

SPACECRAFT TRAJECTORY TARGETING  
BY BOUNDARY-CONDITION ORBIT FITTING

by

DALE GORDON STUART

S.B., Massachusetts Institute of Technology  
(1982)

SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENTS FOR THE  
DEGREE OF

MASTER OF SCIENCE  
IN AERONAUTICS AND ASTRONAUTICS

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY  
June 1984

© Dale G. Stuart

The author hereby grants to M.I.T. permission to reproduce and to  
distribute copies of this thesis document in whole or in part.

Signature of Author \_\_\_\_\_

Department of Aeronautics and Astronautics  
June 4, 1984

Certified by \_\_\_\_\_

Prof. David L. Akin  
Thesis Supervisor

Accepted by \_\_\_\_\_

Prof. Harold Y. Wachman  
Chairman, Departmental Graduate Committee

SPACECRAFT TRAJECTORY TARGETING

BY BOUNDARY-CONDITION ORBIT FITTING

by

DALE GORDON STUART

Submitted to the Department of Aeronautics and Astronautics  
on May 11, 1984 in partial fulfillment of the  
requirements for the degree Master of Science in  
Aeronautics and Astronautics

ABSTRACT

The classic orbital boundary-value problem provides a means of finding feasible orbits that pass through an initial position and a fixed target position. In this thesis, a method is developed that extends the boundary-value problem and allows a choice of any combination of three of the following four target parameters: altitude, downrange angle, velocity, and flight path angle. Special attention is paid to the algorithm allowing choice of target velocity, altitude, and flight path angle, and to the case where the transfer orbit has its apogee at the target altitude. In addition, the targeting technique is adapted for use as a general guidance algorithm for a spacecraft in powered flight, and is implemented in a computer simulation of spacecraft trajectories. Two cases are studied: launch from the earth's surface into Low Earth Orbit (LEO), and transfer from LEO to Geosynchronous Earth Orbit (GEO).

Results from the computer simulations indicate that extremely good targeting accuracy can be achieved, with typically less than 0.001 % error between the actual and desired target values, and resulting delta-v values are very close to optimal for a large number of cases. Overall, the orbit-fitting targeting technique provides a fast and straightforward means of generating a feasible (in some cases close to optimal) trajectory that satisfies a chosen combination of target conditions. The technique is easy to implement, contains no inherent major computational difficulties, and can be applied to a wide range of two-body space flight navigation problems.

Thesis Supervisor: David L. Akin  
Professor of Aeronautics and Astronautics

0750627

## ACKNOWLEDGMENTS

Special thanks to Professor Dave Akin, for being my thesis supervisor, for keeping me on target, and for knowing how to keep me encouraged despite his being the busiest person I know. Thanks also to Professor Battin, who provided inspiration and valuable tips when I most needed them, and whose courses taught me -- always one month too late, however -- how I could have done everything much more easily.

An education at M.I.T. is not provided by the professors alone, and I wish to thank all my fellow friends in the Space Systems Lab -- that's you Mary, Sarah, Janice, Russ, John, Cliff, George, Dan, Ray, Janet, Joe, and too many more to mention -- for reminding me there's a lighter side to M.I.T., for sharing my enthusiasm for space exploration, and for not letting me get hopelessly submerged in work, although I hope you'll forgive me for "always being so organized!" And, of course, to one of the most valuable people in the Aero/Astro department, Susie Fennelly, who knows everything you always wanted to know about surviving The System, many thanks for helping me throughout these years.

My sincere thanks to my parents and sisters, who may have long ago considered me lost in space, but who never ceased providing me support and encouragement all my life. Last, but not at all least, I deeply thank my husband, Dave, to whom I owe more than words can tell.

This work was supported by fellowships from the Fannie and John Hertz Foundation, and from Zonta International.

## TABLE OF CONTENTS

|   | PAGE |
|---|------|
| ABSTRACT .....  | 2    |
| ACKNOWLEDGEMENTS .....  | 3    |
| LIST OF SYMBOLS .....   | 6    |
| LIST OF FIGURES AND TABLES .....  | 9    |
| <br>  |      |
| CHAPTER 1 Introduction .....  | 10   |
| <br>  |      |
| CHAPTER 2 Development of the Orbit-Fitting Targeting Technique                    |      |
| 2.1 Orbital Boundary-Value Problem .....  | 12   |
| 2.2 Solution Technique for Any Choice of Target Parameters .....                  | 15   |
| 2.3 Altitude, Velocity, and Flight Path Angle as Target<br>Parameters .....       | 18   |
| 2.4 Finding the Required Velocity .....   | 19   |
| 2.5 Specialization for Target Point at Apogee .....                               | 25   |
| 2.6 Range of Feasible Velocities .....  | 27   |
| <br>  |      |
| CHAPTER 3 Guidance Algorithm  |      |
| 3.1 Overview .....  | 29   |
| 3.2 Velocity-to-be-Gained Determination .....                                     | 30   |
| 3.3 Thrust Angle Determination .....  | 31   |
| 3.4 Thrust Level Determination .....  | 32   |
| 3.5 Guidance for Infeasible Target Velocities .....                               | 34   |
| <br>  |      |
| CHAPTER 4 Technique Applications and Results                                      |      |
| 4.1 Overview .....  | 37   |
| 4.2 Simulation Technique .....  | 38   |
| 4.3 Orbit-to-Orbit Transfer Trajectories .....                                    | 39   |
| 4.4 Launch Trajectories .....   | 47   |
| <br>  |      |
| CHAPTER 5 Conclusions .....   | 53   |
| <br>  |      |
| APPENDIX A Alternative Choices of Target Parameters                               |      |
| A.1 Altitude, Velocity, and Transfer Angle as Target<br>Parameters .....          | 54   |
| A.2 Velocity, Flight Path Angle, and Transfer Angle as Target<br>Parameters ..... | 56   |
| A.3 Altitude, Flight Path Angle, and Transfer Angle as Target<br>Parameters ..... | 57   |

TABLE OF CONTENTS (cont.)

|   | PAGE |
|---|------|
| APPENDIX B Alternative Technique for Finding the Required<br>Velocity ..... | 58   |
| APPENDIX C Spacecraft Equations of Motion .....                             | 60   |
| APPENDIX D Tangential Thrust Transfers .....                                | 63   |
| APPENDIX E Sample Trajectories  |      |
| E.1 LEO to GEO Transfer .....   | 65   |
| E.2 Earth Launch to LEO .....   | 70   |
| APPENDIX F Computer Program Listing .....                                   | 72   |
| LIST OF REFERENCES .....  | 91   |

## LIST OF SYMBOLS

Variables in parentheses are the variable names used in the computer program.

|           |   |   |
|-----------|---|---|
| A         | - | perpendicular distance from $V_2$ to the radius |
| a         | - | semi-major axis of an elliptical orbit (SMA)    |
| a         | - | coefficient of a term in a polynomial           |
| $a_1$     | - | coefficient of a term in a polynomial           |
| $a_2$     | - | coefficient of a term in a polynomial           |
| $a_{max}$ | - | maximum feasible semi-major axis (Amax)         |
| $a_{min}$ | - | minimum feasible semi-major axis                |
| $a_n$     | - | normal component of acceleration                |
| B         | - | perpendicular distance from $V_2$ to the chord  |
| b         | - | coefficient of a term in a polynomial           |
| C         | - | chord (chord)                                   |
| c         | - | exhaust velocity (c)                            |
| c         | - | coefficient of a term in a polynomial           |
| D         | - | drag (Drag)                                     |
| e         | - | eccentricity of an elliptical orbit             |
| $F_c$     | - | circumferential component of force              |
| $F_r$     | - | radial component of force                       |
| g         | - | gravitational acceleration (g)                  |
| h         | - | angular momentum of an orbit                    |
| $I_{sp}$  | - | specific impulse                                |
| L         | - | lift (Lift)                                     |
| m         | - | vehicle mass                                    |
| $m_x$     | - | vehicle mass at start of time interval k (Mk)   |
| $M_f$     | - | final mass                                      |

LIST OF SYMBOLS (cont.)

|            |   |  |
|------------|---|--|
| $M_0$      | - | initial mass (M0)                                      |
| $P_1$      | - | initial point  |
| $P_2$      | - | target point   |
| $p$        | - | parameter of an elliptical orbit                       |
| $r$        | - | radius (Rk)  |
| $r_1$      | - | initial radius (R0)                                    |
| $r_2$      | - | target radius (R2)                                     |
| $r_a$      | - | apogee radius of an elliptical orbit                   |
| $r_{GEO}$  | - | GEO orbital radius                                     |
| $r_p$      | - | perigee radius of an elliptical orbit                  |
| $T$        | - | thrust (Thrust)  |
| $V$        | - | vehicle velocity (Vk)                                  |
| $V_1$      | - | vehicle's current velocity (Vk)                        |
| $V_2$      | - | target velocity (V2)                                   |
| $V_{2c}$   | - | chordal component of $V_2$                             |
| $V_{2max}$ | - | maximum feasible target velocity (V2Max)               |
| $V_{2min}$ | - | minimum feasible target velocity                       |
| $V_{2p}$   | - | radial component of $V_2$                              |
| $V_{2t}$   | - | a temporarily assigned target velocity                 |
| $V_a$      | - | apogee velocity of an elliptical orbit                 |
| $V_c$      | - | chordal velocity component of $V_2$ and $V_{req}$ (VC) |
| $V_{cr}$   | - | a velocity component (VCR)                             |
| $V_G$      | - | velocity-to-be-gained (VG)                             |
| $V_{GEO}$  | - | GEO orbital velocity                                   |

LIST OF SYMBOLS (cont.)

|                    |   |   |
|--------------------|---|---|
| $V_{LEO}$          | - | LEO orbital velocity  |
| $V_p$              | - | radial velocity component of $V_2$ and $V_{req}$ (VP)               |
| $V_r$              | - | radial component of vehicle velocity (Vrk)                          |
| $V_{rc}$           | - | chordal component of $V_{req}$                                      |
| $V_{rp}$           | - | radial component of $V_{req}$                                       |
| $V_{rr}$           | - | a velocity component (VRR)  |
| $V_{\sigma}$       | - | a velocity component (VSig)   |
| x                  | - | a substitute variable   |
| $\alpha$           | - | thrust angle, angle between $V_1$ and $V_G$ (alpha)                 |
| $\beta$            | - | angle between extension of $V_1$ and $V_G$ (beta)                   |
| $\gamma_2$         | - | angle between $V_c$ and $V_2$ (gamma)                               |
| $\delta$           | - | angle between chord and $r_2$                                       |
| $\zeta$            | - | angle between $V_c$ and $V_p$ at target point                       |
| $\eta$             | - | angle between $V_2$ and extension of the radius                     |
| $\theta$           | - | angle between $V_{req}$ and local horizontal (theta)                |
| $\mu$              | - | gravitational constant of the central body (Mu)                     |
| $\sigma$           | - | transfer angle between $r_1$ and $r_2$ (sigma2)                     |
| $\sigma$           | - | downrange angle between launch site and vehicle position<br>(sigma) |
| $\phi$             | - | vehicle's flight path angle (Phk)                                   |
| $\phi_2$           | - | target flight path angle  |
| $\Delta V_c$       | - | delta-v required for orbit circularization                          |
| $\Delta V_t$       | - | delta-v required during initial thrusting phase (DeltaV)            |
| $\Delta V_{total}$ | - | delta-v required for the entire mission                             |
| $\omega$           | - | angular component of vehicle velocity (Vomk)                        |

## LIST OF FIGURES AND TABLES

| FIGURE   | PAGE |
|--|------|
| 2-1 Geometry of the Orbital Boundary-Value Problem .....       | 13   |
| 2-2 Hyperbolic Velocity Loci at $P_1$ and $P_2$ .....          | 14   |
| 2-3 Detail of Geometry at $P_2$ .....                          | 17   |
| 2-4 Geometry of Velocity Components at $P_1$ and $P_2$ .....   | 21   |
| 3-1 Velocity-to-be-Gained Determination .....                  | 31   |
| 4-1 Sample LEO to GEO Transfer Trajectory .....                | 41   |
| 4-2a Thrust, Thrust Angle vs Time -- LEO to GEO Transfer ..... | 43   |
| 4-2b Altitude vs Time -- LEO to GEO Transfer .....             | 43   |
| 4-2c Velocity vs Time -- LEO to GEO Transfer .....             | 44   |
| 4-2d Flight Path Angle vs Time -- LEO to GEO Transfer .....    | 44   |
| 4-3 Partial LEO to GEO Transfer Trajectory .....               | 46   |
| 4-4 Sample Launch Trajectory .....                             | 48   |
| 4-5a Thrust, Thrust Angle vs Time -- Launch to LEO .....       | 49   |
| 4-5b Altitude vs Time -- Launch to LEO .....                   | 49   |
| 4-5c Velocity vs Time -- Launch to LEO .....                   | 50   |
| 4-5d Flight Path Angle vs Time -- Launch to LEO .....          | 50   |
| TABLE  |      |
| 4-1 $\Delta V$ Comparisons for LEO to GEO Transfer .....       | 47   |

## CHAPTER 1

### Introduction

Since the Apollo era, and especially with the advent of the Space Shuttle program, guidance techniques for spacecraft in powered flight have flourished. Elaborate algorithms exist that find an optimum trajectory for given target conditions, and for a given spaceflight mission, and these algorithms perform the task very well for today's sophisticated spacecraft. It is of course necessary to use an exact, optimal guidance technique with an actual vehicle, but the requirements for exactness and optimality are precisely what lead to the enormous complexity of these algorithms.

When a real vehicle is not involved, and such precise accuracy or optimality is not required, the currently available guidance techniques -- such as those based on Pontryagin's maximum principle, Bellman's dynamic programming, or Raleigh-Ritz type procedures -- can end up being more elaborate and detailed than necessary. Especially early in design studies, when a vehicle can be modelled by a small set of parameters, and a quick, easy method for finding a feasible trajectory is needed, algorithms based on these optimal control theories are too cumbersome and time-consuming to implement. In addition, each particular algorithm is developed around one specified set of target parameters among position, velocity, transfer angle, and flight path angle, and usually cannot be used for any other combination of these parameters.

No technique exists, however, that is easy to implement, close to optimum, and works for more than one combination of target parameters, or for more than a limited type of spaceflight missions. Even the classic calculus-of-variations-derived linear tangent algorithm, in its simplest form, works for only a very narrow range of problems -- so tightly constrained that they aren't very realistic -- and when made more versatile, it too becomes complex and cumbersome. (See Space Shuttle Ascent, Guidance, Navigation, and Control, ref. 1)

This gap in spacecraft guidance techniques can be filled by relaxing the requirement that the trajectory be exactly optimum, and by using basic methods from orbital mechanics to ensure that the target conditions are satisfied. A means already exists, known as the orbital boundary-value problem, for finding an orbit that passes through a given initial point and a selected target point, thus providing a technique for guiding a vehicle to a desired target position. The boundary-value problem, however, allows only a limited choice of target parameters, with the restriction that the target be a fixed position in space. The intent of this thesis, therefore, is to extend the basic boundary-value technique to allow multiple choices of target parameters, including choices better suited to the requirements for current space missions. The technique can thus be adapted for use in a wide variety of two-dimensional spaceflight targeting problems, ranging from planetary launch trajectories to orbital insertion and orbit-to-orbit transfers, as well as interplanetary missions, and most two-body targeting problems in general.

## CHAPTER 2

### Development of the Orbit-Fitting Targeting Technique

#### 2.1 The Orbital Boundary-Value Problem

The starting point for the technique developed in this thesis is the classic two-point orbital boundary-value problem. The problem formulation is: given an initial position and a desired target position measured from a central body, find the feasible orbits that pass through both points, and find the required velocity vectors at the initial point that correspond to flight paths along any of the feasible transfer orbits. The technique for solving this problem has been described in detail by Battin (ref. 2) and will be summarized below.

If a line (the chord) is drawn connecting the initial and target points, then any velocity vector at either point can be defined in terms of a component parallel to the chord and one parallel to the radius vector (see figure 2-1) provided the transfer angle,  $\sigma$ , is not equal to 180 degrees. If  $\bar{V}_{req}$  and  $\bar{V}_2$  are velocity vectors at  $P_1$  and  $P_2$  respectively, and if they correspond to an orbit that passes through  $P_1$  and  $P_2$ , then from fundamental properties of orbital mechanics, the relationships

$$V_{rc} = V_{2c} = V_c$$

$$V_{rp} = -V_{2p} = V_p$$

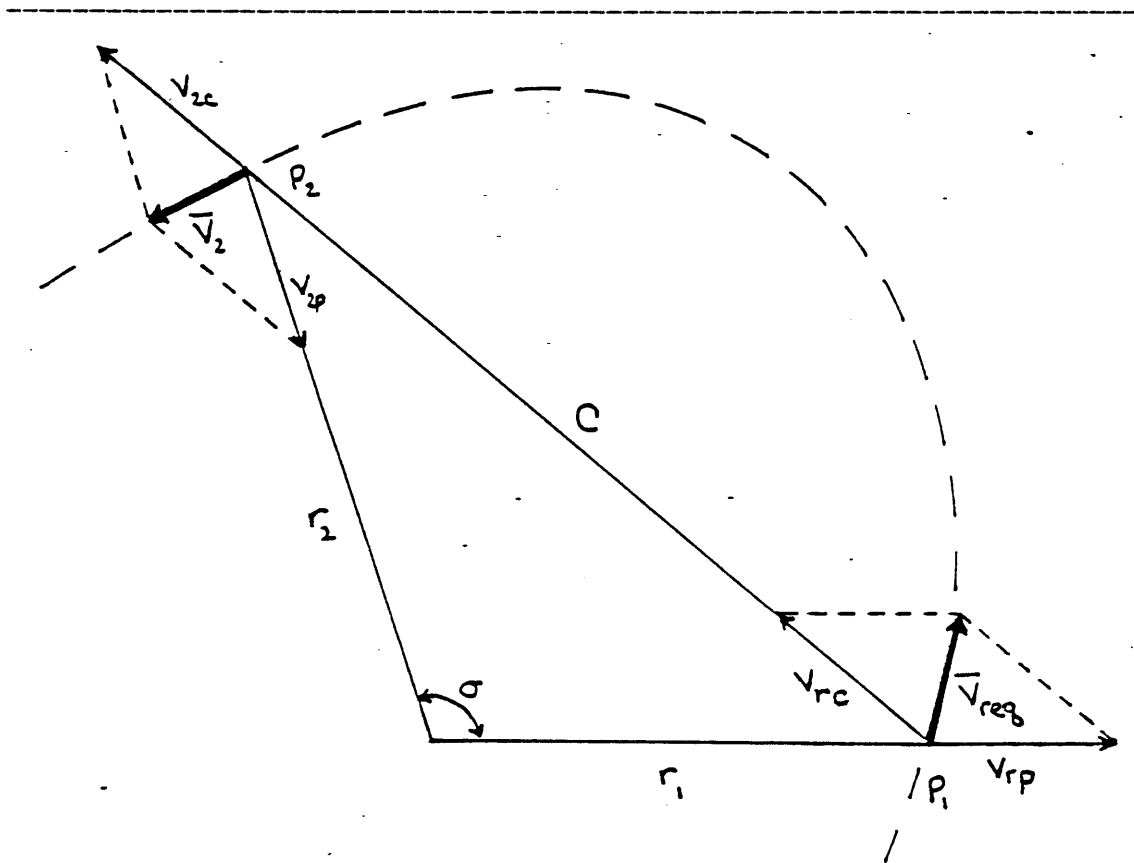


Figure 2-1 Geometry of the Orbital Boundary-Value Problem

can easily be derived, where the subscript \$r\$ denotes current or initial position; 2 denotes target or final position; \$c\$ denotes chordal components; and \$p\$ denotes radial components. Without loss of generality, the target radius can be taken to be greater than the initial radius. The chord can be found from the law of cosines

$$C^2 = r_1^2 + r_2^2 - 2r_1r_2 \cos\sigma \quad (2.1)$$

and the product of the chordal velocity component, \$V\_c\$, and radial velocity component, \$V\_p\$, can then be expressed as a function of the chord, the initial and final radii, and the transfer angle, \$\sigma\$, in the form

$$\frac{v_c v_p}{r_1 r_2} = \frac{\mu C}{(1 + \cos \sigma)} \quad (2.2)$$

This relationship defines an infinite family of velocity vectors at the initial point and at the target point, and is in the asymptotic form of an equation for a hyperbola -- with the chord as one asymptote and the radius as the second asymptote. This hyperbola at  $P_1$  represents the locus of all feasible velocity vectors at the initial point, and the hyperbola at  $P_2$  corresponds to the locus of the resulting velocity vectors at the target point, as shown in figure 2-2. Properties

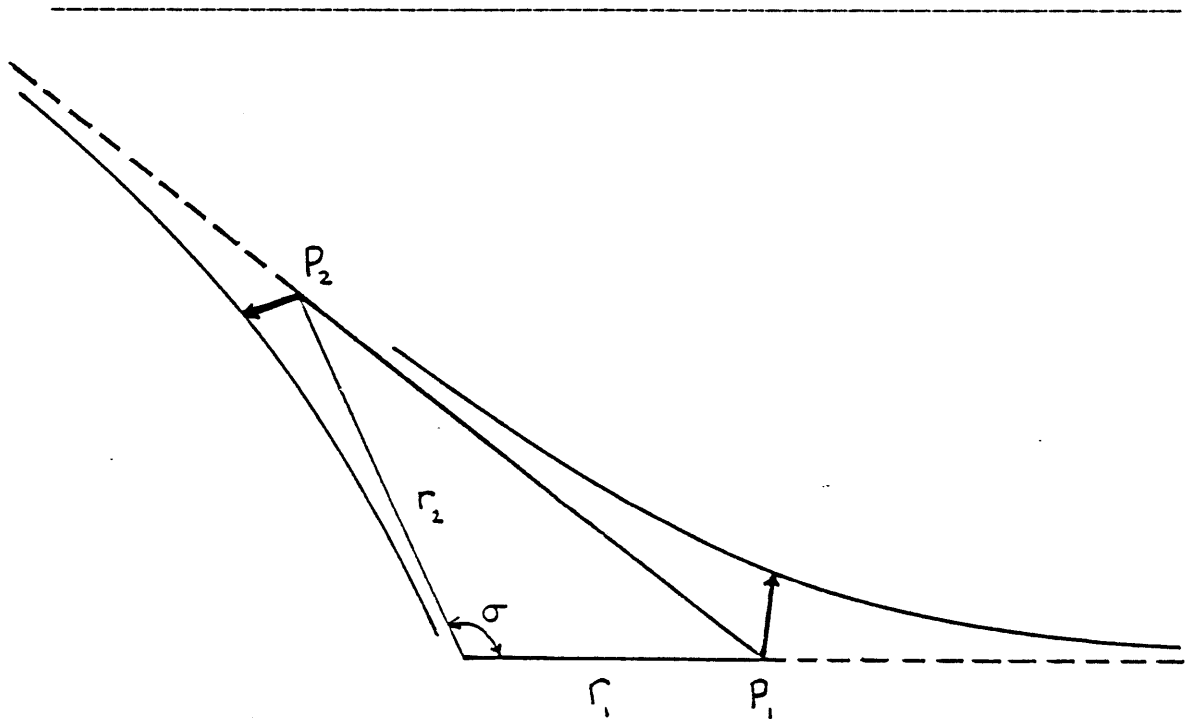


Figure 2-2 Hyperbolic Velocity Loci at  $P_1$  and  $P_2$

of this hyperbolic locus will be exploited later in the development of the orbit-fitting targeting technique. Since the required velocity vector,  $\bar{v}_{req}$ , (that which corresponds to flight along a feasible orbit from  $P_1$  to  $P_2$ ) is not yet uniquely specified, one more parameter, such

as target velocity or target flight path angle, may be specified. Selecting either one of these will uniquely determine a feasible transfer orbit, and hence will also determine a unique required velocity at the initial point. The orbital boundary-value problem, however, does not permit the designation of both the target velocity and target flight path angle as target parameters. Since, for many types of missions, it is desirable to specify both these variables as target parameters, a means must be found to eliminate this restriction from the boundary-value problem.

If either of the target parameters upon which the boundary-value problem is based -- target radius or transfer angle -- is not specified as a target parameter, the restriction is removed and any two of the remaining three unspecified variables may now be selected as target parameters. Thus, of the complete set of four parameters available -- velocity, altitude, flight path angle, and transfer angle -- any three may be chosen as the target parameters, and they will uniquely determine an orbital path that passes through the initial point and achieves the desired target conditions at the target point. In the following sections, the necessary relationships will be established that allow a solution to be found for any of the four such combinations of target parameters.

## 2.2 Solution Technique for Any Choice of Target Parameters

When any three of the target parameters are specified and the initial position is known, the fourth parameter is also determined,

and can be found as a function of the three specified parameters and the initial radius. The fourth parameter must be found in order to completely specify the geometry at the target point and to eventually find the required velocity at the initial point.

To start, the geometry at the target point is expressed in terms of the angles  $\beta$  and  $\eta$ , as shown in figure 2-3. The perpendicular distance, A, from the the radius to the tip of  $V_2$ , can be found as

$$A = V_2 \sin \eta \quad (2.3)$$

and the perpendicular distance from the extension of the chord to  $V_2$  is

$$B = -V_2 \sin(\beta + \eta) \quad (2.4)$$

A and B can also be expressed in terms of the velocity components,  $V_c$  and  $V_p$  as

$$A = V_c \sin \beta \quad (2.5)$$

$$B = V_p \sin \beta \quad (2.6)$$

Since the locus of possible velocities at  $P_2$  is a hyperbola (as established in equation 2.2), using the property of hyperbolas that the product AB is equal to a constant, AB can be found by multiplying equations (2.3) and (2.4), and also by multiplying equations (2.5) and (2.6). Setting the resulting expressions equal to each other gives

$$AB = -V_2^2 \sin \eta \sin(\beta + \eta) = V_c V_p \sin^2 \beta \quad (2.7)$$

Now equation (2.2) can be substituted for the product  $V_c V_p$ , and if the

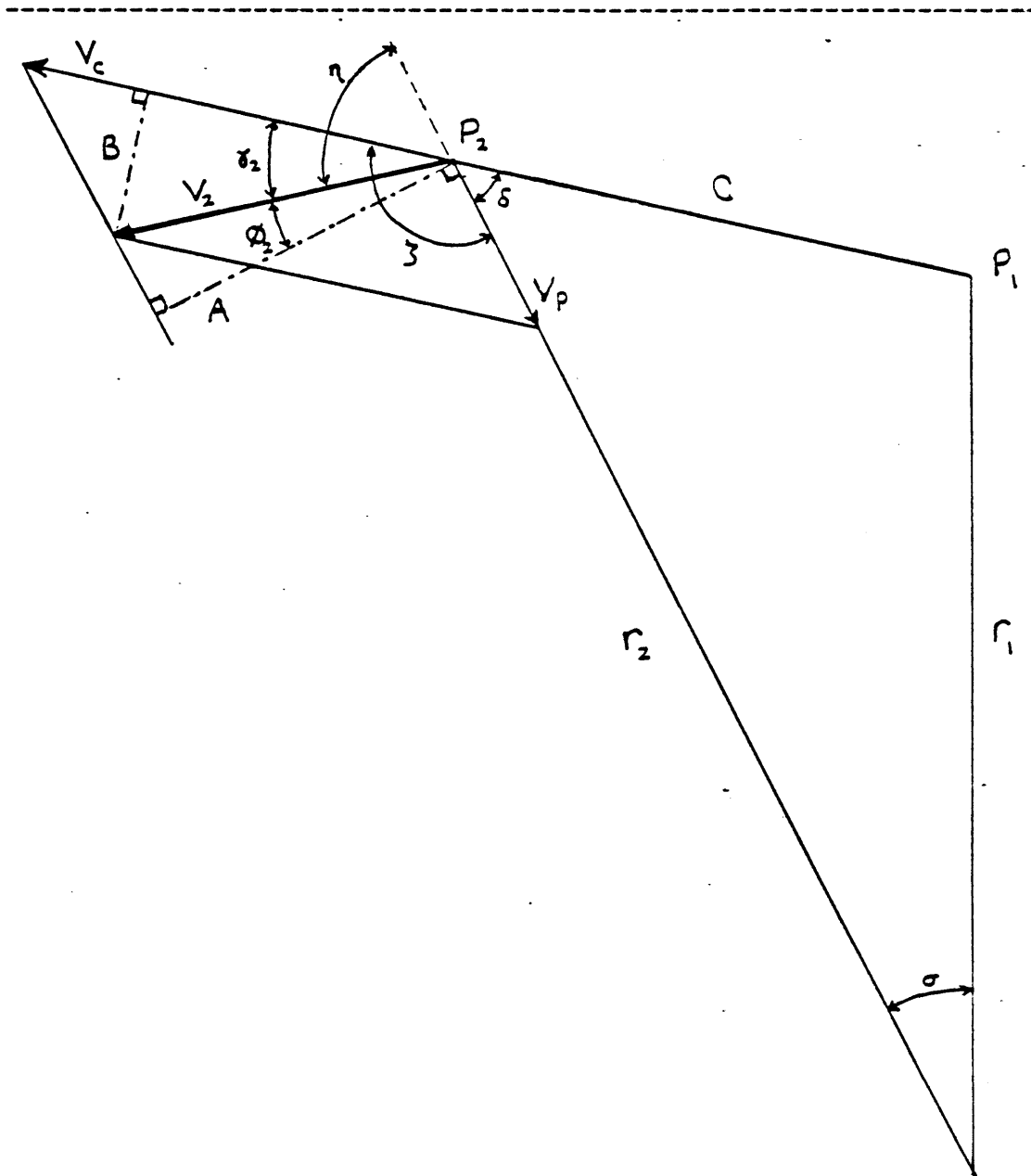


Figure 2-3 Detail of Geometry at  $P_2$

chord, C, is replaced by

$$C = r_1 \frac{\sin \sigma}{\sin \eta} \quad (2.8)$$

the resulting expression is

$$\mu \sin \eta \tan \frac{\sigma}{2} = -r_2 v_2^2 \sin \eta \sin(\zeta + \eta) \quad (2.9)$$

To eliminate  $\gamma$  from this equation, a useful relationship can be developed by expressing  $r_2$  as

$$r_2 = r_1 \cos\sigma - C \cos\gamma \quad (2.10)$$

and substituting equation (2.8) for  $C$ , then solving for  $\gamma$  gives

$$\cot\gamma = \cot\sigma - \frac{r_2}{r_1 \sin\sigma} \quad (2.11)$$

Then  $\gamma$  can be eliminated between equations (2.9) and (2.11), with the final result

$$\mu r_1 (1 - \cos\sigma) + r_1 r_2 V_2^2 \sin\eta \sin(\sigma + \eta) = r_2^2 V_2^2 \sin^2\eta \quad (2.12)$$

Recognizing that  $\eta = \pi/2 - \phi_2$ , equation (2.12) can be written

$$\mu r_1 (1 - \cos\sigma) + r_1 r_2 V_2^2 \cos\phi_2 \cos(\phi_2 - \sigma) = r_2^2 V_2^2 \cos^2\phi_2 \quad (2.13)$$

This is the desired expression which relates all the possible target parameters -- altitude, velocity, flight path angle, and transfer angle -- to each other, and once any three of these parameters are specified, the fourth can be found directly from this relationship.

### 2.3 Altitude, Velocity, and Flight Path Angle as Target Parameters

In this particular case, the target altitude or radius,  $r_2$ , velocity,  $V_2$ , and flight path angle,  $\phi_2$ , are specified, and the remaining variable,  $\sigma$ , must be found in order to completely determine the geometry of the problem. To solve for  $\sigma$  from equation (2.13), the

substitution

$$x = \cot \frac{\sigma}{2} \quad (2.14)$$

can be used, with

$$\sin \sigma = \frac{2x}{1+x^2} \quad \cos \sigma = -\frac{1-x^2}{1+x^2} \quad 1-\cos \sigma = \frac{2}{1+x^2}$$

The result is a quadratic in  $x$ , of the form

$$ax^2 + bx + c = 0$$

where

$$a = -r_2 v_2^2 (r_2 - r_1) \cos^2 \phi_2 \quad (2.15)$$

$$b = 2r_1 r_2 v_2^2 \sin \phi_2 \cos \phi_2 \quad (2.16)$$

$$c = 2\mu r_1 - r_2 v_2^2 (r_1 + r_2) \cos^2 \phi_2 \quad (2.17)$$

The quadratic can easily be solved with the standard quadratic equation, and once  $x$  is found, equation (2.14) gives the answer for  $\sigma$ .

Equation (2.13) can be used for any other combination of target parameters. However, only the case completed above will be explored in further depth, since this case is the most useful for common space missions. (The remaining three cases are solved in Appendix A).

#### 2.4 Finding the Required Velocity

Once the values of all four target variables are known, the

geometry of the feasible transfer orbit is fixed and the velocity at the initial point can be determined. To find the required velocity, the geometry is set up as shown in figure 2-4. The internal angle,  $\delta$ , between the chord and the radius vector to  $P_2$ , can be found from the law of sines

$$\frac{\sin\delta}{r_1} = \frac{\sin\sigma}{C} \quad (2.18)$$

but the geometry also shows that

$$\delta = \frac{\pi}{2} - \phi_2 - \gamma_2 \quad (2.19)$$

Substituting equation (2.19) for  $\delta$  into (2.18) and using a trig identity gives

$$\cos(\gamma_2 + \phi_2) = \frac{r_1}{C} \sin\sigma \quad (2.20)$$

which allows solving for  $\gamma_2$  once  $\phi_2$  and  $\sigma$  are known.

The velocity at the target point,  $\bar{V}_2$ , can be resolved into its component in the chordal direction,  $V_c$ , and its component in the radial direction,  $V_p$

$$V_2 \cos\phi_2 = V_c \cos(\gamma_2 + \phi_2) \quad (2.21)$$

$$V_2 \sin\phi_2 = V_c \sin(\gamma_2 + \phi_2) - V_p \quad (2.22)$$

and  $V_p$  and  $V_c$  can be solved for in terms of  $V_2$  from equations (2.21) and (2.22)

$$V_p = V_2 \left[ \tan(\gamma_2 + \phi_2) \cos\phi_2 - \sin\phi_2 \right] \quad (2.23)$$

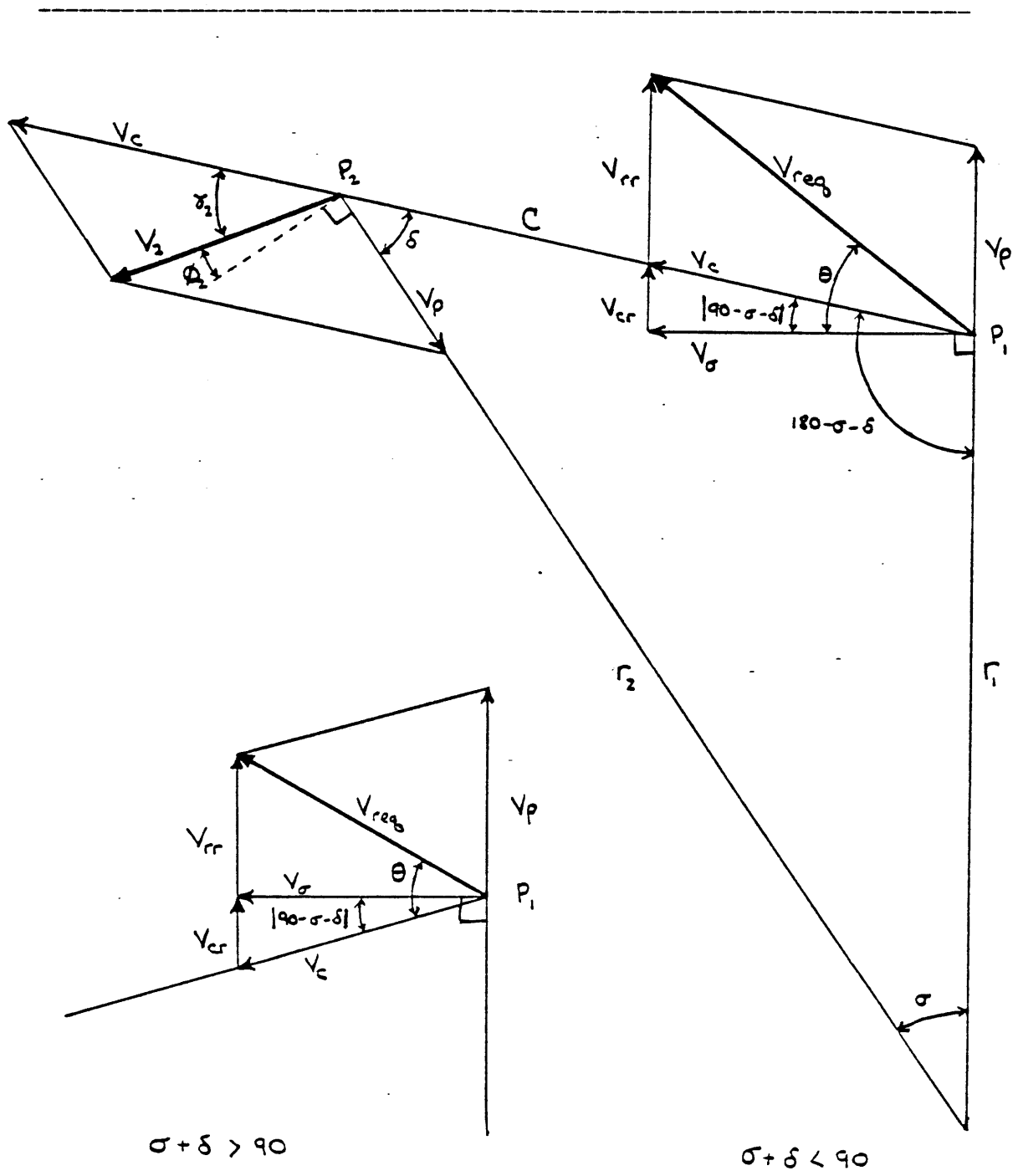


Figure 2-4 Geometry of Velocity Components at  $P_1$  and  $P_2$

and

$$v_c = \frac{v_2 C \cos \phi_2}{\cos(\gamma_2 + \phi_2)} \quad (2.24)$$

If equation (2.20) is used to eliminate  $\gamma_2$  from equations (2.23) and (2.24),  $v_c$  and  $v_p$  can be expressed solely in terms of known target parameters and  $r_1$

$$v_p = v_2 \left[ \frac{\sqrt{1 - \left(\frac{r_1}{C}\right)^2 \sin^2 \sigma}}{\frac{r_1}{C} \sin \sigma} \cos \phi_2 - \sin \phi_2 \right] \quad (2.25)$$

$$v_c = \frac{v_2 C \cos \phi_2}{r_1 \sin \sigma} \quad (2.26)$$

Additional useful relationships are

$$v_\sigma = v_c \sin(\sigma + \delta) \quad (2.27)$$

$$v_{cr} = v_c |\cos(\sigma + \delta)| \quad (2.28)$$

To find the required velocity vector at the initial point, two cases must be considered:

1)  $\sigma + \delta < \pi/2$ :

From the geometry, clearly

$$v_{rr} = v_p \quad (2.29)$$

and the angle,  $\theta$ , between the local horizontal and the required velocity vector is

$$\theta = \arctan \left( \frac{v_p + v_{cr}}{v_\sigma} \right) \quad (2.30)$$

then the magnitude of the required velocity is given by

$$v_{req}^2 = (v_p + v_{cr})^2 + v_\sigma^2 \quad (2.31)$$

2)  $\sigma + \delta > \pi/2$ :

In this case the geometry shows that

$$v_{rr} = v_p - v_{cr} \quad (2.32)$$

$$\theta = \arctan \left( \frac{v_{rr}}{v_\sigma} \right) \quad (2.33)$$

leading to

$$v_{req}^2 = v_{rr}^2 + v_\sigma^2 \quad (2.34)$$

Substituting the appropriate expressions for  $v_{rr}$ ,  $v_\sigma$ , and  $v_{cr}$  into equations (2.30) through (2.34) yields expressions for  $v_{req}$  and  $\theta$  which hold for both cases

$$v_{req}^2 = 2v_p v_c \cos(\sigma + \delta) + v_c^2 \quad (2.35)$$

$$\theta = \arctan \left[ \frac{v_p - v_c \cos(\sigma + \delta)}{v_c \sin(\sigma + \delta)} \right] \quad (2.36)$$

where  $\delta$  is found, from equation (2.18) as

$$\delta = \arcsin \left( \frac{r_1}{c} \sin \sigma \right) \quad (2.37)$$

To eliminate  $\delta$  from equation (2.35) and (2.36),  $\cos(\sigma+\delta)$  and  $\sin(\sigma+\delta)$  are expanded, and equation (2.37) substituted for  $\delta$  to give

$$\cos(\sigma+\delta) = \cos\sigma \sqrt{1 - \left(\frac{r_1}{C}\right)^2 \sin^2\sigma} - \frac{r_1}{C} \sin^2\sigma \quad (2.38)$$

and

$$\sin(\sigma+\delta) = \sin\sigma \left[ \frac{r_1}{C} \cos\sigma + \sqrt{1 - \left(\frac{r_1}{C}\right)^2 \sin^2\sigma} \right] \quad (2.39)$$

and finally, substituting equations (2.38) and (2.39) into equations (2.35) and (2.36) gives the result

$$v_{\text{req}}^2 = 2v_c v_p \left[ \cos\sigma \sqrt{1 - \left(\frac{r_1}{C}\right)^2 \sin^2\sigma} - \frac{r_1}{C} \sin^2\sigma \right] + v_c^2 \quad (2.40)$$

with

$$\theta = \arctan \left\{ \frac{v_p - v_c \left[ \cos\sigma \sqrt{1 - \left(\frac{r_1}{C}\right)^2 \sin^2\sigma} - \frac{r_1}{C} \sin^2\sigma \right]}{v_c \sin\sigma \left[ \frac{r_1}{C} \cos\sigma + \sqrt{1 - \left(\frac{r_1}{C}\right)^2 \sin^2\sigma} \right]} \right\} \quad (2.41)$$

These expressions involve only the chordal and radial velocity components of  $V_2$ , found from equations (2.25) and (2.26), and other known variables, and provide a way of finding the required velocity once all the target parameters and initial point are specified. (An alternative technique for finding  $v_{\text{req}}$  and  $\theta$  is presented in Appendix 3.)

## 2.5 Specialization for Target Point at Apogee

Since most orbital transfers and maneuvers are intended to eventually reach a circular orbit, it is useful to specify that the transfer orbit reach the altitude of the desired circular orbit with a flight path angle equal to zero. This means that the apogee radius of the transfer orbit is the radius of the circular orbit, with velocity vector perpendicular to the radius. The transfer angle,  $\sigma$ , can now be determined directly by using an additional relationship specific to the case of the target point at apogee of the transfer orbit.

Knowing  $v_2$  and  $r_2$ , the semi-major axis,  $a$ , of the transfer orbit can immediately be found from the Vis-Viva equation

$$v_2^2 = \mu \left( \frac{2}{r_2} - \frac{1}{a} \right) \quad (2.42)$$

thus

$$a = \left( \frac{2}{r_2} - \frac{v_2^2}{\mu} \right)^{-1} \quad (2.43)$$

From Battin (ref. 3) the parameter,  $p$ , can be expressed in the form

$$p = \frac{r_1 r_2}{r_2 - r_1 \cos \sigma} (1 - \cos \sigma) \quad (2.44)$$

and the eccentricity,  $e$ , is found from

$$e = \frac{r_2 - r_1}{r_2 - r_1 \cos \sigma} \quad (2.45)$$

where  $\sigma$  still remains to be found. By using the relationship

$$p = a(1-e^2) \quad (2.46)$$

equations (2.44) and (2.45) can be substituted into equation (2.46) for  $p$  and  $e$ , and  $\sigma$  can be found in terms of the semi-major axis, with the result

$$\cos\sigma = \frac{2ar_2 - ar_1 - r_2^2}{ar_1 - r_1r_2} \quad (2.47)$$

Substituting for  $a$  from (2.43) finally gives

$$\cos\sigma = \frac{r_2^2 v_2^2 - \mu r_1}{r_1 r_2 v_2^2 - \mu r_1} \quad (2.48)$$

As a check, this agrees with the result obtained from equation (2.13) if  $\phi_2$  is set equal to zero.

With  $\sigma$  now determined,  $\phi_2$  can be set equal to zero in equations (2.25) and (2.26), and the chordal and radial velocity components become

$$v_c = \frac{v_2 C}{r_1 \sin\sigma} \quad (2.49)$$

$$v_p = \frac{v_2 \sqrt{1 - \left(\frac{r_1}{C}\right)^2 \sin^2\sigma}}{\frac{r_1}{C} \sin\sigma} \quad (2.50)$$

The required velocity and  $\theta$  can now be found from equations (2.40) and (2.41).

## 2.6 Range of Feasible Required Velocities

At this point, it is advantageous to explore the feasible range of  $V_2$  as a function of  $r_1$  and  $r_2$ . The choice of  $V_2$  is restricted, depending on  $r_1$  and  $r_2$  -- there being a maximum value above which no feasible orbit passes through both the initial and target points and also satisfies the target conditions. This is true because once  $r_2$  and  $V_2$  are selected, the semi-major axis of the orbit is fixed from equation (2.43) and the perigee radius of the orbit is

$$r_p = 2a - r_2 \quad (2.51)$$

If  $r_1$  is less than  $r_p$ , then clearly no point in the orbit will pass through the initial point, and the problem cannot be solved. The actual maximum value of  $V_2$  can be found by determining the maximum feasible semi-major axis for the transfer orbit

$$a_{\max} = \frac{r_1 + r_2}{2} \quad (2.52)$$

corresponding to an elliptical orbit with perigee at  $r_1$  and apogee at  $r_2$ . This gives an upper limit on  $V_2$

$$V_{2\max}^2 = \mu \left( \frac{2}{r_2} - \frac{1}{a_{\max}} \right) \quad (2.53)$$

The lower limit of  $V_2$  that still yields a feasible orbit is

$$V_{2\min} = 0 \quad (2.54)$$

and the associated semi-major axis is

$$a_{\min} = \frac{r_2}{2} \quad (2.55)$$

corresponding to an orbit that travels a straight line from  $P_1$  to  $P_2$  and reaches  $P_2$  with zero velocity -- not very useful for practical purposes. The upper limit will become important later, however.

## CHAPTER 3

### Guidance Algorithm

#### 3.1 Overview

The technique developed in the preceding chapter for finding the required velocity vector can easily be incorporated into a guidance algorithm for a spacecraft in powered flight. At any point in the spacecraft's trajectory, the vehicle's current position can be designated the initial point,  $P_1$ , in the boundary-value problem, and the problem can be solved for that particular initial point and a specified set of target conditions. Unless the vehicle has infinite thrust, however, it will not be able to instantaneously achieve the required velocity, and the process must be repeated some time interval later, with the vehicle's updated current position as the new initial point in the problem. This procedure therefore lends itself well to computer trajectory simulations in which the vehicle's position is determined at discrete time intervals by numerically integrating the equations of motion. At each time interval, then, once a set of target conditions has been chosen, the required velocity vector for the vehicle at its current position can be found, and a guidance algorithm followed such that the vehicle's velocity vector eventually equals the required velocity vector. The control variables available for use in this problem are the vehicle thrust level -- most conveniently measured as a per-

centage of the maximum available thrust -- and the thrust angle, the angle between the thrust vector and the velocity vector. While a spacecraft cannot directly control the thrust angle, it does so indirectly by controlling the rocket nozzle gimbal angle, but using the thrust angle as a control variable avoids the necessity of including the dynamic relations between the nozzle gimbal angle and resulting thrust angle; these relations are vehicle-dependent, and contribute nothing to the analysis of the original problem. Hence the choice of thrust angle as the second control variable.

### 3.2 Velocity-to-be-Gained Determination

Once the required velocity for the vehicle's current position has been found, the velocity-to-be-gained,  $\bar{V}_G$ , is defined as the vector difference between the required velocity and the vehicle's current velocity (see figure 3-1).

If  $\phi$  is the current flight path angle of the vehicle, measured from the local horizontal, and  $V_1$  its current velocity, then  $V_G$  can be determined from the law of cosines

$$V_G^2 = V_{req}^2 + V_1^2 - 2V_{req}V_1 \cos(\phi - \theta) \quad (3.1)$$

and the angle,  $\beta$ , between  $\bar{V}_1$  and  $\bar{V}_G$  can be found from the law of sines

$$\beta = \arcsin \left( \frac{V_{req}}{V_G} \sin|\phi - \theta| \right) \quad (3.2)$$

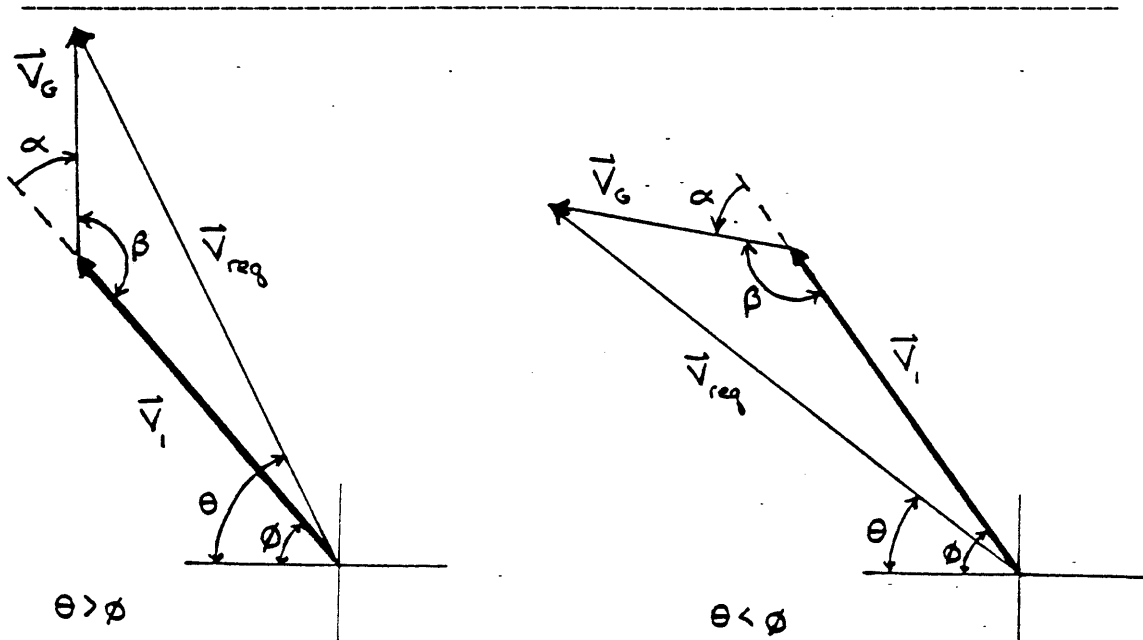


Figure 3-1 Velocity-to-be-Gained Determination

In many cases,  $\beta$  is greater than 90 degrees, but the arcsin function will return values only between -90 and 90 degrees, so for the case where

$$v_G^2 + v_1^2 < v_{req}^2$$

$\beta$  must be corrected to

$$\beta = \pi - \arcsin\left(\frac{v_{req}}{v_G} \sin|\phi - \theta|\right) \quad (3.3)$$

### 3.3 Thrust Angle Determination

For the sake of simplicity, the guidance algorithm to be used will choose the thrust vector such that its direction is aligned with the

velocity-to-be-gained direction. This makes it easy to find the thrust angle,  $\alpha$ , as follows.

IF  $\theta < \phi$ :

$$\alpha = \pi - \beta \quad (3.4)$$

and if  $\theta > \phi$ :

$$\alpha = \beta - \pi \quad (3.5)$$

and of course if  $\theta = \phi$ :

$$\alpha = 0$$

### 3.4 Thrust Level Determination

A spacecraft in powered flight, thrusting in the  $\vec{V}_G$  direction, will be continuously increasing the energy of its orbit (provided  $r_2 > r_1$ ) and must avoid increasing its energy beyond that required for the target transfer orbit. Specifically, the vehicle velocity must be prevented from becoming greater than  $V_{req}$ . During the time interval before this occurs, therefore, the vehicle can either stop thrusting its main engines and coast to the target, using smaller engines to "fine tune" its trajectory, or if the vehicle has throttleable engines, it can reduce its thrust to the level required to exactly match the required velocity without overshooting it. In order to accomplish this, the increment to the vehicle's velocity during the next time interval must be predicted, and then the thrust level must be found that corresponds to a velocity increment that is less than or equal to

$V_G$ . This can be done in the following manner.

The propellant mass flow rate is determined from the vehicle thrust,  $T$ , and exhaust velocity,  $c$

$$\frac{dm}{dt} = - \frac{T}{c} \quad (3.6)$$

and for a given time interval,  $Dt$ , the change in vehicle mass,  $Dm$ , is

$$Dm = \frac{T}{c} Dt \quad (3.7)$$

If  $m_k$  is the vehicle mass at the beginning of the time interval, and  $m_{k+1}$  is the mass at the end of the interval, then

$$m_{k+1} = m_k - Dm \quad (3.8)$$

and substituting equation (3.8) into equation (3.7) for  $Dm$  gives

$$m_{k+1} = m_k - \frac{T}{c} Dt \quad (3.9)$$

The rocket equation can be applied now, since, if  $Dt$  is small enough, the behavior of the vehicle will be as if it had thrusted impulsively at the beginning of the time interval, and the velocity increment,  $DV_k$ , is then

$$DV_k = -c \ln \left( \frac{m_{k+1}}{m_k} \right) \quad (3.10)$$

solving for  $m_{k+1}$  gives

$$m_{k+1} = m_k e^{-\left(\frac{DV_k}{c}\right)} \quad (3.11)$$

finally, substituting for  $m_{k+1}$  from equation (3.9) and solving for T determines the thrust required to achieve a specified DV

$$T = \frac{m_k c}{Dt} \left[ 1 - e^{-\left(\frac{DV_k}{c}\right)} \right] \quad (3.12)$$

At the point where using maximum thrust would cause  $DV_k$  to exceed  $V_G$ ,  $V_G$  should be substituted for  $DV_k$  in equation (3.12) and the new thrust level used for that time interval. At the next time interval, when a new  $V_G$  is found, the process should be repeated and the appropriate thrust level determined again. Once  $V_G$  becomes exactly equal to zero, the vehicle can stop thrusting entirely and coast the rest of the way to the target, since it has achieved perfect alignment with the desired transfer orbit.

### 3.5 Guidance for Infeasible Target Velocities

A problem arises if the selected target velocity,  $V_2$ , is greater than the maximum feasible target velocity,  $V_{2max}$ , for the vehicle at its current position. (Recall that  $V_{2max}$  is the apogee velocity of an elliptical orbit with the vehicle's current altitude as its perigee altitude, as discussed in section 2.6) In fact, if the vehicle thrust is low and the target velocity high, it could be some time before the vehicle has climbed to a high enough altitude that  $V_{2max}$  becomes greater than  $V_2$ . During this phase of the flight a "temporary" guidance

scheme must be used, and several options exist for doing this.

First, a temporary target velocity,  $V_{2t}$ , can be chosen, where  $V_{2t}$  is almost equal to  $V_{2max}$ . (If  $V_{2t}$  were set exactly equal to  $V_{2max}$ , the transfer angle would be at 180 degrees, where a singularity exists in the algorithm.) At each time step, as the vehicle altitude increases,  $V_{2max}$  will increase, and therefore  $V_{2t}$  can also be increased. Eventually  $V_{2max}$  will be greater than the original desired  $V_2$  and the orbit-fitting targeting technique may be used. This technique guarantees a smooth transition from the first phase to the second, with no abrupt changes in the required velocity or flight path angle, but it has the disadvantage that it attempts to keep the vehicle at a flight path angle of zero, since the perigee of the temporary transfer orbit is always placed at the vehicle's position. Since a vehicle under thrust naturally tends to spiral outward and its flight path angle tends to increase, this scheme would require that the vehicle thrust against the radial component of its velocity -- an inefficient procedure and wasteful use of fuel.

A second possible scheme would simply allow the vehicle to thrust tangentially to its flight path and follow a natural gravity-turn spiral trajectory until it has reached the altitude where  $V_{2max}$  is greater than  $V_2$ . At that point, it has intersected the feasible transfer orbit at its perigee, where the flight path angle is zero, and the vehicle will immediately be required to thrust against its radial velocity component. It spends much less time doing so, however, than in the first scheme, since now the flight path angle of the feasible transfer orbit increases with altitude, and the result is better fuel

efficiency than the first scheme. If a constraint is placed on the thrust angle rate of change, abrupt changes in the vehicle's pitch can be avoided, and the transition from this first phase into the orbit-fitting targeting phase can be made smoothly.

The third possible scheme would involve intercepting the feasible transfer orbit at some point other than at perigee. An altitude exists where the difference between the tangential-thrust trajectory velocity and the transfer orbit velocity is a minimum. Presumably, it would be most efficient to perform the transition at this point, but the minimum  $V_G$  point cannot be found analytically, and can only be identified during the trajectory by monitoring  $V_G$  and watching for the point where  $V_G$  just starts to increase after reaching a minimum. By that time, however, the required thrust angle is very large, often greater than 90 degrees, and if the transition is performed at this point, the vehicle is required to thrust at a large angle from its flight path until it aligns itself with the desired transfer orbit. Even if the exact point of minimum  $V_G$  could be found, since the minimum  $V_G$  usually points sharply away from the vehicle's velocity, the result would still be inefficient use of fuel during this transition to the orbit-fitting targeting technique.

The second scheme ends up being the most efficient of the three alternatives described, and this is the technique implemented in the trajectory simulations to handle cases where  $V_2$  is initially greater than  $V_{2max}$ .

## CHAPTER 4

### Technique Applications and Results

#### 4.1 Overview

Two missions were selected as test applications for the targeting technique: LEO to GEO transfer, and earth launch to LEO insertion. These cases represent two of the most frequent maneuvers in current spacecraft operations, and probably in future operations as well. In the case of LEO to GEO transfer, a near-optimum trajectory can be found fairly easily for any given thrust level, and therefore the performance of the technique developed in this paper can easily be compared with known near-optimum values. The results from this case can also be applied to other maneuvers which involve orbit-to-orbit transfers, since this is the broad class of maneuvers of which circular LEO to GEO transfer is but one example.

For the earth launch case, while it is considerably more difficult to find the optimum trajectory, a reasonable value has been well established and the results can be compared with this known value. If the atmospheric effects are neglected or are changed to match those of another body, the earth launch case can be considered an example of launch from any celestial body, and the results may be applied to other situations involving launch and insertion into any orbit.

## 4.2 Simulation Technique

The trajectory simulations were performed by applying a standard Fourth-Order Runge-Kutta integration algorithm to the spacecraft equations of motion. (For a description and development of these equations, see Appendix C.) The set of initial conditions -- altitude, velocity, and flight path angle -- were selected according to the specific mission being analyzed. Constant thrust was used throughout most of the engine-firing phases, without regard to maximum acceleration limits, although acceleration limits could be imposed if desired. For all of the missions studied, however, the acceleration levels never became high enough that constraints would have significantly affected the performance. The vehicle was therefore allowed to use maximum thrust until near the end of the burn when the vehicle velocity approached the required velocity, at which point the thrust, assumed to be throttleable, was reduced as required.

Values could be chosen to limit the maximum thrust angle (hence vehicle pitch rate) and the maximum thrust angle rate of change during the thrusting phase. However, after thrust was reduced and the velocity-to-be-gained fell below 1 m/s, the thrust angle and pitch rate were left unconstrained to allow "fine-tuning" or continuous corrections to the trajectory. While this type of continuous adjustment is not realistic, -- an actual spacecraft would only perform a small number of discrete mid-course correction burns -- by allowing continuous adjustments, the accumulated numerical and roundoff errors end up being nullified. This occurs because the desired transfer orbit is calculated exactly at each time step, with no approximations, and the guidance

algorithm corrects for any deviation from this transfer orbit. The numerical errors arising from the use of an approximate integration technique are regarded simply as deviations from the desired path and are thus corrected for, with the end result that the trajectory is closer to an exact actual trajectory and the target values are reached with extremely high precision. Since allowing continuous correction ends up consuming more fuel than would otherwise be necessary, the delta-v required for the mission is calculated at the point where the thrust is first reduced and the velocity-to-be-gained is below 1 m/s, the point at which an actual vehicle would shut off its engines and coast the rest of the way to the target. (The computer program to perform the simulations is listed in Appendix F.)

Once the vehicle has reached the target altitude, an additional burn must be performed to insert the vehicle into a circular orbit. For the sake of simplicity, and because it is the option that requires the least fuel, this burn was chosen to be impulsive in the simulations, and the required delta-v is simply the difference between the circular orbital velocity and the apogee velocity of the transfer orbit.

#### 4.3 Orbit-to-Orbit Transfer Trajectories

The Orbital Transfer Vehicle (OTV) model was selected on the basis of what might eventually be in use as part of future space activities. For reuseability, liquid fuels, with an engine such as the advanced space engine, would be used, with vehicle parameters and initial and target conditions as follows:

specific impulse: 450 seconds  
initial mass: 25,000 kg  
initial acceleration: 0.03 to 3.0 g's  
initial altitude: LEO at 372 km  
initial velocity: 7684.5 m/s  
target altitude: GEO at 35,863 km  
target velocity: 1612.6 to 2110.0 m/s  
final velocity: GEO velocity -- 3071.9 m/s

The apogee velocity of the transfer orbit was varied between 1612.6 m/s -- the Hohmann transfer orbit apogee velocity -- and 2110.0 m/s, depending on the thrust level. The actual target velocity was arrived at through trial and error as the one that yielded the minimum total mission delta-v for a given thrust level. For high thrust, the optimum target velocity is clearly near the Hohmann orbit apogee velocity, since the Hohmann transfer is optimal for infinite thrust capability. The optimum target velocity will increase as the thrust level decreases, since, for infinitesimal thrust, the target velocity is exactly equal to the circular velocity and the vehicle is very close to circular velocity throughout its trajectory (see ref. 4).

The near-optimum trajectory to be used for performance comparisons was found by requiring the vehicle to thrust in a direction always tangential to its flight path. When the instantaneous orbit -- the orbit it would be in if it stopped thrusting -- had an apogee at GEO altitude, the vehicle was allowed to stop thrusting, and the required delta-v calculated at that point. (For details of how this trajectory can be found, see Appendix D.)

The resulting trajectory for a sample LEO to GEO transfer with initial acceleration of 0.03 g's and thrust angle limit of  $\pm 25$  degrees is shown in polar form in figure 4-1. Since the target velocity of

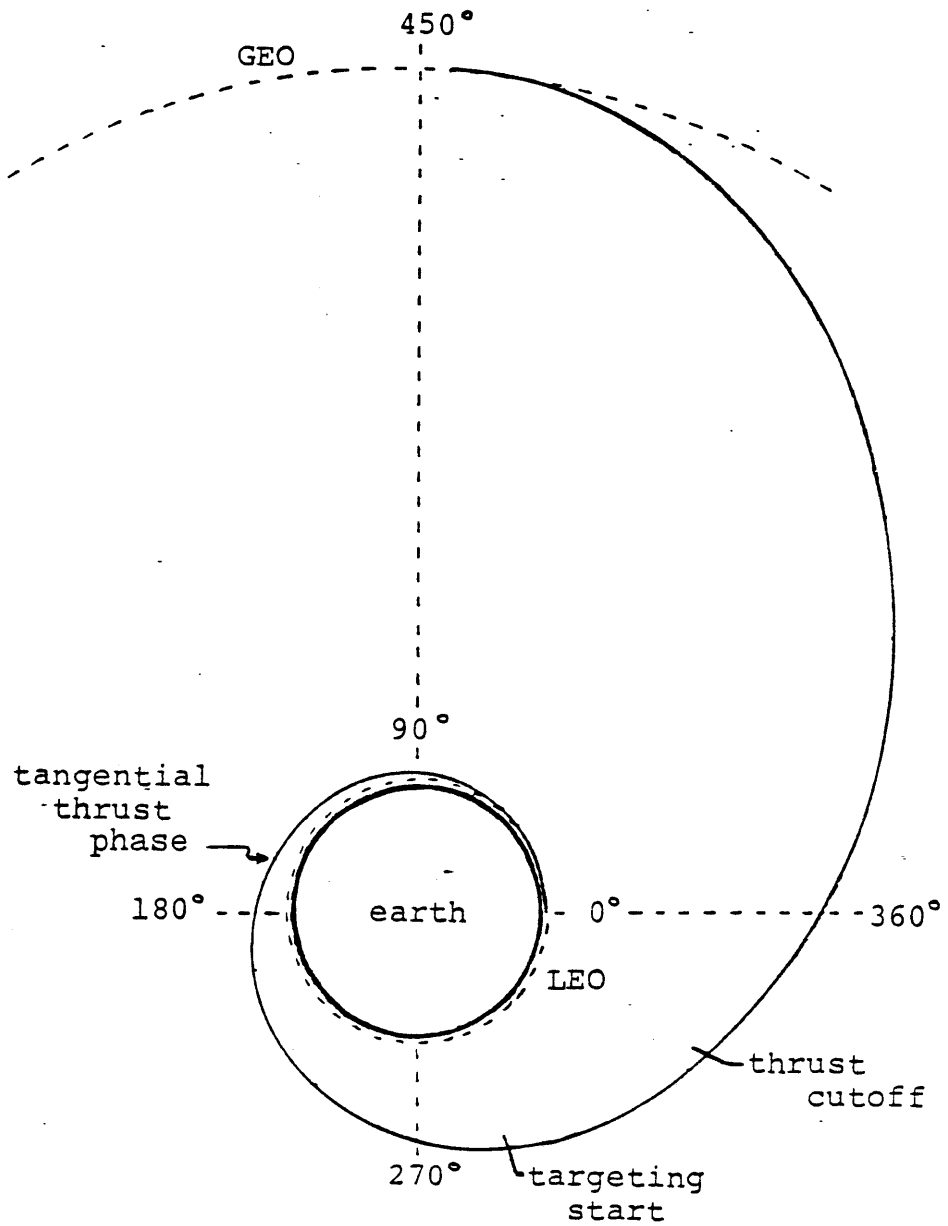


Figure 4-1 Sample LEO to GEO Transfer Trajectory

2110 m/s led to a transfer orbit with a perigee at an altitude of 6600 km, a tangential thrust trajectory was followed, as explained in section 3.5, until the vehicle reached the transfer orbit perigee altitude. After that point, the orbit-fitting targeting technique was used until the vehicle velocity matched the required velocity and could stop

thrusting. For the remainder of the trajectory, the vehicle coasted along the elliptical transfer orbit until it achieved the target conditions.

Figure 4-2 shows the plots of the individual variables versus time, giving the behavior of the trajectory in more detail. The thrust is sustained at its maximum value until cutoff. The thrust angle moves to its lower limit as soon as the targeting is initiated, then climbs to a slightly positive value, and then can be considered equal to zero after thrust cutoff, even though minute corrections are taking place, as discussed in section 4.2. The altitude increases steadily until GEO is reached. The velocity increases at first, then decreases as the vehicle spirals outward, gaining potential energy rather than kinetic energy. When the targeting begins and the vehicle is commanded to pitch down, the velocity increases again until it reaches the required velocity for the vehicle's current altitude. After thrust cutoff, the vehicle coasts, trading kinetic for potential energy as the velocity decreases until it reaches the target value. The flight path angle increases at first as the vehicle spirals outward -- unsteadily because of the changing acceleration as the vehicle burns off fuel. The rate of increase of the flight path angle slows when targeting begins and the vehicle is commanded to thrust slightly downward, then steepens as the thrust angle becomes positive. At thrust cutoff, the flight path angle has matched that required on the transfer orbit, and it follows the transfer orbit profile until the vehicle reaches apogee where the flight path angle crosses zero.

Figure 4-3 shows a partial listing of the actual output from the

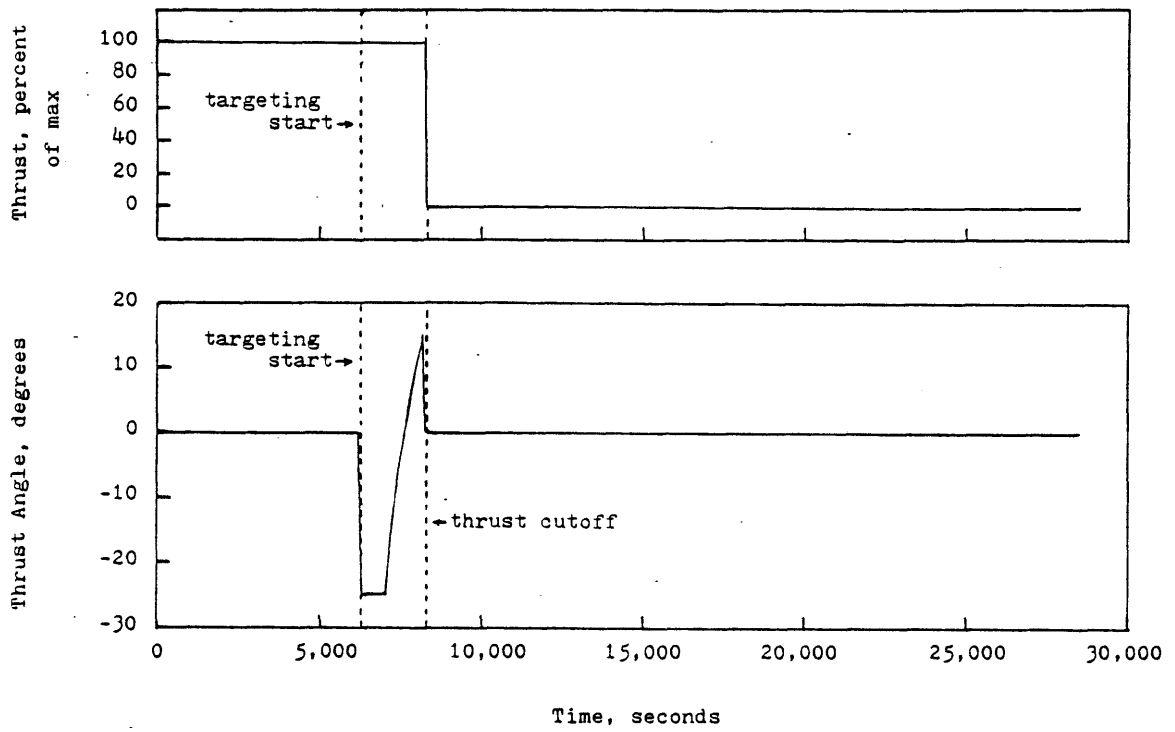


Figure 4-2a Thrust, Thrust Angle vs Time -- LEO to GEO Transfer

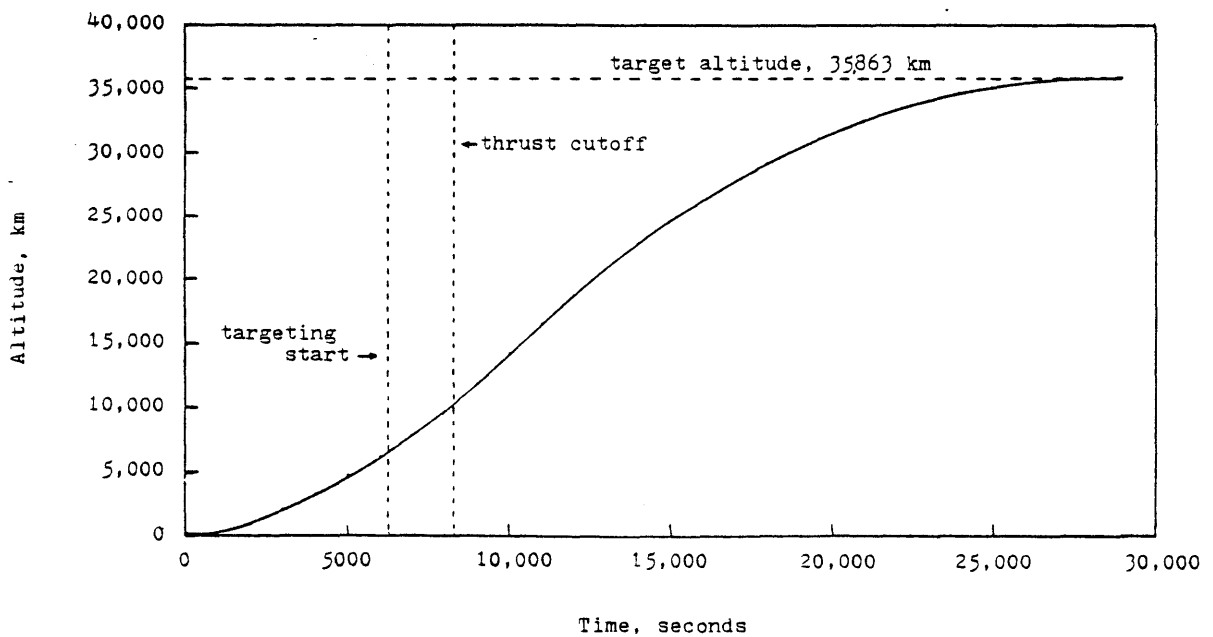


Figure 4-2b Altitude vs Time -- LEO to GEO Transfer

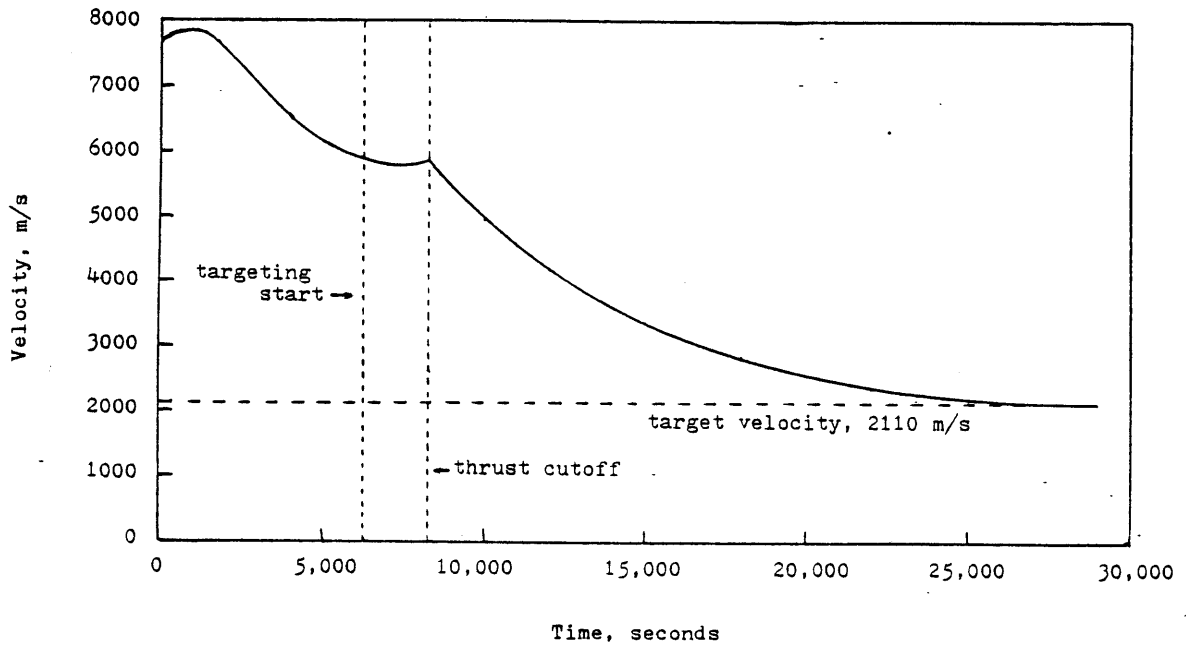


Figure 4-2c Velocity vs Time -- LEO to GEO Transfer

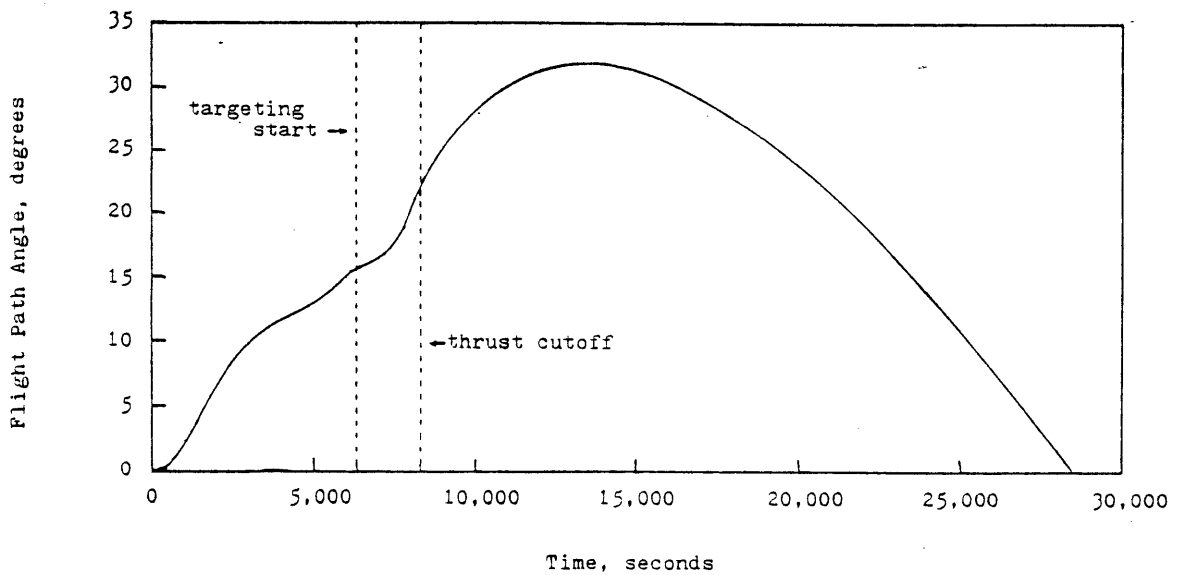


Figure 4-2d Flight Path Angle vs Time -- LEO to GEO Transfer

trajectory, with values printed out every five integration steps. (A complete listing is shown in Appendix E.) The values for the velocity, altitude, and flight path angle in the last line of the trajectory show that the target values are achieved exactly (to as many significant figures as are displayed) at the point where the flight path angle is closest to zero, as desired. This is typical of the accuracy achieved for all cases tested, and demonstrates the basic viability of the technique. In addition, the algorithm remains relatively stable and well-behaved as the vehicle approaches the target.

The delta-v requirement of 3486 m/s for this particular case compares favorably with the tangential-thrust near-optimum value of 3425 m/s for the thrusting phase, with a total delta-v, including the circularization burn, of 4449 m/s, compared with 4410 m/s from the tangential-thrust trajectory. These values differ by only 0.88 %. The performance on this trajectory was improved slightly by having restricted the thrust angle, and without such a constraint, the trajectory would have required a total delta-v of 4515 m/s -- an increase of 66 m/s. The unrestrained trajectory, however, represents the worst-case performance for the orbit-fitting targeting technique, and thus other comparisons were made based on an unrestricted thrust angle trajectory. The figures in table 4-1 show that the technique performed better at higher thrust levels, and above 1 g initial acceleration, the trajectory very closely resembled a Hohmann transfer ellipse -- as it should, since a Hohmann transfer is optimal for high thrust levels.

Targeted Spaceflight Trajectory

LEO to GEO transfer with thrust/MQ=0.03 g.

Thrust= 7355.0 N, C= 4500.0 m/s, MQ= 25000.0 kg, Mfuel= 24000.0 kg  
 Sref= 0.0 m^2, hstart= 0.0 km, Ot= 20.0 sec  
 AlphaMax= 25.0 deg, AlphDotMax=360.0 deg/sec  
 TargetH= 35863.0 km, TargetV= 2110.0 m/s

| t   | V      | h       | Phi   | Sigma | VG   | alpha | Thrust  | DeltaV |
|-----|--------|---------|-------|-------|------|-------|---------|--------|
| 0.0 | 7684.5 | 372.000 | 0.000 | 0.000 | 0.00 | 0.00  | 100.000 | 0.00   |

Starting targeting now.

|        |        |          |       |         |      |      |         |        |
|--------|--------|----------|-------|---------|------|------|---------|--------|
| 100.0  | 7713.9 | 372.074  | .019  | 6.535   | 0.00 | 0.00 | 100.000 | 29.52  |
| 200.0  | 7742.8 | 372.746  | .088  | 13.095  | 0.00 | 0.00 | 100.000 | 59.23  |
| 300.0  | 7770.6 | 374.689  | .206  | 19.677  | 0.00 | 0.00 | 100.000 | 89.14  |
| 400.0  | 7796.3 | 378.562  | .372  | 26.280  | 0.00 | 0.00 | 100.000 | 119.25 |
| 500.0  | 7819.4 | 385.009  | .582  | 32.898  | 0.00 | 0.00 | 100.000 | 149.56 |
| 600.0  | 7839.2 | 394.643  | .835  | 39.526  | 0.00 | 0.00 | 100.000 | 180.08 |
| 700.0  | 7855.1 | 408.031  | 1.127 | 46.167  | 0.00 | 0.00 | 100.000 | 210.80 |
| 800.0  | 7866.6 | 425.699  | 1.454 | 52.784  | 0.00 | 0.00 | 100.000 | 241.74 |
| 900.0  | 7873.3 | 448.107  | 1.813 | 59.399  | 0.00 | 0.00 | 100.000 | 272.89 |
| 1000.0 | 7874.8 | 475.650  | 2.199 | 65.992  | 0.00 | 0.00 | 100.000 | 304.26 |
| 1100.0 | 7871.0 | 508.653  | 2.608 | 72.552  | 0.00 | 0.00 | 100.000 | 335.85 |
| 1200.0 | 7861.6 | 547.362  | 3.035 | 79.071  | 0.00 | 0.00 | 100.000 | 367.66 |
| 1300.0 | 7846.6 | 591.949  | 3.475 | 85.539  | 0.00 | 0.00 | 100.000 | 399.70 |
| 1400.0 | 7826.2 | 642.505  | 3.923 | 91.945  | 0.00 | 0.00 | 100.000 | 431.97 |
| 1500.0 | 7800.5 | 699.049  | 4.376 | 98.280  | 0.00 | 0.00 | 100.000 | 464.47 |
| 1600.0 | 7769.8 | 761.530  | 4.829 | 104.535 | 0.00 | 0.00 | 100.000 | 497.21 |
| 1700.0 | 7734.3 | 829.834  | 5.279 | 110.703 | 0.00 | 0.00 | 100.000 | 530.19 |
| 1800.0 | 7694.5 | 903.790  | 5.721 | 116.777 | 0.00 | 0.00 | 100.000 | 563.41 |
| 1900.0 | 7650.8 | 983.180  | 6.154 | 122.750 | 0.00 | 0.00 | 100.000 | 596.88 |
| 2000.0 | 7603.7 | 1067.746 | 6.574 | 128.616 | 0.00 | 0.00 | 100.000 | 630.60 |
| 2100.0 | 7553.5 | 1157.203 | 6.980 | 134.373 | 0.00 | 0.00 | 100.000 | 664.57 |
| 2200.0 | 7500.8 | 1251.242 | 7.370 | 140.017 | 0.00 | 0.00 | 100.000 | 698.81 |
| 2300.0 | 7446.1 | 1349.542 | 7.743 | 145.546 | 0.00 | 0.00 | 100.000 | 733.30 |
| 2400.0 | 7389.8 | 1451.774 | 8.097 | 150.958 | 0.00 | 0.00 | 100.000 | 768.06 |
| 2500.0 | 7332.2 | 1557.614 | 8.433 | 156.253 | 0.00 | 0.00 | 100.000 | 803.10 |

: : : : : : : : :

|         |        |           |       |         |     |        |      |         |
|---------|--------|-----------|-------|---------|-----|--------|------|---------|
| 27000.0 | 2123.9 | 35731.360 | 4.764 | 444.066 | .00 | 81.05  | .025 | 3506.48 |
| 27100.0 | 2122.1 | 35748.400 | 4.448 | 444.354 | .01 | 85.08  | .088 | 3506.51 |
| 27200.0 | 2120.5 | 35764.270 | 4.131 | 444.641 | .00 | 87.14  | .038 | 3506.55 |
| 27300.0 | 2118.9 | 35778.950 | 3.813 | 444.929 | .00 | -98.54 | .013 | 3506.58 |
| 27400.0 | 2117.5 | 35792.450 | 3.495 | 445.216 | .00 | 90.02  | .029 | 3506.60 |
| 27500.0 | 2116.2 | 35804.770 | 3.177 | 445.503 | .02 | 85.26  | .139 | 3506.62 |
| 27600.0 | 2115.0 | 35815.900 | 2.858 | 445.790 | .00 | 138.93 | .002 | 3506.64 |
| 27700.0 | 2113.9 | 35825.860 | 2.539 | 446.077 | .00 | 90.00  | .004 | 3506.69 |
| 27800.0 | 2113.0 | 35834.640 | 2.219 | 446.363 | .00 | 86.02  | .027 | 3506.74 |
| 27900.0 | 2112.2 | 35842.230 | 1.900 | 446.650 | .02 | 87.50  | .130 | 3506.81 |
| 28000.0 | 2111.5 | 35848.640 | 1.580 | 446.936 | .00 | -99.60 | .010 | 3506.84 |
| 28100.0 | 2111.0 | 35853.870 | 1.260 | 447.223 | .00 | 90.00  | .010 | 3506.87 |
| 28200.0 | 2110.5 | 35857.930 | .940  | 447.509 | .02 | 88.25  | .186 | 3506.91 |
| 28300.0 | 2110.2 | 35860.800 | .621  | 447.795 | .01 | -90.96 | .113 | 3507.01 |
| 28400.0 | 2110.1 | 35862.490 | .299  | 448.081 | .00 | 90.00  | .010 | 3507.19 |

Alignment achieved; coasting now.

|         |        |           |       |         |      |      |       |         |
|---------|--------|-----------|-------|---------|------|------|-------|---------|
| 28500.0 | 2110.0 | 35863.000 | -.044 | 448.368 | 0.00 | 0.00 | 0.000 | 3508.69 |
|---------|--------|-----------|-------|---------|------|------|-------|---------|

Target achieved.

Figure 4-3 Partial LEO to GEO Transfer Trajectory

---

| Thrust/M <sub>0</sub><br>g's | V <sub>2</sub> for<br>min ΔV<br>m/s | ΔV Tangential<br>Thrust<br>m/s | ΔV Targeted<br>Trajectory<br>m/s | % Difference<br>in ΔV's |
|------------------------------|-------------------------------------|--------------------------------|----------------------------------|-------------------------|
| 0.03                         | 2110.0                              | 4409.9                         | 4514.6                           | 2.49                    |
| 0.10                         | 1680.0                              | 3998.8                         | 4081.0                           | 2.06                    |
| 0.30                         | 1620.0                              | 3871.0                         | 3897.2                           | 0.667                   |
| 1.00                         | 1612.6                              | 3866.8                         | 3872.6                           | 0.150                   |
| 3.00                         | 1612.6                              | 3865.9                         | 3867.1                           | 0.031                   |

Table 4-1 ΔV Comparisons for LEO to GEO Transfer

---

#### 4.4 Launch Trajectories

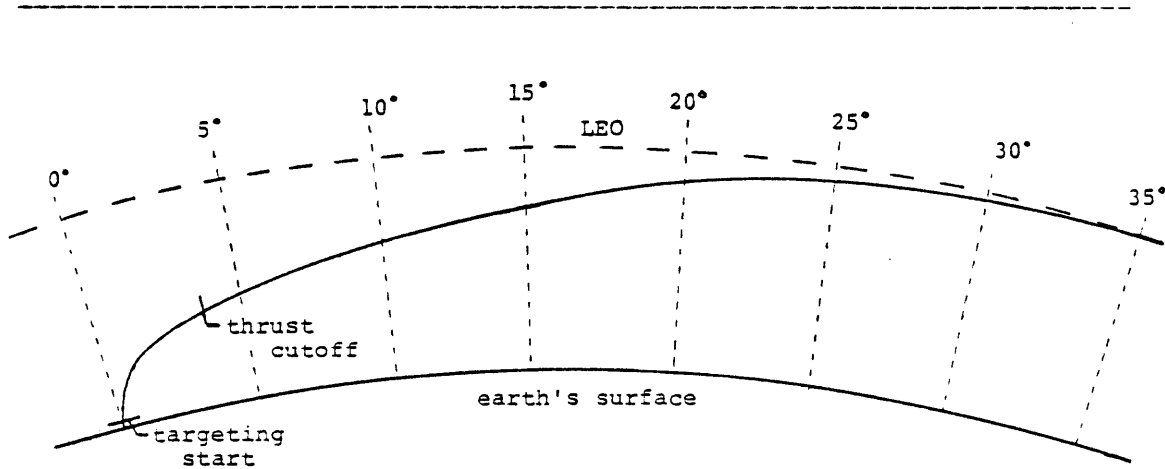
The launch vehicle model was chosen, for simplicity, to represent a single-stage-to-orbit vehicle with a 100,000 kg launch mass. A Space Shuttle Main Engine (SSME) could be used, and the vehicle parameters, initial and target conditions were as follows:

specific impulse: 450 seconds  
 initial mass: 100,000 kg  
 thrust: 1.6 million Newtons  
 initial altitude: 0 km  
 initial velocity: 1 m/s  
 target altitude: LEO at 372 km  
 target velocity: 7000 m/s  
 final velocity: LEO velocity -- 7684.5 m/s

The initial velocity is 1 m/s rather than 0 m/s to avoid a singularity in the equations of motion. The vehicle aerodynamic model was chosen to resemble a streamlined delta-wing shuttle-type vehicle, with a variable reference area which could be scaled with the vehicle size, but a value of 50 sq. m was used for most of the simulations. Since flight near the earth's surface is dominated by atmospheric

effects and it is desirable to climb out of the atmosphere as soon as possible, open-loop guidance is commonly used in the early phase of a launch and the targeting initiated only when atmospheric effects are no longer dominant. The vehicle model in this case followed a vertical ascent, gravity-turn trajectory until the atmospheric effects had diminished and the targeting algorithm could be followed. The altitude at which the targeting began could be set to equal any desired value, and was typically between 5 and 10 km. The thrust angle was limited to  $\pm 10$  degrees, and its maximum allowed rate of change was 2 deg/sec.

The trajectory from a sample launch simulation is shown in polar form in figure 4-4. In this run, targeting was initiated at an altitude of 10 km.



Note: trajectory altitude scale is 2X actual scale

Figure 4-4 Sample Launch Trajectory

The behavior of the individual variables is shown in figure 4-3. As soon as targeting begins, 44 seconds into the flight, the thrust

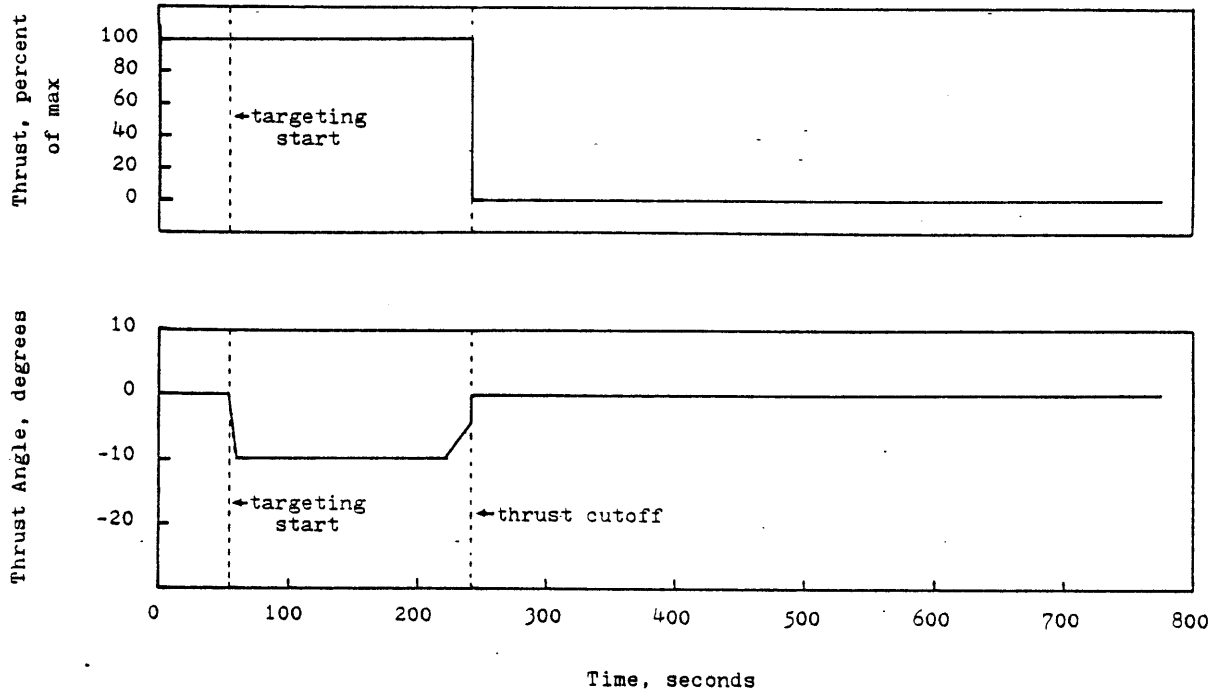


Figure 4-5a Thrust, Thrust Angle vs Time -- Launch to LEO

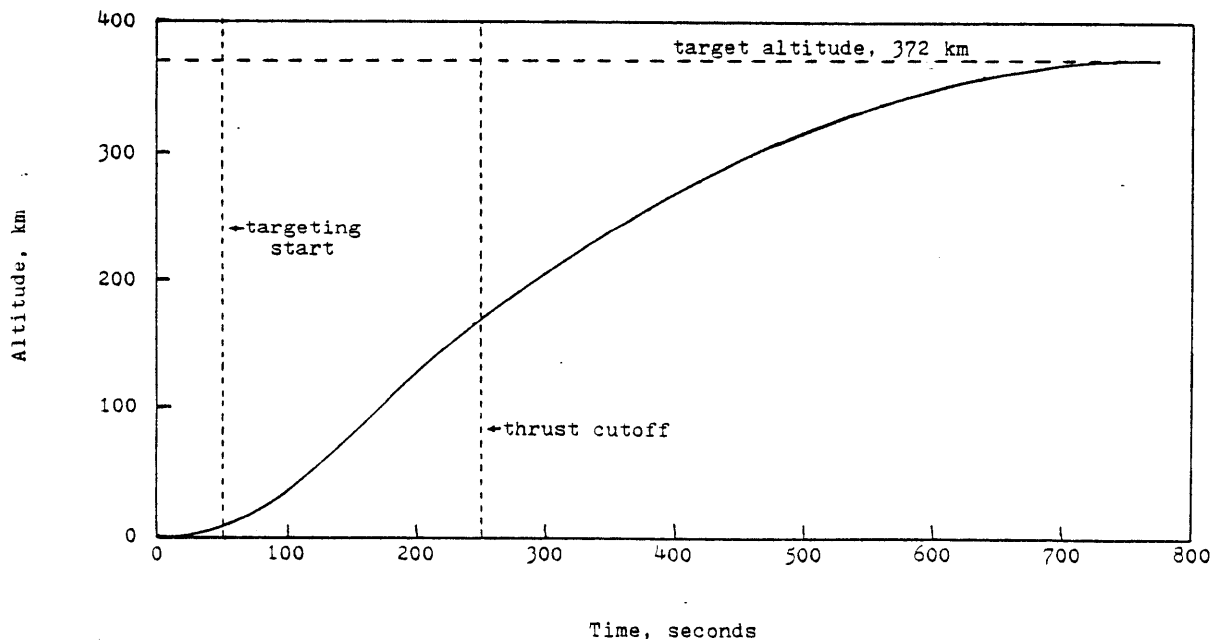


Figure 4-5b Altitude vs Time -- Launch to LEO

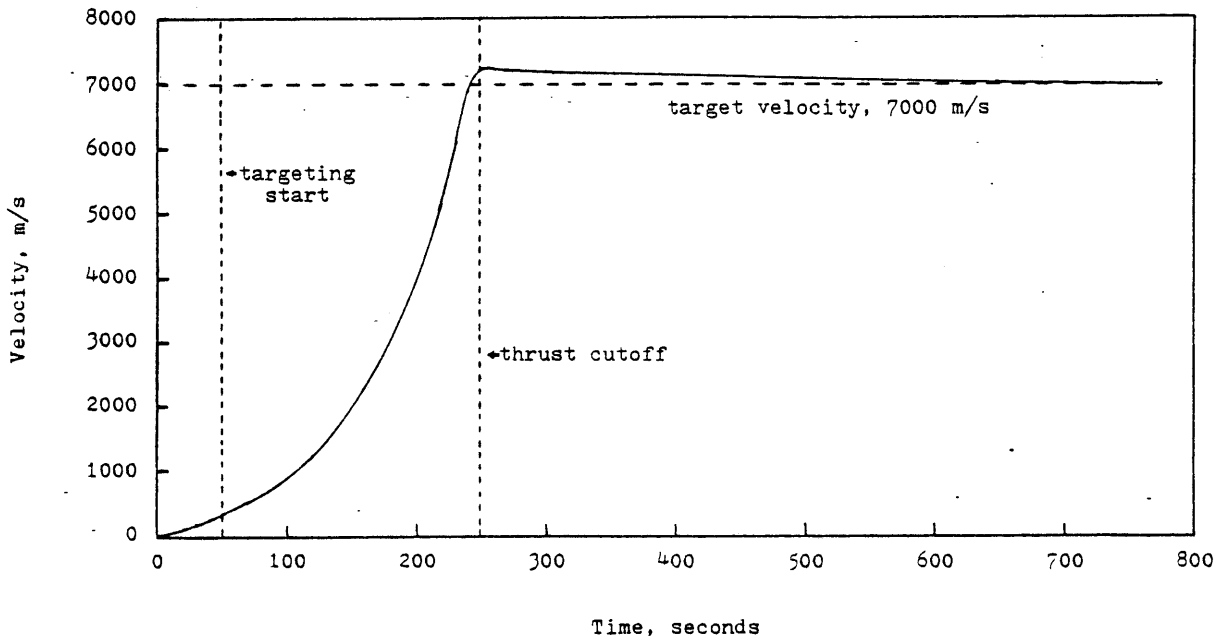


Figure 4-5c Velocity vs Time -- Launch to LEO

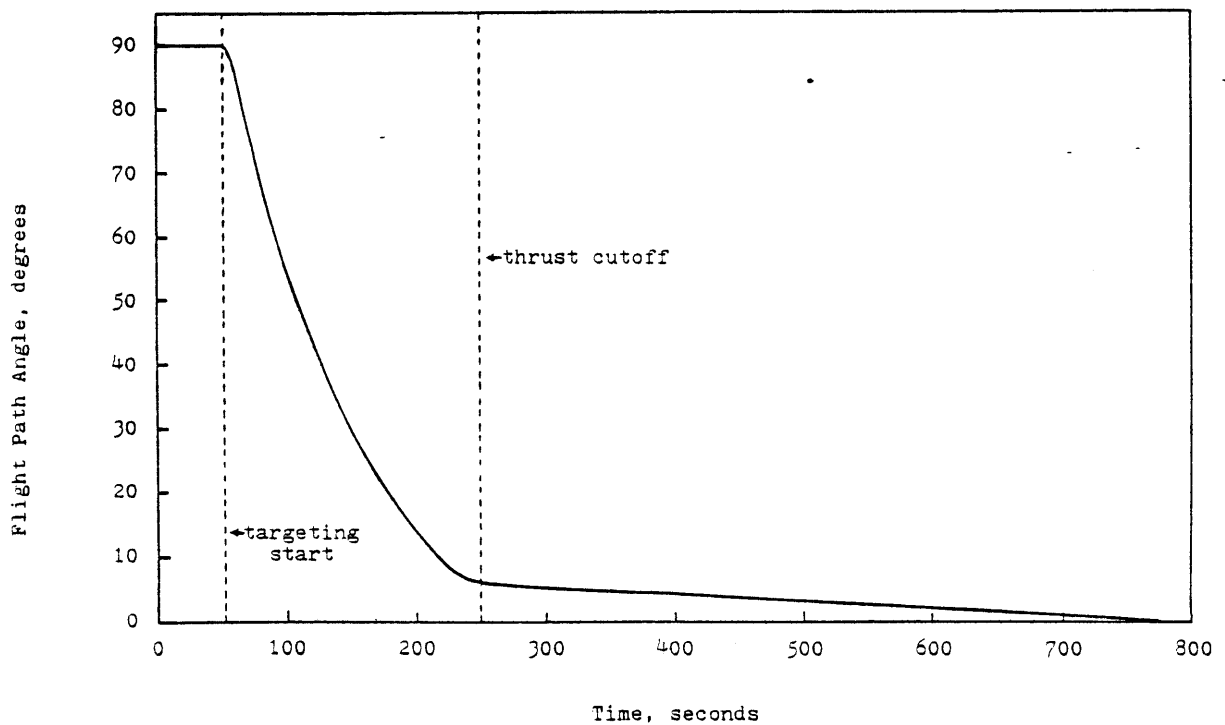


Figure 4-5d Flight Path Angle vs Time -- Launch to LEO

angle decreases at the maximum allowed rate to its maximum negative value of -10 degrees, in order to pitch the vehicle down to start gaining circumferential velocity. If the thrust angle were not constrained, the initial value, when targeting began, would be close to -80 degrees, since the transfer orbit requires a near-horizontal flight path angle at that altitude. If allowed to proceed unconstrained, the vehicle would pitch over very quickly and attempt to maintain the low flight path angle required. But, since the vehicle does not have sufficient velocity or thrust to maintain altitude in near-horizontal flight, and because atmospheric drag further impedes it, it would continue to pitch down and eventually impact the earth. Hence the necessity for constraining the maximum value of the thrust angle and its rate of change. With these constraints, the vehicle is permitted to pitch over only slowly, so that, by the time it approaches a horizontal flight path, it has left the atmosphere and it has sufficient velocity to maintain the desired flight path angle. Targeting could be initiated at a lower altitude than at 10 km, but with penalties in performance since the vehicle would be required to spend more time in the atmosphere because the pitch-over maneuver would be started earlier. For example, the same run with targeting initiated at 5 km requires a delta-v, during the thrusting phase, of approximately 200 m/s more than the run with targeting initiated at 10 km. Initiating targeting at higher altitudes than 10 km produces only small changes in the resulting delta-v, since, at 10 km, the vehicle is already above the densest part of the atmosphere.

The delta-v required for launch and LEO insertion for this run is 9530 m/s. If the rotation of the earth is included, a bonus of 408 m/s

from a 28.5 degree inclination launch site brings the actual delta-v down to 9122 m/s. This compares very well with commonly accepted launch delta-v requirements of 30,000 ft/sec or 9144 m/s, and shows that the technique performs acceptably well for the difficult problem of generating a feasible launch trajectory.

## CHAPTER 5

### Conclusions

The cases studied in the preceding section demonstrate both the excellent targeting accuracy and the good performance achievable with the orbit-fitting targeting technique. While, in some cases, an understanding of the technique is required and appropriate constraints must be applied, the technique is basically easy to implement and to use. It is ideally suited for use in applications such as design studies and mission feasibility studies, where reasonable trajectories must be generated for many different choices of design parameters, but where the exact optimum trajectory is not needed. It also provides a good initial feasible trajectory for use with gradient-search type optimization techniques that require such a trajectory as a starting point.

In addition, the orbit-fitting technique can be used to satisfy multiple choices of target parameter combinations, not just one designated set, thus making it more flexible than most current targeting methods. Because of its general nature, it could be applied to almost any two-dimensional, two-body spaceflight targeting problem involving target parameters among those discussed.

## APPENDIX A

### Alternative Choices of Target Parameters

#### A.1 Altitude, Velocity, and Transfer Angle as Target Parameters

In this case the unknown variable is the flight path angle,  $\phi_2$ , which can be found by using equation (2.13). If the substitution

$$x = \cot \frac{\phi_2}{2} \quad (\text{A.1})$$

is made for  $\phi_2$ , with

$$\sin \phi_2 = \frac{2x}{1+x^2} \quad \cos \phi_2 = -\frac{1-x^2}{1+x^2} \quad (\text{A.2})$$

the result is a quartic in  $x$ , of the form

$$(a_1 - a_2)x^4 - bx^3 - 2(a_1 + a_2)x^2 + bx + (a_1 - a_2) = 0 \quad (\text{A.3})$$

where

$$a_1 = r_2 v_2^2 (r_1 \cos \sigma - r_2) \quad (\text{A.4})$$

$$a_2 = \mu r_1 (1 - \cos \sigma) \quad (\text{A.5})$$

$$b = 2r_1 r_2 v_2^2 \sin \sigma \quad (\text{A.6})$$

Equation (A.3) can be solved for  $x$  by traditional root finding methods, and then equation (A.1) used to find  $\phi_2$ . Two feasible roots will be

found as solutions to the quartic, due to the axial symmetry of the hyperbolic locus of velocities at  $P_1$ . The smaller angle will correspond to a shorter time of flight from  $P_1$  to  $P_2$ , and should therefore be selected as the solution.

A.2 Velocity, Flight Path Angle, and Transfer Angle as Target Parameters

The variable left to be found now is the radius,  $r_2$ . Collecting terms in  $r_2$  from equation (2.13) gives a quadratic in  $r_2$  of the form

$$ar_2^2 + br_2 + c = 0 \quad (\text{A.7})$$

where

$$a = v_2^2 \cos^2 \phi_2 \quad (\text{A.8})$$

$$b = r_1 v_2^2 \cos \phi_2 \cos(\phi_2 - \sigma) \quad (\text{A.9})$$

$$c = \mu r_1 (1 - \cos \sigma) \quad (\text{A.10})$$

Equation (A.7) can be solved easily for  $r_2$  with the quadratic equation.

A.3 Altitude, Flight Path Angle, and Transfer Angle as Target Parameters

This case leaves the velocity,  $V_2$ , to be found. Equation (2.13) can be solved for  $V_2$  directly, giving

$$V_2^2 = \frac{\mu r_1 (1 - \cos \sigma)}{r_1 r_2 \cos \phi_2 \cos(\phi_2 - \sigma) - r_2^2 \cos^2 \phi_2} \quad (\text{A.11})$$

## APPENDIX B

### Alternative Technique for Finding the Required Velocity

The orbit-fitting targeting technique was originally developed by applying the orbital boundary-value problem as a guidance technique. Therefore, the method implemented in the computer routines for finding the required velocity involved the chordal and radial components,  $V_c$  and  $V_p$ , (as developed in section 2.3), since these components were essential in solving the original boundary-value problem. After extending the targeting capabilities of the technique, however, and developing equation (2.13), which makes it possible to determine the values of all the target parameters, regardless of which are used as the designated set, it is no longer necessary to resolve the required velocity into chordal and radial components. Instead, it can be found quite easily, as follows.

The semi-major axis of the transfer orbit can be found by using the values for the radius and velocity at the target point from

$$a = \left( \frac{2}{r_2} - \frac{v_2^2}{\mu} \right)^{-1} \quad (\text{B.1})$$

Since the radius at the initial point is known,  $V_{\text{req}}$  can be determined from

$$v_{\text{req}}^2 = \mu \left( \frac{2}{r_1} - \frac{1}{a} \right) \quad (\text{B.2})$$

Substituting equation (B.1) for  $a$  and simplifying gives

$$v_{\text{req}}^2 = \frac{2\mu(r_2 - r_1)}{r_1 r_2} - v_2^2 \quad (\text{B.3})$$

The angular momentum of the transfer orbit,  $\bar{h}$ , is defined as

$$\bar{h} = \bar{r} \times \bar{v} \quad (\text{B.4})$$

and its magnitude can be found by using values from the target point

$$h = r_2 v_2 \cos \phi_2 \quad (\text{B.5})$$

Since the angular momentum of an orbit is constant,  $h$  can be used to find  $\theta$ , since at the initial point

$$h = r_1 v_{\text{req}} \cos \theta \quad (\text{B.6})$$

Thus, substituting equation (B.5) for  $h$  and solving for  $\theta$  gives

$$\theta = \arccos \left( \frac{r_2 v_2 \cos \phi_2}{r_1 v_{\text{req}}} \right) \quad (\text{B.7})$$

Equations (B.3) and (B.7) now give a simple means for finding  $v_{\text{req}}$  and  $\theta$ , once  $r_2$ ,  $v_2$ , and  $\phi_2$  are known.

## APPENDIX C

### Spacecraft Equations of Motion

The equations of motion for a vehicle in flight around a spherical non-rotating earth, including atmospheric effects, can be found by resolving the forces acting on the vehicle into components in the radial and circumferential directions. The sum of the force components in the radial direction,  $F_r$ , is found to be

$$F_r = T \sin(\alpha + \phi) + L \cos \phi - D \sin \phi - Mg \quad (C.1)$$

and the sum of the components in the circumferential direction,  $F_c$ , is

$$F_c = T \cos(\alpha + \phi) - L \sin \phi - D \cos \phi \quad (C.2)$$

where  $T$  is the vehicle thrust,  $L$  the lift,  $D$  the drag,  $M$  the mass,  $g$  the local gravitational acceleration,  $\alpha$  the angle of attack, and  $\phi$  the flight path angle. These equations can be converted to an earth-fixed reference frame by using the relationships

$$\ddot{r} = \frac{F_r}{M} + r\dot{\sigma}^2 \quad (C.3)$$

$$r\ddot{\sigma} = \frac{F_c}{M} - 2\dot{r}\dot{\sigma} \quad (C.4)$$

where  $r$  is the distance from the center of the earth, and  $\sigma$  is the downrange angle measured from an initial point. The vehicle pitch be-

havior can be found by equating the normal components of acceleration (perpendicular to the velocity),  $a_n$ , with the centrifugal acceleration caused by the vehicle pitching about some point other than its center of mass. The angular pitch rate,  $\dot{\theta}$ , is then

$$\dot{\theta} = \frac{a_n}{V} \quad (C.5)$$

where  $a_n$  is

$$a_n = \frac{L + T \sin \alpha}{M} - g \cos \phi \quad (C.6)$$

The pitch rate seen from the earth-fixed reference frame is then

$$\dot{\phi} = \dot{\theta} + \dot{\sigma} \quad (C.7)$$

All together, the equations of motion, reduced to first order, are

$$\dot{V}_r = \ddot{r} = \frac{T \sin(\alpha + \phi) + L \cos \phi - D \sin \phi}{M} - g + r\Omega^2 \quad (C.8)$$

$$\dot{\Omega} = \ddot{\sigma} = \frac{1}{r} \frac{T \cos(\alpha + \phi) - L \sin \phi - D \cos \phi}{M} - 2 \frac{V}{r} \quad (C.9)$$

$$\dot{r} = V_r \quad (C.10)$$

$$\dot{\sigma} = \Omega \quad (C.11)$$

$$\dot{\phi} = \frac{1}{V} \frac{L + T \sin \alpha}{M} - g \cos \phi + \Omega \quad (C.12)$$

$$\dot{M} = -\frac{T}{c} \quad (C.13)$$

where  $c$  is the characteristic rocket exhaust velocity, found from the specific impulse and the relation  $c = I_{sp}g$ .

## APPENDIX D

### Tangential Thrust Transfers

When performing orbital maneuvers, thrusting tangentially to the flight path generally produces a greater increase in orbital energy per unit mass of propellant expended than does thrusting in any other direction. A near-optimum LEO to GEO transfer trajectory for any thrust level can be found by following this principal. Starting from LEO, the vehicle thrusts tangentially to its flight path, while continuously monitoring the parameters of the instantaneous orbit it would be in if it suddenly stopped thrusting. The parameters to be monitored -- the semi-major axis  $a$ , eccentricity  $e$ , and orbital apogee  $r_a$  -- can be found from the following relationships

$$a = \left( \frac{2}{r} - \frac{v^2}{\mu} \right)^{-1} \quad (D.1)$$

$$e^2 = 1 - \frac{r^2 v^2}{\mu a} \cos^2 \phi \quad (D.2)$$

$$r_a = a (1 + e) \quad (D.3)$$

Whenever the vehicle reaches the point where  $r_a$  is equal to the desired GEO radius,  $r_{GEO}$ , the engines can be shut off and the vehicle allowed to coast the remainder of the way to  $r_{GEO}$ . At apogee the vehicle must circularize its orbit, requiring a delta- $v$  equal to the

difference between GEO circular velocity,  $V_{\text{GEO}}$ , and the transfer orbit apogee velocity,  $V_a$ , found from

$$V_a^2 = \mu \left( \frac{2}{r_a} - \frac{1}{a} \right) \quad (\text{D.4})$$

The delta-v required for circularization,  $\Delta V_c$ , is thus  $\Delta V_c = V_{\text{GEO}} - V_a$ . The delta-v required for the thrusting phase of the transfer,  $\Delta V_t$ , can be found from the rocket equation

$$\Delta V_t = -c \ln \left( \frac{M_f}{M_o} \right) \quad (\text{D.5})$$

where  $M_f$  is the vehicle mass at engine cutoff, and  $M_o$  is its initial launch mass. The total delta-v required for a tangential thrust LEO to GEO transfer is therefore

$$\Delta V_{\text{total}} = \Delta V_t + \Delta V_c \quad (\text{D.6})$$

APPENDIX E

Sample Trajectories

E.1 LEO to GEO Transfer

Targeted Spaceflight Trajectory

LEO to GEO transfer with thrust/MQ=0.03 g.

Thrust= 7355.0 N, C= 4500.0 m/s, MQ= 25000.0 kg, Mfuel= 24000.0 kg  
 Sref= 0.0 m^2, hstart= 0.0 km, Ot= 20.0 sec  
 AlphaMax= 25.0 deg, AlphDotMax=360.0 deg/sec  
 TargetH= 35863.0 km, TargetV= 2110.0 m/s

| t   | V      | h       | Phi   | Siema | VG   | alpha | Thrust  | DeltaV |
|-----|--------|---------|-------|-------|------|-------|---------|--------|
| 0.0 | 7684.6 | 372.000 | 0.000 | 0.000 | 0.00 | 0.00  | 100.000 | 0.00   |

Starting targetings now.

|        |        |          |        |         |      |      |         |         |
|--------|--------|----------|--------|---------|------|------|---------|---------|
| 100.0  | 7713.9 | 372.074  | .019   | 6.535   | 0.00 | 0.00 | 100.000 | 29.52   |
| 200.0  | 7742.8 | 372.746  | .066   | 13.095  | 0.00 | 0.00 | 100.000 | 59.23   |
| 300.0  | 7770.6 | 374.689  | .206   | 19.677  | 0.00 | 0.00 | 100.000 | 89.14   |
| 400.0  | 7796.3 | 378.962  | .372   | 26.280  | 0.00 | 0.00 | 100.000 | 119.25  |
| 500.0  | 7819.4 | 385.009  | .582   | 32.898  | 0.00 | 0.00 | 100.000 | 149.56  |
| 600.0  | 7839.2 | 394.643  | .835   | 39.526  | 0.00 | 0.00 | 100.000 | 180.08  |
| 700.0  | 7855.1 | 408.031  | 1.127  | 46.157  | 0.00 | 0.00 | 100.000 | 210.80  |
| 800.0  | 7866.6 | 425.699  | 1.454  | 52.784  | 0.00 | 0.00 | 100.000 | 241.74  |
| 900.0  | 7873.3 | 448.107  | 1.813  | 59.399  | 0.00 | 0.00 | 100.000 | 272.89  |
| 1000.0 | 7874.8 | 475.650  | 2.199  | 65.992  | 0.00 | 0.00 | 100.000 | 304.26  |
| 1100.0 | 7871.0 | 508.653  | 2.608  | 72.552  | 0.00 | 0.00 | 100.000 | 335.85  |
| 1200.0 | 7861.6 | 547.362  | 3.035  | 79.071  | 0.00 | 0.00 | 100.000 | 367.66  |
| 1300.0 | 7846.6 | 591.949  | 3.475  | 85.539  | 0.00 | 0.00 | 100.000 | 399.70  |
| 1400.0 | 7826.2 | 642.505  | 3.923  | 91.945  | 0.00 | 0.00 | 100.000 | 431.97  |
| 1500.0 | 7800.5 | 699.049  | 4.376  | 98.280  | 0.00 | 0.00 | 100.000 | 464.47  |
| 1600.0 | 7769.6 | 761.530  | 4.829  | 104.535 | 0.00 | 0.00 | 100.000 | 497.21  |
| 1700.0 | 7734.3 | 829.834  | 5.279  | 110.703 | 0.00 | 0.00 | 100.000 | 530.19  |
| 1800.0 | 7694.5 | 903.790  | 5.721  | 116.777 | 0.00 | 0.00 | 100.000 | 563.41  |
| 1900.0 | 7650.8 | 983.180  | 6.154  | 122.750 | 0.00 | 0.00 | 100.000 | 596.88  |
| 2000.0 | 7603.7 | 1067.746 | 6.574  | 128.616 | 0.00 | 0.00 | 100.000 | 630.60  |
| 2100.0 | 7553.5 | 1157.203 | 6.980  | 134.373 | 0.00 | 0.00 | 100.000 | 664.57  |
| 2200.0 | 7500.8 | 1251.242 | 7.370  | 140.017 | 0.00 | 0.00 | 100.000 | 698.81  |
| 2300.0 | 7446.1 | 1349.542 | 7.743  | 145.546 | 0.00 | 0.00 | 100.000 | 733.30  |
| 2400.0 | 7389.8 | 1451.774 | 8.097  | 150.958 | 0.00 | 0.00 | 100.000 | 768.06  |
| 2500.0 | 7332.2 | 1557.614 | 8.433  | 156.253 | 0.00 | 0.00 | 100.000 | 803.10  |
| 2600.0 | 7273.9 | 1666.740 | 8.750  | 161.431 | 0.00 | 0.00 | 100.000 | 838.40  |
| 2700.0 | 7215.2 | 1778.841 | 9.049  | 166.494 | 0.00 | 0.00 | 100.000 | 873.99  |
| 2800.0 | 7156.3 | 1893.622 | 9.330  | 171.442 | 0.00 | 0.00 | 100.000 | 909.86  |
| 2900.0 | 7097.7 | 2010.804 | 9.594  | 176.277 | 0.00 | 0.00 | 100.000 | 946.02  |
| 3000.0 | 7039.5 | 2130.127 | 9.840  | 181.002 | 0.00 | 0.00 | 100.000 | 982.47  |
| 3100.0 | 6982.0 | 2251.354 | 10.071 | 185.619 | 0.00 | 0.00 | 100.000 | 1019.22 |
| 3200.0 | 6925.5 | 2374.267 | 10.288 | 190.132 | 0.00 | 0.00 | 100.000 | 1056.27 |
| 3300.0 | 6870.0 | 2498.572 | 10.491 | 194.542 | 0.00 | 0.00 | 100.000 | 1093.53 |
| 3400.0 | 6815.6 | 2624.396 | 10.681 | 198.853 | 0.00 | 0.00 | 100.000 | 1131.30 |
| 3500.0 | 6762.7 | 2751.287 | 10.861 | 203.068 | 0.00 | 0.00 | 100.000 | 1169.29 |

|         |        |           |        |         |         |        |         |         |
|---------|--------|-----------|--------|---------|---------|--------|---------|---------|
| 3600.0  | 6711.1 | 2879.214  | 11.030 | 207.190 | 0.00    | 0.00   | 100.000 | 1207.60 |
| 3700.0  | 6661.0 | 3008.068  | 11.191 | 211.223 | 0.00    | 0.00   | 100.000 | 1246.24 |
| 3800.0  | 6612.4 | 3137.759  | 11.345 | 215.169 | 0.00    | 0.00   | 100.000 | 1285.22 |
| 3900.0  | 6565.5 | 3268.216  | 11.493 | 219.031 | 0.00    | 0.00   | 100.000 | 1324.54 |
| 4000.0  | 6520.1 | 3399.383  | 11.636 | 222.813 | 0.00    | 0.00   | 100.000 | 1364.20 |
| 4100.0  | 6476.4 | 3531.222  | 11.775 | 226.516 | 0.00    | 0.00   | 100.000 | 1404.22 |
| 4200.0  | 6434.2 | 3663.710  | 11.912 | 230.145 | 0.00    | 0.00   | 100.000 | 1444.59 |
| 4300.0  | 6393.7 | 3796.840  | 12.047 | 233.702 | 0.00    | 0.00   | 100.000 | 1485.33 |
| 4400.0  | 6354.8 | 3930.616  | 12.182 | 237.188 | 0.00    | 0.00   | 100.000 | 1526.45 |
| 4500.0  | 6317.5 | 4065.056  | 12.318 | 240.607 | 0.00    | 0.00   | 100.000 | 1567.94 |
| 4600.0  | 6281.7 | 4200.189  | 12.456 | 243.961 | 0.00    | 0.00   | 100.000 | 1609.82 |
| 4700.0  | 6247.4 | 4336.053  | 12.596 | 247.253 | 0.00    | 0.00   | 100.000 | 1652.09 |
| 4800.0  | 6214.6 | 4472.699  | 12.740 | 250.483 | 0.00    | 0.00   | 100.000 | 1694.76 |
| 4900.0  | 6183.3 | 4610.184  | 12.889 | 253.655 | 0.00    | 0.00   | 100.000 | 1737.84 |
| 5000.0  | 6153.4 | 4748.575  | 13.042 | 256.769 | 0.00    | 0.00   | 100.000 | 1781.34 |
| 5100.0  | 6124.9 | 4887.945  | 13.202 | 259.829 | 0.00    | 0.00   | 100.000 | 1825.26 |
| 5200.0  | 6097.8 | 5028.377  | 13.368 | 262.835 | 0.00    | 0.00   | 100.000 | 1869.62 |
| 5300.0  | 6071.9 | 5169.956  | 13.542 | 265.789 | 0.00    | 0.00   | 100.000 | 1914.41 |
| 5400.0  | 6047.3 | 5312.776  | 13.724 | 268.693 | 0.00    | 0.00   | 100.000 | 1959.66 |
| 5500.0  | 6024.0 | 5456.935  | 13.914 | 271.548 | 0.00    | 0.00   | 100.000 | 2005.36 |
| 5600.0  | 6001.9 | 5602.537  | 14.114 | 274.355 | 0.00    | 0.00   | 100.000 | 2051.54 |
| 5700.0  | 5980.9 | 5749.688  | 14.323 | 277.116 | 0.00    | 0.00   | 100.000 | 2098.19 |
| 5800.0  | 5961.1 | 5898.500  | 14.542 | 279.831 | 0.00    | 0.00   | 100.000 | 2145.33 |
| 5900.0  | 5942.5 | 6049.089  | 14.772 | 282.502 | 0.00    | 0.00   | 100.000 | 2192.98 |
| 6000.0  | 5924.9 | 6201.572  | 15.012 | 285.130 | 0.00    | 0.00   | 100.000 | 2241.13 |
| 6100.0  | 5908.3 | 6356.070  | 15.264 | 287.715 | 0.00    | 0.00   | 100.000 | 2289.80 |
| 6200.0  | 5892.9 | 6512.708  | 15.527 | 290.259 | 0.00    | 0.00   | 100.000 | 2339.00 |
| 6300.0  | 5878.4 | 6671.612  | 15.802 | 292.763 | 1869.24 | -25.00 | 100.000 | 2388.75 |
| 6400.0  | 5864.7 | 6831.826  | 15.880 | 295.227 | 1472.30 | -25.00 | 100.000 | 2439.05 |
| 6500.0  | 5844.7 | 6992.376  | 15.964 | 297.653 | 1277.14 | -25.00 | 100.000 | 2489.92 |
| 6600.0  | 5830.4 | 7153.369  | 16.056 | 300.043 | 1131.78 | -25.00 | 100.000 | 2541.36 |
| 6700.0  | 5817.7 | 7314.919  | 16.154 | 302.398 | 1013.24 | -25.00 | 100.000 | 2593.43 |
| 6800.0  | 5806.6 | 7477.140  | 16.261 | 304.720 | 912.25  | -25.00 | 100.000 | 2646.09 |
| 6900.0  | 5797.0 | 7640.152  | 16.377 | 307.009 | 823.91  | -25.00 | 100.000 | 2699.37 |
| 7000.0  | 5789.0 | 7804.079  | 16.501 | 309.267 | 745.26  | -25.00 | 100.000 | 2753.29 |
| 7100.0  | 5782.7 | 7969.065  | 16.642 | 311.494 | 674.31  | -21.62 | 100.000 | 2807.86 |
| 7200.0  | 5779.4 | 8135.504  | 16.833 | 313.692 | 609.22  | -17.00 | 100.000 | 2863.11 |
| 7300.0  | 5778.7 | 8303.985  | 17.077 | 315.862 | 548.30  | -12.64 | 100.000 | 2919.04 |
| 7400.0  | 5780.0 | 8475.090  | 17.375 | 318.004 | 490.18  | -8.57  | 100.000 | 2975.67 |
| 7500.0  | 5782.9 | 8649.389  | 17.727 | 320.118 | 433.75  | -4.78  | 100.000 | 3033.03 |
| 7600.0  | 5787.0 | 8827.425  | 18.130 | 322.204 | 378.13  | -1.28  | 100.000 | 3091.13 |
| 7700.0  | 5792.0 | 9009.718  | 18.583 | 324.262 | 322.61  | 1.96   | 100.000 | 3149.98 |
| 7800.0  | 5797.7 | 9196.755  | 19.084 | 326.292 | 266.60  | 4.94   | 100.000 | 3209.62 |
| 7900.0  | 5803.9 | 9388.996  | 19.631 | 328.293 | 209.67  | 7.68   | 100.000 | 3270.06 |
| 8000.0  | 5810.3 | 9586.866  | 20.220 | 330.265 | 151.43  | 10.22  | 100.000 | 3331.32 |
| 8100.0  | 5816.8 | 9790.762  | 20.850 | 332.206 | 91.62   | 12.62  | 100.000 | 3393.42 |
| 8200.0  | 5823.5 | 10001.060 | 21.520 | 334.117 | 30.01   | 15.18  | 100.000 | 3456.40 |
| 8300.0  | 5797.9 | 10217.570 | 22.135 | 335.994 | .10     | 68.16  | .794    | 3486.87 |
| 8400.0  | 5743.5 | 10437.440 | 22.654 | 337.824 | .10     | 66.82  | .769    | 3487.37 |
| 8500.0  | 5689.3 | 10659.910 | 23.152 | 339.606 | .10     | 65.92  | .752    | 3487.86 |
| 8600.0  | 5635.5 | 10884.720 | 23.630 | 341.343 | .09     | 67.01  | .727    | 3488.33 |
| 8700.0  | 5582.1 | 11111.600 | 24.088 | 343.034 | .09     | 64.57  | .714    | 3488.79 |
| 8800.0  | 5529.2 | 11340.300 | 24.527 | 344.682 | .09     | 66.18  | .684    | 3489.24 |
| 8900.0  | 5476.7 | 11570.580 | 24.947 | 346.288 | .09     | 65.63  | .670    | 3489.67 |
| 9000.0  | 5424.8 | 11802.230 | 25.350 | 347.853 | .09     | 63.48  | .661    | 3490.09 |
| 9100.0  | 5373.5 | 12035.040 | 25.736 | 349.378 | .08     | 64.92  | .625    | 3490.50 |
| 9200.0  | 5322.7 | 12268.820 | 25.104 | 350.866 | .08     | 64.01  | .604    | 3490.90 |
| 9300.0  | 5272.6 | 12503.390 | 25.467 | 352.316 | .08     | 63.22  | .588    | 3491.28 |
| 9400.0  | 5223.1 | 12738.580 | 25.794 | 353.731 | .07     | 63.47  | .567    | 3491.65 |
| 9500.0  | 5174.3 | 12974.240 | 27.116 | 355.111 | .07     | 62.81  | .554    | 3492.01 |
| 9600.0  | 5126.1 | 13210.210 | 27.423 | 356.458 | .07     | 61.96  | .531    | 3492.35 |
| 9700.0  | 5078.6 | 13446.370 | 27.715 | 357.773 | .07     | 62.71  | .520    | 3492.69 |
| 9800.0  | 5031.8 | 13682.580 | 27.996 | 359.057 | .07     | 61.64  | .510    | 3493.02 |
| 9900.0  | 4985.6 | 13918.720 | 28.263 | 360.312 | .06     | 61.36  | .489    | 3493.33 |
| 10000.0 | 4940.1 | 14154.700 | 28.517 | 361.537 | .06     | 61.20  | .470    | 3493.64 |
| 10100.0 | 4895.3 | 14390.390 | 28.758 | 362.734 | .06     | 60.98  | .459    | 3493.93 |
| 10200.0 | 4851.2 | 14625.720 | 28.988 | 363.905 | .06     | 61.77  | .440    | 3494.22 |
| 10300.0 | 4807.8 | 14860.580 | 29.207 | 365.050 | .05     | 61.29  | .426    | 3494.50 |
| 10400.0 | 4765.0 | 15094.890 | 29.414 | 366.170 | .05     | 59.89  | .421    | 3494.76 |
| 10500.0 | 4722.9 | 15328.590 | 29.611 | 367.265 | .05     | 60.28  | .396    | 3495.03 |

|         |        |           |        |         |     |       |      |         |
|---------|--------|-----------|--------|---------|-----|-------|------|---------|
| 10600.0 | 4681.4 | 15561.600 | 29.798 | 368.337 | .05 | 59.75 | .382 | 3495.28 |
| 10700.0 | 4640.5 | 15793.840 | 29.974 | 369.387 | .05 | 60.73 | .363 | 3495.52 |
| 10800.0 | 4600.4 | 16025.270 | 30.141 | 370.415 | .05 | 60.16 | .365 | 3495.76 |
| 10900.0 | 4560.8 | 16255.820 | 30.298 | 371.422 | .05 | 60.15 | .350 | 3495.99 |
| 11000.0 | 4521.8 | 16485.440 | 30.447 | 372.409 | .05 | 59.40 | .348 | 3496.21 |
| 11100.0 | 4483.5 | 16714.080 | 30.586 | 373.376 | .04 | 58.29 | .337 | 3496.43 |
| 11200.0 | 4445.8 | 16941.700 | 30.718 | 374.325 | .04 | 59.99 | .316 | 3496.64 |
| 11300.0 | 4408.6 | 17168.260 | 30.840 | 375.255 | .04 | 59.41 | .312 | 3496.84 |
| 11400.0 | 4372.1 | 17393.710 | 30.955 | 376.167 | .04 | 60.04 | .302 | 3497.04 |
| 11500.0 | 4336.1 | 17618.020 | 31.062 | 377.062 | .04 | 60.33 | .289 | 3497.23 |
| 11600.0 | 4300.6 | 17841.160 | 31.162 | 377.941 | .04 | 58.66 | .283 | 3497.42 |
| 11700.0 | 4265.7 | 18063.090 | 31.254 | 378.804 | .04 | 58.22 | .287 | 3497.60 |
| 11800.0 | 4231.4 | 18283.790 | 31.340 | 379.651 | .03 | 59.71 | .262 | 3497.78 |
| 11900.0 | 4197.6 | 18503.240 | 31.418 | 380.483 | .03 | 58.54 | .268 | 3497.95 |
| 12000.0 | 4164.3 | 18721.400 | 31.489 | 381.301 | .03 | 57.89 | .262 | 3498.12 |
| 12100.0 | 4131.5 | 18938.260 | 31.554 | 382.104 | .03 | 60.02 | .241 | 3498.28 |
| 12200.0 | 4099.2 | 19153.800 | 31.613 | 382.894 | .03 | 57.05 | .249 | 3498.44 |
| 12300.0 | 4067.3 | 19368.000 | 31.666 | 383.671 | .03 | 59.10 | .233 | 3498.59 |
| 12400.0 | 4036.0 | 19580.830 | 31.712 | 384.435 | .03 | 58.90 | .230 | 3498.75 |
| 12500.0 | 4005.2 | 19792.300 | 31.753 | 385.187 | .03 | 57.72 | .222 | 3498.89 |
| 12600.0 | 3974.8 | 20002.370 | 31.787 | 385.927 | .03 | 58.27 | .216 | 3499.03 |
| 12700.0 | 3944.8 | 20211.050 | 31.817 | 386.655 | .03 | 58.16 | .218 | 3499.17 |
| 12800.0 | 3915.3 | 20418.310 | 31.840 | 387.372 | .03 | 59.06 | .201 | 3499.31 |
| 12900.0 | 3886.2 | 20624.160 | 31.859 | 388.077 | .03 | 59.60 | .201 | 3499.44 |
| 13000.0 | 3857.6 | 20828.560 | 31.872 | 388.772 | .02 | 60.05 | .193 | 3499.57 |
| 13100.0 | 3829.4 | 21031.530 | 31.880 | 389.457 | .03 | 57.31 | .199 | 3499.70 |
| 13200.0 | 3801.6 | 21233.060 | 31.883 | 390.132 | .02 | 58.02 | .182 | 3499.82 |
| 13300.0 | 3774.1 | 21433.120 | 31.882 | 390.797 | .02 | 58.69 | .184 | 3499.94 |
| 13400.0 | 3747.1 | 21631.730 | 31.875 | 391.452 | .02 | 57.86 | .178 | 3500.06 |
| 13500.0 | 3720.5 | 21828.870 | 31.864 | 392.099 | .02 | 56.26 | .184 | 3500.17 |
| 13600.0 | 3694.2 | 22024.540 | 31.848 | 392.736 | .02 | 57.28 | .178 | 3500.29 |
| 13700.0 | 3668.3 | 22218.740 | 31.828 | 393.365 | .02 | 58.86 | .168 | 3500.40 |
| 13800.0 | 3642.8 | 22411.460 | 31.804 | 393.985 | .02 | 58.58 | .159 | 3500.50 |
| 13900.0 | 3617.6 | 22602.700 | 31.775 | 394.597 | .02 | 59.89 | .157 | 3500.61 |
| 14000.0 | 3592.8 | 22792.450 | 31.742 | 395.201 | .02 | 58.59 | .163 | 3500.71 |
| 14100.0 | 3568.4 | 22980.730 | 31.705 | 395.798 | .02 | 57.79 | .159 | 3500.81 |
| 14200.0 | 3544.2 | 23167.510 | 31.663 | 396.386 | .02 | 57.88 | .157 | 3500.91 |
| 14300.0 | 3520.4 | 23352.820 | 31.618 | 396.968 | .02 | 60.13 | .149 | 3501.00 |
| 14400.0 | 3496.9 | 23536.640 | 31.569 | 397.542 | .02 | 58.05 | .147 | 3501.10 |
| 14500.0 | 3473.8 | 23718.960 | 31.516 | 398.109 | .02 | 58.81 | .138 | 3501.19 |
| 14600.0 | 3450.9 | 23899.810 | 31.459 | 398.670 | .02 | 60.69 | .134 | 3501.28 |
| 14700.0 | 3428.4 | 24079.160 | 31.398 | 399.223 | .02 | 58.81 | .138 | 3501.37 |
| 14800.0 | 3406.2 | 24257.040 | 31.334 | 399.771 | .02 | 60.03 | .132 | 3501.46 |
| 14900.0 | 3384.2 | 24433.420 | 31.266 | 400.311 | .02 | 58.14 | .136 | 3501.54 |
| 15000.0 | 3362.6 | 24608.330 | 31.195 | 400.846 | .02 | 57.62 | .134 | 3501.62 |
| 15100.0 | 3341.2 | 24781.750 | 31.120 | 401.375 | .02 | 57.46 | .134 | 3501.71 |
| 15200.0 | 3320.1 | 24953.690 | 31.041 | 401.898 | .02 | 58.69 | .128 | 3501.79 |
| 15300.0 | 3299.3 | 25124.160 | 30.959 | 402.416 | .01 | 61.39 | .115 | 3501.86 |
| 15400.0 | 3278.8 | 25293.150 | 30.874 | 402.928 | .02 | 58.77 | .119 | 3501.94 |
| 15500.0 | 3258.5 | 25460.670 | 30.786 | 403.434 | .01 | 62.06 | .109 | 3502.02 |
| 15600.0 | 3238.6 | 25626.720 | 30.694 | 403.935 | .02 | 58.08 | .117 | 3502.09 |
| 15700.0 | 3218.8 | 25791.300 | 30.599 | 404.431 | .02 | 57.53 | .124 | 3502.17 |
| 15800.0 | 3199.3 | 25954.410 | 30.501 | 404.922 | .01 | 59.36 | .115 | 3502.24 |
| 15900.0 | 3180.1 | 26116.060 | 30.399 | 405.408 | .01 | 61.10 | .105 | 3502.31 |
| 16000.0 | 3161.1 | 26276.250 | 30.295 | 405.889 | .01 | 61.48 | .103 | 3502.38 |
| 16100.0 | 3142.4 | 26434.990 | 30.187 | 406.366 | .01 | 60.02 | .103 | 3502.45 |
| 16200.0 | 3123.9 | 26592.270 | 30.076 | 406.838 | .01 | 60.27 | .103 | 3502.51 |
| 16300.0 | 3105.6 | 26748.100 | 29.963 | 407.306 | .01 | 59.06 | .111 | 3502.58 |
| 16400.0 | 3087.6 | 26902.480 | 29.846 | 407.769 | .01 | 61.65 | .096 | 3502.65 |
| 16500.0 | 3069.8 | 27055.420 | 29.726 | 408.228 | .01 | 59.96 | .098 | 3502.71 |
| 16600.0 | 3052.3 | 27206.920 | 29.604 | 408.682 | .01 | 61.77 | .096 | 3502.77 |
| 16700.0 | 3034.9 | 27356.980 | 29.478 | 409.133 | .01 | 62.25 | .094 | 3502.83 |
| 16800.0 | 3017.8 | 27505.610 | 29.350 | 409.580 | .01 | 61.50 | .094 | 3502.89 |
| 16900.0 | 3000.9 | 27652.810 | 29.219 | 410.023 | .01 | 59.67 | .100 | 3502.95 |
| 17000.0 | 2984.2 | 27798.580 | 29.085 | 410.462 | .01 | 61.85 | .092 | 3503.01 |
| 17100.0 | 2967.7 | 27942.930 | 28.948 | 410.897 | .01 | 59.53 | .096 | 3503.07 |
| 17200.0 | 2951.4 | 28085.860 | 28.808 | 411.329 | .01 | 58.01 | .096 | 3503.13 |
| 17300.0 | 2935.4 | 28227.380 | 28.666 | 411.757 | .01 | 61.04 | .090 | 3503.18 |
| 17400.0 | 2919.5 | 28367.480 | 28.521 | 412.182 | .01 | 63.81 | .073 | 3503.24 |
| 17500.0 | 2903.8 | 28506.160 | 28.373 | 412.603 | .01 | 60.63 | .082 | 3503.30 |

|         |        |           |        |         |     |       |      |         |
|---------|--------|-----------|--------|---------|-----|-------|------|---------|
| 17600.0 | 2888.4 | 28643.480 | 28.223 | 413.021 | .01 | 61.07 | .090 | 3503.35 |
| 17700.0 | 2873.1 | 28779.390 | 28.070 | 413.436 | .01 | 61.44 | .084 | 3503.40 |
| 17800.0 | 2858.0 | 28913.880 | 27.915 | 413.847 | .01 | 66.33 | .071 | 3503.45 |
| 17900.0 | 2843.1 | 29046.980 | 27.756 | 414.256 | .01 | 62.82 | .075 | 3503.50 |
| 18000.0 | 2828.4 | 29178.700 | 27.595 | 414.661 | .01 | 59.26 | .086 | 3503.55 |
| 18100.0 | 2813.9 | 29309.030 | 27.432 | 415.064 | .01 | 64.73 | .080 | 3503.60 |
| 18200.0 | 2799.6 | 29437.970 | 27.266 | 415.463 | .01 | 65.27 | .077 | 3503.65 |
| 18300.0 | 2785.4 | 29565.540 | 27.098 | 415.860 | .01 | 63.48 | .080 | 3503.70 |
| 18400.0 | 2771.5 | 29691.730 | 26.927 | 416.254 | .01 | 65.55 | .063 | 3503.75 |
| 18500.0 | 2757.7 | 29816.550 | 26.753 | 416.645 | .01 | 65.02 | .071 | 3503.80 |
| 18600.0 | 2744.1 | 29940.000 | 26.577 | 417.033 | .01 | 65.33 | .063 | 3503.85 |
| 18700.0 | 2730.6 | 30062.100 | 26.399 | 417.419 | .01 | 62.74 | .069 | 3503.89 |
| 18800.0 | 2717.4 | 30182.830 | 26.218 | 417.803 | .01 | 66.45 | .056 | 3503.94 |
| 18900.0 | 2704.3 | 30302.200 | 26.035 | 418.183 | .01 | 65.25 | .063 | 3503.98 |
| 19000.0 | 2691.4 | 30420.230 | 25.849 | 418.562 | .01 | 62.11 | .082 | 3504.03 |
| 19100.0 | 2678.6 | 30536.900 | 25.661 | 418.938 | .01 | 62.76 | .069 | 3504.07 |
| 19200.0 | 2666.0 | 30652.230 | 25.471 | 419.311 | .01 | 64.20 | .061 | 3504.12 |
| 19300.0 | 2653.6 | 30766.210 | 25.278 | 419.682 | .01 | 65.24 | .059 | 3504.16 |
| 19400.0 | 2641.4 | 30878.860 | 25.083 | 420.052 | .01 | 67.43 | .065 | 3504.20 |
| 19500.0 | 2629.3 | 30990.160 | 24.886 | 420.418 | .01 | 66.74 | .056 | 3504.24 |
| 19600.0 | 2617.3 | 31100.140 | 24.686 | 420.783 | .01 | 63.70 | .065 | 3504.28 |
| 19700.0 | 2605.6 | 31208.790 | 24.484 | 421.145 | .01 | 64.05 | .056 | 3504.33 |
| 19800.0 | 2594.0 | 31316.110 | 24.280 | 421.506 | .01 | 63.01 | .063 | 3504.36 |
| 19900.0 | 2582.5 | 31422.110 | 24.073 | 421.864 | .01 | 67.53 | .054 | 3504.41 |
| 20000.0 | 2571.2 | 31526.790 | 23.864 | 422.221 | .01 | 68.46 | .071 | 3504.44 |
| 20100.0 | 2560.1 | 31630.160 | 23.653 | 422.575 | .01 | 68.60 | .063 | 3504.48 |
| 20200.0 | 2549.1 | 31732.210 | 23.440 | 422.928 | .01 | 65.58 | .059 | 3504.52 |
| 20300.0 | 2538.2 | 31832.950 | 23.225 | 423.278 | .01 | 66.48 | .056 | 3504.56 |
| 20400.0 | 2527.5 | 31932.390 | 23.007 | 423.627 | .01 | 68.35 | .061 | 3504.60 |
| 20500.0 | 2517.0 | 32030.530 | 22.787 | 423.974 | .01 | 63.54 | .054 | 3504.64 |
| 20600.0 | 2506.6 | 32127.370 | 22.565 | 424.320 | .01 | 66.07 | .050 | 3504.67 |
| 20700.0 | 2496.4 | 32222.910 | 22.341 | 424.663 | .01 | 65.32 | .063 | 3504.71 |
| 20800.0 | 2486.3 | 32317.150 | 22.115 | 425.005 | .01 | 66.83 | .046 | 3504.74 |
| 20900.0 | 2476.3 | 32410.110 | 21.886 | 425.345 | .01 | 73.46 | .046 | 3504.78 |
| 21000.0 | 2466.5 | 32501.770 | 21.656 | 425.684 | .01 | 74.70 | .050 | 3504.82 |
| 21100.0 | 2456.8 | 32592.150 | 21.423 | 426.021 | .01 | 65.19 | .063 | 3504.85 |
| 21200.0 | 2447.3 | 32681.250 | 21.189 | 426.357 | .01 | 62.53 | .065 | 3504.88 |
| 21300.0 | 2438.0 | 32769.070 | 20.952 | 426.691 | .01 | 66.15 | .046 | 3504.92 |
| 21400.0 | 2428.7 | 32855.610 | 20.714 | 427.023 | .01 | 77.80 | .044 | 3504.95 |
| 21500.0 | 2419.6 | 32940.880 | 20.473 | 427.354 | .01 | 68.80 | .056 | 3504.99 |
| 21600.0 | 2410.7 | 33024.870 | 20.230 | 427.684 | .01 | 69.73 | .054 | 3505.02 |
| 21700.0 | 2401.9 | 33107.600 | 19.986 | 428.012 | .01 | 72.59 | .050 | 3505.05 |
| 21800.0 | 2393.2 | 33189.060 | 19.739 | 428.339 | .00 | 72.96 | .038 | 3505.09 |
| 21900.0 | 2384.7 | 33269.260 | 19.491 | 428.664 | .01 | 68.89 | .052 | 3505.12 |
| 22000.0 | 2376.3 | 33348.190 | 19.240 | 428.989 | .01 | 74.58 | .042 | 3505.15 |
| 22100.0 | 2368.0 | 33425.870 | 18.986 | 429.311 | .01 | 68.24 | .061 | 3505.18 |
| 22200.0 | 2359.9 | 33502.290 | 18.734 | 429.633 | .01 | 72.10 | .048 | 3505.21 |
| 22300.0 | 2351.9 | 33577.460 | 18.478 | 429.954 | .01 | 67.95 | .050 | 3505.24 |
| 22400.0 | 2344.1 | 33651.370 | 18.220 | 430.273 | .01 | 74.12 | .048 | 3505.28 |
| 22500.0 | 2336.4 | 33724.040 | 17.960 | 430.591 | .01 | 73.14 | .046 | 3505.31 |
| 22600.0 | 2328.8 | 33795.460 | 17.699 | 430.908 | .01 | 72.10 | .054 | 3505.34 |
| 22700.0 | 2321.4 | 33865.630 | 17.435 | 431.224 | .01 | 75.85 | .046 | 3505.37 |
| 22800.0 | 2314.1 | 33934.560 | 17.170 | 431.539 | .01 | 71.28 | .052 | 3505.40 |
| 22900.0 | 2306.9 | 34002.260 | 16.903 | 431.852 | .01 | 75.43 | .044 | 3505.43 |
| 23000.0 | 2299.9 | 34068.710 | 16.635 | 432.165 | .01 | 73.02 | .059 | 3505.46 |
| 23100.0 | 2292.9 | 34133.930 | 16.364 | 432.477 | .01 | 75.88 | .038 | 3505.49 |
| 23200.0 | 2286.2 | 34197.920 | 16.092 | 432.787 | .00 | 74.87 | .029 | 3505.51 |
| 23300.0 | 2279.5 | 34260.670 | 15.819 | 433.097 | .01 | 79.15 | .040 | 3505.54 |
| 23400.0 | 2273.0 | 34322.190 | 15.543 | 433.406 | .00 | 75.91 | .031 | 3505.57 |
| 23500.0 | 2266.6 | 34382.490 | 15.266 | 433.714 | .01 | 70.68 | .073 | 3505.60 |
| 23600.0 | 2260.4 | 34441.550 | 14.986 | 434.020 | .01 | 79.20 | .040 | 3505.63 |
| 23700.0 | 2254.3 | 34499.400 | 14.708 | 434.327 | .00 | 79.87 | .031 | 3505.66 |
| 23800.0 | 2248.3 | 34556.020 | 14.426 | 434.632 | .01 | 77.30 | .052 | 3505.68 |
| 23900.0 | 2242.4 | 34611.430 | 14.143 | 434.936 | .00 | 78.68 | .038 | 3505.71 |
| 24000.0 | 2236.7 | 34665.610 | 13.859 | 435.240 | .00 | 85.35 | .023 | 3505.74 |
| 24100.0 | 2231.1 | 34718.570 | 13.573 | 435.542 | .00 | 77.17 | .033 | 3505.76 |
| 24200.0 | 2225.6 | 34770.330 | 13.285 | 435.844 | .00 | 78.89 | .019 | 3505.78 |
| 24300.0 | 2220.2 | 34820.870 | 12.996 | 436.146 | .00 | 82.87 | .031 | 3505.81 |
| 24400.0 | 2215.0 | 34870.190 | 12.706 | 436.448 | .01 | 76.98 | .042 | 3505.84 |
| 24500.0 | 2209.9 | 34918.310 | 12.414 | 436.746 | .01 | 76.12 | .054 | 3505.86 |

|         |        |           |        |         |     |         |      |         |
|---------|--------|-----------|--------|---------|-----|---------|------|---------|
| 24600.0 | 2205.0 | 34965.200 | 12.121 | 437.045 | .01 | 74.70   | .042 | 3505.89 |
| 24700.0 | 2200.1 | 35010.900 | 11.827 | 437.344 | .00 | 81.53   | .038 | 3505.91 |
| 24800.0 | 2195.4 | 35055.390 | 11.531 | 437.641 | .01 | 75.93   | .069 | 3505.94 |
| 24900.0 | 2190.9 | 35098.670 | 11.234 | 437.938 | .01 | 74.24   | .042 | 3505.97 |
| 25000.0 | 2186.4 | 35140.750 | 10.936 | 438.235 | .01 | 75.98   | .046 | 3505.99 |
| 25100.0 | 2182.1 | 35181.630 | 10.637 | 438.531 | .01 | 79.14   | .069 | 3506.01 |
| 25200.0 | 2177.9 | 35221.310 | 10.336 | 438.826 | .01 | 79.08   | .040 | 3506.04 |
| 25300.0 | 2173.8 | 35259.790 | 10.035 | 439.121 | .00 | 81.17   | .025 | 3506.06 |
| 25400.0 | 2169.9 | 35297.070 | 9.732  | 439.415 | .00 | 83.57   | .033 | 3506.09 |
| 25500.0 | 2166.1 | 35333.150 | 9.428  | 439.709 | .01 | 81.22   | .050 | 3506.11 |
| 25600.0 | 2162.4 | 35368.030 | 9.124  | 440.003 | .01 | 78.72   | .107 | 3506.13 |
| 25700.0 | 2158.8 | 35401.720 | 8.818  | 440.295 | .00 | 90.00   | .004 | 3506.17 |
| 25800.0 | 2155.4 | 35434.220 | 8.511  | 440.588 | .01 | 80.02   | .054 | 3506.19 |
| 25900.0 | 2152.1 | 35465.520 | 8.203  | 440.880 | .01 | 80.89   | .071 | 3506.21 |
| 26000.0 | 2148.9 | 35495.630 | 7.895  | 441.171 | .00 | 99.03   | .013 | 3506.24 |
| 26100.0 | 2145.8 | 35524.550 | 7.585  | 441.462 | .00 | 158.55  | .002 | 3506.26 |
| 26200.0 | 2142.9 | 35552.280 | 7.275  | 441.753 | .00 | 79.44   | .021 | 3506.29 |
| 26300.0 | 2140.1 | 35578.820 | 6.963  | 442.043 | .00 | 75.21   | .008 | 3506.31 |
| 26400.0 | 2137.4 | 35604.170 | 6.651  | 442.333 | .00 | 163.07  | .006 | 3506.34 |
| 26500.0 | 2134.8 | 35628.330 | 6.338  | 442.622 | .01 | 84.08   | .073 | 3506.36 |
| 26600.0 | 2132.4 | 35651.310 | 6.025  | 442.912 | .01 | 84.68   | .061 | 3506.38 |
| 26700.0 | 2130.1 | 35673.100 | 5.711  | 443.201 | .00 | -103.12 | .008 | 3506.41 |
| 26800.0 | 2127.9 | 35693.700 | 5.396  | 443.489 | .01 | 81.54   | .077 | 3506.43 |
| 26900.0 | 2125.9 | 35713.120 | 5.080  | 443.778 | .00 | 86.22   | .029 | 3506.45 |
| 27000.0 | 2123.9 | 35731.360 | 4.764  | 444.066 | .00 | 81.05   | .025 | 3506.48 |
| 27100.0 | 2122.1 | 35748.400 | 4.448  | 444.354 | .01 | 85.08   | .088 | 3506.51 |
| 27200.0 | 2120.5 | 35764.270 | 4.131  | 444.641 | .00 | 87.14   | .038 | 3506.55 |
| 27300.0 | 2118.9 | 35778.950 | 3.813  | 444.929 | .00 | -98.54  | .013 | 3506.58 |
| 27400.0 | 2117.5 | 35792.450 | 3.495  | 445.216 | .00 | 90.02   | .029 | 3506.60 |
| 27500.0 | 2116.2 | 35804.770 | 3.177  | 445.503 | .02 | 85.26   | .159 | 3506.62 |
| 27600.0 | 2115.0 | 35815.900 | 2.858  | 445.790 | .00 | 138.93  | .002 | 3506.66 |
| 27700.0 | 2113.9 | 35825.860 | 2.539  | 446.077 | .00 | 90.00   | .004 | 3506.69 |
| 27800.0 | 2113.0 | 35834.640 | 2.219  | 446.363 | .00 | 86.02   | .027 | 3506.76 |
| 27900.0 | 2112.2 | 35842.230 | 1.900  | 446.650 | .02 | 87.50   | .130 | 3506.81 |
| 28000.0 | 2111.5 | 35848.640 | 1.580  | 446.936 | .00 | -99.60  | .010 | 3506.84 |
| 28100.0 | 2111.0 | 35853.870 | 1.260  | 447.223 | .00 | 90.00   | .010 | 3506.87 |
| 28200.0 | 2110.5 | 35857.930 | .940   | 447.509 | .02 | 88.25   | .186 | 3506.91 |
| 28300.0 | 2110.2 | 35860.800 | .621   | 447.795 | .01 | -90.96  | .113 | 3507.01 |
| 28400.0 | 2110.1 | 35862.490 | .299   | 448.081 | .00 | 90.00   | .010 | 3507.19 |

Alignment achieved, coasting now.

|         |        |           |       |         |      |      |       |         |
|---------|--------|-----------|-------|---------|------|------|-------|---------|
| 28500.0 | 2110.0 | 35863.000 | -.044 | 448.368 | 0.00 | 0.00 | 0.000 | 3508.69 |
|---------|--------|-----------|-------|---------|------|------|-------|---------|

Target achieved.

E.2 Earth Launch to LEO

Targeted Spaceflight Trajectory

Launch from Surface -- With Atmosphere

Thrust=1600000.0 N, C= 4500.0 m/s, MO=100000.0 kg, Mfuel= 90000.0 kg  
 Sref= 50.0 m^2, hstart= 10.0 km, Dt= .5 sec  
 AlphaMax= 10.0 deg, AlphDotMax= 2.0 deg/sec  
 TargetH= 372.0 km; TargetV= 7000.0 m/s

| t    | V     | h     | Phi    | Sigma | VG   | alpha | Thrust  | DeltaV |
|------|-------|-------|--------|-------|------|-------|---------|--------|
| 0.0  | 1.0   | 0.000 | 90.000 | 0.000 | 0.00 | 0.00  | 100.000 | 0.00   |
| 10.0 | 65.4  | 0.329 | 90.002 | 0.000 | 0.00 | 0.00  | 100.000 | 162.92 |
| 20.0 | 133.5 | 1.321 | 90.005 | 0.000 | 0.00 | 0.00  | 100.000 | 331.95 |
| 30.0 | 203.7 | 3.004 | 90.008 | 0.000 | 0.00 | 0.00  | 100.000 | 507.59 |
| 40.0 | 275.9 | 5.401 | 90.012 | 0.000 | 0.00 | 0.00  | 100.000 | 690.36 |
| 50.0 | 351.9 | 8.535 | 90.017 | 0.000 | 0.00 | 0.00  | 100.000 | 880.87 |

Starting targeting now.

|       |        |         |        |        |         |         |         |         |
|-------|--------|---------|--------|--------|---------|---------|---------|---------|
| 60.0  | 431.4  | 12.454  | 86.896 | 0.000  | 7391.50 | -10.00  | 100.000 | 1079.80 |
| 70.0  | 515.6  | 17.125  | 77.764 | 0.006  | 7294.80 | -10.00  | 100.000 | 1287.94 |
| 80.0  | 620.8  | 22.548  | 69.324 | 0.021  | 7173.83 | -10.00  | 100.000 | 1536.17 |
| 90.0  | 748.1  | 28.745  | 61.672 | 0.046  | 7026.56 | -10.00  | 100.000 | 1735.50 |
| 100.0 | 898.0  | 35.708  | 54.774 | 0.085  | 6852.74 | -10.00  | 100.000 | 1977.20 |
| 110.0 | 1070.7 | 43.396  | 48.543 | 0.140  | 6650.93 | -10.00  | 100.000 | 2232.57 |
| 120.0 | 1266.9 | 51.728  | 42.912 | 0.212  | 6421.24 | -10.00  | 100.000 | 2503.35 |
| 130.0 | 1487.6 | 60.611  | 37.824 | 0.306  | 6163.49 | -10.00  | 100.000 | 2791.45 |
| 140.0 | 1734.7 | 69.936  | 33.233 | 0.422  | 5877.10 | -10.00  | 100.000 | 3099.26 |
| 150.0 | 2010.4 | 79.588  | 29.096 | 0.564  | 5560.80 | -10.00  | 100.000 | 3429.69 |
| 160.0 | 2317.9 | 89.450  | 25.368 | 0.735  | 5212.49 | -10.00  | 100.000 | 3786.32 |
| 170.0 | 2661.2 | 99.410  | 22.006 | 0.936  | 4828.93 | -10.00  | 100.000 | 4173.66 |
| 180.0 | 3045.4 | 109.356 | 18.971 | 1.172  | 4405.45 | -10.00  | 100.000 | 4597.51 |
| 190.0 | 3477.7 | 119.175 | 16.226 | 1.447  | 3935.41 | -10.00  | 100.000 | 5065.47 |
| 200.0 | 3967.7 | 128.755 | 13.738 | 1.763  | 3409.44 | -10.00  | 100.000 | 5587.80 |
| 210.0 | 4529.2 | 137.981 | 11.474 | 2.127  | 2814.08 | -10.00  | 100.000 | 6178.82 |
| 220.0 | 5182.5 | 146.730 | 9.404  | 2.546  | 2129.41 | -10.00  | 100.000 | 6859.38 |
| 230.0 | 5960.5 | 154.892 | 7.606  | 3.029  | 1324.60 | -7.84   | 100.000 | 7661.51 |
| 240.0 | 6919.6 | 162.621 | 6.038  | 3.587  | 346.85  | -5.28   | 100.000 | 8638.33 |
| 250.0 | 7255.3 | 170.246 | 5.988  | 4.215  | 0.00    | -7.43   | 0.002   | 8985.21 |
| 260.0 | 7245.6 | 177.747 | 5.889  | 4.845  | 0.01    | 75.24   | 0.010   | 8985.30 |
| 270.0 | 7236.2 | 185.113 | 5.789  | 5.475  | 0.01    | 66.06   | 0.010   | 8985.40 |
| 280.0 | 7227.0 | 192.343 | 5.688  | 6.102  | 0.00    | -118.79 | 0.007   | 8985.47 |
| 290.0 | 7217.9 | 199.437 | 5.586  | 6.729  | 0.00    | 79.71   | 0.005   | 8985.56 |
| 300.0 | 7209.0 | 206.395 | 5.484  | 7.354  | 0.01    | 76.63   | 0.014   | 8985.62 |
| 310.0 | 7200.4 | 213.214 | 5.380  | 7.978  | 0.00    | 90.00   | 0.001   | 8985.70 |
| 320.0 | 7191.8 | 219.897 | 5.276  | 8.600  | 0.01    | -118.19 | 0.009   | 8985.80 |
| 330.0 | 7183.5 | 226.442 | 5.171  | 9.221  | 0.00    | 107.75  | 0.005   | 8985.92 |
| 340.0 | 7175.4 | 232.846 | 5.066  | 9.841  | 0.00    | -142.53 | 0.003   | 8986.02 |
| 350.0 | 7167.4 | 239.112 | 4.959  | 10.460 | 0.00    | 59.86   | 0.006   | 8986.11 |
| 360.0 | 7159.6 | 245.238 | 4.852  | 11.078 | 0.01    | 55.06   | 0.010   | 8986.22 |
| 370.0 | 7152.0 | 251.224 | 4.745  | 11.695 | 0.01    | 74.48   | 0.012   | 8986.30 |
| 380.0 | 7144.6 | 257.070 | 4.636  | 12.310 | 0.01    | 33.56   | 0.015   | 8986.38 |
| 390.0 | 7137.4 | 262.774 | 4.527  | 12.924 | 0.01    | -110.94 | 0.014   | 8986.47 |
| 400.0 | 7130.4 | 268.336 | 4.417  | 13.538 | 0.00    | 90.00   | 0.005   | 8986.58 |
| 410.0 | 7123.5 | 273.757 | 4.307  | 14.150 | 0.00    | -90.00  | 0.003   | 8986.68 |
| 420.0 | 7116.8 | 279.036 | 4.196  | 14.762 | 0.01    | 77.86   | 0.020   | 8986.77 |
| 430.0 | 7110.3 | 284.172 | 4.084  | 15.372 | 0.01    | -102.47 | 0.011   | 8986.88 |
| 440.0 | 7104.0 | 289.165 | 3.972  | 15.981 | 0.01    | 79.80   | 0.019   | 8987.00 |
| 450.0 | 7097.9 | 294.015 | 3.860  | 16.590 | 0.01    | 80.64   | 0.010   | 8987.10 |
| 460.0 | 7092.0 | 298.720 | 3.746  | 17.198 | 0.00    | -90.00  | 0.002   | 8987.20 |
| 470.0 | 7086.2 | 303.281 | 3.632  | 17.805 | 0.00    | 123.45  | 0.003   | 8987.34 |
| 480.0 | 7080.7 | 307.699 | 3.518  | 18.411 | 0.01    | -101.46 | 0.012   | 8987.49 |

|       |        |         |       |        |     |         |      |         |
|-------|--------|---------|-------|--------|-----|---------|------|---------|
| 490.0 | 7075.3 | 311.971 | 3.403 | 19.016 | .02 | 78.85   | .030 | 8987.63 |
| 500.0 | 7070.1 | 316.099 | 3.288 | 19.620 | .00 | -97.29  | .006 | 8987.75 |
| 510.0 | 7065.1 | 320.082 | 3.172 | 20.224 | .01 | -106.17 | .015 | 8987.83 |
| 520.0 | 7060.3 | 323.919 | 3.056 | 20.827 | .01 | 83.02   | .020 | 8987.92 |
| 530.0 | 7055.6 | 327.610 | 2.940 | 21.430 | .01 | 86.79   | .015 | 8988.07 |
| 540.0 | 7051.2 | 331.156 | 2.823 | 22.031 | .00 | -112.75 | .006 | 8988.20 |
| 550.0 | 7046.9 | 334.555 | 2.705 | 22.632 | .01 | 82.61   | .019 | 8988.35 |
| 560.0 | 7042.8 | 337.807 | 2.587 | 23.233 | .01 | -106.18 | .020 | 8988.51 |
| 570.0 | 7038.9 | 340.913 | 2.469 | 23.833 | .01 | 90.02   | .015 | 8988.70 |
| 580.0 | 7035.2 | 343.872 | 2.351 | 24.432 | .00 | 103.03  | .007 | 8988.89 |
| 590.0 | 7031.7 | 346.684 | 2.232 | 25.031 | .02 | 84.59   | .035 | 8989.06 |
| 600.0 | 7028.3 | 349.348 | 2.113 | 25.630 | .01 | -96.33  | .015 | 8989.31 |
| 610.0 | 7025.2 | 351.866 | 1.993 | 26.227 | .02 | -94.99  | .038 | 8989.53 |
| 620.0 | 7022.2 | 354.236 | 1.873 | 26.825 | .01 | -93.11  | .015 | 8989.81 |
| 630.0 | 7019.4 | 356.457 | 1.753 | 27.422 | .00 | 90.02   | .001 | 8990.08 |
| 640.0 | 7016.8 | 358.531 | 1.633 | 28.019 | .01 | -90.00  | .014 | 8990.33 |
| 650.0 | 7014.4 | 360.457 | 1.513 | 28.615 | .01 | 90.00   | .025 | 8990.65 |
| 660.0 | 7012.2 | 362.234 | 1.392 | 29.211 | .01 | 100.76  | .013 | 8990.90 |
| 670.0 | 7010.2 | 363.864 | 1.271 | 29.807 | .04 | -93.95  | .060 | 8991.29 |
| 680.0 | 7008.3 | 365.345 | 1.150 | 30.402 | .01 | 90.00   | .012 | 8991.55 |
| 690.0 | 7006.7 | 366.678 | 1.029 | 30.998 | .00 | -82.65  | .006 | 8991.84 |
| 700.0 | 7005.2 | 367.862 | .908  | 31.593 | .07 | -92.99  | .110 | 8992.21 |
| 710.0 | 7003.9 | 368.897 | .786  | 32.187 | .02 | -91.81  | .026 | 8992.78 |
| 720.0 | 7002.8 | 369.784 | .665  | 32.782 | .01 | 92.17   | .022 | 8993.31 |
| 730.0 | 7001.8 | 370.522 | .543  | 33.377 | .04 | -92.86  | .065 | 8993.82 |
| 740.0 | 7001.1 | 371.112 | .421  | 33.971 | .14 | 88.56   | .228 | 8994.91 |
| 750.0 | 7000.6 | 371.552 | .299  | 34.565 | .01 | -90.00  | .024 | 8996.13 |
| 760.0 | 7000.2 | 371.843 | .177  | 35.159 | .07 | 88.11   | .124 | 8998.51 |
| 770.0 | 7000.0 | 371.986 | .057  | 35.754 | .22 | 89.50   | .376 | 9001.44 |

Alignment achieved, coasting now.

|       |        |         |       |        |      |      |       |         |
|-------|--------|---------|-------|--------|------|------|-------|---------|
| 775.0 | 7000.0 | 372.002 | -.003 | 36.051 | 0.00 | 0.00 | 0.000 | 9004.19 |
|-------|--------|---------|-------|--------|------|------|-------|---------|

Target achieved.

APPENDIX F

Computer Program Listing

```

-----
----- TARGET.TEXT - April 9, 1984 -----
-----

```

```

PROGRAM Target;
($S)

```

```

USES ($U /CCUTIL/CCLIB) CCcIIO;
($U /SYSTEM/PASCLIB) Atmosphere, Extend;

```

```

LABEL 1, 2, 3;

```

```

CONST REarth=6.378165e6; (M)
      GEarth=9.80665; (M/S^2)
      Mu=3.986032e14; (M^3/S^2)
      RadToDeg=57.2957795131;
      printout='/PRINTER';
      outfile='/VDALE/TARGET.OUTPUT';

```

```

TYPE headrecord = RECORD
  TitleH : STRING[80];
  ThrustIH : REAL;
  MOH : REAL;
  MfuelH : REAL;
  CH : REAL;
  SrefH : REAL;
  AlphaDMaxH : REAL;
  AlphODMaxH : REAL;
  Oth : REAL;
  Vkh : REAL;
  HH : REAL;
  PHIDH : REAL;
  hstartH : REAL;
  HZH : REAL;
  VZOH : REAL;
END;

```

```

datarecord = RECORD
  time : REAL;
  mass : REAL;
  Vtotal : REAL;
  altitude : REAL;
  fitangle : REAL;
  Vradial : REAL;
  Vangular : REAL;
  rangeang : REAL;
  Vrequired : REAL;
  VTgained : REAL;
  thrustang : REAL;
  thrustivi : REAL;
  delta_V : REAL;
  message : STRING[75];
END;

```

```

String60 = STRING[60];

```

```

VAR   BVR, BOM, BR, BS, BP, BM:      ARRAY[0..4] OF REAL;
      Vr, Vom, Om, V, R, S, Phi, M:  ARRAY[1..4] OF REAL;

      h, Vk, Vnk, Vomk, Omk, Rk, Sk, Phk, Mk, MQ, Mfuel, PhiD, c, Dt, t,
      density, pressure, temp, g, vsound, Q, mult, Cd, Cl, Drag, Lift,
      Sref, Thrust, ThrustI, sigmaD, hstart, R0, R2, H2, VZ, VZ0, chord,
      VC, VP, VRR, VSig, VCR, VG, VReq, alpha, alphaD, oldAlpha, theta,
      beta, sigma2, gamma, AlphaMax, AlphaDMax, AlphDotMax, AlphDotDMax,
      DeltaV, DeltaVk, SMA, epsilon, DeltaSigT, oldSk, Amax, VZMax,
      ThrustPercent, Vstep, newval:  REAL;

      Option, N, count, interval:  INTEGER;

      FirstTarget, EndTarget, terminate, TargetReached, savedata:  BOOLEAN;

      ans:                               STRING[3];
      Title, newtitle:                   STRING[80];
      headname:                           STRING[20];
      printer:                             INTERACTIVE;
      DateString, TimeString:             CkStr40;
      headerset:                           headrecord;
      headerfile:                          FILE OF headrecord;
      dataset:                              datarecord;
      datafile:                             FILE OF datarecord;

FUNCTION BrkPress:  BOOLEAN;  EXTERNAL;

```

```

-----
{ GetInput: Interactively prompts user for run initialization data }
-----

```

PROCEDURE GetInput;

```

-----
{ GetInput.ReadHeader: Reads data from file for selected run }
{ initialization }
-----

```

PROCEDURE ReadHeader;

```

BEGIN
  RESET(headerfile, CONCAT(headname, '.HEADER'));
  READ(headerfile, headerset);
  Title      := headerset.TitleH;
  ThrustI    := headerset.ThrustIH;
  MO         := headerset.MOH;
  Mfuel      := headerset.MfuelH;
  C          := headerset.CH;
  Sref       := headerset.SrefH;
  AlphaDMax := headerset.AlphaDMaxH;
  AlphDotDMax := headerset.AlphDotDMaxH;
  Dt         := headerset.DtH;
  Vk         := headerset.VkH;
  n          := headerset.nH;
  PhiD       := headerset.PhiDH;
  hstart     := headerset.hstartH;
  HZ         := headerset.HZH;
  VZD        := headerset.VZDH;
  CLOSE(headerfile, LOCK);
END;
```

```

-----
{ GetInput.WriteHeader: Writes data to file for selected run }
{ initialization }
-----

```

PROCEDURE WriteHeader;

```

BEGIN
  REWRITE(headerfile, CONCAT(headname, '.HEADER'));
  WITH headerset DO BEGIN
    TitleH      := Title;
    ThrustIH    := ThrustI;
    MOH         := MO;
    MfuelH      := Mfuel;
    CH          := C;
    SrefH       := Sref;
    AlphaDMaxH := AlphaDMax;
    AlphDotDMaxH := AlphDotDMax;
    DtH         := Dt;
    VkH         := Vk;
    nH          := n;
    PhiDH       := PhiD;
    hstartH     := hstart;
    HZH         := HZ;
    VZDH        := VZD;
  END;
  WRITE(headerfile, headerset);
  CLOSE(headerfile, LOCK);
END;
```

```

-----
}
{ GetInput
}
-----

```

```

BEGIN (GetInput)

WRITELN;
WRITELN('Initialization options are:');
WRITELN;
WRITELN(' 1) Use values from an existing run');
WRITELN(' 2) Change values of an existing run');
WRITELN(' 3) Create a new run');
WRITELN;
WRITE('Enter option number: ');
READLN(option);

IF NOT (option IN [1..3]) THEN
  REPEAT
    WRITELN;
    WRITE('option 3 is not an option. Please choose 1, 2, or 3: ');
    READLN(option);
  UNTIL (option IN [1..3]);

CASE option OF

1: BEGIN (Use existing run)
  WRITELN;
  WRITE('Enter name of existing run: ');
  READLN(headname);
  ReadHeader;
END;

2: BEGIN (Change existing run)

  WRITELN;
  WRITE('Enter name of existing run: ');
  READLN(headname);
  ReadHeader;

  WRITELN;
  WRITELN('Title: ',Title);
  WRITELN;
  WRITE('New title (return for no change): ');
  READLN(newtitle);
  IF NOT(newtitle='') THEN Title:=newtitle;
  WRITELN;
  WRITELN('Enter values (negative for no change):');
  WRITELN;
  WRITE('Thrust= ',ThrustI:9:1,' N. New value=');
  READLN(newval);
  IF (newval)>=0 THEN ThrustI:=newval;
  WRITE('Initial mass, MO= ',MO:8:1,' kg. New value=');
  READLN(newval);
  IF (newval)>=0 THEN MO:=newval;
  WRITE('Fuel mass, Mfuel= ',Mfuel:8:1,' kg. New value=');
  READLN(newval);
  IF (newval)>=0 THEN Mfuel:=newval;
  WRITE('Exhaust velocity, C= ',c:7:1,' m/s. New value=');
  READLN(newval);
  IF (newval)>=0 THEN c:=newval;
  WRITE('Reference area, Sref= ',Sref:5:1,' m^2. New value=');
  READLN(newval);
  IF (newval)>=0 THEN Sref:=newval;
  WRITE('Max alpha, AlphaMax= ',AlphaDMax:6:1,' deg. New value=');
  READLN(newval);
  IF (newval)>=0 THEN AlphaDMax:=newval;
  WRITE('Max alphadot, AlphDotMax= ',AlphDotDMax:6:1,
    ' deg/sec. New value=');

```

```

READLN(newval);
IF (newval)>=0 THEN AlphaDotDMax:=newval;
WRITE('Time increment, Dt=',Dt:6:1,' secs. New value=');
READLN(newval);
IF (newval)>=0 THEN Dt:=newval;

WRITELN;
WRITELN('Enter initial conditions:');
WRITELN;
WRITE('Velocity, V=',Vk:7:1,' m/s. New value=');
READLN(newval);
IF (newval)>=0 THEN Vk:=newval;
WRITE('Altitude, h=',h/1000:7:1,' km. New value=');
READLN(newval);
IF (newval)>=0 THEN h:=newval*1000;
WRITE('Flight angle, Phi=',Phi:5:1,' deg. New value=');
READLN(newval);
IF (newval)>=0 THEN PhiD:=newval;

WRITELN;
WRITELN('Enter target values:');
WRITELN;
WRITE('Altitude at which to start targeting, hstart=',hstart/1000:6:1,
      ' km. New value=');
READLN(newval);
IF (newval)>=0 THEN hstart:=newval*1000;
WRITE('Target altitude, H2=',H2/1000:8:1,' km. New value=');
READLN(newval);
IF (newval)>=0 THEN H2:=newval*1000;
R0:=h+REarth;
R2:=H2+REarth;
Amax:=(R0+R2)/2;
VZMax:=SQRT(Mu*(2/R2-1/Amax));
WRITELN('Target velocity (max feasible is ',SQRT(Mu/R2):9:3,' m/s)');
WRITELN(' (Hohmann transfer velocity is ',VZMax:9:3,' m/s)');
WRITE(' VZ=',VZ0:9:3,' m/s. New value=');
READLN(newval);
IF (newval)>=0 THEN VZ0:=newval;

WRITELN;
WRITE('Run name under which to save this data: ');
READLN(headname);
WriteHeader;

END;

3: BEGIN (Create new run)

WRITELN;
WRITE('Title: ');
READLN(Title);
WRITELN;
WRITELN('Enter values (in Metric units):');
WRITELN;
WRITE('Thrust (N), =');
READLN(ThrustI);
WRITE('Initial mass (kg), M0=');
READLN(M0);
WRITE('Fuel mass (kg), Mfuel=');
READLN(Mfuel);
WRITE('Exhaust velocity (m/s), C=');
READLN(c);
WRITE('Reference area (m^2), Sref=');
READLN(Sref);
WRITE('Max alpha (deg), AlphaMax=');
READLN(AlphaMax);
WRITE('Max alphasdot (deg/sec), AlphaDotMax=');
READLN(AlphaDotMax);

```

```

WRITE('Time increment (secs), Dt=');
READLN(Dt);

WRITELN;
WRITELN('Enter initial conditions:');
WRITELN;
WRITE('Velocity (m/s), V=');
READLN(VK);
WRITE('Altitude (km), h=');
READLN(h);
h:=h*1000;
WRITE('Flight angle (deg), Phi=');
READLN(Phi0);

WRITELN;
WRITELN('Enter target values:');
WRITELN;
WRITE('Altitude at which to start targeting (km), hstart=');
READLN(hstart);
hstart:=hstart*1000;
WRITE('Target altitude (km), H2=');
READLN(H2);
H2:=H2*1000;
R0:=h+REarth;
R2:=H2+REarth;
Amax:=(R0+R2)/2;
VZMax:=SQRT(Mu*(2/R2-1/Amax));
WRITELN('Target velocity (max feasible is ',SQRT(Mu/R2):9:3,' m/s)');
WRITE (' (Hohmann transfer velocity is ',VZMax:9:3,' m/s), VZ=');
READLN(VZ0);

WRITELN;
WRITE('Name of headerfile to which this data is to be saved: ');
READLN(headname);
WriteHeader;

END;

END; {CASE}

WRITELN;
WRITE('Save the output for this run? ');
READLN(ans);
IF (ans[1]='Y') OR (ans[1]='y') THEN savedata:=TRUE
ELSE savedata:=FALSE;
WRITELN;
WRITE('Data output interval: ');
READLN(interval);

END; {GetInput}

```

```
{-----}  
{ Putout: Writes out data to screen and disk file at each time step }  
{-----}
```

```
PROCEDURE Putout;
```

```
  BEGIN
```

```
    WRITE(t:7:1);  
    WRITE(Vk:9:1);  
    WRITE(h/1000:11:3);  
    WRITE(Phi0:8:3);  
    WRITE(Sigma0:9:3);  
    WRITE(VG:9:2);  
    WRITE(Alpha0:9:2);  
    WRITE(ThrustPercent:9:3);  
    WRITELN(DeltaV:9:2);
```

```
  WITH dataset DO BEGIN
```

```
    time:=t;  
    mass:=Mk;  
    Vtotal:=Vk;  
    altitude:=h;  
    fitangle:=Phi0;  
    Vradial:=Vrk;  
    Vangular:=Vomk;  
    Vrequired:=VReq;  
    VTBgained:=VG;  
    rangeang:=Sigma0;  
    thrustang:=Alpha0;  
    thrust[V]:=ThrustPercent;  
    delta_V:=DeltaV;  
    message:='';
```

```
  END;
```

```
  IF savedata THEN WRITE(datafile, dataset);
```

```
END; (Putout)
```

```

-----]
[ Setup:  Initializes internal variables and writes run initialization
[          information to screen
-----]

```

```
PROCEDURE Setup;
```

```
BEGIN
```

```

t:=0;
Lift:=0;
Drag:=0;
Vreq:=0;
VG:=0;
DeltaV:=0;
oldSk:=0;
Sk:=0;
Mk:=MO;
Rk:=h+REarth;
Phk:=PhiD/RadToDeg;
Vrk:=Vk*SIN(Phk);
Vomk:=Vk*COS(Phk);
Omek:=Vk*COS(Phk)/Rk;
ThrustPercent:=100;
Thrust:=ThrustI;
RZ:=H2+REarth;
BVR[0]:=0;
BOM[0]:=0;
BR[0]:=0;
BS[0]:=0;
BM[0]:=0;
alpha:=0;
alphaD:=0;
sigmaD:=0;
AlphaMax:=AlphaDMax/RadToDeg;
AlphDotMax:=AlphDotDMax/RadToDeg;
FirstTarget:=TRUE;
EndTarget:=FALSE;
terminate:=FALSE;
TargetReached:=FALSE;

REWRITE(printer, printout);
REWRITE(datafile, outfile);

WRITELN;
WRITELN;
WRITELN(Title);
WRITELN;
WRITELN(
'Thrust=',ThrustI:9:1,' N,      C=',c:7:1,' m/s,      MO=',MO:8:1,' kg',
'Mfuel=',Mfuel:8:1,' kg/');
WRITELN('Sref=',Sref:5:1,' m^2,      hstart=',hstart/1000:6:1,' km',
'Qt=',Qt:6:1,' sec');
WRITELN(
'AlphaMax=',AlphaDMax:6:1,' deg, AlphDotMax=',AlphDotDMax:6:1,
' deg/sec');

WRITELN('TargetH=',H2/1000:8:1,' km, TargetV=',V20:7:1,' m/s');
WRITELN;
WRITELN(
'  t      V      h      Phi      Sigma      VG      alpha
'  Thrust  DeltaV');

WRITELN;

Putout;

END; (Setup)

```

```

-----
[ DoTarget:   Guidance algorithm to find target trajectory and control
               variable values to steer vehicle onto target trajectory
-----

```

```
PROCEDURE DoTarget;
```

```

-----
[ DoTarget.FindThrust:  Finds thrust from rocket equation that will keep
               Vx<VReq at each time step
-----

```

```
PROCEDURE FindThrust;
```

```

BEGIN
  Vstep:=0.99*VG;
  Thrust:=(C*Mk/dt)*(1-EXP(-Vstep/c));
  IF Thrust > ThrustI THEN Thrust:=ThrustI;
  ThrustPercent:=Thrust/ThrustI*100;
END; {FindThrust}

```

```

-----
[ DoTarget.Cutoff:   Ends targetting procedure and resets variables for
               coasting flight when closeness to target would cause
               a numerical error in ARCCOS or ARCSIN
-----

```

```
PROCEDURE Cutoff(reason: String80);
```

```

BEGIN
  Thrust:=0;
  alpha:=0;
  alpha0:=0;
  VReq:=0;
  VG:=0;
  ThrustPercent:=0;
  EndTarget:=TRUE;
  WRITELN;
  WRITELN('Alignment achieved, coasting now.');
```

```

  WRITELN(' ',reason);
  WRITELN;
  dataset.message:='Alignment achieved, coasting now.';
  WRITE(datafile, dataset);
  EXIT (DoTarget);
END; {Cutoff}

```

```

-----
[ DoTarget.OneMinusCOS:  Solves 1-COS(X) accurately for small X
-----

```

```
FUNCTION OneMinusCOS (X: REAL): REAL;
```

```

BEGIN
  OneMinusCOS:=2*SQR(SIN(X/2));
END;

```

```

-----
( DoTarget )
-----

```

```

BEGIN (DoTarget)

-IF FirstTarget THEN BEGIN
  WRITELN;
  WRITELN('Starting targeting now. ');
  WRITELN;
  dataset.message:='Starting targeting now.';
  WRITE(datafile, dataset);
  FirstTarget:=FALSE;
END;

IF Rk>=R2 THEN Cutoff('Rk >= R2');
Amax:=(Rk+R2)/Z;
V2Max:=SQRT(Mu*(Z/R2-1/Amax));

IF V20 <= 0.9999*V2Max THEN BEGIN

  V2:=V20;
  SMA:=(R2*Mu)/(Z*Mu-R2*SQR(V2));
  IF ((Z*SMA*R2-SMA*Rk-SQR(R2))/(SMA*Rk-Rk*R2) >= 1) THEN Cutoff
    ('sigma = ARCCOS (arg>=1)');
  sigma2:=ARCCOS((Z*SMA*R2-SMA*Rk-SQR(R2))/(SMA*Rk-Rk*R2));
  epsilon:=R2-Rk;
  chord:=SQRT(Z*(SQR(Rk)+Rk*epsilon)*(OneMinusCOS(sigma2))+SQR(epsilon));
  gamma:=(Pi/Z)-ARCSIN((Rk/chord)*SIN(sigma2));

  VC:=V2/COS(gamma);
  VP:=V2*TAN(gamma);
  VSig:=VC*COS(ABS(sigma2-gamma));
  VCR:=VC*SIN(ABS(sigma2-gamma));

  IF gamma>=sigma2 THEN BEGIN
    VRR:=VP;
    theta:=ARCTAN((VP+VCR)/VSig);
    VReq:=SQRT(SQR(VP+VCR)+SQR(VSig));
  END
  ELSE IF gamma<sigma2 THEN BEGIN
    VRR:=VP-VCR;
    theta:=ARCTAN(VRR/VSig);
    VReq:=SQRT(SQR(VRR)+SQR(VSig));
  END;

  epsilon:=VReq-Vk;
  VG:=SQRT(Z*(SQR(Vk)+Vk*epsilon)*OneMinusCOS(ABS(Pk-theta))
    +SQR(epsilon));

  IF (VG=0) THEN BEGIN
    Thrust:=0;
    alpha:=0;
    alphaD:=0;
    VG:=0;
    ThrustPercent:=0;
    EXIT (DoTarget);
  END
  ELSE FindThrust;

  IF ((VReq/VG)*SIN(ABS(Pk-theta)) >= 1) THEN beta:=Pi/2
  ELSE beta:=ARCSIN((VReq/VG)*SIN(ABS(Pk-theta)));
  IF (SQRT(SQR(VG)+SQR(Vk))<VReq) THEN beta:=Pi-beta;

  didalpha:=alpha;
  IF theta>Pk THEN alpha:=Pi-beta;
  IF theta<Pk THEN alpha:=beta-Pi;

END ELSE alpha:=0;

```

```
IF (ThrustPercent>1.0) AND (-VG>1.0) AND (Vk<0.995*VReq) THEN BEGIN
  IF (ABS(alpha-oldAlpha)/Dt > AlphDotMax) THEN
    alpha:=SIGN(alpha-oldAlpha)*(AlphDotMax*Dt)+oldalpha;
  IF (ABS(alpha) > AlphaMax) THEN
    alpha:=SIGN(alpha)*AlphaMax;
END;

alphaD:=alpha*RadToDeg;

IF BrkPress THEN BEGIN Terminate:=TRUE; GOTO 2; END;

END: (DoTarget)
```

```

-----
[ Integrate: Performs Fourth-Order Runge Kutta integration on inertial
[ frame equations of motion
-----

```

```
PROCEDURE Integrate;
```

```
VAR I: INTEGER;
```

```

-----
[ Integrate.FindDrag: Finds aerodynamic forces on vehicle using 1976
[ standard atmosphere model
-----

```

```
PROCEDURE FindDrag;
```

```
BEGIN
```

```
FindAtmosphere (density, pressure, temp, g, vsound, h);
```

```
Cl:=1*alpha;
```

```
Cd:=0.1+2*Cl*Cl; [Cd0 = 0.1]
```

```
Q:=0.5*density*SQR(Vk);
```

```
Drag:=Q*Cd*Sref;
```

```
Lift:=Q*Cl*Sref;
```

```
END; [FindDrag]
```

```

-----
[ Integrate.FindB: Finds intermediate state variable increments
-----

```

```
PROCEDURE FindB;
```

```

-----
[ Integrate.FindB.Impact: Prevents flight beneath the earth's surface
-----

```

```
PROCEDURE Impact;
($S Cases)
```

```
BEGIN
```

```
WRITELN;
```

```
WRITELN('Impact on Earth at V=4,VCN3:7:1);
```

```
WRITELN('End of run.');
```

```
dataset.message:='Impact on Earth. End of Run.');
```

```
GOTO 3;
```

```
END; [Impact]
```

```

-----
( Integrate.FindB.FuelGone: Allows choice of coasting or ending
( trajectory if fuel burns out
-----

```

```

PROCEDURE FuelGone;
($S Cases)

```

```

BEGIN

```

```

WRITELN;
WRITE('Fuel has burned out, do you want to coast? ');
READLN(ans);

```

```

IF (ans[1]='Y') OR (ans[1]='y') THEN BEGIN

```

```

Thrust:=0;
alpha:=0;
alphaD:=0;
VReq:=0;
VG:=0;
EndTarget:=TRUE;
WRITELN('Coasting now. ');
WRITELN;
dataset.message:='Fuel gone, coasting now.';
WRITE(datafile, dataset);
GOTO 1;
END

```

```

ELSE BEGIN

```

```

WRITELN('Ending trajectory. ');
dataset.message:='Fuel gone, ending trajectory.';
GOTO 2;
END;

```

```

END; (FuelGone)

```

```

-----
( Integrate.FindB
-----

```

```

($S)
BEGIN (FindB)

```

```

IF R[N]<REarth THEN Impact;
IF M[N]<MO-Mfuel THEN FuelGone;

```

```

BVR[N]:=Dt*((Thrust*SIN(alpha+Phi[N])+Lift*COS(Phi[N])
-Drag*SIN(Phi[N]))/M[N]-g+R[N]*SQR(Om[N]));

```

```

BOM[N]:=Dt*((Thrust*COS(alpha+Phi[N])-Lift*SIN(Phi[N])
-Drag*COS(Phi[N]))/(M[N]*R[N])-2*Vr[N]*Om[N]/R[N]);

```

```

BR[N]:=Dt*Vr[N];

```

```

BS[N]:=Dt*Om[N];

```

```

BP[N]:=Dt*((Lift+Thrust*SIN(alpha))/M[N]-g*COS(Phi[N]))/Vr[N]+Om[N];

```

```

BM[N]:=Dt*(-Thrust/c);

```

```

END; (FindB)

```

```

[-----]
[ Integrate ]
[-----]

```

```

BEGIN (Integrate)
IF (Sref>0) THEN FindDrag ELSE g:=GEarth*SQR(REarth/Rk);

FOR I:=1 TO 4 DO BEGIN
  IF (I=2) OR (I=3) THEN mult:=0.5
  ELSE mult:=1;

  VR[I]:=Vrk+BVR[I-1]*mult;
  OM[I]:=Omk+BOM[I-1]*mult;
  R[I]:=Rk+BR[I-1]*mult;
  S[I]:=Sk+BS[I-1]*mult;
  Phi[I]:=Phk+BP[I-1]*mult;
  M[I]:=Mk+BM[I-1]*mult;
  V[I]:=SQR(SQR(VR[I])+SQR(R[I]*OM[I]));

  N:=I;

  FindB;

END;

Vrk:=Vrk+(BVR[1]+2*BVR[2]+2*BVR[3]+BVR[4])/6;
Omk:=Omk+(BOM[1]+2*BOM[2]+2*BOM[3]+BOM[4])/6;
Rk:=Rk+(BR[1]+2*BR[2]+2*BR[3]+BR[4])/6;
Sk:=Sk+(BS[1]+2*BS[2]+2*BS[3]+BS[4])/6;
Phk:=Phk+(BP[1]+2*BP[2]+2*BP[3]+BP[4])/6;
Mk:=Mk+(BM[1]+2*BM[2]+2*BM[3]+BM[4])/6;

Vk:=SQR(SQR(Vrk)+SQR(Rk*Omk));

IF (h>=hstart) AND (NOT EndTarget) THEN BEGIN
  DeltaSigT:=Sk-OldSk;
  DoTarget;
END;

t:=t+Dt;
OldSk:=Sk;
h:=Rk-REarth;
Vomk:=Rk*Omk;
PhiD:=Phk*RadToDeg;
SigmaD:=Sk*RadToDeg;
DeltaV:=-c*LN(Mk/M0);

END; (Integrate)

```

```

-----
[ PrintData: Sends all output to printer if user requests data printout ]
-----

```

```

PROCEDURE PrintData;
($S Cases)

```

```

BEGIN

  RESET(datafile, outfile);
  ClkDate2(DateString);
  ClkTime2(TimeString);
  WRITELN(printer,
'Targeted Spaceflight Trajectory
',
TimeString, ', ', DateString);

  WRITELN(printer);
  WRITELN(printer);
  WRITELN(printer, Title);
  WRITELN(printer);
  WRITELN(printer);
  Thrust=' ', ThrustI:9:1, ' N, C=' ', c:7:1, ' m/s, MO=' ', MO:8:1, ' kg, ',
  ' Mfuel=' ', Mfuel:8:1, ' kg';
  WRITELN(printer);
  Sref=' ', Sref:5:1, ' m^2, hstart=' ', hstart/1000:6:1, ' km, ',
  ' Ot=' ', Ot:6:1, ' sec';
  WRITELN(printer);
  AlphaMax=' ', AlphaOMax:5:1, ' deg, AlphDotMax=' ', AlphDotOMax:5:1,
  ' deg/sec');

  WRITELN(printer);
  TargetH=' ', H2/1000:8:1, ' km, TargetV=' ', V20:7:1, ' m/s');
  WRITELN(printer);
  WRITELN(printer);
  ' t V h Phi Sigma VG alpha ',
  ' Thrust DeltaV');

  WRITELN(printer);

  WHILE NOT EOF(datafile) DO BEGIN

    READ(datafile, dataset);

    WITH dataset DO BEGIN

      IF message = '' THEN BEGIN
        WRITE(printer, time:7:1);
        WRITE(printer, Vtotal:9:1);
        WRITE(printer, altitude/1000:11:3);
        WRITE(printer, fitangle:8:3);
        WRITE(printer, rangeang:9:3);
        WRITE(printer, VTBgained:9:2);
        WRITE(printer, thrustang:9:2);
        WRITE(printer, thrustivi:9:3);
        WRITELN(printer, delta_v:9:2);
      END
      ELSE BEGIN
        WRITELN(printer);
        WRITELN(printer, message);
        IF (message='Trajectory terminated by user.')
        OR (message='Target achieved.')
        OR (message='Fuel gone, ending trajectory.') THEN BEGIN
          WRITELN(printer, 'Delta-V=' ', delta_V:9:2, ' m/s');
          WRITELN(printer, 'Final Mass=' ', mass:8:1, ' kg');
        END;
        WRITELN(printer);
      END;
    END;
  END;
END;

```

```
IF BrkPress THEN BEGIN
  CLOSE (datafile, LOCK);
  EXIT (PrintData);
END;
END; (WHILE)
CLOSE(datafile, LOCK);
END; (PrintData)
```

```

-----
[ Target ]
-----

```

```

(##)
BEGIN {Target}

  WRITELN;
  WRITELN('Targeted Spaceflight Trajectory Analysis');

  IF (Sref>0) THEN InitAtmosphere;
  GetInput;
  Setup;
  count:=0;

  1: REPEAT

    Integrate;

    count:=count+1;
    IF (count=interval) THEN BEGIN Putout; count:=0; END;
    IF BrkFpress THEN terminate:=TRUE;

    IF ((h>=H2) AND ((Phk<0) OR (h>H2*1.01)))
      OR ((Phk<0) AND (H2-h < H2*1.0e-3)) THEN TargetReached:=TRUE;

  UNTIL terminate OR TargetReached;

  2: IF terminate THEN BEGIN
    IF (count>0) THEN putout;
    WRITELN;
    WRITELN('Trajectory terminated by user. ');
    dataset.message:='Trajectory terminated by user. ';
    END
  ELSE IF TargetReached THEN BEGIN
    IF (count>0) THEN putout;
    WRITELN;
    WRITELN('Target achieved. ');
    dataset.message:='Target achieved. ';
    END;

  WRITELN;
  WRITELN('DeltaV= ',DeltaV:9:2,' m/s');
  WRITELN;
  WRITELN('Final Mass=',Mk:8:1,' kg', ' Mass Ratio=',Mk/M0:7:5);

  3: WRITE(datafile, dataset);

  IF savedata THEN BEGIN
    CLOSE(datafile, LOCK);
    WRITELN;
    WRITE('Print results? (Y/N) ');
    READLN(ans);
    IF (ans[1]='Y') OR (ans[1]='y') THEN PrintData;
  END
  ELSE CLOSE(datafile, PURGE);
END.

```

UNIT Atmosphere;

{1976 Standard Atmosphere}

INTERFACE

CONST Ge=9.80665;  
Re=6.356756e6;  
Rain=287.0;  
SLtemp=288.15;

VAR ht: REAL;  
ltemp, l: ARRAY[1..8] OF REAL;  
a, b: ARRAY[1..14] OF REAL;  
upper: ARRAY[0..15] OF REAL;  
level: -1..15;

PROCEDURE InitAtmosphere;

PROCEDURE FindAtmosphere (VAR density, pressure, temp, g, vsound: REAL;  
h: REAL);

IMPLEMENTATION

PROCEDURE InitAtmosphere;

BEGIN

upper[0]:=0;  
upper[1]:=11; a[1]:=1.2250; b[1]:=-0.1101; ltemp[1]:=288.15; l[1]:=-6.5;  
upper[2]:=20; a[2]:=2.0484; b[2]:=-0.1569; ltemp[2]:=216.65; l[2]:=0.0;  
upper[3]:=32; a[3]:=2.0435; b[3]:=-0.1567; ltemp[3]:=216.65; l[3]:=1.0;  
upper[4]:=47; a[4]:=1.4919; b[4]:=-0.1469; ltemp[4]:=228.49; l[4]:=2.8;  
upper[5]:=51; a[5]:=0.5382; b[5]:=-0.1252; ltemp[5]:=269.68; l[5]:=0.0;  
upper[6]:=71; a[6]:=0.5803; b[6]:=-0.1267; ltemp[6]:=270.65; l[6]:=-2.8;  
upper[7]:=86; a[7]:=4.5685; b[7]:=-0.1558; ltemp[7]:=216.65; l[7]:=-2.0;  
upper[8]:=91; a[8]:=30.459; b[8]:=-0.1778; ltemp[8]:=186.95; l[8]:=0.0;  
upper[9]:=110; a[9]:=31.134; b[9]:=-0.1781;  
upper[10]:=120; a[10]:=0.2875; b[10]:=-0.1355;  
upper[11]:=150; a[11]:=5.3104e-4; b[11]:=-0.0830;  
upper[12]:=300; a[12]:=2.249e-7; b[12]:=-3.1236e-2;  
upper[13]:=600; a[13]:=4.2681e-9; b[13]:=-1.802e-2;  
upper[14]:=1000; a[14]:=7.6343e-11; b[14]:=-9.973e-3;  
upper[15]:=1e30;  
level:=1;

END; {InitAtmosphere}

```

PROCEDURE FindAtmosphere;
BEGIN
  ht:=h/1000;

  REPEAT
    IF ht>=upper[level] THEN level:=level+1
    ELSE IF ht<upper[level-1] THEN level:=level-1
  UNTIL (upper[level-1]<=ht) AND (ht<upper[level]);

  CASE level OF
    1, 2, 3, 4, 5, 6, 7, 8:
      BEGIN
        temp:=ttemp[level]+t[level]*(ht-upper[level-1]);
        density:=a[level]*EXP(b[level]*ht);
        pressure:=density*Rain*temp;
      END;
    9:
      BEGIN
        temp:=263.1905-76.3231*SQRT(1-SQR((ht-71)/(-19.92429)));
        density:=a[level]*EXP(b[level]*ht);
        pressure:=density*Rain*temp;
      END;
    10:
      BEGIN
        temp:=240.0+12.0*(ht-110);
        density:=a[level]*EXP(b[level]*ht);
        pressure:=density*Rain*temp;
      END;
    11, 12, 13, 14:
      BEGIN
        temp:=1000.0-640.0*EXP(-0.01875*(ht-120));
        density:=a[level]*EXP(b[level]*ht);
        pressure:=density*Rain*temp;
      END;
    15:
      BEGIN
        density:=0.0;
        pressure:=0.0;
      END;
  END; (CASE)

  vsound:=331.475*SQRT(temp/SLtemp);
  g:=Ge*SQRT(Re)/(SQRT(ht+Re));
END; (FindAtmosphere)
END.

```

## REFERENCES

- (1) McHenry, R.L., Brand, T.J., et al, "Space Shuttle Ascent, Guidance, Navigation, and Control", Journal of the Astronautical Sciences, Vol. XXVII, No. 1, pp. 1-38, January-March, 1979
- (2) Battin, R.H., An Introduction to the Mathematics and Methods of Astrodynamics, in publication, 1984
- (3) Battin, R.H., Astronautical Guidance, McGraw-Hill, 1964
- (4) Stuhlinger, E., Ion Propulsion for Space Flight, McGraw-Hill, 1964
- (5) U.S. Standard Atmosphere, 1976, U.S. Government Printing Office, Washington, D.C., October 1976