

UNCLASSIFIED

AD 267 325

*Reproduced
by the*

**ARMED SERVICES TECHNICAL INFORMATION AGENCY
ARLINGTON HALL STATION
ARLINGTON 12, VIRGINIA**



UNCLASSIFIED

NOTICE: When government or other drawings, specifications or other data are used for any purpose other than in connection with a definitely related government procurement operation, the U. S. Government thereby incurs no responsibility, nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use or sell any patented invention that may in any way be related thereto.

62-1-4
XEROX

WADD TECHNICAL REPORT 61-133
VOLUME III

CATALOGED BY **ASTIA**
AS AD NO. **267325**

PASSIVE AERODYNAMIC ATTITUDE STABILIZATION OF NEAR EARTH SATELLITES

Volume III
MATHEMATICAL TECHNIQUES AND COMPUTER PROGRAM

O. C. JUELICH

NORTH AMERICAN AVIATION, INC.
COLUMBUS, OHIO

JULY 1961

DEC 11 1961
TIPDR

642 400

AERONAUTICAL SYSTEMS DIVISION

WADD TECHNICAL REPORT 61-133
VOLUME III

**PASSIVE AERODYNAMIC ATTITUDE STABILIZATION
OF NEAR EARTH SATELLITES**

**Volume III
MATHEMATICAL TECHNIQUES AND COMPUTER PROGRAM**

O. C. JUELICH

*NORTH AMERICAN AVIATION, INC.
COLUMBUS, OHIO*

JULY 1961

FLIGHT DYNAMICS LABORATORY
CONTRACT Nr. AF 33(616)-7100
PROJECT Nr. 1366
TASK Nr. 13967

AERONAUTICAL SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
WRIGHT-PATTERSON AIR FORCE BASE, OHIO

McGregor & Werner, Inc., Dayton, O.
400 - November 1961 - 10-342

FOREWORD

The research reported herein was performed by North American Aviation, Inc., Columbus, Ohio for the Hypersonic Flight Section, Flight Branch of the Flight Dynamics Laboratory, Wright Air Development Division. The work was accomplished under Air Force Contract No. AF 33(616)-7100, Project No. 1366, Task No. 13967, "A Study of Aerodynamically Oriented and Stabilized Satellites." This research was carried out by the Engineering Research and Aerothermodynamics Development Groups of the Columbus Division, North American Aviation, Inc., with Dr. D. M. Schrello, Engineering Research Group, as Project Engineer. Mr. Joseph Ondrejka, Flight Dynamics Laboratory, was WADD Project Engineer.

The results of this study are reported in a series of three volumes of which this is Volume III. The other reports in this series are:

VOLUME I: "Librations due to Combined Aerodynamic and Gravitational Torques," by D. M. Schrello

VOLUME II: "Aerodynamic Analysis," by Paul H. Davison

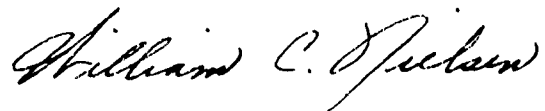
ABSTRACT

A computer program is presented which integrates numerically the pitch equation of an aerodynamically stabilized satellite. The mathematical theory of the linear second order equation with periodic coefficients and the integration procedure are discussed to clarify the structure and application of the program.

PUBLICATION REVIEW

This report has been reviewed and is approved.

FOR THE COMMANDER:



WILLIAM C. NIELSEN
Colonel, USAF
Chief, Flight Dynamics Laboratory

TABLE OF CONTENTS

	<u>Page No.</u>
INTRODUCTION.....	1
DISCUSSION	
1. DISCUSSION OF THE LINEAR SECOND ORDER DIFFERENTIAL EQUATION WITH PERIODIC COEFFICIENTS.....	3
1.1 General Case.....	3
1.2 Resonant Case.....	6
1.3 Stability Criteria.....	9
1.4 Short Term Boundedness.....	10
1.5 A "Built-In" Check on the Accuracy of the Numeric Integration.....	11
2. INTEGRATION PROCEDURE.....	12
2.1 Derivation of Predictor Formula.....	12
2.2 Corrector Formula.....	14
2.3 Integration Scheme.....	15
3. COMPUTER PROGRAM.....	17
3.1 Integration Portion of the Program.....	17
3.2 Card-Punching Portion of the Program.....	18
3.3 Atmospheric Properties Subroutine.....	19
3.4 A Potential Application.....	19
APPENDIX A - PROGRAM DESCRIPTION - PITCHING MOTION OF A SATELLITE.....	21
APPENDIX B - PROGRAM DESCRIPTION - ATMOSPHERE SUBROUTINE.....	51
APPENDIX C - PROGRAM DESCRIPTION - FIXED DECIMAL CARD READ ROUTINE WITH ERROR DETECTION.....	57
APPENDIX D - PROGRAM DESCRIPTION - PUNCH SCALING FUNCTION.....	61
APPENDIX E - PROGRAM DESCRIPTION - OUTPUT TAPE WRITE ROUTINE.....	65
APPENDIX F - SUMMARY OF LIBRARY ROUTINES.....	68
APPENDIX G - SAMPLE INPUT AND OUTPUT.....	69
BIBLIOGRAPHY.....	88

LIST OF SYMBOLS

A, B, C, D	Coefficients used in the derivation of Eq. (32)
a, b	Initial condition constants
a_n	The nth coefficient of Taylor's Series
F	Number of a card field
f	Function generated in the card-punch program
h	Altitude, increment in t
I	Identity matrix
i	Orbit inclination, Counting index
M	Inertia parameter
n	Counting index, integral multiplier
P, P_p, Q, Q_p	Aerodynamic parameters
r	Distance from center of earth
r_E	Mean radius of earth
S_n	Matrix defined in Eq. (22)
s	Constant appearing in error term of Taylor's series
t	Independent variable
V	Velocity of vehicle in inertial space
V_R	Velocity of vehicle relative to atmosphere
v	True anomaly of vehicle
W	Matrix whose determinant is the Wronskian of the system (y_1, y_2)
W	Wronskian of the system (y_1, y_2)
w	Eigenvalue bound for W , defined in Eq. (23)
w_p, w_1, w_2, w_γ	Weighting constants in card-punch program
y	Solution of differential equation, Eqs. (5) or (6)
α	Angle of attack

α, β, γ	Periodic functions of t
Γ, Γ_e	Aerodynamic parameters
γ	Inclination of orbit to local horizontal
ϵ	Eccentricity, error
θ	Pitch angle relative to local horizontal
ρ	Atmospheric density
σ_{ni}	Eigenvalues of matrix S_n
ω_e	Earth rotation rate
ω	Eigenvalue of the matrix W

Subscripts (except counting indices)

e	even part
n	operator defined in Eq. (13)
o	odd part
p	perigee
p	periodic, calculated from predictor formula
s	calculated from Simpson's rule
0	particular solution
$1, 2$	homogeneous solution

Notation

\dot{x}	Derivative with respect to t
x'	Derivative with respect to ν
\bar{x}	Maximum magnitude, see Eq. (11)
x^*	An instance of x
Δx	An increment of x
$\frac{d}{dx}$	Differentiation
\cong	Approximate equality
\equiv	Identity

INTRODUCTION

The equation of satellite pitching motion derived in Volume I of this series of reports is:

$$\theta'' + 2Q\theta' + \{3M[1 - \epsilon \cos v] + P\}\theta = 2Q + \gamma P \quad (1)$$

where v , the true anomaly, is the independent variable,

θ is the pitch angle from the local horizontal,

ϵ is the orbital eccentricity,

M is the inertia ratio parameter,

and the remaining quantities are defined by the following relations:

$$\gamma = \frac{\epsilon \sin v}{1 + \epsilon \cos v}, \quad (2)$$

$$P = P_p (\rho/\rho_p) (V_R/V)^2 [1 + 2\epsilon(1 - \cos v)],$$

$$Q = Q_p (\rho/\rho_p) (V_R/V) [1 - \epsilon(1 - \cos v)] - \epsilon \sin v,$$

in which

$$(V_R/V) = 1 - (\omega_E / \dot{v}_p) \cos i,$$

$$\dot{v}_p = \frac{1}{r_p} \sqrt{\frac{\mu(1+\epsilon)}{r_p}}, \quad (3)$$

the subscript P denotes perigee conditions,

μ is the product of the universal gravitational constant and the mass of the earth,

ω_E is the Earth's rotation rate,

i is the orbit inclination,

ρ , the atmospheric density, is a tabular function of the altitude h , see for instance Minzner, (Ref. 1). The altitude h in turn is defined by

Manuscript released by the author February 1961 for publication as a WADD Technical Report.

$$h = r - r_E \quad (4)$$

with r_E the mean earth radius, and

$$r = r_p (1 + \epsilon) / (1 + \epsilon \cos v).$$

The quantities P_p , Q_p are defined by

$$P_p = \Gamma \rho_p r_p^2 / 2$$

$$Q_p = \Gamma_q \rho_p r_p / 8$$

where Γ is a pitching moment parameter, and

Γ_q is a damping in pitch parameter.

These parameters are defined and discussed in Volume I of this series of reports. It is desired to calculate θ and possibly the angle of attack $\alpha = \theta - \gamma$ for "all values" of the true anomaly.

Even though the differential equation is a linear second order equation it does not lend itself readily to solution in terms of elementary or tabulated functions particularly since the variation of density with altitude is a tabulated function. Direct numeric integration appears to offer a rather straightforward approach, provided its limitation to a finite range of v can be circumvented.

The means used to obtain results valid for all v constitute the first section of this report. The second section describes the method of numeric integration adopted. The final section discusses the computer program written to implement these results while the program itself is given in the appendices.

1. DISCUSSION OF THE LINEAR SECOND ORDER DIFFERENTIAL EQUATION WITH PERIODIC COEFFICIENTS

It is the purpose of this section of the report to show that three integrations carried out over one orbital period suffice to furnish information about the boundedness or stability of solutions of Eq. (1). Equations with arbitrary periodic coefficients are discussed to insure independence of the tabulated atmospheric density function. In this form the presentation is an amplification of the concise discussion given by Coddington and Levinson, (Ref. 2) and a minor extension as well since the reference discusses homogeneous equations only.

1.1 General Case

Let

$$\ddot{y} + \alpha \dot{y} + \beta y = \gamma \quad (5)$$

be a linear differential equation in which α , β , and γ are periodic functions of t , with common period of 2π . The dots denote differentiation with respect to t . Eq. (1) is of this type. Let

$$\ddot{y} + \alpha \dot{y} + \beta y = 0 \quad (6)$$

be the associated homogeneous equation. Now let y_0 be that solution of Eq. (5) that has

$$y_0(0) = 0, \quad \dot{y}_0(0) = 0$$

and let y_1, y_2 be solutions of Eq. (6) that have respectively

$$y_1(0) = 1, \quad \dot{y}_1(0) = 0,$$

$$y_2(0) = 0, \quad \dot{y}_2(0) = 1.$$

Since these solutions are obviously linearly independent, any solution of Eq. (5) may be written

$$y = y_0 + ay_1 + by_2. \quad (7)$$

For $t = 0, 2\pi$ Eq. (7) yields

$$\left. \begin{aligned} y(0) &= y_0(0) + a y_1(0) + b y_2(0) = a \\ \dot{y}(0) &= \dot{y}_0(0) + a \dot{y}_1(0) + b \dot{y}_2(0) = b \end{aligned} \right\}$$

$$\left. \begin{aligned} y(2\pi) &= y_0(2\pi) + a y_1(2\pi) + b y_2(2\pi) \\ \dot{y}(2\pi) &= \dot{y}_0(2\pi) + a \dot{y}_1(2\pi) + b \dot{y}_2(2\pi) \end{aligned} \right\}$$

(8)

If it is now required that y have period 2π ,

$$y(2\pi) = y(0), \quad \dot{y}(2\pi) = \dot{y}(0)$$

(9)

Eqs.(8) may be combined to yield

$$\left. \begin{aligned} y_0(2\pi) &= a [1 - y_1(2\pi)] - b y_2(2\pi) \\ \dot{y}_0(2\pi) &= -a \dot{y}_1(2\pi) + b [1 - \dot{y}_2(2\pi)] \end{aligned} \right\}$$

(10)

Let

$$W = \begin{bmatrix} y_1(2\pi) & y_2(2\pi) \\ \dot{y}_1(2\pi) & \dot{y}_2(2\pi) \end{bmatrix}$$

then Eq. (10) may be written

$$\begin{bmatrix} y_0(2\pi) \\ \dot{y}_0(2\pi) \end{bmatrix} = (I - W) \begin{bmatrix} a \\ b \end{bmatrix}$$

A unique periodic solution to Eq. (5) is thus seen to exist if $(I - W)$ is non-singular. It will be shown below that

$$\det(I - W) = 0$$

implies a resonance in the conventional sense. In the non-resonant case let the periodic particular solution be denoted by the subscript p :

$$y_p = y_0 + a y_1 + b y_2$$

where a and b are the solution to the system of equations given by Eqs. (10).

Now define the notation

$$\bar{y} = \max_{0 \leq t \leq 2\pi} (|y|). \quad (11)$$

Then

$$\bar{y}_p \leq \bar{y}_0 + |a| \bar{y}_1 + |b| \bar{y}_2,$$

and y_p exists and is bounded whenever y_0 , y_1 , y_2 , exist on the interval $[0, 2\pi]^p$, and $\det(I-W) \neq 0$.

Under these conditions, any solution of Eq. (5) may be written

$$y = y_p + a_0 y_1 + b_0 y_2 \quad (12)$$

where a_0 and b_0 are initial condition constants. The only change from Eq. (7) is the choice of the particular solution. The notation

$$t_n = t - 2n\pi \quad (13)$$

will be useful. In terms of the conditions at $t = 2n\pi$ Eq. (12) becomes

$$y(t) = y_p(t_n) + a_n y_1(t_n) + b_n y_2(t_n)$$

and in turn, when $t_n = 2\pi$,

$$y(2\pi + 2n\pi) = y_p(2\pi) + a_n y_1(2\pi) + b_n y_2(2\pi)$$

$$= y_p(0) + a_{n+1} y_1(0) + b_{n+1} y_2(0) = y_p(0) + a_{n+1}.$$

Since $y_p(2\pi) = y_p(0)$,

$$a_{n+1} = a_n y_1(2\pi) + b_n y_2(2\pi).$$

Similarly

$$b_{n+1} = a_n \dot{y}_1(2\pi) + b_n \dot{y}_2(2\pi).$$

Thus

$$\begin{bmatrix} a_{n+1} \\ b_{n+1} \end{bmatrix} = W \begin{bmatrix} a_n \\ b_n \end{bmatrix} = W^{n+1} \begin{bmatrix} a_0 \\ b_0 \end{bmatrix}.$$

The solutions of Eq. (5) remain bounded if W^n remains bounded for all n . This in turn is the case if the magnitudes of the eigenvalues of W are at most 1. In this case an upper bound on the amplitude of solutions is

$$\bar{y} \leq \bar{y}_p + \sqrt{a_0^2 + b_0^2} (\bar{y}_1 + \bar{y}_2). \quad (14)$$

1.2 Resonant Case

If $(I-W)$ is singular, the system of equations

$$0 = (I-W) \begin{bmatrix} a \\ b \end{bmatrix} \quad (15)$$

has an infinity of non-trivial solutions. Let one of these be

$$a = a^*, \quad b = b^* \quad (16)$$

and let

$$y^* = a^* y_1 + b^* y_2. \quad (17)$$

Then Eq. (15) may be rewritten

$$0 = (I-W) \begin{bmatrix} a^* \\ b^* \end{bmatrix}$$

or

$$I \begin{bmatrix} a^* \\ b^* \end{bmatrix} = W \begin{bmatrix} a^* \\ b^* \end{bmatrix}$$

whence

$$\begin{bmatrix} y_1(0) & y_2(0) \\ \dot{y}_1(0) & \dot{y}_2(0) \end{bmatrix} \begin{bmatrix} a^* \\ b^* \end{bmatrix} = \begin{bmatrix} y_1(2\pi) & y_2(2\pi) \\ \dot{y}_1(2\pi) & \dot{y}_2(2\pi) \end{bmatrix} \begin{bmatrix} a^* \\ b^* \end{bmatrix},$$

so that

$$\begin{bmatrix} y^*(0) \\ \dot{y}^*(0) \end{bmatrix} = \begin{bmatrix} y^*(2\pi) \\ \dot{y}^*(2\pi) \end{bmatrix} . \quad (18)$$

Thus y^* is a non-trivial solution of period 2π of the homogeneous equation, Eq. (6). Since the period of y^* matches that of the forcing term of Eq. (5) this meets the accepted notions of resonance.

Conversely, if a solution of the homogeneous Eq. (6) satisfies Eq. (18) the argument may be reversed to show that Eq. (15) is valid.

In the resonant case, the system of equations, Eqs. (10), may or may not have a solution. If it does, the analysis for the non-resonant case applies, except that y_p is no longer unique. If, however, the system of equations, Eqs. (10) is inconsistent, no periodic solution of Eq. (5) exists. In this case let

$$y = y_0 + a_0 y_1 + b_0 y_2 \quad (19)$$

be some solution of Eq. (5), where a_0 and b_0 are again initial condition constants. By suitable choice of the initial condition constants Eq. (19) may be written

$$y(t) = y_0(t_n) + a_n y_1(t_n) + b_n y_2(t_n) \quad (20)$$

where t_n is defined in Eq. (13). Now

$$\begin{aligned} y(2\pi + 2n\pi) &= y_0(2\pi) + a_n y_1(2\pi) + b_n y_2(2\pi) \\ &= y_0(0) + a_{n+1} y_1(0) + b_{n+1} y_2(0) = a_{n+1} , \end{aligned}$$

and

$$\begin{aligned} \dot{y}(2\pi + 2n\pi) &= \dot{y}_0(2\pi) + a_n \dot{y}_1(2\pi) + b_n \dot{y}_2(2\pi) \\ &= \dot{y}_0(0) + a_{n+1} \dot{y}_1(0) + b_{n+1} \dot{y}_2(0) = b_{n+1} . \end{aligned}$$

Thus

$$\begin{bmatrix} a_{n+1} \\ b_{n+1} \end{bmatrix} = \begin{bmatrix} y_0(2\pi) \\ \dot{y}_0(2\pi) \end{bmatrix} + W \begin{bmatrix} a_n \\ b_n \end{bmatrix} ,$$

and, proceeding recursively

$$\begin{aligned}
 \begin{bmatrix} a_{n+1} \\ b_{n+1} \end{bmatrix} &= \begin{bmatrix} \gamma_0(2\pi) \\ \dot{\gamma}_0(2\pi) \end{bmatrix} + W \left(\begin{bmatrix} \gamma_0(2\pi) \\ \dot{\gamma}_0(2\pi) \end{bmatrix} + W \begin{bmatrix} a_{n-1} \\ b_{n-1} \end{bmatrix} \right) \\
 &= (I + W) \begin{bmatrix} \gamma_0(2\pi) \\ \dot{\gamma}_0(2\pi) \end{bmatrix} + W^2 \begin{bmatrix} a_{n-1} \\ b_{n-1} \end{bmatrix} \\
 &\dots = \left(\sum_{i=0}^n W^i \right) \begin{bmatrix} \gamma_0(2\pi) \\ \dot{\gamma}_0(2\pi) \end{bmatrix} + W^{n+1} \begin{bmatrix} a_0 \\ b_0 \end{bmatrix}.
 \end{aligned} \tag{21}$$

Now the vector $[a_{n+1} \ b_{n+1}]$ can remain bounded as n increases only if the sum

$$S_n = \sum_{i=0}^{n-1} W^i \tag{22}$$

remains bounded.

Let σ_{ni} be the eigenvalues of S_n and ω_i be those of W , where $i = 1, 2$. The eigenvalues of the matrix polynomial, Eq. (22) have the relation

$$\begin{aligned}
 \sigma_{ni} &= (1 - \omega_i^n) / (1 - \omega_i), & \omega_i &\neq 0, 1 \\
 &= 0, & \omega_i &= 0 \\
 &= n, & \omega_i &= 1
 \end{aligned}$$

Thus, the definition

$$w = \max(|\omega_1|, |\omega_2|) \tag{23}$$

and relation

$$0 < w < 1 \tag{24}$$

imply

$$|\sigma_{ni}| \rightarrow |1/(1-w_i)| < 1/(1-w) \quad \text{as } n \rightarrow \infty.$$

But if the relation, Eq. (24), is satisfied the second term of Eq. (21) is also bounded, and the amplitude of y is limited to

$$|y| \leq \frac{1}{1-w} \sqrt{\gamma_0 (2\pi)^2 + \dot{\gamma}_0 (2\pi)^2} + (\bar{\gamma}_1 + \bar{\gamma}_2) \sqrt{a_0^2 + b_0^2} \quad (25)$$

where the bars are defined in Eq. (11).

The derivation of the bound, given by Eq. (25), did not use the fact of resonance, thus it is applicable even to the non-resonant case, although in the latter case the bound given by Eq. (14) will usually be sharper. (It should be borne in mind that the symbols a_0 and b_0 have different significances in the two expressions.)

1.3 Stability Criteria

The eigenvalues of the matrix W satisfy the equation

$$0 = \begin{vmatrix} \omega - \gamma_1(2\pi) & -\gamma_2(2\pi) \\ -\dot{\gamma}_1(2\pi) & \omega - \dot{\gamma}_2(2\pi) \end{vmatrix} = \omega^2 - [\gamma_1(2\pi) + \dot{\gamma}_2(2\pi)]\omega + \det W$$

so that

$$\omega = \frac{\gamma_1(2\pi) + \dot{\gamma}_2(2\pi)}{2} \pm \sqrt{\left[\frac{\gamma_1(2\pi) + \dot{\gamma}_2(2\pi)}{2} \right]^2 - \det W} \quad (26)$$

When the eigenvalues are complex, they are conjugates, so

$$\begin{aligned} \omega_1 \omega_2 &= \left[\frac{\gamma_1(2\pi) + \dot{\gamma}_2(2\pi)}{2} \right]^2 + \det W - \left[\frac{\gamma_1(2\pi) + \dot{\gamma}_2(2\pi)}{2} \right]^2 \\ &= \det W, \end{aligned}$$

and thus

$$|\omega_1| = \sqrt{\det W}.$$

When the eigenvalues are real, let the one larger in absolute value be ω . Then

$$|\omega| \geq \left| \frac{y_1(2\pi) + \dot{y}_2(2\pi)}{2} \right| \geq \sqrt{\det W}.$$

For bounded motions, therefore, it is necessary that

$$\det W \leq 1.$$

It is to be noted that $\det W$ is the Wronskian of the system (y_1, y_2) , evaluated at $t = 2\pi$. Abel's formula for the Wronskian is:

$$W(x) = W(0) e^{\int_0^x \alpha(t) dt},$$

Thus

$$\det W = W(2\pi) = \begin{vmatrix} y_1(0) & y_2(0) \\ \dot{y}_1(0) & \dot{y}_2(0) \end{vmatrix} e^{\int_0^{2\pi} \alpha(t) dt} = e^{\int_0^{2\pi} \alpha(t) dt}.$$

Now if α is identically zero, or even if it has an average value of zero on the interval $(0, 2\pi)$,

$$\det W = 1.$$

Then, referring to Eq. (23),

$$w \geq 1.$$

It may be concluded, therefore, that in the absence of the damping term, the resonant solutions are unbounded unless Eqs. (10) happen to be dependent. The non-resonant solutions were shown above, Eq. (14), to be bounded, unless $w > 1$.

1.4 Short Term Boundedness

In the application of the results obtained above to Eq. (1) it must be recalled from Volume I, Appendix C of this series of reports, that the constants (or elements) describing the orbit are themselves subject to drift as the orbit decays. Thus the results are applicable to some large but finite number of periods. Under such a limitation an ultimately unbounded motion

may still be acceptable if its rate of growth is small. For the n th period the bound expressed by Eq. (25) may be written:

$$\bar{y}(t_{n-1}) \leq \frac{\omega^n - 1}{\omega - 1} \sqrt{y_0 (2\pi)^2 + \dot{y}_0 (2\pi)^2} + \omega^{n-1} (\bar{y}_1 + \bar{y}_2) \sqrt{y(0)^2 + \dot{y}(0)^2} \quad (27)$$

while in the non-resonant case the bound given by Eq. (14) becomes

$$\bar{y}(t_{n-1}) \leq \bar{y}_p + \omega^{n-1} \sqrt{[y_p(0) - y(0)]^2 + [\dot{y}_p(0) - \dot{y}(0)]^2}. \quad (28)$$

Now the total lifetime of the satellite may be divided into regions during each of which the parameters of the orbit may be considered constant, and during each of which the growth or decay of the pitch amplitude may be estimated by means of Eq. (27), or in a non-resonant case by Eq. (28).

1.5 A "Built-In" Check on the Accuracy of the Numeric Integration

Let Eq. (5) have a unique periodic solution y_p and let

$$y_p = y_e + y_o$$

where y_e is the even part of y_p and y_o is the odd part of y_p . Further, suppose that in Eq. (5) the functions α and γ are odd functions of t , and β is an even function of t . Then Eq. (5) becomes

$$(\ddot{y}_e + \alpha \dot{y}_e + \beta y_e) + (\ddot{y}_o + \alpha \dot{y}_o + \beta y_o) = \gamma.$$

The parity properties of functions guarantee that the first parenthesis on the left is an even function of t , while the second parenthesis is an odd function. Since the right member is an odd function, the first parenthesis must vanish. If there existed a non-trivial even function y_e the uniqueness of y_p would be destroyed. Thus

$$y_e \equiv 0,$$

and y_p is an odd function, so that $y_p(0) = 0$. Then in the solution of Eqs. (10):

$$a = 0,$$

and

$$y_p = y_o + b y_2.$$

This situation arises in Eq. (1) when the damping in pitch parameter, Γ_q , vanishes. In this case the size of a computed from Eqs. (10) becomes a measure of the accuracy of the numeric integration used to obtain the coefficients in these equations.

2. INTEGRATION PROCEDURE

The choice of integration procedure involves a balance between accuracy, speed, stability, and auxiliary "housekeeping" requirements. Gill, (Ref. 3) has derived a very elegant version of the Runge-Kutta fourth order procedure in which the housekeeping is minimized, in that only one storage cell need be assigned to each variable to be integrated. Milne (Ref. 4, Section 38) earlier gave a predictor-corrector method which obtains the same accuracy with half as many substitutions into the differential equation, but requires that a history of past values of the variables be kept. Milne's predictor formula requires a history of four previous values of the derivative, and if it is desired to double the step-size in the course of the integration a history of seven previous values must be available, along with a record of the time at which the step size was last doubled. The starting procedure for the predictor-corrector method is necessarily relatively involved. Since the corrector formula, Simpson's Rule, requires a history of only two previous values, it would be desirable to have a predictor formula with an equally short time-base, even at the cost of some accuracy and stability. Milne (Ref. 4, Sections 30 and 31) gives a procedure for deriving such formulas to arbitrary specifications and for estimating their accuracy. The procedure is here used to obtain the desired formula.

2.1 Derivation of Predictor Formula

In the identity:

$$y(t_0 + h) - y(t_0 - h) = \int_{t_0 - h}^{t_0 + h} \dot{y} dt \quad (29)$$

the assumption that the second term on the right is a linear combination of $y(t_0)$, $y(t_0 - h)$, $h\dot{y}(t_0)$ and $h\dot{y}(t_0 - h)$, may be written

$$\int_{t_0 - h}^{t_0 + h} \dot{y} dt = A y(t_0) + B y(t_0 - h) + C h \dot{y}(t_0) + D h \dot{y}(t_0 - h). \quad (30)$$

The coefficients A, B, C, D may be determined by requiring that Eq. (30) be exact for the first four terms of the Taylor expansion of about . Thus, let

$$y(t) = a_0 + a_1(t - t_0) + a_2 \frac{(t - t_0)^2}{2} + a_3 \frac{(t - t_0)^3}{6} + \dots \quad (31)$$

where

$$a_n = \frac{d^n y(t_0)}{dt^n} .$$

Substitution of Eq. (31) into Eq. (29) and Eq. (30) gives respectively

$$\int_{t_0-h}^{t_0+h} \dot{y} dt \cong 2a_1 h + \frac{1}{3} a_3 h^3,$$

$$\int_{t_0-h}^{t_0+h} y dt \cong A a_0 + B (a_0 - a_1 h + \frac{a_2}{2} h^2 - \frac{a_3}{6} h^3)$$

$$+ C a_1 h + D (a_1 h - a_2 h^2 + \frac{a_3}{2} h^3).$$

Equating the coefficients of a_0 , a_1 , a_2 , and a_3 respectively, yields:

$$\left. \begin{aligned} A + B &= 0 \\ C + D - B &= 2 \\ \frac{1}{2} B - D &= 0 \\ \frac{1}{2} D - \frac{1}{6} B &= \frac{1}{3} \end{aligned} \right\}.$$

The solution to this system is

$$A = -4, \quad B = 4, \quad C = 4, \quad D = 2.$$

Then Eq. (30) becomes

$$\int_{t_0-h}^{t_0+h} \dot{y} dt \cong -4y(t_0) + 4y(t_0-h) + 4h\dot{y}(t_0) + 2h\dot{y}(t_0-h),$$

which yields in Eq. (29):

$$\begin{aligned} y(t_0+h) &\cong y(t_0-h) + 4 \left\{ y(t_0-h) - y(t_0) + \frac{h}{2} [\dot{y}(t_0-h) + 2\dot{y}(t_0)] \right\} \\ &= y_p(t_0+h) \end{aligned} \quad (32)$$

where the subscript p stands for "predicted value". Its derivation insures that Eq. (32) is exact to third order terms.

The Taylor series, Eq. (31) may be made exact by including the residue term:

$$y(t) = a_0 + a_1(t-t_0) + \frac{a_2(t-t_0)^2}{2} + \frac{a_3(t-t_0)^3}{6} + \frac{d^4 y(s)}{dt^4} \cdot \frac{(t-t_0)^4}{24} \quad (33)$$

where s is on the interval $[t_0, t]$.

Now the error committed by the use of $y_p(t_0+h)$ in place of $y(t_0+h)$ may be obtained by subtracting the right member of Eq. (32) from its left member. If each term in this difference is expressed in terms of Eq. (33) or its first derivative, the coefficients of a_0 , a_1 , a_2 , and a_3 will be zero because Eq. (32) is accurate to third order terms. The residue terms will remain:

$$y(t_0+h) - y_p(t_0+h) = \frac{h^4}{24} \left[\frac{d^4 y(s_1)}{dt^4} - 5 \frac{d^4 y(s_2)}{dt^4} + 8 \frac{d^4 y(s_3)}{dt^4} \right],$$

where s_1 is between t_0 and t_0+h , while s_2 and s_3 are between t_0 and t_0-h . The three distinct values arise from the fact that s in Taylor's Theorem, Eq. (33), cannot be expected to be the same for the expansions of $y(t_0+h)$, $y(t_0-h)$, and $\dot{y}(t_0-h)$. However, Eq. (32) may be shown to satisfy Milne's Theorem 2, (Ref. 4, Section 31) which in effect permits the replacement of s_1 , s_2 , and s_3 by a single s between t_0-h and t_0+h . The error becomes

$$\frac{d^4 y(s)}{dt^4} \cdot \frac{h^4}{6},$$

and Eq. (32) may be rewritten:

$$y(t_0+h) = y(t_0-h) + 4 \left\{ y(t_0-h) - y(t_0) \right. \\ \left. + \frac{h}{2} [\dot{y}(t_0-h) + 2\dot{y}(t_0)] \right\} + \frac{d^4 y}{dt^4} \cdot \frac{h^4}{6}. \quad (34)$$

2.2 Corrector Formula

Simpson's rule is

$$y(t_0+h) \cong y(t_0-h) + \frac{h}{3} [\dot{y}(t_0-h) + 4\dot{y}(t_0) + \dot{y}(t_0+h)] = y_s(t_0+h) \quad (35)$$

where the subscript s denotes "calculated from Simpson's rule". An error analysis for Simpson's rule yields

$$y(t_0+h) = y(t_0-h) + \frac{h}{3} [\dot{y}(t_0-h) + 4\dot{y}(t_0) + \dot{y}(t_0+h)] - \frac{d^5 y}{dt^5} \cdot \frac{h^5}{90}. \quad (36)$$

In the derivation of the error term in Eq. (35) the assumption was made that the other terms on the right-hand side are known precisely. If this assumption

is replaced by the assumption that $\dot{y}(t_0+h)$ was itself computed by Eq. (32) the error term is larger, in fact

$$y(t_0+h) = y(t_0-h) + \frac{h}{3} [\dot{y}(t_0-h) + 4\dot{y}(t_0) + \dot{y}_p(t_0+h)] + 4 \frac{d^5 y}{dt^5} \cdot \frac{h^5}{90} \quad (37)$$

If, however, $\dot{y}_s(t_0+h)$ is used in place of $\dot{y}(t_0+h)$ the error term is about

$$\frac{6}{5} \cdot \frac{d^5 y}{dt^5} \cdot \frac{h^5}{90} \quad .$$

2.3 Integration Scheme

The predictor-corrector scheme is then:

- a) With $y = \theta'$ use Eq. (34) to predict $\theta'(v+\Delta v)$ from $\theta'(v)$, $\theta'(v-\Delta v)$, $\theta''(v)$, $\theta''(v-\Delta v)$.
- b) With $y = \theta$ use Eq. (36) to predict $\theta(v+\Delta v)$ from $\theta(v-\Delta v)$, $\theta'(v-\Delta v)$, $\theta'(v)$ and the predicted $\theta'(v+\Delta v)$.
- c) Use the predicted $\theta(v+\Delta v)$ and $\theta'(v+\Delta v)$ in Eq. (1) to predict $\theta''(v+\Delta v)$.
- d) Use the predicted $\theta''(v+\Delta v)$ in Eq. (35) with $y = \theta'$ to calculate the corrected $\theta'(v+\Delta v)$.
- e) Use the corrected $\theta'(v+\Delta v)$ in Eq. (35) with $y = \theta$ to calculate the corrected $\theta(v+\Delta v)$.
- f) Use the corrected $\theta(v+\Delta v)$ and $\theta'(v+\Delta v)$ in Eq. (1) to calculate the corrected $\theta''(v+\Delta v)$.

The small size of the error term in Eq. (36) makes it unlikely that an iteration of steps d, e, f would improve the result, if the prediction of steps a, b, c is at all adequate. This adequacy may be tested by comparing the predicted and corrected values of $\theta''(v+\Delta v)$. Since the error estimates for the prediction exceed those for the correction the difference between predicted and corrected values is a pessimistic estimate of the error incurred at each step of the integration. Let $\epsilon(\Delta v)$ be the magnitude of the difference between the predicted and corrected values of $\theta''(v+\Delta v)$, and let $\bar{\epsilon}$ be the maximum tolerable difference. Then if

$$\epsilon(\Delta v) \leq \bar{\epsilon} / 32$$

the error estimates for Eq. (35) suggest that

$$\epsilon(2\Delta v) \leq \bar{\epsilon}.$$

Thus a sufficiently small ϵ justifies a doubling of the step size. If, however

$$\epsilon(\Delta v) > \bar{\epsilon}$$

the integration must be restarted with a smaller value of Δv .

Starting, as well as restarting may be accomplished by Taylor's formulas

$$\theta'(v + \Delta v) = \theta'(v) + \theta''(v)\Delta v + \frac{d^3\theta(s)}{dv^3} \cdot \frac{(\Delta v)^2}{2}, \quad (38)$$

$$\theta(v + \Delta v) = \theta(v) + \theta'(v)\Delta v + \frac{\theta''(v)(\Delta v)^2}{2} + \frac{d^3\theta(s)(\Delta v)^3}{6}, \quad (39)$$

where s is between v and $v + \Delta v$. The relatively low powers of Δv appearing in the error terms of Eq. (38) and (39) suggest that small values of Δv be used to retain accuracy. Since the predictor-corrector scheme makes it easy to double the step size there is but small penalty for setting the initial value of Δv too small.

Computational experiments have been conducted comparing the procedure presented above with Gill's version (Ref. 3) of the Runge-Kutta fourth order procedure and with the Kutta fourth order formula as quoted by Milne (Ref. 5). All three processes gave essentially the same accuracy for the same step size; as mentioned above, the present process requires half as many substitutions into the differential equation. It should, however, be noted that the relatively large coefficients appearing in Eq. (34) tend to favor the build up of "inherited" errors, so that the present method cannot be expected to be as stable as the Runge-Kutta process for differential equations with unstable solutions.

The potential build-up of inherited error is not of concern in the present study for three reasons. Firstly, the range of integration is relatively short so that inherited error cannot build up very far. Secondly, the most critical cases are the most unstable cases, but in these the demonstration of instability is obvious. Thirdly, application of the accuracy criterion developed in Section 1.5 to the computer results justifies, a posteriori, the assertion that the error build-up was nil to four decimal places.

3. COMPUTER PROGRAM

Program Descriptions of the various components of the Computer Program written to calculate solutions of Eq. (1) are included in Appendices A to F. Sample input and output data are in Appendix G. The program has been checked out and used at the North American Aviation, Inc. Columbus IBM 704 computing installation to generate the numerical results discussed in Volume I of this series of reports. Certain systems features of this program, such as the choice of input medium (magnetic tape or punch-cards), assignment of magnetic tape unit numbers, labelling of library routines, etc. may have to be varied if this program is to operate successfully at other installations or on other computing machines. It is believed that the information furnished is adequate to enable anyone familiar with the IBM Fortran Manuals (for instance Ref. 6 and 7), and the operating system of his installation to adapt the source language deck to that installation.

The program was written in two portions. The first portion integrates the differential equation, Eq. (1), and its homogeneous part, and records the results on a magnetic tape as well as on the output medium. The second portion reads the magnetic tape and produces punched cards suitable for automatic plotting. The portions are discussed separately below and a brief discussion of the atmosphere table look-up routine is also given. This section concludes with some remarks on a potential application of the computer program.

3.1 Integration Portion of the Program

All input data is read by means of a "sparse data" routine so that information unchanged from one case to the next need not be written or key-punched repetitively. The only item of input data modified by the program is the identification number used to identify records on the intermediate storage tape, which is automatically advanced from record to record. To further facilitate the preparation of input data the primary input variables are presented in lists; the program systematically uses each combination of list entries. All angles in input and output are expressed in degrees; however, inside the computer and on the intermediate storage tape they are carried in radian units.

In the integration of the differential equations care is taken to provide common abscissas from one integration to the next. This is accomplished by using a computing interval obtained from the print interval by successive halving, and by not doubling the step size (when doubling is permitted by the criteria of Section 2) if this would cause the end of the print interval to fall inside a computing interval. The accuracy of the integration is further insured by a test which aborts the calculation if the step size does not expand in the first three intervals. If the step size fails to expand, the accuracy of the starting formulas Eqs. (38), (39) is less than that implied by the predictor-corrector method. The calculation should be attempted again with a smaller minimum step size. The integration is also aborted if the magnitude of any of the θ 's exceeds 10^7 , such an event would imply a gross data error, or a machine failure. The number of computing steps taken during

each print interval is printed out so that the step size used may be deduced, the total number of steps for each integration is printed out as an aid to future computing time estimates. As compiled for the IBM 704, the program will integrate about 8.5 steps per second.

3.2 Card-Punching Portion of the Program

This section may be combined with the first section in a single computer run, or it may be used separately if the intermediate storage tape is saved from run to run. The second option permits inspection and editing of the curves to be punched into cards. Since the on-line punching of cards is a relatively expensive process the card format was designed to admit up to fifteen curves per card deck, while retaining complete flexibility as to size and placement of the plots. This portion also reads its input data (specifying the editing information) with the "sparse data" routine. The only input item modified by the program is the card deck identification number which serves as a signal that all information to be punched has been accumulated. This number is reset to zero after the card deck has been punched to indicate that the store of information is again incomplete.

The punched-card output for automatic plotting assumes a plotting device capable of accepting one coordinate pair per card. Abscissa and ordinate may be placed in any of sixteen four digit fields, or the abscissa may be suppressed in favor of an abscissa punched for another curve. The abscissa field will contain the independent variable ψ . The ordinate may contain any function of the form

$$f = |w_p| \theta_0 + w_1 \theta_1 + w_2 \theta_2 + w_\gamma \gamma$$

where θ_0 is the particular solution of Eq. (1) corresponding to γ_0 of Section 1,

θ_1, θ_2 are the solutions of the homogeneous part of Eq. (2) corresponding to γ_1, γ_2 of Section 1,

γ is defined in Eq. (2)

and the w 's are input data.

For $w_\gamma = 0, w_p = 1, f$ will be a solution of the differential equation, Eq. (1). When $w_\gamma = -1, f$ will be the angle of attack, α . When $w_p = -1$, the input values of w_1 and w_2 will be ignored, instead the program will use the pre-computed values needed to make f periodic, of orbital period.

The card decks produced will contain, in addition to the abscissa-ordinate pairs, four terminating cards. The first two of these are calibration check cards which will position the plotting head to the lower left hand and upper right hand corners of the plot respectively, the third contains an 11-punch in Column 1 which may be used to stop the plotting device, and the fourth is completely blank, serving as a separator between decks. The body of the deck, and the first two terminating cards have a 0-punch in Column 1, sequence number in Columns 5-8, and a deck identification number in Columns 2-4. The four

digit fields consisting of Columns 9-12, 13-16, etc to 69-72 are numbered $F = 1, 2, \dots, 16$ respectively and are the data fields proper (Columns 73-80 are not readily accessible to the IBM 704 computer).

The tape file of data is searched completely for the record required by the input data, so that in principle these records could be demanded in any order. The search is most efficient however if the records are demanded in the order in which they are stored on the tape.

3.3 Atmospheric Properties Subroutine

The atmospheric properties routine used in the program is capable of simulating atmosphere tables based on the ideal gas law and on linear variations of temperature, gas constant, and specific heat ratio with geopotential altitude. In the present program the routine was used to obtain the density distribution of the 1959 ARDC (Minzner) Atmosphere, (Ref. 1), which was represented by use of the Molecular Scale Temperature in place of the usual temperature. The detail input data are given in Appendix G.

3.4 A Potential Application

In its present form the program can furnish information about the stability in pitch of a given satellite in a given orbit. If the detailed motion is not of interest, only the first portion of the program is required. For a design study, knowledge of the stability boundaries in various parameter planes of the differential equation, Eq. (1), would be desirable. These occur when the magnitude of the larger eigenvalue ω given by Eq. (26) passes through the value 1. With minor modifications the subroutine INTEG of the first part of the present computer program could be used as the basis of a program to search iteratively for these stability boundaries. Points of departure for these searches could be obtained from the eigenvalues of the sine problem to which the present problem reduces itself when the orbital eccentricity is zero.

A NOTE ON THE APPENDICES

Appendices A to E are presented in the format of a standard SHARE program description.

APPENDIX APROGRAM DESCRIPTION - PITCHING MOTION OF A SATELLITE
NORTH AMERICAN AVIATION, INC. COLUMBUS
ENGINEERING PROGRAM DESCRIPTION1. Identification

- a. Pitching Motion of a Satellite, IM 091
- b. O. C. Juelich, December 1960
- c. Research Group - 350
- d. Fortran Source Deck is up to date.

2. Purpose

Numerically integrates one particular and two homogeneous solutions of the pitch-equation. Solutions are printed, and optionally punched into cards in scaled form suitable for automatic plotting.

3. Restrictions

- a. Uses Tape 5 as input tape
Tape 6 as printer output tape
Tape 2 for intermediate storage of results
Tape 7 to store part of the program
- b. Operates in N.A.A. monitor for IBM 704 Fortran

4. Method

- a. The mathematical method is described in Report WADD TR 61-133 Vol. III, of which this description is a part.
- b. The program is designed to terminate calculations if the step size has not adequately expanded in the first three print intervals or if any of the solutions exceed 10 million degrees in amplitude.
- c. Punched Card output is described in Section 3.2 of this report.

5. Use

The program consists of two parts. The first part generates solutions of the differential equation and records these on magnetic tape. The second part retrieves selected solutions and scales them for automatic plotting. The two parts can be combined in one machine run or they can be used on separate occasions if the intermediate storage tape (tape 2) is preserved. The latter alternative permits inspection of the printed results before card-punching is undertaken.

6. Coding Information

- a. The first part consists of a main program and two subroutines labeled INTEG, DIFFC. In addition subroutine ATMOS (1F113) and Utility File

subroutine FXRCH are used, as well as Library subroutines SQRT, COS, SIN, CHAIN, and Fortran system routines.

- b. The second part is Chain 1 on Tape 7. It consists of a main program and a subroutine labelled SCALE. In addition Utility File Subroutines FXRCH, NPSCL, WRITE, are used, along with Fortran system routines.
- c. Descriptions of subroutines ATMOS, FXRCH, NPSCL, and WRITE are given separately.

7. Input Data

- a. All input data to both parts of the program is read by the Utility File Routine FXRCH into the Fortran COMMON Data Region.
- b. Part one requires the following data:

<u>Location</u>	<u>Symbol</u>	<u>Designation, Description</u>
7	dv_{min}	Minimum computing step size bound, degrees The minimum computing step size is the largest number of the form $dv_{print}/2^n$ (n integral) and $\leq dv_{min}$
9	dv_{print}	Print Interval, degrees Should be a factor of 360° .
28	RECID	Record Identification Number or Control Number. If this number is negative or zero the remaining data set is ignored. If RECID is negative part II is loaded and given control. If RECID is zero tape 2 is rewound and computation ends. If RECID is positive it becomes the identification number of the data case, if it is omitted the value of RECID from the previous data case increased by 1. is used.
30	r_E	Radius of Earth, feet Used in Eq. (4), but not in subroutine ATMOS.
31	μ	Product of Universal gravitational constant and mass of earth, ft^3/sec^2 Used in Eq. (3).
32	TBL	Table used for routine ATMOS (1F113) See Appendix B.

<u>Location</u>	<u>Symbol</u>	<u>Designation, Description</u>
201	n_i	Number of orbit inclinations,
202	i_1	first orbit inclination, degrees
203	i_2	second orbit inclination
et seq.		
211	n_h	Number of perigee altitudes,
212	h_1	first perigee altitude, feet
213	h_2	second perigee altitude
et seq.		
221	n_e	Number of eccentricities,
222	e_1	first eccentricity, dimensionless
223	e_2	second eccentricity
et seq.		
231	n_r	Number of pitching moment parameters,
232	r_1	first pitching moment parameter
233	r_2	second pitching moment parameter
et seq.		
241	n_{r_d}	Number of damping in pitch parameters
242	r_{d1}	first damping in pitch parameter
243	r_{d2}	second damping in pitch parameter
et seq.		
251	n_M	Number of inertia ratio parameters
252	M_1	first inertia ratio parameter
253	M_2	second inertia ratio parameter
et seq.		

The $n_i, n_h, n_e, n_r, n_{r_d}, n_M$ combinations of the listed values of i, h, e, r, r_d, M will be used systematically. The results will be identified by successive values of RECID.

c. Part two requires the following data:

<u>Location</u>	<u>Symbol</u>	<u>Designation, Description</u>
1.	RECID	Identification number of tape record to be plotted.
2.	w_p	Weight of function θ_0 in curve to be plotted Should be 1, 0, or -1.
3.	w_γ	Weight of $\gamma(v)$ in curve to be plotted Should be 0 or -1.
4.	w_1	Weight of function θ_1 in curve to be plotted Ignored when $w_p = -1$.
5.	w_2	Weight of function θ_2 in curve to be plotted Ignored when $w_p = -1$.

<u>Location</u>	<u>Symbol</u>	<u>Designation, Description</u>
6.	f_{min}	Minimum value of function in plotting interval, degrees If f is less than this value the point will be plotted on the bottom margin of the paper.
7.	C_{min}	Plotter counts for f_{min} .
8.	f_{max}	Maximum value of function in plotting interval, degrees If f exceeds this value the point will be plotted on the top margin of the paper.
9.	C_{max}	Plotter counts for f_{max} .
10.	C_{edge}	Plotter counts for top margin of the paper. The bottom margin is at zero counts.
11.	F_f	Number of the card field in which f is to appear. Should be between 1. and 16.
12.	v_{min}	Minimum value of v in plotting interval, degrees If v is less than this value the point will be plotted on the left margin of the paper.
13.	d_{min}	Plotter counts for v_{min}
14.	v_{max}	Maximum value of v in plotting interval, degrees. If v exceeds this value the point will be plotted on the right margin of the paper.
15.	d_{max}	Plotter counts for v_{max}
16.	d_{edge}	Plotter counts for right margin of the paper. The left margin is at zero counts.
17.	F_v	Number of the card field in which v is to appear, or control number. If F_v is zero, v will not be entered on the punched cards.
18.	CID	Identification number for card deck to be punched, or control number. If CID is zero the scaled data are accumulated in core storage for later punching.

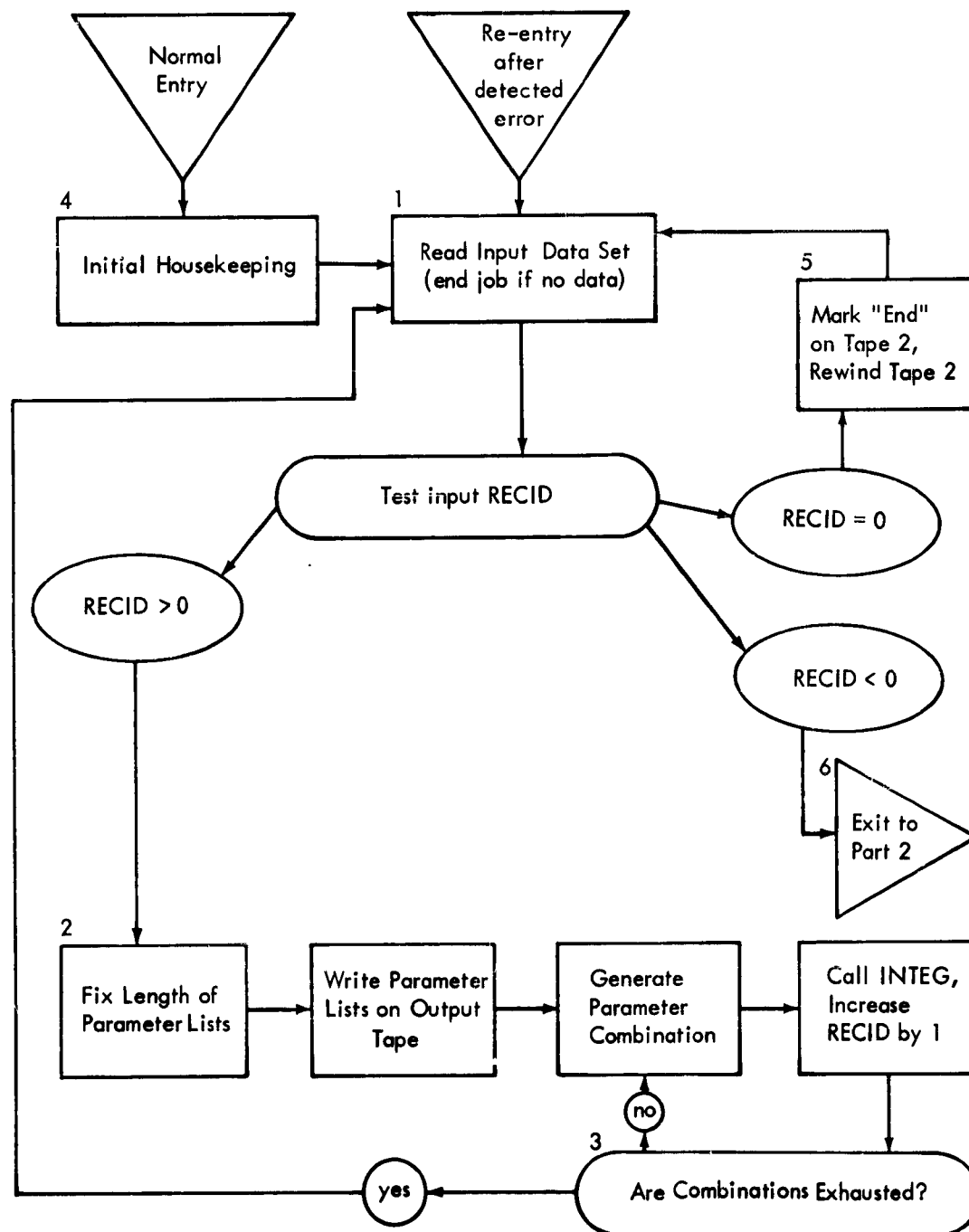
<u>Location</u>	<u>Symbol</u>	<u>Designation, Description</u>
-----------------	---------------	---------------------------------

If CID is not zero the accumulated information is punched into cards and CID becomes an identification number in columns 2 to 4 of each card. After punching CID is reset to zero.

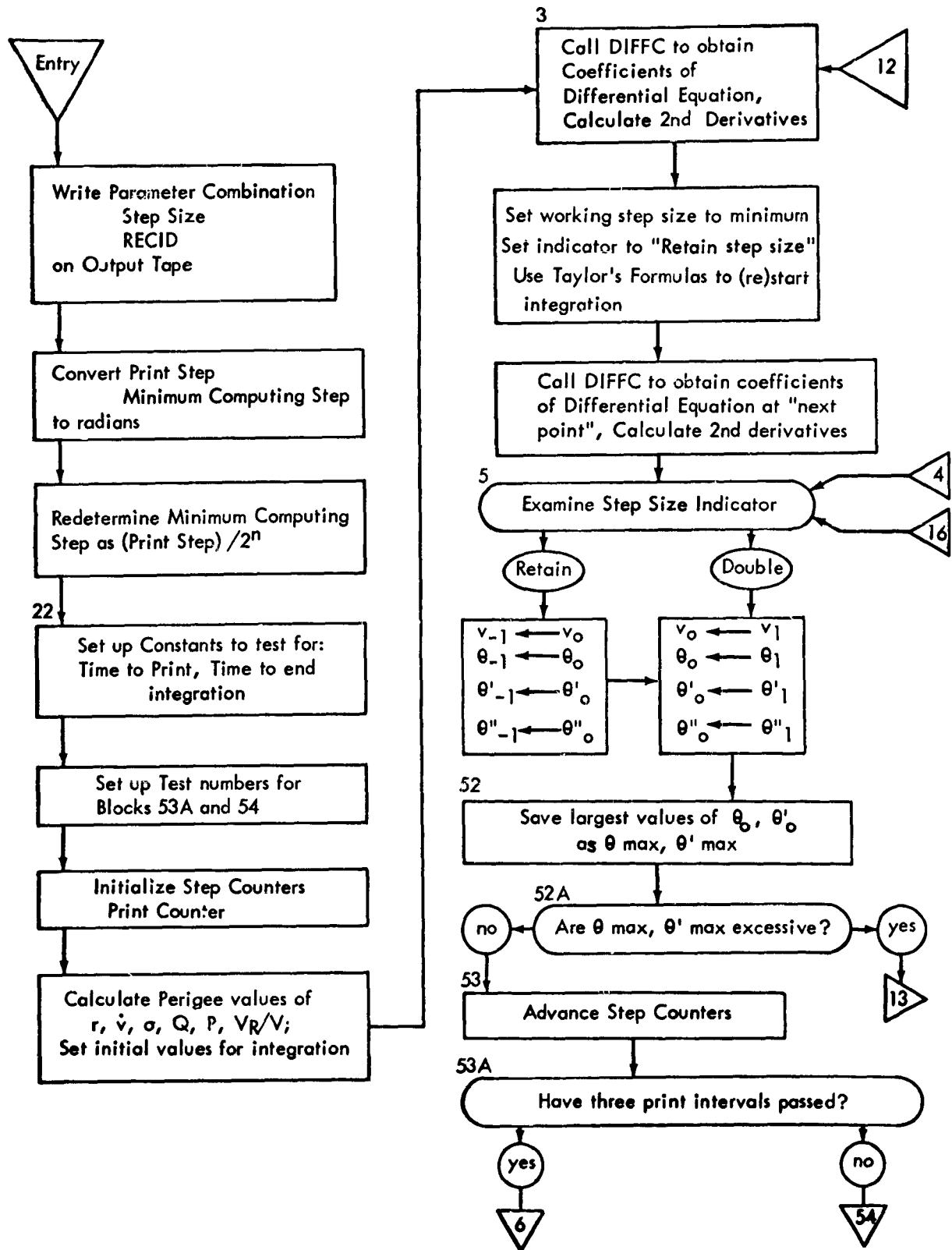
- d. Sample Input Data with resulting output follow the compiled listings of the components of this program.

Note: The program symbol for v is PHI, for *det (I-W)* is DELTA.

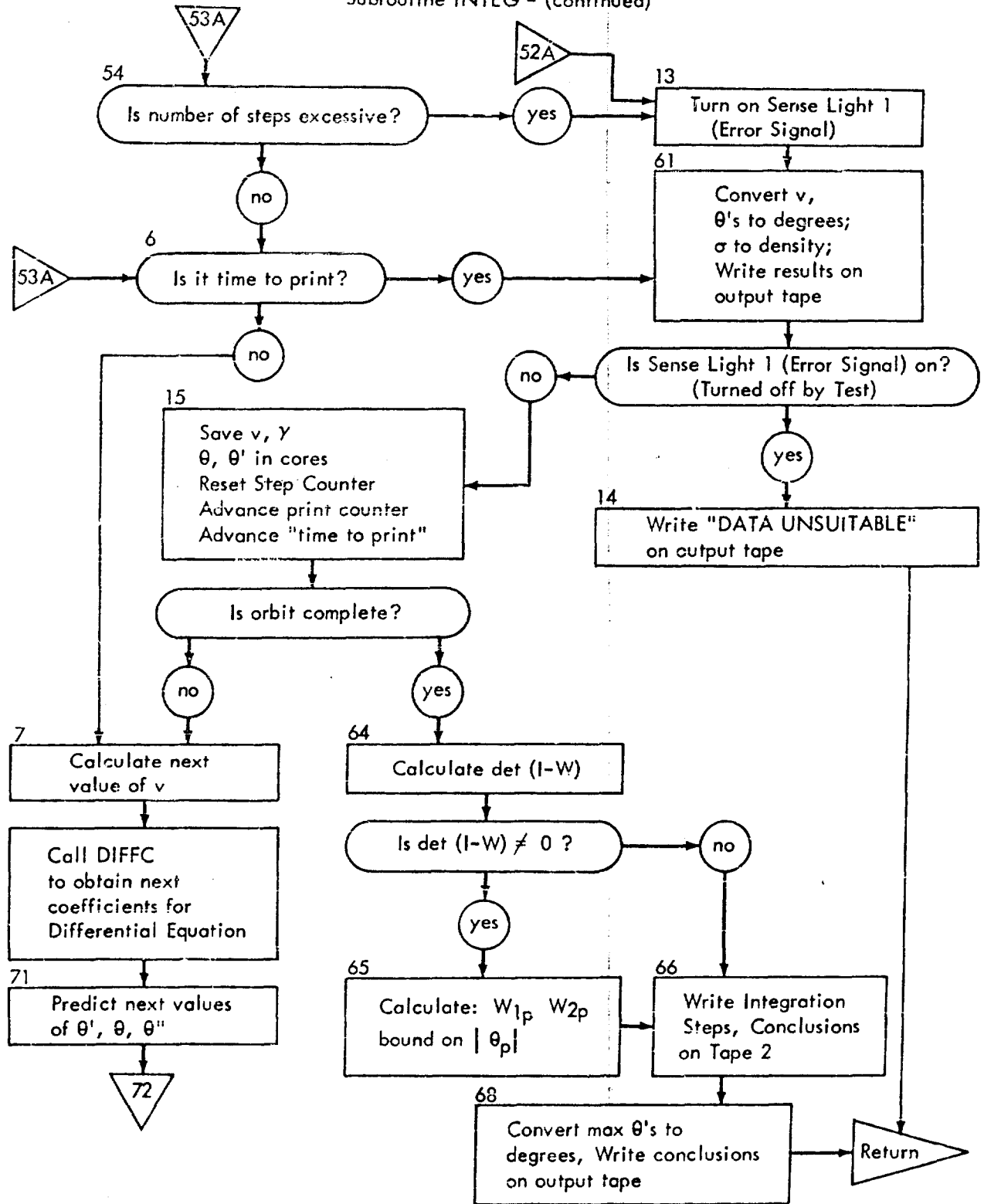
Main Program for Part I



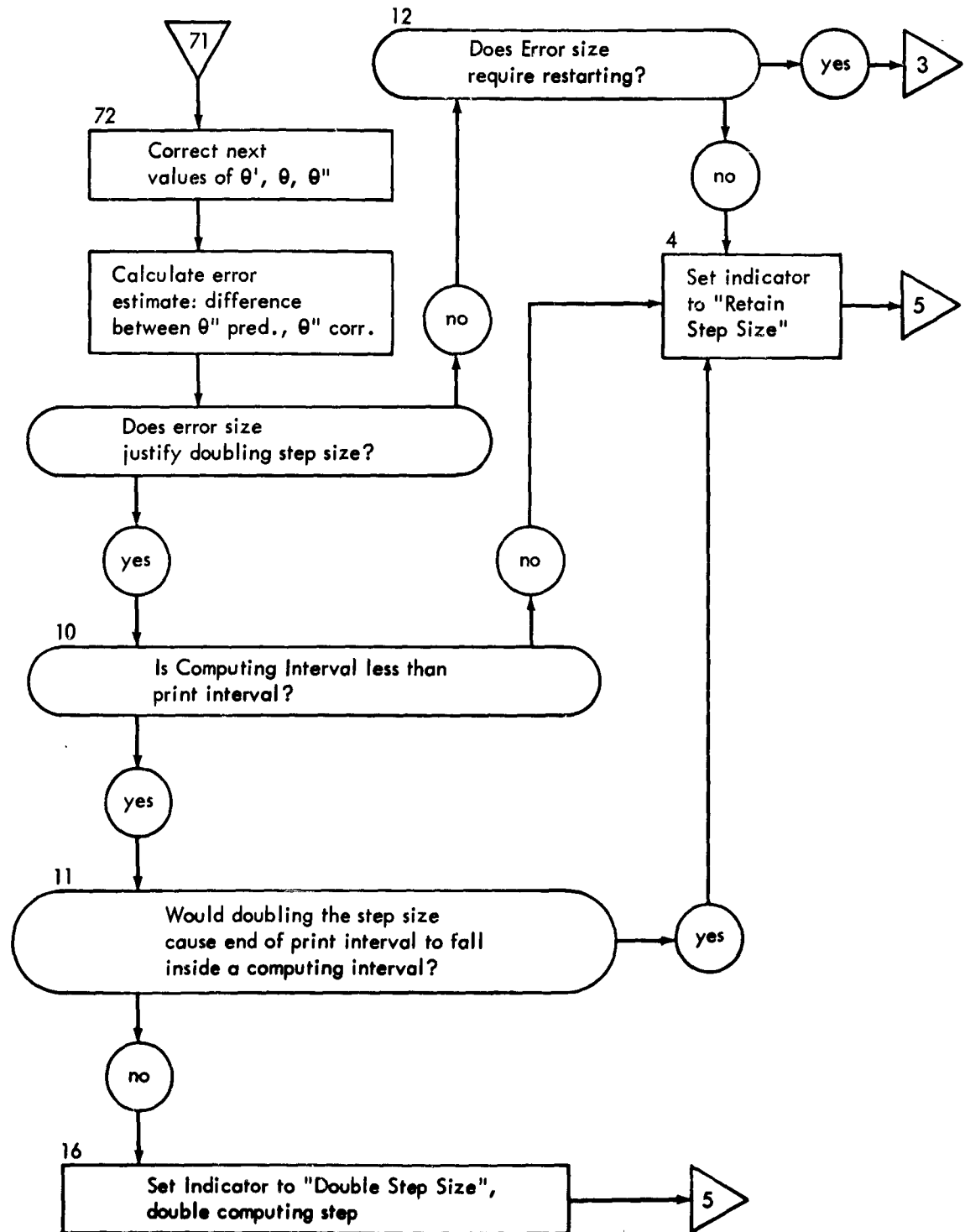
Subroutine INTEG



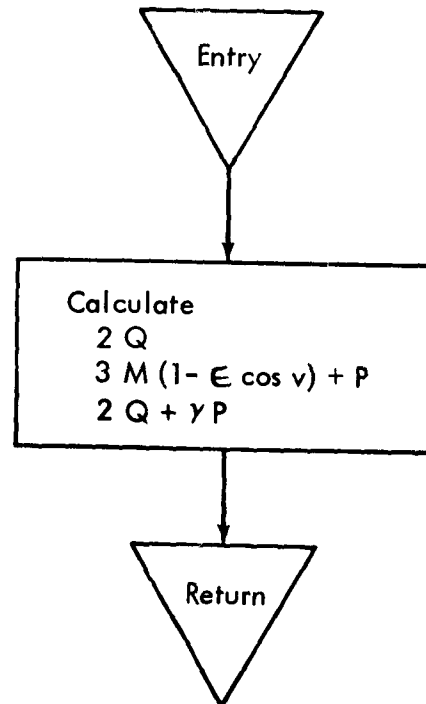
Subroutine INTEG - (continued)



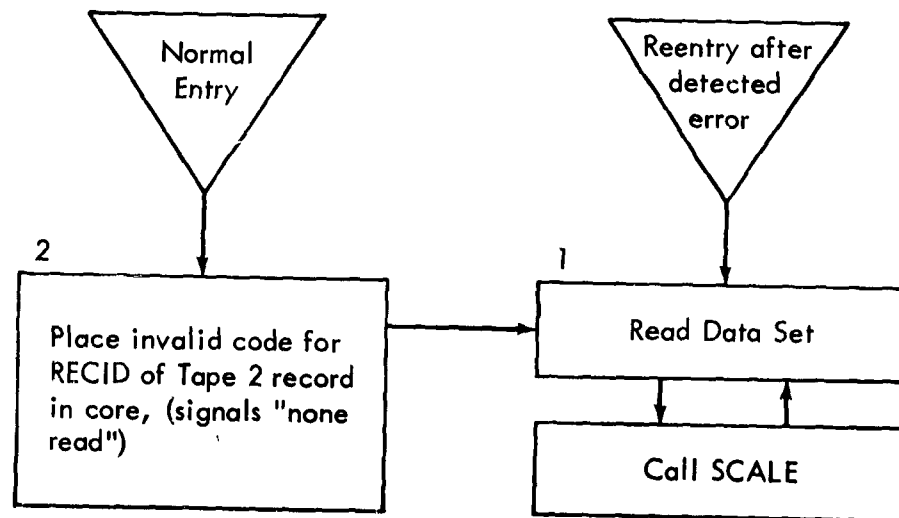
Subroutine INTEG - (concluded)



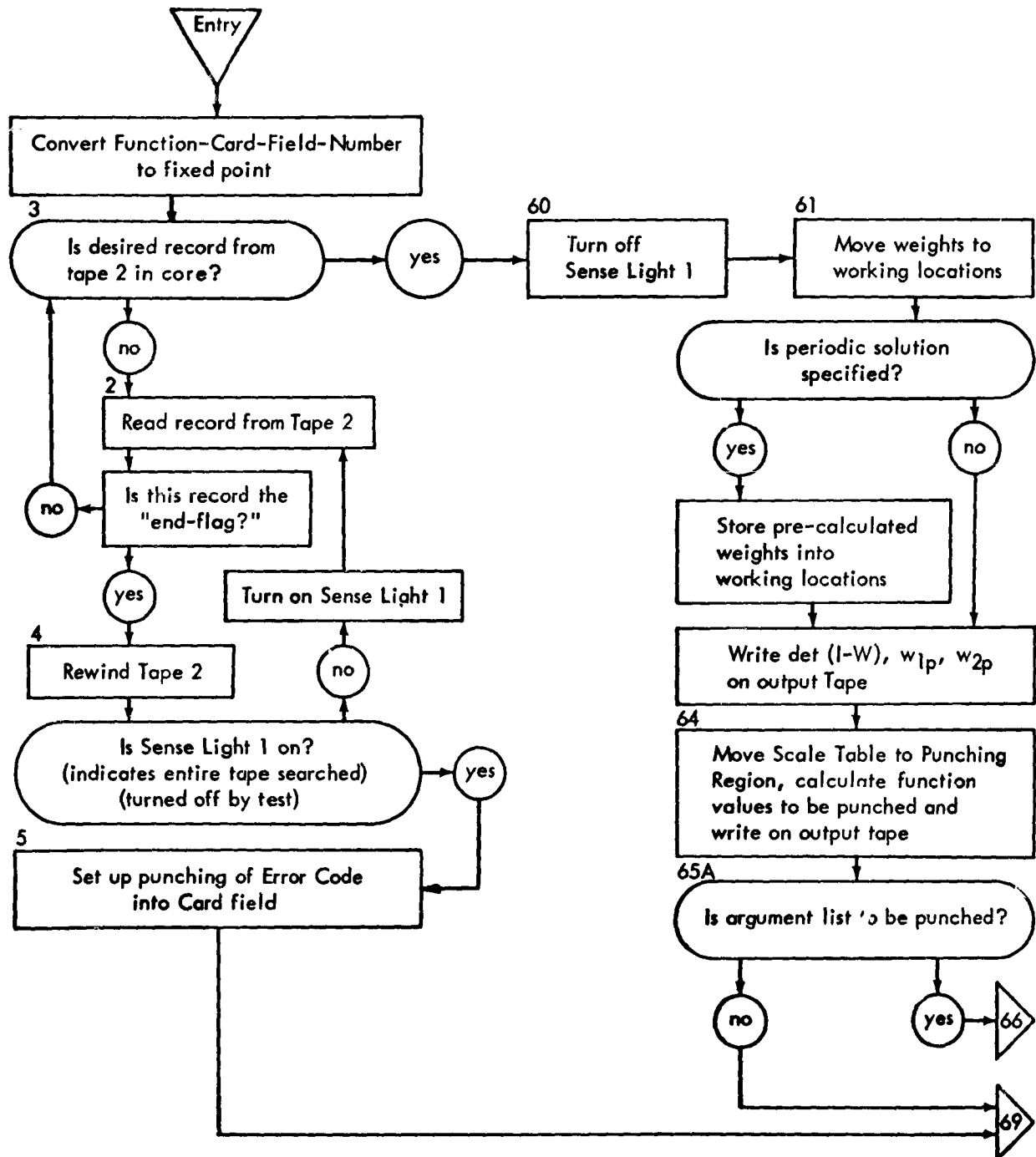
Subroutine DIFFC



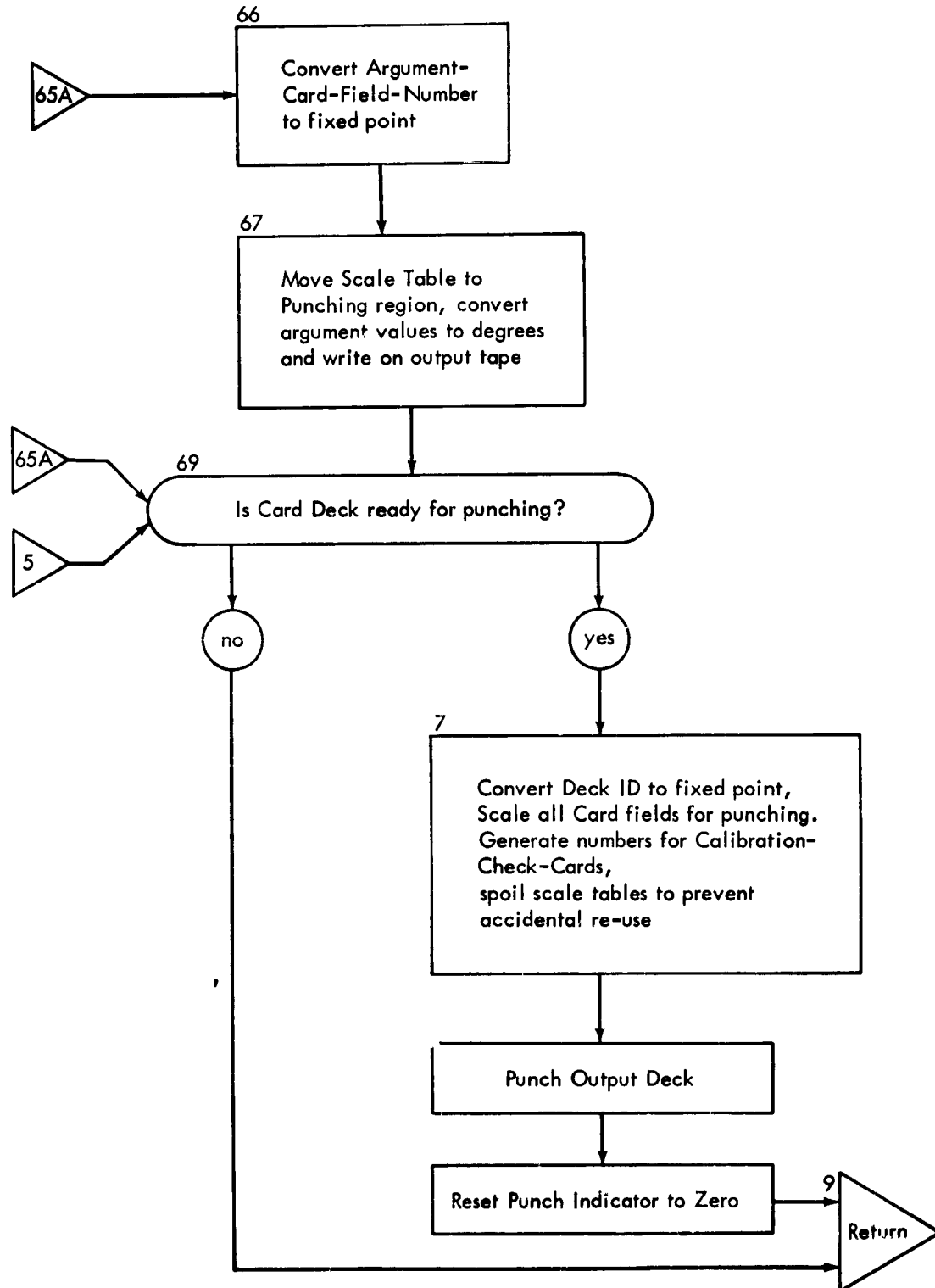
Main Program for Part II



Subroutine SCALE



Subroutine SCALE - (concluded)



C	MAIN PROGRAM TO VARY THROUGH LISTS OF VARIABLES	1M091300
	DIMENSION C(260),D(260)	1M091302
	COMMON C	1M091304
	EQUIVALENCE (C,D)	1M091306
	IF DIVIDE CHECK 1,4	1M091307
4	REWIND 2	1M091308
	YD=-1.	1M091309
	NS=1	1M091310
1	CALL FXRCH(D)	1M091312
	IF (C(28)) 6,5,2	1M091314
2	CONTINUE	1M091316
	IM=D(201)	1M091318
	JM=D(211)	1M091320
	KM=D(221)	1M091322
	LM=D(231)	1M091324
	MM=D(241)	1M091326
	NM=D(251)	1M091328
	WRITE OUTPUT TAPE 6,11,(D(I+201),I=1,IM)	1M091330
	WRITE OUTPUT TAPE 6,12,(D(I+211),I=1,JM)	1M091332
	WRITE OUTPUT TAPE 6,13,(D(I+221),I=1,KM)	1M091334
	WRITE OUTPUT TAPE 6,14,(D(I+231),I=1,LM)	1M091336
	WRITE OUTPUT TAPE 6,15,(D(I+241),I=1,MM)	1M091338
	WRITE OUTPUT TAPE 6,16,(D(I+251),I=1,NM)	1M091340
	DO 3 I=1,IM	1M091342
	DO 3 J=1,JM	1M091344
	DO 3 K=1,KM	1M091346
	DO 3 L=1,LM	1M091348
	DO 3 M=1,MM	1M091350
	DO 3 N=1,NM	1M091352
	C(1)=D(I+201)	1M091354
	C(2)=D(J+211)	1M091356
	C(3)=D(K+221)	1M091358
	C(4)=D(L+231)	1M091360
	C(5)=D(M+241)	1M091362
	C(6)=D(N+251)	1M091364
	CALL INTEG	1M091366
	C(28)=C(28)+1.	1M091367
3	CONTINUE	1M091368
	GO TO 1	1M091370
11	FORMAT (18H1 PARAMETER LIST -/7H0 OI =(9F10.2))	1M091372
12	FORMAT (7H0 HP =(9F10.0))	1M091374
13	FORMAT (7H0 EC =(9F10.6))	1M091376

14 FORMAT (7H0 GAM =(9F10.3))
15 FORMAT (7H0GAMQ =(9F10.3))
16 FORMAT (7H0 OM =(9F10.6))
5 WRITE TAPE 2,YD,NS,(C(I),I=1,15)
END FILE 2
REWIND 2
GO TO 1
6 C(8)=CHAINF(1,7)
STOP
END(0,0,0,0,0)

1M091378
1M091380
1M091382
1M091384
1M091386
1M091388
1M091390
1M091392
1M091394

STORAGE FOR VARIABLES APPEARING IN COMMON SENTENCES

DEC OCT DEC OCT DEC OCT DEC OCT
 D 32562 77462 C 32562 77462

EXTERNAL FORMULA NUMBERS WITH CORRESPONDING INTERNAL FORMULA NUMBERS AND OCTAL LOCATIONS

EFN	IFN	LOC	EFN	IFN	LOC	EFN	IFN	LOC	EFN	IFN	LOC
4	5	00012	4	5	00012	1	8	00017	1	9	00017
2	11	00027	3	68	00343	11	70	00000	12	71	00000
13	72	00000	14	73	00000	15	74	00000	16	75	00000
5	76	00366	5	82	00400	6	86	00405			

STORAGE NOT USED BY PROGRAM

DEC OCT DEC OCT
 317 00475 32302 77056

LOCATIONS OF NAMES IN TRANSFER VECTOR

DEC	OCT	DEC	OCT	DEC	OCT	DEC	OCT
INTEG	2 00002	FXRCH	7 00007	CHAIN	0 00000	(FIL)	3 00003
(IOH)O	5 00005	(LEV)	6 00006	(RTN)	1 00001	(STH)	4 00004

STORAGE LOCATIONS FOR VARIABLES NOT APPEARING IN DIMENSION EQUIVALENCE OR COMMON SENTENCES

DEC	OCT	DEC	OCT	DEC	OCT	DEC	OCT
IM	316 00474	JM	315 00473	KM	314 00472	LM	313 00471
MM	312 00470	NM	311 00467	NS	310 00466	YD	309 00465

STORAGE LOCATIONS FOR SYMBOLS NOT APPEARING IN SOURCE PROGRAM

DEC	OCT	DEC	OCT	DEC	OCT	DEC	OCT
E)10K	198 00306	E)10J	191 00277	D)20L	203 00313	D)20K	199 00307
D)20J	192 00300	D)20I	185 00271	C)G5	308 00464	C)G4	307 00463
C)G3	306 00462	C)G2	305 00461	C)G1	304 00460	8)G	280 00430
8)F	284 00434	8)E	288 00440	8)D	292 00444	8)C	296 00450
8)B	303 00457	2)	269 00415	3)	271 00417	6)	272 00420

SUBROUTINES NOT PUNCHED FROM LIBRARY

FXRCH	(LEV)	(IOH)O	(STH)	(FIL)	INTEG	(RTN)	CHAIN

```

SUBROUTINE INTEG
  PREDF(YL,YO,YPL,YPO)=YL+4.*(YL-YO+DPC*(YPL+YPO
  2 +YPO)/2.)
  SIMPF(YL,YPL,YPO,YPN)=YL+DPC*(YPL+4.*YPO+YPN)/3.
  RELERF(A,B)=ABSF(A-B)/MAX1F(ABSF(A),ABSF(B),1.E-7)
  DIMENSION C(200),VMAX(2,3),PHI(3),V(3,4,3),CO(3),
  2 SAV(2,4,360),PRBL(8)
  COMMON C
  EQUIVALENCE (C(32),TBL(1),SIG(14),R(15),H(16),QP(17),
  2 PP(18),VRVV(19),SIGP(20),THPC(21),THO(22),PHIO(23),
  3 EC(30),HP(31),OI(32),PHMAX(5),OM(27),GAMQ(28),GAM(29))
  4 ,(GAMMA(4),RE(3),OMU(2),C(32) ),(N,PRBL(8) )
  WRITE OUTPUT TAPE 6,1,(C(I),I=1,7),C(9),C(28)
  FORMAT (1H18X2H0113X2HHP13X2HEC13X3HGAM11X4HGAMQ12X
  2 2HOM/F13.2,F15.0,F15.0,1P17.4,E15.4,0PF13.6/1H0
  3 4X9HD PHI MIN5X11HD PHI PRINT51X9HRECORD ID/
  4 1P2E15.4,0PF60.0/1H08X3HPHI12X1HH7X3HRHO16X5HGAMMA
  5 6X8HALPHA H16X8HALPHA H2 6X8ALPHA PA6X11HN)
  DPPR=C(9)/57.295780
  DPLIM=C(7)/57.295780
  DPMIN=DPPR
  DPTST=DPMIN/2.
  23 IF (DPMIN-DPLIM) 22,22,21
  21 DPMIN=DPMIN/2.
  GO TO 23
  22 PHPR=DPPR-DPTST
  PHEND=6.2831853-DPTST
  FNRA=DPPR/DPTST
  NRAT=FNRA
  N=0
  NT=0
  NS=1
  RP=HP+RE
  PHDP=SQRTF(OMU*(1.+EC)/RP)/RP
  CALL ATMOS (HP,TBL,R)
  SIGP=SIG
  QP=GAMQ*SIG*RP*2.9711463E-4
  PP=GAM*SIG*RP**2*1.1884585E-3
  VRVV=1.-7.292 1151E-5*COSF(OI/57.295780)/PHDP
  PHI(2)=0.
  V(1,2,1)=1.7453293E-2
  VMAX(1,1)=1.7453293E-2

```

```

1M091416
1M091418
1M091420
1M091422
1M091424
1M091426
1M091427
1M091428
1M091430
1M091432
1M091434
1M091436
1M091438
1M091440
1M091442
1M091444
1M091446
1M091448
1M091450
1M091452
1M091454
1M091456
1M091458
1M091460
1M091462
1M091464
1M091466
1M091468
1M091470
1M091472
1M091474
1M091476
1M091478
1M091480
1M091482
1M091484
1M091486
1M091488
1M091490
1M091492
1M091494
1M091496

```

```

V(1,2,2)=0.
VMAX(1,2)=0.
V(1,2,3)=0.
VMAX(1,3)=0.
V(2,2,1)=0.
VMAX(2,1)=0.
V(2,2,2)=1.
VMAX(2,2)=1.
V(2,2,3)=0.
VMAX(2,3)=0.
CALL DIFFC(PHI(2),CO)
DO 31 I=1,3
V(3,2,1)=-V(2,2,1)*CO(1)-V(1,2,I)*CO(2)
V(3,2,3)=V(3,2,3)+CO(3)
DPC=DPMIN
M=1
PHI(3)=PHI(2)+DPC
CALL DIFFC(PHI(3),CO)
DO 32 I=1,3
V(1,3,I)=V(1,2,I)+DPMIN*(V(2,2,I)+DPMIN*V(3,2,I)/2.)
V(2,3,I)=V(2,2,I)+DPMIN*V(3,2,I)
V(3,3,I)=-V(2,3,I)*CO(1)-V(1,3,I)*CO(2)
V(3,3,3)=V(3,3,3)+CO(3)
DO 51 J=M,2
PHI(J)=PHI(J+1)
DO 51 K=1,3
DO 51 I=1,3
V(I,J,K)=V(I,J+1,K)
DO 52 J=1,3
DO 52 I=1,2
VMAX(I,J)=MAXIF(VMAX(I,J),ABSF(V(I,2,J) ) )
IF (MAXIF(VMAX(1,1),VMAX(1,2),VMAX(1,3) )-174532.93) 53,13,13
N=N+1
NT=NT+1
IF (PHI(2)-3.*DPPR) 54,6,6
IF (NT-NRAT) 6,6,13
IF (PHI(2)-PHPR) 7,61,61
PRBL(1)=57.295780*PHI(2)
PRBL(2)=H
PRBL(4)=57.295780*GAMMA
PRBL(3)=2.376 917E-3*SIG
DO 62 I=1,3

```

```

1M091498
1M091500
1M091502
1M091504
1M091506
1M091508
1M091510
1M091512
1M091514
1M091516
1M091518
1M091520
1M091522
1M091524
1M091526
1M091528
1M091530
1M091532
1M091534
1M091536
1M091538
1M091540
1M091542
1M091544
1M091546
1M091548
1M091550
1M091552
1M091554
1M091556
1M091558
1M091560
1M091562
1M091564
1M091566
1M091568
1M091570
1M091572
1M091574
1M091576
1M091578
1M091580

```

```

62 PRBL(I+4)=57.295780*V(1,2,I)-FRBL(4)
WRITE OUTPUT TAPE 6,2,(PRBL(I),I=1,8)
2 FORMAT (F14.3,F14.0,1PE14.4,OP4F14.3,17)
IF (SENSELIGHT I) 14,15
14 WRITE OUTPUT TAPE 6,17
17 FORMAT (16HODATA UNSUITABLE)
GO TO 9
15 SAV(1,1,NS)=PHI(2)
SAV(2,1,NS)=GAMMA
DO 63 J=1,3
DO 63 I=1,2
63 SAV(I,J+1,NS)=V(1,2,J)
N=0
NS=NS+1
PHPR=PHPR+DPPR
IF (PHI(2)-PHEND) 7,7,64
64 PRBL(1)=(1.-57.295780*V(1,2,1) )*(1.-V(2,2,2) )
2 -V(2,2,1)*V(1,2,2)/1.7453293E-2
PRBL(2)=0.
PRBL(3)=0.
PRBL(4)=0.
IF (ABSF(PRBL(1) )-1.E-6) 66,66,65
65 PRBL(2)=57.295780*(V(1,2,3)*(1.-V(2,2,2) )+V(2,2,3)
2 *V(1,2,2) )/PRBL(1)
PRBL(3)=( 1.-57.295780*V(1,2,1) )*V(2,2,3)+V(2,2,1)
2 *V(1,2,3)/1.7453293E-2 )/PRBL(1)
PRBL(4)=VMAX(1,3)+ABSF(PRBL(2) )*VMAX(1,1)
2 +ABSF(PRBL(3) )*VMAX(1,2)
66 DO 67 I=1,3
67 PRBL(I+4)=VMAX(1,I)
NS=8*NS-8
C DIMENSION SAV1(2880)
EQUIVALENCE (SAV1,SAV)
C
WRITE TAPE 2,C(28),NS8,(SAV1(I),I=1,NS8),(PRBL(I),I=1,7)
DO 68 I=1,4
68 PRBL(I+3)=57.295780*PRBL(I+3)
WRITE OUTPUT TAPE 6,1,(C(I),I=1,7),C(9)
WRITE OUTPUT TAPE 6,69,C(28),FRBL(1),(V(2,2,I),I=1,3),
2 NT,PRBL(4),PRBL(2),(PRBL(I+4),I=1,3),PRBL(3),(VMAX(2,I),I=1,3)
69 FORMAT (1H+F89.0/1H031X5HDELTA23X15HEND DERIVATIVES28X2HNT/

```

```

IM091582
IM091584
IM091586
IM091588
IM091590
IM091592
IM091594
IM091596
IM091598
IM091600
IM091602
IM091604
IM091606
IM091608
IM091610
IM091612
IM091614
IM091616
IM091618
IM091620
IM091622
IM091624
IM091626
IM091628
IM091630
IM091632
IM091634
IM091636
IM091638
IM091640
IM091642
IM091643
IM091644
IM091646
IM091647
IM091648
IM091650
IM091652
IM091654
IM091656
IM091658
IM091660

```

```

1M091662
1M091664
1M091666
1M091668
1M091670
1M091672
1M091674
1M091676
1M091678
1M091680
1M091682
1M091684
1M091686
1M091688
1M091690
1M091692
1M091694
1M091696
1M091698
1M091700
1M091702
1M091704
1M091706
1M091708
1M091710
1M091711
1M091712
1M091714
1M091716
1M091718
1M091720
1M091722

2 1PE42.4,0PI4X3FI4.3,I7//20X27HITH PER BD WEIGHT THILTA H1
3 13X10HMAX THETAS/I6XIP2E12.4,I4XOP3F14.3//32X
4 15HWEIGHT THETA H213X15HMAX DERIVATIVES/IPE42.4,14X
5 OP3FI4.3)
9 RETURN
7 PHI(3)=PHI(2)+DPC
CALL DIFFC(PHI(3),CO)
DO 71 I=1,3
V(2,4,I)=PREDF(V(2,1,I),V(2,2,I),V(3,1,I),V(3,2,I) )
V(1,4,I)=SIMP(V(1,1,I),V(2,1,I),V(2,2,I),V(2,4,I) )
V(3,4,I)=-V(2,4,I)*CO(1)-V(1,4,I)*CO(2)
V(3,4,3)=V(3,4,3)+CO(3)
DO 72 I=1,3
V(2,3,I)=SIMP(V(2,1,I),V(3,1,I),V(3,2,I),V(3,4,I) )
V(1,3,I)=SIMP(V(1,1,I),V(2,1,I),V(2,2,I),V(2,3,I) )
V(3,3,I)=-V(2,3,I)*CO(1)-V(1,3,I)*CO(2)
V(3,3,3)=V(3,3,3)+CO(3)
ERR=0.
DO 73 I=1,3
ERR=MAXIF(ERR,RELERF(V(3,4,I),V(3,3,I) ) )
IF (ERR-1.0E-6) IO,IO,12
10 IF (DPC+DPC-DPPR) I1,I1,4
11 TEST=.5*(PHPR+DPTST-PHI(3) )/DPC
IF (INTF((TEST-INTF(TEST+.5) )*FNKRAT) ) 4,16,4
16 M=2
DPC=DPC+DPC
GO TO 5
4 M=1
GO TO 5
12 IF (ERR-1.0E-4) 4,4,3
13 SENSELIGHT 1
GO TO 61
END(0,0,0,0,0)

```

STORAGE FOR VARIABLES APPEARING IN COMMON SENTENCES

DEC	OCT	DEC	OCT	DEC	OCT	DEC	OCT
H 32546	77442	HP 32561	77461	GAM 32559	77457	GAMQ 32558	77456
GAMMA 32534	77426	EC 32560	77460	C 32562	77462	OI 32562	77462
OM 32557	77455	OMU 32532	77424	PHIO 32553	77451	PHMAX 32535	77427
PP 32548	77444	QP 32547	77443	RE 32533	77425	R 32545	77441
SIGP 32550	77446	SIG 32544	77440	TBL 32531	77423	THO 32552	77450
THPO 32551	77447	VRVV 32545	77445				

NAMES OF ARITHMETIC STATEMENT FUNCTIONS WITH CORRESPONDING INTERNAL FORMULA NUMBERS AND OCTAL LOCATIONS

PRED	IFN	LOC	IFN	LOC	IFN	LOC	IFN	LOC
	2	01517	SIMP	3	01533	RELER	4	01543

EXTERNAL FORMULA NUMBERS WITH CORRESPONDING INTERNAL FORMULA NUMBERS AND OCTAL LOCATIONS

EFN	IFN	LOC	EFN	IFN	LOC	EFN	IFN	LOC
1	14	00000	23	18	00053	21	20	00063
3	50	00236	3	51	00236	31	53	00244
5	65	00330	51	65	00354	52	72	00377
54	77	00446	6	78	00452	61	79	00457
2	91	00000	14	92	00525	14	94	00534
15	97	00537	63	101	00557	64	106	00610
66	114	00714	67	115	00716	68	130	00753
9	153	01052	7	155	01056	71	161	01117
73	170	01205	10	172	01230	11	173	01235
4	178	01267	12	180	01272	13	181	01277

STORAGE NOT USED BY PROGRAM

DEC	OCT	DEC	OCT
3854	07416	32362	77152

LOCATIONS OF NAMES IN TRANSFER VECTOR

DIFFC	DEC	OCT	DIFFC	DEC	OCT
(FIL)	1	00001	(IOH)O	2	00002
(STH)	5	00005	COS	7	00007
	6	00006	(10H)O	8	00010
			ATMOS	3	00003
			(LEV)	8	00010
			SQRT	4	00004
			(RTN)	0	00000

STORAGE LOCATIONS FOR VARIABLES APPEARING IN DIMENSION AND EQUIVALENCE SENTENCES

DEC	OCT	DEC	OCT	DEC	OCT	DEC	OCT
CO	962 01702	N	3846 07406	PHI	965 01705	PRBL	3853 07415
SAV1	3845 07405	SAV	3845 07405	VMAX	959 01677	V	953 01671

STORAGE LOCATIONS FOR VARIABLES NOT APPEARING IN DIMENSION, EQUIVALENCE OR COMMON SENTENCES

DEC	OCT	DEC	OCT	DEC	OCT	DEC	OCT
FNRRAT	917 01625	ERR	916 01624	DPTST	915 01623	DPPR	914 01622
DPMIN	913 01621	DPLIM	912 01620	DPC	911 01617	M	910 01616
NRAT	909 01615	NS8	908 01614	NS	907 01613	NT	906 01612
PHDP	905 01611	PHEND	904 01610	PHPR	903 01607	RP	902 01606
TEST	901 01605						

STORAGE LOCATIONS FOR SYMBOLS NOT APPEARING IN SOURCE PROGRAM

DEC	OCT	DEC	OCT	DEC	OCT	DEC	OCT
EJM	281 00431	C11G4	90C 01604	A11G4	838 01506	8125	776 01410
81H	780 01414	812	786 01422	811	837 01505	11	896 01600
111	693 01575	21	705 01301	31	710 01306	411	889 01571
6)	727 01327	71	888 01570				

(LEV)	(IOH)O	(STH)	SUBROUTINES NOT PUNCHED FROM LIBRARY	COS	DIFFC
(RTN)		(FIL)	SORT ATMOS		

```

SUBROUTINE DIFFC(PHI,V)
DIMENSION C(200),V(3)
COMMON C
EQUIVALENCE (C(32),TBL(1),SIG(14),R(15),H(16),QP(17),
2 PP(18),VRVV(19),SIGP(20),THPO(21),THO(22),PHIO(23),
3 EC(30),HP(31),OI(32),PHMAX(5),OM(27),GAMQ(28),GAM(29))
4 ,(GAMMA(4),RE(3),OMU(2),C(32))
SINP=SINF(PHI)
COSP=COSE(PHI)
GAMMA=EC*SINP/(1.+EC*COSP)
EOMCP=EC*(1.-COSP)
RV=(HP+RE)/(1.+EC)/(1.+EC*COSP)
H=RV-RE
CALL ATMOS(H,TBL,R)
RRVV=SIG*VRVV/SIG
TWOQ=QP*RRVV*(1.+EOMCP)-EC*SINP
V(1)=TWOQ+TWOQ
PTIL=PP*RRVV*VRVV*(1.+EOMCP+EOMCP)
V(2)=OM*(1.-EC*COSP)/.33333333+PTIL
V(3)=GAMMA*PTIL+TWOQ+TWOQ
RETURN
END(0,0,0,0,0)

```

STORAGE FOR VARIABLES APPEARING IN COMMON SENTENCES

DEC	OCT	DEC	OCT	DEC	OCT	DEC	OCT				
H	32546	77442	HP	32561	77461	GAM	32559	77457	GAMQ	32558	77456
GAMMA	32534	77426	EC	32560	77460	C	32562	77462	OI	32562	77462
OM	32557	77455	OMU	32532	77424	PHIO	32553	77451	PHMAX	32535	77427
PP	32548	77444	QP	32547	77443	RE	32533	77425	R	32545	77441
SIGP	32550	77446	SIG	32544	77440	TBL	32531	77423	THQ	32552	77450
THPO	32551	77447	VRVV	32549	77445						

STORAGE NOT USED BY PROGRAM

DEC	OCT	DEC	OCT
133	00205	32362	77152

LOCATIONS OF NAMES IN TRANSFER VECTOR

DEC	OCT	ATMOS	DEC	OCT	SIN	DEC	OCT	
COS	1	00001	ATMOS	0	00000	SIN	2	00002

STORAGE LOCATIONS FOR VARIABLES NOT APPEARING IN DIMENSION-EQUIVALENCE OR COMMON SENTENCES

DEC	OCT	DEC	OCT	DEC	OCT	DEC	OCT				
EOMCP	132	00204	COSP	131	00203	PTIL	130	00202	RRVV	129	00201
RV	128	00200	SINP	127	00177	TWQQ	126	00176			

STORAGE LOCATIONS FOR SYMBOLS NOT APPEARING IN SOURCE PROGRAM

DEC	OCT	DEC	OCT	DEC	OCT	DEC	OCT				
9)	121	00171	1)	123	00173	3)	114	00162	6)	116	00164
7)	122	00172									

SIN COS ATMOS SUBROUTINES NOT PUNCHED FROM LIBRARY

```
C      CHAIN(1) PROGRAM TO SCALE FOR PUNCHING
      DIMENSION C(20)
      COMMON C
      IF DIVIDE CHECK 1,2
      C(20)=0.
      CALL FXRCH(C)
      CALL SCALE
      GO TO 1
      END(0,0,0,0,0)
```

```
1M091800
1M091801
1M091802
1M091803
1M091804
1M091805
1M091806
1M091807
```

STORAGE FOR VARIABLES APPEARING IN COMMON SENTENCES

	DEC	OCT		DEC	OCT		DEC	OCT
C	32562	77462						

EXTERNAL FORMULA NUMBERS WITH CORRESPONDING INTERNAL FORMULA NUMBERS AND OCTAL LOCATIONS

EFN	IFN	LOC	EFN	IFN	LOC	EFN	IFN	LOC
2	4	00004	1	5	00006	1	6	00006

STORAGE NOT USED BY PROGRAM

DEC	OCT
20	00024
DEC	OCT
32542	77436

LOCATIONS OF NAMES IN TRANSFER VECTOR

FXRCH	1	00001	SCALE	0	00000			
	DEC	OCT		DEC	OCT		DEC	OCT

STORAGE LOCATIONS FOR SYMBOLS NOT APPEARING IN SOURCE PROGRAM

3)	DEC	OCT		DEC	OCT		DEC	OCT
	14	00016	6)	15	00017			

SUBROUTINES NOT PUNCHED FROM LIBRARY

FXRCH	SCALE
-------	-------

	SUBROUTINE SCALE	1M091820
	DIMENSION C(20),SAV(8,120),SAV1(960),PNCH(121,16),IPNCH(121,16),	1M091822
	2 SPNCH(5,16),PRBL(7),ISPNC(5,16)	1M091824
	COMMON C	1M091826
	EQUIVALENCE (SAV,SAV1),(PNCH,IPNCH),(SPNCH,ISPNC)	1M091827
	WRITE OUTPUT TAPE 6,1,(C(I),I=1,18)	1M091828
1	FORMAT (1H16X11HTAPE RECORD5X26HWEIGHT PAR WEIGHT GAMMA	1M091830
	2 4X9HWEIGHT HI6X9HWEIGHT H2/F18.0,1PE16.6,9E15.6/	1M091832
	3 33HOSCALING -- MIN VALUE MIN COUNTS5X9HMAX VALUE6X	1M091834
	4 39HMAX COUNTS EDGE COUNTS CARD FIELD/	1M091836
	5 9H FUNCTIONOPF11.3,F13.0,F14.3,F16.0,F15.0,F14.0/	1M091838
	6 9H ARGUMENT F11.3,F13.0,F14.3,F16.0,F15.0,F14.0/	1M091840
	7 10HODECK ID -F7.0)	1M091842
	J=C(11)+5	1M091844
3	IF (INTF(C(20)+.5)-INTF(C(1)+.5)) 2,60,2	1M091846
2	READ TAPE 2,C(20),NS8,(SAV1(I),I=1,NS8),(PRBL(I),I=1,7)	1M091848
4	IF (C(20)) 4,3,3	1M091850
	REWIND 2	1M091852
5	IF (SENSELIGHT 1) 5,6	1M091854
	SPNCH(1,J)=SPNCH(3,J)	1M091856
6	GO TO 69	1M091858
	SENSELIGHT 1	1M091860
	GO TO 2	1M091862
60	IF (SENSELIGHT 1) 61,61	1M091864
61	WH1=C(4)	1M091866
	WH2=C(5)	1M091868
	WP=ABSF(C(2))	1M091870
	IF (C(2)) 62,63,63	1M091872
62	WH1=PRBL(2)	1M091874
	WH2=PRBL(3)	1M091876
63	NS=NS8/8	1M091878
	CALL WRITE(6HOD WHS,PRBL,3)	1M091879
	DO 64 I=1,5	1M091880
64	SPNCH(I,J)=C(I+5)	1M091882
	PNCH(1,J)=WH1	1M091884
	DO 65 I=1,NS	1M091886
65	PNCH(I+1,J)=(C(3)*SAV(2,I)+WP*SAV(7,I)+WH1*SAV(3,I)	1M091888
	2 +WH2*SAV(5,I))*57.295780	1M091890
	N=NS+1	1M091892
	CALL WRITE(6HOFNCTN,PNCH(1,J),N)	1M091894
	IF (C(17)) 69,69,66	1M091896
66	J=C(17)+5	1M091898

67	DO 67 I=1,5 SPNCH(I,J)=C(I+11) PNCH(1,J)=0. DO 68 I=1,NS PNCH(I+1,J)=57.295780*SAV(1,I) CALL WRITE (6H0 ARG,PNCH(1,J),N) IF (C(18)) 9,9,7 ID=C(18)+5 DO 72 J=1,16 ISPNC(5,J)=SPNCH(5,J) DO 71 I=1,N IPNCH(I,J)=XMINOF(ISPNC(5,J),NP SCL(PNCH(I,J),SPNCH(1,J))) SPNCH(3,J)=SPNCH(1,J) ISPNC(4,J)=0 N1=N+1 N2=N1+1 PUNCH 73,(ID,I,(IPNCH(I,J),J=1,16),I=1,N),ID,N1, 2 (ISPNC(4,J),J=1,16),ID,N2,(ISPNC(5,J),J=1,16) 73 FORMAT (1H013,17I4) PUNCH 74,ID 74 FORMAT (1H-13/1H) C(18)=0. 9 RETURN END(0,0,0,0,0)	1M091900 1M091902 1M091904 1M091906 1M091908 1M091910 1M091912 1M091914 1M091916 1M091918 1M091920 1M091922 1M091923 1M091924 1M091926 1M091928 1M091930 1M091932 1M091934 1M091936 1M091938 1M091940 1M091942
----	--	--

STORAGE FOR VARIABLES APPEARING IN COMMON SENTENCES

DEC OCT DEC OCT DEC OCT DEC OCT

C 32562 77462

EXTERNAL FORMULA NUMBERS WITH CORRESPONDING INTERNAL FORMULA NUMBERS AND OCTAL LOCATIONS

EFN	IFN	LOC	EFN	IFN	LOC	EFN	IFN	LOC	EFN	IFN	LOC
1	11	00000	3	13	00053	2	14	00066	2	23	00111
4	25	00116	4	25	00116	5	27	00122	6	29	00125
60	31	00127	61	32	00131	62	36	00143	63	38	00147
64	42	00175	65	45	00222	66	50	00270	67	52	00316
68	55	00343	69	58	00363	7	59	00367	71	63	00421
71	64	00432	72	66	00455	73	85	00000	74	89	00000
9	91	00574									

STORAGE NOT USED BY PROGRAM

DEC OCT DEC OCT

3498 06652 32542 77436

49

LOCATIONS OF NAMES IN TRANSFER VECTOR

DEC	OCT	WRITE	DEC	OCT	DEC	OCT	DEC	OCT			
NPSC	1	00001	WRITE	2	00002	(FIL)	4	00004	(IOHJO)	6	00006
(LEV)	7	00007	(RTN)	3	00003	(SCH)	0	00000	(STH)	5	00005

STORAGE LOCATIONS FOR VARIABLES APPEARING IN DIMENSION AND EQUIVALENCE SENTENCES

DEC	OCT	DEC	OCT	DEC	OCT	DEC	OCT	
ISPNC	601	01131	IPNCH	2537	04751	PNCH	2537	04751
SAV1	3497	06651	SAV	3497	06651	SPNCH	601	01131
						PRBL	521	01011

STORAGE LOCATIONS FOR VARIABLES NOT APPEARING IN DIMENSION, EQUIVALENCE OR COMMON SENTENCES

DEC	OCT	DEC	OCT	DEC	OCT	DEC	OCT	
I	514	01002	ID	513	01001	J	512	01000
N2	510	00776	N	509	00775	NS8	508	00774
WH1	506	00772	WH2	505	00771	WP	504	00770
						NS	507	00773
						NI	511	00777

STORAGE LOCATIONS FOR SYMBOLS NOT APPEARING IN SOURCE PROGRAM

APPENDIX B

PROGRAM DESCRIPTION - ATMOSPHERE SUBROUTINE
NORTH AMERICAN AVIATION, INC. COLUMBUS
ENGINEERING PROGRAM DESCRIPTION

1. Identification

- a. Atmosphere Subroutine, 1F113
- b. Sarah Crooke, August 1959
- c. Aero Advanced Design Group - 214
- d. Fortran Source Deck is up to date.

2. Purpose

Calculates standard atmosphere properties at a specified altitude.

3. Restrictions

Fortran 2 subroutine subprogram.

4. Method

Table look-up of properties at base geopotentials followed by application of the ideal gas law.

5. Use

Subroutine is reached by Fortran Statement

CALL ~~ATMOS~~ (H, TBL, R)

where H is the input altitude
 TBL is the table of base geopotentials and properties
 R is the results array.

6. Coding Information

- a. This routine occupies 309 Locations.
- b. Inferior Subroutines are Library routines DUMP, EXP, SQRT, and the Fortran system routine EXP(3).

7. Data Arrangement

- a. The array TBL must contain the following data:

<u>Location</u>	<u>Description</u>	<u>Symbol</u>
1	Sea Level Temperature, °R	T_0
2	Sea Level Pressure, lb/ft ²	P_0

<u>Location</u>	<u>Description</u>	<u>Symbol</u>
3	Sea Level Gas Constant, $\text{ft}^2/\text{sec}^{20}\text{R}$	R_0
4	Sea Level Density, slugs/ft^3	ρ_0
5	Number of Base Geopotentials in Table	n
6	Base Geopotentials, ft	h_0
⋮		
5 + n		
6 + n	Base Gas Constants, $\text{ft}^2/\text{sec}^{20}\text{R}$	R_0
⋮		
5 + 2n		
6 + 2n	Base Temperatures, $^{\circ}\text{R}$	T_0
⋮		
5 + 3n		
6 + 3n	Base Specific Heat Ratios, dimensionless	γ_0
⋮		
5 + 4n		
6 + 4n A_1	$A_i \times 10^{\delta_i}$ is the i th Base Pressure Ratio, dimensionless	δ_0
7 + 4n B_1		
⋮		
4 + 6n A_n		
5 + 6n B_n		

b. On return the array R will contain the following data:

<u>Location</u>	<u>Description</u>	<u>Symbol</u>
1	Geopotential Altitude, ft	h
2	Density Ratio, dimensionless	σ
3	Speed of Sound, ft/sec	a
4	Speed of Sound, knots	A
5	Pressure Ratio, dimensionless	δ
6	Local Acceleration of Gravity, ft/sec^2	g

<u>Location</u>	<u>Description</u>	<u>Symbol</u>
7	Temperature, °R	T
8	Gas Constant, ft ² /sec ² °R	R
9	Incompressible Dynamic Pressure at Mach 1, lb/ft ²	G
10	Ratio of Specific Heats, dimensionless	γ

8. Formulas

In the formulas below H is the input altitude. The remaining symbols are those listed in 7 above. The subscript i refers to the value of the quantity at the i th base geopotential altitude, where

$$i = 1 \text{ if } h < h_2,$$

$$i = j \text{ if } h_j \leq h < h_{j+1}, \quad j = 2, \dots, n-1,$$

$$i = n-1 \text{ if } h_n \leq h.$$

The prefix Δ denotes the increment of the quantity from the i th to the $(i+1)$ th base geopotential altitude.

$$h = \frac{20891000H}{H + 20891000}$$

$$T = T_0 + (h - h_0) \frac{\Delta T}{\Delta h}$$

$$R = R_0 + (h - h_0) \frac{\Delta R}{\Delta h}$$

$$\delta = \delta_0 \left[\frac{R_0}{R} \frac{T}{T_0} \right]^{32.174049 \Delta h / (T_0 \Delta R - R_0 \Delta T)},$$

$$\text{if } T_0 \Delta R - R_0 \Delta T \neq 0$$

$$= \delta_0 e^{-32.174049 (h - h_0) / \Delta h},$$

$$\text{if } T_0 \Delta R - R_0 \Delta T = 0$$

$$\sigma = \delta \frac{T_0}{T} \frac{R_0}{R}$$

$$\gamma = \gamma_0 + (h - h_0) \frac{\Delta \gamma}{\Delta h}$$

$$g = 32.174049 \left[\frac{20\,891\,000}{H + 20\,891\,000} \right]^2$$

$$a = \sqrt{\frac{\gamma P_0 \delta}{\rho_0 \sigma}}$$

$$A = \frac{a}{1.6878099}$$

$$G = \frac{1}{2} (\rho_0 \sigma) a^2$$

```

SUBROUTINE ATMOS(H, TABLE, RES)
DIMENSION TABLE(100), RES(10), R(10)
R(1)=H*20891000./(H+20891000.)
I=TABLE(5)
IF(I-1)5,5,10
R(1)=DUMPF(1)
DO20K=2,I
IF(TABLE(K+5)-R(1))20,15,15
CONTINUE
K=I
HB=TABLE(K+4)
R(3)=TABLE(K+5)-HB
J=I+K
RB=TABLE(J+4)
DRDH=(TABLE(J+5)-RB)/R(3)
J=J+I
TB=TABLE(J+4)
DTDH=(TABLE(J+5)-TB)/R(3)
J=J+I
GAMB=TABLE(J+4)
DGDH=(TABLE(J+5)-GAMB)/R(3)
J=J+K
PBPO=TABLE(J+2)*10.**TABLE(J+3)
R(9)=R(1)-HB
R(7)=TB+DTDH*R(9)
R(8)=RB+DRDH*R(9)
R(10)=TB*DRDH-RB*DTDH
IF(R(10))30,25,30
R(2)=EXP(-32.174049*R(9)/(R(7)*R(8)))
GOTO35
R(2)=(RB*R(7)/(R(8)*TB))**(32.174049/R(10))
R(5)=R(2)*PBPO
R(2)=R(5)*TABLE(1)*TABLE(3)/(R(7)*R(8))
R(10)=GAMB+DGDH*R(9)
R(6)=32.174049*(20891000./(20891000.+H))**2
R(3)=SQRTF(R(10)*R(5)*TABLE(5)*TABLE(2)/(TABLE(4)*R(2)))
R(4)=R(3)/1.6878099
R(9)=.5*R(3)*R(3)*TABLE(4)*R(2)
DO40I=1,10
RES(I)=R(I)
RETURN
END(0,0,0,0,0)

```

```

1F040100
1F040105
1F040110
1F040115
1F040120
1F040125
1F040130
1F040135
1F040140
1F040145
1F040150
1F040155
1F040160
1F040165
1F040170
1F040175
1F040180
1F040185
1F040190
1F040195
1F040200
1F040205
1F040210
1F040215
1F040220
1F040225
1F040230
1F040235
1F040240
1F040245
1F040250
1F040255
1F040260
1F040265
1F040270
1F040275
1F040280
1F040285
1F040290
1F040295
1F040300

```

EXTERNAL FORMULA NUMBERS WITH CORRESPONDING INTERNAL FORMULA NUMBERS AND OCTAL LOCATIONS

EFN	IFN	LOC	EFN	IFN	LOC	EFN	IFN	LOC	EFN	IFN	LOC
5	6	00106	10	7	00113	20	9	00132	15	11	00141
25	29	00254	30	31	00272	35	32	00312	40	40	00401

STORAGE NOT USED BY PROGRAM

DEC	OCT
309	00465
	32562 77462

LOCATIONS OF NAMES IN TRANSFER VECTOR

EXP(3)	DEC	OCT	EXP	DEC	OCT	DUMP	DEC	OCT	DE	OCT	DE	OCT
	2	00002	1	00001	3	00003					0	00000

STORAGE LOCATIONS FOR VARIABLES APPEARING IN DIMENSION AND EQUIVALENCE SENTENCES

DEC	OCT	DEC	OCT
308	00464		

STORAGE LOCATIONS FOR VARIABLES NOT APPEARING IN DIMENSION+EQUIVALENCE OR COMMON SENTENCES

DEC	OCT	DEC	OCT	DEC	OCT	DEC	OCT
I	298 00452	HB	297 00451	GAMB	296 00450	DTDH	295 00447
DRDH	294 00446	DGDH	293 00445	J	292 00444	K	291 00443
PBPO	290 00442	RE	289 00441	TB	288 00440		

STORAGE LOCATIONS FOR SYMBOLS NOT APPEARING IN SOURCE PROGRAM

DEC	OCT	DEC	OCT	DEC	OCT	DEC	OCT
E)9	170 00252	E)5	86 00126	C)G2	287 00437	C)G1	286 00436
9)	277 00425	1)	283 00433	2)	265 00411	3)	267 00413
6)	272 00420	7)	282 00432				

DUMP EXP(3) SUBROUTINES NOT PUNCHED FROM LIBRARY SORT

APPENDIX C

PROGRAM DESCRIPTION - FIXED DECIMAL CARD READ
ROUTINE WITH ERROR DETECTION
NORTH AMERICAN AVIATION, INC. COLUMBUS
ENGINEERING PROGRAM DESCRIPTION

1. Identification

- a. Fixed Decimal Card Read Routine with Error Detection - Deck NCF-FXRCH
- b. O. C. Juelich, March 1960
- c. Aero Advanced Design Group - 214
- d. Fortran Source Deck is up to date.

2. Purpose

- a. Load Input Data for Fortran II programs.
- b. Permit resumption of loading after cards with zero or fractional origin are found.
- c. List such error cards in the print-out.

3. Restrictions

- a. Reads input tape 5, writes output tape 6 in case of error.
- b. Data cards are written on Form 114-C-17 (Fortran Fixed 10 Digit Decimal Data).
- c. Uses Fortran II Error Procedure.

4. Method

- a. Card images are read from the input tape. The first word specifies the location to which the data is to be transmitted, and whether the data set is complete.
- b. Filled-in card fields are transmitted to the calling program, blank fields are not transmitted.
- c. Error Card images are written on the output tape. If error cards are found the error procedure is invoked at the end of the data set.

5. Use

- a. Subroutine is reached by the Fortran Statement:

CALL FXRCH (D) .

- b. A card image is read. The first word on the card is the subscript n , written with decimal point. The $(i+2)$ th word on the card; $i = 0, 1, 2, 3, 4$; is transmitted to location $D(|n|+i)$ as a floating point number unless it is blank or -0. If the $(i+2)$ th word is blank or -0 location $D(|n|+i)$ is unmodified. If n is positive this procedure is repeated. If n is negative control returns to the calling program. (Subscripting as in EQUIVALENCE statements.)

6. Coding Information

- a. This routine requires 17⁴ locations.
- b. Inferior subroutines are the Library routine ERRØR and Fortran System Subroutines.

C FIXED DECIMAL CARD READ ROUTINE WITH ERROR DETECTION

```

SUBROUTINE FXRCH(D)
DIMENSION A(2),C(6),D(6)
I=1
1 READ INPUT TAPE 5,101,(C(J),J=1,6),A(1),A(2)
J=ABSF(C(1))
4 IF (J) 2,99,2
2 IF (C(1))-INTF(C(1)) 99,3,99
3 DO 13 K=2,6
IF (C(K)) 12,11,12
11 IF (SIGNF(1,C(K))) 13,13,12
12 D(J)=C(K)
13 J=J+1
14 IF (SIGNF(1,C(1))) 20,20,1
20 GO TO (40,30),I
30 CALL ERROR
40 RETURN
99 GO TO (97,98),I
97 I=2
WRITE OUTPUT TAPE 6,102
98 WRITE OUTPUT TAPE 6,103,(C(J),J=1,6),A(1),A(2)
GO TO 14
101 FORMAT (6F12.0,A6,A2)
102 FORMAT (10H1BAD CARDS)
103 FORMAT (F16.8,5E16.8,A6,A2)
FREQUENCY 4(0,0,1),2(0,1,0),11(1,0,1),14(1,0,5),20(1,0) ,
2 99(1,0)
END(0,0,0,0,0)

```

EXTERNAL FORMULA NUMBERS WITH CORRESPONDING INTERNAL FORMULA NUMBERS AND OCTAL LOCATIONS

EFN	IFN	LOC	EFN	IFN	LOC	EFN	IFN	LOC	EFN	IFN	LOC
1	4	00027	1	9	00050	4	11	00063	2	12	00065
3	13	00075	11	15	00104	12	16	00112	13	17	00116
14	18	00126	20	19	00134	30	20	00140	40	21	00143
99	23	00151	97	24	00153	98	27	00172	98	32	00213
101	34	00000	102	35	00000	103	36	00000			

STORAGE NOT USED BY PROGRAM

DEC OCT
174 00256
DEC OCT
32562 77462

LOCATIONS OF NAMES IN TRANSFER VECTOR

DEC	OCT	(FIL)	(RTN)	(IOH)I	(STH)	DEC	OCT	(IOH)O	(TSH)	DEC	OCT
3	00003			0	00000	6	00006			2	00002
7	00007			4	00004	1	00001			5	00005

STORAGE LOCATIONS FOR VARIABLES APPEARING IN DIMENSION AND EQUIVALENCE SENTENCES

DEC	OCT	A	DEC	OCT	DEC	OCT
171	00253		173	00255		

STORAGE LOCATIONS FOR VARIABLES NOT APPEARING IN DIMENSION, EQUIVALENCE OR COMMON SENTENCES

DEC	OCT	J	DEC	OCT	DEC	OCT
165	00245		164	00244		

STORAGE LOCATIONS FOR SYMBOLS NOT APPEARING IN SOURCE PROGRAM

DEC	OCT	E)H	E)F	E)D	DEC	OCT	DEC	OCT
121	00171			103	00147	95	00137	
77	00115	D)401	D)401	22	00026	D)20F	104	00150
85	00125	C)G3	C)G3	163	00243	C)G2	162	00242
160	00240	8)37	8)37	153	00231	8)36	156	00234
142	00216	3)	3)	144	00220	6)	145	00221

(LEV)	(IOH)I	(TSH)	ERROR	(IOH)O	(STH)	(FIL)

APPENDIX DPROGRAM DESCRIPTION - PUNCH SCALING FUNCTION
NORTH AMERICAN AVIATION, INC. COLUMBUS
ENGINEERING PROGRAM DESCRIPTION1. Identification

- a. Punch Scaling Function, Deck - NCF NPSCL.
- b. J. B. Burnett, May 1960.
- c. Digital Computing Group - 331.
- d. Source Deck is up to date.

2. Purpose

Converts floating point data to fixed point data which is scaled suitably for automatic plotting equipment.

3. Restrictions

Operates as a Fortran II Function Subprogram.

4. Method

Linear Scaling.

5. Use

- a. Reached through a Fortran Statement such as

$$IPNCH = NPSCL (A, S)$$

where A is the number to be scaled

S is an array of dimension 4 containing the scale table.

S(2) holds the plotter counts corresponding to the left or bottom edge of the graph

S(4) holds the plotter counts corresponding to the right or top edge of the graph

S(1) holds the value of A corresponding to S(2)

S(3) holds the value of A corresponding to S(4)

- b. S(2) and S(4) must be between 0.0 and 999.0, S(3) must exceed S(1). If the array S does not meet these specifications IPNCH will be set to -0.
- c. If A is less than S(1), IPNCH will be set to 0. If A exceeds S(3) IPNCH will be set to 999. (The user must make provision to confine the plotting head to the plotting bed.)

d. It is expected that the plotting machine be set to 20 counts per centimeter, with (0, 0) counts at the lower left hand corner of the grid.

6. Coding Information

- a. This routine occupies 119 locations.
- b. There are no inferior subroutines.

	FUNCTION NPSC(A,S)	NPSC300
	DIMENSION S(4)	NPSC305
	IF (S(3)-S(1)) 98,98,1	NPSC310
1	DEN=S(3)-S(1)	NPSC315
	IF (S(4)) 98,2,2	NPSC320
2	IF (S(2)) 98,3,3	NPSC325
3	IF (S(4)-999) 14,14,98	NPSC330
14	IF (S(2)-999) 4,4,98	NPSC332
4	ENUM=S(4)-S(2)	NPSC335
	IF (A-S(1)) 99,5,5	NPSC340
5	IF (A-S(3)) 6,6,100	NPSC345
6	NPSC=(A-S(1))*ENUM/DEN+S(2)+.5	NPSC350
	GO TO 101	NPSC355
98	NPSC=-.0	NPSC360
	GO TO 101	NPSC365
99	NPSC=0	NPSC370
	GO TO 101	NPSC375
100	NPSC=999	NPSC380
101	RETURN	NPSC385
	END(0,0,0,0,0)	

EXTERNAL FORMULA NUMBERS WITH CORRESPONDING INTERNAL FORMULA NUMBERS AND OCTAL LOCATIONS

EFN	IFN	LOC	EFN	IFN	LOC	EFN	IFN	LOC	EFN	IFN	LOC
1	4	00055	2	6	00064	3	7	00070	14	8	00074
4	9	00100	5	11	00110	6	12	00114	98	14	00131
99	16	00134	100	18	00137	101	19	00141			

STORAGE NOT USED BY PROGRAM

DEC	OCT	DEC	OCT
119	00167	32562	77462

STORAGE LOCATIONS FOR VARIABLES NOT APPEARING IN DIMENSION, EQUIVALENCE OR COMMON SENTENCES

ENUM	DEC	OCT	DEC	OCT	DEC	OCT
118	00166	117	00165	NP5CL	116	00164

STORAGE LOCATIONS FOR SYMBOLS NOT APPEARING IN SOURCE PROGRAM

9)	DEC	OCT	DEC	OCT	DEC	OCT
	111	00157	2)	102	00146	3)
						6)
				104	00150	106
						00152

APPENDIX EPROGRAM DESCRIPTION - OUTPUT TAPE WRITE ROUTINE
NORTH AMERICAN AVIATION, INC. COLUMBUS
ENGINEERING PROGRAM DESCRIPTION1. Identification

- a. Output Tape Write Routine, Deck - NCF WRITE.
- b. O. C. Juelich, June 1960.
- c. Research Group - 330.
- d. Fortran Source Deck is up to date.

2. Purpose

Writes up to ten floating point numbers per line retaining seven digit accuracy for most numbers.

3. Restrictions

- a. Writes output tape 6.
- b. Intended primarily for numbers between 10^{-3} and 10^7 in magnitude. Seven digit accuracy is retained for numbers between 1 and 10^7 .
- c. Source Deck includes 704 SAP instructions.

4. Method

A Format Statement is generated, using F conversion for numbers less than 10^7 and E conversion for numbers 10^7 or larger in magnitude.

5. Use

Reached through a Fortran Statement such as

```
CALL WRITE (6HiIDENT, A, N)
```

where i is the pre-print space control character

IDENT is the indicative word for the first line of print.

(The second line, if any, will be identified by an 11, the third by 21, etc.)

A is the first word to be printed.

N is the number of words to be printed, $N > 0$.

6. Coding Information

- a. This routine requires 171 locations.
- b. All inferior subroutines are part of the Fortran System.

```

SUBROUTINE WRITE(A,F,N)
  SFMTA ALF (A6) WRITE000
  SFMTE ALF E10.2 WRITE010
  SFMTF ALF F10.0 WRITE020
  SFMTI ALF I16 WRITE030
  SFMTX ALF WRITE040
  DIMENSION F(25),FMT(12)
  EQUIVALENCE (B,IB)
  FMT(1)=FMTA
  B=A
  J=1
  K=XMINOF(J+9,N)
  DO 2 I=J,K
  IT=I+2-J
  X=ABSF(F(I) )
  L=1
  DO 3 M=1,7
  IF (X-9 999 999A) 4,6,5
  4 X=10*X
  5 L=L+1
  6 IF (L) 6,7,7
  FMT(IT)=FMTE
  GO TO 8
  S7 CLA L
  S ARS 12
  S ORA FMTF
  S SLW FMT(IT)
  8 FMT(IT+1)=FMTX
  2 CONTINUE
  WRITE OUTPUT TAPE 6,FMT,B,(F(I),I=J,K)
  9 IF (K-N) 9,10,9
  IB=J
  FMT(1)=FMTI
  GO TO 1
  10 RETURN
  CARDS
  WRITE050
  WRITE060
  WRITE070
  WRITE080
  WRITE090
  WRITE100
  WRITE110
  WRITE120
  WRITE130
  WRITE140
  WRITE150
  WRITE160
  WRITE170
  WRITE180
  WRITE190
  WRITE200
  WRITE210
  WRITE220
  WRITE230
  WRITE240
  WRITE250
  WRITE260
  WRITE270
  WRITE280
  WRITE290
  WRITE300
  WRITE310
  WRITE320
  WRITE330
  WRITE340
  WRITE350
  WRITE360

```

EXTERNAL FORMULA NUMBERS WITH CORRESPONDING INTERNAL FORMULA NUMBERS AND OCTAL LOCATIONS

EFN	IFN	LOC	EFN	IFN	LOC	EFN	IFN	LOC	EFN	IFN	LOC
1	12	00035	4	19	00077	3	20	00102	5	21	00112
6	22	00117	7	24	00122	8	28	00126	2	29	00130
9	38	00165	10	42	00175						

STORAGE NOT USED BY PROGRAM

DEC	OCT	DEC	OCT
171	00253	32562	77462

LOCATIONS OF NAMES IN TRANSFER VECTOR

(FIL)	DEC	OCT	(IOH)O	DEC	OCT	(LEV)	DEC	OCT	(STH)	DEC	OCT
	0	00000		2	00002	3	00003			1	00001

STORAGE LOCATIONS FOR VARIABLES APPEARING IN DIMENSION AND EQUIVALENCE SENTENCES

DEC	OCT	DEC	OCT	DEC	OCT	DEC	OCT	
18	170	00252	FMT	169	00251	B	170	00252

STORAGE LOCATIONS FOR VARIABLES NOT APPEARING IN DIMENSION, EQUIVALENCE OR COMMON SENTENCES

DEC	OCT	DEC	OCT	DEC	OCT	DEC	OCT	
17	157	00235	I	156	00234	J	155	00233
L	153	00231	M	152	00230	X	151	00227
						K	154	00232

STORAGE LOCATIONS FOR SYMBOLS NOT APPEARING IN SOURCE PROGRAM

DEC	OCT	DEC	OCT	DEC	OCT	DEC	OCT				
E18	91	00133	E18	77	00115	DEC	71	00107			
D1407	73	00111	D1401	28	00034	E16	D1601	27	00033		
C161	148	00224	91	145	00221	C1200	C162	149	00225		
31	138	00212	61	140	00214	11	146	00222	21	129	00201

(LEV)	(IOH)O	(STH)	(FIL)

APPENDIX FSUMMARY OF LIBRARY ROUTINES

SQRT Evaluates the square root of a non-negative floating-point number. Invokes **ERROR** if argument is less than zero.

COS Evaluates the cosine of an angle expressed in radians in floating-point form.

SIN Evaluates the sine of an angle expressed in radians in floating-point form.

EXP Evaluates the exponential function (e^x) of a floating-point number less than 88. Invokes **ERROR** if the argument exceeds 88.

ERROR

- a. Produces a "back-trace" through the program, listing the point and subroutine at which the error was detected, the point at which the subroutine was called, up to the main program.
- b. Turns on the Divide Check indicator and tests the first instruction of the main program. If this is "IF DIVIDE CHECK" control goes to this statement, otherwise **DUMP** is invoked.

CHAIN Loads a program identified by Record and Tape Number from Tape to Core and gives it control.

DUMP Gives a full Memory Print-out and invokes **EXIT**.

EXIT Returns control to the Monitor to end the job. (This function is also performed by the input tape reading system routine if an "End of File" is encountered on the input tape.)

APPENDIX G

SAMPLE INPUT AND OUTPUT

FORTRAN FIXED 10 DIGIT DECIMAL DATA

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

LINE NO.	NUMBER	IDENTIFICATION	DESCRIPTION DO NOT KEY PUNCH
1	3 2 .		TBL (1)
13	5 1 8 . 6 8 8		T ₀
25	2 1 1 6 . 2 1 6 9		P ₀
37	1 7 1 6 . 4 8 2 7		R ₀
49	. 0 0 2 3 7 6 9 1 7	73	P ₀
61	1 3 .	1	n
1	3 7 .		TBL (6)
13	- 5 0 0 . 0 1 1 9 7		h _A 1
25	3 6 0 8 9 . 2 3 8 8		h _A 2
37	8 2 0 2 0 . 9 9 7 3		h _A 3
49	1 5 4 1 9 9 . 4 7 5	73	h _A 4
61	1 7 3 8 8 4 . 5 1 4	2	h _A 5
1	4 2 .		TBL (11)
13	2 5 9 1 8 6 . 3 5 1		h _A 6
25	2 9 5 2 7 5 . 5 9 0		h _A 7
37	3 4 4 4 8 8 . 1 8 8		h _A 8
49	5 2 4 9 3 4 . 3 8 2	73	h _A 9
61	5 5 7 7 4 2 . 7 8 1	3	h _A 10
1	4 7 .		TBL (16)
13	6 5 6 1 6 7 . 9 7 8		h _A 11
25	1 2 8 3 0 0 0 .		h _A 12
37	2 6 2 3 2 8 9 . 0 6 3		h _A 13
49	1 7 1 6 . 4 8 2 7	73	R _A 1
61	1 7 1 6 . 4 8 2 7	4	R _A 2

70

FORTRAN FIXED 10 DIGIT DECIMAL DATA

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

LINE NO.	NUMBER	IDENTIFICATION	DESCRIPTION	DO NOT KEY PUNCH
1	5 2 .		TBL (21)	
13	1 7 1 6 . 4 8 2 7		R _A 3	ft ² /sec ² °R
25	1 7 1 6 . 4 8 2 7		R _A 4	ft ² /sec ² °R
37	1 7 1 6 . 4 8 2 7		R _A 5	ft ² /sec ² °R
49	1 7 1 6 . 4 8 2 7	73	R _A 6	ft ² /sec ² °R
61	1 7 1 6 . 4 8 2 7	73	R _A 7	ft ² /sec ² °R
1	5 7 .	5	TBL (26)	
13	1 7 1 6 . 4 8 2 7		R _A 8	ft ² /sec ² °R
25	1 7 1 6 . 4 8 2 7		R _A 9	ft ² /sec ² °R
37	1 7 1 6 . 4 8 2 7		R _A 10	ft ² /sec ² °R
49	1 7 1 6 . 4 8 2 7	73	R _A 11	ft ² /sec ² °R
61	1 7 1 6 . 4 8 2 7	73	R _A 12	ft ² /sec ² °R
1	6 2 .	6	TBL (31)	
13	1 7 1 6 . 4 8 2 7		R _A 13	ft ² /sec ² °R
25	5 2 0 . 4 7 1		T _A 1	°R
37	3 8 9 . 9 8 8		T _A 2	°R
49	3 8 9 . 9 8 8	73	T _A 3	°R
61	5 0 8 . 7 8 8	73	T _A 4	°R
1	6 7 .	7	TBL (36)	
13	5 0 8 . 7 8 8		T _A 5	°R
25	2 9 8 . 1 8 8		T _A 6	°R
37	2 9 8 . 1 8 8		T _A 7	°R
49	4 0 6 . 1 8 8	73	T _A 8	°R
61	2 3 8 6 . 1 8 8	73	T _A 9	°R

FORTRAN FIXED 10 DIGIT DECIMAL DATA

DECK NO.	PROGRAMMER	DATE	PAGE	of	JOB NO.	DESCRIPTION	DO NOT KEY PUNCH
1	7 2 .					TBL (41)	
13	2 5 6 6 . 1 8 8					T _B 10	°R
25	2 8 3 6 . 1 8 8					T _A 11	°R
37	4 0 3 9 . 8 5 6					T _A 12	°R
49	6 6 1 3 . 5 3 2 8					T _A 13	
61	1 . 4 0 1 1					Y _A 1	dimensionless
1	7 7 .					TBL (46)	
13	1 . 4 0 1 1					Y _A 2	dimensionless
25	1 . 4 0 1 1					Y _A 3	dimensionless
37	1 . 4 0 1 1					Y _A 4	dimensionless
49	1 . 4 0 1 1					Y _A 5	dimensionless
61	1 . 4 0 1 1					Y _A 6	dimensionless
1	8 2 .					TBL (51)	
13	1 . 4 0 1 1					Y _A 7	dimensionless
25	1 . 4 0 2 5					Y _A 8	dimensionless
37	1 . 4 1 9 5					Y _A 9	dimensionless
49	1 . 4 2 6 8					Y _A 10	dimensionless
61	1 . 4 5 4 5					Y _A 11	dimensionless
1	8 7 .					TBL (56)	
13	1 . 5 9 2 0					Y _A 12	dimensionless
25	1 . 6 5 8 3 4					Y _A 13	dimensionless
37	1 . 0 1 8 2 0 2 0					SA 1 A	dimensionless
49	0 .					x 10 ⁸	
51	. 2 2 3 3 5 8 8 5 8 7					SA 2 A	dimensionless

FORTRAN FIXED 10 DIGIT DECIMAL DATA

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

LINE NO.	NUMBER	IDENTIFICATION	DESCRIPTION	DO NOT KEY PUNCH
1	9 2 .		TBL (61)	
13	0 .			$\times 10^8$
25	. 2 4 5 6 0 5 2 3 1		$\delta_A 3 A$	dimensionless
37	1 .			$\times 10^8$
49	. 1 1 8 8 6 5 7 6	80	$\delta_A 4 A$	dimensionless
61	2 .	1 3		$\times 10^8$
1	9 7 .		TBL (66)	
13	. 5 7 5 5 7 3 3 4		$\delta_A 5 A$	dimensionless
25	3 .			$\times 10^8$
37	. 9 9 6 2 3 3 3 5	80	$\delta_A 6 A$	dimensionless
49	5 .	1 4		$\times 10^8$
61	. 1 0 3 0 6 9 4 2		$\delta_A 7 A$	dimensionless
1	1 0 2 .		TBL (71)	
13	5 .			$\times 10^8$
25	. 7 3 5 5 0 4 9 2		$\delta_A 8 A$	dimensionless
37	7 .			$\times 10^8$
49	. 3 5 7 2 6 2 5 5	80	$\delta_A 9 A$	dimensionless
61	8 .	1 5		$\times 10^8$
1	1 0 7 .		TBL (76)	
13	. 2 7 8 6 6 4 7 6		$\delta_A 10 A$	dimensionless
25	8 .			$\times 10^8$
37	. 1 4 0 6 7 3 9 1	80	$\delta_A 11 A$	dimensionless
49	8 .	1 6		$\times 10^8$
61	. 4 4 5 2 1 0 8 7		$\delta_A 12 A$	dimensionless

FORTRAN FIXED 10 DIGIT DECIMAL DATA

DECK NO. _____ PROGRAMMER _____ DATE _____ PAGE _____ of _____ JOE NO. _____

DESCRIPTION DO NOT KEY PUNCH

LINE NO.	NUMBER	IDENTIFICATION	DESCRIPTION
1	112.		TBL (81)
13	-10.		$\times 10^8$
25	.36222		<i>dimensionless</i>
37	-12.		$\times 10^8$
49			
61			
1	28.		
13	22230100.		RECID
25			
37	20902131.		
49	1.407698+16		<i>ft</i>
61			ft^3/sec^2
1	7.		
13	.05		<i>d v min</i>
25			<i>degrees</i>
37	5.		<i>d v print</i>
49			<i>degrees</i>
61			
1	201.		
13	1.		<i>hi</i>
25	0.		<i>i</i>
37			<i>degrees</i>
49			
61			

PORTION: FIXED 10 DIGIT DECIMAL DATA

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

NUMBER	IDENTIFICATION	DESCRIPTION	DO NOT KEY PUNCH
1	73	h ₃ h = 1.6 × 10 ⁶	H
13			
25			
37			
49			
61			
1	73	2.1	n ₆ ε ₁ ε ₂ <i>dimensionless</i>
13			
25			
37			
49			
61			
1	73	2.2	n ₀ Γ
13			
25			
37			
49			
61			
1	73	2.3	n ₀ Γ ₁
13			
25			
37			
49			
61			
1	73	2.4	n ₀ Γ ₁
13			
25			
37			
49			
61			

FORTRAN FIXED 10 DIGIT DECIMAL DATA

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

LINE NO.	NUMBER	IDENTIFICATION	DESCRIPTION DO NOT KEY PUNCH
1	- 2 5 1 .		Last Card of First Data Set.
13	1 .		MM
25	0 .		M
37			
49			
61			
1	- 2 8 .	2 5	Last Card of Second Data Set.
13	0 .		RECID = 0, Mark "end" on Tape 2
25			
37			
49			
61			
1	- 2 8 .	2 6	Last Card of Third Data Set.
13	1 .		RECID < 0, exit to Card-Punch Program via "CHAIN."
25			
37			
49			
61			
1	1 .	2 7	
13	2 2 2 3 0 1 0 1 .		RECID
25	1 .		WR
37	0 .		WR
49	0 .		WR
61	0 .	2 8	WR

FORTRAN FIXED 10 DIGIT DECIMAL DATA

DECK NO. _____ PAGE _____ of _____ 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		
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		

FORTRAN: FIXED 10 DIGIT DECIMAL DATA

DECK NO. _____ PROGRAMMER _____ DATE _____ PAGE _____ of _____ 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		

1.1.

3.

Ff

-17.

0.

Last Card of Data Set

Fv (do not punch v - same argument field as before)

PARAMETER LIST -

OI = 0.
HP = 160000.
EC = 0. 0.025000
GAM = 1.000
GAMQ = 0.
OM = 0.

PHI	H	RHO	GAMMA	ALPHA H1	ALPHA H2	ALPHA PA	N
5.000	1600000.	5.0380E-15	0.	0.996	4.993	-0.	8
10.000	1600000.	5.0390E-15	0.	0.983	9.944	0.	1
15.000	1600000.	5.0380E-15	0.	0.962	14.810	-0.	1
20.000	1600000.	5.0380E-15	0.	0.933	19.551	0.	1
25.000	1600000.	5.0380E-15	0.	0.896	24.126	-0.	1
30.000	1600000.	5.0380E-15	0.	0.851	28.497	0.	1
35.000	1600000.	5.0380E-15	0.	0.799	32.626	-0.	1
40.000	1600000.	5.0380E-15	0.	0.741	36.479	0.	1
45.000	1600000.	5.0380E-15	0.	0.676	40.022	-0.	1
50.000	1600000.	5.0380E-15	0.	0.605	43.227	0.	1
55.000	1600000.	5.0380E-15	0.	0.529	46.065	-0.	1
60.000	1600000.	5.0380E-15	0.	0.449	48.512	0.	1
65.000	1600000.	5.0380E-15	0.	0.365	50.549	-0.	10
70.000	1600000.	5.0380E-15	0.	0.278	52.157	0.	1
75.000	1600000.	5.0380E-15	0.	0.188	53.323	-0.	1
80.000	1600000.	5.0380E-15	0.	0.097	54.037	0.	1
85.000	1600000.	5.0380E-15	0.	0.005	54.294	-0.	1
90.000	1600000.	5.0380E-15	0.	-0.087	54.090	0.	1
95.000	1600000.	5.0380E-15	0.	-0.178	53.428	-0.	1
100.000	1600000.	5.0380E-15	0.	-0.268	52.313	0.	1
105.000	1600000.	5.0380E-15	0.	-0.355	50.755	-0.	1
110.000	1600000.	5.0380E-15	0.	-0.440	48.766	0.	1
115.000	1600000.	5.0380E-15	0.	-0.520	46.364	-0.	1
120.000	1600000.	5.0380E-15	0.	-0.597	43.570	0.	1
125.000	1600000.	5.0380E-15	0.	-0.668	40.406	-0.	1
130.000	1600000.	5.0380E-15	0.	-0.734	36.900	0.	1
135.000	1600000.	5.0380E-15	0.	-0.793	33.081	-0.	1
140.000	1600000.	5.0380E-15	0.	-0.846	28.981	0.	1
145.000	1600000.	5.0380E-15	0.	-0.891	24.636	-0.	1
150.000	1600000.	5.0380E-15	0.	-0.929	20.083	0.	1
155.000	1600000.	5.0380E-15	0.	-0.959	15.359	-0.	1
160.000	1600000.	5.0380E-15	0.	-0.981	10.505	0.	1
165.000	1600000.	5.0380E-15	0.	-0.995	5.562	-0.	1
170.000	1600000.	5.0380E-15	0.	-1.000	0.571	0.	1
175.000	1600000.	5.0380E-15	0.	-0.997	-4.424	-0.	1
180.000	1600000.	5.0380E-15	0.	-0.985	-9.381	0.	1

RECORD ID
22230100.

OM 0.
GAMQ 0.
GAM 1.0000E 00
EC 0.
HP 1600000.
D PHI PRINT 5.0000E 00

O1 0.
D PHI MIN 5.0000E-02

185.000	1600000.	5.0380E-15	-0.	-0.965	-14.259	-0.	1
190.000	1600000.	5.0380E-15	-0.	-0.937	-19.017	0.	1
195.000	1600000.	5.0380E-15	-0.	-0.900	-23.613	-0.	1
200.000	1600000.	5.0380E-15	-0.	-0.857	-28.009	0.	1
205.000	1600000.	5.0380E-15	-0.	-0.806	-32.167	-0.	1
210.000	1600000.	5.0380E-15	-0.	-0.748	-36.053	0.	1
215.000	1600000.	5.0380E-15	-0.	-0.683	-39.634	-0.	1
220.000	1600000.	5.0380E-15	-0.	-0.613	-42.878	0.	1
225.000	1600000.	5.0380E-15	-0.	-0.538	-45.760	-0.	1
230.000	1600000.	5.0380E-15	-0.	-0.458	-48.253	0.	1
235.000	1600000.	5.0380E-15	-0.	-0.375	-50.338	-0.	1
240.000	1600000.	5.0380E-15	-0.	-0.288	-51.995	0.	1
245.000	1600000.	5.0380E-15	-0.	-0.199	-53.213	-0.	1
250.000	1600000.	5.0380E-15	-0.	-0.108	-53.979	0.	1
255.000	1600000.	5.0380E-15	-0.	-0.016	-54.288	-0.	1
260.000	1600000.	5.0380E-15	-0.	0.076	-54.137	0.	1
265.000	1600000.	5.0380E-15	-0.	0.168	-53.527	-0.	1
270.000	1600000.	5.0380E-15	-0.	0.258	-52.463	0.	1
275.000	1600000.	5.0380E-15	-0.	0.345	-50.955	-0.	1
280.000	1600000.	5.0380E-15	-0.	0.430	-49.015	0.	1
285.000	1600000.	5.0380E-15	-0.	0.511	-46.659	-0.	1
290.000	1600000.	5.0380E-15	-0.	0.588	-43.908	0.	1
295.000	1600000.	5.0380E-15	-0.	0.660	-40.785	-0.	1
300.000	1600000.	5.0380E-15	-0.	0.726	-37.317	0.	1
305.000	1600000.	5.0380E-15	-0.	0.786	-33.532	-0.	1
310.000	1600000.	5.0380E-15	-0.	0.840	-29.463	0.	1
315.000	1600000.	5.0380E-15	-0.	0.886	-25.144	-0.	1
320.000	1600000.	5.0380E-15	-0.	0.925	20.612	0.	1
325.000	1600000.	5.0360E-15	-0.	0.956	-15.906	-0.	1
330.000	1600000.	5.0380E-15	-0.	0.979	-11.065	0.	1
335.000	1600000.	5.0380E-15	-0.	0.994	-6.130	-0.	1
340.000	1600000.	5.0380E-15	-0.	1.000	-1.143	0.	1
345.000	1600000.	5.0380E-15	-0.	0.997	3.854	-0.	1
350.000	1600000.	5.0380E-15	-0.	0.987	8.818	0.	1
355.000	1600000.	5.0380E-15	-0.	0.968	13.707	-0.	1
360.000	1600000.	5.0380E-15	-0.	0.940	18.481	0.	1

O1	HP	EC	GAM	GAMG	OM	
0.	160000.	0.	1.0000E 00	0.	0.	
D PHI MIN	D PHI PRINT				RECORD ID	
5.0000E-02	5.0000E 00				22230100.	
		DELTA	END DERIVATIVES			NT
		1.1942E-01	-0.006	0.940		79
	TH PER BD	WEIGHT THETA H1	MAX THETAS	54.294		
	0.	0.	1.000			
		WEIGHT THETA H2	MAX DERIVATIVES	1.000		
		-0.	0.018			

OI 0.0
 D PHI MIN 5.0000E-02
 HP 160000.0
 EC 0.0250C0
 GAM 1.0000E 00
 GAMG 0.0
 OM 0.0
 RECORD ID 22230101.

PHI	H	RHO	GAMMA	ALPHA H1	ALPHA H2	ALPHA PA	N
5.000	1602088.	4.9958E-15	0.122	0.874	4.871	-0.122	8
10.000	1608341.	4.8720E-15	0.243	0.740	9.704	-0.244	1
15.000	1618717.	4.6739E-15	0.362	0.600	14.460	-0.366	1
20.000	1633147.	4.4132E-15	0.479	0.455	19.108	-0.488	1
25.000	1651539.	4.1042E-15	0.592	0.307	23.621	-0.611	1
30.000	1673771.	3.7625E-15	0.701	0.156	27.978	-0.734	1
35.000	1699695.	3.4034E-15	0.805	0.005	32.167	-0.859	1
40.000	1729139.	3.0412E-15	0.903	-0.145	36.183	-0.986	1
45.000	1761906.	2.6879E-15	0.995	-0.293	40.024	-1.115	1
50.000	1797773.	2.3527E-15	1.080	-0.438	43.695	-1.247	1
55.000	1836495.	2.0423E-15	1.157	-0.577	47.207	-1.384	1
60.000	1877804.	1.7606E-15	1.225	-0.710	50.569	-1.525	1
65.000	1921412.	1.5095E-15	1.285	-0.836	53.795	-1.672	1
70.000	1967010.	1.2889E-15	1.335	-0.954	56.900	-1.825	1
75.000	2014273.	1.0975E-15	1.375	-1.063	59.896	-1.985	1
80.000	2062857.	9.3334E-16	1.405	-1.162	62.797	-2.152	1
85.000	2112407.	7.9366E-16	1.424	-1.252	65.616	-2.328	1
90.000	2162553.	6.7569E-16	1.432	-1.332	68.364	-2.512	1
95.000	2212918.	5.7662E-16	1.430	-1.401	71.051	-2.705	1
100.000	2263118.	4.9379E-16	1.417	-1.460	73.686	-2.907	1
105.000	2312764.	4.2478E-16	1.393	-1.508	76.277	-3.117	1
110.000	2361468.	3.6743E-16	1.358	-1.546	78.830	-3.337	1
115.000	2408844.	3.1985E-16	1.312	-1.573	81.351	-3.565	1
120.000	2454511.	2.8044E-16	1.256	-1.589	83.844	-3.802	1
125.000	2498098.	2.4785E-16	1.190	-1.597	86.312	-4.047	1
130.000	2539249.	2.2094E-16	1.115	-1.594	88.760	-4.300	1
135.000	2577620.	1.9877E-16	1.031	-1.583	91.187	-4.560	1
140.000	2612891.	1.8058E-16	0.939	-1.564	93.597	-4.826	8
145.000	2644765.	1.6574E-16	0.839	-1.537	95.989	-5.099	2
150.000	2672969.	1.5375E-16	0.732	-1.503	98.365	-5.376	1
155.000	2697260.	1.4420E-16	0.619	-1.463	100.722	-5.658	1
160.000	2717431.	1.3677E-16	0.502	-1.418	103.063	-5.943	1
165.000	2733305.	1.3123E-16	0.380	-1.369	105.384	-6.231	1
170.000	2744744.	1.2739E-16	0.255	-1.317	107.685	-6.520	1
175.000	2751647.	1.2514E-16	0.128	-1.262	109.964	-6.810	1
180.000	2753955.	1.2439E-16	0.000	-1.206	112.220	-7.099	1

185.000	2751647.	1.2514E-16	-0.128	-1.150	114.450	-7.386	1
190.000	2744744.	1.2739E-16	-0.255	-1.095	116.651	-7.671	1
195.000	2733305.	1.3123E-16	-0.380	-1.041	118.821	-7.952	1
200.000	2717431.	1.3677E-16	-0.502	-0.990	120.957	-8.228	1
205.000	2697261.	1.4420E-16	-0.619	-0.942	123.055	-8.498	1
210.000	2672969.	1.5375E-16	-0.732	-0.899	125.112	-8.760	1
215.000	2644765.	1.6574E-16	-0.839	-0.862	127.124	-9.015	1
220.000	2612892.	1.8058E-16	-0.939	-0.830	129.085	-9.260	1
225.000	2577620.	1.9877E-16	-1.031	-0.805	130.992	-9.495	1
230.000	2539249.	2.2094E-16	-1.115	-0.788	132.837	-9.718	1
235.000	2498099.	2.4785E-16	-1.190	-0.779	134.614	-9.928	1
240.000	2454511.	2.8044E-16	-1.256	-0.777	136.316	-10.125	1
245.000	2408844.	3.1985E-16	-1.312	-0.785	137.933	-10.306	1
250.000	2361469.	3.6742E-16	-1.358	-0.801	139.454	-10.471	1
255.000	2312765.	4.2478E-16	-1.393	-0.826	140.867	-10.619	1
260.000	2263118.	4.9379E-16	-1.417	-0.860	142.158	-10.747	1
265.000	2212918.	5.7662E-16	-1.430	-0.903	143.308	-10.855	1
270.000	2162553.	6.7569E-16	-1.432	-0.954	144.297	-10.940	1
275.000	2112408.	7.9366E-16	-1.424	-1.013	145.100	-11.001	1.84
280.000	2062858.	9.3333E-16	-1.405	-1.079	145.688	-11.036	1
285.000	2014274.	1.0975E-15	-1.375	-1.151	146.025	-11.041	1
290.000	1967011.	1.2889E-15	-1.335	-1.229	146.074	-11.015	1
295.000	1921412.	1.5095E-15	-1.285	-1.310	145.787	-10.953	1
300.000	1877804.	1.7606E-15	-1.225	-1.394	145.114	-10.853	1
305.000	1836495.	2.0423E-15	-1.157	-1.479	143.997	-10.711	1
310.000	1797773.	2.3527E-15	-1.080	-1.563	142.373	-10.523	1
315.000	1761906.	2.6879E-15	-0.995	-1.644	140.176	-10.284	1
320.000	1729140.	3.0412E-15	-0.903	-1.720	137.340	-9.991	1
325.000	1699695.	3.4034E-15	-0.805	-1.789	133.799	-9.639	1
330.000	1673771.	3.7624E-15	-0.701	-1.849	129.492	-9.227	1
335.000	1651539.	4.1042E-15	-0.592	-1.898	124.369	-8.750	1
340.000	1633147.	4.4132E-15	-0.479	-1.934	118.392	-8.210	1
345.000	1618717.	4.6739E-15	-0.362	-1.955	111.543	-7.605	1
350.000	1608341.	4.8720E-15	-0.243	-1.961	103.824	-6.937	1
355.000	1602088.	4.9958E-15	-0.122	-1.950	95.259	-6.212	1
360.000	1600000.	5.0380E-15	-0.000	-1.923	85.897	-5.434	1

OI HP EC GAM GAMQ OM
 0. 1600000. 0.0250C0 1.0000E 00 0. 0.

D PHI MIN D PHI PRINT
 5.0000E-02 5.0000E 00

DELTA
 5.8469E 0C

TH PER 8D WEIGHT THETA H1
 2.1596E 01 -8.7613E-07

WEIGHT THETA H2
 6.3257E-02

END DERIVATIVES ~1.923 0.185 NT
 0.031 144.739 12.440 87

MAX THETAS
 2.643

MAX DERIVATIVES 1.923 0.185

0.031

RECORD ID
 22230101.

TAPE RECORD
22230101.

WEIGHT PAR 1.000000E 00
WEIGHT GAMMA 0.

WEIGHT HI 0.

WEIGHT H2 0.

SCALING - MIN VALUE
FUNCTION -30.000
ARGUMENT 0.

MIN COUNTS 200.
100.

MAX VALUE 30.000
360.000

MAX COUNTS 800.
820.

EDGE COUNTS 850.
850.

CARD FIELD 2.
1.

DECK ID - 0.

D WHS 5.846886 -0.000001 0.063257

FNCTN	0.	-0.000145	-0.001167	-0.003979	-0.009561	-0.018978	-0.033398	-0.054080	-0.082365	-0.119653
11	-0.167370	-0.226938	-0.299741	-0.367090	-0.490200	-0.610158	-0.747910	-0.904235	-1.079742	-1.274852
21	-1.489798	-1.724617	-1.979151	-2.253047	-2.545762	-2.856563	-3.184539	-3.528604	-3.887509	-4.259852
31	-4.644092	-5.038556	-5.441461	-5.850924	-6.264979	-6.681594	-7.098686	-7.514138	-7.925817	-8.331587
41	-8.729327	-9.116940	-9.492370	-9.853608	-10.19871	-10.52577	-10.83298	-11.11858	-11.38085	-11.61814
51	-11.82881	-12.01126	-12.16383	-12.28485	-12.37254	-12.42501	-12.44018	-12.41578	-12.34928	-12.23790
61	-12.07857	-11.86800	-11.60273	-11.27925	-10.89417	-10.44445	-9.927682	-9.342381	-8.688284	-7.966612
71	-7.180254	-6.333826	-5.433610	15.00000	20.00000	25.00000	30.00000	35.00000	40.00000	45.00000
ARG	0.	5.000000	10.00000	15.00000	20.00000	25.00000	30.00000	35.00000	40.00000	45.00000
11	50.00000	55.00000	60.00000	65.00000	70.00000	75.00000	80.00000	85.00000	90.00000	95.00000
21	100.0000	105.0000	110.0000	115.0000	120.0000	125.0000	130.0000	135.0000	140.0000	145.0000
31	150.0000	155.0000	160.0000	165.0000	170.0000	175.0000	180.0000	185.0000	190.0000	195.0000
41	200.0000	205.0000	210.0000	215.0000	220.0000	225.0000	230.0000	235.0000	240.0000	245.0000
51	250.0000	255.0000	260.0000	265.0000	270.0000	275.0000	280.0000	285.0000	290.0000	295.0000
61	300.0000	305.0000	310.0000	315.0000	320.0000	324.9999	329.9999	334.9999	339.9999	344.9999
71	349.9999	354.9999	359.9999	364.9999	370.0000	375.0000	380.0000	385.0000	390.0000	395.0000

8

BIBLIOGRAPHY

1. Minzner, R. A., "Higher Atmospheric Densities and Temperatures Demanded by Satellite and Recent Rocket Measurements", American Rocket Society Preprint Number 781-59.
2. Coddington, E. A. and Levinson, N., Theory of Ordinary Differential Equations, McGraw-Hill, New York, 1955, pp. 218-220.
3. Gill, S., "A Process for the Step-by-Step Integration of Differential Equations in an Automatic Digital Computing Machine", Proceedings of the Cambridge Philosophical Society, Volume 47, 1951, pp. 96 to 108.
4. Milne, W. E., Numerical Calculus, Princeton University Press, Princeton, 1949.
5. Milne, W. E., Numerical Solution of Differential Equations, John Wiley & Sons, Inc., New York, 1953.
6. The Fortran Automatic Coding System for the IBM 704 EDPM, International Business Machines Corporation, New York, 1956.
7. Reference Manual - Fortran II - for the IBM 704 Data Processing System, International Business Machines Corporation, New York, 1958.