

General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

N69-25228

(ACCESSION NUMBER) 127	(THRU)
(PAGES) CR 98430	(CODE) 30
(NASA CR OR TX OR AD NUMBER)	(CATEGORY)

SD 69-3

**PROGRAM FOR OPTIMIZATION OF
TWO-IMPULSE TRANSFERS
BY CONTOURING AND STEEP DESCENT**

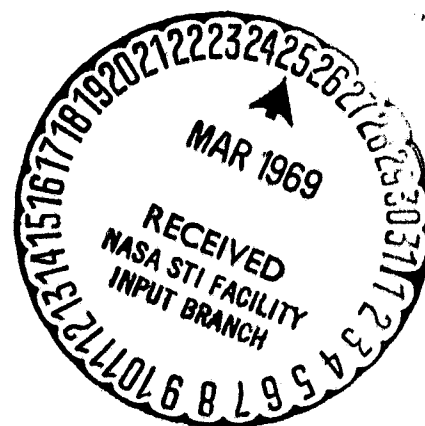
3 January 1969

Prepared by

**G. A. McCue
D. F. Bender
J. M. Morford**

Approved by


**C. R. Faulders, Manager
Flight Technology
Science & Technology**



**SPACE DIVISION
NORTH AMERICAN ROCKWELL CORPORATION**

SD 69-3

PROGRAM FOR OPTIMIZATION OF
TWO-IMPULSE TRANSFERS
BY CONTOURING AND STEEP DESCENT

3 January 1969

Prepared by

G. A. McCue
D. F. Bender
J. M. Morford

Approved by



C. R. Faulders, Manager
Flight Technology
Science & Technology

SPACE DIVISION
NORTH AMERICAN ROCKWELL CORPORATION

PRECEDING PAGE BLANK NOT FILMED.

FOREWORD

This work was principally performed under R&DA Nos. 2141 and 6552. Additional support was derived from the National Aeronautics and Space Administration's Marshall Space Flight Center under Contract Nos. NAS8-5211, NAS8-1582, NAS8-20238, and NAS8-21077.

ABSTRACT

Mathematical formulations of the general optimum two-impulse orbital transfer problem lead to expressions which, except for special cases, are analytically intractable. Numerical techniques therefore were developed and used to study optimum transfer modes—many orbit pairs yield four useful relative optima. A parameter space was adopted to represent the impulse function; that is, impulse was defined as a function of take-off point, arrival point, and transfer orbit semi-latus rectum. This abstraction allows visualization and detailed examination of surfaces of constant impulse—thus revealing the structure of impulse functions. Digital computer function contouring methods are utilized to accomplish this initial optimization and function exposure. This entire procedure requires about 10 seconds of IBM 360 computer time.

Once the optima are located by contouring they may be further studied by an "adaptive steep-descent" procedure. The program will automatically investigate the suspected optima revealed by contouring by conducting a numerical search for the exact optimum. With this descent technique, an optimum is found by taking trial and error search steps in a direction approximating the negative gradient of the impulse function. The procedure

SPACE DIVISION OF NORTH AMERICAN ROCKWELL CORPORATION

is an advance over older methods in that the step size is adjusted automatically to handle scaling and accuracy problems that are encountered when detailed analyses of optima are desired.

In this report, the program and its subroutines have been documented and a sample problem presented. This function contouring and adaptive steep descent program is written in double precision FORTRAN H. Depending upon the complexity of the optimum regions, the program will contour the entire function, produce detailed maps of all suspected optima and converge to the minimum impulsive transfer in less than a minute of IBM 360 time. The program's reliability and versatility have been demonstrated repeatedly during studies of finite thrust orbital transfer. If the user desires he may input only the central body name, the unit system desired and the orbital elements. With only this minimal data the program will carry out the complete optimization procedure without additional input parameter specifications. It therefore can be successfully used by relatively untrained analysts.

SPACE DIVISION of NORTH AMERICAN ROCKWELL CORPORATION

CONTENTS

Section		Page
I	INTRODUCTION	1
II	TWO-IMPULSE ORBITAL TRANSFER FORMULATION	4
III	IMPULSE FUNCTION INVESTIGATION	9
IV	p-OPTIMIZATION	12
V	EXACT OPTIMIZATION BY STEEP DESCENT	16
VI	CONCLUSION	30
VII	REFERENCES	31
	APPENDIXES	
	I. COMPUTER PROGRAM DESCRIPTION	34
	II. EXAMPLE OF THE USE OF THE ORBITAL TRANSFER OPTIMIZATION PROGRAM	47
	III. COMPUTER PROGRAM LISTINGS	69

ILLUSTRATIONS

Figure		Page
1	Orbital Transfer Geometry	5
2	Schematic View of Impulse Function Space and Impulse Contours	10
3	Optimum Impulse Contours for an Inclined Asymmetric Orbit Pair (Contour Interval = 500 ft/sec)	15
4	Descent Paths Plotted on Optimum-Impulse Contour Map.	25
5	Convergence of Adaptive Descent and Stepwise Steepest Descent Methods	26
6	Path Followed During Adaptive Descent Optimization of a Function Having Long Narrow "Valleys".	29
7	Modular Tree Structure of Two-Impulse Orbital Transfer Optimization Program	37
8	Data Input Format and Data Description.	40
9-27	Computer Output of a Sample Optimization Problem	50
28	MAIN Program Listing	70
29	Subroutine DATSET Listing	75
30	Subroutine INPUT Listing	77
31	Subroutine DEX Listing	81
32	Subroutine SURVEY Listing	83
33	Subroutine STEEP Listing	87
34	Subroutine IMP2 Listing	93
35	Subroutine ELIPSE Listing	100

SPACE DIVISION of NORTH AMERICAN ROCKWELL CORPORATION

Figure		Page
36	Subroutine CØNTUR Listing	105
37	Subroutine CØNGRD Listing	107
38	Subroutine CØNSET Listing	109
39	Subroutine TRISEL Listing	110
40	Subroutine TRICØN Listing	112
41	Subroutine PLØT3 Listing	114
42	Function DBLDØT Listing	115
43	Subroutine DBLCRS Listing	115
44	Function DBLUNT Listing	116
45	Subroutine DBLVEC Listing	116
46	Function DØT Listing	117
47	Function UNIT Listing	117

NOMENCLATURE

Scalars

a	Semimajor axis
α	Magnitude of step size
e	Eccentricity (magnitude of $\underline{e}$)
i	Inclination
p	Semilatus rectum
r	Radius to satellite
s_1, s_2, s_3	Scaling parameters
$\Delta\theta$	Transfer angle (true anomaly difference in transfer orbit plane)
μ	Gravitation constant
ϕ_1	Angle from reference axis to departure position in initial orbit
ϕ_2	Angle from reference axis to arrival position in terminal orbit
ω	Argument of perigee, angle from reference axis to perigee point

Vectors

$\underline{e}$	Orbit shape and orientation vector
$\underline{g}$	Unit vector in gradient direction
$\underline{I}$	Impulse vector
$\underline{N}$	Unit vector denoting reference direction (line of intersection of initial and final orbit planes)

SPACE DIVISION of NORTH AMERICAN ROCKWELL CORPORATION

Vectors

<u>r</u>	Geocentric satellite position vector
<u>U₁</u>	Unit vector directed toward point of departure from initial orbit
<u>U₂</u>	Unit vector directed toward point of arrival in final orbit
<u>V</u>	Velocity vector
<u>W</u>	Unit vector directed along orbit's angular momentum vector

Subscripts

1	Initial orbit
2	Final orbit
t	Transfer orbit
t1	Transfer orbit departure point
t2	Transfer orbit arrival point

SPACE DIVISION OF NORTH AMERICAN ROCKWELL CORPORATION

I. INTRODUCTION

Success of a variety of space missions depends ultimately upon an ability to maneuver in orbit. The high fuel requirements usually associated with orbit-changing maneuvers make it essential that optimum orbital transfer modes be investigated. While these modes might not be employed in the actual mission, they do represent lower bounds against which vehicle design compromises may be measured. Furthermore, the optimum two-impulse transfer has proven to be a very good approximation to its finite thrust counterpart. It will therefore suffice for most design studies.

This paper concerns a program for computing the optimum two-impulse orbital transfer between any pair of unperturbed elliptical orbits. The objective of the orbital transfer maneuver is to transform the five orbital elements, p , e , ω , i , Ω of an initial orbit to match corresponding elements of a final orbit. This is accomplished by instantaneous velocity changes (impulses) which may be applied at arbitrary points on the initial and final orbits. Impulse optimization is accomplished for free end points; i. e., impulse is minimized with respect to take-off and arrival points, as well as transfer orbit geometry.

Little analytic information is known about the optimum orbital transfer when end points are unspecified. Like many problems of celestial mechanics, the formulation is straightforward and, except for special cases, the equations

SPACE DIVISION of NORTH AMERICAN ROCKWELL CORPORATION

are analytically intractable. Thus, for the general problems, one either accepts simplifications in the mathematical model or he must utilize numerical techniques.

This report describes a program in which numerical methods are employed to allow visualization of an "impulse function space"—a geometrical abstraction of the impulse function. This technique allows concise graphical presentation of all optimum impulse information concerning any pair of elliptical orbits. Prior work associated with the development of the contouring program which is described herein may be found in Refs. 1-10.

It was found that a computer contouring technique that produced a visual "map" of the function was a powerful tool for examining the nature and structure of the entire impulse function. This technique made it possible to identify all possible regions of a given function which represent optimum transfer conditions.

Function contouring proves adequate for locating minima and for providing insight, but at times it does not yield the numerical accuracy that would be required for actual orbital transfer. For example, for some interesting orbit pairs the impulse functions have long, narrow "valleys" containing several minima. In such cases it is difficult to determine the optimum transfer circumstances with a precision adequate for use in engineering design studies.

Several steep descent processes (numerical techniques) were used in an attempt to accurately and quickly compute the optimum conditions.

SPACE DIVISION OF NORTH AMERICAN ROCKWELL CORPORATION

Experience with these processes demonstrated, however, that a more powerful "adaptive descent" technique was needed. Such a technique has now been included as part of this computer program. This numerical optimization method has been applied successfully to the minimization of numerous orbital transfer functions, and it is particularly effective for optimizing functions which have long, narrow structural features.

The program described here combines much prior work in a form that allows maximum flexibility in the solution of two-impulse transfer problems. Depending upon the complexity of the optimum regions the program will contour the entire function, produce detailed maps of all suspected optima and converge to the minimum impulsive transfer in less than a minute of IBM 360 time. The program's reliability and versatility have been demonstrated repeatedly during studies of finite thrust orbital transfer. If the user desires he may input only the orbital elements and the program will carry out the complete optimization procedure without additional input parameters. As a further aid to the user, the program tabulates numerous parameters concerning the optimum transfers. Experience has shown that the given output parameter set is usually desired for further studies.

II. TWO-IMPULSE ORBITAL TRANSFER FORMULATION

Referring to Fig. 1, consider a two-impulse transfer process between an initial orbit with elements p_1, e_1, ω_1, i and a final orbit defined by p_2, e_2, ω_2 . The formulation assumes Keplerian orbits and results from choosing the final orbit as the reference plane. That is, i is the relative inclination of the two orbit planes ($\cos i = \underline{W}_1 \cdot \underline{W}_2$, where $\underline{W}_1$ and $\underline{W}_2$ are unit vectors directed along the angular momentum vectors of the initial and final orbits). For coplanar orbits, the line of nodes or reference direction ($\underline{N}$) is arbitrary, but for inclined orbits $\underline{N}$ is defined as the line of intersection of the two orbit planes ($\underline{N} = \underline{W}_2 \times \underline{W}_1 / |\underline{W}_2 \times \underline{W}_1|$).

For the general case, there is a three-parameter family of transfer orbits joining any two specific orbits. The angles from the reference line to departure point ($\emptyset_1$) and from the reference line to the arrival point ($\emptyset_2$) are a natural choice for two of the three independent variables, since they, along with the given orbital elements, specify position and velocity in the known orbits (Fig. 1). The semilatus rectum (p_t) of the transfer orbit was the third parameter used for this study. It was chosen since it simplified the structure of the impulse function, $I(\emptyset_1, \emptyset_2, p_t)$. It also avoided several undesirable discontinuities that are present in other formulations. (A discussion of the use of alternate variables may be found in Refs. 3 and 11.) A

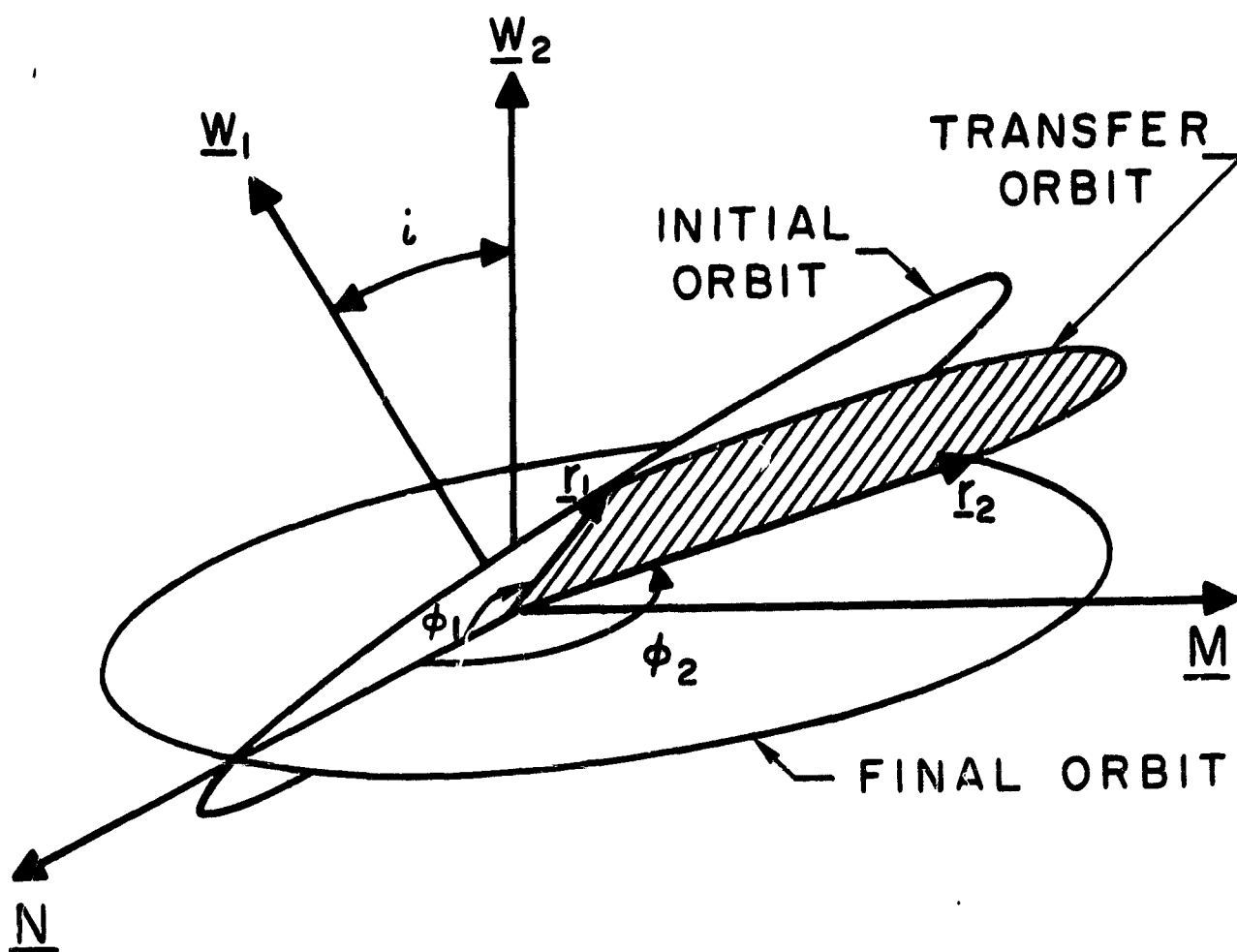


Figure 1. Orbital Transfer Geometry

discussion of the properties of the impulse function considered here appears in Ref. 1. Additional properties of this function were obtained by Lee.⁽⁹⁾

TRANSFER GEOMETRY

Geocentric unit vectors ($\underline{U}_1$ and $\underline{U}_2$) and radius vectors ($\underline{r}_1$ and $\underline{r}_2$) directed toward the departure and arrival points may be computed from θ_1 , θ_2 and the elements of the initial and final orbits:*

$$\underline{U}_1 = [\cos \theta_1, \sin \theta_1 \cos i, \sin \theta_1 \sin i] \quad (1)$$

$$\underline{U}_2 = [\cos \theta_2, \sin \theta_2, 0] \quad (2)$$

$$\underline{r}_j = \left[\frac{p_j}{1 + e_j \cos (\theta_j - \omega_j)} \right] \underline{U}_j \quad j = 1, 2 \quad (3)$$

Unit vectors normal to the three orbit planes are defined as follows:

$$\underline{W}_1 = [0, -\sin i, \cos i] \quad (4)$$

$$\underline{W}_2 = [0, 0, 1] \quad (5)$$

$$\underline{W}_t = \underline{U}_1 \times \underline{U}_2 / |\underline{U}_1 \times \underline{U}_2| \quad \underline{U}_1 \times \underline{U}_2 \neq 0 \quad (6)$$

Two vectors that define the shape and orientation of the initial and final orbits complete the transfer geometry:

$$\underline{e}_j = e_j [\cos \omega_j, \sin \omega_j \cos i_j, \sin \omega_j \sin i_j] \quad j = 1, 2 \quad (7)$$

* The subscripts 1, 2, and t denote initial, final and transfer orbits.

SPACE DIVISION OF NORTH AMERICAN ROCKWELL CORPORATION

The true anomaly interval traversed in the transfer orbit ($\Delta\theta$) may be determined directly:

$$\cos \Delta\theta = (\underline{U}_1 \cdot \underline{U}_2) \quad 0^\circ < \Delta\theta < 180^\circ \quad (8)$$

No generality is lost if the true anomaly interval is limited to the first two quadrants. Although this does restrict the problem to "short transfers", the "long transfers" ($180^\circ < \Delta\theta < 360^\circ$) may be computed by changing the signs of the velocity vectors in the transfer orbit. If $\sin \Delta\theta = 0$, the plane of the transfer is no longer uniquely determined and an alternate formulation is required to avoid this singularity.⁽¹²⁾

IMPULSE COMPUTATION

The function to be minimized is the total impulse for the two-impulse maneuver:

$$I = |\underline{I}_1| + |\underline{I}_2| \quad (9)$$

where

$$\underline{I}_1 = \pm \underline{V}_{t1} - \underline{V}_1 \quad (10)$$

$$\underline{I}_2 = \underline{V}_2 \mp \underline{V}_{t2} \quad (11)$$

(When a double sign is used, the upper sign refers to a "short transfer").

Velocity vectors in the initial and final orbits at the departure and arrival points ($\underline{V}_1$ and $\underline{V}_2$) and the corresponding velocity vectors in the transfer orbit ($\underline{V}_{t1}$ and $\underline{V}_{t2}$) are computed as follows:

$$\underline{V}_1 = (\mu/p_1)^{1/2} \underline{W}_1 \times (\underline{e}_1 + \underline{U}_1) \quad (12)$$

SPACE DIVISION of NORTH AMERICAN ROCKWELL CORPORATION

$$\underline{V}_2 = (\mu/p_2)^{1/2} \underline{W}_2 \times (\underline{e}_2 + \underline{U}_2) \quad (13)$$

$$\underline{V}_{t1} = (\mu/p_t)^{1/2} \underline{W}_t \times (\underline{e}_t + \underline{U}_1)^* \quad (14)$$

$$\underline{V}_{t2} = (\mu/p_t)^{1/2} \underline{W}_t \times (\underline{e}_t + \underline{U}_2)^* \quad (15)$$

The final impulse equations are obtained from Eqs. 10-17 by substituting Eq. 6 and performing several algebraic manipulations:

$$\underline{I}_1 = \pm [\underline{v} + z\underline{U}_1] - \underline{V}_1 \quad (16)$$

$$\underline{I}_2 = \underline{V}_2 \mp [\underline{v} - z\underline{U}_2] \quad (17)$$

where,

$$\underline{v} = \left[(\mu p_t)^{1/2} (\underline{r}_2 - \underline{r}_1) \right] / \left| \underline{r}_1 \times \underline{r}_2 \right| \quad (18)$$

$$z = (\mu/p_t)^{1/2} \tan (\Delta\theta/2) \quad (19)$$

Impulses corresponding to "long" and "short" transfers are compared by changing the double sign in Eqs. 16 and 17, and the combination producing the lesser impulse is used for the remaining computations.

* Note that e_t is an intermediate variable similar to e_1 and e_2 (as in Eq. 7), but no formula is given since it later drops out of the derivations.

III. IMPULSE FUNCTION INVESTIGATION

While the formulation of the expressions for impulse computation is relatively straightforward, the exact mathematical expressions are not susceptible to analytical solution. One must therefore resort to numerical techniques for optimizing the solution vector $I(\theta_1, \theta_2, p_t)$.

At best, conventional numerical search techniques only provide information concerning a function's local properties.⁽¹³⁾ Yet a detailed understanding of the impulse function's structure is essential for interpretation of numerical results. The number, shape, and relative importance of minima are requisites to a complete analysis of transfer problems since numerical optimization methods (e.g., steepest descent) find only that minimum impulse which is closest to the point of search initiation.

A geometric (" θ -space") representation of the impulse function was adopted. In Figure 2, p_t , θ_1 , and θ_2 are measured along the principal axes of an orthogonal coordinate system. All possible transfers between a pair of orbits (excluding the $\Delta\theta = 0^\circ$ and $\Delta\theta = 180^\circ$ function singularities) lie within the volume defined by $0^\circ \leq \theta < 360^\circ$, $0^\circ \leq \theta_2 < 360^\circ$, $0 < p_t < +\infty$. Of course, θ_1 and θ_2 undergo cyclical repetition outside this volume.

Studying the impulse function in θ -space required visualization of surfaces which are the loci of points having the same impulse. These "impulse surfaces" are generally closed and surround optima in much the same manner

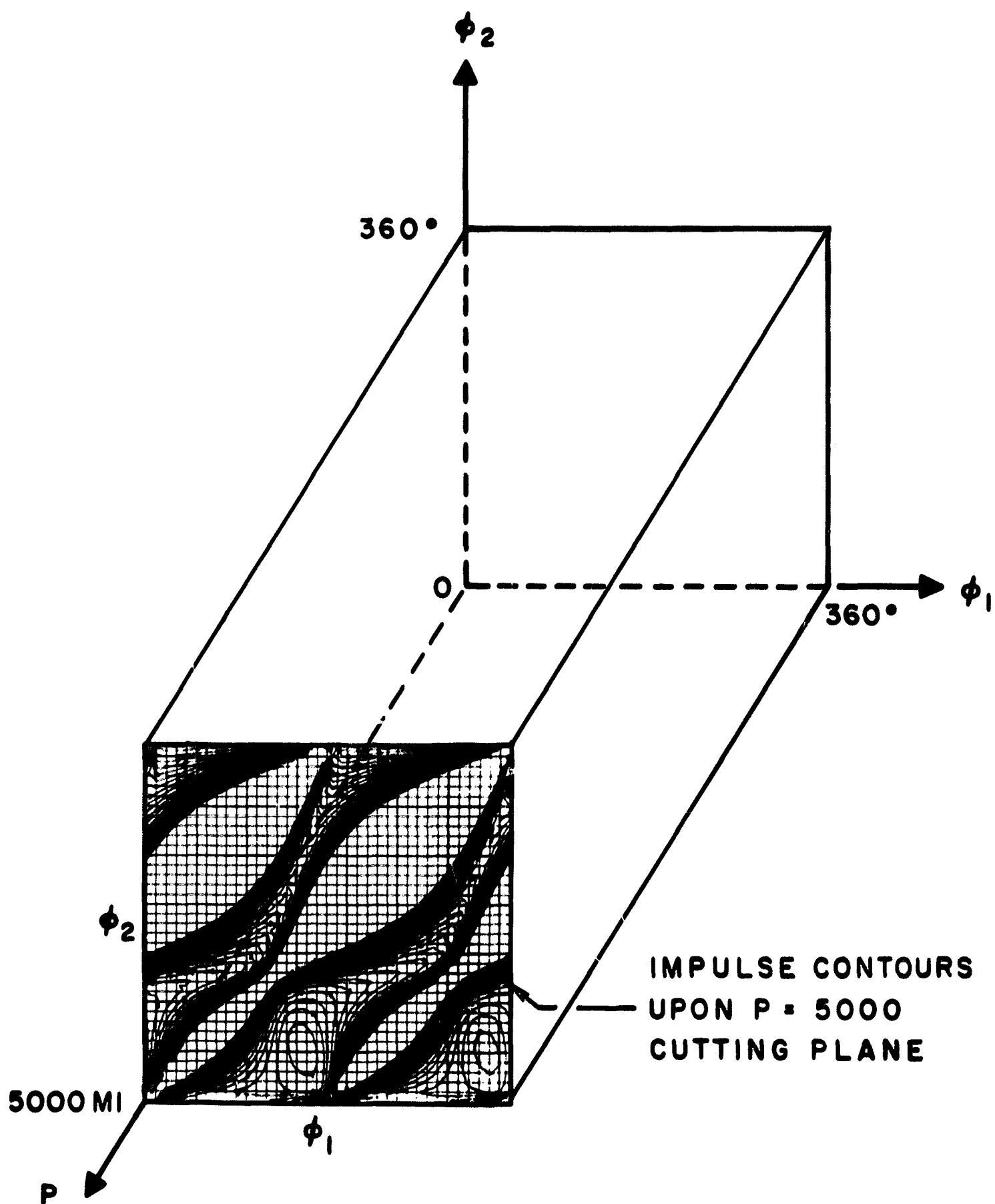


Figure 2. Schematic View of Impulse Function Space and Impulse Contours

that the successive layers of an onion enclose its center. Impulse surfaces may be visualized and studied by considering their traces upon a cutting plane (Fig. 2). The resulting contours of equivalent impulse that appear in most of the figures presented here have been found to provide a wealth of data in a concise easily understood form.

These contour maps are produced from an array of impulse values which are parametrically generated from Equation 9 with ϕ_1 , ϕ_2 , or p_t being held constant. Constant impulse contour lines, consisting of short straight-line segments, are then fitted between the survey points. It is important to note that the fidelity of the contouring technique depends directly upon the number of survey points. This fact should be remembered when examining certain structural details of the contour maps produced by the computer program.

Several seconds of IBM 360 time are required to contour a typical array of survey data (500-1,000 points). An SC-4020 CRT is then employed to plot the contour maps. A description of the contouring technique appears in Refs. 5 and 7. These contouring routines form an important package which is part of this complete computer program.

IV. p-OPTIMIZATION

The technique of passing a number of cutting planes through a given $\emptyset$ -space suffices to isolate optima and other pertinent features. However, as a $\emptyset$ -space becomes more complex, this technique requires that numerous contour maps be generated. A "p-optimization" technique was developed to overcome the above difficulties and thus present one contour map containing only optimum impulse information. Given $\emptyset_1$ and $\emptyset_2$ (i. e., $\underline{r}_1$ and $\underline{r}_2$), it is desired to find that p_t which minimizes the impulse defined by Equation 9.

The differentiation of Equation 9 yields

$$dI = \frac{\underline{I}_1 \cdot d\underline{I}_1}{|\underline{I}_1|} + \frac{\underline{I}_2 \cdot d\underline{I}_2}{|\underline{I}_2|} \quad (20)$$

Since $\underline{V}_1$ and $\underline{V}_2$ are independent of p_t

$$\frac{\partial \underline{I}_1}{\partial p_t} = \pm \frac{\partial \underline{V}_{t1}}{\partial p_t} \quad (21)$$

$$\frac{\partial \underline{I}_2}{\partial p_t} = \mp \frac{\partial \underline{V}_{t2}}{\partial p_t} \quad (22)$$

This leads to an expression for the derivative of impulse with respect to p_t which may be set equal to zero.

$$\frac{\partial I}{\partial p_t} = \pm \frac{1}{2p_t} \left[\frac{\underline{I}_1 \cdot (\underline{v} - z \underline{U}_1)}{|\underline{I}_1|} - \frac{\underline{I}_2 \cdot (\underline{v} + z \underline{U}_2)}{|\underline{I}_2|} \right] = 0 \quad (23)$$

Equation 23 is used to direct a numerical search that seeks the value of p_t that minimizes impulse for the given $\emptyset_1$ and $\emptyset_2$. Though Equation 23

has several roots, usually only those roots which correspond to elliptical transfer orbits are useful for impulse minimization. For certain elliptical orbit pairs, it can be shown that optimum two-impulse transfers between particular groups of end points result in hyperbolic transfer orbits.⁽⁹⁾ The range of θ_1 which produces an elliptical transfer orbit is bounded. These boundaries (parabolic orbit limits) are defined as follows:

$$P_{\min} = \frac{r_1 r_2 - \underline{r}_1 \cdot \underline{r}_2}{r_1 + r_2 + \left[2(r_1 r_2 + \underline{r}_1 \cdot \underline{r}_2) \right]^{1/2}} \quad (24)$$

$$P_{\max} = \frac{r_1 r_2 - \underline{r}_1 \cdot \underline{r}_2}{r_1 + r_2 - \left[2(r_1 r_2 + \underline{r}_1 \cdot \underline{r}_2) \right]^{1/2}} \quad (25)$$

Except for certain special cases, only one minimum impulse will occur between these limits.⁽⁹⁾ An effective numerical technique has been devised to seek the required optimum impulse solutions where $P_{\min} < P_t < P_{\max}$.

The contour maps presented throughout the remainder of this paper were generated using this technique. Approximately 500 p-optimizations were required for the generation of each contour map, a process which requires 5-10 seconds of IBM 360 time.

Figure 3 is a p-optimized survey of the $\emptyset$ -space associated with the orbit pair which was examined in Figure 2. Note that the optimum impulse (feet per second) for any $\emptyset_1, \emptyset_2$ may be read directly from the contour map. The advantages of the p-optimization technique should be apparent from a comparison of Figures 2 and 3. Complete description of a $\emptyset$ -space requires the generation of numerous cuts at various values of one of the variables.

Once these cuts are available, it is difficult to extract the optimum impulse information. The p-optimization technique allows complete description of the optimum impulse regions of a $\emptyset$ -space in one contour map. One may easily proceed from these contour maps to numerical searching programs which permit exact detailed examination of regions of particular interest.

Note that Figure 3 readily illustrates the circumstances of the four optimum transfers. At the same time it is possible to note their relative importance and to infer certain other properties such as error sensitivity, i. e., $\partial I / \partial \emptyset_1$ and $\partial I / \partial \emptyset_2$ may be approximated from the contour maps. Other features of $\emptyset$ -space structure such as regions of extreme impulse are apparent. Contour levels are usually printed beside the maps and areas not contoured require more impulse than the highest level contoured.

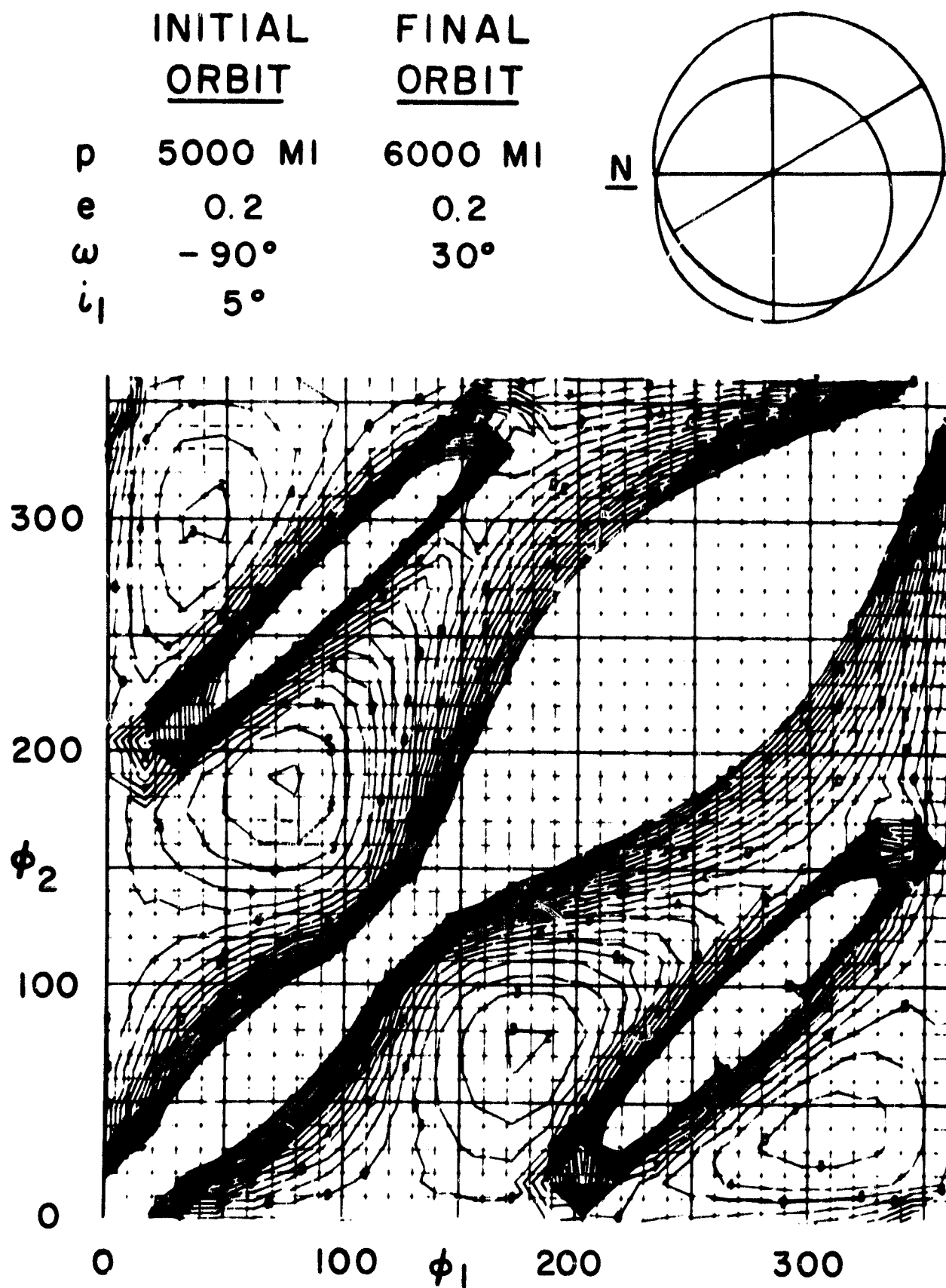


Figure 3. Optimum Impulse Contours for an Inclined Asymmetric Orbit Pair
(Contour Interval = 500 ft/sec)

V. EXACT OPTIMIZATION BY STEEP DESCENT

INTRODUCTION

Within subroutine SURVEY which produces the impulse contour map of Fig. 3, it has been possible to construct logic to identify points which are possible local minima. The program may be instructed to investigate each of these points by conducting a detailed search for a local minimum. The logic and procedures required for such investigations are detailed in this section along with some examples of the program's functioning.

IMPULSE MINIMIZATION

Minimization of Eq. 9 by a steepest descent technique requires computation of the gradient. Recalling that differentiation of Eq. 9 provided the following expression:

$$dI = \frac{(\underline{I}_1 \cdot d\underline{I}_1)}{|\underline{I}_1|} + \frac{(\underline{I}_2 \cdot d\underline{I}_2)}{|\underline{I}_2|}$$

One may proceed with further differentiation as follows:

$$\frac{\partial I}{\partial p_t} = \frac{\left(\underline{I}_1 \cdot \frac{\partial \underline{I}_1}{\partial p_t} \right)}{|\underline{I}_1|} + \frac{\left(\underline{I}_2 \cdot \frac{\partial \underline{I}_2}{\partial p_t} \right)}{|\underline{I}_2|} \quad (26)$$

$$\frac{\partial I}{\partial \phi_1} = \frac{\left(\underline{I}_1 \cdot \frac{\partial \underline{I}_1}{\partial \phi_1} \right)}{|\underline{I}_1|} + \frac{\left(\underline{I}_2 \cdot \frac{\partial \underline{I}_2}{\partial \phi_1} \right)}{|\underline{I}_2|} \quad (27)$$

$$\frac{\partial I}{\partial \phi_2} = \frac{\left(\underline{I}_1 \cdot \frac{\partial \underline{I}_1}{\partial \phi_2} \right)}{|\underline{I}_1|} + \frac{\left(\underline{I}_2 \cdot \frac{\partial \underline{I}_2}{\partial \phi_2} \right)}{|\underline{I}_2|} \quad (28)$$

Terms in the above expressions may be expanded.

$$\begin{aligned} \frac{\partial \underline{I}_1}{\partial \mathbf{p}_t} &= \pm \frac{\partial \underline{V}_{t1}}{\partial \mathbf{p}_t} - \frac{\partial \underline{V}_1}{\partial \mathbf{p}_t} \\ \frac{\partial \underline{I}_2}{\partial \mathbf{p}_t} &= \mp \frac{\partial \underline{V}_{t2}}{\partial \mathbf{p}_t} + \frac{\partial \underline{V}_2}{\partial \mathbf{p}_t} \\ \frac{\partial \underline{I}_1}{\partial \phi_1} &= \pm \frac{\partial \underline{V}_{t1}}{\partial \phi_1} - \frac{\partial \underline{V}_1}{\partial \phi_1} \\ \frac{\partial \underline{I}_1}{\partial \phi_2} &= \pm \frac{\partial \underline{V}_{t1}}{\partial \phi_2} - \frac{\partial \underline{V}_1}{\partial \phi_2} \\ \frac{\partial \underline{I}_2}{\partial \phi_1} &= \mp \frac{\partial \underline{V}_{t2}}{\partial \phi_1} + \frac{\partial \underline{V}_2}{\partial \phi_1} \\ \frac{\partial \underline{I}_2}{\partial \phi_2} &= \mp \frac{\partial \underline{V}_{t2}}{\partial \phi_2} + \frac{\partial \underline{V}_2}{\partial \phi_2} \end{aligned} \quad (29)$$

Noting that $\partial \underline{V}_1 / \partial \mathbf{p}_t$ and $\partial \underline{V}_2 / \partial \mathbf{p}_t$ are each zero, a simplified expression for

$\partial I / \partial \mathbf{p}_t$ may be obtained through several algebraic manipulations:

$$\frac{\partial \underline{I}}{\partial \underline{p}_t} = \pm \frac{1}{2\underline{p}_t} \left[\frac{\underline{I}_1 \cdot (\underline{v} - z\underline{U}_1)}{|\underline{I}_1|} - \frac{\underline{I}_2 \cdot (\underline{v} + z\underline{U}_2)}{|\underline{I}_2|} \right] \quad (30)$$

Additional expressions are obtained from Eqs. 26 - 29 by direct differentiation of the vector equations.

$$\begin{aligned} \frac{\partial \underline{V}_{t1}}{\partial \phi_1} = (\mu/p_t)^{1/2} & \left\{ \frac{p_t}{r_1} \underline{U}_2 \left[\frac{\partial \csc \Delta \theta}{\partial \phi_1} - \frac{\csc \Delta \theta}{r_1} \frac{\partial r_1}{\partial \phi_1} \right] \right. \\ & + \left(1 - \frac{p_t}{r_2} \right) \left[\underline{U}_1 \frac{\partial \csc \Delta \theta}{\partial \phi_1} + \csc \Delta \theta \frac{\partial \underline{U}_1}{\partial \phi_1} \right] \\ & \left. - \underline{U}_1 \frac{\partial \cot \Delta \theta}{\partial \phi_1} - \cot \Delta \theta \frac{\partial \underline{U}_1}{\partial \phi_1} \right\} \quad (31) \end{aligned}$$

$$\frac{\partial \underline{V}_1}{\partial \phi_1} = (\mu/p_1)^{1/2} \left[-\cos \phi_1, -\sin \phi_1 \cos i, -\sin \phi_1 \sin i \right] \quad (32)$$

$$\begin{aligned} \frac{\partial \underline{V}_{t1}}{\partial \phi_2} = (\mu/p_t)^{1/2} & \left\{ \frac{p_t}{r_1} \left[\underline{U}_2 \frac{\partial \csc \Delta \theta}{\partial \phi_2} + \csc \Delta \theta \frac{\partial \underline{U}_2}{\partial \phi_2} \right] \right. \\ & + \underline{U}_1 \left[\left(1 - \frac{p_t}{r_2} \right) \frac{\partial \csc \Delta \theta}{\partial \phi_2} + \frac{p_t}{r_2^2} \csc \Delta \theta \frac{\partial r_2}{\partial \phi_2} \right. \\ & \left. \left. - \frac{\partial \cot \Delta \theta}{\partial \phi_2} \right] \right\} \quad (33) \end{aligned}$$

$$\frac{\partial \underline{V}_1}{\partial \phi_2} = 0 \quad (34)$$

$$\begin{aligned} \frac{\partial \underline{V}_{t2}}{\partial \phi_1} = (\mu/p_t)^{1/2} \left\{ -\frac{p_t}{r_2} \left[\underline{U}_1 \frac{\partial \csc \Delta \theta}{\partial \phi_1} + \csc \Delta \theta \frac{\partial \underline{U}_1}{\partial \phi_1} \right] \right. \\ \left. + \underline{U}_2 \left[\left(1 - \frac{p_t}{r_1} \right) \frac{\partial \csc \Delta \theta}{\partial \phi_1} - \frac{p_t}{r_1^2} \csc \Delta \theta \frac{\partial r_1}{\partial \phi_1} \right. \right. \\ \left. \left. + \frac{\partial \cot \Delta \theta}{\partial \phi_1} \right] \right\} \end{aligned} \quad (35)$$

$$\frac{\partial \underline{V}_2}{\partial \phi_1} = 0 \quad (36)$$

$$\begin{aligned} \frac{\partial \underline{V}_{t2}}{\partial \phi_2} = (\mu/p_t)^{1/2} \left\{ \frac{p_t}{r_2} \underline{U}_1 \left[-\frac{\partial \csc \Delta \theta}{\partial \phi_2} + \frac{1}{r_2} \csc \Delta \theta \frac{\partial r_2}{\partial \phi_2} \right] \right. \\ \left. - \left(1 - \frac{p_t}{r_1} \right) \left[\underline{U}_2 \frac{\partial \csc \Delta \theta}{\partial \phi_2} + \csc \Delta \theta \frac{\partial \underline{U}_2}{\partial \phi_2} \right] \right. \\ \left. + \underline{U}_2 \frac{\partial \cot \Delta \theta}{\partial \phi_2} + \cot \Delta \theta \frac{\partial \underline{U}_2}{\partial \phi_2} \right\} \end{aligned} \quad (37)$$

$$\frac{\partial \underline{V}_2}{\partial \phi_2} = (\mu/p_2)^{1/2} [-\cos \phi_2, -\sin \phi_2, 0] \quad (38)$$

The remaining terms in Eqs. 31-38 may be computed from the following expressions

$$\frac{\partial \underline{U}_1}{\partial \phi_1} = [-\sin \phi_1, \cos \phi_1 \cos i, \cos \phi_1 \sin i] \quad (39)$$

$$\frac{\partial \underline{U}_2}{\partial \phi_2} = [-\sin \phi_2, \cos \phi_2, 0] \quad (40)$$

$$\frac{\partial r_1}{\partial \phi_1} = \frac{r_1^2 e_1 \sin(\phi_1 - \omega_1)}{p_1} \quad (41)$$

$$\frac{\partial r_2}{\partial \phi_2} = \frac{r_2^2 e_2 \sin(\phi_2 - \omega_2)}{p_2} \quad (42)$$

$$\frac{\partial \csc \Delta \theta}{\partial \theta_j} = - \csc \Delta \theta \cot \Delta \theta \frac{\partial \Delta \theta}{\partial \theta_j} \quad j = 1, 2 \quad (43)$$

$$\frac{\partial \cot \Delta \theta}{\partial \theta_j} = - \csc^2 \Delta \theta \frac{\partial \Delta \theta}{\partial \theta_j} \quad j = 1, 2 \quad (44)$$

The following convenient expression for $\Delta \theta$ allows computation of the remaining derivatives:

$$\Delta \theta = \cos^{-1} (\cos \theta_1 \cos \theta_2 + \sin \theta_1 \sin \theta_2 \cos i) \quad (45)$$

$$\frac{\partial \Delta \theta}{\partial \theta_1} = \frac{-(-\sin \theta_1 \cos \theta_2 + \cos \theta_1 \sin \theta_2 \cos i)}{\left[1 - (\cos \theta_1 \cos \theta_2 + \sin \theta_1 \sin \theta_2 \cos i)^2 \right]^{1/2}} \quad (46)$$

$$\frac{\partial \Delta \theta}{\partial \theta_2} = \frac{-(-\cos \theta_1 \sin \theta_2 + \sin \theta_1 \cos \theta_2 \cos i)}{\left[1 - (\cos \theta_1 \cos \theta_2 + \sin \theta_1 \sin \theta_2 \cos i)^2 \right]^{1/2}} \quad (47)$$

NUMERICAL OPTIMIZATION

Since the necessary conditions for a minimum impulse (setting Eqs. 26, 27, and 28 equal to zero) yield an intractable set of equations, one is faced with the problem of numerically solving an ordinary calculus problem involving the minimization of a function of three variables. Early attempts to minimize the impulse function considered here were successful but often required excessive amounts of computer time.⁽¹¹⁾ The scheme reported in Ref. 11 was a stepwise steep descent procedure similar to one of several methods discussed by Kelley:⁽¹⁴⁾ the local negative gradient was computed and a step-by-step search in that direction was continued until a minimum was reached. Then, a new gradient was computed and the search was continued in some new direction. This whole process was repeated until the minimum impulse was found.

A procedure which was found to be computationally superior to the steep descent scheme reported in Ref. 11 was to make each step extremely small and to recompute the gradient after each step. In this manner one may cause the search to proceed stepwise in the approximate direction of the steepest descent. Ref. 12 contains a comparison of the convergence of the two methods. Although these two methods were later shown to be inadequate for minimizing the functions to be considered here, they were capable of producing valuable results (some of these results appear in Refs. 15, 16, and 17).

Due to the wide variety of impulse functions which may occur, it was desirable to develop a descent method having sufficient flexibility to overcome the vagaries of any particular function. Further, a method which would provide information concerning the structure of a function's "valleys" was considered very desirable. In practice, this information is valuable because non-optimum orbital transfers corresponding to points on a "valley floor" often must be considered in order to satisfy rendezvous constraints.⁽¹⁵⁾ Rapid convergence was not considered to be a necessity. Instead, it was desired to insure a methodical search to the nearest local minimum. A rapidly convergent process which might converge upon some other nearby minimum was to be avoided.

COMPUTATIONAL PROCEDURES

The "adaptive descent" process conducts a step-by-step numerical search employing Eqs. 1 - 47. The search is terminated when the following necessary conditions for a local minimum have been achieved:

$$\left| \frac{\partial I}{\partial \phi_1} \right| \leq \epsilon_1, \quad \left| \frac{\partial I}{\partial \phi_2} \right| \leq \epsilon_1, \quad \left| \frac{\partial I}{\partial p_t} \right| \leq \epsilon_1, \quad 0 < \epsilon_1 \ll 1.0 \quad (48)$$

During the n 'th step of the search the gradient vector is computed and the $n + 1$ st coordinate vector is determined as follows:

$$\begin{bmatrix} \phi_1 \\ \phi_2 \\ p_t \end{bmatrix}_{n+1} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ p_t \end{bmatrix}_n - \alpha \begin{bmatrix} \frac{s_1}{s_j} & 0 & 0 \\ 0 & \frac{s_2}{s_j} & 0 \\ 0 & 0 & \frac{s_3}{s_j} \end{bmatrix} \cdot \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix}, \quad j = 1, 2, \text{ or } 3 \quad (49)$$

where α modifies the step size, the s_j are variable scaling parameters, and g_1 , g_2 , and g_3 are the components of a unit vector in the gradient direction (see below). Note that the scaling factors are normalized relative to one of the scaling parameters (s_j). This normalization is necessary since it provides a reference magnitude for the determination of the step size. Note that this normalization in no way restricts the direction of step since the ratios $s_1 : s_2 : s_3$ and $s_1/s_j : s_2/s_j : s_3/s_j$ are equivalent.

Eq. 49 is employed to generate points at which the impulse function, $I(\phi_1, \phi_2, p_t)$, is evaluated. An additional constraint on the process requires that the sequence of computed impulses $I(\phi_1, \phi_2, p_t)_n$ be monotone decreasing.

The control logic for the optimization process is as follows:

(1) If the inequality

$$I(\phi_1, \phi_2, p_t)_{n+1} < I(\phi_1, \phi_2, p_t)_n \quad (50)$$

is not satisfied, α is decreased by a prescribed percentage, and a new coordinate vector $(\theta_1, \theta_2, p_t)_{n+1}$ is computed. This n 'th stage of the process is repeated until Eq. 50 is satisfied or $\alpha \leq \epsilon_2$, $\alpha \leq \epsilon_2 \ll 1.0$. When $\alpha \leq \epsilon_2$ sufficient convergence has been achieved and the process is terminated.

- (2) If Eq. 50 is satisfied, the $(n+1)$ coordinate vector is adopted and the n 'th step is considered complete. If Eq. 50 is satisfied during each of a number of steps this successful behavior is rewarded by increasing α by a prescribed percentage.
- (3) Since the scaling parameters are normalized, only their relative size is of interest. To produce a change in relative size, one may simply penalize undesirable behavior by decreasing a given scale factor by a predetermined percentage. In practice, a scaling parameter (s_j) is decreased each time the corresponding component of the gradient vector (g_j) changes sign. No provision for increasing s_j is required since decreasing s_i , ($i \neq j$), automatically causes s_j to become relatively more influential.

This process control philosophy is simply a trial and error problem-solving procedure. It does, however, provide a flexible and reliable method of handling the inevitable scaling problems that were associated with steep descent or standard gradient methods.

Kelley⁽¹⁴⁾ notes that it is usually necessary to employ a normalization procedure when the variables have different dimensions. He also notes that it would be desirable to find a transformation that would map a complicated

impulse function into a more symmetrical form; e. g. , concentric spheres. The "adaptive descent" process tends to achieve both of these desirable objectives, and yet, it is a simple, reliable, method which is easily programmed for digital computer solution. The fact that only about one second of IBM 360 time is required to determine a typical local minimum of the impulse functions considered here is offered as further testimony to the practicality of this procedure.

CONVERGENCE PROPERTIES

An impulse function associated with a pair of inclined elliptical orbits, ($p_1 = 5000$ mi, $p_2 = 6000$ mi, $e_1 = e_2 = 0.2$, $\omega_1 = -90^\circ$, $\omega_2 = +30^\circ$, $i = 5^\circ$), was investigated with an IBM 360 FORTRAN H double precision computer program incorporating the adaptive descent technique. This particular function was previously studied in Ref. 1 by generating an optimum impulse contour map (Fig. 3).

The descent paths associated with a number of starting points have plotted in Fig. 4. This impulse function has well defined minima and therefore offers no significant optimization problems. The four minima predicted by contouring are quickly established with required accuracy (13 significant figures). Table 1 contains the parameters associated with each minimum.

The first curve in Fig. 5 illustrates the convergence of the adaptive descent program during an optimization involving one of the minima in Fig. 4. For contrast, an impulse function having nearly flat-bottomed

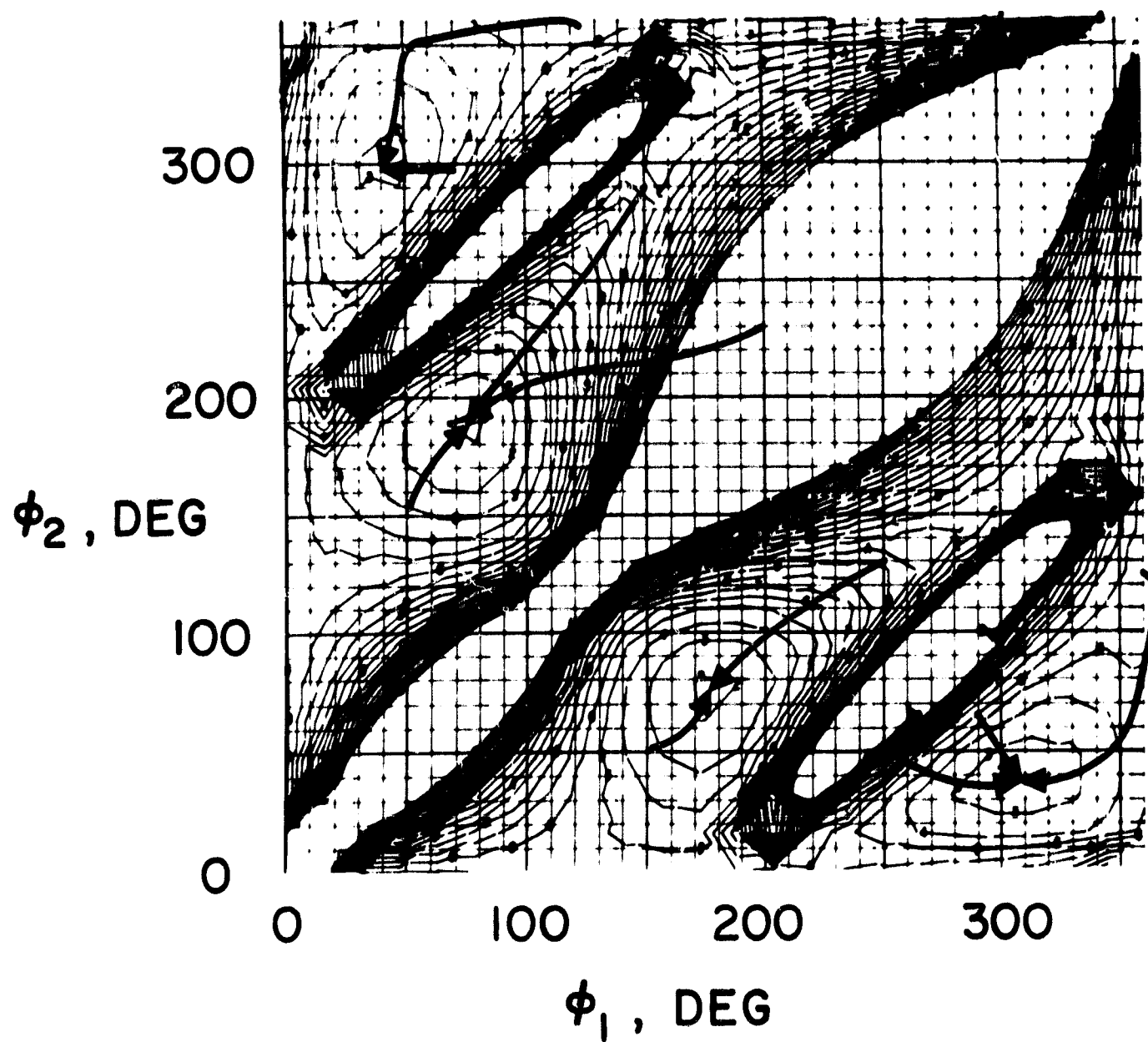


Figure 4. Descent Paths Plotted on Optimum-Impulse Contour Map

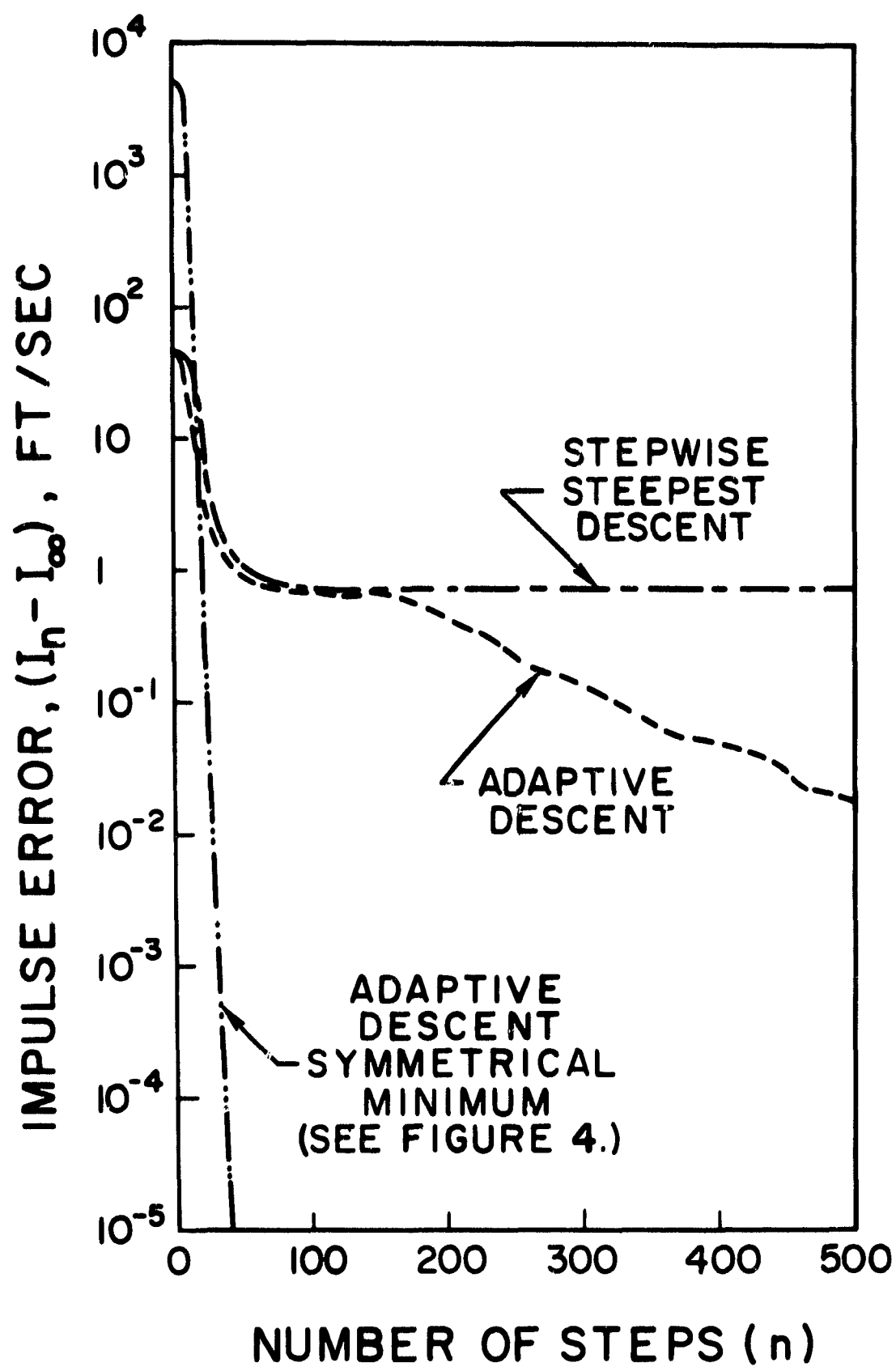


Figure 5. Convergence of Adaptive Descent and Stepwise Steepest Descent Methods

SPACE DIVISION OF NORTH AMERICAN ROCKWELL CORPORATION

Table 1. Optimum Transfer Parameters

Initial Orbit $p_1 = 5000.$ mi, $e_1 = 0.2$, $\omega_1 = -90.^\circ$, $i = 5.^\circ$				
Final Orbit $p_2 = 6000.$ mi, $e_2 = 0.2$, $\omega_2 = +30.^\circ$				
Optimum	ϕ_1 Deg	ϕ_2 , Deg	p_t , mi	Impulse, fps
1	73.8152	187.5568	6644.8496	4902.651223852
2	40.8343	298.2634	6617.7904	5343.148693477
3	177.8114	73.6465	4611.8023	5393.781144757
4	308.2034	37.7403	4592.8574	5654.191209679

long narrow "valleys" was optimized by the adaptive program and by stepwise steepest descent. This data is also plotted in Fig. 5. Although convergence is not as rapid for this particular function, it is clear that the adaptive method is superior to stepwise steepest descent.

Fig. 6 compares the two methods by plotting the search paths in the ϕ_1, p_t plane. The impulse level (I) corresponding to the number of steps in the search is also noted. By the end of the first 100 steps the adaptive program had found the bottom of the valley and had ascertained its long narrow shape. During the next 100 steps the program made rapid progress down the valley. Note that the impulse differences soon were in the 7th significant figure. It is also apparent that most of the movement throughout the space is concerned with these last few significant figures. The poor progress of a stepwise steepest descent process is plotted for comparison. One may observe that the adaptive descent program tends to move along the true path of steepest descent in spite of the fact that individual steps are not taken in the local negative gradient direction.

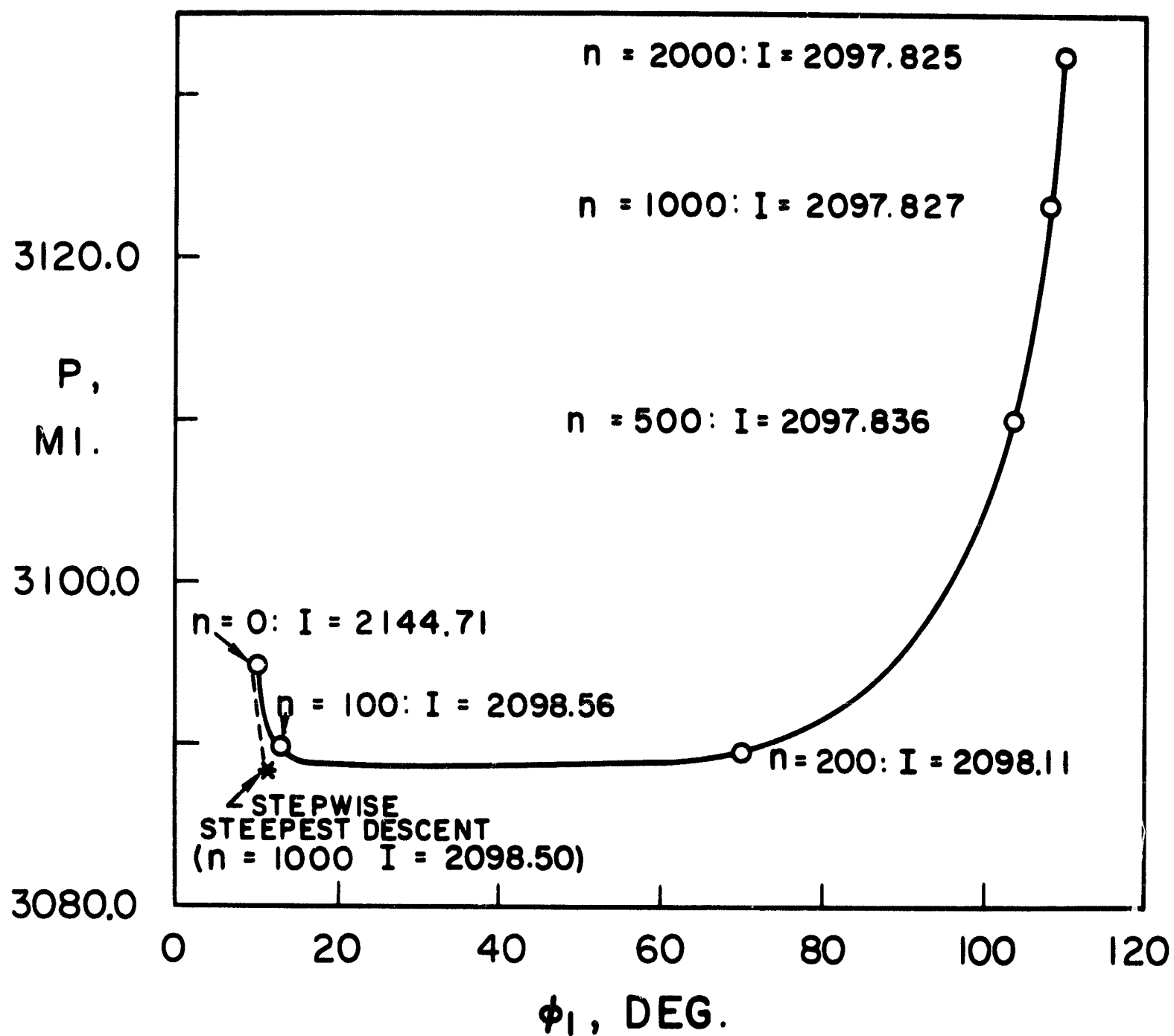


Figure 6. Path Followed During Adaptive Descent Optimization of a Function Having Long Narrow "Valleys"

SPACE DIVISION of NORTH AMERICAN ROCKWELL CORPORATION

Further numerical results may be found in References 18, 19 and 20.

There, extensive studies using the adaptative descent program show that two-impulse transfers between certain orbits require less velocity change than a continuous thrust transfer known as the Lawden Spiral. Until recently, the Lawden Spiral had been considered optimal.

VI. CONCLUSION

Methods for systematically studying optimum two-impulse orbital transfer between any pair of elliptical orbits have been developed and used to investigate the properties of an "impulse function space". A contouring technique was utilized to present large amounts of optimum impulse information in a concise easily understood form. The results obtained from function contouring have been verified through the use of steepest descent optimization procedures. An "adaptive" numerical method for the computation of optimum two-impulse transfers between inclined non-coapsidal elliptical orbits has been developed and incorporated into the overall optimization program. This program has proved to be a fast means of computing optimum transfer conditions with the accuracy needed for research and engineering studies. Optimum impulse transfers obtained from the double precision steep descent optimization always indicate a smaller impulse than those obtained from contouring. In all instances the insight gained through contouring methods proves to be an invaluable aid to subsequent precise optimization and/or research into the nature of optimization problems.

The appendix of this report contains a computer listing of the required IBM 360 FORTRAN H digital computer program. A program description and a sample problem which demonstrates typical usage are also presented.

VII. REFERENCES

1. McCue, G.A., "Optimum Two-Impulse Orbital Transfer and Rendezvous Between Inclined Elliptical Orbits", AIAA J., 1, 1865-1872 (1963).
2. McCue, G.A., "Optimization and Visualization of Functions", AIAA J., 2, 99-100 (January 1964).
3. DesJardins, P.R., Bender, D.F., and McCue, G.A., "Orbital Transfer and Satellite Rendezvous", North American Aviation, Inc., SID 62-870 (31 August 1962).
4. McCue, G.A., "Visualization of Functions by Stereographic Techniques", North American Aviation, Inc., SID 63-170 (20 January 1963).
5. McCue, G.A., "Optimization by Function Contouring Techniques", North American Aviation, Inc., SID 63-171 (10 February 1963).
6. McCue, G.A., and Bender, D.F., "Optimum Transfers Between Nearly Tangent Orbits", North American Aviation, Inc., SID 65-1097 (May 1, 1965).
7. McCue, G.A., and DuPrie, H.J., "Improved Fortran IV Function Contouring Program", North American Aviation, Inc., SID 65-672 (1 April 1965).
8. McCue, G.A., Dworetsky, M. and DuPrie, H.J., "Fortran IV Stereographic Function Representation and Contouring Program", North American Aviation, Inc., SID 65-1182 (1 September 1965).

SPACE DIVISION OF NORTH AMERICAN ROCKWELL CORPORATION

9. Lee, G., "An Analysis of Two-Impulse Orbital Transfer", AIAA J., 2, 1767-1773 (October 1964).
10. McCue, G. A., and DuPrie, H. J., "Optimum Two-Impulse Orbital Transfer Function Contouring Program," North American Aviation, Inc., SID 65-1181 (1 September 1965).
11. Kerfoot, H. P., Bender, D. F., and DesJardins, P. R., "Analytical Study of Satellite Rendezvous (Final Report)", North American Aviation, Inc., MD 59-272 (October 20, 1960).
12. DesJardins, P. R., and Bender, D. F., "Extended Satellite Rendezvous Study", (First Quarterly Report), North American Aviation, Inc., SID 61-304 (September 14, 1961).
13. McCue, G. A. and Hoy, R. C., "Optimum Two-Impulse Orbital Transfer Program", North American Aviation, Inc., SID 65-1119 (August 1, 1965).
14. Kelley, H. J., "Method of Gradients," Optimization Techniques, edited by G. Leitmann (Academic Press, New York, 1962), pp. 206-210.
15. Bender, D. F., "Rendezvous Possibilities with the Impulse of Optimum Two-Impulse Transfer," Space Rendezvous Rescue and Recovery, edited by N. V. Petersen (Western Periodicals Co., North Hollywood, Calif., 1963) pp. 271-291.
16. Hoelker, R. F., "Orbit Transfer Studies by Numerical Processes," Marshall Space Flight Center, AERO-61-24 (March 24, 1961).

SPACE DIVISION OF NORTH AMERICAN ROCKWELL CORPORATION

17. Bender, D. F., "Optimum Coplanar Two-Impulse Transfer Between Elliptic Orbits," J. Aerospace Eng., 21, 44-52 (1962).
18. McCue, G. A., and Bender, D. F., "Numerical Investigation of Minimum Impulse Orbital Transfer," SID 64-423, North American Aviation, Inc. (October 1, 1964).
19. Bender, D. F., and McCue, G. A., "Optimal One-Impulse Transfers Between Coplanar Elliptical Orbits," SID 64-1859, North American Aviation, Inc. (October 1, 1964).
20. McCue, Gary A., and Bender, David F., "Numerical Investigation of Minimum Impulse Orbital Transfer," AIAA J., 3, 2328-2334 (December 1965).

APPENDIX I
COMPUTER PROGRAM DESCRIPTION

APPENDIX I

COMPUTER PROGRAM DESCRIPTION

INTRODUCTION

This appendix contains a discussion of the nature and structure of the authors' two-impulse orbital transfer optimization program. The program is the result of 10 years of work in the field of orbit mechanics. It therefore represents a distillation and compilation of numerous methods and techniques which have proved their usefulness.

The authors have long been interested in developing a single program that provided a maximum amount of useful information about a particular optimization and yet allowed an unfamiliar user to input minimal parameters concerning his problem. Many of the program's functions that would ordinarily be specified by the analyst are performed automatically — thus providing rather complete information in a single computer run. In the event additional runs prove necessary the initial run will have provided sufficient information for accurately choosing the parameters required for the supplementary runs. In short, the program is highly problem-oriented and should prove very useful to the systems analyst or engineer who is primarily interested in answers rather than theory. Of course, the program will still prove very useful for guiding theoretical work aimed at exploring the nature of orbital transfer maneuvers.

PROGRAM STRUCTURE

The program has been written in the FORTRAN H programming language for the IBM 360 Model 65 digital computer. Of course, many of the subroutines were originally written in FORTRAN II and FORTRAN IV and therefore reflect the restrictions inherent in those earlier languages.

The program is basically concerned with three kinds of operations:
(1) inputting the data and parameters and performing preliminary

transformations, (2) performing surveys of impulse functions and generating contour maps, and (3) the precise optimization of particular transfer maneuvers by steep descent. This program structure is prominently reflected in the modular tree structure diagram appearing as Figure 7. The numerous subroutines required by the program are noted in the diagram together with their line of communication. The functions of the subroutines are best discussed with reference to Table 2, which provides a brief description of the duties of each individual module. The main program calls five high level subroutines directly; i.e., DATSET, INPUT, DEX, SURVEY, and STEEP. The DATSET subroutine simply provides reasonable values for all input data parameters which would otherwise be read through the use of subroutine INPUT. The INPUT subroutine functions in such a manner that it is necessary to enter only those data parameters which are being modified for a particular run. That is, a data field may be left blank and the program will not change the previously existing value of the parameter which is in storage. The data cards are each numbered in columns 79 and 80. The number appearing in these columns identifies the data card and thereby tells the program what it should expect to be on that particular card. For instance, card 01 contains the elements of the initial orbit. Because of this procedure, the data cards may be read into the program in any order desired. The program will then read and translate each card individually. A non-zero entry in column 78 instructs the program to terminate the card-reading operation and return to the calling program. The above procedures allow one to process an additional case by changing only one data parameter on one card and by simply adding that data card at the end of the data deck. This process is illustrated further in the sample data sheets of Figure 8. Once the data is entered the DEX subroutine performs transformations which return the orbital elements to the form of Equations 1 through 8, corresponding to Figure 1. That is, the fundamental plane is the plane of the final orbit and the reference direction is the line of intersection of the initial and final orbit planes.

SPACE DIVISION of NORTH AMERICAN ROCKWELL CORPORATION

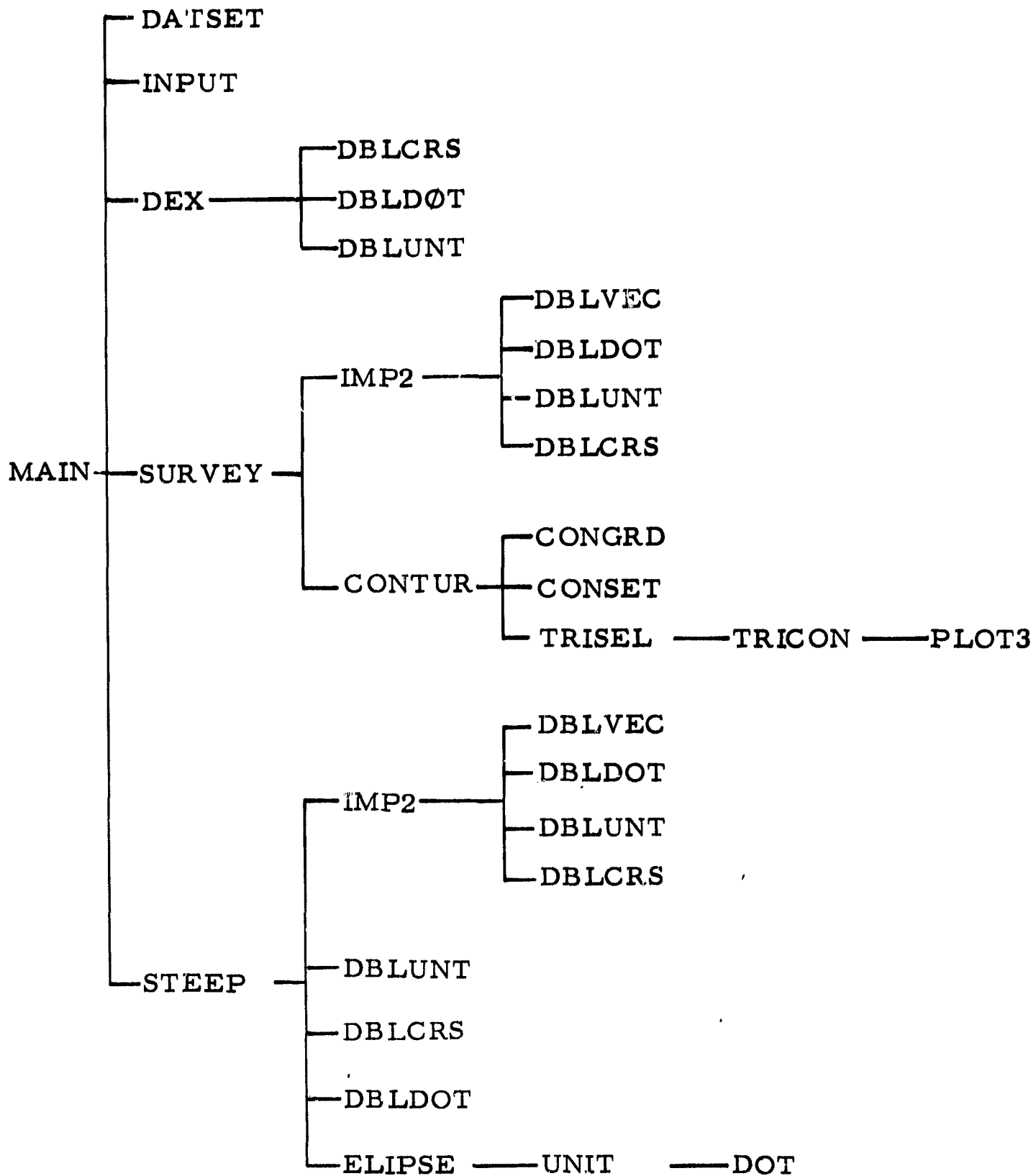


Figure 7. Modular Tree Structure of Two-Impulse Orbital Transfer Optimization Program

Table 2. Functions of Program Modules

<u>Module Name</u>	<u>Module Function</u>
MAIN	Executive program for the optimizing package. Performs common assignments and handles the logical operations associated with solving a particular optimization problem.
DATSET	Provides a standard set of data that will suffice for most problems.
INPUT	Reads and prints initial orbit geometry and other data parameters to be used in impulse computations. Reads only those parameters that are specified on data cards.
DEX	Computes orbit geometry information to be used in impulse computations. Performs coordinate transformations.
SURVEY	Generates an array of the impulse and p_t values which are to be contoured. Calls the contouring package and generates a p-optimized map of the impulse function. It then scans the array of impulse points and identifies possible local minima for further investigation.
STEEP	Executive subroutine for conducting an adaptive steep descent search for a local optimum.
ELIPSE	Projects initial and transfer orbits on final orbit plane. Plots all three orbits, node line, impulse vectors, perigee lines and radius vectors to the transfer points. Labels the resulting figure.

Table 2. Functions of Program Modules (Continued)

Module Name	Module Function
IMP2	Performs optional p-optimization. Computes impulse and its gradient as a function of $\emptyset_1$, $\emptyset_2$, p_t .
CØNTUR	Executive subroutine for the contouring package (see Reference 7 for further details).
CONGRD	Generates grid (if desired) and scaling information for subsequent plotting of contour map. Plots contour identification table, parameter table, and titles.
CØNSET	Computes and/or sets appropriate contour intervals.
TRISEL	Selects triangle to be contoured.
TRICØN	Performs actual contouring operation including interpolation to compute the end points of the line segments which are then plotted by PLØT3.
PLØT3	Plots individual line segments which form contour lines. Labels contour lines.
DBLDØT	Double precision function that computes the dot-product of two vectors.
DBLCRS	Double precision subroutine that takes the cross-product of two vectors.
DBLUNT	Double precision function that unitizes a vector.
DBLVEC	Double precision subroutine that sets up a vector by stipulation of its components.
DØT	Single precision function that computes the dot-product of two vectors.
UNIT	Single precision function that unitizes a vector.

FORTTRAN FIXED IO DIGIT DECIMAL DATA

DECK NO. _____ PROGRAMMER _____ DATE _____ PAGE 1 of 4 JOB NO. _____

NUMBER	IDENTIFICATION	DESCRIPTION	DO NOT KEY PUNCH
1	TEST ONE		
13			
25			
37			
49			
61			
1	UNIT SPECIFICATION	TITLE: Case title information	
13			
25			
37			
49			
61			
1	EARTH	BODY: Name of body (e.g., EARTH, MOON, SUN...) (left justified)	
13			
25			
37			
49			
61			
1	BRITISH	UNSYS: Unit system ("BRITISH", "METERS", "KILOMETERS", OR "BODY") (left justified)	
13			
25			
37			
49			
61			
1		GXM1: Gravitation constant used with "BODY" specification	
13			
25			
37			
49			
61			
1	BODY UNIT		
13			
25			
37			
49			
61			
1	5 0 0 0 . 0	Elements of Initial orbit	
13			
25			
37			
49			
61			
1	0 . 4		
13			
25			
37			
49			
61			
1	0 . 5		
13			
25			
37			
49			
61			
1	0 . 0		
13			
25			
37			
49			
61			
1	- 9 0 . 0		
13			
25			
37			
49			
61			
1	0 1		
13			
25			
37			
49			
61			
1	6 0 0 0 . 0	Elements of final orbit	
13			
25			
37			
49			
61			
1	0 . 2		
13			
25			
37			
49			
61			
1	0 . 0		
13			
25			
37			
49			
61			
1	0 . 0		
13			
25			
37			
49			
61			
1	3 0 . 0		
13			
25			
37			
49			
61			

FORM 114-C-17 REV. 7-68-VELLUM

Figure 8. Data Input Format and Data Description

FORTRAN FIXED 10 DIGIT DECIMAL DATA

DECK NO. _____		PROGRAMMER _____		DATE _____		PAGE 2 of 4 _____		JOB NO. _____	
NUMBER		IDENTIFICATION		DESCRIPTION		DO NOT KEY PUNCH			
1	1 2 0 . 0			θ_1	Search initiation point for steepest descent				
13	2 3 0 . 0			θ_2					
25	5 0 0 0 . 0			r_1					
37									
49									
61									
1									
13	2 .				AMPINC: Reward factor to multiply (increase) step size α				
25	. 5				AMPDEC: Penalty factor to multiply (decrease) step size α				
37	1 . - 0 8				FREEZE: Step size amplitude tolerance, ϵ_2 , for iteration step				
49	2 .				XAMP: Initial value of α				
61	2 .				SAVE: Maximum factor by which any component of the stepping vector may increase from one iteration to the next				
1	1 .				BL46: If non-zero ellipses drawn on CRT				
13	. 8 5								
25	1				SCINFC: Factor to decrease scale s_1 ; changing step direction = j , determines to which s_j the scaling factors are normalized				
37	5				ISCL: Number of successful iterations before α is rewarded				
49									
61									
1	1 .				s_1	Scale factors for movement in parameter space (initial values) (if $s_3 \neq 0$, program will not minimize)			
13	1 .				s_2				
25	1 0 0 .				s_3				
37	1 . - 0 5				ϵ_1	Tolerance for partial derivative convergence			
49	2 0 0				MAXIT: Maximum number of iterations allowed				
61	5				IPRIN: Controls printout of iterations; e.g., every 5'th iteration				

FORM 114-C-17 REV. 7-53 - VELLUM

FORM 114-C-17 REV. 7-83-VELLUM

Figure 8. Data Input Format and Data Description (Continued)

FORTRAN FIXED 10 DIGIT DECIMAL DATA			
DECK NO.	PROGRAMMER	DATE	PAGE 3 of 4 JOB NO.
NUMBER	IDENTIFICATION	DESCRIPTION	DO NOT KEY PUNCH
1	4	INAPS: Maximum number of best rank suspected minima to be contoured in detail	
13	5	NSTEEP: Maximum number of steep descents into the best ranking suspected minima	
25	0	IUSE: Perform a final steep descent using the search initiation point of data card 03 if IUSE = 0	
37	. 2	SCXY: θ_1 and θ_2 scale factor for local area maps	
49	1	SCIMP: Impulse scale factor for local area maps	
61	2 0		
1	2 4 .	SPAR(1,1): Number of points in contour grid (e.g., 24 x 24)	
13	1 5 .	SPAR(2,1): θ_1 interval for contour grid	
25	1 5 .	SPAR(3,1): θ_2 interval for contour grid	
37	1 8 0 .	SPAR(4,1): θ_1 of center of contour map	
49	1 8 0 .	SPAR(5,1): θ_2 of center of contour map	
61	5 0 0 .	SPAR(6,1): Impulse contour interval	
1	1	} Second Case: New data for card 20	
13	1		
25	0		
37	. 1		
49	. 2		
61	2 0		
1		} Second Case: Single parameter on card 01 changed	
13			
25			
37	. 1 1 1		
49			
61	1 0 1		

FORM 114-C-17 REV. 7-68-VELLUM

Figure 8. Data Input Format and Data Description (Continued)

FORTRAN FIXED 10 DIGIT DECIMAL DATA

DECK NO.	PROGRAMMER	DATE	PAGE 4 of 4	JOB NO.
NUMBER	IDENTIFICATION	DESCRIPTION	DO NOT KEY PUNCH	
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				
49				
61				
1				
13				
25				
37				

If the program has been instructed to produce a contour map of the p-optimized impulse function for the entire function space, the main program will transfer control to subroutine SURVEY. Within SURVEY numerous two-impulse transfers are computed so as to produce a grid of data points for subsequent contouring. Each point is produced by entering the IMP2 subroutine and finding a two-impulse transfer for the given take-off and arrival points such that impulse is minimized with respect to p_t . Once the grid of points is available the contouring package (Reference 7) is instructed to produce a contour map similar to that of Figure 3. Since the entire space is usually searched during this initial optimization, and since one usually desires to know if local minima exist within the survey space; the program next examines each point to determine if it exhibits less impulse than its immediate neighbors. Since such points usually correspond to local minima they are tabulated and ranked for further computations.

At this point in the computations the user has the option of instructing the program to perform a detailed examination of each suspected minima. If desired the "n" suspected minima with least impulse will be examined by first producing a small scale contour map of the region immediately surrounding the suspected minimum and then performing a steep descent optimization using the suspected minimum as an initial iterate. If no contour maps are desired the user still has the option of conducting a steep descent search beginning at each suspected minimum. The program also draws a picture of the initial, final, and transfer orbits associated with each minimum found by steep descent. Parameter tables giving the user many necessary input and optimized results are included in the final output summary for each optimization.

From the above description it should be clear that the program is extremely versatile. The above quick description will be supplemented in Appendix II by examining the results of running a typical sample problem. The reader should note that considerable attention has been given to providing output which is adequate and meaningful but not superfluous. The experienced

programmer, when he gets into the details of the program, will discover that he has considerably more options than those presented in the brief description and sample problem given here.

DATA STRUCTURE

Figure 8 contains a description of the data input formats, and Table 3 lists the input formats associated with each type of data card. Note that the data cards are classified according to the number appearing in columns 79 and 80 except for the first two cards which are read independently and only once at the beginning of each computer run. The first card is merely for case description and allows the programmer to insert his own alphanumeric identification. The second card is interesting in that it allows the user to specify the central body and the unit system he wishes. For instance, he may write EARTH, MOON, SUN, VENUS, MARS, or JUPITER in the first field and the program will utilize the appropriate gravitational constant for those bodies. Similarly, BRITISH, METERS, KILOMETERS, or BODY may be written in the second field and the program will use the corresponding unit system. In the case of the BODY specification, the program will expect the user to supply an appropriate constant for g-times-mass of the attracting center. These first two cards are unique in that they are read only once at the beginning of the computer run and may not be reread. All other cards may be repeated as often as desired to produce additional cases.

At this point in the computations the program enters subroutine INPUT and reads the orbital elements, constants and control parameters from the remaining cards. As noted earlier, each card and its format are identified by the number appearing in columns 79 and 80. The program continues reading until it encounters a non-zero entry in column 78. Figure 8 also includes a description of each data parameter beside the field where it appears on the data card. These descriptions should suffice, but a few additional words of clarification are offered in Appendix II. Of course, one who requires complete knowledge of the functions of the input parameters may resort to examining the program's listings in Appendix III.

Table 3. FORTRAN Data Input Formats

Card Identification	Format
TITLE	(18A4)
BODYUNIT	(6A4, D12.0)
01	(6D12.0)
02	(6D12.0)
03	(6D12.0)
04	(6D12.0)
05	(2D12.0, 2I12, 2D12.0)
06	(4D12.0, 2I12)
20	(3I12, 2E12.0), I12)
21	(6E12.0)
22	(6E12.0)
23	(6E12.0)
24	(6E12.0)
25	(6E12.0)

APPENDIX II
EXAMPLE OF THE USE OF THE
ORBITAL TRANSFER OPTIMIZATION PROGRAM

APPENDIX II

SAMPLE OPTIMIZATION PROBLEM

Much of the foregoing discussion is difficult to understand without a realistic example. For this reason this appendix presents a rather comprehensive sample problem.

Figure 9 contains the images of several data cards which were used to produce the computer output appearing as Figures 9 through 27. The orbits utilized for this example are inclined, elliptical, and non-coapsidal. Since the British unit system was used the output parameters will appear in miles and ft/sec. Figure 9 represents the first and second pages of printed output and it contains the image of the data cards in the order in which they were used. Note that not all of the data cards defined in Appendix I, Figure 8, are included and that this means that many of the deleted input data parameters were supplied by subroutine DATSET. Figure 10 contains a printout of the data array common block showing the status of each data parameter just prior to execution of the case. As noted earlier subroutine DEX transforms the elements to a standard orientation. These transformed elements appear next in the printed output (Figure 11).

The program then generates the contour map of the entire function space which appears in Figure 12. In this case, four possible minima were identified and noted by printing the numbers 1, 2, 3, and 4 (in groups of five) on the contour map and by the printed output appearing in Figure 11. Since the data parameters indicated that further investigation of the suspected minima was desired, the program generated four detailed contour maps (Figures 13-16). The program then automatically conducted a steep descent search for a local minimum within each suspected optimum region. The results of the searches are tabulated in Figures 17-20.

The printout for the steep-descent optimization is rather straightforward. It includes the three independent variables (θ_1 , θ_2 , p_t), their partial derivatives and the impulse required. The "GAINS" are the current values of the s_i , "AMP" is the final value of α , and "IFAIL" is the number of times Equation 50 is not satisfied. The last two columns of each iteration printout contain the iteration number and the current value of IFAIL.

Figures 21-24 give the initial, final, and transfer orbits associated with each local minimum. The descent path followed by the program is noted by a sequence of characters plotted near the center of each contour map. The final search point has been darkened for additional emphasis. Note that in this case the search initiation point was quite near the search termination point. The four transfer orbits corresponding to the four locally minimum transfers between the initial and final orbits are quite interesting to examine. Note that two of these transfers are internal—that is, they tend to be inside both the initial and final orbits—and that two of the orbits are external. Note also that the transfer angle for two of the cases is greater than 180° . Data card 22 was used to produce the additional map which resulted in the output seen in Figures 25-27. In this case an additional detailed contour map was requested and its scale was changed. Note that the program finds only one local minimum in this second map.

At this point the analyst could have included any number of additional data cards and the program would have generated the specified additional cases. For instance, entirely new orbital elements could have been introduced by including new cards 01 and 02. This procedure is illustrated in Figure 8.

SPACE DIVISION OF NORTH AMERICAN ROCKWELL CORPORATION

DATE (1/18/59, TIME = 74.896

CASE IDENTIFICATION
TEST OF NEW UNIT SPECIFICATION

RDV IS FORTH
UNIT SYSTEM IS BRITISH
CXM = 0.056450 C5

DATE 3/1/69 TIME = 74.306

DATA CARD IMAGES	
12.	15.
5000.	.4
6000.	.2
120.	230.

181.1	0	22
-5C.	0	1
30.	0	2
	1	3

00
70.

15.50

15.
4.
2
230.

12.
500.
600.
120.

Figure 9.

SPACE DIVISION OF NORTH AMERICAN ROCKWELL CORPORATION

[illegible]

Figure 10.

DATE 01/10/50, TIME = 74.056

TO ANSWER 247 FLEET

Case	Age	Sex	Occupation	Duration of illness	Site of lesion	Pathological changes	Microscopic findings	Diagnosis
1	45	M	Farmer	10 years	Brain	Chronic	Microscopic	Chronic
2	55	F	Housewife	5 years	Brain	Chronic	Microscopic	Chronic
3	65	M	Teacher	15 years	Brain	Chronic	Microscopic	Chronic
4	75	F	Retired	20 years	Brain	Chronic	Microscopic	Chronic
5	85	M	Unemployed	25 years	Brain	Chronic	Microscopic	Chronic

СВЯТЫЙ ПРАВОСЛАВНЫЙ ЦЕРКОВЬ

[illegible]

אין ענין דאס ווילן. 173. 1

Rank	1-100	0.1-1	0.01-0.1	D	Impulse
2	0.6313661	0.2	0.2651370	0.7504730	0.5651010
1	0.7576760	0.2	0.2759300	0.7574700	0.5466610
2	0.7352520	0.2	0.2636230	0.4225530	0.6374340
4	0.7675570	0.3	0.2709090	0.4227710	0.6307940

Parameters on cards 21-25 are printed for each basic map generated.

Figure 11.

SPACE DIVISION of NORTH AMERICAN ROCKWELL CORPORATION

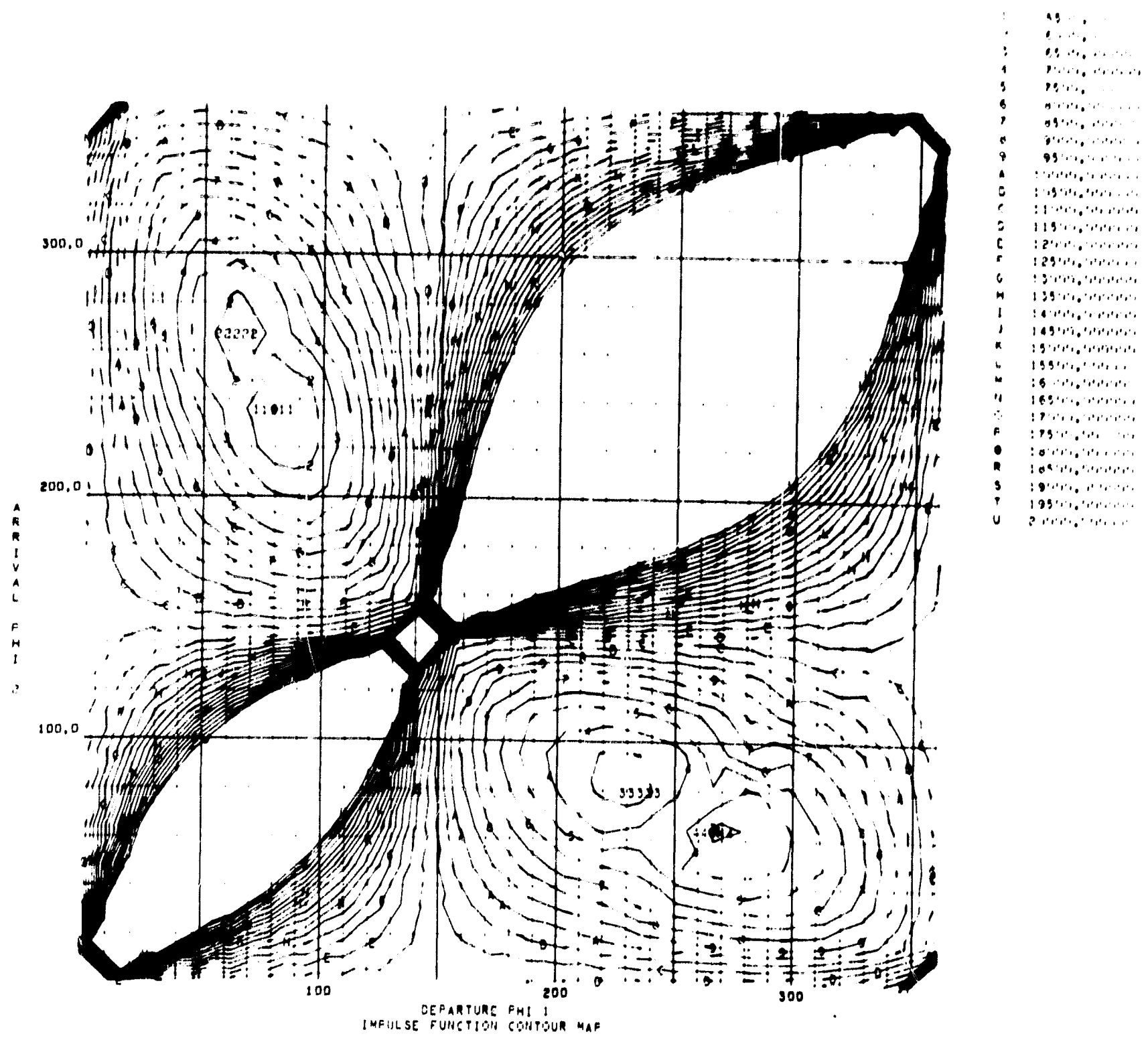
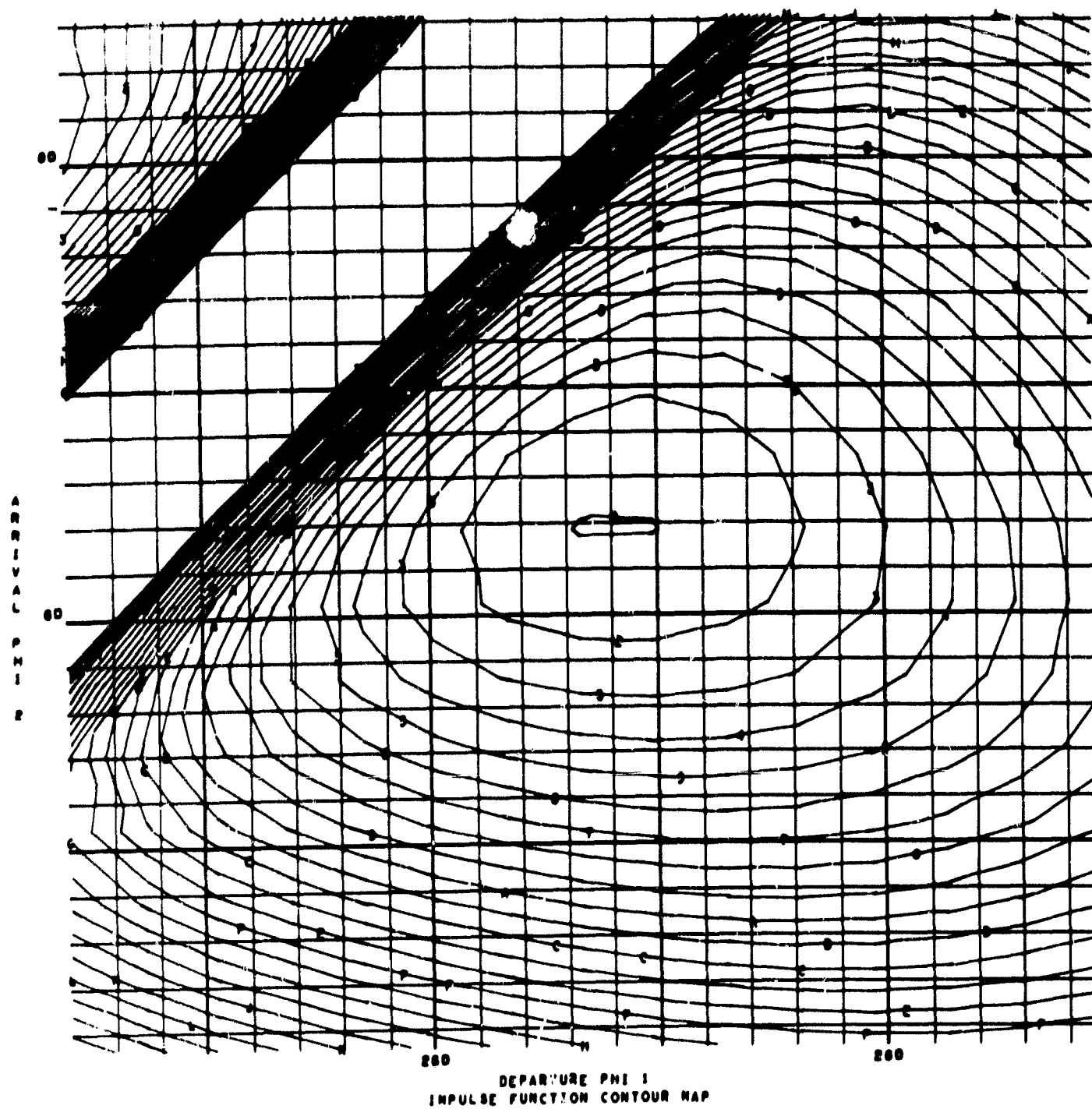


Figure 12.

SPACE DIVISION OF NORTH AMERICAN ROCKWELL CORPORATION

RANK OF OPTIMUM 4



I	6399.992167
Z	6449.988791
3	6499.984373
4	6549.980469
5	6599.976562
6	6649.972656
7	6699.968750
8	6749.964844
9	6799.960937
A	6849.957031
B	6899.953125
C	6949.949219
D	6999.945312
E	7049.941406
F	7099.937500
G	7149.933594
H	7199.929687
I	7249.925781
J	7299.921875
K	7349.917969
L	7399.914062
M	7449.910156
N	7499.906250
O	7549.902344
P	7599.898437
Q	7649.894531
R	7699.890625
S	7749.886719
T	7799.882812
U	7849.878906

Figure 13.

SPACE DIVISION OF NORTH AMERICAN ROCKWELL CORPORATION

RANK OF OPTIMUM : 3

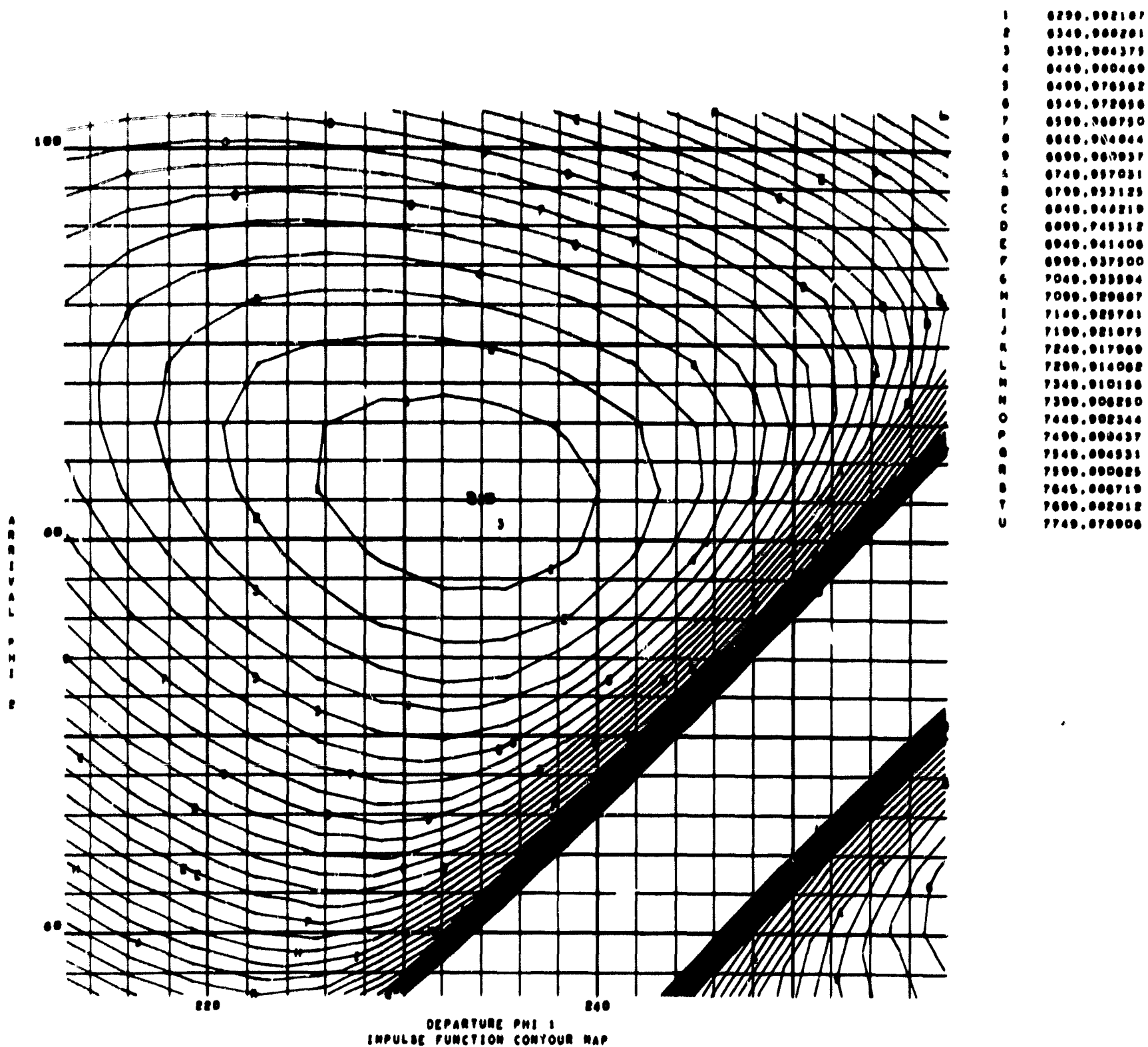


Figure 14.

DATA OF OPTIMUM 7

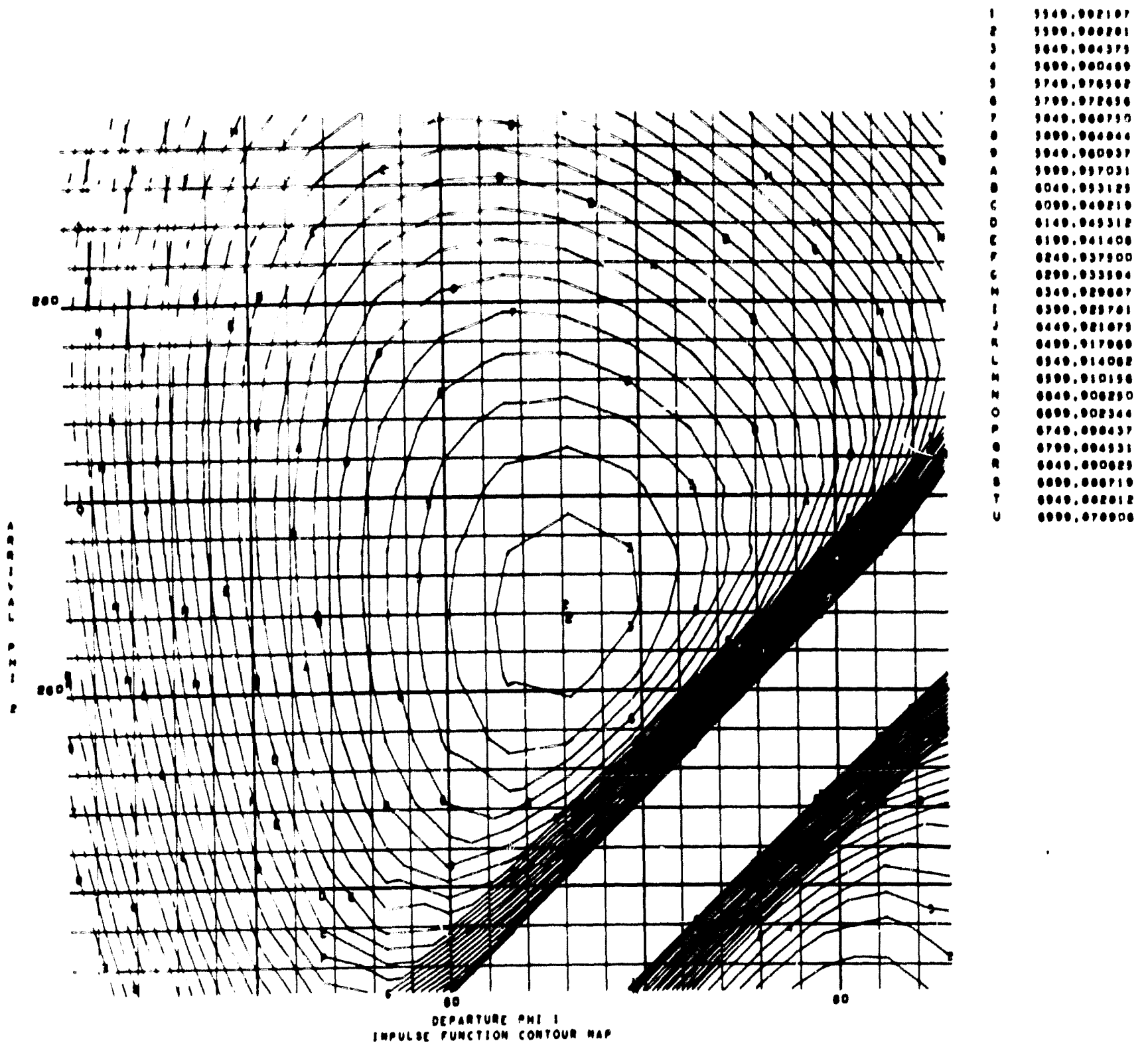


Figure 15.

RANK OF OPTIMUM : 1

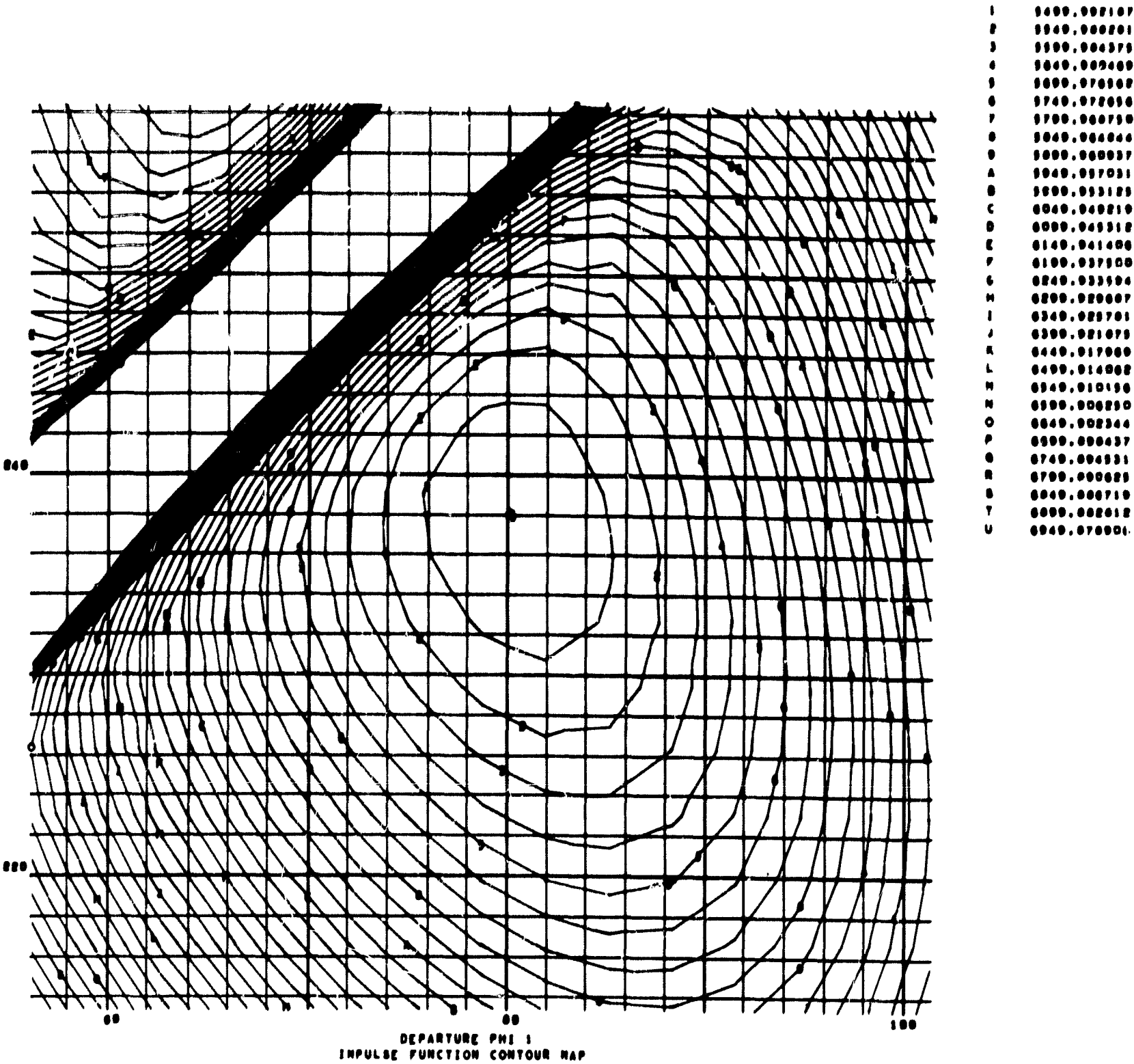


Figure 16.

ADVISORY BOARD OF THE NATIONAL YOUTH SERVICE

[illegible]

*** FINAL TRANSFER CRADITIONS ***

	V CPM0CAFMF	Z CPM0NEMF	W SPTIME
VFCT00	-0.262710390130	04	0.263154307770
VFCT01	-0.355778510750	04	0.306307458600
TOTAL			0.6395541766160

Figure 17.

ADAPTIVE STEEP DESCENT OPTIMIZATION OF TWO-IMPULSE TRANSFER

PHI1	PHI2	P	NI/DPI	NI/DP2	NI/DP	-----IMPULSE-----	-----GAINS-----	ITWC								
235.2524	75.3608	4.2555350	03	25.321364	-82.398534	0.005230	6.27434531011526300	03	1.00	00	1.00	02	0	0		
234.5569	80.7919	4.2551830	03	4.972354	-13.362065	12.546405	6.26702412126041500	03	1.00	00	1.00	02	5	1		
234.5487	82.1245	4.2455570	03	12.697227	6.354399	2.226101	6.26213255177474600	03	1.00	00	8.50	-01	2.70	02	10	7
234.4727	82.1128	4.2488950	03	14.650191	1.637290	0.100651	6.26194771231007200	03	1.00	00	8.50	-01	1.90	02	15	9
234.3172	82.1005	4.2486890	03	12.285702	0.779608	0.091492	6.26172472243028600	03	1.00	00	8.50	-01	1.90	02	20	9
234.0050	82.0546	4.2482090	03	7.763949	-0.602229	-0.001746	6.26141971979463100	03	1.00	00	7.20	-01	1.60	02	25	8
233.6241	82.0553	4.2480750	03	0.106593	-0.486111	1.252451	6.26126260211751700	03	1.00	00	5.20	-01	1.40	02	30	8
233.5531	82.1011	4.2474500	03	1.547355	-2.516786	-0.234187	6.26121785525059800	03	1.00	00	5.20	-01	8.50	01	35	13
233.5252	82.1220	4.2474940	03	0.693517	-1.156525	0.164494	6.26121120485866500	03	1.00	00	5.20	-01	7.20	01	40	13
233.4815	82.1622	4.2472710	03	0.454132	-0.611375	-0.036037	6.26120355560851700	03	1.00	00	5.20	-01	4.40	01	45	13
233.4578	82.1791	4.2472310	03	0.014541	0.012295	0.073455	6.26120242343893900	03	1.00	00	4.40	-01	2.00	01	50	15
233.4513	82.1810	4.2472030	03	0.049939	-0.079781	0.011856	6.26120227962435800	03	1.00	00	3.90	-01	1.40	01	55	17
233.4478	82.1850	4.2471790	03	0.020304	-0.030279	-0.009953	6.26120223585110000	03	1.00	00	3.90	-01	1.20	01	60	19
233.4473	82.1855	4.2471910	03	-0.000257	0.000935	0.000221	6.26120223411640800	03	1.00	00	3.90	-01	1.20	01	65	24
233.4473	82.1854	4.2471310	03	0.000054	0.000015	-0.000038	6.26120223411537500	03	1.00	00	5.20	-01	1.20	01	70	26

AMP= C.762936450-05

IFAIL=

AMP= C.762935450-05 IFALL= 26

IMPULSE	*** FINAL TRANSFER CONDITIONS ***	CENTER IS	EARTH	UNIT SYSTEM...MILES, FT/SEC
FIRST VECTOR	X COMPONENT	Z COMPONENT	MAGNITUDE	
SECOND VECTOR	Y COMPONENT	-0.275241294330	03	C.247930535850 04
TOTAL	-0.423066114550 05	-0.290253690540 03		C.378189687560 04
	INITIAL CRBIT	TRANSFER CRBIT		C.626120223410 03
P	C.530000000000 04	0.424718074540 04		
F	C.430000000000 00	0.237783533910 00		
I (DEG)	C.530000000000 00	0.835299165030 00		
R A NOCE (DEG)	C.C	-0.978145552790 02		
ARC PER (DEG)	-C.530000000000 02	0.302828113680 02		
A	C.595239095240 04	0.450171208520 04		
PERIOD	C.533058713630 04	0.613676007240 04		
TIME PT RAC	0.375408371260 04			
PHI-1, PHI-2, P	0.233447316930 03	0.424718074540 04		
RANK OF OPTIMUM =	3			

Figure 18.

SPACE DIVISION OF NORTH AMERICAN ROCKWELL CORPORATION

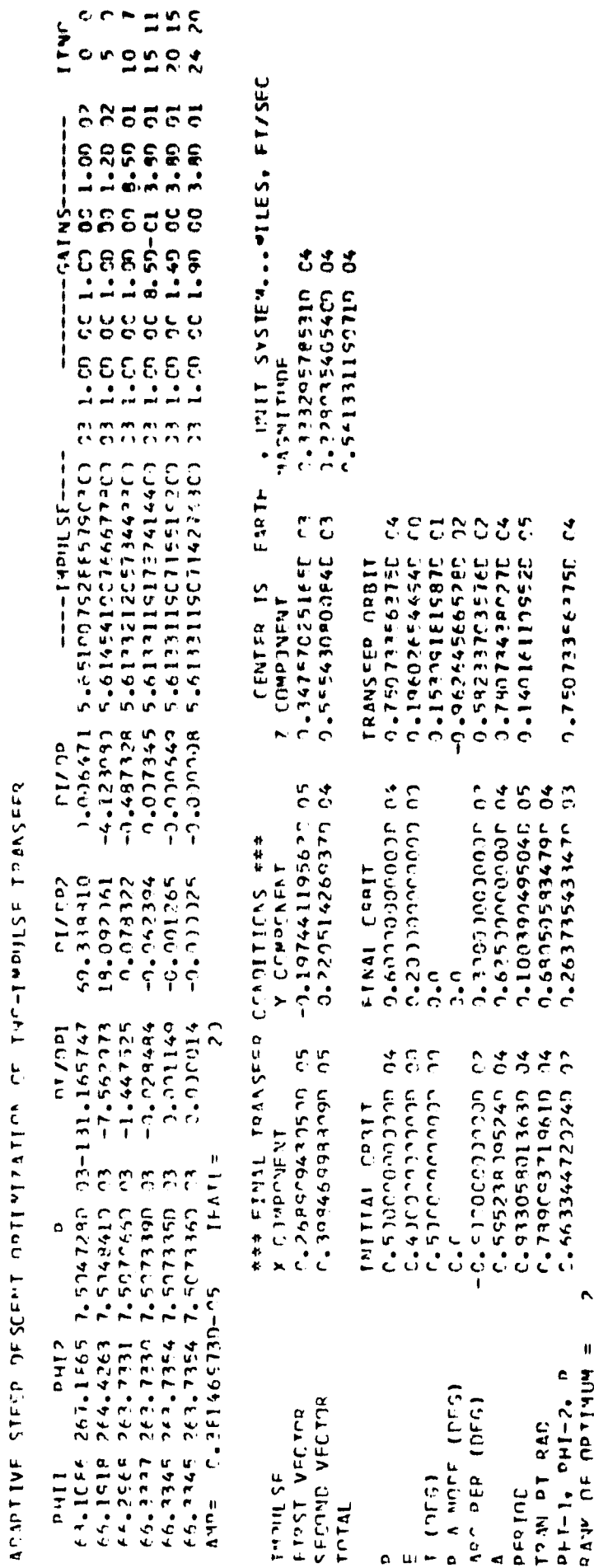


Figure 19.

SPACE DIVISION OF NORTH AMERICAN ROCKWELL CORPORATION

ADAPTIVE STEP DESCENT OPTIMIZATION OF TWO-IMPULSE TRANSFER									
PHI	PHI2	PHI	PHI2	PHI	PHI2	PHI	PHI2	PHI	PHI2
79.7608	235.8824	7.5745560	03	-57.179363	-47.846362	0.036958	5.46590875641009200	03	1.00 00 1.00 02
87.1642	238.0768	7.5758560	03	-5.067732	1.350778	0.150892	5.45671296989282000	03	1.00 00 1.20 00 1.00 02
87.9772	238.0243	7.5763550	03	-0.710599	0.603109	0.050828	5.45668053370510400	03	1.00 00 1.60 00 1.00 02
87.2910	239.0016	7.5763820	03	0.009170	-0.057648	-0.044343	5.45667951283698000	03	1.00 00 1.60 00 1.00 02
87.2531	237.5555	7.5764390	03	-0.019485	0.075580	0.033744	5.45667940808084700	03	1.00 00 1.60 00 1.40 02
87.2532	237.5577	7.5764380	03	-0.001929	-0.000514	-0.000555	5.45667943681933000	03	1.00 00 2.70 00 1.20 02
87.2532	237.5577	7.5764390	03	-0.000716	0.000496	0.000062	5.45667940677824000	03	1.00 00 2.30 00 1.00 02
87.2924	237.5577	7.5764390	03	0.000742	-0.000331	-0.000346	5.45667940677751900	03	1.00 00 2.30 00 1.50 01
AMP= C.381465730-05 IFAIL= 21									
*** FINAL TRANSFER CONDITIONS ***									
X COMPONENT									
-0.314153304380 04									
-0.195660565510 04									
Y COMPONENT									
0.772878130540 03									
0.911485115340 03									
Z COMPONENT									
0.6000000000 04									
0.2000000000 03									
0.0									
0.0									
0.3000000000 03									
0.6251000000 04									
0.1000000000 05									
0.728680059240 04									
0.237307657150 03									
INITIAL ORBIT									
0.5000000000 04									
0.4000000000 03									
0.5000000000 03									
0.0									
-0.5000000000 02									
0.555234025240 04									
0.613058713630 04									
0.825455155540 04									
0.802933407980 02									
FINAL ORBIT									
0.757643537320 04									
0.1261003775140 03									
0.129837754010 01									
0.579976571560 02									
-0.108337477530 03									
0.769868066010 04									
0.127245157000 05									
0.757643537320 04									
CENTER IS EARTH									
7 COMPONENT									
0.242542001120 02									
0.442125355320 03									
UNIT SYSTEM.....MILS. FT/SEC									
ALTITUDE									
0.325325149170 04									
0.220338751510 04									
0.545667940680 04									
IMPULSE									
FIRST VECTOR									
SECOND VECTOR									
TOTAL									
P									
F									
T (DEG)									
P A ANGLE (DEG)									
ARG PER (DEG)									
A									
PERIOD									
TRAIL PT RAD									
PHI-1, PHI-2, P									
RAVE OF OPTIMUM = 1									

Figure 20.

SPACE DIVISION of NORTH AMERICAN ROCKWELL CORPORATION

	*** FINAL TRANSFER CONDITIONS ***		CENTER IS EARTH , UNIT SYSTEM...MILES, FT/SEC	
IMPULSE	X COMPONENT	Y COMPONENT	Z COMPONENT	MAGNITUDE
FIRST VECTOR	-0.242218380180 04	0.455759171590 03	-0.377944254600 03	0.293154307770 04
SECOND VECTOR	-0.355228918750 04	0.145267926670 04	-0.447794255710 03	0.386387450400 04
TOTAL				0.639941766160 04
P	INITIAL ORBIT	FINAL ORBIT	TRANSFER ORBIT	
E	0.300000000000 04	0.800000000000 04	0.427644754600 04	
I (DEG)	0.400000000000 00	0.200000000000 00	0.199067704730 00	
R A MODE (DEG)	0.500000000000 00	0.0	0.123732800360 01	
ARG PER (DEG)	0.0	0.0	0.642853790310 02	
A	-0.900000000000 02	0.300000000000 02	-0.148360902860 03	
PERIOD	0.595238095240 04	0.625000000000 04	0.445280713200 04	
TRAN PT RAD	0.933098013630 04	0.100390495040 03	0.603723424620 04	
PHI-1, PHI-2, P	0.357198800420 04	0.914911664110 04		
RANK OF OPTIMUM =	0.268102811640 03	0.642853790310 02	0.427644754600 04	

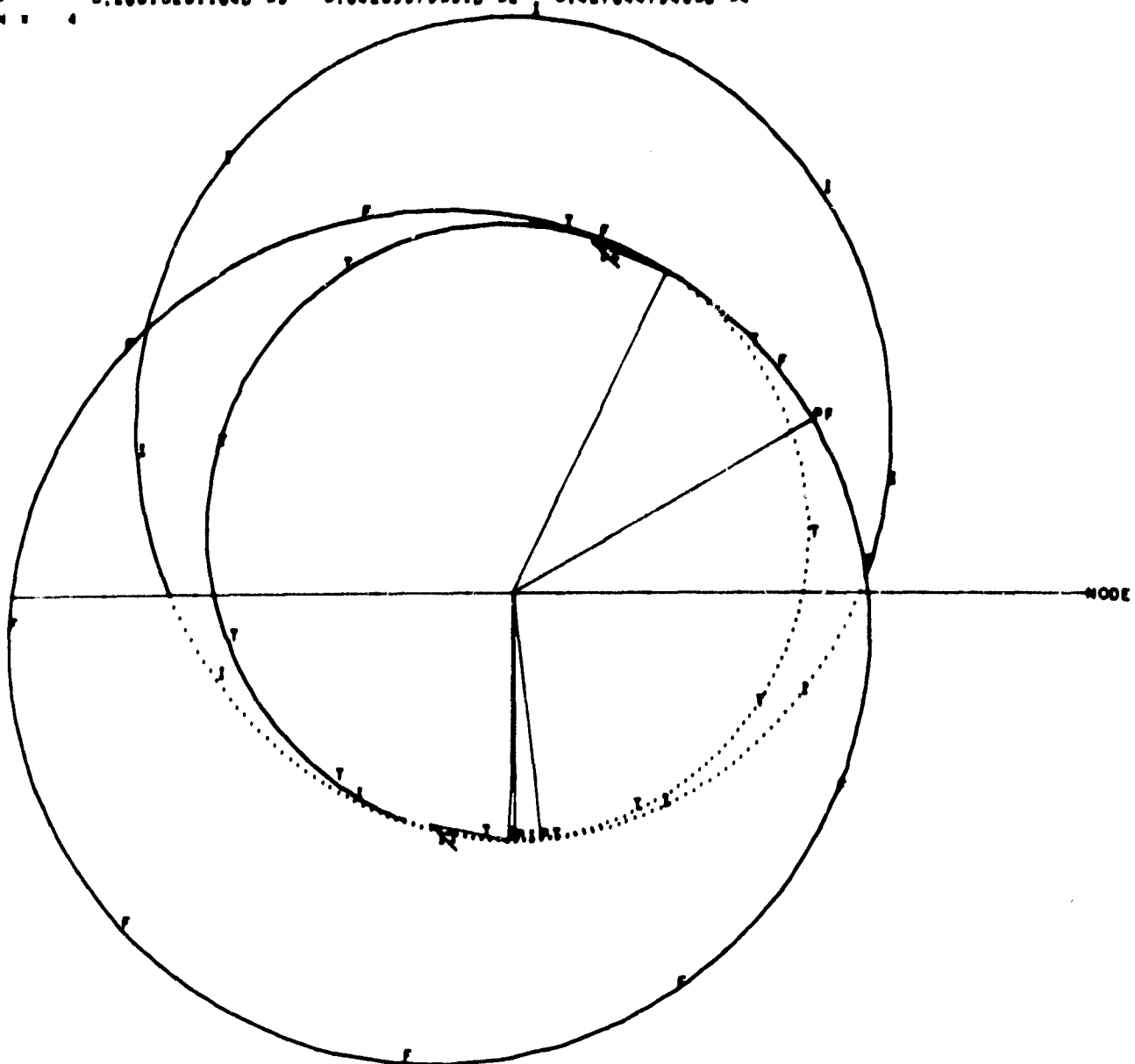


Figure 21.

SPACE DIVISION OF NORTH AMERICAN ROCKWELL CORPORATION

	*** FINAL TRANSFER CONDITIONS ***			CENTER IS EARTH	UNIT SYSTEM...MILES, FT/SEC
	X COMPONENT	Y COMPONENT	Z COMPONENT		MAGNITUDE
IMPULSE					
FIRST VECTOR	-0.934148290680 03	0.263977362380 03	-0.275241394330 03		0.247930533890 04
SECOND VECTOR	-0.423066114550 03	0.122004933640 03	-0.290293690540 03		0.370109687980 04
TOTAL					0.626120223410 04
	INITIAL ORBIT	FINAL ORBIT	TRANSFER ORBIT		
P	0.900000000000 04	0.600000000000 04	0.424718074940 04		
E	0.400000000000 00	0.200000000000 00	0.237763533910 00		
I (DEG)	0.900000000000 00	0.0	0.839299169030 00		
R A NODE (DEG)	0.0	0.0	-0.978145552790 02		
ARG PER (DEG)	-0.900000000000 02	0.300000000000 02	0.302028113880 02		
A	0.395238095240 04	0.625000000000 04	0.450171206920 04		
PERIOD	0.933098013630 04	0.100390499040 03	0.613676007240 04		
TRAN PT RAD	0.378406371260 04	0.534463282900 04			
PHI-1, PHI-2, P	0.253447316930 03	0.821854447210 02	0.424718074940 04		
RANK OF OPTIMUM :	3				

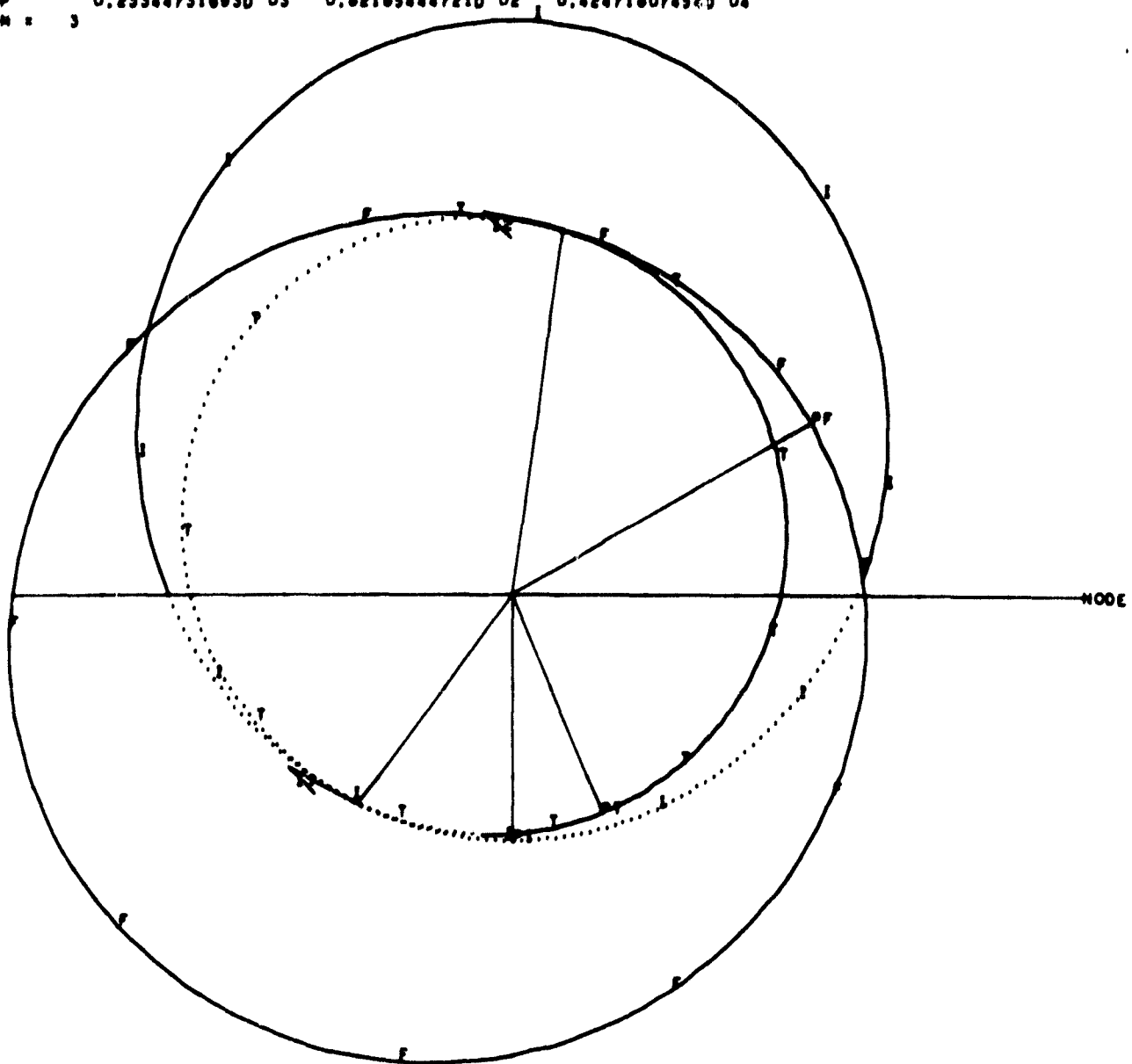


Figure 22.

SPACE DIVISION of NORTH AMERICAN ROCKWELL CORPORATION

	*** FINAL TRANSFER CONDITIONS ***				CENTER IS EARTH , UNIT SYSTEM...MILES, FT/SEC	
	X COMPONENT	Y COMPONENT	Z COMPONENT		MAGNITUDE	
IMPULSE	0.288909430500 05	-0.197441195620 05	0.347570251650 03		0.333293765510 04	
FIRST VECTOR	0.398469888090 05	0.220514269370 04	0.595430800840 03		0.228035405400 04	
SECOND VECTOR					0.561331190710 04	
TOTAL						
P	INITIAL ORBIT	FINAL ORBIT	TRANSFER ORBIT			
E	0.500000000000 04	0.600000000000 04	0.750733563750 04			
I (DEG)	0.400000000000 00	0.200000000000 00	0.196026544940 00			
R A NODE (DEG)	0.500000000000 00	0.0	0.153001819870 01			
ARG PER (DEG)	0.0	0.0	-0.962645665200 02			
A	-0.900000000000 02	0.300000000000 02	0.582337035700 02			
PERIOD	0.595238095240 04	0.625000000000 04	0.780734380270 04			
TRAN PT RAD	0.933058013630 04	0.100380495040 05	0.140161109520 05			
PHI-1, PHI-2, P	0.789093719610 04	0.680505634780 04	0.750733563750 04			
RANK OF OPTIMUM = 2	0.663344720240 02	0.263735433470 03				

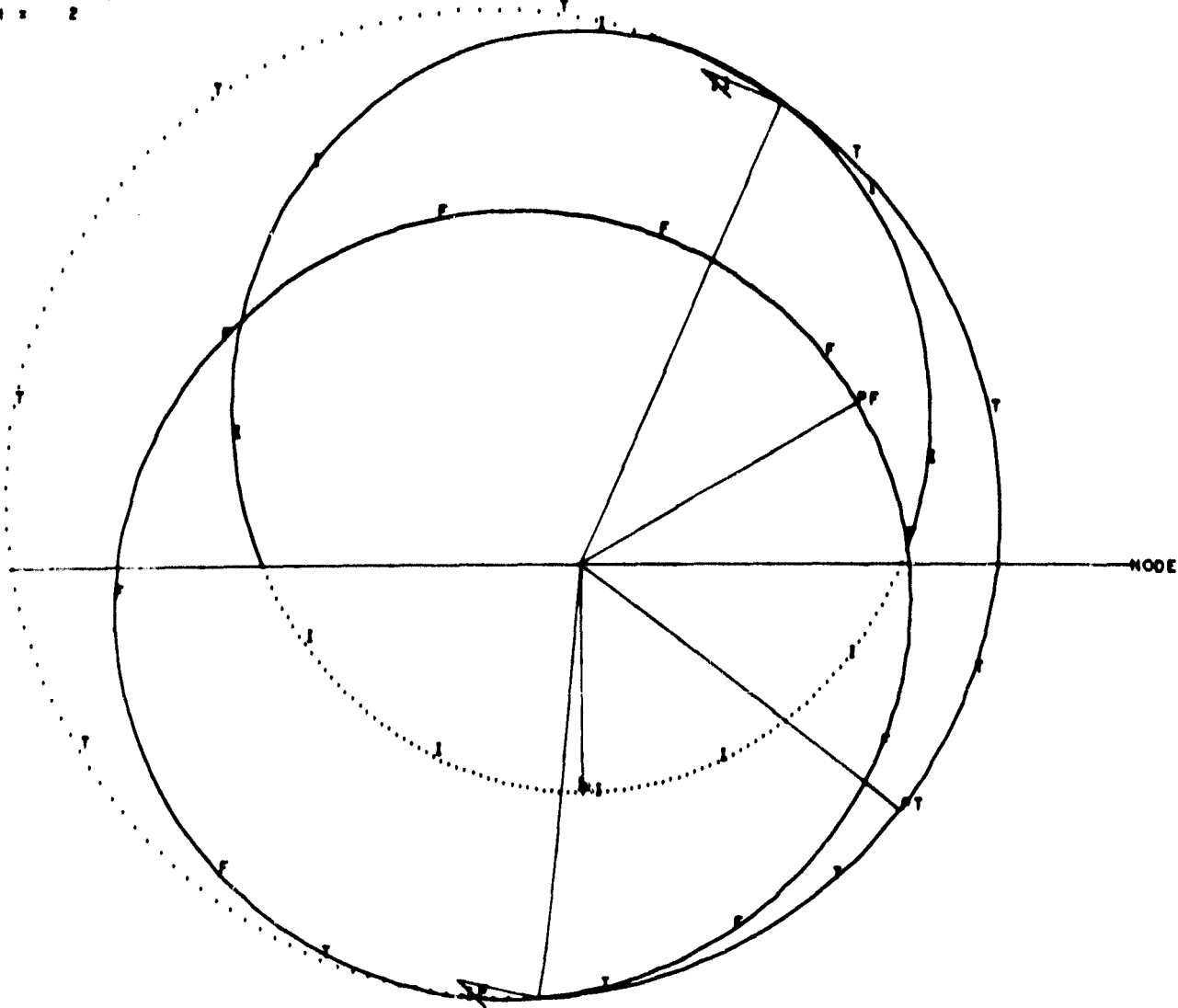


Figure 23.

SPACE DIVISION OF NORTH AMERICAN ROCKWELL CORPORATION

	000 FINAL TRANSFER CONDITIONS 000			CENTER IS EARTH , UNIT SYSTEM...MILES, FT/SEC	
	X COMPONENT	Y COMPONENT	Z COMPONENT	MAGNITUDE	
IMPULSE					
FIRST VECTOR	-0.314193304000 04	0.772070130040 03	0.342042001120 03	0.323320140170 04	
SECOND VECTOR	-0.105600000010 04	0.011400110340 03	0.442125300320 03	0.220330701010 04	
TOTAL				0.343007040000 04	
	INITIAL ORBIT	FINAL ORBIT	TRANSFER ORBIT		
P	0.000000000000 04	0.000000000000 04	0.707043003730 04		
E	0.400000000000 00	0.200000000000 00	0.120000770140 00		
I (DEG)	0.000000000000 00	0.0	0.120000770400 01		
R A NODE (DEG)	0.0	0.0	0.070070071000 02		
ARG PER (DEG)	-0.000000000000 02	0.300000000000 02	-0.100307477030 03		
A	0.000230000000 04	0.025000000000 04	0.700000000000 04		
PERIOD	0.033000013630 04	0.100300400040 03	0.137245107000 03		
TRAN PT RAD	0.000400100040 04	0.720000000000 04			
PHI-1, PHI-2, P	0.000000000000 02	0.237007007100 03	0.707043003730 04		
RANK OF OPTIMUM =	1				

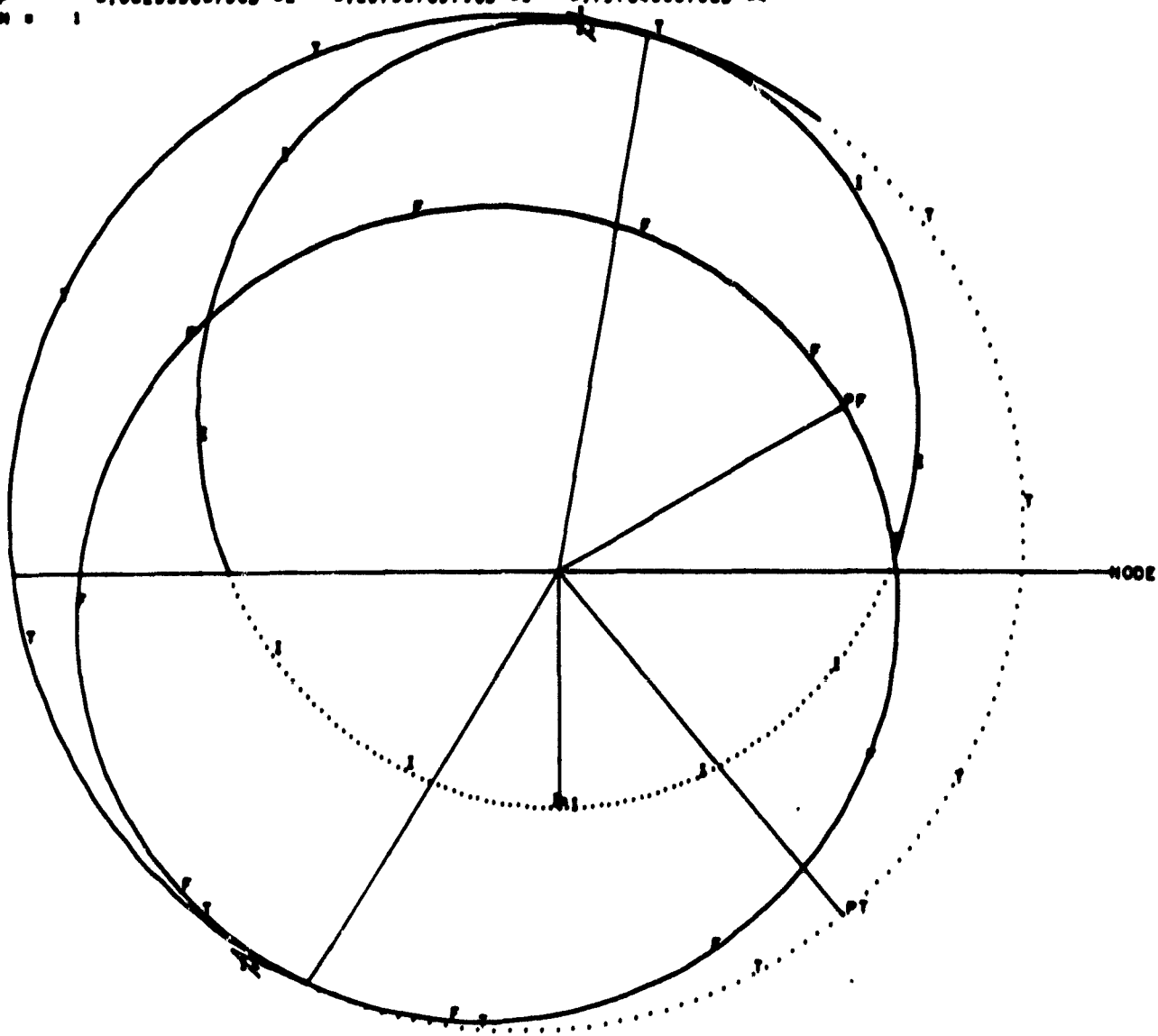


Figure 24.

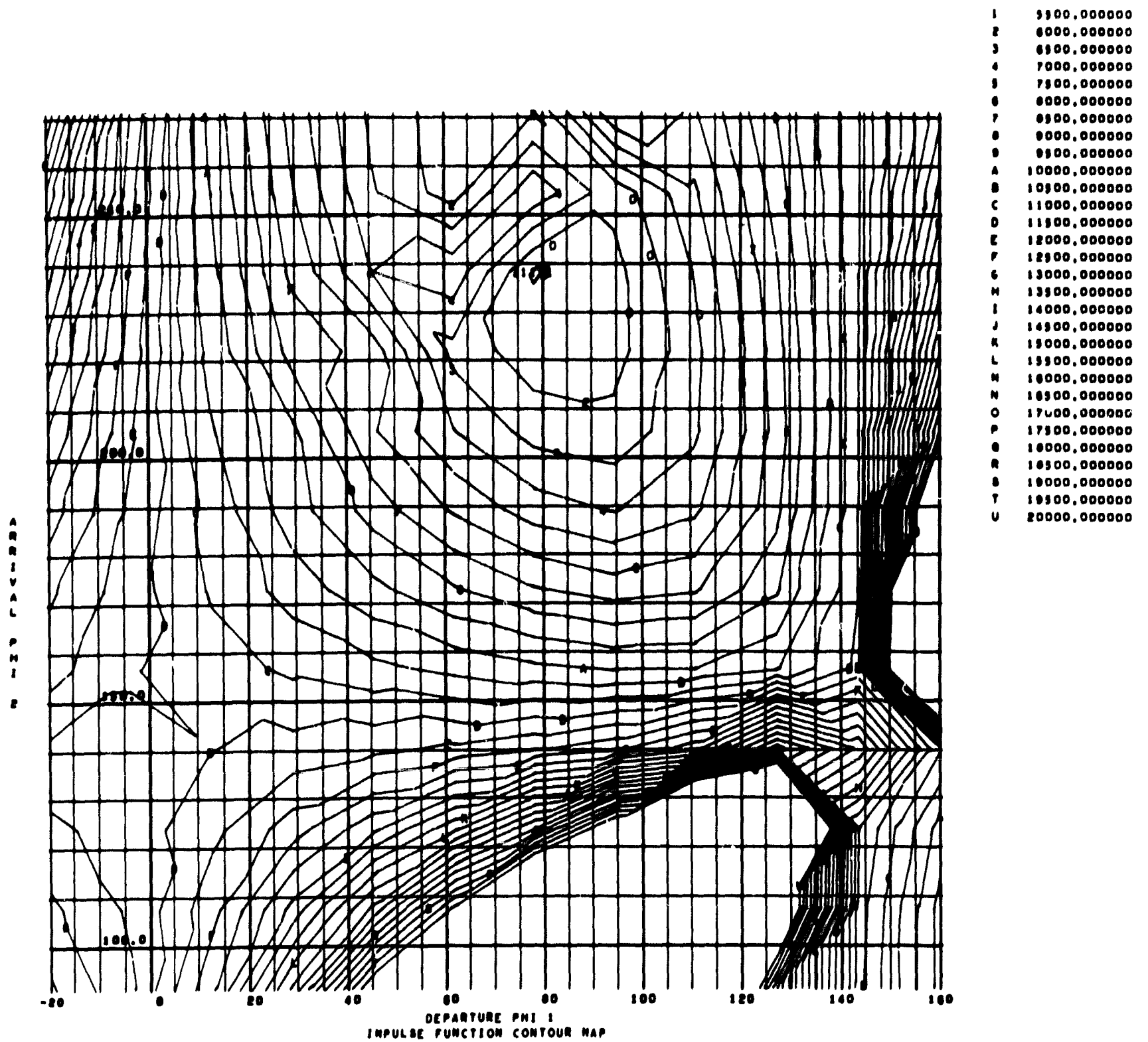


Figure 25.

REFERENCE	DATE	CHART	MAP
0.120000E 02	0.150000E 02	0.700000E 02	0.181100E 03
0.500000E 03			

LOCAL MINIMA SUSPECTED AT

PHI-1	PHI-2	P	IMPULSE	RANK
0.7618160 02	0.2388730 03	0.7566930 04	0.5445070 04	1

ADAPTIVE SPEED REGULATION OF TURBO-IMPULSE TRANSFER

PH1	PH2	P	Q1/QP1	Q1/QP2	Q1/QP	-----IMPUL SF-----	-----GAINS-----	ITNC
120.0000	220.0000	7.3576650	031201.251546	141.341185	-0.000108	7.8702012474832400	03 1.00 00 1.00 02	0 0
1112.0240	220.4913	7.3576650	03 590.670856	-42.539259	-19.579006	7.1606676252488250	03 1.00 00 8.50 -01 1.00 02	5 0
57.7566	226.8554	7.3576740	03 -5.890515	-31.871318	-0.059362	6.8144755181292400	03 1.00 00 1.00 00 1.00 02	10 0
101.5016	241.5385	7.3576560	03 60.032515	-55.501697	-50.658917	6.7427774663089300	03 1.00 00 1.60 00 1.90 02	15 0
50.0020	223.2395	7.4281450	03 438.092365	140.015424	-23.423422	6.23816328176114500	03 1.00 00 2.00 00 3.00 02	20 3
02.1420	242.6502	7.5500740	03 102.222376	67.232435	-21.827933	5.51422526007214500	03 1.00 00 2.00 00 2.00 02	25 3
03.7226	220.1553	7.5717500	03 33.201573	14.256791	-3.736435	5.453107169235452400	03 1.00 00 2.00 00 1.40 02	30 6
07.0850	229.1300	7.5749850	03 -5.876029	0.256924	-0.339900	5.456677119020431300	03 1.00 00 2.30 00 1.20 02	35 10
00.2820	229.0013	7.5764110	03 -0.442285	0.086913	0.024127	5.45667762621580500	03 1.00 00 2.30 00 1.00 02	40 15
00.2926	223.0005	7.5764350	03 -0.31459	0.070313	0.006737	5.456677341627271400	03 1.00 00 1.60 00 1.00 02	45 14
00.2032	227.0081	7.5764340	03 0.007922	-0.002346	-0.003572	5.45667940753537800	03 1.00 00 3.10 00 1.00 02	50 19
01.2932	227.5577	7.5764300	03 -0.001565	0.000297	-0.000214	5.45667940679126500	03 1.00 00 3.10 00 1.20 02	55 22
01.0524	227.0576	7.5764400	03 0.000153	-0.000003	0.000028	5.45667940677753400	03 1.00 00 3.10 00 1.20 02	60 27
01.2524	227.0576	7.5764400	03 -0.000119	0.000007	0.000037	5.45667940677745100	03 1.00 00 3.10 00 1.20 02	65 27

*** FINAL TRANSFER CONDITIONS ***		CENTER IS EARTH		* UNIT SYSTEM...MILES, FT/SEC	
X COMPONENT	Y COMPONENT	Z COMPONENT	RASTIME	RASTIME	RASTIME
-0.314153355R1D 04	0.77247716519C 03	0.24254165525C 03	0.32532915241D 04		
-0.1956602153D 04	0.91148618276D 03	0.4432124921C6D 03	0.2223375026D 04		
INITIAL ORBIT		TRANSFER ORBIT			
0.53000000000D 04	0.60000000000D 04	0.75764395071D 04			
0.41200000000D 00	0.22000000000D 00	0.12600000000D 00			
0.50000000000D 00	0.0	0.12908768556D 01			
0.0	0.0	0.57007645200D 02			
-0.05000000000D 02	0.30000000000D 02	-0.10033745636D 03			
0.555218009524D 04	0.42500000000D 04	0.76584806477D 04			
0.53105211361D 04	0.10039049504D 05	0.1724515615C 05			
0.8254551682D 04	0.72866900754D 04				
0.80233263074D 02	0.23739764530D 03	0.75764395071D 04			
ELAPSED EXECUTION TIME = 90.309					

Figure 26.

SPACE DIVISION OF NORTH AMERICAN ROCKWELL CORPORATION

	*** FINAL TRANSFER CONDITIONS ***			CENTER IS EARTH , UNIT SYSTEM...MILES, FT/SEC	
	X COMPONENT	Y COMPONENT	Z COMPONENT	MAGNITUDE	
IMPULSE					
FIRST VECTOR	-0.214133335810 04	0.772877169190 03	0.342941659850 03	0.829329132410 04	
SECOND VECTOR	-0.193669021590 04	0.911486182760 03	0.442124901060 03	0.820330788260 04	
TOTAL				0.945667940660 04	
	INITIAL ORBIT	FINAL ORBIT	TRANSFER ORBIT		
P	0.300000000000 04	0.600000000000 04	0.757643950710 04		
E	0.400000000000 00	0.200000000000 00	0.126000703840 00		
I (DEG)	0.300000000000 00	0.0	0.129887685960 01		
R A NODE (DEG)	0.0	0.0	0.970978483000 02		
ARG PER (DEG)	-0.900000000000 02	0.300000000000 02	-0.100387486360 03		
A	0.399238099240 04	0.623000000000 04	0.760000006470 04		
PERIOD	0.933098013630 04	0.100390495040 09	0.137245156750 09		
TRAN PY RAD	0.823455188820 04	0.728680079440 04			
PHI-1, PHI-2, P	0.802933690740 02	0.237997645300 03	0.757643950710 04		

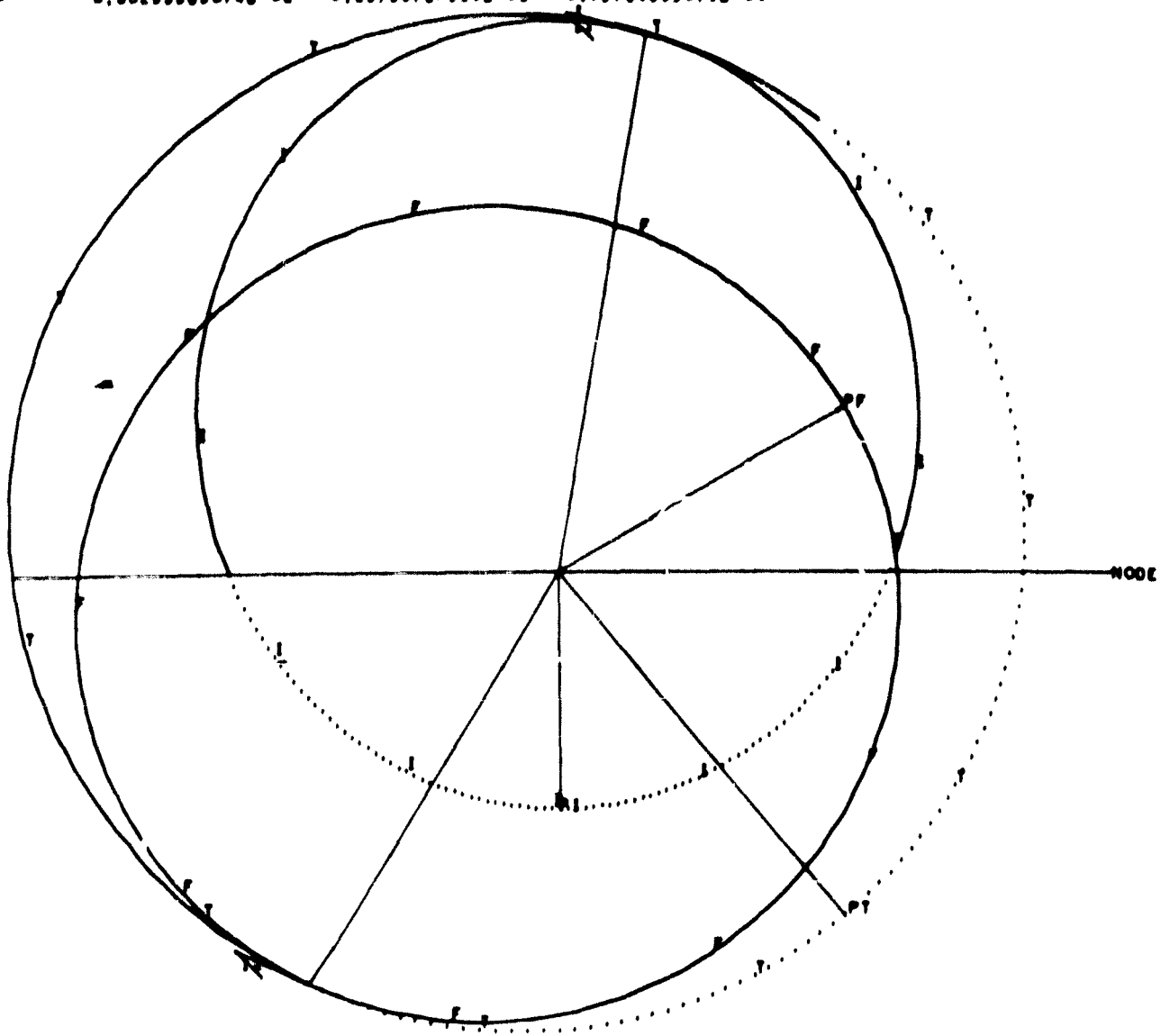


Figure 27.

APPENDIX III
COMPUTER PROGRAM LISTINGS

SD 69-3

Figure 28. MAIN Program Listing

```

C      DATA CONVR / 5280.00, 1000.00, 1.00, 1.00, 1.00 /
C      DATA G/ 95634.50100, 1176.23500, 3.17872790+10, 77787.100,
C      * 10293.700, 3.0348570+07, 398601.500, 4902.7500, 13.249480+10,
C      * 324230.00, 42906.00, 12.64980+07 /
C      DATA GR / .0060984800, .0.811776800, .009811776800, 1.00 /
C      GR IS WEIGHT/MASS OR ACCEL OF GRAV AT EARTH (BRITISH, METRIC, _TRIC)
C      G(I,J) ARE GXM IN (J=1) MI**3/SEC**2, OR (J=2) K**3/SEC**2
C      EARTH, MOON, SUN, VENUS, MARS, JUPITER
C      FOP I=1 I=2 I=3 I=4 I=5 I=6
C      EQUIVALENCE( DATA, XOPB1 )
C      *****
C      CONSTANTS
C      PI = 3.14159 26535 89793 24
C      RAD = 180.00/PI
C      FPSD = -1.
C      ZETEST = 1.0
C      KONTAU=3
C      FPSI = 1.E-05
C      CALL CLOCK( TIME )
C      CALL CDATEV( DATE )
C      CALL COUNTV
C      WRITE(6,10) DATE, TIME
C      10 FORMAT(1 DATE, A8, ' TIME =', F8.3 // )
C      TWO CARDS A, B ARE REQUIRED. ON CARD A THE TITLE AT USER'S
C      DISCRETION (72 COLUMNS).
C      THE ATTRACTING CENTER CAN BE EARTH, MOON, SUN, VENUS, MARS,
C      JUPITER, OR ANY BODY SPECIFIED (12 COLUMNS STARTING AT COLUMN 1,
C      CARD B)
C      IF BODY IS USED, GXM IS REQUIRED IN COLUMNS 25-36, CARD B.
C      THIS IS AN OPTIONAL OVERRIDE OTHERWISE.
C      THE UNIT SYSTEM CAN BE BRITISH (MILES, FT/SEC),
C      METRIC (KM, METERS/SEC), KMETRIC (KM, KM/SEC), OR BODY (STARTING
C      AT COLUMN 13 CARD B)
C      WHEN BODY UNITS ARE USED, THE ACCELERATION OF GRAVITY AT EARTH'S
C      SURFACE IS PUT = 1.00. THUS, XISP (ITEM 2, CARD 12) MUST BE
C      FUEL EXHAUST VELOCITY IN BODY SYSTEM UNITS.
C      TIME IS IN SECONDS FOR ALL SYSTEMS EXCEPT BODY SYSTEM UNITS.

```

Figure 28. MAIN Program Listing (Continued)

SPACE DIVISION of NORTH AMERICAN ROCKWELL CORPORATION

```

C      READ A TITLE AND UNIT DEFINITION CARD.
      READ (5,5) TITLE, BODY, UNSYS, GXM1
      5 FORMAT (19A4/ 5A4, D12.6)
      JJ = 2
      IF ( UNSYS(1) .EQ. TITEST( 1) ) JJ = 1
      IF ( UNSYS(1) .EQ. TITEST( 2) ) JJ = 2
      IF ( UNSYS(1) .EQ. TITEST( 3) ) JJ = 3
      IF ( UNSYS(1) .EQ. TITEST( 4) ) JJ = 4
      JJ = 2 IS CHOSEN IF NO MATCH IS FOUND ( CONV = 1000. )
      ON 8 I = 1,6
      II = 1
      IF ( BODY(1) .EQ. TITEST(1+4) ) GO TO 16
      3 CONTINUE
      IF NO MATCH IN NO 8 LOOP PROGRAM ASSUMES BODY WITH GXM = GXM1.
      GXM = GXM1
      JJ = 7
      IF ( GXM1 .NE. 0.00 ) GO TO 18
      WRITE (6,9)
      9 FORMAT ('PROGRAM UNABLE TO DETERMINE ATTRACTING CENTER, EXIT CALL
      *FD. ')
      WRITE (6,7) TITLE, BODY, UNSYS, GXM
      CALL EXIT
      16 JG = JJ
      IF ( JJ .EQ. 3 ) JG = 2
      GXM = G(II,JG)
      IF ( GXM1 .NE. 0.00 ) GXM = GXM1
      CONV = CONVR(JJ)
      GO = GR(JJ)
      XMM = GXM
      18 WRITE (6,7) TITLE, BODY, UNSYS, GXM
      7 FORMAT ('OCCASIDENTIFICATION' / 1H, 19A4 / 'BODY IS ', 3A4 /
      * 'UNIT SYSTEM IS ', 3A4 / ' GXM = ', D12.6 )
      INIT = 0
      NCARDS = 25
      CALL INPUT(
      MASK,INIT,NCARDS)
      CALL DATSET

```

Figure 28. MAIN Program Listing (Continued)

```

      INIT = 3
      100 WRITE(6,10) DATE , TIME
          CALL INPUT( MASK,INIT,NCARDS)
          WRITE(6,10) DATE , TIME
      6  FORMAT (6F16.6)
      105 CALL DEX( XORB1, XORB2,P1,F1,W1, X11,P2,E2,W2,PER1,PER2 )
          XII = XII * PAD
          W1 = W1 *PAD
          W2 = W2 *PAD
          WRITE(6,7002)
      7002 FORMAT(11,'O TRANSFORMED ELEMENTS',/ , ,T10,'P1',T30,'E1',T50,'W1',
          *T67,'INCL')
          WRITE(6,303) P1, E1, W1, X11, P2, E2, W2
      303 FORMAT (1H 4D19.11/ , ,T10,'P2',T30,'E2',T50,'W2',/1H 3D19.11)
          AEST4 = (P1 + P2) / 2.00
          XII = XII / PAD
          W1 = W1 /RAD
          W2 = W2 /RAD
          -----
      00 3000 JCARD = 1,5
      00 33  I1 = 1,10
      33  IRANK(I1) = 0
          NPT = SPAR(1,JCARD)
          IF ( NPT .EQ. 0 ) GO TO 3000
          WRITE (6,41)
      41  FORMAT ('1GENERATE CONTOUR MAP')
          WRITE (6,46) (SPAR(I,JCARD),I=1,6)
      46  FORMAT (6F16.6)
          CALL SURVEY ( Z, XORD, YORD, NPT, SPAR(2,JCARD), SPAR(3,JCARD),
          * SPAR(4,JCARD), SPAR(5,JCARD), AEST4 , SPAR(6,JCARD), 1 )
          00 2020  I = 1,10
          IF ( IRANK(I) .EQ. 0 ) GO TO 2040
          NNXY = NXV ( SNGL (XMIN(I) ) )
          NNYY = NYV ( SNGL (YMIN(I) ) )
          IXSD = 24
          00 2020  J = 1,5
          IXSP = IXSP - 8
      2020 CALL PLOTV ( (NNXY+IXSP), NNYY, IRANK(I) )
      2040 IF ( JCARD .GT. 1 ) GO TO 3000

```

Figure 28. MAIN Program Listing (Continued)

```

00 2000 IST = 1,10
IST1 = 11 - IST
IRKSV = 0
00 2050 IPK = 1,10
2050 IF ( IST1 .EQ. IRANK(IPK) ) IRKSV = IPK
IF ( IRKSV .EQ. 0 ) GO TO 2900
IF ( IST1.GT. IMAPS ) GO TO 2100
S2XY = SPAR(2,1) * SCXY
S3XY = SPAR(3,1) * SCXY
S6IMP = SPAR(6,1) * S6IMP
NPT1 = 15
CALL SURVEY ( 7, XPRD, YPRD, NPT1, S2XY, S3XY, XMIN( IRKSV ),
* YMIN( IRKSV ), PMIN( IRKSV ), S6IMP, 0 )
WRITE (16,2090) IRANK( IRKSV )
2090 FORMAT ( ' RANK OF OPTIMUM = ', I4 )
2100 IF( IST1 .GT. NSTEEP ) GO TO 2900
A(1) = XMIN( IRKSV )
A(2) = YMIN( IRKSV )
A(3) = PMIN( IRKSV )
CALL STEEP ( -A , IRKSV )
WRITE ( 6,2090) IRANK( IRKSV )
WRITE (16,2090) IRANK( IRKSV )
2000 CONTINUE
2900 CONTINUE
IF ( IUSE .EQ. 0 ) CALL STEEP( AFST , 1 )
-----
X11 = X11 * RAD
W1 = W1 * RAD
W2 = W2 * RAD
CALL TIMEV( TIME1 )
WRITE(6,301) TIME1
301 FORMAT('OELAPSED EXECUTION TIME = ',F12.3 )
CALL COUNTV
GO TO 100
END

```

Figure 28. MAIN Program Listing (Continued)

```

SUBROUTINE DATSET
-----
      THIS SUBROUTINE SUPPLIES REASONABLE VALUES FOR ALL PARAMETERS
      REQUIRED TO RUN A COMPLETE CASE.
-----
      IMPLICIT REAL*(A-H, O-Z)
      LOGICAL RATES, PEINT, CRYOUT
      REAL*4 RESET, TWORRN, SCXY, SCIMP, SPAR, PAD
      COMMON /INDATA/ XORR1(6), XORB2(6), AEST(6)
      A, AMPINC , AMPDEC , FREEZE , XAMP , SAVE , BL46
      B, SCLINC , SCLDEC , ISCL , IAMP , BL55 , BL55
      C, SCALE(3) , , TOL , MAXIT , IPRIN
      D, XIN(14), XFN(14), RATES, INIMFN
      I, BETASV , XISP , TEPRT , PERT , COAST1 , BL126
      J, BRNINC , RMAX1 , BMAX2 , BSH1 , BSH2 , TWORRN
      K, RFSET , KMHIT , NDRPT , PRNJ , BL145 , BL146
      L, INTSTD , PEINT , CRYOUT , ITSQL , IQPRIN , IQCYCL
      M, HRAPTL , PHOS1 , BL163 , BL164 , BL165 , BL166
      N, XNR , XMC , QTR , DTC , JMAX , IBL176
      O , PAD(12)
      Q , IMAPS , NSTEEP , IUSE, SCXY , SCIMP , IPAD3
      P , SPAR(4,5)
-----
      CARD 3 AEST(3) = 5000.00
      CARD 4 AMPINC = 2.00
      AMPDEC = .500
      FREEZE = 1.0-08
      XAMP = 2.00
      SAVE = 2.00
      BL46 = 1.00
      CARD 5 SCLINC = 0.00
      SCLDEC = .8500
      ISCL = 1
      IAMP = 5
      BL55 = 0.00
      BL56 = 0.00

```

Figure 29. Subroutine DATSET Listing

```
C      CARD 6
      SCALE(1) = 1.00
      SCALE(2) = 1.00
      SCALE(3) = 100.00
      TOL      = 1.0-05
      MAXIT    = 200
      IPBIN    = 5
      -----
C      CARDS 7-19 ARE REQUIRED BY THE FINITE THRUST OPTIMIZATION PROGRAM
C      -----
C      CARD 20
      IMADS    = 4
      NSTEEP   = 5
      IUSE     = 0
      SCXY     = .2
      SCIMP    = .1
      IPAD3    = 0
C      CARD 21
      SPAD(1,1) = 24.
      SPAR(2,1) = 15.
      SPAR(3,1) = 15.
      SPAD(4,1) = 180.5
      SPAR(5,1) = 181.1
      SPAR(6,1) = 500.0
C      CARD 22
      SPAR(1,2) = 0.0
      RETURN
      END
```

DATS0390
DATS0400
DATS0410
DATS0420
DATS0430
DATS0440
DATS0450
-----DATS0460
DATS0470
DATS0480

DATS0490
DATS0500
DATS0510
DATS0520
DATS0530
DATS0540
DATS0550
DATS0560
DATS0570
DATS0580
DATS0590
DATS0600
DATS0610
DATS0620
DATS0630
DATS0640
DATS0650
DATS0660

Figure 29. Subroutine DATSET Listing (Continued)

```

SUBROUTINE INPUT(      MASK,INIT,NCARDS )      INPT0000
-----INPT0005
C      READS AND PRINTS INITIAL ORBIT GEOMETRY AND OTHER DATA PARAMETERS  INPT0010
C      TO BE USED IN IMPULSE COMPUTATIONS.  READS ONLY THOSE PARAMETERS  INPT0015
C      THAT ARE SPECIFIED ON DATA CARDS.  INPT0020
C      -----INPT0025
C      REAL*8      DBL      INPT0030
C      LOGICAL      LN      INPT0040
C      DIMENSION  IN( 1 ) , AL( 1 ) ,      DBL( 1 ) , LO( 1 )      INPT0050
C      COMMON /INDATA/ DATA(300), XPOS(6,25)      INPT0060
C      DIMENSION MASK(6,25)      INPT0070
C      A , ALF(3,6), RUJF(3) , FORM(7) , DFORM(20),SPACE( 6)      INPT0080
C      DATA BLANK /' ' /      INPT0090
C      DATA FORM( 7) /' ' / , FFORM(11) /'(1H+* /      INPT0100
C      DATA SPACE /',13X,30X,47X,64X,81X,98X* /      INPT0110
C      DATA DFORM /',1P1E17.6      , 1P1D17.8      ,117      INPT0120
C      A 3A4      ,L17      ' /      INPT0130
C      EQUIVALENCE( DATA , IN , AL , LO , DBL )      INPT0140
C      -----INPT0150
C      INIT = 0 --- MASK ASSUMED DEFINED IN CALLING PROG, SET DATA = 0. INPT0160
C      INIT = 1 --- SET MASK = 1., READ MASK, SET DATA = 0. INPT0170
C      INIT = 2 --- SET MASK = 1., READ MASK      INPT0180
C      INIT = 3 --- READ DATA CARDS      INPT0190
C      -----INPT0200
C      IF( INIT .EQ. 3 ) GO TO 100      INPT0210
C      IF( INIT .EQ. 0 ) GO TO 49      INPT0220
C      FILL DATA MASK WITH ONES AT START      INPT0230
C      DO 10 I=1,6      INPT0240
C      DO 10 J=1,25      INPT0250
C      MASK(I,J)=1      INPT0260
C      READ(5,48) ((MASK(I,J),I=1,6),J=1,NCARDS )      INPT0270
C      48 FORMAT(7?11 )      INPT0280
C      49 IF( INIT .EQ. 2 ) GO TO 55      INPT0290
C      DO 50 I=1,300      INPT0300
C      DATA(I) = 0.0      INPT0310

```

Figure 30. Subroutine INPUT Listing

```

C      SET UP DATA ARRAY INDEXING PARAMETERS
55  NUM = 1
   DO 90 J=1,NCARDS
   DO 90 K=1,6
   MPCS(K,J) = NUM
   IS = MASK(K,J)
   GO TO(60,70,80,90), IS
   WRITE(6,50) K,J,IS
59  FORMAT(20H MASK ERROR
      3112      )
   MASK(K,J) = 2
   GO TO 70
60  NUM = NUM + 1
   GO TO 90
70  NUM = NUM + 2
   GO TO 90
80  NUM = NUM + 3
90  CONTINUE
   IF ( INIT.EQ. 0 ) RETURN
   WRITE(6,91)
   DO1 FORMAT(40H DATA MASK
   DO2 FORMAT( 8I12 )
   DO 93 JD=1,NCARDS
   DO3 WRITE (6,92) JD, (MASK(I,JD), I=1,6), JD
   RETURN
C      READ DATA CARDS
100  WRITE(6,103)
103  FORMAT(40H DATA CARD IMAGES
105  READ(5,110) ((ALF(I,J),I=1,3),J=1,6),IX,ID
110  FORMAT(18A4,5X,I1,I2 )
   WRITE(6,111)((ALF(I,J),I=1,3),J=1,6),IX,ID
111  FORMAT( 2X,6(4X,3A4),2I4 )
INPT0320
INPT0330
INPT0340
INPT0350
INPT0360
INPT0370
INPT0380
INPT0390
INPT0400
INPT0410
INPT0420
INPT0430
INPT0440
INPT0450
INPT0460
INPT0470
INPT0480
INPT0485
INPT0490
INPT0500
INPT0510
INPT0520
INPT0530
INPT0540
INPT0550
INPT0560
INPT0570
INPT0580
INPT0590
INPT0600
INPT0610

```

Figure 30. Subroutine INPUT Listing (Continued)

```

C
      DO 150 J = 1,6
      CHECK FOR PLANKS
      DO 120 I = 1,3
      IF (PLANK .NE. ALF(I,J)) GO TO 125
      120 CONTINUE
      GO TO 150
      125 CALL INFILQ( BUF , 12 )
      WRITE(99,126) ( ALF(I,J), I=1,3 )
      126 FORMAT(3A4)
      CALL INFILQ( BUF , 12 )
      NGN = MASK(J,IN)
      MPN = MPNS(J,IN)
      GO TO (140,142,144,146,148),NGN
      GO TO 105
      140 READ(99,141) DATA(MPN)
      141 FORMAT(1E12.0)
      GO TO 150
      142 MPI = (MPN+1)/2
      READ(99,143) ORL(MPI)
      143 FORMAT(1D12.0)
      GO TO 150
      144 READ(99,145) IN(MPN)
      145 FORMAT( 112 )
      GO TO 150
      146 READ(99,147)(AL(MPN-1+K),K=1,3)
      147 FORMAT( 3A4 )
      GO TO 150
      148 READ(99,149) LG(MPN)
      149 FORMAT( 112 )
      150 CONTINUE
      IF (IX .EQ. 0 ) GO TO 105

```

```

INPT0620
INPT0630
INPT0640
INPT0650
INPT0660
INPT0670
INPT0680
INPT0690
INPT0700
INPT0710
INPT0720
INPT0730
INPT0740
INPT0750
INPT0760
INPT0770
INPT0780
INPT0790
INPT0800
INPT0810
INPT0820
INPT0830
INPT0840
INPT0850
INPT0860
INPT0870
INPT0880
INPT0890
INPT0900
INPT0910
INPT0920

```

Figure 30. Subroutine INPUT Listing (Continued)

SPACE DIVISION OF NORTH AMERICAN ROCKWELL CORPORATION

```

C      WRITE ENTIRE DATA FIELD IN PROPER FORMAT
      WRITE(6,200)
200  FORMAT(40H1 DATA ARRAY IMAGE
      DO 250 I=1,NCARDS
      WRITE(6,201) I
201  FORMAT(' CARD NO.=',I3)
      DO 250 J=1,6
      FORM(2) = SPACE(J)
      NF = (MASK(J,I) - 1) * 4
      DO 230 K=1,4
      FORM( K + 2 ) = FORM(NF + K)
      NG = MASK(I,I)
      MP = MODS(J,I)
      GO TO (441,442,443,444),NG
441  WRITE(6,FORM) DATA(MP)
      GO TO 250
442  MP1 = (MP+1) / 2
      WRITE(6,FORM) MP1
      GO TO 250
443  WRITE(6,FORM)DATA(MP),DATA(MP+1),DATA(MP+2)
250  CONTINUE
      RETURN
      END
INPT0930
INPT0940
)INPT0950
INPT0960
INPT0970
INPT0980
INPT0990
INPT1000
INPT1010
INPT1020
INPT1030
INPT1040
INPT1050
INPT1060
INPT1070
INPT1080
INPT1090
INPT1100
INPT1110
INPT1120
INPT1130
INPT1140
INPT1150
```

Figure 30. Subroutine INPUT Listing (Continued)

```

C      SUBROUTINE DEX( XORB1, XORB2,P1,E1,W1,X11,P2,E2,W2,PER1,PER2 )
C      -----
C      COMPUTES ORBIT GEOMETRY INFORMATION TO BE USED IN IMPULSE
C      COMPUTATIONS. PERFORMS COORDINATE TRANSFORMATIONS.
C      -----
C      IMPLICIT REAL*8(A-H,O-Z)
C      DIMENSION Q(3,3,2),XN(3),XM(3),XI(6,2),XORB1(6),XORB2(6)
C      COMMON/ZCONST/ PI,RAD,GXM
C      DO 10 J=1,6
C      XI(J,1) = XORB1(J)
C      XI(J,2) = XORB2(J)
C      IF( XI(3,1) ) 15,14,15
C      IF( XI(3,2) ) 15,900,15
C      DO 100 J = 1,2
C      XA1 = DSIN( XI(4,J) / RAD )
C      XA2 = DCOS( XI(4,J) / RAD )
C      XB1 = DSIN( XI(5,J) / RAD )
C      XB2 = DCOS( XI(5,J) / RAD )
C      XC1 = DSIN( XI(3,J) / RAD )
C      XC2 = DCOS( XI(3,J) / RAD )
C      Q(1,1,J) = XB2*XA2 - XB1*XA1*XC2
C      Q(2,1,J) = XB2*XA1 + XB1*XA2*XC2
C      Q(3,1,J) = XB1*XC1
C      Q(1,2,J) = -XB1*XA2 - XB2*XA1*XC2
C      Q(2,2,J) = -XB1*XA1 + XB2*XA2*XC2
C      Q(3,2,J) = XB2*XC1
C      Q(1,3,J) = XA1*XC1
C      Q(2,3,J) = -XA2*XC1
C      Q(3,3,J) = XC2
100 Q(3,3,J) = XC2

```

Figure 31. Subroutine DEX Listing

```

X11 = DBLDT( Q(1,3,1) , Q(1,3,2) , 3 )
X11 = DATAN( DSQRT( 1.00 - X11**2) / X11 )
CALL DBLCRS( Q(1,3,2) , Q(1,3,1) , XN )
CALL DBLUNT( XN, XN, 3 )
CALL DBLCRS( Q(1,3,1) , XN , XM )
PX = DBLDT( XN , Q(1,1,1) , 3 )
PY = DBLDT( XM , Q(1,1,1) , 3 )
W1 = DATAN2( PY , PX )
CALL DBLCRS( Q(1,3,2) , XN , XM )
PX = DBLDT( XN , Q(1,1,2) , 3 )
PY = DBLDT( XM , Q(1,1,2) , 3 )
W2 = DATAN2( PY , PX )
GO TO 950
000 X11 = 0.00
W1 = X1(5,1) / RAD
W2 = X1(5,2) / RAD
950 E2 = XNRB2(2)
F1 = XNRB1(2)
P1 = XNRB1(1)
P2 = XNRB2(1)
XNRB1(6) = P1/ (1.00 - F1**2)
XNRB2(6) = P2/ (1.00 - E2**2)
1000 RETURN
END

```

DEX 1155
 DEX 1160
 DEX 1165
 DEX 1170
 DEX 1175
 DEX 1180
 DEX 1185
 DEX 1190
 DEX 1195
 DEX 1200
 DEX 1205
 DEX 1210
 DEX 1215
 DEX 1220
 DEX 1225
 DEX 1230
 DEX 1235
 DEX 1240
 DEX 1245
 DEX 1250
 DEX 1255
 DEX 1260
 DEX 1265
 DEX 1270

Figure 31. Subroutine DEX Listing (Continued)

SD 69-3

Figure 32. Subroutine SURVEY Listing

```

C      SET UP GRID PARAMETERS
      DO 2 I = 1,15
      2 A7(I) = 0.1
      DO 3 I = 1,5
      XLAB(I) = XLA(I)
      3 YLAB(I) = YLA(I)
      DO 4 I = 1,9
      4 BL(I) = XBL(I)
      ZCUT = 100000.
      ZM = 2000.
      NDC = 30
      ZMX = 100000.
      JN = 0
      IF(SURV3) 10, 5, 10
      5 SURV3 = SURV2
      10 SURV1 = NPT
      DXQ = SURV1 / ( SURV1 - 1. )
      DX1 = DXQ * SURV2
      DX2 = DXQ * SURV3
      XSURV = SURV1 * SURV2 / 2.
      XSURV2 = SURV1 * SURV3 / 2.
      A1Q = AEST1 - XSURV
      A2Q = AEST2 - XSURV2
      XL = A1Q
      YB = A2Q
      XR = A1Q + 2. * XSURV
      YT = A2Q + 2. * XSURV2
      XORD(1) = XL
      YORD(1) = YB
      C      CALCULATE GRID COORDINATES
      DO 100 L = 2,NPT
      XORD(L) = XORD(L-1) + DX1
      YORD(L) = YORD(L-1) + DX2
      100 PLOT GRID FOR POINT IDENTIFICATION
      C      CALL CAMRAV(9)

```

Figure 32. Subroutine SURVEY Listing (Continued)

```

C      GENERATE FUNCTION
DO 9000 I = 1,NPT
DO 9000 J = 1,NPT
C      FILL --Z-- ARRAY
A(1) = XORD(J)
A(2) = YORD(I)
A(3) = AEST3
CALL IMP2(A,GRAD,1,3)
Z(J,I,1) = DVT(3)
9000 Z(J,I,2) = DVT(1)
      CALL CONTOUR ( NPT, NPT, XORD, YORD, Z(1,1,1), NOC, ZM, ZMX,
*      ZCUT, DELZ, 6., 6., -1, 1, 1, -1, 2 )
      NPT = NPT - 1
      IF ( IPANKS .EQ. 0 ) RETURN
      IMIN = 1
DO 500 I = 2,NPT
DO 500 J = 2,NPT
      IF ( Z(I,J,1) .GT. Z(I-1,J-1,1) ) GO TO 500
      IF ( Z(I,J,1) .GT. Z(I-1,J,1) ) GO TO 500
      IF ( Z(I,J,1) .GT. Z(I-1,J+1,1) ) GO TO 500
      IF ( Z(I,J,1) .GT. Z(I,J-1,1) ) GO TO 500
      IF ( Z(I,J,1) .GT. Z(I,J+1,1) ) GO TO 500
      IF ( Z(I,J,1) .GT. Z(I+1,J-1,1) ) GO TO 500
      IF ( Z(I,J,1) .GT. Z(I+1,J,1) ) GO TO 500
      IF ( Z(I,J,1) .GT. Z(I+1,J+1,1) ) GO TO 500
      LOCAL MINIMUM FOUND
      7MINS(IMIN) = Z(I,J,1)
      XMINS(IMIN) = XORD(I)
      YMINS(IMIN) = YORD(J)
      PMINS(IMIN) = Z(I,J,2)
      IF ( IMIN .LT. 10 ) IMIN = IMIN + 1
500 CONTINUE
      MIN = IMIN - 1
      IF ( MIN .NE. 0 ) GO TO 540
      WRITE (6,520)
520 FORMAT ('ONO MINIMA FOUND')
      RETURN

```

SURV0630
 SURV0640
 SURV0650
 SURV0660
 SURV0670
 SURV0680
 SURV0690
 SURV0700
 SURV0710
 SURV0720
 SURV0730
 SURV0740
 SURV0750
 SURV0760
 SURV0770
 SURV0780
 SURV0790
 SURV0800
 SURV0810
 SURV0820
 SURV0830
 SURV0840
 SURV0850
 SURV0860
 SURV0870
 SURV0880
 SURV0890
 SURV0900
 SURV0910
 SURV0920
 SURV0930
 SURV0940
 SURV0950
 SURV0960
 SURV0970
 SURV0980
 SURV0990

Figure 32. Subroutine SURVEY Listing (Continued)

```
540  DO 552  I = 1,MIN
      DO 552  J = 1,MIN
552  ZMN(I,J) = ZMINS(J)
      DO 590  I = 1,MIN
      II = I + 1
      L = 1
      DO 555  J = 1,MIN
      JJ = J + 1
      IF ( JJ .GT. MIN ) JJ = MIN
      IF ( ZMN(I,JJ) .LT. ZMN(I,L) ) L = JJ
555  ZMN(II,J) = ZMN(I,J)
      ZOUT(I,L) = ZMN(I,L)
      IRANK(L) = I
590  ZMN(II,L) = ZMN(II,L) * 1.E+20
      WRITE (6,592)
582  FORMAT ('LOCAL MINIMA SUSPECTED AT' //
*      *      P      IMPULSE      RANK      ' )
*      *      PHI-1      PHI-2
      DO 585  I = 1,MIN
585  WRITE (6,595) XMINS(I), YMINS(I), PMINS(I), ZMINS(I), IRANK(I)
595  FORMAT (4D16.6, I5)
      RETURN
      END
```

Figure 32. Subroutine SURVEY Listing (Continued)

```

C      SUBROUTINE STEEP ( A , IRKSV )
C      -----
C      EXECUTIVE SUBROUTINE FOR CONDUCTING AN ADAPTIVE STEEP DESCENT
C      SEARCH FOR A LOCAL OPTIMUM.
C      -----
C      IMPLICIT REAL*8(A-H,O-Z)
C      REAL*4 PAD, SPAR, SCXY, SCIMP, S2XY, S3XY, S6IMP
C      DIMENSION A(3), XGRAD(3), VN(3), XB(3), GRAD(3), STEP1(3), GSAV
C      1(3), SX(3), UTAM(3), SSV(3)
C      DIMENSION THX1(2,3), THX2(2,3)
C      DIMENSION SRAD1(3), SPAD2(3)
C      DIMENSION FORM(7), FRM2(8)
C      INPUT DATA, COMMON BLOCK
C      COMMON /INDATA/ XORR1(6), XORB2(6), AEST(6)
C      A, AMPINC , AMPDEC , FREEZE , XAMP , SAVE , BL46
C      B, SCLINC , SCLDEC , ISCL , IAMP , RL55 , BL56
C      C, SCALE(3) , TOL , MAXIT , IPRIN
C      COMMON/ZCONST/ PI,RAD,GXM,RETA,CONV,II,JJ
C      COMMON /ORRCON/ PI,EI,WI, XII,P2,F2,W2,P,E,W,ETRN(6)
C      COMMON /STCOM/ Q1(3), Q2(3), XE(3), XU(3), XV(3), AUX(3), C
C      1XI,DVT(5),RIE,R2E,SXI,TAM(3),UN1(3),UN2(3),VE1(3),VE2(3)
C      2),XL1,XL2,XS1,XS2,PERT,PER1,PER2,RAD1(3),RAD2(3),UMAG,XIM1(3,2),XI
C      3M2(3,2),DTHETA,FPSI,ZETEST,
C      4JN , LOS ,KONTAU, IBOX
C      COMMON /OPTS/ ZMINS(10), XMINS(10), YMINS(10), PMINS(10),IRANK(10),
C      DATA FORM// EARTH MOON SUN VENUS MARS JUPITER X BODY
C      * , /
C      DATA FRM2 /MILFS, FT/SEC KM, METERS/SEC KM, KM/SEC BODY
C      *NIT SYSTEM* /
C      -----
C      WRITE (6,2210)
C      2210 FORMAT('ADAPTIVE STEEP DESCENT OPTIMIZATION OF TWO-IMPULSE TRANSF
C      *ER')
C      2221 WRITE (6,2222)
C      2222 FORMAT(1H0,3X,4HPHI1,6X,4HPHI2,AX,1HP,9X,6HDI/DP1,5X,6HDI/DP2,
C      * 5X,5HDI/DP,7X,15H----IMPULSF----;6X,19H-----GAINS-----,
C      * 3X,4HITNO)

```

Figure 33. Subroutine STEEP Listing

```

49 SX(1) = 5.00
   SX(2) = 5.00
   SX(3) = 100.00
   DVTI=1.00+20
104 DO 50 I=1,3
50 XR(I) = SCALF(I)
   AMP = XAMP
   JN=0
   ION=0
   IFPO=0
   IFAIL=0
   JG = 1
C SOLUTION VECTOR
C INITIAL P-OPTIMIZATION
   IGRAD = 2
   IF(SCALF(3)) 222, 223, 222
222 IGRAD = 1
223 CALL IMP2 (1, GRAD, IGRAD, 2)
   DO 225 I=1,3
   G = GRAD(I)
225 SSV(I) = DSIGN(1.00, G)
   A(3)=P
C -----
C START ITERATION LOOP
   ITNX = 0
   WRITE (6,500)A, GRAD, DVT(3), XR, ITNX, IFAIL
500 FORMAT (2F9.4, 10D13.6, 3F11.6, 023.16, 3E8.1, I4, I3)
   DO 1000 ITNI = 1, MAXIT
227 JAMP=1
   CALL IMP2 (1, GRAD, 2, 2)
   DO 6000 I=1,3
   AUX(I) = A(I)
6000 XGRAD(I) = GRAD(I)
   DO 231 I=1,3
   IF( DARS( GRAD(I)) - TOL ) 231, 231, 232
231 CONTINUE
   GO TO 950

```

STEF0370
STEE0380
STEE0390
STEE0400
STEE0410
STEE0420
STEE0430
STEE0440
STEE0450
STEE0460
STEF0470
STEE0480
STEE0490
STEF0500
STEE0510
STEE0520
STEE0530
STEF0540
STEE0550
STEE0560
STEE0570
STEE0580
STEF0590
STEE0600
STEE0610
STEE0620
STEE0630
STEE0640
STEE0650
STEE0660
STEE0670
STEE0680
STEE0690
STEE0700
STEE0710
STEE0720
STEE0730

Figure 33. Subroutine STEEP Listing (Continued)

```

232 IF( DVT(3) - DVTL ) 270, 260 , 260
260 AMP=AMP*AMPDEC
    IERR = 0
    DO 262 I=1,3
    A(I)=A(I)-STEP1(I)
262 GRAD(I)=GSAV(I)
    JAMP=2
    IFAIL=IFAIL+1
    IF( AMP - FREE7E )950,950,401
270 IERR = IERR + 1
    IF( IERR-IAMP )279,271,271
271 IERP = 0
    AMP=AMP*AMPINC
279 DO 330 I=1,3
    IF( XB(I) )309,330,309
309 IF( SSV(I)*GRAD(I) )310,310,330
310 XB(I) = XB(I) * SCLDEC
330 CONTINUE
399 BMAG=XB(ISC1)
    IF (BMAG .EQ. 0.) BMAG = 1.
    DO 3990 I=1,3
3990 XB(I)=XB(I)/BMAG
    XG=DBLUNT (GRAD,GRAD,3)
    DO 400 I=1,3
    G = GRAD(I)
400 SSV(I) = DSIGN(1.00 , G )
401 DO 603 I =1,3
    STEP = - GRAD(I)* XB(I) * AMP
    IF( DABS( STEP ) - SX(I) )602 ,602 , 601
601 G = SX(I)
    STEP = DSIGN( G , STEP )
602 A(I)=A(I)+STEP
    SX(I)=DABS( STEP*SAVE )
    GSAV(I)=GRAD(I)
    STEP1(I)=STEP
603 CONTINUE

```

STEE0740
STEE0750
STEE0760
STEE0770
STEE0780
STEE0790
STEE0800
STEE0810
STEE0820
STEE0830
STEE0840
STEE0850
STEE0860
STEE0870
STEE0880
STEE0890
STEE0900
STEE0910
STEE0920
STEE0930
STEE0940
STEE0950
STEE0960
STEE0970
STEE0980
STEE0990
STEE1000
STEE1010
STEE1020
STEE1030
STEE1040
STEE1050
STEE1060
STEE1070
STEE1080
STEE1090

Figure 33. Subroutine STEEP Listing (Continued)

```

          IF (JAMP.EQ.2) GO TO 227
          DVTL=DVT(3)
          IF (MOD ( ITNO , IPRIN ) ) 1000,1001,1000
1001 WRITE (6,500) AUX , XGRAD ,DVT(3) , XB , ITNO , IFAIL
          NNXY = NXV( SNGL( AUX(1) ) )
          NNZY = NYV( SNGL( AUX(2) ) )
          CALL PLOTV( NNXY , NNZY , IRANK(IRKSV) )
1000 CONTINUE
          WRITE (6,500) AUX , XGRAD ,DVT(3) , XB , ITNO , IFAIL
          WRITE (6,3019) AMP, IFAIL
3019 FORMAT(2X,4HAMP=,F16.8,4X,6HIFAIL= ,I10)
          GO TO (410,404), LOS
404 DO 405 K=1,3
405 TAM(K)=-TAM(K)
          DTHTETA = 2.00 * PI - DTHTETA
410 UMAG = DBLUNT( TAM ,UTAM , 3 )
929 ETRN(1) = P
          ETRN(6) = P / (1.00 - F**2 )
          ETRN( 2 ) = E
          IF(X11) 935 , 930 , 935
930 ETRN(3) = 0.00
          ETRN(4) = 0.00
          ETRN(5)=DATAN2(XE(2),XE(1))
          ETRN(5)=ETR(5)*RAD
          W = ETRN(5)
          GO TO 936
          ETRN(3) = DATAN2( DSQRT(1.00 - UTAM(3)**2 ) , UTAM(3) ) * RAD
          CALL DBLCRS ( Q2 , UTAM , VN )
          DUMMY=DBLUNT(VN,VN,3)
          ETRN(4) = DATAN2( VN(2) , VN(1) ) * RAD
          CALL DBLCRS( UTAM , VN , XU )
          ETRN(5) = RAD * DATAN2(DBLUNT(XU,XE,3) , DBLUNT(VN,XE,3) )
          W = ETRN(5)
936 DO 7010 I=1,3
          DO 7010 J=1,2
          XIM1(I,J) = XIM1(I,J) * CONV
          XIM2(I,J) = XIM2(I,J) * CONV

```

```

STEEL1100
STEEL1110
STEEL1120
STEEL1130
STEEL1140
STEEL1150
STEEL1160
STEEL1170
STEEL1180
STEEL1190
STEEL1200
STEEL1210
STEEL1220
STEEL1230
STEEL1240
STEEL1250
STEEL1260
STEEL1270
STEEL1280
STEEL1290
STEEL1300
STEEL1310
STEEL1320
STEEL1330
STEEL1340
STEEL1350
STEEL1360
STEEL1370
STEEL1380
STEEL1390
STEEL1400
STEEL1410
STEEL1420
STEEL1430
STEEL1440
STEEL1450
STEEL1460

```

Figure 33. Subroutine STEEP Listing (Continued)

```

XSI = XSI * CONV
XS2 = XS2 * CONV
XLI = XLI * CONV
XL2 = XL2 * CONV
PIG = 2.00 * PI / DSQRT( GXM )
PERT= PIG*DSQRT (ETRN(6)
      **3)
AI= P1/(1.-EI**2)
A2= P2/(1.-E2**2)
ZER0=0.0 DO
IF ( RL46 .EQ. 0.00 ) GO TO 671
DO 640 J=1,3
THX1(1,J) = XIM1(J,LOS)
640 THX2(1,J) = XIM2(J,LOS)
ET3 = ETRN(3) / RAD
ET4 = ETRN(4) / RAD
ET5 = ETRN(5) / RAD
DO 645 I=1,3
SRAD1(I) = SNGL( RAD1(I) )
SRAD2(I) = SNGL( RAD2(I) )
645 CALL ELIPSE (P1, XI1, W1, P2, ZER0, W2, P, ET3, ET5, ET4, THX1,
* THX2, EI, E, E2, SRAD1, SRAD2 )
671 XI1 = XI1 * RAD
W1 = W1 *RAD
W2 = W2 *RAD
CALL SCOUTV
CALL SCSETV( 1,14)
TOTAL1 = XSI + XS2
TOTAL2 = XLI + XL2
J = 2 * (JJ - 1 ) + 1
WRITE (6,504) FORM(11), FRM2(J), FRM2(J+1)
WRITE(16,504) FORM(11), FRM2(J), FRM2(J+1)
504 FORMAT('0',22X,'*** FINAL TRANSFER CONDITIONS ***', 10X,'CENTER IS
* , AB, ', ', 'UNIT SYSTEM...', 2A8 )
IF ( LOS .EQ. 1 ) WRITE ( 6,505) (XIM1(I,1),I=1,3), XSI,(XIM2(I,1),1)STEEL800
*,I=1,3), XS2, TOTAL1, P1, P2, P, EI, E2, XI1, ZFRO, ETRN(3), ZERSTEEL1810
*0, ZER0, ETRN(4), W1, W2, ETRN(5), A1, A2, ETRN(6), PER1, PER2, PESTEEL1820
*RT, RIF, P2E , AUX

```

Figure 33. Subroutine STEEP Listing (Continued)

```

IF ( LOS .EQ. 1 ) WRITE (16,505) (XIM1(I,1),I=1,3), XS1,(XIM2(I,1))STEE1840
*,I=1,3), XS2, TOTAL1, P1, P2, P, E1, E2, E,XI1, ZERO, ETRN(3), ZERSTEE1850
*,0, ZERO, FTRN(4), W1, W2, FTRN(5), A1, A2, ETRN(6), PER1, PER2, PESTEE1860
*,RT, R1E, R2E , AUX STEE1870
IF ( LOS .NE. 1 ) WRITE ( 6,505) (XIM1(I,1),I=1,3), XL1,(XIM2(I,1))STEE1880
*,I=1,3), XL2, TOTAL2, P1, P2, P, F1, E2, E,XI1, ZERO, ETRN(3), ZERSTEE1890
*,0, ZERO, FTRN(4), W1, W2, FTRN(5), A1, A2, ETRN(6), PER1, PER2, PESTEE1900
*,RT, R1E, R2E , AUX STFF1910
IF ( LOS .NE. 1 ) WRITE (16,505) (XIM1(I,1),I=1,3), XL1,(XIM2(I,1))STEE1920
*,I=1,3), XL2, TOTAL2, P1, P2, P, E1, E2, E,XI1, ZERO, ETRN(3), ZERSTEE1930
*,0, ZERO, FTRN(4), W1, W2, FTRN(5), A1, A2, ETRN(6), PER1, PER2, PESTEE1940
*,RT, R1F, R2E , AUX STEE1950
505 FORMAT (
*      X COMPONENT      Y COMPONENT      Z COMPONENT
*MAGNITUDE',', FIRST VECTOR      ',4D20.11/', SECOND VECTOR      ', STEE1960
*4D20.11 /', TOTAL      ', 60X, D20.11 / 23X,'INITIAL ORBIT STEE1970
*      FINAL ORBIT      TRANSFER ORBIT',', P      ', STEE2000
*3D20.11 / , E      ',3D20.11 / , I (DFG)      ', STEE2010
*3D20.11 / , R A NODE (DEG)      ',3D20.11 / , ARG PER (DEG)      ', STEE2020
*3D20.11 / , A      ', 3D20.11 / , PERIOD      ', STEE2030
*3D20.11 / , TRAN PT RAD      ',2D20.11 / , PHI-1, PHI-2, P      ', STEE2040
* 3D20.11) STEE2050
      XI1 = XI1 / RAD STEE2060
      W1 = W1 /RAD STEE2070
      W2 = W2 /RAD STEE2080
      RETURN STEE2090
      END STEE2100

```

Figure 33. Subroutine STEEP Listing (Continued)

```

C      SUBROUTINE IMP2( A , GRAD , NOPT , NTAU )
C      -----
C      PERFORMS OPTIONAL P-OPTIMIZATION. COMPUTES IMPULSE AND ITS
C      GRADIENT AS A FUNCTION OF PHI-1, PHI-2, P (TRANSFER).
C      -----
C      G. A. MCCUE-----SPACE SCIENCES LAB.
C      DECEMBER 1962
C      -----
C      SEMI-LATUS RECTUM FORMULATION , ( PHI 1 , PHI 2 , P ) -----
C      -----COMMON VAR. REGION-----
C      IMPLICIT REAL*8(A-H,O-Z)
C      COMMON/ZCONST/ PI,RAD,GXM,BETAS,CONV,II,JJ
C      COMMON /ORRCON/ P1,E1,W1,XI1,P2, E2,W2,P,E,W,ETRN(6)
C      COMMON /STCOM/ Q1(3), Q2(3),XE(3),XU(3),XV(3),AUX(3), CIMP20120
C      1XI,DVT(5),RIE,R2E,SXI,TAM(3),UN1(3),UN2(3),VE1(3),VE2(3
C      2),XLI,XL2,XS1,XS2,PERT,PER1,PER2,RAD1(3),RAD2(3),UMAG,XIMI(3,2),XIIMP20140
C      3M2(3,2),DIHETA,FPSI,ZETEST,
C      4JN , LOS , KONTAU, IBOX
C      DIMENSION A(3), XGRAD(3), VN(3), XB(3),GRAD(3), STEP1(3), GSAV
C      1(3),SX(3),UTAM(3),UQ(3),AUX1(3),DX11(3),DX12(3),DX21(3),DX22(3),XRIMP20180
C      2AD(3),VSAVE1(3),VSAVE2(3),DVT1P1(3),DVT1P2(3),DVT2P1(3),DVT2P2(3),IMP20190
C      3DVI1PH1(3),DV2PH2(3),DU1DP1(3),DU2DP2(3),DI1DP1(3),DI1DP2(3),DI2DP1IMP20200
C      4(3),DI2DP2(3), V(3,2)
C      IF( JN ) 10, 10 , 17
C      FUNCTIONS OF INITIAL AND FINAL ORBITS
C      REFERENCE VECTOR INITIALIZATION
C      10 CALL DBLVEC( Q1 , 0.00 , -DSIN( XI1 ) ,DCOS(XI1) )
C      CALL DBLVEC( Q2 , 0.00 , 0.00 , 1.00 )
C      CALL DBLVEC(VE1 ,DCOS(W1)*E1 ,DSIN(W1)*DCOS(XI1)*E1 ,
C      *DSIN(W1) * DSIN(XI1)*E1 )
C      CALL DBLVEC( VE2,DCOS(W2)*E2 , DSIN(W2)*E2 , 0.00 )
C      JN =1
C      PIG = 6.2831853D+00
C      PIG=PIG/DSQRT(GXM)
C      PER1 = PIG * DSQRT(( P1 / (1.00 - E1**2)) **3 )
C      PER2 = PIG * DSQRT (( P2 / (1.00 - E2**2)) **3 )
C      RMU1=DSQRT(GXM/P1)
C      RMU2=DSQRT(GXM/P2)
C      SXI=DSIN(XI1)
C      CXI=DCOS(XI1)
C      -----
C      IMP20000
C      IMP20010
C      IMP20020
C      IMP20030
C      IMP20040
C      IMP20050
C      IMP20060
C      IMP20070
C      IMP20080
C      IMP20090
C      IMP20100
C      IMP20110
C      IMP20120
C      IMP20130
C      IMP20140
C      IMP20150
C      IMP20160
C      IMP20170
C      IMP20180
C      IMP20190
C      IMP20200
C      IMP20210
C      IMP20220
C      IMP20230
C      IMP20240
C      IMP20250
C      IMP20260
C      IMP20270
C      IMP20280
C      IMP20290
C      IMP20300
C      IMP20310
C      IMP20320
C      IMP20330
C      IMP20340
C      IMP20350
C      IMP20360
C      IMP20370
C      IMP20380
C      IMP20390

```

Figure 34. Subroutine IMP2 Listing

```

C      FUNCTIONS OF PHI 1 AND PHI 2
17  A1=A(1)/RAD
    A2=A(2)/RAD
    P=A(3)
    R1F = P1/ (1.00 + E1 *DCOS(A1-W1))
    R2E = P2/ (1.00 + E2 *DCOS(A2-W2))
    CPI=DCOS(A1)
    SPI=DSIN(A1)
    CP2=DCOS(A2)
    SP2=DSIN(A2)
-----
C      RADIUS UNIT VECTORS
C      CALL DBLVEC (UN1, CPI, SPI*CXI, SPI*SXI )
C      CALL DBLVEC (UN2 , CP2 , SP2 , 0.00 )
-----
C      RADIUS VECTORS
C      DO 20 I=1,3
    RAD1(I)=UN1(I)*R1E
    RAD2(I)=UN2(I)*P2E
20  XRAD(I)=RAD2(I)-RAD1(I)
    THX=DBLDOT (UN1,UN2,3)
    THY = DSQRT(1.00 - THX**2 )
    DTHETA=DATAN2(THY,THX)
    DTSAV=DTHETA
    SINDTH=DSIN(DTSAV)
    TANDTH = DSIN( DTSAV /2.00) /DCOS(DTSAV / 2.00 )
    CALL DBLCRS (RAD1,RAD2,TAM)
    UMAG=DBLUNT (TAM,UTAM,?)
    XS1= DBLUNT(XRAD , XU , 3 )
    CALL DBLCRS (UTAM,XU,XV)
-----
C      VELOCITY VECTOR COMPUTATION
C      DO 50 I=1,3
50  AUX(I)=(VE1(I)+UN1(I))*RMU1
    CALL DBLCRS (Q1,AUX,VSAVE1)
    DO 60 I=1,3
60  V(I,I)=VSAVE1(I)
    AUX(I)=(VE2(I)+UN2(I))*RMU2

```

Figure 34. Subroutine IMP2 Listing (Continued)

```

CALL DBLCRS (Q2,AUX,VSAVE2)
DO 62 I=1,3
62 V(I,2) = VSAVE2(I)
-----
C
C
C   COMPUTE PARABOLIC LIMITS
IF (NOPT.EQ.2) GO TO 90
61 RRI=R1E*R2E-DBLDDOT(RAD1,RAD2,3)
   RR2 = DSQRT(2.00 * (-RRI + 2.00 * R1E * R2E ))
   PPMIN=RR1/(R1E+R2E+RR2)
   PPMAX=RR1/(R1E+R2E-RR2)
   POMIN = (PPMIN + PPMAX )/ 2.00
   PN=PPMIN-PPMIN
   PX=PPMAX-PPMIN
   P=PPMIN
-----
C
C   IMPULSE COMPUTED AS A FUNCTION OF (P)
IH = 1
IDR = 1
ISTEP = 0
DVTL = 1.00
90 DO 2000 INPT=1,25
99 Y=DSQRT(P/PI)
   XY=TANDTH/Y
   VV=Y*PI/UMAG
   DO 100 I=1,3
   COMP=RMU1*(VY*XRAD(I)+XY*UN1(I))
   XIM1(I,1)=COMP-V(I,1)
   XIM1(I,2)=-COMP-V(I,1)
   COMP=RMU1*(VY*XRAD(I)-XY*UN2(I))
   XIM2(I,1)=-COMP+V(I,2)
   XIM2(I,2)=+COMP+V(I,2)
100 DO 101 I=1,3
   DX1(I)=XIM1(I,1)
   DX1(I)=XIM1(I,2)
   DX2(I)=XIM2(I,1)
   DX2(I)=XIM2(I,2)
101

```

Figure 34. Subroutine IMP2 Listing (Continued)

```

IMP21140
IMP21150
IMP21160
IMP21170
IMP21180
IMP21190
IMP21200
IMP21210
IMP21220
IMP21230
IMP21240
IMP21250
IMP21260
IMP21270
IMP21280
IMP21290
IMP21300
IMP21310
IMP21320
IMP21330
IMP21340
IMP21350
IMP21360
IMP21370
IMP21380
IMP21390
IMP21400
IMP21410
IMP21420
IMP21430
IMP21440
IMP21450
IMP21460
IMP21470

XS1=DBLUNT(DX11,UQ,3)
XS2=DBLUNT(DX21,UQ,3)
XL1=DBLUNT(DX12,UQ,3)
XL2=DBLUNT(DX22,UQ,3)
DVT(1)=XS1 + XS2
DVT(2)=XL1 + XL2
-----
C
C      ELLIPTICAL LONG OR SHORT CHOICE
700  IBOX=50
    LOS=1
      DVT(3)=DVT(1)
    IF(DVT(2)-DVT(1))705,710,710
705  IBOX=35
    LOS=2
      DVT(3)=DVT(2)
-----
C
C      SEARCH FOR OPTIMUM (I) AS A FUNCTION OF (P)
710  ERROR = ( DVTL - DVT(3))* CONV
    IF(DABS(ERROR)-EPSI)2050,2050,1050
-----
C
C      DERIVATIVE COMPUTATION
1050  XRMU1 = RMU1 / 2.00 / P
    DO 1100 I=1,3
      AUX(I) = (YY*XRAD(I)-XY*UN1(I))
1100  AUX1(I) = (YY*XRAD(I)+XY*UN2(I))
-----
C
C      SHORT OR LONG DERIVATIVES
    IF (LOS .EQ. 2) GO TO 1300
1200  DIOP=XRMU1*(DBLDDT(DX11,AUX,3)/XS1
      * -DBLDDT(DX21,AUX1,3)/XS2)
    GO TO 1400
1300  DIOP=-XRMU1*(DBLDDT(DX12,AUX,3)/XL1
      * -DBLDDT(DX22,AUX1,3)/XL2)
-----
C

```

Figure 34. Subroutine IMP2 Listing (Continued)

```

C      CORRECT P
1400 IF (NOPT.EQ.2) GO TO 2050
1401 SDR = - DESIGN( 1.00 , DIDP )
1405 STEP = (PX + PN + DESIGN(PX-PN,SDR)) / 4.00 * ZETEST
      GO TO 1600
1410 IF (IDR .EQ.2) GO TO 1550
1500 IF(SSV-SDR) 1520,1600,1520
1520 IDR=2
1550 STEP = STEP +0.500
1600 P=P+SDR*STEP
      IF (IDR .EQ.2) GO TO 1800
1650 ISTEP=ISTEP+1
      IF (ISTEP-2) 1800,1800,1700
1700 ISTEP=0
      STEP = STEP * 2.00
1800 IF(P)1801,1801,1900
1801 P = 10.00
      STEP = ZETEST
1900 DVTL=DVT(3)
      SSV=SDR
2000 CONTINUE
2050 DVT(3) = DVT(3) * CONV
2100 DVT(1)=P
2101 PT1 = P/R1E -1.00
      PT2 = P/R2E -1.00
      DO 450 I=1,3
450  AUX(I)=(PT1*UN2(I) - PT2*UN1(I))/SINDTH
      CALL DBLCRS (AUX,UTAM,XE)
      F=DBLUNT (XE,XE,3)
2300 COSDTH=DCOS(DTSAV)
      IF ( NTAU .EQ. 3 ) RETURN

```

```

IMP21480
IMP21490
IMP21500
IMP21510
IMP21520
IMP21530
IMP21540
IMP21550
IMP21560
IMP21570
IMP21580
IMP21590
IMP21600
IMP21610
IMP21620
IMP21630
IMP21640
IMP21650
IMP21660
IMP21670
IMP21680
IMP21690
IMP21700
IMP21710
IMP21720
IMP21730
IMP21740
IMP21750
IMP21760
IMP21770
IMP21780
IMP21790

```

Figure 34. Subroutine IMP2 Listing (Continued)

SD 69-3

Figure 34. Subroutine IMP2 Listing (Continued)

```

DVT2P1(N)=RMUT*(-XR4*(UN1(N)*CSP1+DU1DP1(N)/SINDTH)-UN2(N)*
*   XR3*CSP1+XR2*DR1DP1/SINDTH-COTP1))
-----
C
4000 DVT2P2(N)=RMUT*(UN1(N)*(-XR4*CSP2+XR5*DR2DP2/SINDTH)-XR3*
*   (UN2(N)*CSP2+DU2DP2(N)/SINDTH)+UN2(N)*COTP2+DU2DP2(N)*COSDTH/
*   SINDTH)
-----
C
C   FINAL DERIVATIVES
IF (LOS .EQ.2) GO TO 5200
-----
C
C   SHORT DERIVATIVES
5100 DO 5150 N=1,3
      DI1DP1(N)=DVT1P1(N)-DV1PH1(N)
      DI2DP1(N)=-DVT2P1(N)
      DI1DP2(N)=DVT1P2(N)
      DI2DP2(N)=DV2PH2(N)-DVT2P2(N)
      DI1DP1=OBLDOT(DX11,DI1DP1,3)/XS1+ DBLDDOT(DX21,DI2DP1,3)/XS2
      DI2DP2=OBLDDOT(DX11,DI2DP2,3)/XS1+ DBLDDOT(DX21,DI2DP2,3)/XS2
      GO TO 5500
-----
C
C   LONG DERIVATIVES
5200 DO 5250 N=1,3
      DI1DP1(N)=-DVT1P1(N)-DV1PH1(N)
      DI2DP1(N)=DVT2P1(N)
      DI1DP2(N)=-DVT1P2(N)
      DI2DP2(N)=DV2PH2(N)+DVT2P2(N)
      DI1DP1=OBLDDOT(DX12,DI1DP1,3)/XL1+ DBLDDOT(DX22,DI2DP1,3)/XL2
      DI2DP2=OBLDDOT(DX12,DI2DP2,3)/XL1+ DBLDDOT(DX22,DI2DP2,3)/XL2
      5500 CALL DBLVFC(GRAD,DI1DP1*CONV / RAD , DI2DP2*CONV / RAD ,
*   DI1DP*CONV )
3000 RETURN
      END
-----
IMP22140
IMP22150
IMP22160
IMP22170
IMP22180
IMP22190
IMP22200
IMP22210
IMP22220
IMP22230
IMP22240
IMP22250
IMP22260
IMP22270
IMP22280
IMP22290
IMP22300
IMP22310
IMP22320
IMP22330
IMP22340
IMP22350
IMP22360
IMP22370
IMP22380
IMP22390
IMP22400
IMP22410
IMP22420
IMP22430
IMP22440
IMP22450

```

Figure 34. Subroutine IMP2 Listing (Continued)

```

C SUBROUTINE ELIPSE ( DPI, DXI, DXF, DXF, DPF, DPT, DXT, DMT, ELIP0000
C * DDMT, DTH1, DTH2, DEI, DET, DEF, RAD1, RAD2 ) ELIP0010
C ----- ELIP0011
C PROJECTS INITIAL AND TRANSFER ORBITS ON FINAL ORBIT PLANE. ELIP0012
C PLOTS ALL THREE ORBITS, NOOF LINE, IMPULSE VECTORS, PERIGEE LINES ELIP0013
C AND RADIIUS VECTORS TO THE TRANSFER POINTS. LABELS THE RESULTING ELIP0014
C FIGURE. ELIP0015
C ----- ELIP0016
C REAL*8 DPI, DXI, DMI, DPF, DXF, DPT, DXT, DMT, DDMT, ELIP0020
C * DTH1(2,3),DTH2(2,3), DEI, DET, DEF ELIP0030
C DIMENSION VPI(3),VQI(3),VPT(3),VQT(3),VPF(3),VQF(3),XYZI(3,181), ELIP0040
C * XYZI(3,181),XYZF(3,181),SPI(3), SPT(3),SPF(3),TH1(2,3),TH2(2,3), ELIP0050
C * RAD1(2,3),RAD2(2,3),YEAR(2) ELIP0060
C EXTERNAL NXV,NYV ELIP0070
C PI = SNGL(DPI) ELIP0080
C XI = SNGL(DXI) ELIP0090
C WI = SNGL(DWI) ELIP0100
C PF = SNGL(DPF) ELIP0110
C XF = SNGL(DXF) ELIP0120
C WF = SNGL(DWF) ELIP0130
C PT = SNGL(DPT) ELIP0140
C XT = SNGL(DXT) ELIP0150
C WT = SNGL(DWT) ELIP0160
C OMT = SNGL(DDMT) ELIP0170
C EI = SNGL(DEI) ELIP0180
C ET = SNGL(DEF) ELIP0190
C EF = SNGL(DEF) ELIP0200
C DO 5 I = 1,2 ELIP0210
C DO 5 J = 1,3 ELIP0220
C TH1(I,J) = SNGL( DTH1(I,J) ) ELIP0230
C TH2(I,J) = SNGL( DTH2(I,J) ) ELIP0240
C CALL FRAMEV(2) ELIP0250
C CALL SETMIV(0,0,0,0) ELIP0260
C SFT UP P AND Q VECTORS ELIP0270
C VPI(1) = COS(WI) ELIP0280
C VPI(2) = SIN(WI)*COS(XI) ELIP0290
C VPI(3) = SIN(WI)*SIN(XI) ELIP0300

```

Figure 35. Subroutine ELIPSE Listing

```

VQI(1) = -SIN(WI)
VQI(2) = COS(WI)*COS(XI)
VQI(3) = COS(WI)*SIN(XI)
VPF(1) = COS(WF)
VPF(2) = SIN(WF)
VPF(3) = 0.
VQF(1) = -SIN(WF)
VQF(2) = COS(WF)
VQF(3) = 0.
VPT(1) = COS(WT) * COS(OMT) - SIN(WT) * SIN(OMT) * COS(XT)
VPT(2) = COS(WT) * SIN(OMT) + SIN(WT) * COS(OMT) * COS(XT)
VPT(3) = SIN(WT) * SIN(XT)
VQT(1) = -SIN(WT) * COS(OMT) - COS(WT) * SIN(OMT) * COS(XT)
VQT(2) = -SIN(WT) * SIN(OMT) + COS(WT) * COS(OMT) * COS(XT)
VQT(3) = COS(WT) * SIN(XT)
C
TRANSFER THRUST AND RADIUS VECTORS, AND UNITIZE THRUST VECTORS
K = 1
DO 10 J = 1,3
  RAD1(K,1) = RAD1(1,J)
  RAD2(K,1) = RAD2(1,J)
  TH1(K,1) = TH1(1,J)
  TH2(K,1) = TH2(1,J)
10 K = K + 1
ZIP = UNIT (TH1,TH1,3) + UNIT(TH2,TH2,3)
SET UP PLOTTING ARRAY
DO 20 J=1,181
  ANG = J-1
  ANG = ANG*2./57.296
  SINT = SIN(ANG)
  COST = COS(ANG)
  RI = PI/(1.+EI*COST)
  RT = PT/(1.+ET*COST)
  RF = PF/(1.+EF*COST)
  DO 20 K=1,3
    XYZI(K,J) = RI*(COST*VPI(K) + SINT*VQI(K) )
    XYZT(K,J) = RT*(COST*VPT(K) + SINT*VQT(K) )
    XYZF(K,J) = RF*(COST*VPF(K) + SINT*VQF(K) )
20

```

Figure 35. Subroutine ELIPSE Listing (Continued)

```

C      FIND RANGE OF GRAPH
XS = XYZI(1,1)
XL = XS
YS = XYZI(2,1)
YL = YS
      DO 30 J=1,181
        XS = AMINI(XS,XYZI(1,J),XYZT(1,J), XYZF(1,J) )
        XL = AMAXI(XL,XYZI(1,J),XYZT(1,J), XYZF(1,J) )
        YS = AMINI(YS,XYZI(2,J),XYZT(2,J), XYZF(2,J) )
        YL = AMAXI(YL,XYZI(2,J),XYZT(2,J), XYZF(2,J) )
30      IF (XS .LT. YS) GO TO 32
        XS = YS
        GO TO 34
32      YS = XS
34      IF (XL .GT. YL) GO TO 36
        XL = YL
        GO TO 38
36      YL = XL
38      CONTINUE
        XR = (XL-XS)/20.
        YR = (YL-YS)/20.
        XS = XS - XR
        XL = XL + XR
        YS = YS - YR
        YL = YL + YR
        SET SCALES AND MARGINS
        CALL XSCALV(XS,XL,100,100)
        CALL YSCALV(YS,YL, 0,200 )
        PLOT ORBITS
        IDI = 1
        IDT = 7
        IDF = 13
        DO 40 J=2,181
          CALL LINEV(NXV(XYZF(1,J-1)),NVV(XYZF(2,J-1)),NXV(XYZF(1,J)),
                    NYV(XYZF(2,J)))
          IDF = MOD(IDF+1,18)
          IF (IDF.EQ.0) CALL PLOTV(NXV(XYZF(1,J))+4,NV(XYZF(2,J))+6,22)
40      *
C
C

```

Figure 35. Subroutine ELIPSE Listing (Continued)

```

      IF((XYZT(3,J-1) -LE-0.)-OR.(XYZT(3,J)-LT-0.)) GO TO 40
      CALL LINEV(NXV(XYZT(1,J-1)),NYV(XYZT(2,J-1)),NXV(XYZT(1,J)),
      NYV(XYZT(2,J)))
      GO TO 50
40    CALL PLOTV(NXV(XYZT(1,J-1)),NYV(XYZT(2,J-1)),42)
50    IDT = MOD(IDT+1,18)
      IF(IDT.EQ.0) CALL PLOTV(NXV(XYZT(1,J))+4,NYV(XYZT(2,J))+6,51)
      IF((XYZI(3,J-1)-LE-0.)-OR.(XYZI(3,J)-LT-0.)) GO TO 60
      CALL LINEV(NXV(XYZI(1,J-1)),NYV(XYZI(2,J-1)),NXV(XYZI(1,J)),
      NYV(XYZI(2,J)))
      GO TO 70
60    CALL PLOTV(NXV(XYZI(1,J-1)),NYV(XYZI(2,J-1)),42)
70    IDI = MOD(IDI+1,18)
      IF(IDI.EQ.0) CALL PLOTV(NXV(XYZI(1,J))+4,NYV(XYZI(2,J))+6,25)
80    CONTINUE
      C      PLOT NODE LINE
      CALL LINEV(NXV(XS+XR),NYV(0.),NXV(XL-XR),NYV(0.))
      C      PLOT P VECTORS
      DI = SQRT (XYZI(1,1)**2 + XYZI(2,1)**2)/16.
      DT = SQRT (XYZT(1,1)**2 + XYZT(2,1)**2)/16.
      DF = SQRT (XYZF(1,1)**2 + XYZF(2,1)**2)/16.
      C      VECTORS
      NXVO = NXV(0.)
      NYVO = NYV(0.)
      CALL LINEV(NXVO,NYVO,NXV(XYZI(1,1)),NYV(XYZI(2,1)))
      CALL LINEV(NXVO,NYVO,NXV(XYZT(1,1)),NYV(XYZT(2,1)))
      CALL LINEV(NXVO,NYVO,NXV(XYZF(1,1)),NYV(XYZF(2,1)))
      C      LABELS
      CALL PRINTV( 2,2HPI,NXV(XYZI(1,1))+.3*DI),NYV(XYZI(2,1))+.3*DI))
      CALL PRINTV( 2,2HPT,NXV(XYZT(1,1))+.3*DT),NYV(XYZT(2,1))+.3*DT))
      CALL PRINTV( 2,2HPF,NXV(XYZF(1,1))+.3*DF),NYV(XYZF(2,1))+.3*DF))
      CALL PRINTV ( 4,4HNODE,NXV (XL-XR),NYVO)
      C      PLOT RADIUS VECTORS AND THRUST VECTORS
      CALL LINEV(NXVO,NYVO,NXV(RAD1(1,1)),NYV(RAD1(2,1)))
      CALL LINEV(NXVO,NYVO,NXV(RAD2(1,1)),NYV(RAD2(2,1)))

```

Figure 35. Subroutine ELIPSE Listing (Continued)

ELIP1400
 ELIP1410
 ELIP1420
 ELIP1430
 ELIP1440
 ELIP1450
 ELIP1460
 ELIP1470
 ELIP1480
 ELIP1490
 ELIP1500
 ELIP1510
 ELIP1520
 ELIP1530
 ELIP1540
 ELIP1550
 ELIP1560
 ELIP1570
 ELIP1580
 ELIP1590

```

IRAD1Y = NXV(RAD1(1,1) + X*TH1(1,1)*1.5)
IRAD1Y = NYV(RAD1(2,1) + Y*TH1(2,1)*1.5)
IRAD2X = NXV(RAD2(1,1) + X*TH2(1,1)*1.5)
IRAD2Y = NYV(RAD2(2,1) + Y*TH2(2,1)*1.5)
CALL LINEV(NXV(RAD1(1,1)),NYV(RAD1(2,1)),IRAD1X,IRAD1Y)
CALL LINEV(NXV(RAD2(1,1)),NYV(RAD2(2,1)),IRAD2X,IRAD2Y)
NPADIX = IRAD1X + 18
NRAD1Y = IRAD1Y + 18
NRAD2X = IRAD2X + 18
NRAD2Y = IRAD2Y + 18
IF (IRAD1X.GT.NXV(RAD1(1,1))) NR DIX = NRAD1X - 36
IF (IRAD2X.GT.NXV(RAD2(1,1))) IRAD2X = NRAD2X - 36
IF (IRAD1Y.GT.NYV(RAD1(2,1))) NRAD1Y = NRAD1Y - 36
IF (IRAD2Y.GT.NYV(RAD2(2,1))) NRAD2Y = NRAD2Y - 36
CALL LINEV(IRAD1X,IRAD1Y,NRAD1X,NRAD1Y)
CALL LINEV(IRAD2X,IRAD2Y,NRAD2X,NRAD2Y)
CALL PRINTV( 2,2HI1,IRAD1X+8,IRAD1Y-8)
CALL PRINTV( 2,2HI2,IRAD2X+8,IRAD2Y-8)
RETURN
END

```

Figure 35. Subroutine ELIPSE Listing (Continued)

```

C      SUBROUTINE CONTUR(NOX,NOY,X,Y,Z,NOC,ZMN,ZMX,ZCUT,DELZ,DMAPX,
C      *DMAPY,IDT,IGRD,ITAB,IA7,ICON)
C      -----
C      EXECUTIVE SUBROUTINE FOR THE CONTOURING PACKAGE.
C      -----
C      COMMON /CNGRD/A7(15),XLAB(5),VLAB(5),RL(9),ZPLAN(36)
C      DIMENSION Z(NOX,NOY),X(NOX),Y(NOY)
C      SET UP A7(11) TO A7(14)
C      ZMIN=ZMN
C      ZMAX=ZMX
C      A7(11)=IDT
C      A7(12)=IGRD
C      A7(13)=ITAB
C      A7(14)=IA7
C      A7(15)=1.
C      IF((ITAB.LT.0).AND.(IA7.LT.0)) A7(15)=-1.
C      SET UP MARGINS
C      I IN. = 134 RASTERS
C      NORMAL MARGINS ARE--LEFT=24, RIGHT=0, BOTTOM=24, TOP=24
C      ML=10
C      MR=0
C      MB=24
C      MT = 22
C      CALL OXDYV (2,Y(1),Y(NOY),DY,M,J,NY,20.,IERR)
C      BOOST MARGIN EVEN THOUGH GRID MAY NOT BE WANTED, SINCE SCALING
C      ROUTINES LEAVE ROOM FOR GRID LABELS
C      ML = ML + NY*8 + 8
C      IF((IDT.LT.0).AND.((ITAB.GT.0).OR.(IA7.GT.0))) MR=125
C      CHECK FOR HOLDING DIMENSIONS OF MAP
C      IF((DMAPX.LT.0.).AND.(DMAPY.LT.0.)) GO TO 90
C      MA=ML+MR
C      MU=MB+MT
C      MAPH =(DMAPX*134.)+.5
C      MAPV =(DMAPY*134.)+.5
C      IF(MAPH.LT.0) MAPH=1023-MA
C      IF(MAPV.LT.0) MAPV=1023-MU

```

Figure 36. Subroutine CØNTUR Listing

```

MR1 = 1023 - ML - MAPH
MT1=1023-MR-MAPV
IF(IGRD.LT.0) GO TO 50
MT1=MT1-17
MR1=MR1-17
50 CONTINUE
MR = MAX0(MR,MR1)
MT = MAX0(MT,MT1)
MAP IS NOW SCALED (IF IT WILL FIT). FORCE SQUARE IF NECESSARY
IF((DMAPX.NE.DMAPY).OR.((DMAPX.LT.0.).OR.(DMAPY.LT.0.))) GO TO 90
MAPH= 1023-MR-ML
MAPV=1023-MT-MR
MAPSIZ=MIN0(MAPH,MAPV)
MR=1023-ML-MAPSIZ
MT=1023-MR-MAPSIZ
C ADJUST TO FINAL MARGINS AS SEEN BY CRT ROUTINES
90 IF(IGRD.GT.0) GO TO 100
MR = MR + NY*8 + 8
MT = MT + 17
100 ML = ML - NY*8 - 8
MT = MT - 17
IF ( MT.LT.5) MT = 5
CALL FRAMEV (2)
CALL SETMIV(ML,MR,MR,MT)
CALL XSCALV(X(1),X(NOX),ML,MR)
CALL YSCALV(Y(1),Y(NOY),MR,MT)
CALL CONGRD(X(1),X(NOX),Y(1),Y(NOY),NOC,1)
CALL CONSET(NOX,NOY,Z,NOC,ZMIN,ZMAX,DELZ,ZCUT,ICON)
CALL TRISEL(NCX,NCY,X,Y,Z,NOC,ZCUT,DX,DY,ZMIN,ZMAX,DELZ)
CALL CONGRD(X(1),X(NOX),Y(1),Y(NOY),NOC,2)
CALL SETMIV(24,0,24,24)
RETURN
END

```

Figure 36. Subroutine CØNTUR Listing (Continued)

```

C      SUBROUTINE CONGRD (XL,XR,YB,YT,NOC,ILAB)
C      -----
C      GENERATES GRID (IF DESIRED) AND SCALING INFORMATION FOR SUBSEQUENT
C      PLOTTING OF CONTOUR MAP. PLOTS CONTOUR IDENTIFICATION TABLE,
C      PARAMETER TABLE, AND TITLES.
C      -----
C      COMMON /CNGRD/A7(15),XLAB(5),YLAB(5),BL(9),ZPLAN(36)
C      CALL BRITVEV
C      GO TO (10,200),ILAB
10    CALL DXDYV(2,YB,YT,DY,M,J,NY,20.,IER)
      IF(A7(12).LT.0.)GO TO 20
      WANT GRID
      CALL DXDYV(1,XL,XR,DX,N,I,NX,20.,IER)
      CALL GRIDIV(2,XL,XR,YB,YT,DX,DY,N,M,I,J,NX,NY)
      PLOT LABELS
20    CALL SETMOV(ML,MR,MR,MT)
      LCH= ((1023-ML-MR)/2)+ML
      LCV = ((1023-MR-MT)/2)+MR
      LPH=LCH -72
      IF(LPH.LT.0) LPH=0
      CALL PRINTV( 20,XLAB,LPH,14)
      LPH=LCH-144
      IF(LPH.LT.0) LPH=0
      CALL PRINTV(-36,BL,LPH,0)
      LPV=LCV +72
      IF (LPV.LT.166) LPV=166
      CALL APRNTV(0,-14, 20,YLAB,0,LPV)
      IF (A7(12).GT.0.) GO TO 300
      *FRAME* NON-GRIDED AREA
      MT= 1023-MT
      MR= 1023-MR
      DO 30 IT=1,2
      CALL LINEV (ML,MB,MR,MR)
      CALL LINEV (ML,MB,ML,MT)
      CALL LINEV (MR,MB,MR,MT)
      CALL LINEV (ML,MT,MR,MT)
30    GO TO 300

```

Figure 37. Subroutine CONGRD Listing

```

C      SET UP ID AND A7 TABLE
200  IF(A7(15).LT.0.)GO TO 300
    IF(A7(11).GT.0.)GO TO 290
C      IDENTIFICATION WANTED ON SAME FRAME
    IDH = 894
210  IDV=970
    IF(A7(13).LT.0.) GO TO 250
    IDV=IDV-32
    CALL APRNTV(0,-16, NDC,36H123456789ABCDEFGHIJKLMNPPQRSTUVWXYZ
    *IDH+8,IDV)
    IDH=IDH+32
    DO 230 I=1,NDC
        CALL LABLV(ZPLAN(I),IDH,IDV,7,1,6)
        CALL LABLV(ABS(AMOD(ZPLAN(I),1.)),IDH+48,IDV,7,1,0)
230  IDV=IDV-16
        IDV=IDV-24
    *LOT A7 TABLE
C      250  IF(A7(14).LT.0.) GO TO 300
        CALL PRINTV( 10,10HPARAMETERS,IDH+16,IDV)
        IDV=IDV-32
        DO 270 I=1,10
            CALL LABLV(A7(I),IDH,IDV,7,1,6)
            CALL LABLV(ABS(AMOD(A7(I),1.)),IDH+48,IDV,7,1,0)
270  IDV=IDV-16
        GO TO 300
290  IDH=10
        CALL FRAMEV (2)
        GO TO 210
300  RETURN
    END

```

CONGI170
 CONGI175
 CONGI180
 CONGI185
 CONGI190
 CONGI195
 CONGI200
 CONGI210
 CONGI215
 CONGI220
 CONGI225
 CONGI230
 CONGI235
 CONGI240
 CONGI245
 CONGI250
 CONGI255
 CONGI260
 CONGI265
 CONGI270
 CONGI275
 CONGI280
 CONGI285
 CONGI290
 CONGI295
 CONGI300
 CONGI305
 CONGI310
 CONGI315
 CONGI320

Figure 37. Subroutine CONGRD Listing (Continued)

```

C      SUBROUTINE CONSET (NNX,NNY,Z,NOC,ZMIN,ZMAX,DELZ,ZCUT,ICON)
C      CNSE1005
C      CNSE1006
C      COMPUTES AND/OR SETS APPROPRIATE CONTOUR INTERVALS.
C      CNSE1007
C      CNSE1008
C      COMMON/CNGRD/A7(15),XLAB(5),YLAB(5),BL(9),ZPLAN(36)
C      CNSE1010
C      DIMENSION Z(NNX,NNY)
C      CNSE1015
C      GO TO(5000,1000,1000),ICON
C      CNSE1020
C      1000 ZMIN=Z(1,1)
C      CNSE1030
C      DO 1005 I=1,NNY
C      CNSE1035
C      DO 1005 J=1,NNX
C      CNSE1040
C      1005 ZMIN = AMIN1(Z(J,I),ZMIN)
C      CNSE1045
C      GO TO(2000,2000,1100),ICON
C      CNSE1050
C      1100 ZMAX=0.0
C      CNSE1060
C      DO 1200 I=1,NNY
C      CNSE1065
C      DO 1200 J=1,NNX
C      CNSE1070
C      IF (Z(J,I)-ZCUT) 1150,1150,1200
C      CNSE1075
C      1150 ZMAX=AMAX1 (ZMAX,Z(J,I))
C      CNSE1080
C      1200 CONTINUE
C      CNSE1085
C      DELZ=AIN1((ZMAX-ZMIN)/(FLOAT(NOC)-1.)*100.)/100.
C      CNSE1090
C      2000 ZMIN=ZMIN-AMOD(ZMIN,DELZ)+DELZ
C      CNSE1095
C      5000 ZPLAN(1)=ZMIN
C      CNSE1100
C      DO 5100 I=2,NOC
C      CNSE1105
C      ZPLAN(I)=ZPLAN(I-1)+DELZ
C      CNSE1110
C      ZMIN=ZMIN-DELZ
C      CNSE1115
C      ZMAX=ZPLAN(NOC)
C      CNSE1120
C      RETURN
C      CNSE1125
C      END
C      CNSE1130

```

Figure 38. Subroutine CONSET Listing

```

C      SUBROUTINE TRISEL(NNX,NNY,XX,YY,Z,NOC,ZCUT,DX,DY,ZMIN,ZMAX,DELZ)  TRSEL005
C      -----TRSEL006
C      SELECTS TRIANGLE TO BE CONTOURED.  TRSEL007
C      -----TRSEL008
C      DIMENSION Z(NNX,NNY),XX(NNX),YY(NNY)  TRSEL010
C      DIMENSION ZT(4)  TRSEL015
C      COMMON /CNGRD/A7(15),XLAR(5),YLAR(5),BL(9),ZPLAN(36)  TRSEL020
10  DO 2200  NY=2,NNY  TRSEL025
20  ZT(1)=Z(1,NY-1)  TRSEL030
    ZT(4)=Z(1,NY)  TRSEL035
    DY=YY(NY)-YY(NY-1)  TRSEL040
100  DO 2000  NX=2,NXX  TRSEL045
    ZT(2)=Z(NX,NY-1)  TRSEL050
    ZT(3)=Z(NX,NY)  TRSEL055
    DX=XX(NX)-XX(NX-1)  TRSEL060
    ID=0  TRSEL065
    INO=2  TRSEL070
    DO 150  K=1,4  TRSEL075
    IF(ZT(K).LE.ZCUT.AND.ZT(K).GE.ZMIN) GO TO 150  TRSEL080
    INO=INO-1  TRSEL085
    ID=K  TRSEL090
    IF(INO) 1000,1900,150  TRSEL095
150  CONTINUE  TRSEL100
    GO TO (200,160),INO  TRSEL105
200  GO TO (201,202,203,204),ID  TRSEL115

```

Figure 39. Subroutine TRISEL Listing

```
160 IF (ABS(ZT(3))-ZT(1))-ABS(ZT(4)-ZT(2))) 202,201,201
201 CALL TRICON(XX(NX),YY(NY),-DX,-DY,NOC,ZMIN,ZT(4),ZT(3),ZT(2),DELZ)
IF (ID) 1900,203,1900
202 CALL TRICON(XX(NX-1),YY(NY),DX,-DY,NOC,ZMIN,ZT(3),ZT(4),ZT(1),
*DELZ)
IF (ID) 1900,204,1900
203 CALL TRICON(XX(NX-1),YY(NY-1),DX,DY,NOC,ZMIN,ZT(2),ZT(1),ZT(4),
*DELZ)
GO TO 1900
204 CALL TRICON(XX(NX),YY(NY-1),-DX,DY,NOC,ZMIN,ZT(1),ZT(2),ZT(3),
*DELZ)
1900 ZT(1)=ZT(2)
ZT(4)=ZT(3)
2000 CONTINUE
2200 CONTINUE
RETURN
END
```

TRSE1125
TRSE1130
TRSE1135
TRSE1140
TRSE1145
TRSE1150
TRSE1155
TRSE1160
TRSE1165
TRSE1170
TRSE1175
TRSE1180
TRSE1185
TRSE1190
TRSE1195
TRSE1200
TRSE1205

Figure 39. Subroutine TRISEL Listing (Continued)

	SUBROUTINE TRICON (X,Y,DX,DY,NOC,ZMIN,ZLX,ZV,ZY,DELZ)	TRC01000
	-----	TRC01005
C	PERFORMS ACTUAL CONTOURING OPERATION INCLUDING INTERPOLATION TO	TRC01006
C	COMPUTE THE END POINTS OF THE LINE SEGMENTS WHICH ARE THEN	TRC01007
C	PLOTTED BY PLOT3.	TRC01008
C	-----	TRC01009
	DIMENSION VEC(36,2,2),IX(4)	TRC01010
	COMMON /CNGRD/A7(15),XLAB(5),YLAB(5),BL(9),ZPLAN(36)	TRC01015
	10 IX(1)=(ZX-ZMIN)/DELZ	TRC01020
	20 IX(2)=(ZV-ZMIN)/DELZ	TRC01025
	30 IX(3)=(ZY-ZMIN)/DELZ	TRC01030
	40 IX(4)=IX(1)	TRC01035
	50 IMIN=MINO(IX(1),IX(2),IX(3)) +1	TRC01040
	60 IMAX=MINO(MAXO(IX(1),IX(2),IX(3)),NOC)	TRC01045
	70 INT=IMAX-IMIN	TRC01050
	IF (INT) 1000,72,72	TRC01055
	72 I2=1	TRC01060
	DO 500 K=1,3	TRC01065
	MIN=MINO(IX(K+1),IX(K)) + 1	TRC01070
	MAX=MINO(MAXO(IX(K+1),IX(K)),NOC)	TRC01075
	INZ=MAX-MIN	TRC01080
	75 IF(INZ) 500,80,80	TRC01085
	80 IF(INZ-INZ)90,85,90	TRC01090
	85 GO TO (95,90),I2	TRC01095
	90 I2=1	TRC01105
	GO TO 99	TRC01110
	95 I2=2	TRC01115

Figure 40. Subroutine TRICØN Listing

```

99 DO 400 J=MIN,MAX
   GO TO (100,200,300),K
100 VEC(J,1,I2)=X+DX*(ZPLAN(J)-ZV) / (ZX-ZV)
   VEC(J,2,I2)=Y
   GO TO 400
200 VEC(J,1,I2)=X
   VEC(J,2,I2)=Y+DY*(ZPLAN(J)-ZV) / (ZY-ZV)
   GO TO 400
300 VEC(J,1,I2)=X+DX*(ZPLAN(J)-ZY) / (ZX-ZY)
   VEC(J,2,I2)=Y+DY*(ZPLAN(J)-ZX) / (ZY-ZX)
400 CONTINUE
500 CONTINUE
   DO 600 J= IMIN, IMAX
600 CALL PLOT3(J,VEC(J,1,1),VEC(J,2,1),VEC(J,1,2),VEC(J,2,2))
1000 RETURN
      END

```

TRC01120
 TRC01125
 TRC01135
 TRC01140
 TRC01145
 TRC01150
 TRC01155
 TRC01160
 TRC01165
 TRC01170
 TRC01175
 TRC01180
 TRC01185
 TRC01190
 TRC01195
 TRC01200

Figure 40. Subroutine TRICØN Listing (Continued)

```

C      SUBROUTINE PLOT3 (NS, X1, Y1, X2, Y2)
C      -----
C      PLOTS INDIVIDUAL LINE SEGMENTS WHICH FORM CONTOUR LINES.
C      LABELS CONTOUR LINES.
C      -----
C      EXTERNAL NXV,NYV
C      DIMENSION ICH(50)
C      10 IF(ICH(1).EQ.1) GO TO 100
C      20 DO 30 K=1,9
C      ICH(K) = K
C      ICH(K+9) = K + 16
C      ICH(K+18) = K + 32
C      30 ICH(K+27) = K + 49
C      100 IPLNT = MOD (IPLNT+1,10)
C      IF (IPLNT ) 180, 170, 180
C      PLOT I.D. CHARACTER
C      170 CALL PLOTV (NXV(X1), NYV(Y1), ICH(NS) )
C      180 CALL LINEV (NXV(X1), NYV(Y1), NXV(X2), NYV(Y2) )
C      1000 RETURN
C      END
C      PLOT1005
C      PLOT1006
C      PLOT1007
C      PLOT1008
C      PLOT1009
C      PLOT1010
C      PLOT1015
C      PLOT1020
C      PLOT1025
C      PLOT1030
C      PLOT1035
C      PLOT1040
C      PLOT1045
C      PLOT1050
C      PLOT1055
C      PLOT1060
C      PLOT1065
C      PLOT1070
C      PLOT1075
C      PLOT1080

```

Figure 41. Subroutine PLOT3 Listing

```

C      DOUBLE PRECISION FUNCTION DBLDDOT (X,Y,N)
C      -----
C      DOUBLE PRECISION FUNCTION THAT COMPUTES THE DOT-PRODUCT OF
C      TWO VECTORS
C      -----
C      DOUBLE PRECISION X(3),Y(3)
C      N MUST BE EQUAL TO OR LESS THAN 3
C      DBLDDOT = 0. DO
C      DO 10 I=1,N
C      10 DBLDDOT=DBLDDOT+X(I)*Y(I)
C      RETURN
C      END
DBLD1000
DBLD1005
DBLD1010
DBLD1015
DBLD1020
DBLD1025
DBLD1030
DBLD1035
DBLD1040
DBLD1045
DBLD1050
DBLD1055
```

Figure 42. Function DBLDDOT Listing

```

C      SUBROUTINE DBLCRS (X,Y,Z)
C      -----
C      DOUBLE PRECISION SUBROUTINE THAT TAKES THE CROSS-PRODUCT OF
C      TWO VECTORS
C      -----
C      DOUBLE PRECISION X(3), Y(3), Z(3)
C      Z=X CROSS Y
C      Z(1)=X(2)*Y(3)-Y(2)*X(3)
C      Z(2)=X(3)*Y(1)-Y(3)*X(1)
C      Z(3)=X(1)*Y(2)-Y(1)*X(2)
C      RETURN
C      END
DBLC1000
DBLC1005
DBLC1010
DBLC1011
DBLC1012
DBLC1015
DBLC1020
DBLC1025
DBLC1030
DBLC1035
DBLC1040
DBLC1045
```

Figure 43. Subroutine DBLCRS Listing

```
C      DOUBLE PRECISION FUNCTION DBLUNT (P,U,N)      DBLU1010
-----      DBLU1015
C      DOUBLE PRECISION FUNCTION THAT UNITIZES A VECTOR  DBLU1020
-----      DBLU1025
C      THE VALUE OF THE FUNCTION IS THE MAGNITUDE
C      DOUBLE PRECISION P(3),U(3), SAVE
C      *,DBLUNT,DSQRT
C      N MUST BE EQUAL TO OR LESS THAN 3
C      SAVE=DSQRT(DBLUNT(P,P,N))
C      DO 10 I=1,N
C      10 U(I)=P(I)/SAVE
C      DBLUNT=SAVE
C      RETURN
C      FND
```

Figure 44. Function DBLUNT Listing

```
C      SUBROUTINE DBLVEC (X,A,B,C)      DBLV1000
-----      DBLV1005
C      DOUBLE PRECISION SUBROUTINE THAT SETS UP A VECTOR BY
C      STIPULATION OF ITS COMPONENTS      DBLV1010
C      DBLV1015
C      DBLV1020
C      DOUBLE PRECISION A, B, C, X(3)      DBLV1025
C      A,B, AND C ARE THE X,Y, AND Z COMPONENTS OF VECTOR X RESPECTIVELY
C      X(1)=A      DBLV1030
C      X(2)=B      DBLV1035
C      X(3)=C      DBLV1040
C      RETURN      DBLV1045
C      FND      DBLV1050
C      DBLV1055
```

Figure 45. Subroutine DBLVEC Listing

```

C
C
C
C
FUNCTION DOT (X,Y,N)
-----
C
C SINGLE PRECISION FUNCTION THAT COMPUTES THE DOT-PRODUCT OF
C TWO VECTORS
C
-----
C
C DIMENSION X(N), Y(N)
C DOT=0.0
C DO 10 I=1,N
C 10 DOT=DOT+X(I)*Y(I)
C RETURN
C END
DOT 1005
DOT 1010
DOT 1015
DOT 1016
DOT 1020
DOT 1025
DOT 1030
DOT 1035
DOT 1040
DOT 1045
DOT 1050

```

Figure 46. Function DOT Listing

```

C
C
C
C
FUNCTION UNIT (P,U,N)
-----
C
C SINGLE PRECISION FUNCTION THAT UNITIZES A VECTOR
C
-----
C
C DIMENSION P(3), U(3)
C N MUST BE EQUAL TO OR LESS THAN 3
C SAVE=SQRT(DOT(P,P,N))
C DO 10 I=1,N
C 10 U(I)=P(I)/SAVE
C UNIT=SAVE
C RETURN
C END
UNIT1005
UNIT1010
UNIT1015
UNIT1020
UNIT1025
UNIT1030
UNIT1035
UNIT1040
UNIT1045
UNIT1050
UNIT1055
UNIT1060

```

Figure 47. Function UNIT Listing