is held to the launch rail direction. This is appropriate for most launch situations and will also permit acceleration constraints under other conditions. In segments containing the LAUNCH segment data block acceleration is initially computed as $^{\oplus}\vec{a}$ in the manner described in Equation 43, then adjusted to account for the rail and friction effects as follows.

$$^{\oplus}\vec{a}_L = \left(^{\oplus}\vec{a} \cdot \hat{x}_b - \mu\sqrt{(^{\oplus}\vec{a} \cdot \hat{y}_b)^2 + (^{\oplus}\vec{a} \cdot \hat{z}_b)^2} \right) \hat{x}_b \qquad (95)$$

Keyword	Description	Default Value
CFSTAT	coefficient of static friction	0.0
CFSLID	coefficient of sliding friction	0.0

Table 22: LAUNCH Data Block Keywords

4.15 LIMITS Data Block

The LIMITS data block may be used to limit the MAX or MIN of any of the vehicle attitude angles obtained in segments employing one or more or the iterative guidance policies (see Tables 13 and 15 in Section 4.8). The default limit in effect for such segments constrains ALFAT to less than 45°. The default limit will be superseded by any limits specified in a LIMITS data block. A limit specification includes the angle to be constrained, the type of constraint (MAX or MIN), and a constraining value. A typical use of the LIMITS data block might be

```
LIMITS  PITCHGD MIN = 30.0       MAX YAWGD = 87
        ALFAT   MAX = 10.0
```

Note that the angle name may either precede or follow the MIN or MAX keyword.

For segments which do not employ iterative guidance policies no attitude angle limits are imposed. Hence, the user is free to set the vehicle attitude as desired by directly specifying three control angles as described in Section 4.8. However, in segments for which iterative guidance policies are in effect the directly specifed control angles may be changed in an effort to satisfy the specified limits.

4.16 NOPRINT Data Block

The standard printout file for a problem consists of as many as six parts: (1) a verbatim listing of the input problem file; (2) a summary of the table sets read from the table file; (3) an echoing of the input problem data as read by the TSAP input processor; (4) a history of one or more iteration processes which may occur; (5) a printout of the computed trajectory giving the time, state, derivatives and forces at each print interval in block format; and (6) a printout of optional summary information (see Sections 4.18 and 4.25). The NOPRINT data block may be used to turn off input echoing and/or to eliminate the block trajectory output.

Unless specifically requested in an OUTPUT data block (Section 4.18), TSAP will not echo the input data. Once activated, input echoing can be canceled at any point in the process of reading the problem input file with a line beginning with NOPRINT followed by any or all of the keywords INPUT, ECHO, or ALL. In this way the OUTPUT and NOPRINT data blocks may be paired to bracket portions of the input file for which the echo function is desired.

The OUTPUT and NOPRINT data blocks also relate to the printing of trajectory data in block format. Block output of trajectory data can be quite voluminous and tedious to read so by default will be deleted by the request of one or more summary tables in an OUTPUT data block. It can also be deleted using the NOPRINT data block either alone on a line in the problem input file or with one or more of the keywords BLOCK, TRAJECTORY, or ALL. These keywords may be combined with those for input echoing so as to control both functions on one line. An example of such a line is

```
NOPRINT  INPUT ECHO  BLOCK
```

Note that the output of trajectory information can be totally suppressed so it is advisable to use this data block with caution.

4.17 OPTIMIZE Data Block

The trajectory optimization capability of TSAP is based on the recursive quadratic programming (RQP) alogithm developed by Powell and the associated software known as VF02AD.[17,2] As many as 25 parameters in the problem input file may be varied to meet trajectory conditions (constraints) and minimize or maximize an objective function (performance index). The parameters are specified in the problem file via keywords in a manner similar to that used to define survey and search values. The performance index, constraints, initial parameter values, and iteration control data are specified through the OPTIMIZE data block.

A TSAP problem may contain up to three optimization subproblems, or loops, labeled A, B, and C. The segments integrated for each must be distinct; that is, the iteration loops may not overlap or be nested. Experience tends to suggest that most well formulated problems can be handled by RQP techniques in a single loop. Although search loops (see Section 4.23) may contain nested optimization loops (the converse is not true), it is almost always more efficient to include the function of the search loop as part of the optimization problem. In contrast, surveys used in conjuction with optimization problems are often useful. By surveying constraint values one can generate an array of performance limits. The initial guess of parameter values input in the problem deck is used only for the first survey value; successive survey passes use the optimization results of the previous path to initialize the next optimization parameter set. This will often reduce the number of iterations required.

Example portions of a problem file defining an optimization problem are given in Figures 18 and 19. (The lowercase notes are comments not allowed in an actual input file.) These examples contain all the basic components used to specify an optimization problem and also exercises many of the more advanced features available for use. The remainder of this section describes what the keywords associated with parameter optimization mean.

Perhaps the most important component leading to a successful solution of a trajectory optimization problem is proper problem formulation. Only by carefully defining what must be optimized, what parameters are available to do so and what the constraints on the trajectory are can one expect a relevant solution. For example, some problems require the use of the integral type of user-defined variables. Subtle areas of difficulty include over-constrained problems, discontinuities in the solution field and scaling problems. Each of these topics is treated at some length in reference 2.

4.17.1 Parameter Definition

To define the parameters, keywords of the form *OPTx-k* are placed in the problem file where a value might otherwise go. Here *x* is one of A, B, or C, indicating which of the three optimization loops the parameter belongs to; and *k* indexes an individual parameter within

```
SEGMENT  5  TURN
  INTEG  DTPRNT=1  DT=.5
  AERO TABLE RVAERO  SREF=2.5
  FLY  ALFA  VRS  TSEG  INTERP-1       (recommend this interpolation choice)
       OPTA-1        0
       OPTA-2       40
       OPTA-3       60
       OPTA-4      100
       OPTA-5      OPTA-11             (node times will be scaled)
  FLY  BETA=0
  FLY  BANKGC  VRS  TSEG  INTERP-1
       OPTA-6        0
       OPTA-7       20
       OPTA-8       40
       OPTA-9      100
       OPTA-10     OPTA-11
  WHEN TSEG    =   OPTA-11  GOTO 8
*
SEGMENT  8  PUSHOVER, THEN FLY BALLISTIC UNTIL IMPACT
  INTEG  DTPRNT=1  DT=.5
  AERO TABLE RVAERO  SREF=2.5
  FLY  ALFA  VRS  TSEG  INTERP-1
       0.0          0.0
       OPTA-13      1.0
       OPTA-14     10.0
       0.0         12.0
       0.0         90.0                (node times will not be scaled)
  FLY  BETA   = 0
  FLY  BANKGC = 0
  WHEN TSEG   = OPTA-12  STOP
```

Figure 18: Example portion of an input file defining optimization problem parameters.

the loop. (If n parameters are to be used, they must be indexed 1 through n, inclusive.) The *OPTx-k* keyword(s) may not appear in SEARCH, SUMMARIZE, or SURVEY data blocks and must precede the OPTIMIZE data block. Otherwise, any input value which has meaning as an input to a real-valued function may be defined as a parameter. A parameter definition may appear more than once in the problem thereby slaving the values of two or more input variables together. (Note, for example, OPTA-11 in Figure 18.)

```
OPTIMIZE A FOR VEL=MAX ON SEGMENT 8       (optional PRINT at end of line)
             PARAMETER VALUES             (optional if constraints end block)
  FEST=1000     ACC=.001   MAXFUN=500
*               ALFA PARAMETERS
  PAR-1=1.0     HI-1=15    LO-1=0         (parameter bounds are optional)
  PAR-2=14.0    HI-2=15    LO-2=0
  PAR-3=2.0     HI-3=15    LO-3=0
  PAR-4=10.0    HI-4=15    LO-4=0
  PAR-5=5.0     HI-5=15    LO-5=0
  PAR-13=-10    HI-13=15   LO-13=-15
  PAR-14=-10    HI-14=15   LO-14=-15
*               BANKGC PARAMETERS
  PAR-6=154.    LO-6=0     HI-6=180
  PAR-7=50.     LO-7=0     HI-7=180
  PAR-8=97.     LO-8=0     HI-8=180
  PAR-9=82.     LO-9=0     HI-9=180
  PAR-10=27.    LO-10=0    HI-10=180
*               TIME PARAMETERS
  PAR-11=120.   LO-11=0
  PAR-12=30     HI-12=120  LO-12=5
                   CONSTRAINTS            (switch to reading constraints)
  GAMGD=0     ON SEGMENT 5  SCALE=0.10
  ALT=20000   ON SEGMENT 5                (only 1 constraint spec. per line)
  VEL>4000    ON SEGMENT 5
  ALT=0       SCALE=0.001                 (default segment is 8)
  GAMGD<-30                               (default scale factor is 1)
  CRSRNG=SURV-1                           (note that surveys are permitted)
```

Figure 19: Example OPTIMIZE Data Block

4.17.2 The OPTIMIZE Data Block

The OPTIMIZE data block is used to provide an initial estimate and bounds for the parameters, set iteration control parameters, and to define the performance index and problem constraints. It is placed in the problem file after the segments in which the parameters are defined. The OPTIMIZE keyword is followed by the loop specifier A, B, or C and the word FOR. Next the performance index is specified with the name of an output variable, followed by the MAX or MIN indicator, the words ON SEGMENT, and the segment number at which the performance index is to be evaluated. (As for SEARCH variables, the evaluation is done at the end of the segment.) Figure 19 contains an example of the data block which is consistent with the parameters defined in the example of Figure 18.

The iteration control variables, initial parameter values, and constraints are listed as symbol/value pairs on the lines which follow with the constraints in a group separate from the others. The two groups are separated by lines containing the words PARAMETER VALUES and CONSTRAINTS, respectively. (The former is optional if that group precedes the constraints group.)

The variable FEST is used to provide the optimization algorithm with an order-of-magnitude estimate of $F(\vec{x})$ for scaling purposes. If, for example, maximum velocity is required for a given problem and 2000 ft/sec is a reasonable guess for the result, then FEST=1000 would be appropriate. If not specified, FEST is assigned a value of one.

There are two iteration control variables available for controlling the RQP iteration process. One is ACC, which is a convergence tolerance bound. The recommended range for this variable is 0.01–0.000001; the default value is 0.01. Some simple or well-posed problems may converge quickly to even stricter convergence criteria, while more complex problems may have difficulty achieving convergence to an ACC value which is too small. Hence an appropriate value for ACC is problem dependent.

The number of path integrations performed while iterating within an optimization loop is limited by MAXFUN. If the number of path integrations exceeds the value of MAXFUN after the gradient calculations are complete and the linear search process converges, the iteration process is stopped and the status flag INF (defined below) is set to ten. While the default value of MAXFUN is $10n$, the recommended range extends to $50n$. A value of one will cause TSAP to compute a trajectory only for the initial values of the parameters.

The initial parameter values are set with symbol/value pairs of the form ***PAR-k***=value where the ***k*** corresponds to that in the ***OPTx-k*** used to specify parameter interpretation. Although VF02AD is often capable of converging from even poor initial conditions, it is wise to start with a good estimate. Avoid values of zero so as not to defeat a parameter normalization scheme useful when absolute parameter values are expected to vary widely.

Lower or upper bounds may be placed on the parameter ***k*** using symbol/value pairs of the form ***LO-k***=value or ***HI-k***=value as appropriate. Such bounds are treated as inequality constraints and are highly recommended for all parameters even if not explicitly a part of the problem. Bounding the parameters will prevent the optimization scheme from trying unreasonable values, and hence may reduce computation time. For example, angles can usually be constrained to a 360° range without loss of generality, and segment times can often be given upper and/or lower bounds which will not unduly constrain the solution.

Output variable constraints may be defined following a line containing the CONSTRAINTS keyword. A complete constraint specification includes an output variable name, type of constraint indicator, a constraint reference value, a segment number on which to evaluate the constraint, and a scale factor. One constraint is specified per line so as to group this information. Any of the standard output variables listed in Appendix B may be constrained as may any of the user defined variables discussed in Sections 4.4 and 4.25. The optimization software requires at least one constraint or parameter bound.

Obviously, the number of equality constraints must not exceed the number of parameters, nor should the inequality constraints and parameter bounds overconstrain the problem.

4.17.3 Output of Iteration History Information

During the linear search process the line

```
OPTx:     #  LOOP EVALUATIONS
```

is printed in the status file to indicate the progress in the number of path integrations. The output file contains more complete iteration information. Listed are the values of the parameters, the performance index, and the constraints at the end of each linear search. The VF02AD notation has been retained so these are labeled X =, F =, and C =, respectively. A header serves to identify each of these items. Status flag INF and linear-search step-size information is also printed. An example portion of an output file listing these items is given in Figure 20.

```
        OPTA PARAMETER OPTIMIZATION ITERATION HISTORY:

 X = OPTA- 1              OPTA- 2

 F = ALTD     /    -0.1000000000E+04          ON SEGMENT 50

 C = LONG     SEG 50     LATGD    SEG 50     LO- 1              HI- 1

    ITERATIONS =     1     CALLS OF VF02AD =     1

 X = 0.8608000000E+02   0.6131000000E+02

 F = 0.9764531931E+01

 C = 0.6017428418E-05  -0.2176867497E-03   0.9128888889E+00   0.4355555556E-01
   1 LINEAR SEARCH ITERATION,     STATUS FLAG INF = 0
   LINEAR SEARCH STEP-SIZE: ALPHA =   1.00000

    ITERATIONS =     2     CALLS OF VF02AD =     2

 X = 0.8607280975E+02   0.6116579920E+02

 F = 0.9764431562E+01

 C = 0.5836680894E-07   0.1875220976E-06   0.9127291055E+00   0.4363544727E-01
```

Figure 20: Output File Labels for Optimization Information

4.18 OUTPUT Data Block

The OUTPUT data block may be used to specify the way in which trajectory data is printed. The options include a block format style made up of a fixed set of variables and summary tables in which the output variables for each column of data are chosen by the user. As many as ten summary tables may be defined per problem with the option of putting them in different output files. The OUTPUT data block may also be used to enable echoing of the input data as it is read in.

If no summary tables are requested as described below, then in the absence of a NOPRINT data block (Section 4.16), the trajectory data will appear in the standard printout file in block format. When summary tables are requested, it is assumed block output is not needed but may be obtained by following the data block name with the two keywords BLOCK and TRAJECTORY. Similarly, input data echoing may be activated with the pair of keywords INPUT and ECHO. Use of either set of keywords puts the data block in *switch mode* and precludes definition of a summary table on the same line. An example of the OUTPUT data block which performs both these functions follows.

```
OUTPUT   INPUT ECHO   BLOCK TRAJECTORY
```

When switch mode keyword pairs are not present, entries following the data block name are assumed to be positional parameters. The first parameter designates the file that the output data is to be written on. If the first parameter is PRINT, the trajectory data will go to the standard printout file and column headings and other information will be written at the top of each page. Otherwise, the parameter is the name of the file to which the data will be written, and column headings will be applied only at the beginning of the file.

While block format style output is printed as the trajectory is being integrated, summary tables and output files are created after all trajectory calculations are complete from data stored during the path integration process. If a non-fatal error occurs during the trajectory calculations, summary tables will be printed up to the point of the error which may require TSAP to reintegrate some segments. The only fatal errors known to occur involve the optimization software and such things as disk quota violations. Even so, if summary tables seem an insufficiently robust form of output in the face of unexpected program behavior, it may be useful to activate block format output as explained above.

In a standard summary table values are printed with FORTRAN F-format statements. In each column the format will be such that the number of largest magnitude, when within the range 1.0 to 10^{10}, will be printed with nine significant digits. When the value of largest magnitude in a column is 10^{10} or greater, an E-format which provides five significant digits will be used for that column. Standard F-format summary tables are obtained by omitting the second positional parameter (that is, by default). An example summary table in standard F-format is given in Figure 21.

Additional digits of output can be obtained through use of an optional format in which the values are printed exclusively with FORTRAN E-format statements. Use of this option

TIME	ALT	MACH	LONG	LATGD
280.000000	125000.000	15.0191228	-162.000000	21.4450000
282.000000	115718.134	15.0378484	-162.078905	21.4141233
284.000000	106927.179	15.0878069	-162.157043	21.3835040
286.000000	98662.598	15.0297286	-162.234609	21.3530657
288.000000	90940.581	14.9544974	-162.311704	21.3227710
290.000000	83771.397	14.8809605	-162.388370	21.2926037
292.000000	77163.420	14.7977239	-162.464599	21.2625679
294.000000	71123.641	14.6950211	-162.540323	21.2326919
296.000000	65658.997	14.5618917	-162.615423	21.2030231
298.000000	60775.947	14.3448859	-162.689735	21.1736279
300.000000	56479.233	14.0949328	-162.763068	21.1445839
302.000000	52771.056	13.8102355	-162.835213	21.1159748
304.000000	49650.252	13.4934103	-162.905965	21.0878848
306.000000	47111.555	13.1503580	-162.975139	21.0603895
308.000000	45145.637	12.7888710	-163.042587	21.0335505
310.000000	43739.654	12.4180588	-163.108204	21.0074121
312.000000	42878.182	12.0477902	-163.171936	20.9819988
314.000000	42544.411	11.6869334	-163.233778	20.9573149
314.296906	42538.800	11.6345886	-163.242799	20.9537121

Figure 21: A Summary Table in Standard F-Format

permits 14 significant digits to be printed for each value in exponential notation. E-format summary tables are obtained by using the keyword E-FORMAT as the second positional parameter. An example summary table in E-format is given in Figure 22.

The line or lines which follow the the first line of the OUTPUT data block (which contains the keyword OUTPUT followed by a file name and optionally a file format) may specify one to ten output variables to list in the summary table. Since only five columns of E-format data fit on each line, a request for more than five variables in such a table will result in the data being spread over two lines. Any combination of standard output variables (Appendix B) or variables defined in the DEFINE (Section 4.4) or SUMMARIZE (Section 4.25) data blocks may be used although in the latter two cases the data blocks which define the variables must appear in the problem input file prior to their reference in an OUTPUT data block. When no output variables are provided, the ten default variables TIME, ALT, LONG, LATGD, VEL, GAMGD, PSIGD, ALFA, BETAE, and DYNPRS will be printed. The data block which produced Figure 21 in the printout file is given in the following example.

```
OUTPUT  PRINT
   TIME  ALT  MACH  LONG  LATGD
```

MACH	ALT	RANGE
1.5019122757366E+01	1.2499999997775E+05	0.0000000000000E+00
1.5037848367984E+01	1.1571813420402E+05	4.7871444360742E+00
1.5087806956796E+01	1.0692717897831E+05	9.5295192482761E+00
1.5029728645828E+01	9.8662597473051E+04	1.4239088865258E+01
1.4954497449077E+01	9.0940580169536E+04	1.8921768069537E+01
1.4880960548128E+01	8.3771394764066E+04	2.3580176775247E+01
1.4797723942354E+01	7.7163417643528E+04	2.8213782008095E+01
1.4695021177088E+01	7.1123637452397E+04	3.2818352129949E+01
1.4561891705457E+01	6.5658992106050E+04	3.7386678299624E+01
1.4344885816592E+01	6.0775940992568E+04	4.1908719206947E+01
1.4094932688821E+01	5.6479225478526E+04	4.6372716821594E+01
1.3810235279678E+01	5.2771047139205E+04	5.0766002333387E+01
1.3493409981814E+01	4.9650240849674E+04	5.5075903813957E+01
1.3150357439667E+01	4.7111542771995E+04	5.9291094626418E+01
1.2788870314748E+01	4.5145622423675E+04	6.3402423792109E+01
1.2418057873786E+01	4.3739638211679E+04	6.7403389928868E+01
1.2047789048341E+01	4.2878164219171E+04	7.1290571623540E+01
1.1686931995295E+01	4.2544390286028E+04	7.5063580673343E+01
1.1634585709938E+01	4.2538779421870E+04	7.5614080478160E+01

Figure 22: A Summary Table Produced in E-Format

Data from the same trajectory presented in an E-format summary table was generated by the following data block with the result shown in Figure 22. The data went to the file REENTRY.DAT, as specified following the OUTPUT keyword.

```
OUTPUT  REENTRY.DAT  E-FORMAT
  MACH  ALT  RANGE
```

A special heating output format is also available. It may be activated by setting the second positional parameter on the OUTPUT line to HEATING. The output file (first positional parameter) must be something other than PRINT. The variables TIME, VEL, ALT, and ALFAT will be printed. Any additional variables requested will not be printed. The following is an example of such a data block.

```
OUTPUT  HEAT.DAT  HEATING
```

4.19 PLOT Data Block

TSAP has the capability to generate two-dimensional plots of any output variable versus any other. A special case of such plots is a ground track plot in which the vehicle's flight path is projected onto the earth's surface and (optionally) map features (such as coastlines) are also drawn. All other plots are known as history plots since the pair of values defining each data point on the plot correspond in (mission) time. As many as 20 plots can be generated per problem, one for each PLOT data block listed. The plots are created by the DISSPLA graphics package and placed in an output file in meta-file format for post processing to a variety of different output devices.[3]

Positional parameters on the same line as the PLOT keyword specify the type of plot generated. History plots are produced with data blocks such as

```
PLOT  ALT  VRS  RANGE
```

which generated the plot shown in Figure 23. The first positional parameter gives the y-axis variable (the dependent variable), and the third positional parameter gives the x-axis variable (the independent variable). The parameter VRS is required to indicate a history type of plot. Any two standard output variables or user defined variables can be plotted. If a user defined variable is listed in a PLOT data block, it must have been defined previously with a DEFINE or SUMMARIZE data block. A list of the standard output variables is given in Appendix B. The title specifed in the TITLE data block will be placed on the

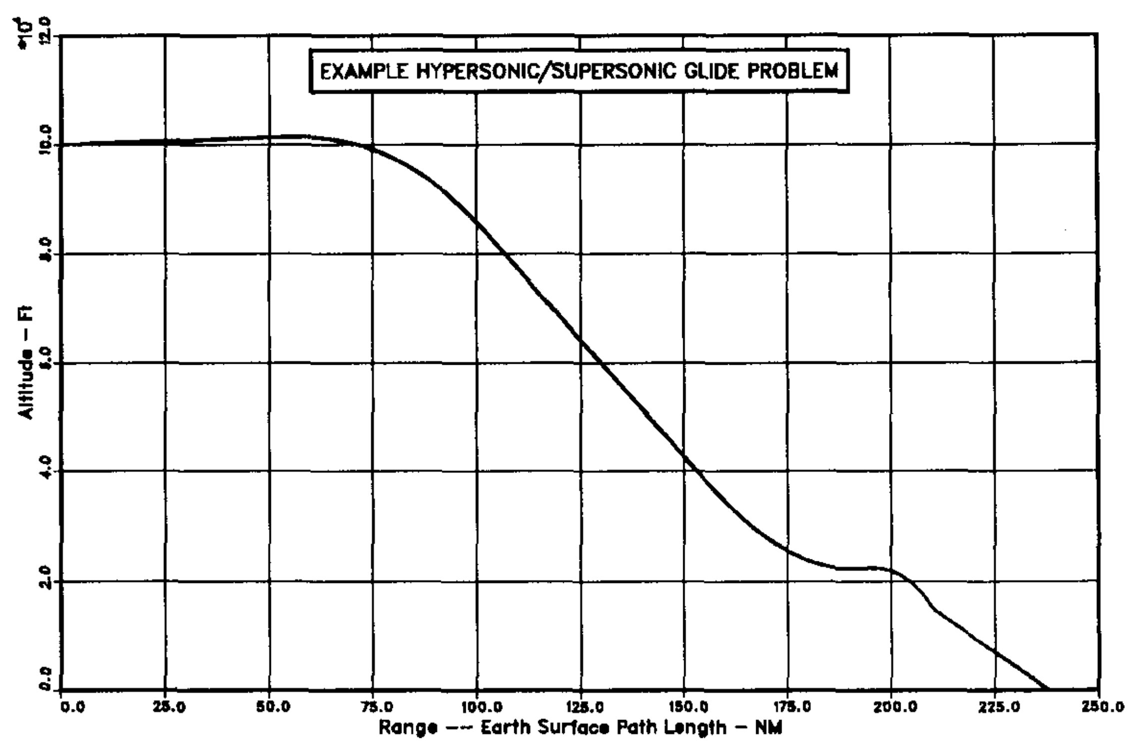

Figure 23: An Example History Plot

Keyword	Description
DX	horizontal axis numbering increment
XLO	lower bound for the horizontal axis
XHI	upper bound for the horizontal axis
XLABEL	label for the horizontal axis
DY	vertical axis numbering increment
YLO	lower bound for the vertical axis
YHI	upper bound for the vertical axis
YLABEL	label for the vertical axis
AUXFILE	name of auxillary data file
MAPFILE	name of DISSPLA map file to be used on ground track
STYLE	lettering style (SIMPLX, DUPLX, or SWISSM)
WIDTH	data trace line thickness (THICK or THIN)

Table 23: Optional Keywords for the PLOT Data Block

upper portion of the plot. Other optional plot parameters may be specified through the keywords given in Table 23.

Ground track plots are essentially plots of latitude versus longitude but use cartographic scaling conventions and may include optional map data. They are produced with the data block

```
PLOT GROUND TRACK
```

One of the optional keywords GEODETIC or GEOCENTRIC may be appended to the end of the above line to specify the type of latitude to be used (geodetic being the default). A Mercator map projection is used so distortion near the poles can be expected. By default, map data from the DISSPLA coastline data file COAST will be included in the ground track plot. The map data can be deleted by pairing the symbol MAPFILE with the keyword NONE. Alternately, other DISSPLA map files (as many as five per plot) may

Keyword	Description	Advertised Resolution (deg)
HERSHEY	world coastlines and lakes	0.5
MAPDTA	world coastlines	0.5
COAST	complete coastlines and major lakes	0.1
POLITICAL	complete political boundaries	0.1
USAHIGH	continental U.S. state boundaries	0.01
NONE	delete map data	-

Table 24: Keywords for DISSPLA map files

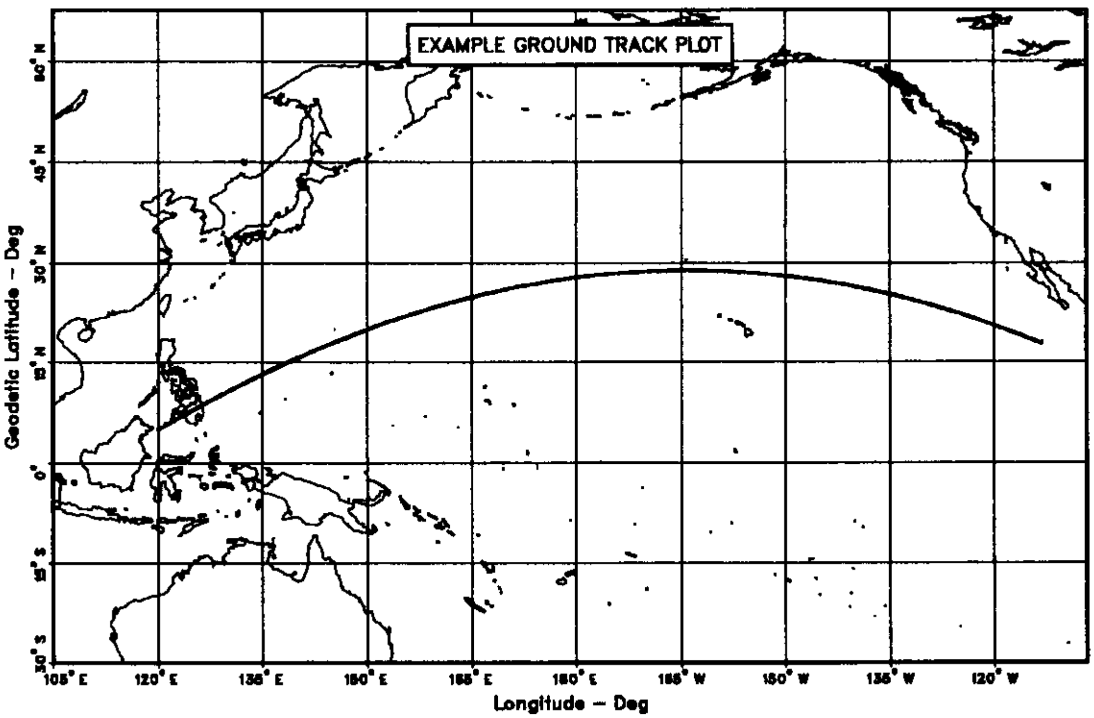

Figure 24: An Example Ground Track Plot

be specified by replacing NONE with the map file name. The most useful of these files are listed in Table 24; more complete information can be found in reference 3. Map data created by the user can be included through an auxillary data file, discussed below.

Normally the plot scales are selected automatically to include the range of values observed during the trajectory; however, fixed scales can be input that override the automatic scaling. This may be desirable to ensure the use of consistent scales. The (horizontal) x-axis scales are input with the symbol/value pairs XLO, XHI, and DX; and the (vertical) y-axis scales are input with the symbol/value pairs YLO, YHI, and DY. The values of DX and DY are checked to determine whether fixed or automatic scales should be used. For instance, if the value of DX is zero (as it is by default), automatic scales are used for the x-axis; if it is non-zero, fixed scales are used. All of the plot scaling parameters default to zero.

An example PLOT data block using fixed scales is

```
PLOT  ALT  VRS  TIME
   XLO=0    XHI=200      DX=50
   YLO=0    YHI=200000   DY=50000
```

where XLO is the starting value of the x-axis, XHI is the last value along the x-axis, DX is the x-axis labeling increment, YLO is the starting value of the y-axis, YHI is the last value of the y-axis, and DY is the y-axis labeling increment. Although both the x and y

SIMPLX DUPLX **SWISSM**

Figure 25: Three lettering styles are available for plots.

axis scales were input in this example, the scales for just one of the axes may be input and the other will be scaled automatically.

Plot scales can be given for ground track plots similarly. For ground track plots the x-axis is always longitude and the y-axis is always latitude. East longitude values are entered positive and west longitude values are entered negative. (Be sure that XLO is less than XHI. If the dateline must be crossed, choose either positive or negative longitude values and extend one bound beyond ±180°.) Similarly, North latitude values are entered positive, and South latitude values are entered negative. An example data block follows.

```
PLOT  GROUND  TRACK
  XLO=-195  XHI=-165  DX=10
  YLO=10    YHI=30    DX=10
```

The standard output variables listed in Appendix B each have an associated label for default use on plots which consists of a descriptive name and a set of units. The default plot labels for user defined variables (Sections 4.4 and 4.25) are simply the variable names. The user may specify alternate labels for each plot through use of the keywords XLABEL and YLABEL. Every entry on a line to the right of the label keyword (other than the delimiter symbols which immediately follow the keyword) are assumed to be part of the desired label. Hence, keyword combinations following XLABEL and YLABEL on a line become part of a label rather than perform their usual function. Note also that plot labels are restricted to being no more than 40 characters in length. In the data block which follows the horizontal axis label is specified as ***TIME SINCE FIRST STAGE IGNITION - SEC*** and the vertical axis label as ***ALTITUDE ABOVE WGS-84 ELLIPSOID - FT***.

```
PLOT ALT VRS TIME  XLABEL = TIME SINCE FIRST STAGE IGNITION - SEC
                   YLABEL = ALTITUDE ABOVE WGS-84 ELLIPSOID - FT
```

The data is usually traced with a thick line to help it stand out. Line thickness can be controlled with the WIDTH symbol by matching it to the keywords THICK or THIN, as desired.

Three character (lettering) styles are available, selected with the keyword STYLE followed by one of the style names SIMPLX, DUPLX, or SWISSM. The default character style is known as the DUPLX style. The SIMPLX style uses fewer vectors to generate characters and so will plot faster than the other styles. Conversely, the SWISSM style produces high quality characters by using many vectors; although slow as output, the SWISSM characters may be more suitable to create plots used for view graph presentations or reports. Examples of all three styles are shown in Figure 25.

Sometimes it is useful to include data from another source on a TSAP plot. This can be done by following the keyword AUXFILE with the name of a file containing the data of interest as the first two entries on each line. That is, the file should contain two columns

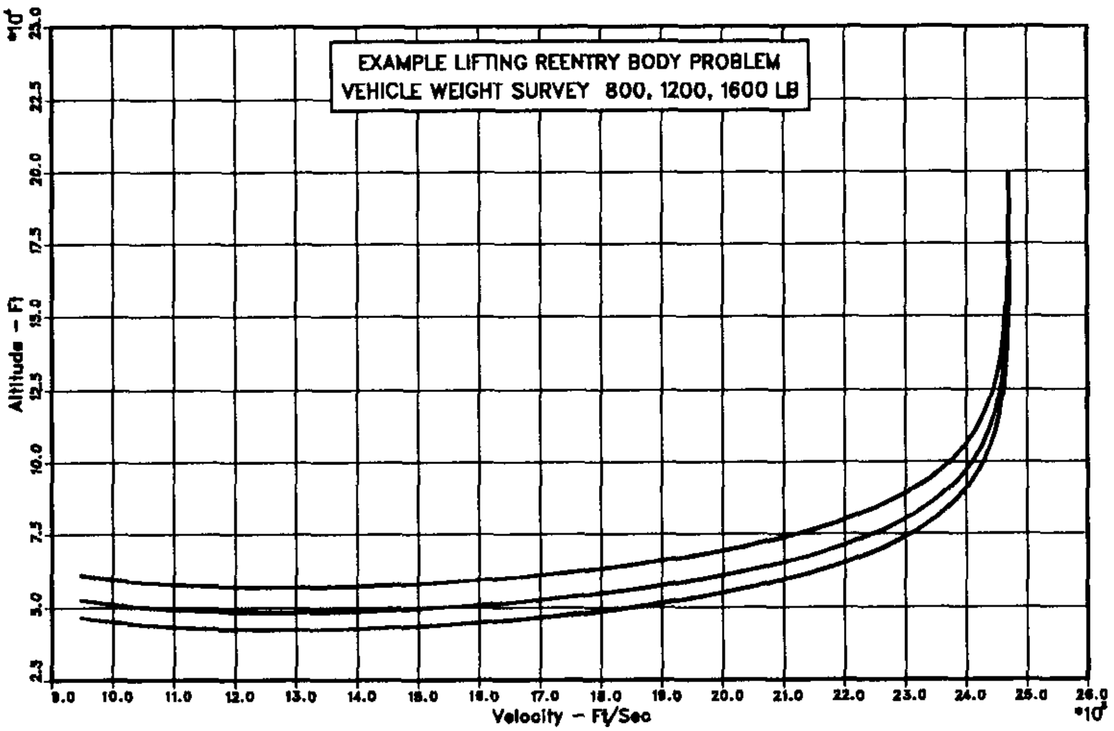

Figure 26: All SURVEY paths appear on each plot.

of data, the first column corresponding to the horizontal axis and the second column the vertical axis. Blank lines and comments are permitted in the file, the latter preceded by an asterisk in column 1. A thin line will be traced through the data points which lie within the plot boundaries; the auxillary data will not affect automatic scaling. Discontinuities in the data over which a line is not desired can be handled by beginning a line with the keyword BREAK. Other character data in the auxillary data file will cause an error message to be generated, but the effect on the plot will be identical to that produced with the BREAK keyword. The following example data block produces a plot with SWISSM characters, with thin lines, and overlays data from the file OLDTRACK.DAT. Map data from the HERSHEY data file is also included.

```
PLOT  GROUND TRACK  MAPFILE=HERSHEY   WIDTH=THIN
   STYLE=SWISSM     AUXFILE=OLDTRACK.DAT
```

Normally a single trajectory is shown on each plot. However, if parametric surveys have been defined with the SURVEY data block (Section 4.26), all the trajectories produced with the surveys are shown on the same plot. This can be used to illustrate the changes in the flight path caused by changing the survey variable. Figure 26 is an example of such a plot.

4.20 PROP Data Block and Segment Data Block

The PROP data block specifies the name of a table set to be interpolated for the propulsive forces and mass flow rates and can also specify the direction relative to the vehicle in which the thrust is applied. When external to a segment, the specified name will be used during the integration of all segments prior to encountering a PROP data block within a segment. The thrust table named inside a SEGMENT data block will be used until a segment is encountered containing a PROP segment data block which names a different table. If no PROP data blocks are entered, the propulsion forces and mass flow rates are zero.

The first part of the PROP data block has the same syntax as the AERO and HEATING data blocks; that is, two positional parameters following the keyword PROP. The first parameter is the word TABLE and the second parameter is the propulsion table set identification name; for instance, the data block

```
PROP   TABLE   PROPDAT2
```

gives PROPDAT2 as the propulsion table set name. An error message would be produced and trajectory integration halted if a propulsion table set with the name PROPDAT2 was not provided in the table input file.

In the example shown above only a propulsion table set name is given. For such cases the propulsive forces are assumed to be aligned with $\hat{x}_b$ (the x-axis of the vehicle body, positive forward). To specify propulsive forces other than along $\hat{x}_b$ a table of thrust vector angles must be given in the PROP data block. The two thrust vector angles, defined in Figure 27, are similar in concept to the total angle of attack and the windward meridian. The angles, called EPS1 and EPS2, may be defined as a function of one of the standard output variables listed in Tables 16 and 17 of Section 4.8.

The thrust vector angle table begins on the line following the PROP keyword. The first line of the table defines the independent state variable, the thrust vector angle (EPS1 and EPS2) column headings, and the interpolation method. The independent variable name

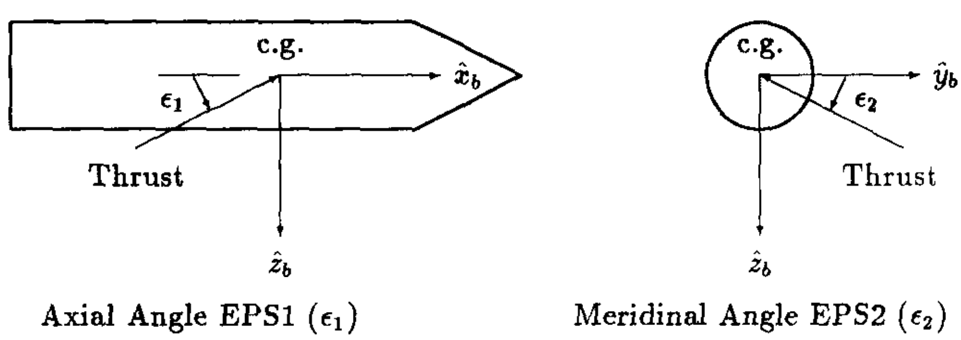

Figure 27: Thrust vector angles are defined relative to the vehicle body.

must be the first variable given as a column heading; the thrust vector angles, EPS1 and EPS2, are given next in any order; and the optional interpolation method is given last. The next set of lines gives the thrust vector angle values as a tabulated function of the independent variable (values of the independent variable must be strictly increasing). For example, the data block shown below defines the thrust vector angles as a function of the flight path time.

```
PROP  TABLE  PROP200
   TIME     EPS1     EPS2    INTERP-3
     0       0.0       0
    20       1.0       0
    25       2.0       0
    27       0.0       0
   100       0.0       0
```

For this example the thrust will be vectored to 1 deg by 20 seconds into the flight, to 2 deg by 25 seconds, and back to 0 deg by 27 seconds. A maximum of 25 data points can be used to define the thrust vector functions.

Three interpolation methods are available for the thrust vector table. The keywords and definitions for each are in Table 9. The interpolation method is input on the second line of the data block as the fourth positional parameter (following the variable names EPS1 and EPS2) as shown in the example above or is assumed to be INTERP-3 by default.

4.21 RADAR Data Block

The RADAR data block may be used to define a radar position which is fixed to the earth. The default station position is that of the initial vehicle conditions, and the default earth shape is identical to that specified in the EARTH data block (or WGS-84 by default). To change the defaults, values can be given to match symbols defined in Table 25. The orientation of the axis system on which the radar azimuth and elevation is computed is based upon ALT, LONG, and LATGD. Hence, adjustments to the position of the origin of the axis system by the offset values DISTN, DISTE, and DISTD (which are zero by default) should be small to maintain the validity of this orientation. Furthermore, these offset distances are cartesian in nature; i.e., they do not curve with the surface of the earth.

The earth shape parameters are available to permit the radar calculations to be based on values which might be standard for a particular radar site, but which may differ from those being used for the trajectory calculations. They affect only the output variables associated with the radar; i.e. RADRNG, RADAZIM, RADELEV, and their derivatives. As for the EARTH data block, the earth shape parameters will be checked for consistency (see Section 2.1.4).

The radar standard output variables are computed only if a radar data block is present in the problem file. Available are the slant range, azimuth, and elevation to the vehicle (RADRNG in nm, RADAZIM and RADELEV in deg), their first derivatives (RADRNGD in ft/sec, RADAZD and RADELD in deg/sec), and their second derivatives (RADRNGDD in ft/sec^2, RADAZDD and RADELDD in deg/sec^2). The first step in computing these output variables is to define a coordinate system for the radar station. The origin of the system is fixed with respect to the earth and has a radius vector $\vec{P}$ computed using the equations at the end of Section 2.1.4 with the earth shape and position values of the

Keyword	Units	Description
ALT	ft	station geodetic altitude
LONG	deg	station longitude
LATGD	deg	station geodetic latitude
DISTN	ft	offset north from given position
DISTE	ft	offset east from given position
DISTD	ft	offset down from given position
REQTR	ft	radius of the earth at the equator
FLAT		earth flatness parameter $(0.0 \leq FLAT < 1.0)$
ECC		earth eccentricity parameter $(0.0 \leq ECC < 1.0)$
RPOLAR	ft	radius of the earth at the poles

Table 25: Keywords for the RADAR Data Block

station. Local geodetic horizon unit vectors $\hat{x}_r$, $\hat{y}_r$, and $\hat{z}_r$ are defined at $\vec{P}$ in the manner discussed in Section 2.1.6.

The position of the vehicle relative to the radar station is given by $\vec{r}_r = \vec{r} - \vec{P}$ (of magnitude r_r, the slant range RADRNG). A unit vector along the line of sight may be defined as

$$\hat{u}_r = \frac{\vec{r}_r}{r_r} \tag{96}$$

The azimuth and elevation angles (RADAZIM and RADELEV) may then be defined as

$$\alpha_r = \tan^{-1}\left(\frac{\vec{r}_r \cdot \hat{y}_r}{\vec{r}_r \cdot \hat{x}_r}\right) \tag{97}$$

$$\varepsilon_r = \sin^{-1}(-\hat{u}_r \cdot \hat{z}_r) \tag{98}$$

Performing vector differentiation with respect to the (earth fixed) radar frame, the first two derivatives of the range term (RADRNGD and RADRNGDD) are given by

$$\dot{r}_r = {}^{\oplus}\vec{V} \cdot \hat{u}_r \tag{99}$$

$$\ddot{r}_r = {}^{\oplus}\vec{a} \cdot \hat{u}_r + {}^{\oplus}\vec{V} \cdot \dot{\hat{u}}_r \tag{100}$$

where

$$\dot{\hat{u}}_r = \frac{1}{r_r}\left[{}^{\oplus}\vec{V} - {}^{\oplus}\vec{V} \cdot \hat{u}_r \hat{u}_r\right] \tag{101}$$

Furthermore,

$$\ddot{\hat{u}}_r = \frac{1}{r_r}\left[{}^{\oplus}\vec{a} - \ddot{r}_r \hat{u}_r - 2\,{}^{\oplus}\vec{V} \cdot \hat{u}_r \dot{\hat{u}}_r\right] \tag{102}$$

The first two derivatives of the azimuth angle (RADAZD and RADAZDD) are

$$\dot{\alpha}_r = \frac{{}^{\oplus}\vec{V} \cdot \hat{y}_r \vec{r}_r \cdot \hat{x}_r - {}^{\oplus}\vec{V} \cdot \hat{x}_r \vec{r}_r \cdot \hat{y}_r}{(\vec{r}_r \cdot \hat{x}_r)^2 + (\vec{r}_r \cdot \hat{y}_r)^2} \tag{103}$$

$$\ddot{\alpha}_r = \frac{{}^{\oplus}\vec{a} \cdot \hat{y}_r \vec{r}_r \cdot \hat{x}_r - {}^{\oplus}\vec{a} \cdot \hat{x}_r \vec{r}_r \cdot \hat{y}_r - 2\dot{\alpha}_r({}^{\oplus}\vec{V} \cdot \hat{x}_r \vec{r}_r \cdot \hat{x}_r + {}^{\oplus}\vec{V} \cdot \hat{y}_r \vec{r}_r \cdot \hat{y}_r)}{(\vec{r}_r \cdot \hat{x}_r)^2 + (\vec{r}_r \cdot \hat{y}_r)^2} \tag{104}$$

The first two derivatives of the elevation angle (RADELD and RADELDD) are

$$\dot{\varepsilon}_r = -\dot{\hat{u}}_r \cdot \hat{z}_r \sec \varepsilon \tag{105}$$

$$\ddot{\varepsilon}_r = \left[\dot{\varepsilon}_r^2 \sin \varepsilon_r - \ddot{\hat{u}}_r \cdot \hat{z}_r\right] \sec \varepsilon_r \tag{106}$$

4.22 RESET Segment Data Block

The RESET segment data block may be used to change directly time and state vector values at the beginning of a segment. The format used and the values which may be changed are identical to those for the related INCREMENT segment data block. Refer to Section 4.10 and Table 18 for more information. An example use of the data block is given below.

```
SEGMENT 20     STAGING:  SET 2ND STAGE WEIGHT, RESET TIME COUNTER
  RESET        WT=2325.24   TMARK=0  FUEL=0.0
  PROP  TABLE STAGE_2P
  AERO  TABLE STAGE_2A
  FLY   ALFA  = 0
  FLY   BETAE = 0
  FLY  BANKGD = 0
  WHEN TMARK  = 12.83  GOTO 30
```

4.23 SEARCH Data Block

Often when defining a flight path, one or more of the input parameters must be chosen to satisfy a constraint. Such values can only be determined by generating several flight paths with different input values and selecting the appropriate input values that satisfy the known flight path constraints. The SEARCH data block can be used to define as many as six independent searches for path constraints so that designated input variables will automatically be varied to meet specified conditions.

An example, shown in Figure 28, will help clarify the definition of a search. This flight path required a descending turn from a 90 degree heading to a 120 degree heading to be completed when the vehicle reached an altitude of 40,000 feet. The segment guidance policies fix the bank angle at zero (BANKGD=0), command a −5 degree descending glide slope (GAMGD=−5), and specify an as yet unknown heading rate (PSIGDD=SRCH-1). The value SRCH-1 is a special keyword which will be replaced with various values in an iterative search for the one required to complete the turn at 40,000 feet. If SRCH-1 had appeared more than once in the problem, each corresponding parameter would be assigned the same value by the iteration software. Several flight paths, or portions of flight paths,

```
(EXAM-SRCH)
AERO  TABLE  TESTLIB
   SREF=2.3      LREF=1.7
INITIAL
   ALT=50000     LONG=-130     LAT=20
   GAMA=-5.0     PSI=90        VEL=8000
   WT=550        TIME=0        RANGE=0
*
SEGMENT  1  PERFORM A DESCENDING TURN TO 120 DEG HEADING
  FLY   BANKGD =  0
  FLY   GAMGD  = -5
  FLY   PSIGDD = SRCH-1
  WHEN  PSIGD  = 120   STOP
*
*  DEFINE SEARCH SO THAT TURN ON SEGMENT 1
*  IS COMPLETED AT AN ALTITUDE OF 40,000 FT
*
SEARCH  1  VARY PSIGDD UNTIL  ALT=40000 ON  SEGMENT  1
  XEST=2.0     DX=1.0     XLO=0.5     XHI=10.0
  MAXITR=10    TOL=20.0  YREF=10000  XREF=1
EXECUTE
END
```

Figure 28: A search can be defined to satisfy a path constraint.

will be computed in order to determine this value.

By setting the heading rate to SRCH-1 instead of a value, the user has indicated that the value for the heading rate will be defined as a search variable for search number 1. A SEARCH data block must be added to the problem that gives the constraint (or solution function) and the search control parameters.

The first parameter in the SEARCH data block defines the search number. It must be an integer 1 through 6 corresponding to that of the search variable (for the example, corresponding to the 1 in SRCH-1). As for segment numbers, search loop numbers are labels with no particular numerical significance. Positional parameters defining the solution function follow (in this example, ALT=40000 ON SEGMENT 1). The solution function is always computed at the end of a segment so the example definition is interpreted as *find the value of heading rate such that the altitude is 40,000 feet at the end of segment number 1.*

The remaining symbol/value pairs give an initial estimate (XEST) of the search variable, its estimated accuracy (DX), the search limits (XLO and XHI), the search tolerance (TOL), the maximum number of iterations (MAXITR), the solution function reference value (YREF), and the search function reference value (XREF). In Figure 28 the heading rate will be varied between 0.5 deg/sec and 10 deg/sec with an initial estimate of 2 deg/sec plus or minus 1 deg/sec. A maximum of 10 paths will be computed while trying to solve for a final altitude of 40,000 feet plus or minus 20 feet.

The reference values are used to avoid numerical problems when curve fitting x-y points that are several orders of magnitude apart. The search is actually performed with the quantities x/x_{ref} and y/y_{ref} where x is the search variable and y is the solution function. Values for x_{ref} and y_{ref} are input with the variables XREF and YREF. In the example shown above, YREF has been set to 10,000 because the solution function involves altitudes on the order of 10,000 ft. The value of XREF has been set to one because the search variable (heading rate) is on the order of one. The objective is to get both x/x_{ref} and y/y_{ref} approximately equal to one.

All of the SEARCH data block input variables in the example shown above are required except for XREF and YREF, which default to one. The search function can be defined to search for a specific value, as shown above, or it can be defined to search for a maximum or minimum value. The syntax of the SEARCH data block is outlined below where the positional parameters (in order) are

(1) - The search number (1, 2, 3, 4, 5, or 6)
(2) - The word FOR or an arbitrary *name* given in the format VARY *name* UNTIL
(3) - A standard or user defined output variable name
(4) - A constant or the words MIN or MAX
(5) - The word ON
(6) - The word SEGMENT
(7) - The segment number where the value is constrained
(8) - Left blank or the word PRINT

and the symbol/value pairs are

XEST -	Initial estimate of the search parameter
DX -	Estimate of the accuracy of the initial estimate
XLO -	Lower limit of the search parameter
XHI -	Upper limit of the search parameter
TOL -	Tolerance on the search function
MAXITR -	Maximum number of search iterations
XREF -	Search variable reference value
YREF -	Solution function reference value

All of the standard output variables (Appendix B) are available for the third positional parameter defining the solution function. User defined and summary variables, input with the DEFINE and SUMMARY data blocks, respectively, can also be used to define the solution function.

Another example of a search data block is given below.

```
SEARCH  3  FOR  VEL=MAX  ON SEGMENT 4  PRINT
  XLO=2   XHI=20      XEST=7      DX=2
  TOL=5   MAXITR=15   YREF=1000   XREF=1
```

Here, one or more variables in the flight path definition (not shown) paired with the symbol SRCH-3 will be varied between 2 and 20 with an initial guess of 7 plus or minus 2 until the velocity at the end of segment 4 is a maximum within 5 ft/sec. When a search is defined, the printout will show the values of the search variable and the solution function on each iteration. If the second postional parameter had been in the VARY *name* UNTIL form (see the first example), the given *name* (up to ten characters) would be used to label the output. Here the keyword FOR is used, so the symbol SRCH-3 will be used instead. Normally the trajectory will not be printed until all of the searches have converged, but a block printout of each path computed during the search can be requested by entering the last positional parameter PRINT (as in this example) rather than leaving it blank. (The block output format must also be active; see Section 4.18 for details.) This can be helpful in finding the reason a search did not converge.

During the iteration process those segments between the first appearance of the search variable and the objective function segment are integrated repeatedly. Such a set of segments is known as a search loop. If more than one search is defined over the same set of segments, the iteration software will determine a nesting procedure for iterating the search loops. The innermost search will be converged repeatedly as the outer loops are converged. Defining more than three nested search loops is likely to cause convergence problems, however. See Section 4.17 on optimization for the recommended procedure for satisfying several constraints simultaneously.

4.24 SEGMENT Data Block

Flight path segments are defined within SEGMENT data blocks. Each SEGMENT data block consists of a segment number and (optionally) a title followed by subordinate data blocks (known as segment data blocks) which are available to define aerodynamic, heating, and propulsion table set names, guidance policies, integration controls, state variable changes, user defined constants for the aerodynamics and propulsion tables, and final conditions. The segment data blocks (defined elsewhere in Section 4) include AERO, CONSTANTS, FLY, HEATING, INCREMENT, INERTIAL, INTEG, LAUNCH, PROP, RESET, and WHEN.

The first line of a SEGMENT data block contains the keyword SEGMENT followed by a positional parameter giving the segment number and a segment title. Segment numbers must be integers between 1 and 99 but are otherwise used only as labels without numeric significance. Segment titles are printed on each page of the block trajectory output to help identify where the vehicle is in the flight path. Each successive line which follows is assumed to apply to the segment until a line is encountered for which the first keyword on the line is not a segment data block name. Only two segment data blocks are required for each segment: the guidance policies (normally consisting of three FLY segment data blocks but may be as few as one) and the final conditions (via the WHEN segment data block).

Typical SEGMENT data blocks are given below

```
SEGMENT  1   PERFORM A LEVEL TURN TO -90 DEG
   INTEG  DTPRNT=1.0  DT=0.05
   FLY  HGDD   = 0.0
   FLY  PSIGDD = 5.0
   FLY  BANKGD = 0
   WHEN  PSIGD = -90  GOTO 4
   WHEN  TSEG  = 50    STOP
   WHEN  ALT   < 0     STOP
*
SEGMENT  4   DIVE AT A 10 DEG SLOPE TO 50,000 FT
   INTEG  DTPRNT=1.0  DT=0.05
   FLY  PSIGD  = -90
   FLY  BANKGD =   0
   FLY  GAMGD  =  -10
   WHEN  ALT   = 50000  GOTO  5
   WHEN  TSEG = 50      STOP
   WHEN  ALT  <  0      STOP
```

The segment titles may extend to the end of the line (Column 80). In both of these segments, integration controls have been specified in the INTEG data block (this is recommended although not required), guidance policies have been given in three FLY data blocks, and segment final conditions have been given in three WHEN data blocks.

4.25 SUMMARIZE Data Block

The SUMMARIZE data block is available for defining a special type of user defined output variable known as a summary variable. These variables have some segment specific and extreme value capability unavailable in the DEFINE data block (Section 4.4). Moreover, a table of final summary variable values is printed when all path calculations are complete. As many as five SUMMARIZE data blocks may appear in a problem, each defining a summary variable.

The first line of the data block begins with the SUMMARIZE keyword followed by a name (1 to 8 characters) for the variable unique amongst the output variables. Up to ten more lines may be used to complete the definition, each employing a single mathematical operation involving the trajectory output data. The available math operations are defined in Table 26 where *value* refers to the current value of the summary variable and *variable* refers to the trajectory output variable or constant input with the math operation. (Trajectory output variables include those previously defined by the user in DEFINE or SUMMARIZE data blocks. However, the value of those from DEFINE data blocks may be from the previous integration step so caution is advised.) Note that the math operations

ADD	Add the variable to the value
SUB	Subtract the variable from the value
MULT	Multiply the value by the variable
DIV	Divide the value by the variable
IDIV	Divide the variable by the value
EXP	Take the value to the variable power
IEXP	Take the variable to the value power
MAX	Limit the value to less than the variable
MIN	Limit the value to greater than the variable
ABS	Take the absolute value of the value
NEG	Negate the value
SQR	Square the value
SQRT	Take the square root of the value
LN	Take the natural log of the value
LOG	Take the base 10 log of the value
E	Take e to the power of the value
SIN	Take the sine of the value (in degrees)
COS	Take the cosine of the value (in degrees)
TAN	Take the tangent of the value (in degrees)
ASIN	Take the arcsine of the value (in degrees)
ACOS	Take the arccosine of the value (in degrees)
ATAN	Take the arctangent of the value (in degrees)

Table 26: Math Operations Available for the SUMMARIZE Data Block

available here are a subset of those available with the DEFINE data block.

For each calculation the value of the summary variable is initially set to zero and modified in accordance with each successive mathematical operation. For example, the data block

```
SUMMARIZE  DIST5-2
   ADD  RANGE  ON  SEGMENT  5
   SUB  RANGE  ON  SEGMENT  2
```

defines a summary variable called DIST5-2 as the difference between the ground range of segment 5 and the ground range of segment 2. The first mathematical operation, ADD, adds the ground range (RANGE) from segment 5 to the initial summary value (zero). The second mathematical operation, SUB, subtracts the segment 2 ground range from the current summary value resulting in the range difference between the segments. Until trajectory integration has begun on all segments referenced in a summary variable definition the value of the variable will be zero. If for this case segment 5 occurs after segment 2, as segment 5 is integrated, the final range on segment 2 will be subtracted from the current range. Subsequent segments will not change the value of the variable, and the value printed in the table of summary variables will be the range at the end of segment 5 minus the range at the end of segment 2.

The example above illustrates only the basic syntax of the SUMMARIZE data block. On term lines (those containing mathematical operations) the item following the mathematical operation defines how to retrieve the output variable from the trajectory data. The term lines also define the type of value required, the output variable name, and the scope of interest. If the entry following the math operation is MAX or MIN and followed by an output variable name, the current maximum or minimum value of the variable within the defined scope is used. Otherwise, the second item is an output variable name (as above) and the latest value of the variable over the given scope is used. In either case the entries following the variable name define the scope of the term. The scope is defined as a specific segment with the keywords ON SEGMENT n (where n is a segment number) or over the entire path with the keywords ON PATH (default).

Some examples may help clarify the syntax and the options available when defining summary variables:

```
SUMMARIZE  MAX-ALT
  ADD  MAX  ALT  ON PATH
  MULT 0.0003048
*
SUMMARIZE  RANGE-KM
  ADD   RANGE    ON  SEGMENT 5
  MULT  6076.115486
  MULT  0.0003048
*
SUMMARIZE  MIN-ELV
  ADD  MIN  RADELV  ON SEGMENT 3
```

The first summary variable, MAX-ALT, becomes the maximum altitude along the flight path in units of kilometers. The next summary variable, RANGE-KM, is the range at the end of segment 5 in kilometers (the conversion could have been accomplished with the single mathematical operation MULT 1.852). The last summary variable is assigned the minimum value of the radar elevation angle on segment 3.

When a SUMMARIZE data block is defined a table listing the survey and summarize variable values for each path is created at the end of the standard printout file after all path computations for a problem are complete. In this way aspects of the computed trajectories which are of prime interest may be compared in a single table. Some example SURVEY and SUMMARIZE data blocks are given in Figure 29, along with the resulting table.

```
SURVEY 1 HEADING
     XLO=30 XHI=60  DX=60
SURVEY 3 VERT-FPA
     XLO=-60 XHI=-20  DX=20
*
SUMMARIZE RANGE-KM
  ADD  RANGE
  MULT 1.852
SUMMARIZE DELT-VEL
  ADD  VEL  ON SEGMENT 4
  SUB  VEL  ON SEGMENT 1
  MULT 0.3048
```

PATH	HEADING	VERT-FPA	RANGE-KM	DELT-VEL
1	30.00000000	-60.0000000	112.7860306	-1200.01888
2	60.00000000	-60.0000000	112.9813747	-1201.00422
3	30.00000000	-40.0000000	232.7048995	-1607.52669
4	60.00000000	-40.0000000	233.1736772	-1610.91113
5	30.00000000	-20.0000000	536.5533231	-2807.25234
6	60.00000000	-20.0000000	539.6132380	-2829.10602

Figure 29: SURVEY and SUMMARIZE Data Blocks and the Resulting Summarize Table

4.26 SURVEY Data Block

Through use of the SURVEY data block, almost all numeric inputs in the other data blocks may be systematically varied over a set range of values. Up to three parameter surveys may be defined, labeled 1, 2, and 3. For each value in a survey a new path is started. Survey loop nesting is such that survey 3 (when defined) will be the outermost loop, and survey 1 the innermost. Moreover, SEARCH and OPTIMIZE loop nesting may be performed within each survey loop, since each path is restarted from the beginning for each survey value. Plots defined for the problem will display all survey paths on each plot, and the summary variable table associated with the SUMMARIZE data blocks (if any) will be printed at the end of the standard printout file with data from all paths that have been integrated.

To define a survey, one or more variables to be surveyed must be selected and input as SURV-*n*, where *n* is the survey loop number, and a SURVEY data block must be included in the problem. In Figure 30 the initial altitude, given in the INITIAL data block, has been selected as the survey variable. The SURVEY data block at the end of the problem gives the values for the survey variable. In this case, the initial altitude will be varied from 30,000 ft (XLO) to 60,000 ft (XHI) in steps of 10,000 ft (DX), and the special value of 36,089 ft will be inserted in the survey. Thus, five trajectories will be generated each with

```
(EXAM-SURV)
AERO  TABLE  TESTLIB
   SREF=2.3     LREF=1.7
INITIAL
   ALT  = SURV-1   LONG = -145   LAT = 20.
   GAMA = 0.0      PSI  = 90.    VEL = 8000.   WT = 550.
LIMITS    MAX ALFAT = 20.0
*
SEGMENT  1  PERFORM A LEVEL TURN TO 120 DEG HEADING
  FLY  PSIGDD = 1.0
  FLY  GAMGD  = 0.0
  FLY  BANKGD = 0.0
  WHEN PSIGD  = 120   STOP
  WHEN TSEG   = 100   STOP
*
*  RUN FOR ALT OF 30K, 36K, 40K, 50K, AND 60K
*
SURVEY  1
  XLO=30000   XHI=60000   DX=10000   XVAL1=36089
EXECUTE
END
```

Figure 30: Parametric surveys can quickly produce a series of flight paths.

Keyword	Description
XLO	beginning survey value
XHI	final survey value
DX	survey increment
XVAL1	extra survey value 1
XVAL2	extra survey value 2
XVAL*k*	extra survey value *k*, where *k* is an integer between 1 and 10

Table 27: Keywords Associated with the SURVEY Data Block

a different initial altitude (30000, 36089, 40000, 50000, and 60000 ft).

The first line of the SURVEY data block begins with the data block name followed by the survey loop number (1, 2, or 3). An optional survey name up to 10 characters in length may be entered as the final positional paramter on the line (FLT-PA in Figure 31). The survey name will be used to label the resulting output. This is followed by lines which pair values to the keywords given in Table 27.

Up to ten extra values can be input defining specific values of the survey variable. When n extra values are required the XVAL symbols 1 through n should be used (i.e., don't skip numbers). If the desired survey values do not fit the standard *low, high, increment* pattern, the survey may be input with the least value for XLO, the greatest value for XHI, the difference between XLO and XHI for DX, and the remaining values for XVAL1 through XVAL10. If for some reason a negative value for DX is desired, XLO should still be the initial (and hence greatest) value and XHI the final value.

A survey loop can control the value of more than one input variable; for example, in Figure 31 the value of the guidance variable GAMGD will be changed in the INITIAL data block and in both segments 1 and 2 for each flight path produced by the survey. This example generates three trajectories with flight path angles of -5, -10, and -15 degrees.

```
(EXAM-SURV)
AERO  TABLE  TESTLIB
   SREF=2.3     LREF=1.7
PROP  TABLE  SUSTAIN
INITIAL
   ALT=100000     LONG=-130    LAT=50
   GAMA=SURV-1    PSI=170      VEL=15000    WT=1000
LIMITS  ALFAT MAX = 20.0
*
SEGMENT  1  TURN RIGHT TO 120 DEG HEADING
  FLY  PSIGDD=2
  FLY  GAMGD=SURV-1
  FLY  BANKGD=0
  WHEN PSIGD=120  GOTO 2
*
SEGMENT  2  TURN LEFT BACK TO 90 DEG HEADING
  FLY  PSIGDD=-2
  FLY  GAMGD=SURV-1
  FLY  BANKGD=0
  WHEN PSIGD=90   STOP
*
SURVEY  1   FLT-PA
  XLO=-5   XHI=-15   DX=-5
EXECUTE
END
```

Figure 31: Surveys can change more than one parameter.

4.27 TARGET Data Block

The TARGET data block is similar in format to the INITIAL data block. Target initial conditions may be specified in spherical or cartesian coordinates using the keywords GEODETIC (default), GEOCENTRIC, ECFC, or ECIC along with the associated symbol/value pairs. Target motion is (by default or by using the keyword LINEAR) propagated assuming the velocity components remain constant at the values specified in the data block. Alternately, use of the keyword ORBITAL will cause the target state to be propagated under the influence of gravity (but no propulsion or aerodynamics are possible).

The target mission time TTARG corresponding to the given target conditions may be set using the TIME keyword which by default corresponds to the problem initial time. One may specify a time offset between the vehicle initial conditions and the given target conditions by matching a value to the keyword TOFFSET (a positive value indicates that the given target state occurs after the initial problem time). TSAP will integrate the given target state over the TOFFSET interval prior to problem execution to obtain the target state corresponding to the problem initial time. Such integration uses a time step of DTOFFSET (which is 1.0 sec by default). A list of all symbol/value pair keywords is listed in Table 28. Unless otherwise specified, all have default values of zero.

Keyword	Units	Description	Coordinate System
ALT	ft	altitude	Geodetic
DTOFFSET	sec	integration time step used to propagate target state to time corresponding to initial problem time when TOFFSET $\neq 0$ (default is 1.0)	
GAMA	deg	vertical flight path angle	Geodetic or Geocentric
LONG	deg	longitude	Geodetic or Geocentric
LAT	deg	latitude	Geodetic or Geocentric
PSI	deg	horizontal flight path angle	Geodetic or Geocentric
RCM	ft	distance to center of the earth	Geocentric
TIME	sec	target mission time of the given conditions	
TOFFSET	sec	difference between the target mission time of the given conditions and the time of the vehicle initial conditions	
VEL	ft/sec	velocity magnitude	Geodetic or Geocentric
X	ft	position X component	ECFC or ECIC
XDT	ft/sec	velocity X component	ECFC or ECIC
Y	ft	position Y component	ECFC or ECIC
YDT	ft/sec	velocity Y component	ECFC or ECIC
Z	ft	position Z component	ECFC or ECIC
ZDT	ft/sec	velocity Z component	ECFC or ECIC

Table 28: TARGET Data Block Keywords

As for the primary vehicle, the target state is propagated in ECFC terms. This allows conversion to geocentric and geodetic quantities to occur in precisely the same manner as well. Hence the target ECFC output variables such as TRGX and TRGXD correspond to those of Section 2.1.1 for the primary vehicle, and the geocentric and geodetic terms such as TRGLATGC and TRGGAMGD correspond to the similarly named variables of Sections 2.1.3 and 2.1.4. The terms TRGRNG, TRGELEV, TRGAZIM, and TRGRNGD (defined analogously to the radar terms described in Section 4.21 but with the primary vehicle state replacing that of the radar station) describe the target state in relation to the primary vehicle state. A complete list of the available standard output variables which describe the target state is available in Appendix B.

4.28 TITLE Data Block

An optional problem title can be defined with the TITLE data block. The title appears on all plots and output files, and it is also printed in the input data section of the standard printout. The first line of the data block consists solely of the keyword TITLE. As many as five lines may follow which contain the title. To prevent interpretation of a title line as a new data block the first word must not be a data block name. For example, the data block

```
TITLE
   THIS IS AN EXAMPLE OF A GOOD TITLE
   DATA BLOCK WITH TWO TITLE LINES
```

will be interpreted correctly because the first words in both lines (THIS and DATA) are not data block keywords. On the other hand, the following data block

```
TITLE
   THIS IS AN EXAMPLE OF A BAD
   TITLE DATA BLOCK WITH THREE LINES
   BECAUSE THE 2ND LINE BEGINS WITH "TITLE"
```

will not be interpreted correctly because the second line begins with TITLE which is a data block keyword. The problem should be inspected for this error whenever input errors occur in the vicinity of the TITLE data block.

4.29 WHEN Segment Data Block

As many as five different conditions for segment termination may be input for each segment with WHEN segment data blocks (one for each final condition). The first final condition to be satisfied during segment integration is executed.

The WHEN data block uses four positional parameters to define the segment termination condition and describe how to proceed. The termination condition is given by an output variable name and a value; for example,

```
WHEN  ALT=20000  GOTO  3
```

has a termination condition of ALT=20000. If the altitude becomes 20,000 feet during execution of this segment, the segment will terminate. The remaining parameters in the data block indicate how path integration is to proceed once the condition is achieved. In the example above, the parameters GOTO 3 cause path integration to continue with segment number 3. The alternative is to stop the path; for instance,

```
WHEN  TSEG = 50  STOP
```

terminates the flight path when the segment time equals 50 seconds.

The special delimiter symbols $<$ and $>$ have application in the WHEN segment data block. At segment initialization such conditions are checked and the requested action taken if the condition is satisfied. The example data block which follows will cause the path to be terminated if, at the beginning of the segment, the vehicle altitude is negative. If not, path integration will procede and the condition will be treated in the same manner as an equality condition.

```
WHEN  ALT < 0.0  STOP
```

A list of the standard output variables that can be used for segment termination is given in Appendix B. In addition, any output variable previously defined by the user in DEFINE or SUMMARIZE data blocks may also be used for segment final conditions.

4.30 WINDS Data Block

A table defining atmospheric winds as a function of a state variable, such as altitude, can be input with the WINDS data block. If the WINDS data block is not input, it is assumed that there are no winds.

The format of the WINDS data block is shown below.

```
WINDS  GEODETIC  INTERP-3
  ALT       VEL     HEADING     DOWN
    0        20          90        0
10000        30          90        0
20000        30          80        0
    .         .           .        .
    .   etc.  .           .        .
```

The first positional parameter following the WINDS keyword should be either GEODETIC or GEOCENTRIC, depending on the coordinate system used to define the wind velocity, heading, and down components.

Three interpolation methods are available for the winds table. The interpolation method is selected with the second positional parameter on the first line as shown above. The keywords used to specify an interpolation method are listed in Table 9, the default method being INTERP-3.

The next line contains four variable names identifying the table columns. The independent variable must be listed first and be among the standard output variables listed in Table 16 of Section 4.8. The state variable acts as the table independent variable, and its values in the table must be strictly increasing. The remaining column headings must contain the keywords VEL, HEADING, and DOWN. They can be listed in any order, but the order shown in the above example is recommended.

The remaining lines in the WINDS data block contain the tabulated data values defining the wind components. A maximum of 50 points can be input defining each function. The wind velocity is given in feet per second, the wind heading is the direction the wind is from in degrees measured clockwise from the north (for example, a heading of 90 deg means the wind is from the east), and the wind down component is given in feet per second.

References

1. Salguero, David E. *Point-Mass Simulation Tool (PMAST) User's Manual*, **Sand85-2039**, March 1986.

2. Outka, David E. *Parameter Optimization Capability in the Trajectory Code PMAST*, **Sand86-2917**, January 1987.

3. **DISSPLA User's Manual**, Integrated Software Systems Corporation, San Diego, CA, 1984.

4. Hannah, Michael, **MASS Reference Manual, Version 1.0**, Computer Consulting and Training Division, Sandia Laboratories.

5. Kane, Thomas R. and Levinson, David A. **Dynamics: Theory and Applications**, McGraw-Hill, New York, 1985.

6. *Department of Defense World Geodetic System 1984*, Defense Mapping Agency WGS 84 Development Committee, **DMA TR 8350.2**, Washington DC, 30 September 1987

7. Seppelin, T. O.,*The Department of Defense World Geodetic System 1972*, World Geodetic System Committee, Inernational Symposium on Problems Related to the Redefinition of North American Geodetic Networks, Frederiction, New Brunswick, Canada, May 1974

8. McDowell, Jerry L., Schutz, Bob E., McKenzie, Richard E., Tapley, Byron D., *Trajectory Analysis Program Mathematical Specifications (Version 770701)*, Center for Advanced Study in Orbital Mechanics, Department of Aerospace Engineering and Engineering Mechanics, University of Texas at Austin, **TR 77-1**, 1977.

9. *Kwajelein Reference Atmosphere 1979*, Air Force Geophysics Laboratory, **AFGL-TR-79-0241**, September 1979.

10. Landrum, D. Brian *Standard Atmosphere Subroutine* , **RS1555/87/010** (U), January 1987.

11. **U.S. Standard Atmosphere, 1962**, National Aeronautics and Space Administration, United States Air Force, and United States Weather Bureau, U.S. Government Printing Office, Washington DC, 1962.

12. **U.S. Standard Atmosphere Supplements, 1966**, Environmental Science Services Administration, National Aeronautics and Space Administration, and the United States Air Force, U.S. Government Printing Office, Washington DC, 1966.

13. **U.S. Standard Atmosphere, 1976**, National Oceanic and Atmospheric Administration, National Aeronautics and Space Administration, and the United States Air Force, U.S. Government Printing Office, Washington DC, 1976.

14. Greene, R. W., *VAX Procedure for Accessing Atmospheric Properties Tables, ATMOS*, Informal Memorandum, Sandia National Laboratories, Albuquerque, NM, 21 April 1988.

15. Burden, Richard L., Faires, J. Douglas, and Reynolds, Albert C., **Numerical Analysis**, Prindle, Weber & Schmidt, Boston, MA, 1978.

16. Stineman, Russell W., *A Consistently Well-Behaved Method of Interpolation,* **Creative Computing**, July 1980, pp. 54-57.

17. Powell, M. J. D. *A Fast Algorithm for Nonlinearly Constrained Optimization Calculations*, **Proceedings of the Biennial Conference on Numerical Analysis, 28 June – 1 July 1977**, Watson, G. A., editor, pp. 144-57. Springer-Verlag, Berlin, Germany 1978.

Appendices

A An Example Problem

A rather extensive example problem is included here for reference. The vehicle involved is a two stage sounding rocket which deploys a hypersonic glide aircraft. The table input file includes several table sets to cover the various flight regimes. Among the problem input file features are simulated vehicle launch, flight path optimization, a path search, and an iterative guidance policy.

The table input file appears on the next three pages. This is followed by the standard printout file, the first few pages of which are an echo of the problem input file file. The two requested plots complete the output presented. The problem was run using the TSAPRUN utility with the CTSS *Batch option*; the job was charged about 8.1 minutes of time on machine w (a Cray X-MP/416).

```
********************************************************** EXAMPLE TABLE FILE
(STAGE-1)
   PROP       1ST STAGE THRUST
   SECTION    THRUST
   THRC       ADD        TMARK
    48
* TIME VALUES
    ØØ.ØØ      ØØ.1Ø      Ø.16       ØØ.2Ø      ØØ.3Ø      Ø1.ØØ
     1.4Ø       1.8Ø      2.2Ø        3.2Ø       4.ØØ      Ø4.2Ø
     5.ØØ       6.2Ø      7.ØØ        9.2Ø      1Ø.ØØ      1Ø.4Ø
    11.ØØ      11.2Ø     11.6Ø       12.ØØ      12.8Ø      13.6Ø
    15.ØØ      17.ØØ     19.2Ø       2Ø.ØØ      21.2Ø      22.ØØ
    23.2Ø      24.ØØ     25.2Ø       26.ØØ      27.ØØ      28.ØØ
    29.ØØ      3Ø.ØØ     31.2Ø       32.ØØ      33.2Ø      34.ØØ
    35.ØØ      36.2Ø     37.ØØ       38.2Ø      39.5Ø      99.99
* VACUUM THRUST VALUES
    ØØØØØ.     22467.    125Ø99.    146231.    145922.    145888.
   145147.    144375.    138114.    135146.    127717.    122373.
   113192.    1ØØ452.     92513.     83421.     76793.     71135.
    68787.     655Ø9.     64ØØ4.     625Ø4.     61417.     6Ø471.
    6Ø2Ø9.     6Ø156.     6Ø414.     6Ø59Ø.     6Ø938.     61286.
    61721.     62113.     6133Ø.     56977.     44735.     28296.
    2Ø777.     15736.     113Ø8.     Ø9124.     Ø65Ø5.     Ø55Ø1.
    Ø3712.     Ø2183.     Ø13Ø9.     ØØ566.     ØØØØØ.     ØØØØØ.
*
   VACTHR     CSTO
*
**** CORRECT FOR ALTITUDE EFFECTS
*
   PRESS      SUB
   3.219      MULT
   VACTHR     ADD
              END
*
**** 1ST STG MASS FLOW SECTION ******************************
*
SECTION    WTDT       SLUG/SEC
   FFC        ADD        TMARK
    48
* TIME VALUES
    ØØ.ØØ      ØØ.1Ø      Ø.16       ØØ.2Ø      ØØ.3Ø      Ø1.ØØ
     1.4Ø       1.8Ø      2.2Ø        3.2Ø       4.ØØ      Ø4.2Ø
     5.ØØ       6.2Ø      7.ØØ        9.2Ø      1Ø.ØØ      1Ø.4Ø
    11.ØØ      11.2Ø     11.6Ø       12.ØØ      12.8Ø      13.6Ø
    15.ØØ      17.ØØ     19.2Ø       2Ø.ØØ      21.2Ø      22.ØØ
    23.2Ø      24.ØØ     25.2Ø       26.ØØ      27.ØØ      28.ØØ
    29.ØØ      3Ø.ØØ     31.2Ø       32.ØØ      33.2Ø      34.ØØ
    35.ØØ      36.2Ø     37.ØØ       38.2Ø      39.5Ø      99.99
* MASS RATE VALUES
    Ø.ØØØØ     2.7556    15.3863    18.1426    18.1376    18.Ø962
   18.Ø811    17.9123    17.1393    16.7674    15.8454    15.1865
   14.Ø435    12.4652    11.4779    1Ø.3497     9.5264     8.8256
    8.5523     8.3Ø37     8.25ØØ     7.9Ø2Ø     7.79Ø2     7.6Ø92
    7.4833     7.4767     7.5Ø88     7.53Ø7     7.5739     7.6172
    7.6713     7.72ØØ     7.6227     7.Ø816     5.56Ø1     3.5169
    2.5824     1.9558     1.4Ø55     1.134Ø     Ø.8Ø85     Ø.6837
    Ø.4614     Ø.2713     Ø.1627     Ø.Ø7Ø3     Ø.ØØØØ     Ø.ØØØØ
          END
************************************************************************************
(STAGE-2)
   PROP       SECOND STAGE THRUST -- ASSUMES EXOATMOSPHERIC OPERATION
   SECTION    THRUST
   THRA       ADD        TMARK
      31
   Ø.         Ø.3        Ø.97       3.32       3.66       4.33
   9.Ø1       11.42      13.82      16.23      17.83      19.44
   21.Ø4      23.46      25.Ø5      29.Ø6      31.47      32.27
   35.48      39.49      41.Ø9      43.5       44.9       45.25
   45.48      45.7       46.6       47.Ø5      47.5       48.39
   99.
   Ø.         12899.     12711.     15576.     15465.     15632.
   17979.     18951.     19712.     2Ø297.     2Ø543.     2Ø666.
   2Ø619.     2Ø318.     19972.     18561.     176Ø7.     17581.
   17781.     17876.     17782.     17378.     17222.     16942.
   16213.     14932.     1942.      421.       136.       Ø.
   Ø.
          END
*
```

```
**** SECONDE STAGE MASS FLOW RATE ************************
*
SECTION   WTDT       SLUG/SEC
MFRA      ADD        TMARK
  31
Ø.        Ø.3        Ø.97       3.32       3.66       4.33
9.Ø1      11.42      13.82      16.23      17.83      19.44
21.Ø4     23.45      25.Ø5      29.Ø6      31.47      32.27
35.48     39.49      41.Ø9      43.5       44.9       45.25
45.48     45.7       46.6       47.Ø5      47.5       48.39
99.
Ø.        1.3565     1.3367     1.638Ø     1.6263     1.6439
1.89Ø7    1.9929     2.Ø729     2.1344     2.16Ø3     2.1732
2.1683    2.1366     2.1ØØ3     1.9519     1.8515     1.8488
1.8698    1.8798     1.87ØØ     1.8275     1.8111     1.7816
1.7Ø5Ø    1.57Ø2     Ø.2Ø42     Ø.Ø443     Ø.Ø143     Ø.
Ø.
          END
*****************************************************************************
(NOTHRUST)
PROP      SIMPLE NO THRUST OR MASS FLOW TABLE
SECTION   WTDT
          ZERO
          END
*
*****************************************************************************
*
(AERO-1T)
AERO      FIRST STAGE DRAG  COEF. WHILE THRUSTING
SECTION   CA/CD      CD          SREF=6.41
CA        ADD        MACH
  19
Ø.        Ø.6        Ø.72       Ø.8        Ø.85       Ø.9
1.Ø2      1.1        1.18       1.5        1.75       2.1
2.9       3.2        4.         5.2        6.2        8.
1Ø.
Ø.73      Ø.72       Ø.74       Ø.76       Ø.82       Ø.95
1.52      1.42       1.33       1.Ø3       Ø.88       Ø.73
Ø.54      Ø.5        Ø.48       Ø.43       Ø.37       Ø.26
Ø.2
          END
****************************************************************************
(AERO-1C)
AERO      FIRST STAGE DRAG  COEF. WHILE COASTING
SECTION   CA/CD      CD          SREF=6.41
CA        ADD        MACH
  19
Ø.        Ø.6        Ø.72       Ø.8        Ø.85       Ø.9
1.Ø2      1.1        1.18       1.5        1.75       2.1
2.9       3.2        4.         5.2        6.2        8.
1Ø.
Ø.8       Ø.79       Ø.82       Ø.84       Ø.9        1.Ø4
1.65      1.56       1.46       1.15       Ø.99       Ø.83
Ø.61      Ø.57       Ø.53       Ø.45       Ø.38       Ø.27
Ø.2
          END
****************************************************************************
(AERO-2C)
AERO      SECOND STAGE DRAG  COEF. WHILE COASTING
SECTION   CA/CD      CD          SREF=5.13
CA        ADD        MACH
  1Ø
Ø.        Ø.8        1.         1.5        2.         3.
4.        6.         1Ø.        2Ø.
Ø.284     Ø.284      Ø.595      Ø.392      Ø.299      Ø.2Ø7
Ø.158     Ø.1Ø7      Ø.Ø76      Ø.Ø65
          END
*
****************************************************************************
*
(EXO-ATMOS)
AERO      SIMPLE NO-AERO-FORCES TABLE
SECTION   CA/CD      CD          SREF=1.Ø
          ZERO
          END
*
****************************************************************************
*
```

```
(RVAERO)
   AERO       SIMPLE AERO DATA FOR AN RV
   SECTION    CA/CD      CA         SREF=2.Ø331
   CA         ADD        ALFAT
      19
*  ALFAT VALUES
   -Ø.62ØØØØØ
    Ø.31ØØØØ   1.2ØØØØØ   2.16ØØØØ   3.15ØØØØ   4.19ØØØØ   5.16ØØØØ
    6.1ØØØØØ   7.ØØØØØØ   7.89ØØØØ   8.82ØØØØ   9.77ØØØØ 1Ø.72ØØØ
   11.65ØØØ   12.6ØØØØ   13.6ØØØØ   14.64ØØØ   15.62ØØØ   15.74ØØØ
*  CA VALUES
    Ø.Ø765Ø
    Ø.Ø772Ø    Ø.Ø775Ø    Ø.Ø778Ø    Ø.Ø799Ø    Ø.Ø836Ø    Ø.Ø875Ø
    Ø.Ø918Ø    Ø.Ø965Ø    Ø.1Ø14ØØ   Ø.1Ø71ØØ   Ø.1128ØØ   Ø.118ØØ
    Ø.1234ØØ   Ø.1293ØØ   Ø.1358ØØ   Ø.1427ØØ   Ø.1495ØØ   Ø.1505ØØ
            END
   SECTION    CN/CL
   CN         ADD        ALFAT
      19
*  ALFAT VALUES
   -Ø.62ØØØØØ Ø.31ØØØØ   1.2ØØØØØ   2.16ØØØØ   3.15ØØØØ   4.19ØØØØ
    5.16ØØØØ   6.1ØØØØØ   7.ØØØØØØ   7.89ØØØØ   8.82ØØØØ   9.77ØØØØ
   1Ø.72ØØØ   11.65ØØØ   12.6ØØØØ   13.6ØØØØ   14.64ØØØ   15.62ØØØ
   15.74ØØØ
*  CN VALUES
   -Ø.Ø23Ø     Ø.Ø1ØØ     Ø.Ø4ØØØ    Ø.Ø74ØØ    Ø.1Ø8ØØØ   Ø.144ØØØ
    Ø.178ØØØØ Ø.21ØØØ    Ø.241ØØØ   Ø.274ØØØ   Ø.3Ø9ØØØ   Ø.346ØØØ
    Ø.379ØØØØ Ø.412ØØ    Ø.447ØØØ   Ø.485ØØØ   Ø.524ØØØ   Ø.561ØØØ
    Ø.567ØØØØ
              END
```

PROBLEM FILE LISTING:

```
(HYPER-X)
TITLE
      TWO STAGE ROCKET LAUNCH OF HYPERSONIC VEHICLE
INITIAL
     ALT=212        LONG=-120.63618      LAT=34.88291
     VEL=0          GAMA=82.0            PSI=-40.0
     TIME=0.0       WT=14987.1
*
OUTPUT    PRINT
  TIME  ALT  LONG  LATGD  VEL  GAMGD  PSIGD  DYNPRS  WT  NX
OUTPUT    PRINT
  TIME  RANGE  ALT  MACH  ALTD  ALFA  BANKGD  GAMGD  PSIGD  DYNPRS
PLOT  ALT  VRS  RANGE
PLOT  GROUNDTRACK    MAPFILE=COAST  MAPFILE=USAHIGH
*
SEGMENT  1   IGNITION TO FIRST MOVEMENT
  INTEG DT=0.01   DTPRNT=0.20
  AERO       TABLE    AERO-1T
  PROP       TABLE    STAGE-1
  LAUNCH     CFSTAT=0.15   CFSLID=0.1
  FLY    PITCHGD  =  OPTA-1
  FLY    YAWGD    = -40.0
  FLY    ROLLGD   =  00.0
  WHEN   VEL      >  00.001       GO TO 2
*
SEGMENT  2   TO END OF LAUNCH RAIL
  INTEG DT=0.01   DTPRNT=0.20
  AERO       TABLE    AERO-1T
  PROP       TABLE    STAGE-1
  LAUNCH     CFSTAT=0.15   CFSLID=0.1
  FLY    PITCHGD  =  *
  FLY    YAWGD    =  *
  FLY    ROLLGD   =  00.0
  WHEN  PATHLENG  >  19.958   GO TO 5
*
SEGMENT  5  ASSUME SOME DYNAMICS AS Q BUILDS
  INTEG DT=0.05   DTPRNT=2.5
  AERO       TABLE    AERO-1T
  PROP       TABLE    STAGE-1
  FLY    ALFA    VRS  TSEG
           *            0.0
         0.3            1.0
         0.3            4.0
         0.2            5.0
         0.1            8.0
         0.0           10.0
         0.0           11.0
         0.0           99.0
  FLY   BETAE   VRS   TSEG
```

```
             *           Ø.Ø
            Ø.3          1.Ø
           -Ø.2          4.Ø
            Ø.2          5.Ø
           -Ø.1          8.Ø
            Ø.Ø         1Ø.Ø
            Ø.Ø         11.Ø
            Ø.Ø         99.Ø
  FLY  BANKGD = Ø.Ø
  WHEN TMARK  > 39.5    GOTO 15
*
SEGMENT  15   ZERO-ALFA COAST
  INTEG DT=Ø.25    DTPRNT=5.Ø
  AERO       TABLE      AERO-1C
  PROP       TABLE      NOTHRUST
  FLY  ALFA   = Ø
  FLY  BETA   = Ø
  FLY  BANKGD = Ø
  WHEN DYNPRS < 1Ø  GOTO 2Ø
*
SEGMENT  2Ø    ASSUME ATTITUDE REMAINS INERTIALLY CONSTANT
  INTEG DT=Ø.5    DTPRNT=5.Ø
  AERO       TABLE      AERO-1C
  PROP       TABLE      NOTHRUST
  FLY  PITCHI = *
  FLY  YAWI   = *
  FLY  ROLLI  = Ø
  WHEN TMARK  > OPTA-2     GOTO 25
*
SEGMENT  25  STAGE AND PROCEED TO REORIENT
  RESET      WT = 3842.7
  INTEG  DT=1.Ø    DTPRNT=1Ø.Ø
  AERO       TABLE      AERO-2C
  PROP       TABLE      NOTHRUST
  FLY  PITCHGD = OPTA-3
  FLY  YAWGD   = OPTA-4
  FLY  ROLLGD  =   Ø
  WHEN TSEG    = 6Ø                GOTO 3Ø
*
SEGMENT  3Ø   SECOND STAGE BURN
* ASSUME SMALL WT DECREASE DUE TO ACS GAS CONSUMPTION
  INCREMENT  WT = -2
  RESET    TMARK =  Ø
  INTEG DT=Ø.Ø5  DTPRNT=2.Ø
  AERO       TABLE      EXO-ATMOS
  PROP       TABLE      STAGE-2
  FLY  PITCHI = *
  FLY  YAWI   = *
  FLY  ROLLI  = Ø
  WHEN TSEG   = 48.4    GOTO 4Ø
*
SEGMENT  4Ø  RELEASE PAYLOAD & COAST TO APOGEE
```

```
SANDIA NATIONAL                TRAJECTORY SIMULATION AND ANALYSIS PROGRAM --- TSAP          PROBLEM:
LABORATORIES                                  (VERSION 89.ØØ1)                              PAGE:     ØØ3

     INTEG DT=5.Ø   DTPRNT=1Ø
     AERO      TABLE      EXO-ATMOS
     PROP      TABLE      NOTHRUST
     RESET   TMARK = Ø      WT = 596.8
     FLY  PITCHI  VRS  GAMGD
           -2Ø           Ø.Ø
            *             *
     FLY  YAWI   =   *
     FLY  ROLLI  =   Ø
     WHEN GAMGD  <   Ø       GO TO 5Ø
   *
   SEGMENT  5Ø  COAST TO Ø.3M FT ALT
     INTEG DT=1.Ø   DTPRNT=1Ø.Ø
     AERO      TABLE      EXO-ATMOS
     PROP      TABLE      NOTHRUST
     FLY  PITCHI =   *
     FLY  YAWI   =   *
     FLY  ROLLI  =   Ø
     WHEN  TMARK >  OPTA-5   GO TO 7Ø
   *
   OPTIMIZE A FOR VEL=MAX ON SEGMENT 5Ø
     FEST  = 1ØØØØ          ACC  = Ø.ØØØØØ1       MAXFUN = 1ØØ
     PAR-1 = 8Ø.Ø           LO-1 = 6Ø              HI-1  = 86
     PAR-2 = 7Ø.            LO-2 = 5Ø              HI-2  = 14Ø
     PAR-3 =  1.            LO-3 =-2Ø              HI-3  = 8Ø
     PAR-4 =-3Ø.            LO-4 =-8Ø              HI-4  = 5Ø
     PAR-5 = 2ØØ.           LO-5 = 1Ø              HI-5  = 1ØØØ
        CONSTRAINTS
     DYNPRS <  1.Ø        ON SEGMENT 2Ø
     ALT    >  Ø.35E6     ON SEGMENT 25
     PSIGD  = -32.Ø       ON SEGMENT 3Ø
     DYNPRS <  Ø.1        ON SEGMENT 4Ø
     ALT    =  Ø.3E6
     GAMGD  > -2Ø.Ø
   *
   SEGMENT  7Ø  HYPERSONIC PULL UP WITH TURN
     INTEG DT=Ø.2   DTPRNT=5.Ø
     AERO      TABLE      RVAERO
     PROP      TABLE      NOTHRUST
     FLY  ALFA   = 8
     FLY  BETAE  = Ø
     FLY  BANKGD = 3Ø
     WHEN GAMGD  > Ø      GOTO 75
     WHEN   ALT  < Ø      STOP
   *
   SEGMENT  75  HYPERSONIC PULL UP PART TWO
     INTEG DT=Ø.2   DTPRNT=2.5
     AERO      TABLE      RVAERO
     PROP      TABLE      NOTHRUST
     FLY  ALFA   = *
     FLY  BETAE  = *
     FLY  BANKGD  VRS    TSEG      INTERP-1
```

```
            *           Ø.Ø
          SRCH-1       1Ø.Ø
          SRCH-1       2Ø.Ø
          8Ø.Ø         3Ø.Ø
     WHEN   TSEG > 3Ø.Ø      GOTO 8Ø
     WHEN DYNPRS < 1ØØØ      STOP
     WHEN    ALT  < Ø        STOP
   *
   SEARCH  1  FOR  ALTD = Ø  ON SEGMENT 75
     XLO=Ø.Ø  XHI=2ØØ.Ø   XEST=11Ø.Ø  DX=2Ø
     TOL=1.Ø  MAXITR=15  YREF=1.Ø  XREF=1ØØ.Ø
   *
   LIMITS  MAX ALFAT = 15.Ø
   *
   SEGMENT  8Ø  FLY LEVEL S-TURNS, THEN DIVE
     INTEG DT=Ø.2   DTPRNT=5.Ø
     AERO       TABLE       RVAERO
     PROP       TABLE       NOTHRUST
     FLY  ALTD      VRS  TSEG  INTERP-1
            Ø.Ø            Ø.Ø
            Ø.Ø           7Ø.Ø
        -1ØØØ.Ø           9Ø.Ø
        -1ØØØ.Ø          12Ø.Ø
            Ø.Ø          15Ø.Ø
            Ø.Ø          17Ø.Ø
         -2ØØ.Ø          18Ø.Ø
         -2ØØ.Ø          2ØØ.Ø
     FLY  BANKGD    VRS  TSEG  INTERP-1
            *              Ø.Ø
            Ø.Ø           1Ø.Ø
          -6Ø.Ø           15.Ø
          -6Ø.Ø           25.Ø
            Ø.Ø           3Ø.Ø
           7Ø.Ø           35.Ø
           7Ø.Ø           45.Ø
            Ø.Ø           5Ø.Ø
          -5Ø.Ø           55.Ø
          -5Ø.Ø           65.Ø
            Ø.Ø           7Ø.Ø
           2Ø.Ø           9Ø.Ø
           2Ø.Ø          2ØØ.Ø
     FLY  BETAE   =  *
     WHEN   TSEG  >  2ØØ     STOP
     WHEN DYNPRS < 1ØØØ      STOP
     WHEN    ALT  < Ø        STOP
   *
   EXECUTE
   END
```

```
SANDIA NATIONAL            TRAJECTORY SIMULATION AND ANALYSIS PROGRAM --- TSAP        PROBLEM:
LABORATORIES                                (VERSION 89.001)                          PAGE:     005

     TABLE-SET STAGE-1    - 1ST STAGE THRUST
     WAS SUCCESSFULLY READ (REQUIRING     313 WORDS OF STORAGE)


     TABLE-SET STAGE-2    - SECOND STAGE THRUST -- ASSUMES EXOATMOSPHERIC OPER
     WAS SUCCESSFULLY READ (REQUIRING     180 WORDS OF STORAGE)


     TABLE-SET NOTHRUST   - SIMPLE NO THRUST OR MASS FLOW TABLE
     WAS SUCCESSFULLY READ (REQUIRING      22 WORDS OF STORAGE)


     TABLE-SET AERO-1T    - FIRST STAGE DRAG  COEF. WHILE THRUSTING
     WAS SUCCESSFULLY READ (REQUIRING      70 WORDS OF STORAGE)


     TABLE-SET AERO-1C    - FIRST STAGE DRAG  COEF. WHILE COASTING
     WAS SUCCESSFULLY READ (REQUIRING      70 WORDS OF STORAGE)


     TABLE-SET AERO-2C    - SECOND STAGE DRAG  COEF. WHILE COASTING
     WAS SUCCESSFULLY READ (REQUIRING      52 WORDS OF STORAGE)


     TABLE-SET EXO-ATMOS  - SIMPLE NO-AERO-FORCES TABLE
     WAS SUCCESSFULLY READ (REQUIRING      24 WORDS OF STORAGE)


     TABLE-SET RVAERO     - SIMPLE AERO DATA FOR AN RV
     WAS SUCCESSFULLY READ (REQUIRING     134 WORDS OF STORAGE)


                     **************************************************
                     *                                                *
                     *   TRAJECTORY SIMULATION AND ANALYSIS PROGRAM   *
                     *             TSAP   (VERSION 89.001)            *
                     *                                                *
                     **************************************************
```

```
SANDIA NATIONAL                TRAJECTORY SIMULATION AND ANALYSIS PROGRAM --- TSAP          PROBLEM:  HYPER-X
LABORATORIES                                (VERSION 89.001)                                PAGE:     006

  ***** HISTORY OF SEARCHES TO SATISFY PATH CONSTRAINTS *****


     OPTA PARAMETER OPTIMIZATION ITERATION HISTORY:

X =  OPTA- 1           OPTA- 2           OPTA- 3           OPTA- 4           OPTA- 5

F =  VEL     /  -0.1000000000E+05        ON SEGMENT 50

C =  PSIGD   SEG 30    ALT     SEG 50    DYNPRS  SEG 20    ALT     SEG 25    DYNPRS  SEG 40
     GAMGD   SEG 50    LO- 1             HI- 1             LO- 2             HI- 2
     LO- 3             HI- 3             LO- 4             HI- 4             LO- 5
     HI- 5

  ITERATIONS =    1     CALLS OF VF02AD =    1

X =  0.8000000000E+02  0.7000000000E+02  0.1000000000E+01 -0.3000000000E+02  0.2000000000E+03

F = -0.1486432615E+01

C = -0.1345413241E-01 -0.4676283813E+00  0.4699663132E+00  0.1742097458E+00  0.9947649060E+00
     0.2635780909E+00  0.3333333333E+00  0.6976744186E-01  0.4000000000E+00  0.5000000000E+00
     0.1050000000E+01  0.9875000000E+00  0.6250000000E+00  0.1600000000E+01  0.1900000000E+02
     0.8000000000E+00
 1 LINEAR SEARCH ITERATION,    STATUS FLAG INF = 0   LINEAR SEARCH STEP-SIZE: ALPHA =  1.00000

  ITERATIONS =    2     CALLS OF VF02AD =    2

X =  0.7031555156E+02  0.7210394792E+02  0.1026989136E+01 -0.2949186956E+02  0.1214728485E+03

F = -0.1576945965E+01

C = -0.4154182702E-02 -0.3924620406E+00 -0.1208171451E+01 -0.5977909837E-01  0.5013102060E+00
     0.5214168672E+00  0.1719258593E+00  0.1823773075E+00  0.4420789583E+00  0.4849718006E+00
     0.1051349457E+01  0.9871626358E+00  0.6313516305E+00  0.1589837391E+01  0.1114728485E+02
     0.8785271515E+00
 1 LINEAR SEARCH ITERATION,    STATUS FLAG INF = 0   LINEAR SEARCH STEP-SIZE: ALPHA =  1.00000
```

```
   ITERATIONS =      3     CALLS OF VF02AD =      3

 X =    0.7204378403E+02   0.7344702192E+02   0.1037306083E+01  -0.2977523136E+02   0.9239632352E+02


 F =   -0.1542234637E+01

 C =   -0.9973730432E-04  -0.4527474504E-01  -0.2667656385E+00  -0.1506164872E-02   0.8769639287E+00
        0.6447058895E+00   0.2007297339E+00   0.1622815810E+00   0.4689404386E+00   0.4753784148E+00
        0.1051865304E+01   0.9870336740E+00   0.6278096080E+00   0.1595504627E+01   0.8239632352E+01
        0.9076036765E+00
  1 LINEAR SEARCH ITERATION,    STATUS FLAG INF = 0   LINEAR SEARCH STEP-SIZE: ALPHA =  1.00000

   ITERATIONS =      4     CALLS OF VF02AD =      4

 X =    0.7166163442E+02   0.7772140834E+02   0.1044148182E+01  -0.2975063564E+02   0.7339118139E+02


 F =   -0.1542716677E+01

 C =   -0.3930793758E-04  -0.3997187195E-02   0.2274527174E+00  -0.1351868842E-02   0.8145920978E+00
        0.6895666624E+00   0.1943605736E+00   0.1667251812E+00   0.5544281667E+00   0.4448470833E+00
        0.1052207409E+01   0.9869481477E+00   0.6281170545E+00   0.1595012713E+01   0.6339118139E+01
        0.9266088186E+00
  1 LINEAR SEARCH ITERATION,    STATUS FLAG INF = 0   LINEAR SEARCH STEP-SIZE: ALPHA =  1.00000
  2 LINEAR SEARCH ITERATIONS,   STATUS FLAG INF = 0   LINEAR SEARCH STEP-SIZE: ALPHA =  0.54070

   ITERATIONS =      5     CALLS OF VF02AD =      6

 X =    0.7065880892E+02   0.9043633591E+02   0.1059379069E+01  -0.2970925568E+02   0.3911584190E+02


 F =   -0.1550418282E+01

 C =    0.6022600398E-03  -0.8875268120E-02   0.7988849573E+00  -0.1205185946E-01   0.1988144710E+00
        0.7358569195E+00   0.1776468154E+00   0.1783859428E+00   0.8087267183E+00   0.3540261720E+00
        0.1052968953E+01   0.9867577616E+00   0.6286343040E+00   0.1594185114E+01   0.2911584190E+01
        0.9608841581E+00
  1 LINEAR SEARCH ITERATION,    STATUS FLAG INF = 0   LINEAR SEARCH STEP-SIZE: ALPHA =  1.00000

   ITERATIONS =      6     CALLS OF VF02AD =      7

 X =    0.7092438006E+02   0.9225831230E+02   0.1059695167E+01  -0.2970635011E+02   0.3831910781E+02


 F =   -0.1547698328E+01

 C =    0.9005958077E-06  -0.6785523176E-04   0.8537283913E+00  -0.6703384672E-04   0.3085933928E+00
        0.7324800122E+00   0.1820730010E+00   0.1752979063E+00   0.8451662461E+00   0.3410120550E+00
        0.1052984758E+01   0.9867538104E+00   0.6286706237E+00   0.1594127002E+01   0.2831910781E+01
        0.9616808922E+00
  1 LINEAR SEARCH ITERATION,    STATUS FLAG INF = 0   LINEAR SEARCH STEP-SIZE: ALPHA =  1.00000
```

```
SANDIA NATIONAL              TRAJECTORY SIMULATION AND ANALYSIS PROGRAM --- TSAP          PROBLEM:  HYPER-X
LABORATORIES                                (VERSION 89.001)                             PAGE:     008
    ITERATIONS =     7     CALLS OF VF02AD =     8

 X =     0.7090608749E+02   0.9293773940E+02   0.1059346282E+01  -0.2970536388E+02   0.3694610015E+02


 F =    -0.1547804097E+01

 C =    -0.2237575039E-05   0.1725649555E-04   0.8644519804E+00  -0.2417001009E-04   0.2674867489E+00
         0.7336038804E+00   0.1817681249E+00   0.1755106105E+00   0.8587547879E+00   0.3361590043E+00
         0.1052967314E+01   0.9887581715E+00   0.6286829515E+00   0.1594107278E+01   0.2694610015E+01
         0.9630538999E+00
  1 LINEAR SEARCH ITERATION,    STATUS FLAG INF = 0   LINEAR SEARCH STEP-SIZE: ALPHA =  1.00000

    ITERATIONS =     8     CALLS OF VF02AD =     9

 X =     0.7078984119E+02   0.9716430317E+02   0.1059980609E+01  -0.2970042081E+02   0.2852581427E+02


 F =    -0.1548488273E+01

 C =     0.1051507332E-04   0.3644273595E-03   0.9143511040E+00  -0.9397968033E-03  -0.8109820397E-01
         0.7401662894E+00   0.1798306866E+00   0.1768823117E+00   0.9432860633E+00   0.3059692631E+00
         0.1052999030E+01   0.9887502424E+00   0.6287447399E+00   0.1594008416E+01   0.1852581427E+01
         0.9714741857E+00
  1 LINEAR SEARCH ITERATION,    STATUS FLAG INF = 0   LINEAR SEARCH STEP-SIZE: ALPHA =  1.00000

    ITERATIONS =     9     CALLS OF VF02AD =    10

 X =     0.7082348408E+02   0.9657907268E+02   0.1058939927E+01  -0.2970205646E+02   0.3000415196E+02


 F =    -0.1548275512E+01

 C =     0.1108375805E-05   0.5893746646E-07   0.9095042207E+00  -0.2168126345E-04  -0.3611710354E-02
         0.7382849744E+00   0.1803914013E+00   0.1764711154E+00   0.9315814635E+00   0.3101494809E+00
         0.1052946996E+01   0.9887632509E+00   0.6287242943E+00   0.1594041129E+01   0.2000415196E+01
         0.9699958480E+00
  1 LINEAR SEARCH ITERATION,    STATUS FLAG INF = 0   LINEAR SEARCH STEP-SIZE: ALPHA =  1.00000

    ITERATIONS =    10     CALLS OF VF02AD =    11

 X =     0.7082470504E+02   0.9654216243E+02   0.1058289222E+01  -0.2970207513E+02   0.3007444273E+02


 F =    -0.1548269629E+01

 C =    -0.1172368442E-06   0.1569664434E-06   0.9091545805E+00  -0.7425683179E-07  -0.1231653480E-04
         0.7382175211E+00   0.1804117507E+00   0.1764569181E+00   0.9308432487E+00   0.3104131255E+00
         0.1052914461E+01   0.9887713847E+00   0.6287240608E+00   0.1594041503E+01   0.2007444273E+01
         0.9699255573E+00

    THE PRINTING OF THE LAST ITERATION GIVES THE VALUES THAT ARE RETURNED BY SUBROUTINE VF02AD
```

```
SANDIA NATIONAL                TRAJECTORY SIMULATION AND ANALYSIS PROGRAM --- TSAP          PROBLEM:  HYPER-X
LABORATORIES                                  (VERSION 89.001)                              PAGE:     009
   0 LINEAR SEARCH ITERATIONS,    STATUS FLAG INF = 1   LINEAR SEARCH STEP-SIZE: ALPHA =  1.00000

    TOTAL FUNCTION EVALUATIONS      74

    * LOOP-1  SRCH-1      = 90.0000000      (ALTD-0) = 253.218434

    * LOOP-1  SRCH-1      = 130.000000      (ALTD-0) = -1757.7078

    * LOOP-1  SRCH-1      = 95.0368517      (ALTD-0) = 44.5573080

    * LOOP-1  SRCH-1      = 96.0739436      (ALTD-0) = -0.0568405

         ***** SEARCH LOOP 1 CONVERGED *****
```

SANDIA NATIONAL LABORATORIES

TRAJECTORY SIMULATION AND ANALYSIS PROGRAM --- TSAP
(VERSION 89.001)

PROBLEM: HYPER-X
PAGE: 010

TITLE: TWO STAGE ROCKET LAUNCH OF HYPERSONIC VEHICLE

PATH 1

TIME	ALT	LONG	LATGD	VEL	GAMGD	PSIGD	DYNPRS	WT	NX
0.000000	212.000	-120.636180	34.8829100	0.0000	0.0000000	0.0000000	0.0000	14987.1000	-0.4510589
0.097877	212.000	-120.636180	34.8829100	0.0010	70.8247050	-40.0000000	0.0000	14982.8533	1.0164874
0.097877	212.000	-120.636180	34.8829100	0.0010	70.8247050	-40.0000000	0.0000	14982.8533	1.0164874
0.200000	212.565	-120.636180	34.8829104	16.5983	70.8247054	-40.0000002	0.3254	14943.5793	9.3330754
0.400000	220.787	-120.636187	34.8829164	70.5026	70.8247106	-40.0000025	5.8692	14826.8689	9.3837616
0.522361	230.851	-120.636194	34.8829238	103.6858	70.8247170	-40.0000054	12.6906	14755.5013	9.4267545
0.522361	230.851	-120.636194	34.8829238	103.6858	70.8247170	-40.0000054	12.6906	14755.5013	9.4267545
2.500000	924.396	-120.636810	34.8835155	653.8837	66.7519771	-40.9543497	494.5337	13622.3419	9.4222432
5.000000	3165.956	-120.639022	34.8858149	1281.9108	64.9971679	-40.9004613	1778.3742	12346.0442	7.4289383
7.500000	6590.191	-120.642622	34.8890289	1735.7743	63.7472504	-41.0330680	2939.5942	11343.5906	5.8826465
10.000000	10894.646	-120.647380	34.8935365	2112.6094	62.6901654	-41.0033969	3808.7704	10497.8653	5.0540898
12.500000	15887.221	-120.653138	34.8989911	2390.0051	61.7359090	-41.0185579	4150.9705	9820.7950	4.0477773
15.000000	21412.952	-120.659759	34.9052604	2655.4326	60.8563869	-41.0340329	4258.0772	9207.3057	4.3892846
17.500000	27512.920	-120.667327	34.9124217	2963.7123	60.0463090	-41.0492422	4281.9502	8605.6463	5.0316722
20.000000	34289.111	-120.676002	34.9206249	3326.4177	59.3071316	-41.0643807	4196.7604	8002.0725	5.7005849
22.500000	41854.488	-120.685962	34.9300379	3749.3791	58.6380735	-41.0796431	3777.2267	7392.2647	6.5758810
25.000000	50351.322	-120.697432	34.9408698	4251.1849	58.0378858	-41.0952390	3233.6696	6774.6898	7.5395406
27.500000	59876.115	-120.710575	34.9532739	4717.7453	57.5002838	-41.1113350	2525.6534	6251.7535	4.6138471
30.000000	69998.985	-120.724824	34.9667117	4877.9587	56.9926451	-41.1276550	1656.2035	6015.8344	1.7938511
32.500000	80246.521	-120.739529	34.9805691	4913.1215	56.4883454	-41.1438469	1022.2685	5901.1124	0.8440846
35.000000	90443.378	-120.754445	34.9946149	4893.2269	55.9783886	-41.1597785	623.0317	5843.1401	0.3149762
37.500000	100498.674	-120.769442	35.0087262	4837.3965	55.4579257	-41.1753790	379.2949	5821.1027	-0.0260542
39.500000	108400.924	-120.781438	35.0200070	4779.6002	55.0315968	-41.1875880	254.8482	5817.4423	-0.1344699
39.500000	108400.924	-120.781438	35.0200070	4779.6002	55.0315968	-41.1875880	254.8482	5817.4423	-0.1342159
40.000000	110354.903	-120.784434	35.0228233	4764.5168	54.9234927	-41.1906009	230.5363	5817.4423	-0.1219541
45.000000	129425.380	-120.814335	35.0509103	4622.5591	53.8079952	-41.2198689	89.7859	5817.4423	-0.0494288
50.000000	147671.207	-120.844172	35.0788986	4490.1231	52.6283342	-41.2475952	38.4056	5817.4423	-0.0218636
55.000000	165113.347	-120.873976	35.1068191	4362.8587	51.3812242	-41.2737997	18.2088	5817.4423	-0.0105907
59.558142	180321.644	-120.901127	35.1322251	4250.0314	50.1820867	-41.2963710	10.0000	5817.4423	-0.0058393
59.558142	180321.644	-120.901127	35.1322251	4250.0314	50.1820867	-41.2963710	10.0000	5817.4423	-0.0058393
60.000000	181760.950	-120.903759	35.1346858	4239.2379	50.0625568	-41.2984924	9.4455	5817.4423	-0.0055161
65.000000	197618.697	-120.933527	35.1625061	4118.7980	48.6677612	-41.3216802	4.9500	5817.4423	-0.0028925
70.000000	212689.480	-120.963286	35.1902851	4001.4519	47.1919673	-41.3433681	2.5934	5817.4423	-0.0015153
75.000000	226975.299	-120.993036	35.2180264	3887.2665	45.6300900	-41.3635604	1.3605	5817.4423	-0.0007943
80.000000	240477.663	-121.022780	35.2457329	3776.4000	43.9769079	-41.3822809	0.7125	5817.4423	-0.0004157
85.000000	253197.803	-121.052518	35.2734072	3669.0703	42.2271561	-41.3994731	0.3744	5817.4423	-0.0002184
90.000000	265136.774	-121.082252	35.3010516	3565.5378	40.3756419	-41.4152000	0.2001	5817.4423	-0.0001166
95.000000	276295.514	-121.111982	35.3286683	3466.0973	38.4173903	-41.4294448	0.1091	5817.4423	-0.0000634
96.542162	279579.930	-121.121151	35.3371810	3436.3019	37.7911394	-41.4335397	0.0908	5817.4423	-0.0000527
96.542162	279579.930	-121.121151	35.3371810	3436.3019	37.7911394	-41.4335397	0.0908	3842.7000	-0.0000159
100.000000	286674.878	-121.141709	35.3562596	3371.0763	36.3478262	-41.4422102	0.0600	3842.7000	-0.0000109
110.000000	305098.532	-121.201156	35.4113740	3195.7096	31.8598182	-41.4633131	0.0197	3842.7000	-0.0000040
120.000000	320413.173	-121.260599	35.4664113	3042.5003	26.8923489	-41.4785278	0.0078	3842.7000	-0.0000017
130.000000	332623.280	-121.320044	35.5213878	2914.7503	21.4503784	-41.4878711	0.0037	3842.7000	-0.0000009
140.000000	341732.418	-121.379495	35.5763198	2815.7683	15.5734977	-41.4913572	0.0021	3842.7000	-0.0000006
150.000000	347743.241	-121.438957	35.6312233	2748.5434	9.3434623	-41.4889982	0.0014	3842.7000	-0.0000004

TITLE: TWO STAGE ROCKET LAUNCH OF HYPERSONIC VEHICLE

TIME	ALT	LONG	LATGD	VEL	GAMGD	PSIGD	DYNPRS	WT	NX
156.542162	349999.974	-121.477867	35.6671344	2722.8801	5.1344438	-41.4842968	0.0013	3842.7000	-0.0000004
156.542162	349999.974	-121.477867	35.6671344	2722.8801	5.1344438	-41.4842968	0.0013	3840.7000	0.0000000
158.000000	350324.050	-121.486691	35.6753544	2858.4501	4.0363608	-40.8992972	0.0014	3783.6542	3.5166392
160.000000	350669.161	-121.499453	35.6875683	3100.4222	2.6636085	-39.9994030	0.0016	3685.4151	4.2141794
162.000000	350900.830	-121.513092	35.7010341	3373.0728	1.4896666	-39.1511026	0.0018	3578.8358	4.5259426
164.000000	351020.403	-121.527690	35.7158678	3673.3065	0.4993816	-38.3697371	0.0022	3465.8336	4.9629034
166.000000	351029.452	-121.543339	35.7322009	4004.0566	-0.3265179	-37.6493152	0.0026	3346.0778	5.4271366
168.000000	350929.733	-121.560139	35.7501723	4365.2679	-1.0073340	-36.9904780	0.0031	3220.4645	5.8882804
170.000000	350723.135	-121.578188	35.7699199	4756.6657	-1.5622902	-36.3918816	0.0037	3089.9992	6.3421259
172.000000	350411.714	-121.597583	35.7915820	5178.3522	-2.0091687	-35.8501046	0.0044	2955.5814	6.8039292
174.000000	349997.726	-121.618426	35.8152984	5630.5589	-2.3638463	-35.3609586	0.0054	2818.0692	7.2693867
176.000000	349483.629	-121.640817	35.8412086	6112.2497	-2.6410435	-34.9211587	0.0065	2678.7727	7.7145313
178.000000	348872.047	-121.664852	35.8694443	6621.0681	-2.8543393	-34.5277790	0.0079	2539.1606	8.0998476
180.000000	348165.722	-121.690616	35.9001234	7153.7890	-3.0158201	-34.1776889	0.0096	2400.8349	8.4821834
182.000000	347367.552	-121.718189	35.9333529	7708.4010	-3.1348676	-33.8685581	0.0116	2264.8586	8.7548484
184.000000	346480.524	-121.747638	35.9692208	8278.8252	-3.2214601	-33.5921537	0.0141	2133.0611	8.9658723
186.000000	345507.669	-121.779016	36.0077934	8863.0593	-3.2815866	-33.3499573	0.0170	2006.0381	9.1740609
188.000000	344452.233	-121.812371	36.0491316	9459.4513	-3.3198884	-33.1359369	0.0206	1884.1845	9.3471817
190.000000	343317.882	-121.847761	36.0933091	10081.0974	-3.3329555	-32.9420895	0.0248	1765.0334	10.0026482
192.000000	342109.834	-121.885322	36.1405048	10750.2744	-3.3137897	-32.7606031	0.0300	1645.1459	10.8073209
194.000000	340834.548	-121.925230	36.1909585	11473.5460	-3.2639340	-32.5905497	0.0365	1524.6711	11.6929197
196.000000	339499.523	-121.967688	36.2449388	12258.3082	-3.1848242	-32.4312292	0.0447	1403.8753	12.7327816
198.000000	338113.553	-122.012924	36.3027529	13111.6585	-3.0782845	-32.2823607	0.0550	1283.3206	13.8081922
200.000000	336686.616	-122.061182	36.3647238	14034.5075	-2.9484560	-32.1448453	0.0680	1164.5420	14.9286737
202.000000	335230.092	-122.112724	36.4311957	15034.3555	-2.7968490	-32.0182873	0.0844	1048.2044	15.5344179
204.000000	333739.750	-122.166996	36.5013685	15412.5073	-2.8379579	-31.9899074	0.0961	1007.5013	0.1614915
204.942162	333011.548	-122.192738	36.5346257	15416.1022	-2.9103972	-31.9999962	0.1000	1007.2752	0.0000000
204.942162	333011.548	-122.192738	36.5346257	15416.1022	-2.9103972	-31.9999962	0.1000	596.8000	0.0000000
204.942162	333011.548	-122.192738	36.5346257	15416.1022	-2.9103972	-31.9999962	0.1000	596.8000	0.0000000
210.000000	328784.858	-122.331287	36.7131416	15424.6299	-3.3026203	-32.0558781	0.1262	596.8000	0.0000000
220.000000	318851.800	-122.606977	37.0659731	15444.6704	-4.0768455	-32.1675553	0.2174	596.8000	0.0000000
230.000000	306823.728	-122.885088	37.4186978	15468.9340	-4.8490990	-32.2808877	0.4194	596.8000	0.0000000
235.016605	300000.047	-123.025549	37.5956240	15482.6963	-5.2356496	-32.3383935	0.6130	596.8000	0.0000000
235.016605	300000.047	-123.025549	37.5956240	15482.6963	-5.2356496	-32.3383935	0.6130	596.8000	-0.0002132
240.000000	292699.048	-123.165719	37.7713741	15497.3613	-5.6186701	-32.3957652	0.9144	596.8000	-0.0003180
245.000000	284850.461	-123.307011	37.9477121	15513.0978	-6.0021194	-32.4536836	1.4029	596.8000	-0.0004878
250.000000	276477.772	-123.448969	38.1240595	15529.8398	-6.3845697	-32.5119071	2.1695	596.8000	-0.0007544
255.000000	267581.179	-123.591607	38.3004234	15547.5597	-6.7658238	-32.5703565	3.3779	596.8000	-0.0011746
260.000000	258161.158	-123.734936	38.4768108	15566.2111	-7.1455768	-32.6288883	5.3316	596.8000	-0.0018539
265.000000	248218.674	-123.878967	38.6532286	15585.7171	-7.5233537	-32.6872567	8.5182	596.8000	-0.0029621
270.000000	237755.511	-124.023708	38.8296836	15605.9504	-7.8983961	-32.7450463	13.7574	596.8000	-0.0047839
275.000000	226774.825	-124.169167	39.0061821	15626.7054	-8.2695074	-32.8015779	22.1728	596.8000	-0.0077102
280.000000	215281.929	-124.315343	39.1827302	15647.6615	-8.6348526	-32.8557872	35.7282	596.8000	-0.0124239
285.000000	203285.640	-124.462229	39.3593333	15668.2943	-8.9914562	-32.9059225	57.6118	596.8000	-0.0200335
290.000000	190800.472	-124.609804	39.5359963	15687.7614	-9.3345697	-32.9491623	92.8284	596.8000	-0.0322795
295.000000	177850.083	-124.758027	39.7127224	15704.7198	-9.6566553	-32.9809992	149.2071	596.8000	-0.0518842
300.000000	164472.698	-124.906819	39.8895135	15717.0133	-9.9456333	-32.9941815	242.1126	596.8000	-0.0841905
305.000000	150732.829	-125.056042	40.0663681	15720.3001	-10.1769609	-32.9740067	414.1629	596.8000	-0.1440181
310.000000	136751.216	-125.205421	40.2432775	15705.3012	-10.2977799	-32.8888743	749.9573	596.8000	-0.2607849

SANDIA NATIONAL LABORATORIES

TRAJECTORY SIMULATION AND ANALYSIS PROGRAM --- TSAP
(VERSION 89.001)

PROBLEM: HYPER-X
PAGE: 012

TITLE: TWO STAGE ROCKET LAUNCH OF HYPERSONIC VEHICLE

TIME	ALT	LONG	LATGD	VEL	GAMGD	PSIGD	DYNPRS	WT	NX
315.000000	122762.182	-125.354412	40.4202167	15654.2075	-10.2060209	-32.6778017	1392.0177	596.8000	-0.4840505
320.000000	109212.131	-125.501947	40.5971183	15533.8477	-9.7089178	-32.2259515	2588.7093	596.8000	-0.9001796
325.000000	96911.538	-125.646031	40.7737957	15292.1316	-8.4949798	-31.3486827	4484.1874	596.8000	-1.5592998
330.000000	87087.383	-125.783463	40.9497621	14882.0476	-6.2586672	-29.8661327	6759.5569	596.8000	-2.3505209
335.000000	81105.122	-125.910284	41.1239784	14308.3878	-2.9712478	-27.7602694	8319.5256	596.8000	-2.8929734
338.931571	79656.742	-126.000504	41.2587514	13804.9363	0.0000319	-25.8666520	8303.2860	596.8000	-2.8873264
338.931571	79656.742	-126.000504	41.2587514	13804.9363	0.0000319	-25.8666520	8303.2860	596.8000	-2.8873264
340.000000	79756.737	-126.023480	41.2949423	13670.3181	0.7676380	-25.2978990	8103.0212	596.8000	-2.8176877
342.500000	80651.574	-126.074310	41.3789796	13370.1028	2.1722578	-23.6472502	7424.3202	596.8000	-2.5816810
345.000000	82157.552	-126.120646	41.4623035	13097.2772	2.9400556	-21.7515564	6627.1803	596.8000	-2.3044892
347.500000	83886.940	-126.162447	41.5450639	12853.8940	3.0727946	-19.8384995	5875.4298	596.8000	-2.0430807
350.000000	85504.408	-126.200043	41.6273174	12637.1341	2.6942225	-18.0667442	5256.3277	596.8000	-1.8277985
352.500000	86853.063	-126.233886	41.7090522	12441.9221	2.2377916	-16.4461250	4777.6809	596.8000	-1.6613572
355.000000	87937.523	-126.264332	41.7902426	12263.4989	1.7871557	-14.9383642	4407.9989	596.8000	-1.5328066
357.500000	88768.497	-126.291656	41.8708680	12098.1626	1.3398080	-13.5163937	4123.7146	596.8000	-1.4339516
360.000000	89355.029	-126.316069	41.9509104	11942.9732	0.9022719	-12.1571851	3908.1650	596.8000	-1.3589979
362.500000	89725.113	-126.337726	42.0303514	11795.5283	0.5410741	-10.8380755	3745.8654	596.8000	-1.3025609
365.000000	89928.155	-126.356747	42.1091710	11653.9738	0.2662019	-9.5500820	3621.4356	596.8000	-1.2592926
367.500000	90010.781	-126.373233	42.1873471	11516.9797	0.0738253	-8.2896839	3522.9569	596.8000	-1.2250483
368.931571	90020.719	-126.381570	42.2318142	11440.2170	-0.0002847	-7.5800629	3474.5125	596.8000	-1.2082026
368.931571	90020.719	-126.381570	42.2318142	11440.2170	-0.0002847	-7.5800629	3474.5125	596.8000	-1.4923020
370.000000	90020.694	-126.387239	42.2648437	11378.1650	-0.0000338	-7.0142558	3436.9272	596.8000	-1.1075448
375.000000	90020.691	-126.410522	42.4180811	11208.5377	0.0000001	-6.0735690	3335.2151	596.8000	-0.8931025
380.000000	90020.691	-126.431846	42.5693682	11062.0703	-0.0000001	-5.9239173	3248.6189	596.8000	-0.8641441
385.000000	90020.691	-126.453658	42.7185921	10907.4150	-0.0000011	-6.6026089	3158.4181	596.8000	-0.9188493
390.000000	90020.691	-126.478815	42.8652643	10738.8844	0.0000001	-7.8040015	3061.5706	596.8000	-0.8983063
395.000000	90020.811	-126.507795	43.0092479	10574.6634	0.0041490	-8.9851796	2968.6334	596.8000	-0.8431987
400.000000	90025.956	-126.539055	43.1509038	10439.2827	0.0000073	-9.2301758	2892.4028	596.8000	-0.7772533
405.000000	90025.957	-126.568941	43.2908071	10278.1248	-0.0000009	-8.1485299	2803.7883	596.8000	-0.9677318
410.000000	90025.957	-126.592484	43.4286710	10066.3605	0.0000001	-6.0396405	2689.4434	596.8000	-0.9489401
415.000000	90025.957	-126.608683	43.5642021	9863.2981	-0.0000001	-3.9990824	2582.0329	596.8000	-0.7687044
420.000000	90029.622	-126.620588	43.6977556	9739.2448	0.0000086	-3.6224037	2517.0538	596.8000	-0.6838680
425.000000	90029.622	-126.632781	43.8298425	9613.9107	-0.0000004	-4.2064472	2452.6867	596.8000	-0.7209512
430.000000	90029.622	-126.647519	43.9595887	9480.1071	-0.0000003	-5.1758254	2384.8903	596.8000	-0.7063535
435.000000	90029.635	-126.665109	44.0875251	9349.0962	0.0009097	-6.1307776	2319.4281	596.8000	-0.6697828
440.000000	89987.468	-126.684495	44.2136398	9232.3803	-0.4699633	-6.3726131	2268.4070	596.8000	-0.6234214
445.000000	88971.706	-126.703888	44.3382148	9130.9904	-2.0612459	-6.4007530	2326.4387	596.8000	-0.6261764
450.000000	86704.599	-126.723269	44.4612870	9028.8850	-3.6730800	-6.4935872	2533.8670	596.8000	-0.6798135
455.000000	83187.492	-126.742814	44.5826923	8919.1980	-5.3293533	-6.6600076	2925.2947	596.8000	-0.7803229
460.000000	78465.577	-126.762642	44.7021667	8788.6247	-6.5511353	-6.7315009	3564.1887	596.8000	-0.9537390
465.000000	73464.231	-126.781631	44.8196709	8633.6294	-6.6512787	-6.4086342	4382.1721	596.8000	-1.1590866
470.000000	68464.231	-126.799388	44.9349168	8442.6509	-6.8024373	-6.0948142	5348.0534	596.8000	-1.4126682
475.000000	63464.231	-126.815875	45.0473550	8209.1754	-6.9968579	-5.7847522	6442.3991	596.8000	-1.6998277
480.000000	58464.231	-126.831055	45.1563615	7928.8425	-7.2455436	-5.4794146	7631.8364	596.8000	-2.0118027
485.000000	53464.231	-126.844900	45.2612546	7597.7840	-7.5630611	-5.1803437	8900.0798	596.8000	-2.3443957
490.000000	48494.359	-126.857352	45.3613153	7210.8749	-7.5665767	-4.7421316	10167.8740	596.8000	-2.6891237
495.000000	44171.533	-126.867557	45.4559997	6770.7524	-6.6242062	-3.9266696	11024.5739	596.8000	-2.9134265
500.000000	40682.938	-126.875274	45.5447986	6304.0403	-5.5928686	-3.0424128	11294.0672	596.8000	-2.9847578
505.000000	38027.676	-126.880548	45.6274512	5834.6218	-4.4009115	-2.0570116	10986.3727	596.8000	-2.9046118

SANDIA NATIONAL LABORATORIES

TRAJECTORY SIMULATION AND ANALYSIS PROGRAM --- TSAP
(VERSION 89.001)

PROBLEM: HYPER-X
PAGE: 013

TITLE: TWO STAGE ROCKET LAUNCH OF HYPERSONIC VEHICLE

TIME	ALT	LONG	LATGD	VEL	GAMGD	PSIGD	DYNPRS	WT	NX
510.000000	36205.748	-126.883444	45.7040037	5383.7650	-2.9924102	-0.9446764	10206.9609	596.8000	-2.7006222
515.000000	35217.153	-126.884036	45.7747516	4968.6331	-1.3191545	0.3164035	9036.4397	596.8000	-2.3938853
520.000000	35031.763	-126.882437	45.8401611	4600.4783	0.0223833	1.5119689	7802.3146	596.8000	-2.0587856
525.000000	35032.661	-126.879587	45.9009126	4287.7640	0.0000021	2.2666781	6777.4178	596.8000	-1.7906854
530.000000	35032.662	-126.875801	45.9576267	4013.6195	0.0000015	3.0846868	5938.4747	596.8000	-1.5702293
535.000000	35032.662	-126.871122	46.0107730	3771.1864	0.0000014	3.9579081	5242.7422	596.8000	-1.3874061
540.000000	35015.560	-126.865610	46.0607409	3556.6599	-0.4838040	4.7103573	4666.2968	596.8000	-1.2285288
545.000000	34609.385	-126.859818	46.1079433	3371.1757	-2.2332765	5.0562618	4258.0458	596.8000	-1.1218740
550.000000	33720.619	-126.853908	46.1526723	3202.1835	-3.6001899	5.5944570	3974.1728	596.8000	-1.0586003
555.000000	32720.061	-126.847383	46.1950642	3041.5099	-3.7703088	6.6242523	3723.5034	596.8000	-0.9916722
560.000000	31720.079	-126.840117	46.2352496	2890.7274	-3.9672798	7.7018202	3491.8439	596.8000	-0.9340214
565.000000	30720.077	-126.832145	46.2733461	2748.9604	-4.1722334	8.8362391	3277.2109	596.8000	-0.8806073
568.931571	29933.762	-126.825401	46.3019055	2643.3906	-4.3391714	9.7695978	3119.4201	596.8000	-0.8413377

SANDIA NATIONAL LABORATORIES

TRAJECTORY SIMULATION AND ANALYSIS PROGRAM --- TSAP
(VERSION 89.001)

PROBLEM: HYPER-X
PAGE: 014

TITLE: TWO STAGE ROCKET LAUNCH OF HYPERSONIC VEHICLE

PATH 1

TIME	RANGE	ALT	MACH	ALTD	ALFA	BANKGD	GAMGD	PSIGD	DYNPRS
0.000000	0.000000	212.000	0.0000000	0.00000	160.824705	-40.0000000	0.0000000	0.0000000	0.0000
0.097877	0.000000	212.000	0.0000009	0.00094	0.000000	-0.0000048	70.8247050	-40.0000000	0.0000
0.097877	0.000000	212.000	0.0000009	0.00094	0.000000	-0.0000048	70.8247050	-40.0000000	0.0000
0.200000	0.000032	212.565	0.0148779	15.67738	0.000000	0.0000000	70.8247054	-40.0000002	0.3254
0.400000	0.000503	220.787	0.0631969	66.59099	-0.000006	-0.0000048	70.8247106	-40.0000025	5.8692
0.522361	0.001079	230.851	0.0929447	97.93315	-0.000012	0.0000000	70.8247170	-40.0000054	12.6906
0.522361	0.001079	230.851	0.0929447	97.93315	-0.000012	0.0000000	70.8247170	-40.0000054	12.6906
2.500000	0.047766	924.396	0.5875509	600.79155	0.300000	0.0000000	66.7519771	-40.9543497	494.5337
5.000000	0.214337	3165.956	1.1609047	1161.77897	0.252545	0.0000000	64.9971679	-40.9004613	1778.3742
7.500000	0.485251	6590.191	1.5911812	1556.73185	0.121565	0.0000000	63.7472504	-41.0330680	2939.5942
10.000000	0.843108	10894.646	1.9673328	1877.13481	0.017966	0.0000000	62.6901654	-41.0033969	3808.7704
12.500000	1.276116	15887.221	2.2680740	2105.05508	0.000000	0.0000000	61.7359090	-41.0185579	4150.9705
15.000000	1.773922	21412.952	2.5753751	2319.25951	0.000000	0.0000000	60.8563869	-41.0340329	4258.0772
17.500000	2.342683	27512.920	2.9475706	2567.84704	0.000000	0.0000000	60.0463090	-41.0492422	4281.9602
20.000000	2.994348	34289.111	3.4073114	2860.43917	0.000000	0.0000000	59.3071316	-41.0643807	4196.7604
22.500000	3.742295	41854.488	3.8730206	3201.58295	0.000000	0.0000000	58.6380735	-41.0796431	3777.2267
25.000000	4.603187	50351.322	4.3913742	3606.69811	0.000000	0.0000000	58.0378858	-41.0952390	3233.6696
27.500000	5.589271	59876.115	4.8733201	3978.91861	0.000000	0.0000000	57.5002838	-41.1113350	2525.6534
30.000000	6.657801	69998.985	5.0241780	4090.65929	0.000000	0.0000000	56.9926451	-41.1276550	1656.2035
32.500000	7.759978	80246.521	5.0247682	4096.43065	0.000000	0.0000000	56.4883454	-41.1438469	1022.2685
35.000000	8.877409	90443.378	4.9698816	4055.63657	0.000000	0.0000000	55.9783886	-41.1597785	623.0317
37.500000	10.000333	100498.674	4.8802195	3984.61203	0.000000	0.0000000	55.4579257	-41.1753790	379.2949
39.500000	10.898198	108400.924	4.7805203	3916.73049	0.000000	0.0000000	55.0315968	-41.1875880	254.8482
39.500000	10.898198	108400.924	4.7805203	3916.73049	0.000000	0.0000000	55.0315968	-41.1875880	254.8482
40.000000	11.122387	110354.903	4.7485090	3899.21108	0.000000	0.0000000	54.9234927	-41.1906009	230.5363
45.000000	13.358735	129425.380	4.4555852	3730.60263	0.000000	0.0000000	53.8079952	-41.2198689	89.7859
50.000000	15.588220	147671.207	4.2001794	3568.36768	0.000000	0.0000000	52.6283342	-41.2475952	38.4056
55.000000	17.813231	165113.347	4.0321522	3408.77121	0.000000	0.0000000	51.3812242	-41.2737997	18.2088
59.558142	19.838626	180321.644	4.0007860	3264.37839	0.000000	0.0000000	50.1820867	-41.2963710	10.0000
59.558142	19.838626	180321.644	4.0007860	3264.37839	0.000000	-0.2404557	50.1820867	-41.2963710	10.0000
60.000000	20.034830	181760.950	3.9998918	3250.41782	0.121801	-0.2424681	50.0625568	-41.2984924	9.4455
65.000000	22.253566	197618.697	3.9896966	3092.77511	1.542263	-0.2644765	48.6677612	-41.3216802	4.9600
70.000000	24.469779	212689.480	3.9792196	2935.60356	3.043678	-0.2850815	47.1919673	-41.3433681	2.5934
75.000000	26.683709	226975.299	3.9683341	2778.77365	4.631136	-0.3042816	45.6300900	-41.3635604	1.3605
80.000000	28.895538	240477.663	3.9460302	2622.21279	6.309862	-0.3220738	43.9769079	-41.3822609	0.7126
85.000000	31.105419	253197.803	3.9043380	2465.87808	8.085123	-0.3384543	42.2271561	-41.3994731	0.3744
90.000000	33.313486	265136.774	3.8619096	2309.74146	9.962115	-0.3534187	40.3756419	-41.4152000	0.2001
95.000000	35.519864	276295.514	3.8189812	2153.78299	11.945817	-0.3669622	38.4173903	-41.4294448	0.1091
96.542162	36.200061	279579.930	3.8056838	2105.71375	12.579912	-0.3708522	37.7911394	-41.4335397	0.0908
96.542162	36.200061	279579.930	3.8056838	2105.71375	-37.320061	-7.2520616	37.7911394	-41.4335397	0.0908
100.000000	37.724675	286674.878	3.7475231	1997.98874	-35.869563	-7.0218874	36.3478262	-41.4422102	0.0600
110.000000	42.130067	305098.532	3.5281682	1686.83236	-31.349086	-6.2716078	31.8598182	-41.4633131	0.0197
120.000000	46.530576	320413.173	3.3123742	1376.17038	-26.328833	-5.3870427	26.8923489	-41.4785278	0.0078
130.300000	50.927106	332623.280	3.0977466	1065.91046	-20.810859	-4.3633915	21.4503784	-41.4878711	0.0037
140.000000	55.320549	341732.418	2.9260098	755.96138	-14.834016	-3.2072029	15.5734977	-41.4913572	0.0021
150.000000	59.711798	347743.241	2.7892638	446.23252	-8.482771	-1.9403533	9.3434623	-41.4889982	0.0014

SANDIA NATIONAL LABORATORIES — TRAJECTORY SIMULATION AND ANALYSIS PROGRAM --- TSAP (VERSION 89.001) — PROBLEM: HYPER-X — PAGE: 015

TITLE: TWO STAGE ROCKET LAUNCH OF HYPERSONIC VEHICLE

TIME	RANGE	ALT	MACH	ALTD	ALFA	BANKGD	GAMGD	PSIGD	DYNPRS
156.542162	62.583861	349999.974	2.7315841	243.67887	-4.186052	-1.0694216	5.1344438	-41.4842968	0.0013
156.542162	62.583861	349999.974	2.7315841	243.67887	-4.238789	-1.3289401	5.1344438	-41.4842968	0.0013
158.000000	63.238542	350324.050	2.8629118	201.20496	-3.098485	-1.0596984	4.0363608	-40.8992972	0.0014
160.000000	64.200089	350669.161	3.0998883	144.08281	-1.675306	-0.7475508	2.6636085	-39.9994030	0.0016
162.000000	65.246573	350900.830	3.3685848	87.68862	-0.461067	-0.5141224	1.4896666	-39.1511026	0.0018
164.000000	66.386125	351020.403	3.6662287	32.01559	0.562527	-0.3448565	0.4993816	-38.3697371	0.0022
166.000000	67.627921	351029.452	3.9961610	-22.81824	1.417476	-0.2257535	-0.3265179	-37.6493152	0.0026
168.000000	68.981645	350929.733	4.3588302	-76.74311	2.125101	-0.1452394	-1.0073340	-36.9904780	0.0031
170.000000	70.456935	350723.135	4.7545612	-129.68445	2.706143	-0.0937124	-1.5622902	-36.3918816	0.0037
172.000000	72.063437	350411.714	5.1841509	-181.55004	3.179496	-0.0635588	-2.0091687	-35.8501046	0.0044
174.000000	73.810922	349997.726	5.6486217	-232.23352	3.561847	-0.0489037	-2.3638463	-35.3609586	0.0054
176.000000	75.709115	349483.629	6.1478201	-281.64380	3.868511	-0.0451658	-2.6410435	-34.9211587	0.0065
178.000000	77.767223	348872.047	6.6803478	-329.70943	4.113507	-0.0488141	-2.8543393	-34.5277790	0.0079
180.000000	79.993497	348165.722	7.2439927	-376.37292	4.309237	-0.0572151	-3.0158201	-34.1776889	0.0096
182.000000	82.395433	347367.552	7.8378188	-421.54521	4.465318	-0.0685949	-3.1348676	-33.8665581	0.0116
184.000000	84.979244	346480.524	8.4567881	-465.23245	4.591899	-0.0814617	-3.2214601	-33.5921537	0.0141
186.000000	87.749617	345507.669	9.1000125	-507.34971	4.695081	-0.0950536	-3.2815866	-33.3499573	0.0170
188.000000	90.710886	344452.233	9.7659317	-547.80217	4.779597	-0.1088838	-3.3198884	-33.1359369	0.0206
190.000000	93.868270	343317.882	10.4359444	-586.09733	4.842135	-0.1232909	-3.3329555	-32.9420895	0.0248
192.000000	97.234252	342109.834	11.1610048	-621.41211	4.875915	-0.1388001	-3.3137897	-32.7606031	0.0300
194.000000	100.825521	340834.548	11.9486672	-653.25310	4.882771	-0.1550735	-3.2639340	-32.5905497	0.0365
196.000000	104.660800	339499.523	12.8074343	-681.03541	4.864459	-0.1718169	-3.1848242	-32.4312292	0.0447
198.000000	108.761483	338113.553	13.7455705	-704.10074	4.823167	-0.1887411	-3.0782845	-32.2823607	0.0550
200.000000	113.150119	336686.616	14.7648872	-721.90077	4.763409	-0.2054515	-2.9484560	-32.1448453	0.0680
202.000000	117.850740	335230.092	15.8740830	-733.59897	4.687098	-0.2217897	-2.7968490	-32.0182873	0.0844
204.000000	122.808306	333739.750	16.3341504	-763.09571	4.807577	-0.2260877	-2.8379579	-31.9899074	0.0961
204.942162	125.157991	333011.548	16.3679243	-782.73984	4.917585	-0.2251085	-2.9103972	-31.9999962	0.1000
204.942162	125.157991	333011.548	16.3679243	-782.73984	4.917585	-0.2251085	-2.9103972	-31.9999962	0.1000
204.942162	125.157991	333011.548	16.3679243	-782.73984	4.917585	-0.2251085	-2.9103972	-31.9999962	0.1000
210.000000	137.775214	328784.858	16.5543545	-888.60804	5.511542	-0.2197626	-3.3026203	-32.0558781	0.1262
220.000000	162.736742	318851.800	16.8383094	-1098.02881	6.684871	-0.2097566	-4.0768455	-32.1675553	0.2174
230.000000	187.723020	306823.728	17.0509752	-1307.61600	7.856626	-0.2005344	-4.8490990	-32.2808877	0.4194
235.016605	200.268470	300000.047	17.1747616	-1412.83017	8.443765	-0.1962161	-5.2356496	-32.3383935	0.6130
235.016605	200.268470	300000.047	17.1747616	-1412.83017	8.000000	30.0000000	-5.2356496	-32.3383935	0.6130
240.000000	212.738806	292699.048	17.2210166	-1517.30316	8.000000	30.0000000	-5.6186701	-32.3957652	0.9144
245.000000	225.259215	284850.461	17.2475424	-1622.13097	8.000000	30.0000000	-6.0021194	-32.4536836	1.4029
250.000000	237.788737	276477.772	17.1157988	-1726.93832	8.000000	30.0000000	-6.3845697	-32.5119071	2.1695
255.000000	250.327939	267581.179	16.9023030	-1831.68375	8.000000	30.0000000	-6.7658238	-32.5703565	3.3779
260.000000	262.877367	258161.158	16.6854343	-1936.29349	8.000000	30.0000000	-7.1455768	-32.6288883	5.3315
265.000000	275.437536	248218.674	16.4660354	-2040.64252	8.000000	30.0000000	-7.5233537	-32.6872567	8.5182
270.000000	288.008901	237755.511	16.2448488	-2144.52005	8.000000	30.0000000	-7.8983961	-32.7450463	13.7574
275.000000	300.591821	226774.825	15.9465781	-2247.58213	8.000000	30.0000000	-8.2695074	-32.8015779	22.1728
280.000000	313.186508	215281.929	15.6333783	-2349.28934	8.000000	30.0000000	-8.6348526	-32.8557872	35.7282
285.000000	325.792936	203285.640	15.3254676	-2448.75357	8.000000	30.0000000	-8.9914562	-32.9059225	57.6118
290.000000	338.410698	190800.472	15.0228885	-2544.54260	8.000000	30.0000000	-9.3345697	-32.9491623	92.8284
295.000000	351.038778	177850.083	14.7253020	-2634.36685	8.000000	30.0000000	-9.6566553	-32.9809992	149.2071
300.000000	363.675207	164472.698	14.5256575	-2714.54253	8.000000	30.0000000	-9.9456333	-32.9941815	242.1126
305.000000	376.316294	150732.829	14.6340136	-2777.60367	8.000000	30.0000000	-10.1769609	-32.9740067	414.1629
310.000000	388.954672	136751.216	14.9536454	-2807.54391	8.000000	30.0000000	-10.2977799	-32.8888743	749.9673

SANDIA NATIONAL LABORATORIES

TRAJECTORY SIMULATION AND ANALYSIS PROGRAM --- TSAP
(VERSION 89.001)

PROBLEM: HYPER-X
PAGE: 016

TITLE: TWO STAGE ROCKET LAUNCH OF HYPERSONIC VEHICLE

TIME	RANGE	ALT	MACH	ALTD	ALFA	BANKGD	GAMGD	PSIGD	DYNPRS
315.000000	401.575443	122762.182	15.2620465	-2773.74025	8.000000	30.0000000	-10.2060209	-32.6778017	1392.0177
320.000000	414.149070	109212.131	15.5138566	-2619.67154	8.000000	30.0000000	-9.7089178	-32.2259515	2588.7093
325.000000	426.618377	96911.538	15.4644196	-2258.99580	8.000000	30.0000000	-8.4949798	-31.3486827	4484.1874
330.000000	438.885650	87087.383	15.1495038	-1622.39983	8.000000	30.0000000	-6.2586672	-29.8661327	6759.5589
335.000000	450.815034	81105.122	14.6249450	-741.67267	8.000000	30.0000000	-2.9712478	-27.7602694	8319.5256
338.931571	459.871846	79656.742	14.1243438	0.00768	8.000000	30.0000000	0.0000319	-25.8666520	8303.2860
338.931571	459.871846	79656.742	14.1243438	0.00768	8.000000	30.0000000	0.0000319	-25.8666520	8303.2860
340.000000	462.278187	79756.737	13.9856532	183.14686	8.000000	37.0595336	0.7676380	-25.2978990	8103.0212
342.500000	467.816873	80651.574	13.6701394	506.77996	8.000000	53.5780195	2.1722578	-23.6472502	7424.3202
345.000000	473.234035	82157.552	13.3774223	671.77421	8.000000	70.0965054	2.9400556	-21.7515564	6627.1803
347.500000	478.543123	83886.940	13.1133689	689.02884	8.000000	86.6149913	3.0727946	-19.8384995	5875.4298
350.000000	483.758469	85504.408	12.8780647	594.01767	8.000000	96.0739436	2.6942225	-18.0687442	5256.3277
352.500000	488.891196	86853.063	12.6675361	485.81853	8.000000	96.0739436	2.2377916	-16.4461250	4777.6809
355.000000	493.948858	87937.523	12.4767116	382.45798	8.000000	96.0739436	1.7871557	-14.9383642	4407.9989
357.500000	498.937243	88768.497	12.3015861	282.87839	8.000000	96.0739436	1.3398080	-13.5163937	4123.7146
360.000000	503.860810	89355.029	12.1389764	188.06558	8.000000	94.3565564	0.9022719	-12.1571851	3908.1650
362.500000	508.722909	89725.113	11.9861168	111.38970	8.000000	90.3380705	0.5410741	-10.8380755	3745.8654
365.000000	513.526063	89928.155	11.8406527	54.14534	8.000000	86.3195846	0.2662019	-9.5500820	3621.4356
367.500000	518.272262	90010.781	11.7008119	14.83956	8.000000	82.3010987	0.0738253	-8.2896839	3522.9569
368.931571	520.965061	90020.719	11.6227461	-0.05684	8.000000	80.0000000	-0.0002847	-7.5800629	3474.5125
368.931571	520.965061	90020.719	11.6227461	-0.05684	12.080892	80.0000000	-0.0002847	-7.5800629	3474.5125
370.000000	522.962135	90020.694	11.5597042	-0.00671	6.634916	71.4525658	-0.0000338	-7.0142558	3436.9272
375.000000	532.211369	90020.691	11.3873704	0.00001	2.539273	31.4525658	0.0000001	-6.0735690	3335.2151
380.000000	541.334799	90020.891	11.2385661	-0.00002	2.293497	-12.8211513	-0.0000001	-5.9239173	3248.6189
385.000000	550.337493	90020.691	11.0814433	-0.00020	4.637046	-60.0000000	-0.0000011	-6.6026089	3158.4181
390.000000	559.205418	90020.891	10.9102239	0.00002	4.819045	-60.0000000	0.0000001	-7.8040015	3061.5706
395.000000	567.936312	90020.811	10.7433818	0.76575	4.127166	-47.1788487	0.0041490	-8.9851796	2968.6334
400.000000	576.544165	90025.956	10.6058043	0.00134	2.669739	14.9580099	0.0000073	-9.2301758	2892.4028
405.000000	585.038558	90025.957	10.4420757	-0.00016	7.874813	70.0000000	-0.0000009	-8.1485299	2803.7883
410.000000	593.373314	90025.957	10.2269335	0.00002	8.244531	70.0000000	0.0000001	-6.0396405	2689.4434
415.000000	601.535079	90025.957	10.0206320	-0.00001	5.132932	55.0419901	-0.0000001	-3.9990824	2582.0329
420.000000	609.584004	90029.622	9.8945753	0.00147	3.080948	-10.6842928	0.0000086	-3.6224037	2517.0538
425.000000	617.494117	90029.822	9.7672423	-0.00007	4.857737	-50.0000000	-0.0000004	-4.2064472	2452.6867
430.000000	625.316467	90029.622	9.6313047	-0.00006	5.020923	-50.0000000	-0.0000003	-5.1758254	2384.8903
435.000000	633.029913	90029.635	9.4982042	0.14844	4.480077	-39.3157072	0.0009097	-6.1307776	2319.4281
440.000000	640.642673	89987.466	9.3798936	-75.72688	-3.387426	1.0684293	-0.4699633	-6.3726131	2266.4070
445.000000	648.163618	88971.706	9.2832478	-328.42134	-2.729838	6.0684293	-2.0612459	-6.4007530	2326.4387
450.000000	655.594872	86704.599	9.1935343	-576.42142	-2.610037	11.0684293	-3.6730800	-6.4935872	2533.8670
455.000000	662.927506	83187.492	9.1035804	-828.42133	-2.396824	16.0684293	-5.3293533	-6.6600076	2925.2947
460.000000	670.146428	78465.577	8.9993138	-1002.69341	2.512949	20.0000000	-6.5511353	-6.7315009	3564.1887
465.000000	677.243721	73464.231	8.8711011	-1000.00009	1.654293	20.0000000	-6.6512787	-6.4086342	4382.1721
470.000000	684.200441	68464.231	8.7050104	-999.99999	1.321550	20.0000000	-6.8024373	-6.0948142	5348.0534
475.000000	690.983872	63464.231	8.4798854	-999.99998	1.054802	20.0000000	-6.9968579	-5.7847522	6442.3991
480.000000	697.556817	58464.231	8.1903081	-999.99998	0.842778	20.0000000	-7.2455436	-5.4794146	7631.8364
485.000000	703.878645	53464.231	7.8483325	-999.99998	0.674053	20.0000000	-7.5630611	-5.1803437	8900.0798
490.000000	709.906368	48494.359	7.4486644	-949.51464	1.628073	20.0000000	-7.5665767	-4.7421316	10167.8740
495.000000	715.604842	44171.533	6.9940282	-781.05245	1.434595	20.0000000	-6.6242062	-3.9266696	11024.5739
500.000000	720.943805	40682.938	6.5119256	-614.38564	1.444006	20.0000000	-5.5928686	-3.0424128	11294.0672
505.000000	725.909031	38027.676	6.0270273	-447.71906	1.544083	20.0000000	-4.4009115	-2.0570116	10986.3727

SANDIA NATIONAL LABORATORIES — TRAJECTORY SIMULATION AND ANALYSIS PROGRAM --- TSAP (VERSION 89.001) — PROBLEM: HYPER-X PAGE: 017

TITLE: TWO STAGE ROCKET LAUNCH OF HYPERSONIC VEHICLE

TIME	RANGE	ALT	MACH	ALTD	ALFA	BANKGD	GAMGD	PSIGD	DYNPRS
510.000000	730.504895	36205.748	5.5613028	-281.05229	1.734925	20.0000000	-2.9924102	-0.9446764	10206.9609
515.000000	734.750929	35217.153	5.1107613	-114.38566	2.043343	20.0000000	-1.3191545	0.3164035	9036.4397
520.000000	738.677130	35031.763	4.7281163	1.79563	1.070881	20.0000000	0.0223633	1.5119689	7802.3146
525.000000	742.325153	35032.661	4.4067437	0.00015	1.384744	20.0000000	0.0000021	2.2666781	6777.4178
530.000000	745.732629	35032.662	4.1249922	0.00011	1.575191	20.0000000	0.0000015	3.0846868	5938.4747
535.000000	748.928334	35032.662	3.8758319	0.00009	1.779399	20.0000000	0.0000014	3.9579081	5242.7422
540.000000	751.936135	35015.560	3.6550706	-30.03198	0.555748	20.0000000	-0.4838040	4.7103573	4666.2968
545.000000	754.779429	34609.385	3.4581246	-131.36851	0.725156	20.0000000	-2.2332765	5.0562618	4258.0458
550.000000	757.475299	33720.619	3.2717314	-201.07736	2.344205	20.0000000	-3.6001899	5.5944570	3974.1728
555.000000	760.034149	32720.081	3.0937961	-200.00002	2.338359	20.0000000	-3.7703088	6.6242523	3723.5034
560.000000	762.465002	31720.079	2.9275103	-200.00012	2.498728	20.0000000	-3.9672798	7.7018202	3491.8439
565.000000	764.775541	30720.077	2.7718207	-200.00026	2.667521	20.0000000	-4.1722334	8.8362391	3277.2109
568.931571	766.512511	29933.762	2.6583150	-200.00014	2.806366	20.0000000	-4.3391714	9.7695978	3119.4201

SANDIA NATIONAL LABORATORIES — TRAJECTORY SIMULATION AND ANALYSIS PROGRAM --- TSAP (VERSION 89.001) — PROBLEM: HYPER-X PAGE: 018

***** JOB COMPLETE *****

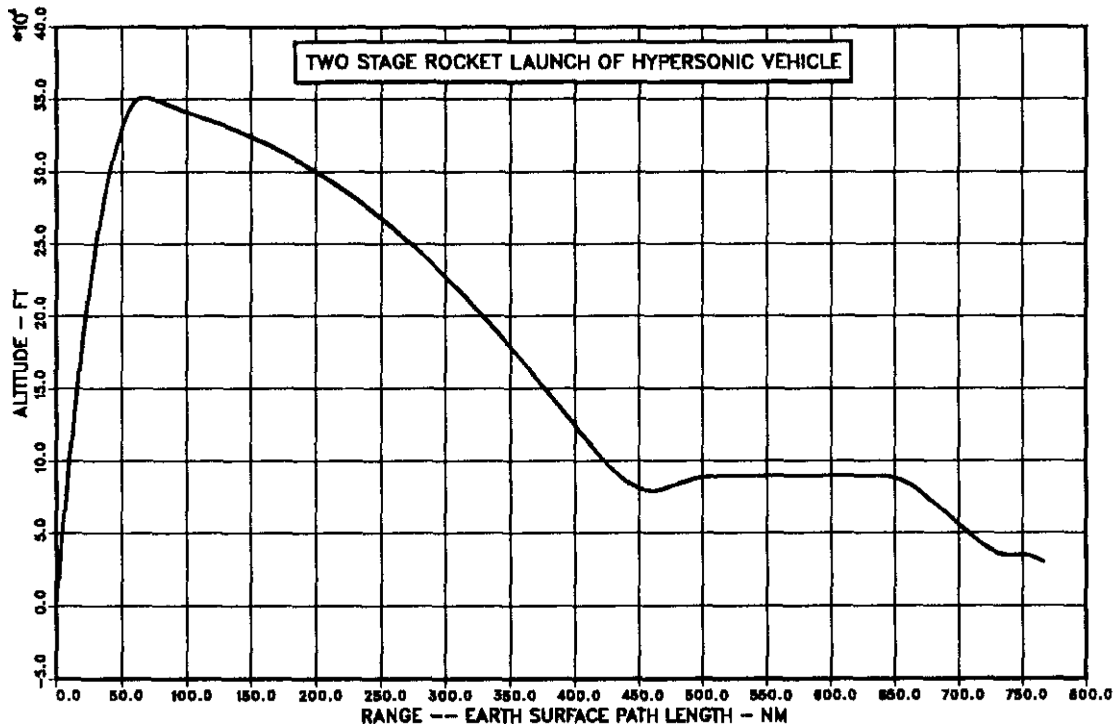

Figure 32: Range versus Altitude History Plot for the Example Problem

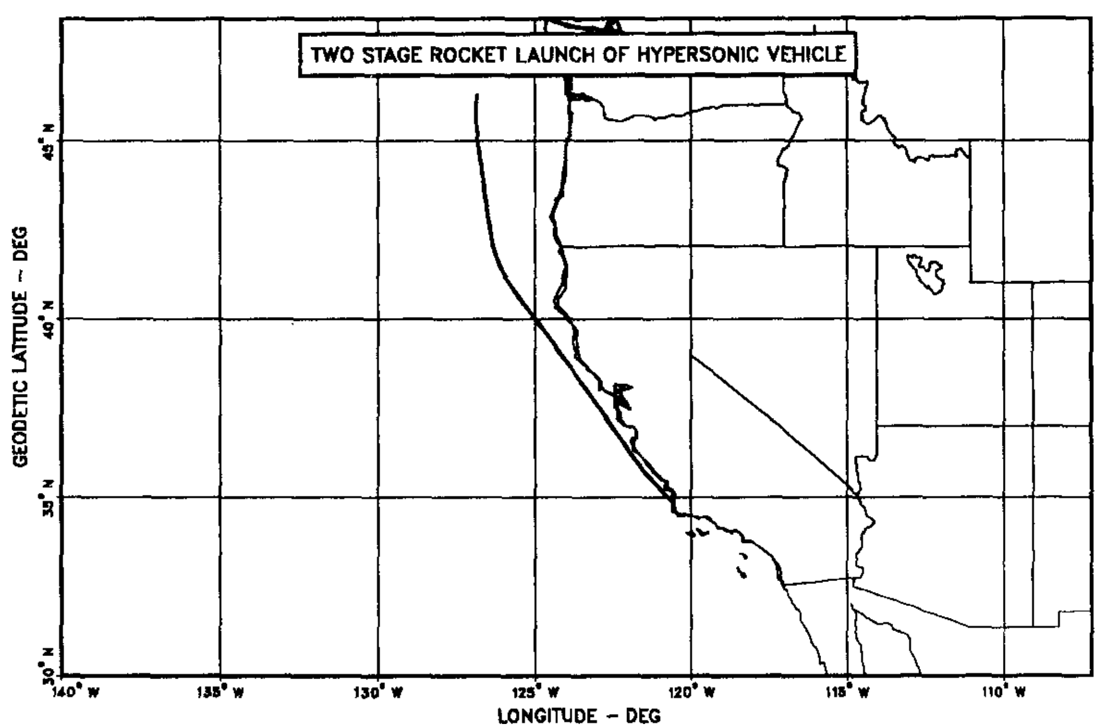

Figure 33: Ground Track Plot for the Example Problem

B Standard Output Variable Summary

Name	Symbol	Section	Description
ALFA	α	2.1.10, 4.8	angle of attack (deg)
ALFAT	α_T	2.1.10, 4.8	total angle of attack (deg)
ALT	h	2.1.3	geodetic altitude (ft)
ALTD	$\dot{h}$	2.2.1	geodetic altitude rate $\dot{h} = -{}^{\oplus}\vec{V} \cdot \hat{z}_{gd}$ (ft/sec)
ALTDD	$\ddot{h}$	2.2.1	vertical (geodetic) acceleration in the earth fixed frame $\ddot{h} = -{}^{\oplus}\vec{a} \cdot \hat{z}_{gd} - {}^{\oplus}\vec{V} \cdot \dot{\hat{z}}_{gd}$ (ft/sec^2)
BALISTIC	β_c		ballistic coefficient $\beta_c = W/C_D S_{ref}$ (lb/ft^2)
BANKGC	μ_{gc}	2.1.10, 4.8	bank angle relative to local geocentric (deg)
BANKGD	μ_{gd}	2.1.10, 4.8	bank angle relative to local geodetic (deg)
BETA	β	2.1.10, 4.8	sideslip angle (deg)
BETAE	β_E	2.1.10, 4.8	Euler sideslip angle (deg)
CA	C_A	2.3.1, 3.2	aerodynamic axial force coefficient
CD	C_D	2.3.1, 3.2	aerodynamic drag force coefficient
CL	C_L	2.3.1, 3.2	aerodynamic lift force coefficient
CM	C_m	3.2	aerodynamic moment coefficient
CMA	C_{m_α}	3.2	aerodynamic moment per deg α_T coefficient
CN	C_N	2.3.1, 3.2	aerodynamic normal force coefficient
CRSRNG		4.5	range normal to specified great circle route (nm)
CS	C_S	2.3.1, 3.2	aerodynamic side force coefficient
CX	C_x	2.3.1, 3.2	aerodynamic vehicle x-axis force coefficient

CY	C_y	2.3.1, 3.2	aerodynamic vehicle y-axis force coefficient
CZ	C_z	2.3.1, 3.2	aerodynamic vehicle z-axis force coefficient
DWNRNG		4.5	range in direction of specified great circle route (nm)
DYNPRS	q	2.3.1	dynamic pressure $q = 0.5\rho^w\vec{V}^2$ (lb/ft^2)
EAST		4.5	distance east of reference point over earth surface (nm)
EPS1	ϵ_1	4.20	thrust vector meridinal angle (deg)
EPS2	ϵ_2	4.20	thrust vector off-axis angle (deg)
FUEL	w_f	4.10	mass (weight) of fuel used (positive for mass lost); mass change due to mass rate terms (lb$_m$)
GAMGC	γ_{gc}	2.1.3	vertical flight path angle with respect to the local geocentric horizontal (deg)
GAMGCD	$\dot{\gamma}_{gc}$	2.2.3	rate of change of γ_{gc} (deg/sec)
GAMGD	γ_{gd}	2.1.4	vertical flight path angle with respect to the local geodetic horizontal (deg)
GAMGDD	$\dot{\gamma}_{gd}$	2.2.3	rate of change of γ_{gd} (deg/sec)
L/D	C_L/C_D	2.3.1, 3.2	lift coefficient over drag coefficient
LATGC	δ_{gc}	2.1.3	geocentric latitude (deg)
LATGCD	$\dot{\delta}_{gc}$	2.2.1	rate of change of δ_{gc} (deg/sec)
LATGD	δ_{gd}	2.1.4	geodetic latitude (deg)
LATGDD	$\dot{\delta}_{gd}$	2.2.1	rate of change of δ_{gd} (deg/sec)
LONG	λ	2.1.4	longitude (deg)
LONGD	$\dot{\lambda}$	2.2.1	rate of change of λ (deg/sec)
LREF	L_{ref}	3.2, 4.1	aerodynamic reference length (ft)
MACH	M		Mach number $M = {}^wV/a$

MASS	m	4.10, 4.12	vehicle mass (slugs)
MASSDT	$\dot{m}$	3.2	mass rate of change (slugs/sec)
NORTH		4.5	distance north of reference point over earth surface (nm)
NTOTAL	N_T	2.2.4	total specific loading magnitude (g)
NX	N_x	2.2.4	specific loading along vehicle x-axis (g)
NY	N_y	2.2.4	specific loading along vehicle y-axis (g)
NZ	N_z	2.2.4	specific loading along vehicle z-axis (g)
PATHLENG	$\int^{\oplus} V dt$	2.2	trajectory pathlength (ft)
PATHLMAR	$\int^{\oplus} V dt$	4.10	trajectory pathlength since last mark (ft)
PATHLSEG	$\int^{\oplus} V dt$	4.10	trajectory pathlength for current segment (ft)
PHI	ϕ_w	2.1.10, 4.1	windward meridian (deg)
PHIWBA	ϕ_{wba}	2.1.10, 4.1	windward meridian from body y-axis (deg)
PITCHGC	Θ_{gc}	2.1.8, 4.8	pitch Euler angle with respect to the geocentric horizon (deg)
PITCHGD	Θ_{gd}	2.1.8, 4.8	pitch Euler angle with respect to the geodetic horizon (deg)
PITCHI	Θ_I	2.1.8, 4.8	pitch Euler angle with respect to the inertial platform horizon (deg)
PRESS	P_∞	4.2	freestream atmospheric pressure (lb/ft^2)
PSIGC	ψ_{gc}	2.1.3	geocentric horizontal flight path angle (deg)
PSIGCD	$\dot{\psi}_{gc}$	2.2.3	rate of change of the geocentric horizontal flight path angle (deg/sec)
PSIGD	ψ_{gd}	2.1.4	geodetic horizontal flight path angle (deg)
PSIGDD	$\dot{\psi}_{gd}$	2.2.3	rate of change of the geodetic horizontal flight path angle (deg/sec)
QWALL-1	Q_1	4.9	heating rate (or auxillary table variable) one

QWALL-2	Q_2	4.9	heating rate (or auxillary table variable) two
QWALL-3	Q_3	4.9	heating rate (or auxillary table variable) three
RADAZIM	α_r	4.21	azimuth from radar station to vehicle (deg)
RADAZD	$\dot{\alpha}_r$	4.21	rate of change of azimuth from radar station to vehicle (deg/sec)
RADAZDD	$\ddot{\alpha}_r$	4.21	second time derivative of azimuth from radar station to vehicle (deg/sec^2)
RADELEV	ε_r	4.21	elevation from radar station to vehicle (deg)
RADELD	$\dot{\varepsilon}_r$	4.21	rate of change of elevation from radar station to vehicle (deg/sec)
RADELDD	$\ddot{\varepsilon}_r$	4.21	second time derivative of elevation from radar station to vehicle (deg/sec^2)
RADRNG	r_r	4.21	slant range from radar station to vehicle (nm)
RADRNGD	$\dot{r}_r$	4.21	rate of change of slant range from radar station to vehicle (ft/sec)
RADRNGDD	$\ddot{r}_r$	4.21	second time derivative of slant range from radar station to vehicle (ft/sec^2)
RANGE	$\int V_g dt$	2.2.2	length of the flight path projected onto the earth's reference ellipsoid (nm)
RCM	r	2.1.3	radial distance from earth's center to the vehicle (ft)
RCMD	$\dot{r}$		rate of change of r, $\dot{r} = \hat{z}_{gc} \cdot {}^{\oplus}\vec{V}$ (ft/sec)
REYPFT	R_e/L_{ref}		Reynolds number per foot $R_e/L_{ref} = \Vert {}^{w}\vec{V} \Vert / \nu$ (ft^{-1})
RHO	ρ	4.2	freestream atmospheric density (slugs/ft^3)
RNGMARK	$\int V_g dt$	4.10	range since last mark (nm)
RNGSEG	$\int V_g dt$	4.10	range during current segment (nm)
ROLLGC	Φ_{gc}	2.1.8, 4.8	geocentric Euler roll angle (deg)
ROLLGD	Φ_{gd}	2.1.8, 4.8	geodetic Euler roll angle (deg)
ROLLI	Φ_I	2.1.8, 4.8	inertial platform Euler roll angle (deg)

SEGMENT		4.24	current segment number
SNDSPD	a	4.2	freestream speed of sound (ft/sec)
SREF	S_{ref}	3.2, 4.1	aerodynamic reference area (ft^2)
TEMP	T_∞	4.2	freestream atmospheric temperature (deg Rankine)
THRUST	T	4.20	thrust force (lb)
TIME	t	2.2	path time (sec)
TMARK	t_m	4.10	time since last mark (sec)
TRGALT	h_T	4.27	altitude of the target (ft)
TRGAZIM		4.27	geodetic azimuth from the vehicle to the target (deg)
TRGEAST		4.5	distance east over earth surface from the reference point to the target (ft)
TRGELEV		4.27	geodetic elevation of the target with respect to the vehicle (deg)
TRGGAMGC	$\gamma_{T_{gc}}$	4.27	geocentric vertical flight path angle of the target (deg)
TRGGAMGD	$\gamma_{T_{gd}}$	4.27	geodetic vertical flight path angle of the target (deg)
TRGLATGC	$\delta_{T_{gc}}$	4.27	geocentric latitude of the target (deg)
TRGLATGD	$\delta_{T_{gd}}$	4.27	geodetic latitude of the target (deg)
TRGLONG	λ_T	4.27	longitude of the target (deg)
TRGNORTH		4.5	distance north over earth surface from the reference point to the target (ft)
TRGPSIGC	$\psi_{T_{gc}}$	4.27	geocentric horizontal flight path angle of the target (deg)
TRGPSIGD	$\psi_{T_{gd}}$	4.27	geodetic horizontal flight path angle of the target (deg)
TRGRCM	r_T	4.27	radial distance of target to earth's center (ft)

TRGRNG		4.27	distance between the vehicle and the target (ft)
TRGRNGD		4.27	magnitude of relative velocity between primary vehicle and the target (ft/sec)
TRGVEL	${}^{\oplus}V^{\tau}$	4.27	target velocity magnitude (ft/sec)
TRGX	$\hat{x}_{\oplus} \cdot \vec{r}^{\tau}$	4.27	target ECFC position X component (ft)
TRGXD	$\hat{x}_{\oplus} \cdot {}^{\oplus}\vec{V}^{\tau}$	4.27	target ECFC velocity X component (ft/sec)
TRGY	$\hat{y}_{\oplus} \cdot \vec{r}^{\tau}$	4.27	target ECFC position Y component (ft)
TRGYD	$\hat{y}_{\oplus} \cdot {}^{\oplus}\vec{V}^{\tau}$	4.27	target ECFC velocity Y component (ft/sec)
TRGZ	$\hat{z}_{\oplus} \cdot \vec{r}^{\tau}$	4.27	target ECFC position Z component (ft)
TRGZD	$\hat{z}_{\oplus} \cdot {}^{\oplus}\vec{V}^{\tau}$	4.27	target ECFC velocity Z component (ft/sec)
TSEG	t_{seg}	4.10	time of current segment (sec)
TTARG	t_{τ}	4.27	target mission time (sec)
VAIR	$\Vert {}^{w}\vec{V} \Vert$	2.1.9	vehicle speed with respect to the air mass (ft/sec)
VEL	${}^{\oplus}V$	2.1.3	earth relative velocity magnitude (ft/sec)
VELD	${}^{\oplus}\dot{V}$		rate of change of the velocity magnitude ${}^{\oplus}\dot{V} = {}^{\oplus}\vec{a} \cdot {}^{\oplus}\vec{V} / {}^{\oplus}V$ (ft/sec^2)
VGROUND	V_g	2.2.2	ground speed (ft/sec)
VISC	ν	4.2	freestream kinematic viscosity (ft^2/sec)
WINDD	${}^{w}\vec{V} \cdot \hat{z}_{gd}$	4.30	downward component (geodetic or geocentric, as specified in WINDS data block) of wind velocity (ft/sec)
WINDE	${}^{w}\vec{V} \cdot \hat{y}_{gc}$	4.30	eastward component of wind velocity (ft/sec)
WINDN	${}^{w}\vec{V} \cdot \hat{x}_{gd}$	4.30	northward component (geodetic or geocentric, as specified in WINDS data block) of wind velocity (ft/sec)
WT	W	4.10, 4.12	vehicle mass (weight) (lb$_m$)

WTDT	$\dot{W}$	3.2	mass (weight) rate of change (lb_m/sec)
XCG	x_{cg}		center of gravity position in vehicle
XECF	$\hat{x}_\oplus \cdot \vec{r}$	2.1.1	ECFC position X component (ft)
XECFD	$\hat{x}_\oplus \cdot {}^\oplus\vec{V}$	2.1.1	ECFC velocity X component (ft/sec)
XECFDD	$\hat{x}_\oplus \cdot {}^\oplus\vec{a}$	2.1.1	ECFC acceleration X component (ft/sec^2)
XECI	$\hat{x}_I \cdot \vec{r}$	2.1.2	ECIC position X component (ft)
XECID	$\hat{x}_I \cdot {}^I\vec{V}$	2.1.2	ECIC velocity X component (ft/sec)
XECIDD	$\hat{x}_I \cdot {}^I\vec{a}$	2.1.2	ECIC acceleration X component (ft/sec^2)
YAWGC	Ψ_{gc}	2.1.8, 4.8	geocentric pitch Euler angle (deg)
YAWGD	Ψ_{gd}	2.1.8, 4.8	geodetic pitch Euler angle (deg)
YAWI	Ψ_I	2.1.8, 4.8	inertial platform pitch Euler angle (deg)
YECF	$\hat{y}_\oplus \cdot \vec{r}$	2.1.1	ECFC position Y component (ft)
YECFD	$\hat{y}_\oplus \cdot {}^\oplus\vec{V}$	2.1.1	ECFC velocity Y component (ft/sec)
YECFDD	$\hat{y}_\oplus \cdot {}^\oplus\vec{a}$	2.1.1	ECFC acceleration Y component (ft/sec^2)
YECI	$\hat{y}_I \cdot \vec{r}$	2.1.2	ECIC position Y component (ft)
YECID	$\hat{y}_I \cdot {}^I\vec{V}$	2.1.2	ECIC velocity Y component (ft/sec)
YECIDD	$\hat{y}_I \cdot {}^I\vec{a}$	2.1.2	ECIC acceleration Y component (ft/sec^2)
ZECF	$\hat{z}_\oplus \cdot \vec{r}$	2.1.1	ECFC position Z component (ft)
ZECFD	$\hat{z}_\oplus \cdot {}^\oplus\vec{V}$	2.1.1	ECFC velocity Z component (ft/sec)
ZECFDD	$\hat{z}_\oplus \cdot {}^\oplus\vec{a}$	2.1.1	ECFC acceleration Z component (ft/sec^2)
ZECI	$\hat{z}_I \cdot \vec{r}$	2.1.2	ECIC position Z component (ft)
ZECID	$\hat{z}_I \cdot {}^I\vec{V}$	2.1.2	ECIC velocity Z component (ft/sec)

ZECIDD	$\hat{z}_I \cdot {}^I\vec{a}$	2.1.2	ECIC acceleration Z component (ft/sec^2)

Distribution

1510	J. W. Nunziato
1520	L. W. Davison
1530	D. B. Hayes
1550	C. W. Peterson
1551	J. K. Cole
1551	K. V. Chavez
1551	R. A. LaFarge
1551	R. J. Weir
1551	W. P. Wolfe
1552	D. D. McBride
1553	S. McAlees, Jr.
1553	D. P. Kelly
1554	D. P. Aeschliman
1555	W. R. Barton
1555	R. W. Greene
1555	T. M. Jordan
1555	D. L. Keese
1555	W. A. Millard
1555	D. E. Outka (20)
1555	L. R. Rollstin
1555	H. A. Wente
1556	W. L. Oberkampf
3141	C. T. F. (5)
3151	W. I. Klein (3)
3154	C. H. Dalin for DOE/OSTI (8)
7523	P. L. Walter
7525	R. L. Eno
7526	L. W. Lathrop
8131	W. R. Bolton, Jr.
8171	R. N. Everett
8165	K. E. Carbiener
8524	SNLL Technical Library
9014	J. M. Parvin
9015	D. E. Bennett, III
9132	T. M. Criel
9132	R. S. Edmunds
9132	J. C. Gilkey
9132	R. Schindwolf
9132	R. D. Tucker
9142	G. W. Mitchell
9142	R. D. Robinett, III
9144	W. E. Williamson
9144	S. A. Kerr
9144	J. L. McDowell
9144	B. A. Rainwater
9144	D. E. Salguero