NASA Technical Memorandum 81636

NASA-TM-81636 19810008913

Computer Program Documentation for the Dynamic Analysis of a Noncontacting Mechanical Face Seal

B. M. Auer and I. Etsion
Lewis Research Center
Cleveland, Ohio

November 1980

NASA

E-650

COMPUTER PROGRAM DOCUMENTATION FOR THE DYNAMIC ANALYSIS OF A NONCONTACTING MECHANICAL FACE SEAL

by

B. M. Auer and I. Etsion
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135

SUMMARY

A computer program is presented which gives a numerical solution to a set of second order ordinary differential equations which describe the dynamic behavior of a noncontacting mechanical face seal. The equations were based on an analytical solution of the Reynolds equation for the fluid film pressure combined with a numerical solution, using time integration techniques, of the equations of motion of the flexibly mounted seal ring.

The face seal consists of two discs, a seal seat attached to a shaft and rotating with it and a seal ring attached to a stationary mount. The seal ring is flexibly supported to allow self alignment and tracking of the mating seal seat. Program input includes variables describing the geometry and the physical characteristics of the face seal and seal operating conditions. Output from the program includes velocities and displacements of the seal ring along and about the axes of an inertial reference system.

Five appendixes are included, containing the nomenclature (A), the dictionary of Fortran symbols (B), the computer program (C), the program flow chart (D), and a sample problem (E).

INTRODUCTION

A dynamic analysis of a noncontacting, coned, mechanical face seal is presented in Ref. [1]. The seal consists of two discs, a rotor which is attached to the shaft and rotates with it (see Fig. 1) and a stator which is flexibly supported to allow self alignment and tracking of the rotor. The stator, or seal ring, can move axially and tilt about two orthogonal diameters while its circumferential rotation is prevented by anti-rotation locks. The motion of the flexibly mounted stator is controlled by the forces and moments acting upon it, by its mass inertia, and by the stiffness and damping properties of the seal system.

The dynamic analysis in Ref. [1] is based on a numerical solution of the equations of motion of the stator. For this purpose, a computer program was prepared. Input data for the program describes the geometry and the physical characteristics of the face seal, as well as its operating conditions. These

N81-17435

are the inner and outer seal ring radii, the seal ring mass and radius of gyration, the coning angle, the external moment acting on the seal ring, the spring constant of the flexible support, the radius of the springs location, the seal design clearance, the inside and outside pressures, and the shaft speed. The computed results at any time t, include the seal ring axial and angular displacement and the minimum clearance between the ring and the seat. The first time derivatives of the ring displacements are also computed.

The object of this report is to describe the equations of motion of the face seal and the method of solution of these equations, to describe the input and output of the program, and to document the computer program.

STATEMENT OF THE PROBLEM

The program solves the set of three second order ordinary differential equations, (1), (2), and (3) for α_x, α_y, and z as a function of time. The full set of equations, (1) through (28), is listed below, and this is followed by a brief description of their derivation.

$$\ddot{\alpha}_x = (M_x + M_e)/I \tag{1}$$

$$\ddot{\alpha}_y = (M_y/I \tag{2}$$

$$\ddot{z} = F_z/m \tag{3}$$

$$I = \frac{1}{2} mR_g^2 \tag{4}$$

$$R_m = \frac{1. + R_i}{2} \tag{5}$$

$$M_x = M_1 \cos(\psi) - M_2 \sin \psi \tag{6}$$

$$M_y = M_1 \sin(\psi) + M_2 \cos \psi \tag{7}$$

$$M_1 = \int_0^{2\pi} \int_{R_i}^{1} PR^2 \cos\theta \, dR_d\theta - \frac{K}{2} R_{sp}^2 \gamma \frac{C_o}{r_o} \tag{8}$$

$$M_2 = \int_0^{2\pi} \int_{R_i}^{1} PR^2 \sin\theta \, dR_d\theta \tag{9}$$

$$F_z = \int_0^{2\pi} \int_{R_i}^{1} PR\ dR\ d\theta - F_c \tag{10}$$

$$F_c = K \frac{C_o}{r_o} Z + \pi R_m (1 - R_i) \left[P_o + P_i + \frac{\beta(1 - R_i)}{2 + \beta(1 - R_i)} (P_o - P_i) \right] \tag{11}$$

$$P = P_d + P_s \qquad \text{If } (P < 0), \text{ then } P = 0 \tag{12}$$

$$P_d = \frac{1}{(1 + Z)^3} \left[\left(\frac{1}{2} - \dot{\psi} \right) \gamma R_m \sin\theta - \dot{Z} - \dot{\gamma} R_m \cos\theta \right] A \left(\frac{R - R_i}{1 - R_i} \right) \tag{13}$$

$$P_s = P_o - (P_o - P_i) \frac{H_i^2 (H_o + H)}{2} A \tag{14}$$

$$\gamma = \sqrt{\alpha_x^2 + \alpha_y^2} \tag{15}$$

$$\dot{\gamma} = (\dot{\alpha}_x \alpha_x + \dot{\alpha}_y \alpha_y)/\gamma \tag{16}$$

$$\psi = \tan^{-1}(\alpha_y/\alpha_x) \tag{17}$$

$$\dot{\psi} = (\dot{\alpha}_y \alpha_x - \dot{\alpha}_x \alpha_y)/\gamma^2 \tag{18}$$

$$\varepsilon = \gamma/(1 + Z) \tag{19}$$

$$\delta = \beta/(1 + Z) \tag{20}$$

$$H = 1 + \varepsilon R \cos(\theta) + \delta(R - R_i) \tag{21}$$

$$H_i = 1 + \varepsilon R_i \cos(\theta) \tag{22}$$

$$H_o = 1 + \varepsilon \cos\theta + \delta(1 - R_i) \tag{23}$$

$$H_m = (H_i + H_o)/2 \tag{24}$$

$$A = (1 - R)/\left[H_m H^2 (1 - R_i) \right] \tag{25}$$

$$H_{MIN1} = 1 - \varepsilon R_i \tag{26}$$

$$H_{MIN2} = 1 - \varepsilon + \delta(1 - R_i) \tag{27}$$

$$H_{MIN} = MIN(H_{MIN1}, H_{MIN2}) \tag{28}$$

The theoretical model is shown in Fig. 2. The seal seat is parallel to the xy plane of an inertial reference system xyz and is rotating at a constant angular velocity ω about the z axis. The primary seal ring has three degrees of freedom. It can move axially along the z axis and tilt about the x and y axes. A rotating coordinate system 123 coincides with the principal axes of the ring so that axis 3 is perpendicular to the plane of the ring. The coordinate system 123 rotates in the inertial reference xyz so that axis 1 always remains in the plane xy and axis 2 is directed to the instantaneous point of maximum film thickness. Thus, the orientation of coordinate system 123 in the inertial reference xyz can be defined by the nutation angle γ^* measured from axis z to axis 3, and the precession angle ψ measured from axis x to axis 1.

Equations (1), (2), and (3) are the general equations of motion of the seal ring where $\ddot{\alpha}_x$ and $\ddot{\alpha}_y$ are the accelerations of the ring about the x and y axes, respectively, and $\ddot{Z}$ is the axial displacement acceleration. Equations (6) and (7) transform the ring moments from the rotating 123 coordinate system to the xyz inertial system. Equations (8) and (9) are used to calculate the moments M_1 and M_2 where $\frac{K}{2} R_{sp}^2 \gamma \frac{C_o}{r_o}$ is the restoring moment provided by the flexible support. Equation (10) calculates the axial force, F_z, and Eq. (11) calculates the closing force, F_c, contributed by both the flexible support and the pressure balance of the seal. Equations (12), (13), and (14) calculate the fluid film pressure distribution, where P_s is the hydrostatic pressure component and P_d is the hydrodynamic and squeeze pressure components. Equations (21), (22), (23), and (24) give the film thickness distributions H, H_i, H_o, and H_m, and Eqs. (19) and (20) give the tilt and the coning parameters ε and δ.

The velocities $\dot{\alpha}_x$, $\dot{\alpha}_y$, and $\dot{Z}$ and the new displacements α_x, α_y, and Z are found from the solution of the system (1), (2), and (3) to give the transient dynamics of the seal. From these values, the nutation γ and the precession ψ and their derivatives can be found by using Eqs. (15), (16), (17), and (18). After each time step, the minimum film thickness, H_{MIN}, is calculated from Eqs. (26), (27), and (28), and is tested against a failure criterion. Since contact can take place on either the inner or the outer radius of the ring, the minimum film thickness is set equal to the smaller of H_{MIN1} or H_{MIN2}.

METHOD OF SOLUTION

The program solves the set of three second order differential equations, (1), (2), and (3), where initial conditions are given for α_x, α_y, Z, $\dot{\alpha}_x$, $\dot{\alpha}_y$, and $\dot{Z}$. These equations were rewritten in terms of the six new variables Y_1, Y_2, Y_3, Y_4, Y_5, and Y_6 as the following set of six first order equations, with initial conditions given for Y_1, Y_2, Y_3, Y_4, Y_5, and Y_6.

$$\dot{Y}_1 = \dot{\alpha}_x \tag{29}$$

$$\dot{Y}_2 = \dot{\alpha}_y \tag{30}$$

$$\dot{Y}_3 = \dot{Z} \tag{31}$$

$$\dot{Y}_4 = (M_x + M_e)/I \tag{32}$$

$$\dot{Y}_5 = M_y/I \tag{33}$$

$$\dot{Y}_6 = F_z/m \tag{34}$$

The dependent variables are given by Eqs. (35) through (40) as follows:

$$Y_1 = \alpha_x \tag{35}$$

$$Y_2 = \alpha_y \tag{36}$$

$$Y_3 = Z \tag{37}$$

$$Y_4 = \dot{\alpha}_x \tag{38}$$

$$Y_5 = \dot{\alpha}_y \tag{39}$$

$$Y_6 = \dot{Z} \tag{40}$$

To find the solution of Eqs. (29) through (34), the program uses the GEAR differential equation solver software package in the Lewis Research Center's International Mathematical and Statistical (IMSL) software library of scientific Univac 1100 Fortran V programs. The Lewis Research Center subscribes to this software package which is supplied by the International Mathematical and Statistical Libraries, Inc. software company in Houston, Texas.

The GEAR package finds the numerical solution of initial value problems for systems of first order ordinary differential equations. This package has the option of using two basic methods of solution, either an implicit multistep variable order Adams predictor corrector method or an implicit multistep backward differentiation formula method, also known as Gear's stiff method [2]. Let Y_1, Y_2, Y_3, Y_4, Y_5, and Y_6 be denoted by $\overline{Y}$ and let $\dot{Y}_1$, $\dot{Y}_2$, $\dot{Y}_3$, $\dot{Y}_4$, $\dot{Y}_5$, and $\dot{Y}_6$ be denoted by $\dot{\overline{Y}}$. Then the call to the GEAR package is CALL DGEAR (N6, FCN, FCNJ, T, DELH, Y, TEND, TOL, METH, MITER, INDEX, IWK, WK, IER), where the parameters passed in the call statement are defined as follows:

N6 - Input giving the number of first order differential equations to be solved.

FCN - User supplied subroutine for evaluating the function $\dot{\overline{Y}}$.

FCNJ - User supplied subroutine to compute the N6xN6 Jacobian Matrix of the partial derivatives of $\dot{\overline{Y}}$ with respect to $\overline{Y}$. FCNJ is needed only when miter = 1.

T - The independent variable. T is input on the first call only, and on the first call passes the initial value to the program. T is output on subsequent calls and passes to the user the current value of the independent variable at which integration has been completed.

DELH - On the first call only, DELH is input and contains the next step size in T to be used. On subsequent calls, DELH is output and contains the step size actually used.

Y - The dependent variable, $\overline{Y}$. On input, Y supplies the initial values of $\overline{Y}$. On output, Y contains the value of $\overline{Y}$ computed at TEND.

TEND - Input giving the value of T at which a solution for $\overline{Y}$ is desired.

TOL - Input giving the relative error bound.

METH - Input indicating whether the Adams Method or the Backward Differentiation Formulae Method is to be used.

MITER - An input indicating whether the chord method or functional iteration is to be used and whether a finite difference or analytic Jacobian is to be used with the chord method.

INDEX - Both an input and an output parameter. Index tells whether this is the first call to DGEAR and whether the integration was successful.

IWK - An integer work vector.

WK - A real work vector.

IER - An output error parameter telling the user whether the integration was halted due to an error. The value of IER indicates the error encountered, as described on the following pages.

The GEAR package is called from the MAIN program and before the first call, the parameters N6, DELH, TOL, METH, MITER, and INDEX are assigned values. On the first call, T is set equal to TZRO, TEND is set equal to TZRO + DTEND, and the Y array is assigned the initial values for the dependent variables. The initial value of the step size, DELH, also must be chosen. This value should be considerably smaller than the average value expected for the problem since DGEAR must begin with a not so highly efficient first-order method. For the first step, as for every step, DGEAR tests for the possibility that the step size was too large to pass the error test (based on TOL), and so adjusts the step size down automatically. This downward adjustment, if any, is noted by IER having the values of 66 or 67. The relative error bound, TOL, should be assigned a value. TOL is an estimate of the local truncation error, as distinguished from the global truncation error. The local truncation error is the error resulting from taking a single step, starting with data which is regarded as exact. TOL should be at least an order of magnitude larger than the unit roundoff but generally not larger than 0.001.

DGEAR solves the differential equations from T = TZRO to T = TZRO + DTEND with the given initial conditions. The value of IER, which is returned to the MAIN program from DGEAR, is tested and the program stops if IER is between 132 and 136. The output error parameter, IER, informs the user of errors encountered by the GEAR package, if any. If the program failed to pass the error test after reducing DELH by a factor of 1.0E10 from its initial value, then IER is set equal to 132. If the program failed to achieve corrector convergence after reducing DELH by a factor of 1.0E10 from its initial value, then IER is set equal to 133. Incorrect input to the GEAR package is flagged by IER = 135 or 136. If IER is less than 132 or greater than 136, the desired output is printed, TEND is reset to TEND = TEND + DTEND, and DGEAR is called again. This process is continued until there is seal failure, TEND = TSTOP, or IER is between 132 and 136. To solve the problem, the GEAR package used the implicit Adams Method and used the following values for its parameters; N6 = 6, DELH = 0.001, TOL = 0.0001, METH = 1, MITER = 0, and INDEX = 1.

To find the moments M_1 and M_2 and the force F_z, the double integrals of Eqs. (8), (9), and (10) have to be evaluated. This was done by the application of the Guassian quadrature integration formula, one of a family of powerful, accurate, numerical integration techniques [3]. To evaluate these integrals, the Fortran program SQUAD1 was used. This program was written at the Lewis Research Center by Charles Goldstein, and it evaluates integrals by repeated application of Gauss' formula [4].

The call to obtain the inner integrals, which give integration with respect to R, is CALL SQUAD1 (MODE, NR, RI, DD1, XR, YR1, ANS1), where the parameters are defined as follows.

MODE - =1, then obtain the arguments XR(NR)
=2, then use the evaluations of the integrand YR1[XR(NR)] to obtain ANS1

NR - The number of points of integration; N = 3(1)16

RI - The lower limit of integration

DD1 - The upper limit of integration

SR - An array containing the variable of integration

YR1 - An array containing values of the integrand

ANS1 - The result of the integration; i.e., the value of the integral

The call to obtain the outer integrals, which gives integration with respect to θ, is CALL SQUAD1 (MODE, NTH, TH1, TH2, XTH, YTH1, ANS1), where the parameters are defined as follows.

MODE - =1, then obtain the arguments XTH(NTH)
=2, then use the evaluations of the integrand YTH1[XTH(NTH)] to obtain ANS1

NTH - The number of points of integration per quadrant; NTH = 3(1)16

TH1 - The lower limit of integration

TH2 - The upper limit of integration

XTH - An array containing the variable of integration

YTH1 - An array containing values of the integrand

ANS1 - The result of the integration; i.e., the value of the integral

The program used 5 points of integration for the inner integral (NR = 5), and 8 points of integration for each quadrant of the outer integral (NTH = 8). Larger values were tried for NR and NTH (e.g., NR = 11 and NTH = 16) and the values of the resulting integrals were compared. It was found that using NR = 5 and NTH = 8 gave sufficient accuracy for the integrals. The variables NR and NTH can take on values between 3 and 16. To compute the outer integrals, whose limits of integration are 0° and 360°, SQUAD1 was used four times over the quadrants 0° to 90°, 90° to 180°, 180° to 270°, and 270° to 360°. This was done because NTH can have a maximum value of only 16, and applying SQUAD1 over four intervals allowed more points to be used over the total interval of integration 0° to 360°. It was also done so that the program would avoid computing the integrand at the points 0° and 180°. The routine SQUAD1 does not compute the integrand at the end points of the interval over which SQUAD1 is being used, so that computation at these points was avoided. Singularities can only occur at 0° and 180°, but because of the complexity of the integrand, it could not easily be determined whether singularities do occur. As a precaution, computation of the integrand was thus avoided at $\theta = 0^{\circ}$ and 180°.

FORTRAN PROGRAM

General Description

The foregoing analysis has resulted in a computer program for the analysis of a noncontacting mechanical face seal. The MAIN executive program handles most of the program flow and logic. It assigns values to program parameters, reads and writes program input, initializes the GEAR package and repeatedly calls the GEAR package to move ahead in time by DTEND at each call. After each call, it writes the output and tests for seal failure or a failure of the GEAR package. The program stops when a failure is detected or when TSTOP is reached. The subroutine FQUAD does most of the computing for the functions which are used in the DGEAR package.

A brief explanation of the function of each program module follows.

DGEAR the general GEAR package for solving a system of first order ordinary differential equations

ERMIN a subroutine called by ERTST to determine when H_{MIN} varies by less than HSTAB during several revolutions. If this condition is met, then ERMIN sends a flag through ERTST to MAIN

ERTST a subroutine called by the MAIN program to compute H_{MIN} and determine if there is seal failure or GEAR package failure

FCN a subroutine which computes the $\dot{\bar{Y}}$ array

FCNJ a dummy subroutine passed to the GEAR package

FQUAD a subroutine used to compute the integrand of the double integrals appearing in Eqs. (8) through (10)

MAIN the Main executive program

SQUAD1 a subroutine called by FQUAD to perform a single Gaussian integration

WRT a subroutine called by the MAIN program which writes the output Z, γ, ψ, $\dot{Z}$, $\dot{\gamma}$, $\dot{\psi}$, α_x, α_y, H_{MIN}, T, $\dot{\alpha}_x$, $\ddot{\alpha}_x$, $\dot{\alpha}_y$, $\ddot{\alpha}_y$, $\ddot{Z}$, DELH, INDEX, IER, and FC after each call to the GEAR package

Using the Program

The program always starts by reading program input through the two namelists, INPUT1 and INPUT2. The input in these namelists are described below (all symbols are repeated in appendix C).

Namelist INPUT1

DELH a parameter in the GEAR package containing the first step size to be tried

DTEND for each call to the GEAR package, solve the differential equations from TEND to TEND + DTEND

HMSTOP the program is stopped when HMIN becomes less than HMSTOP

HSTAB the program is stopped if HMIN vairies by less than HSTAB for several revolutions

INDEX a parameter in the GEAR package indicating the type of call being made to the package

METH a parameter in the GEAR package call indicating the method of solution to be used

MITER a parameter in the GEAR package call indicating the iteration method to be used

NPLT if punch is true, then write every NPLT-TH array point into dataset 15 to be saved to make a movie

NR a parameter in the SQUAD1 call indicating the number of points of integration used to obtain the inner integrals

NTH a parameter in the SQUAD1 call indicating the number of points of integration used to obtain the outer integrals

PUNCH a logical variable. If PUNCH = TRUE, then write and save output in dataset 15

READ1 a logical variable. If READ1 = TRUE, then read in and process another case

TOL a parameter in the GEAR package call giving the relative error bound

TSTOP the differential equations are solved to TEND = TSTOP

TZRO the initial value of time, the independent variable

YIN an input array containing the initial values for α_x, α_y, Z, $\dot{\alpha}_x$, $\dot{\alpha}_y$, and $\dot{Z}$

Namelist INPUT2

BETA normalized coning angle, β

FKROC the dimensionless parameter, $K \frac{C_o}{r_o}$, which is the spring constant multiplied by the equilibrium center-line clearance divided by r_o

FM dimensionless mass, m

FME dimensionless external moment, M_e

PI dimensionless pressure at the inner radius of the ring, P_i

PO dimensionless pressure at the outer radius of the ring, P_o

RG dimensionless ring radius of gyration, R_g

RI dimensionless inner radius, R_i

RSP dimensionless radius of the location of the springs, R_{sp}

If the logical variable READ1 is TRUE, then another seal case is read into the MAIN program through the Namelists INPUT1 and INPUT2 and this case is processed. If READ1 is read in as FALSE, then the program stops.

APPENDIX A

NOMENCLATURE

A	$(1 - R)/H_m H^2 (1 - R_i)$
C	seal center-line clearance
C_o	equilibrium center-line clearance
F_c^*	closing force
F_c	dimensionless closing force, F_c^*/Sr_o^2
F_z^*	axial force
F_z	dimensionless axial force, F_z^*/Sr_o^2
H	dimensionless film thickness, h/C
H_{min}	dimensionless minimum film thickness
h	film thickness
I^*	ring mass moment of inertia about a diameter
I	dimensionless moment of inertia, $I^* \omega^2 C_o/Sr_o^4$
K^*	spring constant
K	dimensionless spring constant, K^*/Sr_o
M^*	moment
M_e^*	external moment
M_e	dimensionless external moment
M	dimensionless moment, M^*/Sr_o^3
m^*	ring mass
m	dimensionless mass, $m^* \omega^2 C_o/Sr_o^2$
n	number of seal revolutions, $t/2\pi$
P	dimensionless pressure, p/S
p	pressure
R	dimensionless radius, r/r_o

r	radial coordinate
r_g	ring radius of gyration
R_g	dimensionless ring radius of gyration
S	seal parameter, $6\mu\omega(r_o/C_o)^2(1 - R_i)^2$
t^*	time
t	dimensionless time, ωt^*
Z^*	axial displacement
Z	dimensionless displacement, Z^*/C_o
α^*	tilt angle
α	normalized tilt, $\alpha^* r_o/C_o$
β^*	coning angle
β	normalized coning, $\beta^* r_o/C_o$
γ^*	nutation
γ	normalized nutation, $\gamma^* r_o/C_o$
δ	coning parameter, $\beta^* r_o/C$
ε	tilt parameter, $\gamma^* r_o/C$
θ	angular coordinate
μ	viscosity
ψ	precession
ω	shaft angular velocity

Subscripts:

1,2,3	axes 1, 2, or 3, respectively
d	hydrodynamic
i	inner radius
m	mid radius
o	outer radius

s	hydrostatic
sp	springs
x,y,z	axes, x, y, or z, respectively

APPENDIX B

FORTRAN SYMBOLS

ANS1	a parameter in the SQUAD1 call containing the value of the integral
BETA	normalized coning angle, β
CAX	normalized tilt, α_x, wrt. the x axis
CAY	normalized tilt, α_y, wrt. the y axis
CAXD	the first derivative of α_x
CAYD	the first derivative of α_y
CZ	the axial dimensionless displacement of the ring, Z
CZD	the first derivative of Z
DD1	the upper limit of integration in the SQUAD1 call
DEL	the coning parameter, δ
DELH	a parameter in the GEAR package containing the step size to be used on the first call to DGEAR and the step size actually used on subsequent calls
DTEND	for each call to the GEAR package, solve the differential equation from TEND to TEND + DTEND
EPS	the tilt parameter, ε
FC	the dimensionless closing force, Fc
FCN	a subroutine used by the GEAR package for evaluating the function $\dot{\bar{Y}}$
FCNJ	a DUMY subroutine used by the GEAR package
FI	the dimensionless moment of inertia, I
FKROC	the parameter $K \frac{C_o}{r_o}$, which is the dimensionless spring constant multiplied by the equilibrium center-line clearance divided by r_o
FM	dimensionless mass, m
FM1	dimensionless moment, M_1, wrt. axis 1
FM2	dimensionless moment, M_2, wrt. axis 2
FME	dimensionless external moment, M_e

FMX	dimensionless moment, M_x, wrt. axis x
FMY	dimensionless moment, M_y, wrt. axis y
FZ	dimensionless axial force, F_z
GAM	normalized nutation, γ
GAMD	the first derivative of the normalized nutation, $\dot{\gamma}$
H	dimensionless film thickness, H
HI	dimensionless film thickness at the inner radius, H_i
HO	dimensionless film thickness at the outer radius, H_o
HM	the average of HI and HO
HMIN	dimensionless minimum film thickness, H_{MIN}
HMIN1	dimensionless minimum film thickness at the inner radius, H_{MIN1}
HMIN2	dimensionless minimum film thickness at the outer radius, H_{MIN2}
HMSTOP	the program is stopped when HMIN becomes less than HMSTOP
HSTAB	the program is stopped if HMIN varies by less than HSTAB for several revolutions
IER	an output from the GEAR package indicating any errors encountered while using the GEAR package
INDEX	an output parameter from the GEAR package call indicating whether the integration was successful, and an input to the GEAR package indicating the type of call being made to the package
IWK	a parameter in the GEAR package call which is used as an integer work vector
IWRT	a parameter indicating whether debug output is wanted
LOGER	a logical variable returned from subroutine ERTST. LOGER = FALSE means that the program should be stopped
METH	a parameter in the GEAR package call indicating whether the Adams method or the backward differentiation formulae method is to be used
MITER	an input in the GEAR package call indicating whether the chord method or functional iteration is to be used and whether a finite difference or analytic Jacobian is to be used with the chord method

N6 a parameter in the GEAR package call indicating the number of first order differential equations to be solved

NPLT a program input indicating how much output to save for movie generation

NR a parameter in the SQUAD1 call indicating the number of points of integration used to obtain the inner integrals

NTH a parameter in the SQUAD1 call indicating the number of points of integration used to obtain the outer integrals

P dimensionless pressure

PD the dimensionless hydrodynamic and squeeze pressure components, P_d

PI the dimensionless pressure at the inner radius of the ring, P_i

PO the dimensionless pressure at the outer radius of the ring, P_o

PS the dimensionless hydrostatic pressure component, P_s

PSI the precession, ψ

PSID the first derivative of the precession, $\dot{\psi}$

PUNCH a logical variable. If punch = true, then write output into dataset 15 to be saved to make movies

RAX, RAY,
RTANG,
RZ arrays in which to save output for movie generation

READ1 a logical variable. If READ1 = TRUE, then read in and process another case

RG the dimensionless ring radius of gyration

RI the dimensionless inner radius, R_i

RM the dimensionless mid radius, R_m

RSP the dimensionless radius of the location of the springs, R_{sp}

T dimensionless time, the independent variable

TEND a parameter to the GEAR package giving the value of T at which a solution for $\overline{Y}$ is desired

TH1 a parameter in the SQUAD1 call giving the lower limit of integration of the outer integral

TH2 — a parameter in the SQUAD1 call giving the upper limit of integration of the outer integral

TOL — a parameter in the GEAR package call giving the relative error bound

TOUCH — a logical variable. TOUCH = T means that the seal ring touches the shaft

TSTOP — the differential equations are solved to TEND = TSTOP

TZRO — the initial value of time, the independent variable

WK — a parameter in the GEAR package call containing a real work vector

XR — a parameter in the SQUAD1 call containing the variable of integration of the inner integrals

XTH — a parameter in the SQUAD1 call containing the variable of integration of the outer integrals

Y — a parameter in the GEAR package call containing the dependent variable, $\overline{Y}$

YIN — an input array containing the initial values for α_x, α_y, Z, $\dot{\alpha}_x$, $\dot{\alpha}_y$, and $\dot{Z}$

YPRIME — a parameter in the FCN call containing the first derivative of the $\overline{Y}$ array

YR1 — a parameter in the SQUAD1 call containing the values of the integrand of the inner integrals

YTH1 — a parameter in the SQUAD1 call containing the values of the integrand of the outer integrals

APPENDIX C

FORTRAN PROGRAMS

```
C   MAIN PROGRAM
C
C         THIS PROGRAM DESCRIBES THE MOTION OF A SEAL RING
C           BY SOLVING SIX FIRST ORDER DIFFERENTIAL EQUAS.  (WHICH
C           WERE DERIVED FROM THREE 2ND ORDER ORDINARY DIFFERENTIAL
C             EQUATIONS)
C
C
C         THE SYSTEM OF DIFFERENTIAL EQUAS. TO BE SOLVED IS AS FOLLOWS
C            YPRIME(1) = AX-DER1
C            YPRIME(2) = AY-DER1
C            YPRIME(3) = Z-DER1
C            YPRIME(4) = AX-DER2
C            YPRIME(5) = AY-DER2
C            YPRIME(6) = Z-DER2
C
C         WHERE THE  Y  ARRAY IS DEFINED AS FOLLOWS
C            Y(1)=AX
C            Y(2)=AY
C            Y(3)=Z
C            Y(4)= DER-AX
C            Y(5)= DER-AY
C            Y(6)= DER-Z
C
C         DTEND   INCREMENT TEND BY DTEND
C
C         HSTAB  -  STOP THE PROGRAM IF HMIN VARIES LESS THAN HSTAB
C                   DURING FIVE REVOLUTIONS
C
C         HMSTOP  - STOP THE PROGRAM IF HMIN BECOMES LESS
C                     THAN HMSTOP
C
C         IWRT  = 0   DO NOT WRITE DEBUG OUTPUT
C               = 1   WRITE COMPLETE DEBUG OUTPUT
C
C         NPLT  -  PUNCH EVERY NPLT PT. FOR THE MOIVE
C                   PLOTTING PROGRAM
C
C         NR    =  NO. OF INTEGRATION INTERVALS IN THE
C                  INTERVAL  1 TO RI
C
C         NTH       = NO. OF INTEGRATION INTERVALS
C                     IN EACH QUADRANT OF THE INTERVAL 0 TO 2*PIE
C
C         PUNCH - PUNCH=.T., THEN PUNCH OUTPUT FOR THE MOVIE
C                   PLOTTING PROGRAM
C                 PUNCH=.F., THEN DO NOT PUNCH THIS OUTPUT
C                   FOR THE MOVIE PLOTTING PROGRAM
C
C         RAX,RAY,RZ,RTANG - INFORMATION PUNCHED FROM THIS PROG.
C                   TO BE READ INTO & USED IN THE MOVIE PLOTTING
C                   PROGRAM
C
C         TEND   FOR EACH CALL TO DGEAR, FIND SOLUTION
C                AT TEND
C
C         TOUCH=.T. MEANS THAT THE SEAL RING TOUCHES THE SHAFT.
C
C         TSTOP   SOLVE DIFF. EQUAS. TO T=TSTOP
C
C         TZRO   INITIAL T
```

```
C
C
C
C
      EXTERNAL FCN,FCNJ
C
      LOGICAL  LOGER,TOUCH,READ1,PUNCH
C
      DIMENSION  YIN(6)
      DIMENSION  Y(6),WK(144),IWK(6)
      DIMENSION RAX(6200),RAY(6200),RZ(6200),RTANG(6200)
C
      COMMON/COM1/ PIE,PIE2,N4PIE,NR,NTH,IWRT
C
      COMMON/COM2/ FKROC,RO,BETA,RI,CO,
     1              RM,PO,PI,FI,FME,FM,RSP
C
      COMMON/COM3/ HMIN,HMIN1,HMIN2
C
C     COMMON/GEAR/ DUMMY(48),SDUMMY(4),IDUMMY(38)
C
      NAMELIST/INPUT1/ READ1,METH,MITER,TOL,INDEX,DELH,
     1                 TZRO,YIN,TSTOP,
     2                 DTEND,NR,NTH,HMSTOP,HSTAB,
     3                  NPLT,PUNCH
C
      NAMELIST/INPUT2/ RI,FKROC,BETA,PI,PO,
     1                 FME,FM,RG,RSP
C
      DATA READ1/.TRUE./
C
      DATA TZRO,TSTOP,DTEND,YIN /0.,13.,.1,0.,.2,4*0./
C
C
      DATA HMSTOP/.1E-20/
C
C
      NR=5
      N4PIE=4
      NTH=8
      N6=6
      IWRT=0
      PUNCH=.FALSE.
      NPLT=1
      PIE=3.14159265359
      PIE2=2.*PIE
      HSTAB=.001
C
      GO TO 14
   20 IF(.NOT.PUNCH) GO TO 14
      WRITE(15,122) RI,FKROC,BETA,PI,PO,FME,FM
      WRITE(15,124) RG,RSP,TZRO,TOL,DELH,NR,NTH,METH,MITER
      WRITE(15,122) YIN
      WRITE(15,112) JT,NPLT,TOUCH
      WRITE(15,114) (RAX(J),J=1,JT,NPLT),RAX(JT)
      WRITE(15,114) (RAY(J),J=1,JT,NPLT),RAY(JT)
      WRITE(15,114) (RZ(J),J=1,JT,NPLT),RZ(JT)
      WRITE(15,114) (RTANG(J),J=1,JT,NPLT),RTANG(JT)
C
C
   14 METH=1
      MITER=0
      INDEX=1
      TOL=.0001
      DELH=.01
      IF(.NOT.READ1) GO TO 99
      READ(5,INPUT1)
```

```
      WRITE(6,101)
      WRITE(6,INPUT1)
      READ(5,INPUT2)
      WRITE(6,INPUT2)
C
C
      FI=.5*FM*RG*RG
      RM=(1.+RI)/2.
C
C
C          INITIALIZE THE INTEGRATION
C
   30 LOGER=.TRUE.
      TOUCH=.FALSE.
      JT=1
C
      T=TZRO
      TEND=TZRO+DTEND
C
      DO 32 J32=1,6
      Y(J32)=YIN(J32)
   32 CONTINUE
C
C
C          WRITE OUTPUT HEADING
      WRITE(6,104)
      WRITE(7,106)
      WRITE(7,102)
C
      CALL ERTST(LOGER,TOUCH,Y,IER,T,DELH,DTEND,HMSTOP,HSTAB)
C
      CALL WRT(N6,JT,Y,T,DELH,TEND,INDEX,IER)
C
      IF(.NOT.PUNCH) GO TO 40
      RAX(JT)=Y(1)
      RAY(JT)=Y(2)
      RZ(JT)=Y(3)
      RTANG(JT)=TZRO
C
C          USE SUB DVOGER TO MOVE ONE STEP IN TIME
C
   40 JT=JT+1
      CALL  DGEAR(N6,FCN,FCNJ,T,DELH,Y,TEND,TOL,
     1            METH,MITER,INDEX,IWK,WK,IER)
C
      IF(.NOT.PUNCH) GO TO 54
      RAX(JT)=Y(1)
      RAY(JT)=Y(2)
      RZ(JT)=Y(3)
      RTANG(JT)=TEND
C
   54 CALL ERTST(LOGER,TOUCH,Y,IER,T,DELH,DTEND,HMSTOP,HSTAB)
C
      IF(.NOT.LOGER) GO TO 20
C
      CALL WRT(N6,JT,Y,T,DELH,TEND,INDEX,IER)
C
C
C
      TEND=TEND+DTEND
C
      IF(TEND.GE.TSTOP) GO TO 20
      GO TO 40
C
C
C
  101 FORMAT(1H1)
```

```
  104 FORMAT(1H1,6X,'CZ',9X,'GAM',8X,'PSI',8X,'CZD',8X,
     1  'GAMD',7X,'PSID',7X,'CAX',8X,'CAY',8X,'HMIN',7X,
     2  'TWRT')
C
  102 FORMAT(1H1,'INDEX',2X,'IER',5X,'AXD',8X,'AYD',9X,
     1  'ZD',9X,'T',10X,'H',9X,'AXDD',7X,'AYDD',7X,
     2  'ZDD',9X,'FC')
C
  106 FORMAT(1H0,/////)
  112 FORMAT(2I4,L4)
  114 FORMAT(8E10.4)
  122 FORMAT(7E11.4)
  124 FORMAT(5E11.4,4I5)
C
   99 STOP
      END
```

```
      SUBROUTINE ERTST(LOGER,TOUCH,Y,IER,T,DELH,DTEND,HMSTOP,HSTAB)
C
C
C          THIS SUBROUTINE TESTS THE VARIABLE IER
C                   FROM THE  DGEAR  CALL AND PRINTS
C                   THE CORRECT ERROR MESSAGE
C
C          THE SUBROUTINE ALSO COMPUTES HMIN
C
C
C
      DIMENSION Y(6)
C
      LOGICAL  LOGER,TOUCH
C
      COMMON/COM1/ PIE,PIE2,N4PIE,NR,NTH,IWRT
C
      COMMON/COM2/  FKROC,RO,BETA,RI,CO,
     1              RM,PO,PI,FI,FME,FM,RSP
C
      COMMON/COM3/ HMIN,HMIN1,HMIN2
C
C               TO COMPUTE AND TEST HMIN
C
      CAX=Y(1)
      CAY=Y(2)
      SQXY=CAX*CAX+CAY*CAY
      GAM= SQRT(SQXY)
C
      CAXX= ABS(CAX)
      CAYY= ABS(CAY)
      PSI= ATAN2(CAYY,CAXX)
      IF(CAX.LT.0.AND.CAY.GT.0) PSI=-PSI
      IF(CAX.GE.0.AND.CAY.LT.0) PSI=-PSI
C
      CPSI=CAX/GAM
C
      CZ1=1.+Y(3)
      EPS=GAM/CZ1
      DEL=BETA/CZ1
C
      HMIN1=1.-EPS*RI
      HMIN2=1.-EPS+DEL*(1.-RI)
      HMIN=HMIN1
      IF(HMIN2.LT.HMIN1) HMIN=HMIN2
      IF(HMIN.GT.HMSTOP) GO TO 20
      LOGER=.FALSE.
      TOUCH=.TRUE.
      WRITE(6,102)  HMIN,HMIN1,HMIN2
      GO TO 99
C
   20 H1=HMIN
      CALL ERMIN(DTEND,H1,LOGER,HSTAB)
C
   22 IF(IER.NE.33) GO TO 24
      LOGER=.FALSE.
      WRITE(6,104) IER
      GO TO 99
C
   24 IF(IER.LT.132.OR.IER.GT.136) GO TO 99
      LOGER=.FALSE.
      WRITE(6,104) IER
C
C
C
  101 FORMAT(1H1)
  102 FORMAT('  FATAL ERROR',5X,'HMIN='E12.5,5X,
```

```
     1          'HMIN1='E12.5,5X,'HMIN2='E12.5)
  104 FORMAT('  FATAL ERROR',5X,'IER='I5)
C
   99 RETURN
      END
```

```
      SUBROUTINE ERMIN(DTEND,H1,LOGER,HSTAB)
C
C
C         THIS SUBROUTINE TESTS HMIN.  IF HMIN DOES NOT VARY BY HSTAB
C                      IN FIVE REVOLUTIONS, THEN SET LOGER TO  .FALSE.
C
C         IN CALLING PROG. ERTST, H1=HMIN
C
C
C
      DATA IDO,HS /0,1.E20/
C
      LOGICAL LOGER
C
      IF(IDO.EQ.1)  GO TO 40
C
      IF(ABS(H1-HS).GT.HSTAB)  GO TO 50
C
      HS=H1
      TS=0.
      IDO=1
      GO TO 99
C
   50 IDO=0
      HS=H1
      GO TO 99
C
   40 IF(ABS(HS-H1).LT.HSTAB)  GO TO 55
      IDO=0
      HS=H1
      GO TO 99
C
   55 TS=TS+DTEND
      IF(TS.GT.31.416)  LOGER=.FALSE.
C
   99 RETURN
      END
```

```
      SUBROUTINE FCN(N,T,Y,YPRIME)
C
C         THIS SUB IS A PARAMETER IN THE CALL
C                TO DGEAR
C
C         THIS SUB COMPUTES YPRIME
C
      DIMENSION  Y(N),YPRIME(N)
C
      COMMON/COM1/ PIE,PIE2,N4PIE,NR,NTH,IWRT
C
      COMMON/COM2/ FKROC,RO,BETA,RI,CO,
     1             RM,PO,PI,FI,FME,FM,RSP
C
      COMMON/COM4/ CAX,CAY,CZ,CAXD,CAYD,CZD
      COMMON/COM5/ FMX,FMY,FZ,GAM,PSI,GAMD,PSID,FC,
     1             FM1,FM1C,FZ1,FM2,EPS,DEL
C
C
C         COMPUTE THE RIGHT HAND SIDE OF THE 6 FIRST ORDER
C         DIFFERENTIAL EQUATIONS
C
   20 CAX=Y(1)
      CAY=Y(2)
      CZ=Y(3)
      CAXD=Y(4)
      CAYD=Y(5)
      CZD=Y(6)
C
      CALL FQUAD
C
      YPRIME(1)=Y(4)
      YPRIME(2)=Y(5)
      YPRIME(3)=Y(6)
      YPRIME(4)=(FMX+FME)/FI
      YPRIME(5)=FMY/FI
      YPRIME(6)=FZ/FM
C
C
   10 RETURN
      END
```

```
      SUBROUTINE FCNJ(N,T,Y,PD)
C      TO COMPUTE THE N*N JACOBIAN MATRIX OF PARTIAL
C            DERIVATES FOR THE DGEAR IMSL ROUTINE
C       THIS IS A DUMY ROUTINE FOR DGEAR
C
      DIMENSION Y(N),PD(N,N)
C     DOUBLE PRECISION  T,Y,PD
C
      DUMY=DUMY
      RETURN
      END
```

```
      SUBROUTINE FQUAD
C
C
C         THIS SUBROUTINE PERFORMES DOUBLE INTEGRATION
C         USING THE GOLDSTEIN QUADRATURE ROUTINES
C         TO FIND M1,M2,AND FZ
C
C         INTEGRATE THE INNER INTEGRAL WRT  R
C         INTEGRATE THE OUTER INTEGRAL WRT  THETA
C
C         NR   = THE NO. OF PTS. IN THE R INTEGRAL
C         NTH  = THE NO. OF PTS IN EACH QUADRANT OF THE THETA
C                INTEGRAL.  WE INTEGRATE FROM 0 TO 2*PIE,
C                & USE NTH PTS. IN EACH QUADRANT
C
C         IWRT  = 0  DO NOT WRITE DEBUG OUTPUT
C               =1   WRITE DEBUG OUTPUT
C
C
      DIMENSION XTH(16),XR(16),YTH1(16),YTH2(16),
     1          YTH3(16),YR1(16),YR2(16),YR3(16)
C
      COMMON/COM1/ PIE,PIE2,N4PIE,NR,NTH,IWRT
C
      COMMON/COM2/ FKROC,RO,BETA,RI,CO,
     1             RM,PO,PI,FI,FME,FM,RSP
C
      COMMON/COM4/ CAX,CAY,CZ,CAXD,CAYD,CZD
C
      COMMON/COM5/ FMX,FMY,FZ,GAM,PSI,GAMD,PSID,FC,
     1             FM1,FM1C,FZ1,FM2,EPS,DEL
C
C
      FN4PIE=N4PIE
      PIEN4=2.*PIE/FN4PIE
C
      DD1=1.
      FM1=0.
      FM2=0.
      FZ1=0.
C
      SQXY=CAX*CAX + CAY*CAY
      GAM=  SQRT(SQXY)
C
      CAXX= ABS(CAX)
      CAYY= ABS(CAY)
      PSI= ATAN2(CAYY,CAXX)
      IF(CAX.LT.0.AND.CAY.GT.0) PSI=-PSI
      IF(CAX.GE.0.AND.CAY.LT.0) PSI=-PSI
C
      CPSI=CAX/GAM
      SPSI=CAY/GAM
C
      GAMD=CAXD*CPSI + CAYD*SPSI
      PSID=(CAYD*CPSI - CAXD*SPSI)/GAM
      IF(IWRT.GE.1) WRITE(11,116) CAX,CAY,CZ,CAXD,CAYD,CZD,GAM,GAMD
C
      RIM1=1.-RI
      Z1=1.+CZ
      EPS=GAM/Z1
      DEL=BETA/Z1
      EPSRI=EPS*RI
      DELRI=DEL*RIM1
      Z3R= 1./Z1/Z1/Z1/RIM1
      PD1=(.5-PSID)*GAM*RM
      PD2=GAMD*RM
      PS1=.5*(PO-PI)
```

```
C
C
      CALL SQUAD1(1,NR,RI,DD1,XR,YR1,ANS1)
C
      TH1=0.
      TH2=0.
C
      IF(IWRT.EQ.1) WRITE(11,102) NR,RI,ANS1,PSI,CPSI,SPSI,
     1  GAM,GAMD,PSID,Z1,EPS,DEL,EPSRI,DELRI,Z3R,PD1,PD2,
     2   PS1,XR,YR1
C
      DO 40 J4=1,N4PIE
      TH1=TH2
      TH2=TH2+PIEN4
      CALL SQUAD1(1,NTH,TH1,TH2,XTH,YTH1,ANS1)
C
      IF(IWRT.EQ.1) WRITE(11,104) J4,NTH,TH1,TH2,ANS1,XTH,YTH1
      DO 30 J3=1,NTH
      THA=XTH(J3)
      STHA= SIN(THA)
      CTHA= COS(THA)
C
      DO 20 J2=1,NR
      R=XR(J2)
      R2=R*R
      ECTHA=EPS*CTHA
      H=1.+R*ECTHA+DEL*(R-RI)
      HI=1.+EPSRI*CTHA
      HO=1.+ECTHA+DELRI
      HM=.5*(HI+HO)
      A=(1.-R)/(HM*H*H*RIM1)
C
      PD=Z3R*(PD1*STHA-CZD-PD2*CTHA)*A*(R-RI)
      PS=PO-PS1*HI*HI*(HO+H)*A
      P=PD+PS
      IF(P.LE.0) P=0.
      YR1(J2)=P*R2*CTHA
      YR2(J2)=P*R2*STHA
      YR3(J2)=P*R
      IF(IWRT.EQ.1) WRITE(11,106) R,R2,ECTHA,H,HI,
     1   HO,HM,A,PD,PS,P
   20 CONTINUE
C
      CALL SQUAD1(2,NR,RI,DD1,XR,YR1,ANS1)
      CALL SQUAD1(2,NR,RI,DD1,XR,YR2,ANS2)
      CALL SQUAD1(2,NR,RI,DD1,XR,YR3,ANS3)
C
      YTH1(J3)=ANS1
      YTH2(J3)=ANS2
      YTH3(J3)=ANS3
C
      IF(IWRT.EQ.1) WRITE(11,108) NTH,THA,STHA,CTHA,ANS1,
     1     ANS2,ANS3,YR1,YR2,YR3
   30 CONTINUE
C
      CALL SQUAD1(2,NTH,TH1,TH2,XTH,YTH1,ANS1)
      CALL SQUAD1(2,NTH,TH1,TH2,XTH,YTH2,ANS2)
      CALL SQUAD1(2,NTH,TH1,TH2,XTH,YTH3,ANS3)
      FM1=FM1+ANS1
      FM2=FM2+ANS2
      FZ1=FZ1+ANS3
      IF(IWRT.EQ.1) WRITE(11,110) TH1,TH2,ANS1,
     1    ANS2,ANS3,YTH1,YTH2,YTH3
   40 CONTINUE
C
      FM1C=.5*FKROC*RSP*RSP*GAM
      FM1=FM1-FM1C
```

```
      FMX=FM1*CPSI-FM2*SPSI
      FMY=FM1*SPSI+FM2*CPSI
C
      FC1=FKROC*CZ
      FC2=BETA*(PO-PI)*RIM1/(2.+BETA*RIM1)
      FC3=PO+PI+FC2
C
      FC=FC1+PIE*RM*RIM1*FC3
      FZ=FZ1-FC
C
      IF(IWRT.GE.1) WRITE(11,112) FMX,FMY,FZ,FC,FM1,
     1              FM2,PSI,PSID,FZ1,ANS1,ANS2,ANS3
C
C
C
  102 FORMAT(1H0,'  FQD2  NR'I2,'  RI='E12.5/
     1        4E14.5/(4E14.5))
  104 FORMAT(1H0,'FQD4  J4'I2,'  NTH'I2,3E15.5/' X'5F8.4/' Y'5F8.4)
C
  108 FORMAT(1H0,'FQD8  NTH',I3,'  THA='E12.5,'  STHA=',E12.5,
     1  '  CTHA=',E12.5,'  ANS1=',E12.5,'  ANS2=',E12.5,
     2  '  ANS3='E12.5,/,(4E15.5))
C
  110 FORMAT(1H0,'FQ10','  TH1='E12.5,'  TH2='E12.5,
     1 '  ANS1='E12.5,'  ANS2='E12.5,'  ANS3='E12.5,/(5E15.5))
C
  112 FORMAT(1H0, 'FMX',E12.5, 'FMY',E12.5,' FZ',E12.5,
     1  ' FC'E12.5,' FM1',E12.5,' FM2',E12.5,' PSI',E12.5,
     2  ' PSID',E12.5,/,' FZ1'E12.5,' ANS1'E12.5,
     3  ' ANS2'E12.5,' ANS3'E12.5)
C
  106 FORMAT('  FQD6','  R='E12.5,' R2=',E12.5,'  ECTHA='E12.5,
     1 '  H='E12.5,'  HI='E12.5/'  HO='E12.5,'  HM='E12.5,
     2 '  A='E12.5,'  PD='E12.5,'  PS='E12.5,'  P='E12.5)
C
  116 FORMAT(1H1  'CAX',E12.5, 'CAY'E12.5,  'CZ',E12.5,
     1        'CAXD',E12.5,'  CAYD',E12.5,'  CZD',E12.5,
     2  ' GAM',E11.4,' GAMD',E11.4)
C
C
      RETURN
      END
```

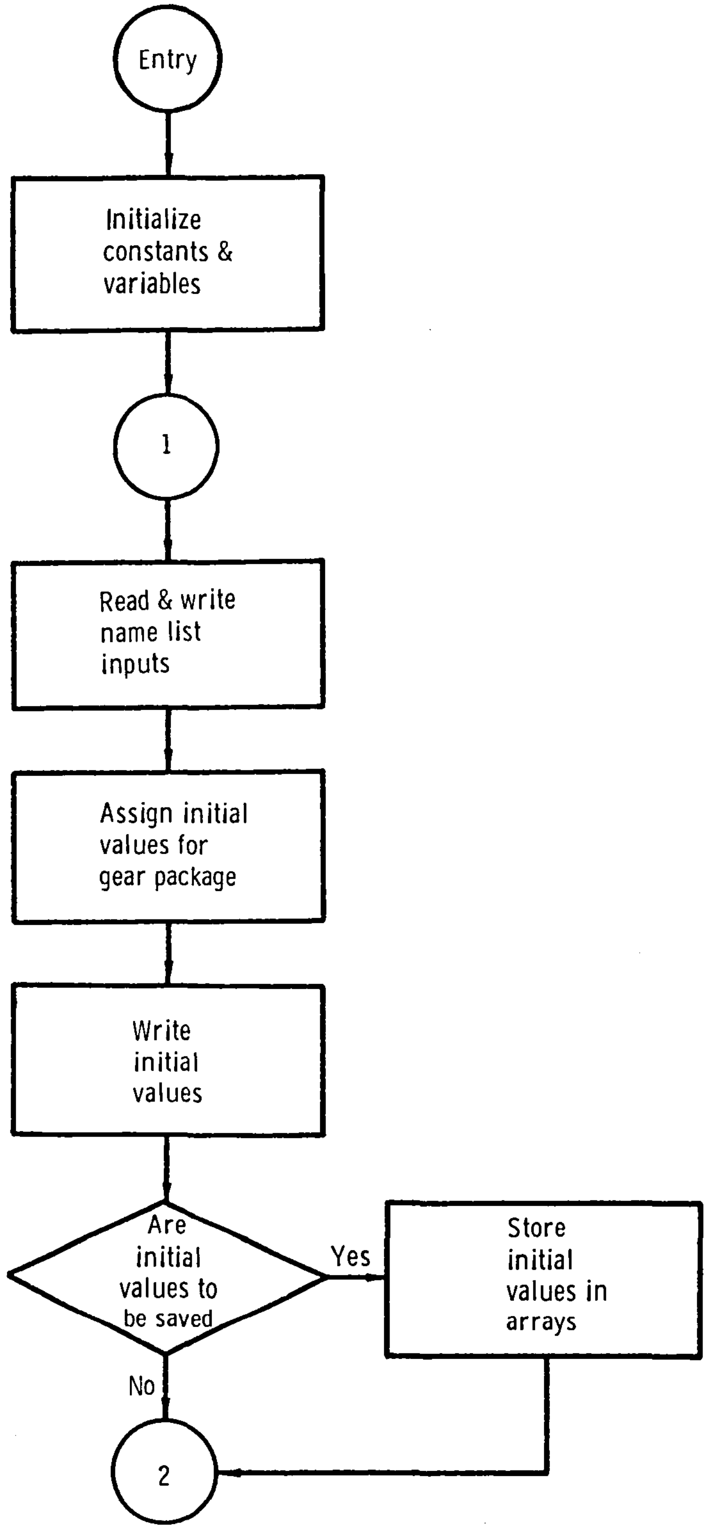
```
      SUBROUTINE SQUAD1(MODE,N,XI,XO,X,Y,ANSWR)
C                                                                       SQU00020
C  NUMERICAL INTEGRATION BY GAUSS  FORMULA                              SQU00030
C                                                                       SQU00040
      DIMENSION A(70),H(70),X(N),Y(N)                                   SQU00050
      DOUBLE PRECISION A,H
      DATA (A(I),H(I),I=1,28)/                                          SQU00060
     1 7.74596669241483D-01, 5.55555555555556D-01,-0.              D-00,SQU00070
     1 8.88888888888889D-01, 8.61136311594053D-01, 3.47854845137454D-01,SQU00080
     1 3.39981043584856D-01, 6.52145154862546D-01, 9.06179845938664D-01,SQU00090
     1 2.36926885056189D-01, 5.38469310105683D-01, 4.78628670499366D-01,SQU00100
     1-0.              D-00, 5.68888888888889D-01, 9.32469514203152D-01,SQU00110
     1 1.71324492379170D-01, 6.61209386466265D-01, 3.60761573048139D-01,SQU00120
     1 2.38619186083197D-01, 4.67913934572691D-01, 9.49107912342759D-01,SQU00130
     1 1.29484966168870D-01, 7.41531185599394D-01, 2.79705391489277D-01,SQU00140
     1 4.05845151377397D-01, 3.81830050505119D-01,-0.              D-00,SQU00150
     1 4.17959183673469D-01, 9.60289856497536D-01, 1.01228536290376D-01,SQU00160
     1 7.96666477413627D-01, 2.22381034453374D-01, 5.25532409916329D-01,SQU00170
     1 3.13706645877887D-01, 1.83434642495650D-01, 3.62683783378362D-01,SQU00180
     1 9.68160239507626D-01, 8.12743883615740D-02, 8.36031107326636D-01,SQU00190
     1 1.80648160694857D-01, 6.13371432700590D-01, 2.60610696402935D-01,SQU00200
     1 3.24253423403809D-01, 3.12347077040003D-01,-0.              D-00,SQU00210
     1 3.30239355001260D-01, 9.73906528517172D-01, 6.66713443086880D-02,SQU00220
     1 8.65063366688985D-01, 1.49451349150581D-01, 6.79409568299024D-01,SQU00230
     1 2.19086362515982D-01, 4.33395394129247D-01, 2.69266719309996D-01,SQU00240
     1 1.48874338981631D-01, 2.95524224714753D-01/                      SQU00250
      DATA (A(I),H(I),I=29,56)/                                         SQU00260
     1 9.78228658146057D-01, 5.56685671161740D-02, 8.87062599768095D-01,SQU00270
     1 1.25580369464905D-01, 7.30152005574049D-01, 1.86290210927734D-01,SQU00280
     1 5.19096129206812D-01, 2.33193764591990D-01, 2.69543155952345D-01,SQU00290
     1 2.62804544510247D-01,-0.              D-00, 2.72925086777901D-01,SQU00300
     1 9.81560634246719D-01, 4.71753363865120D-02, 9.04117256370475D-01,SQU00310
     1 1.06939325995318D-01, 7.69902674194305D-01, 1.60078328543346D-01,SQU00320
     1 5.87317954286617D-01, 2.03167426723066D-01, 3.67831498998180D-01,SQU00330
     1 2.33492536538355D-01, 1.25233408511469D-01, 2.49147045813403D-01,SQU00340
     1 9.84183054718588D-01, 4.04840047653160D-02, 9.17598399222978D-01,SQU00350
     1 9.21214998377280D-02, 8.01578090733310D-01, 1.38873510219787D-01,SQU00360
     1 6.42349339440340D-01, 1.78145980761946D-01, 4.48492751036447D-01,SQU00370
     1 2.07816047536889D-01, 2.30458315955135D-01, 2.26283180262897D-01,SQU00380
     1-0.              D-00, 2.32551553230874D-01, 9.86283808696812D-01,SQU00390
     1 3.51194603317520D-02, 9.28434883663574D-01, 8.01580871597600D-02,SQU00400
     1 8.27201315069765D-01, 1.21518570687903D-01, 6.87292904811685D-01,SQU00410
     1 1.57203167158194D-01, 5.15248636358154D-01, 1.85538397477938D-01,SQU00420
     1 3.19112368927890D-01, 2.05198463721296D-01, 1.08054948707344D-01,SQU00430
     1 2.15263853463158D-01, 9.87992518020485D-01, 3.07532419961170D-02,SQU00440
     1 9.37273392400706D-01, 7.03660474881080D-02/                      SQU00450
      DATA (A(I),H(I),I=57,70)/                                         SQU00460
     1 8.48206583410427D-01, 1.07159220467172D-01, 7.24417731360170D-01,SQU00470
     1 1.39570677926154D-01, 5.70972172608539D-01, 1.66269205816994D-01,SQU00480
     1 3.94151347077563D-01, 1.86161000015562D-01, 2.01194093997435D-01,SQU00490
     1 1.98431485327112D-01,-0.              D-00, 2.02578241925561D-01,SQU00500
     1 9.89400934991650D-01, 2.71524594117540D-02, 9.44575023073233D-01,SQU00510
     1 6.22535239386480D-02, 8.65631202387832D-01, 9.51585116824930D-02,SQU00520
     1 7.55404408355003D-01, 1.24628971255534D-01, 6.17876244402644D-01,SQU00530
     1 1.49595988816577D-01, 4.58016777657227D-01, 1.69156519395003D-01,SQU00540
     1 2.81603550779259D-01, 1.82603415044924D-01, 9.50125098376370D-02,SQU00550
     1 1.89450610455068D-01/                                            SQU00560
C
C
      NN=N
      XIN= XI
      ANS=0.
      IF(XIN .EQ.XO) GO TO (400,500),MODE
      IF (NN.LE. 16) GO TO 202
      WRITE (6,200) NN
  200 FORMAT(47HO*****NUMBER OF POINTS FOR SQUAD1 INTEGRATION =I4,22H IS
     A OUT OF RANGE.*****)
```

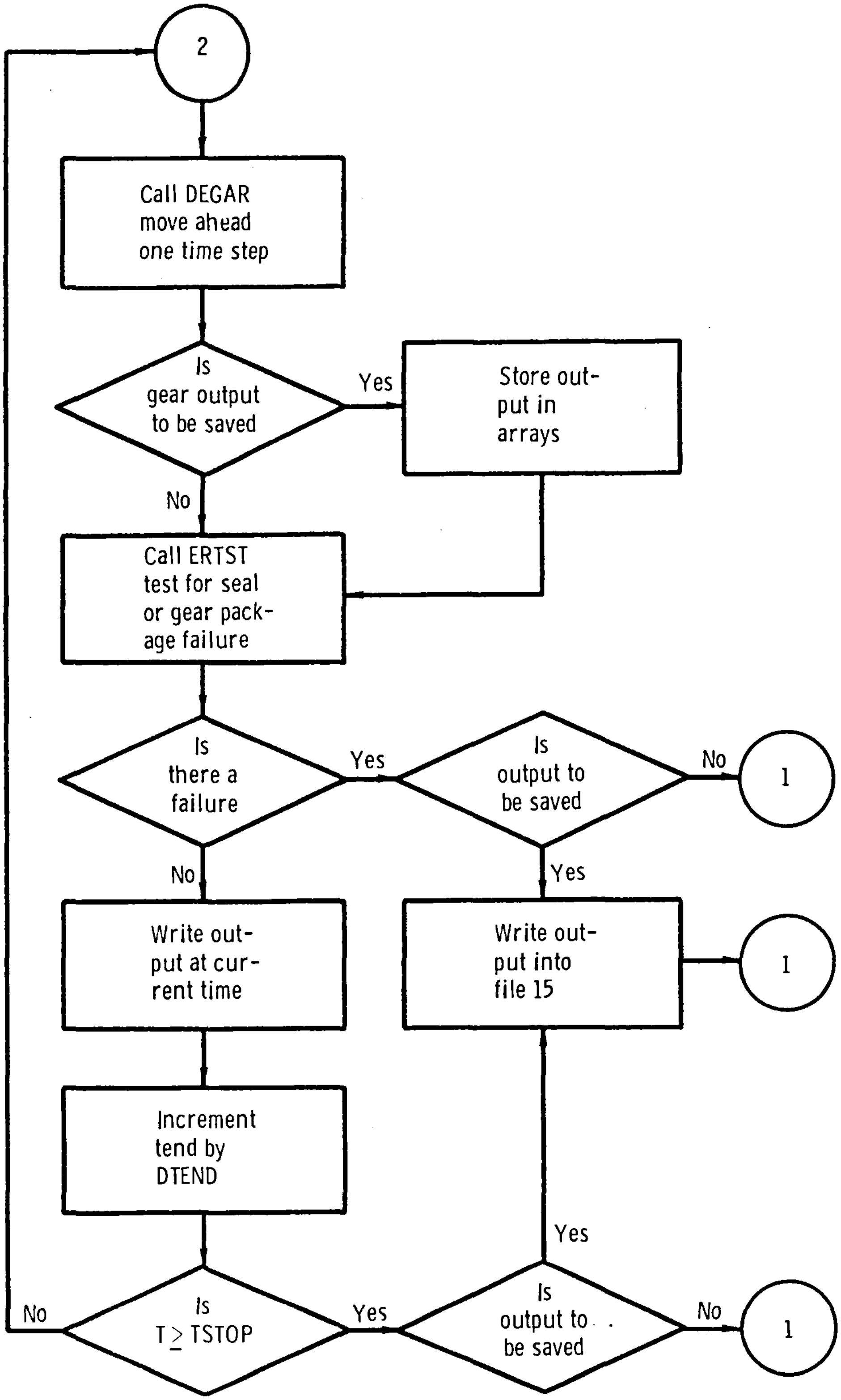
```
      RETURN
  202 INDKT = MOD(NN,2)+1
COMMENT--INDKT=1, N IS EVEN
C              =2, N IS ODD
      GO TO (204,210),INDKT
  204 MIN= (NN*NN)/4 - 1
      MAX = (NN*(NN+2))/4 - 2
      GO TO 215
  210 MIN= (NN*NN-9)/4 + 1
      MAX = (NN*(NN+2)-11)/4
  215 MINM1 = MIN-1
      HDELX= 0.5*(XO-XIN)
      M= NN/2
      DO 220 I=1,M
      IM= I+M
      J= MINM1+I
      GO TO (216,217),MODE
  216 X(I)= XIN + HDELX*(1.-A(J))
      X(IM)= XIN + HDELX*(1.+A(J))
      GO TO 220
  217 ANS= ANS + (Y(I)+Y(IM))*H(J)
  220 CONTINUE
      IF(INDKT .EQ. 2) GO TO (230,240),MODE
      GO TO (501,500),MODE
  230 X(NN)= XIN+HDELX
      RETURN
  240 ANS = ANS + Y(NN)*H(MAX+1)
  500 ANSWR= HDELX*ANS
  501 RETURN
  400 DO 402 I=1,NN
  402 X(I)= XIN
  999 RETURN
      END
```

```
      SUBROUTINE WRT(N6,JT,Y,T,DELH,TEND,INDEX,IER)
C
C
C                 THIS SUB WRITES OUTPUT FOR ETSION'S PROGRAM
C
C
      DIMENSION Y(N6)
C
      COMMON/COM1/ PIE,PIE2,N4PIE,NR,NTH,IWRT
C
      COMMON/COM2/  FKROC,RO,BETA,RI,CO,
     1              RM,PO,PI,FI,FME,FM,RSP
C
      COMMON/COM3/ HMIN,HMIN1,HMIN2
      COMMON/COM4/ CAX,CAY,CZ,CAXD,CAYD,CZD
      COMMON/COM5/ FMX,FMY,FZ,GAM,PSI,GAMD,PSID,FC,
     1              FM1,FM1C,FZ1,FM2,EPS,DEL
C
C
      TWRT=TEND
      IF(JT.LE.1) TWRT=0.
C
      CAX=Y(1)
      CAY=Y(2)
      CZ=Y(3)
      CAXD=Y(4)
      CAYD=Y(5)
      CZD=Y(6)
      CALL FQUAD
      YPRSV1=(FMX+FME)/FI
      YPRSV2=FMY/FI
      YPRSV3=FZ/FM
C
      WRITE(7,102) INDEX,IER,Y(4),Y(5),Y(6),T,DELH,YPRSV1,YPRSV2,YPRSV3,
     1  FC
C
C
C
      SQXY=CAX*CAX + CAY*CAY
      GAM= SQRT(SQXY)
C
      CAXX= ABS(CAX)
      CAYY= ABS(CAY)
      PSI= ATAN2(CAYY,CAXX)
      IF(CAX.LT.0.AND.CAY.GT.0) PSI=-PSI
      IF(CAX.GE.0.AND.CAY.LT.0) PSI=-PSI
C
      CPSI=CAX/GAM
      SPSI=CAY/GAM
C
      GAMD=CAXD*CPSI+CAYD*SPSI
      PSID=(CAYD*CPSI-CAXD*SPSI)/GAM
C
C
      TWRTP=TWRT/PIE2
      WRITE(6,104) CZ,GAM,PSI,CZD,GAMD,PSID,CAX,
     1              CAY,HMIN,TWRTP
C
C
  104 FORMAT(1H ,10E11.4)
  102 FORMAT(2I5,9E11.4)
C
   90 RETURN
      END
```

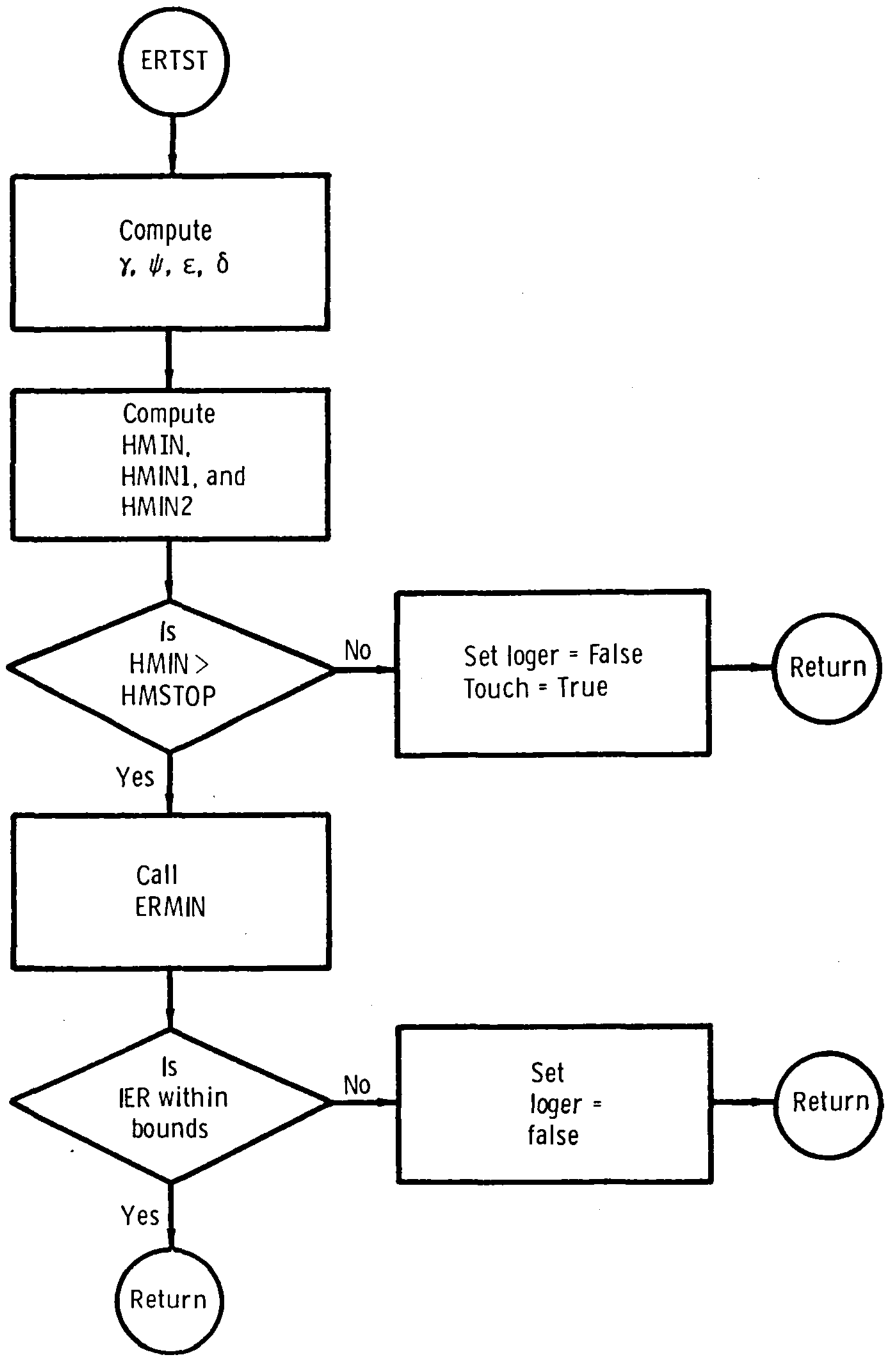
APPENDIX D - FLOW CHARTS

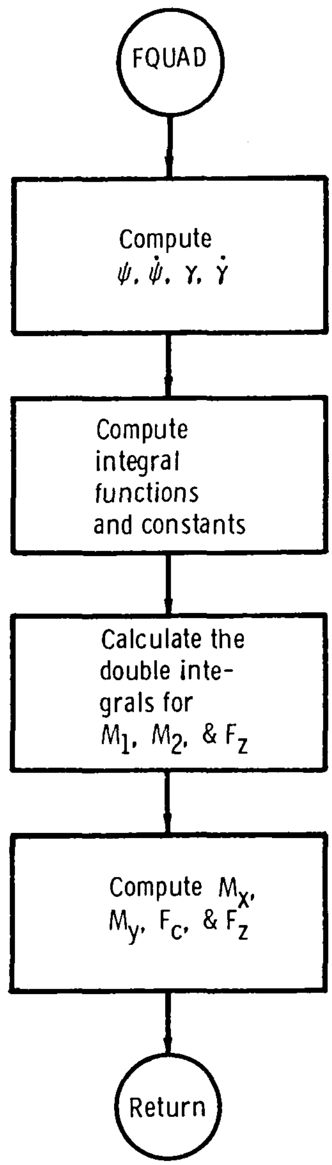
MAIN EXECUTIVE PROGRAM

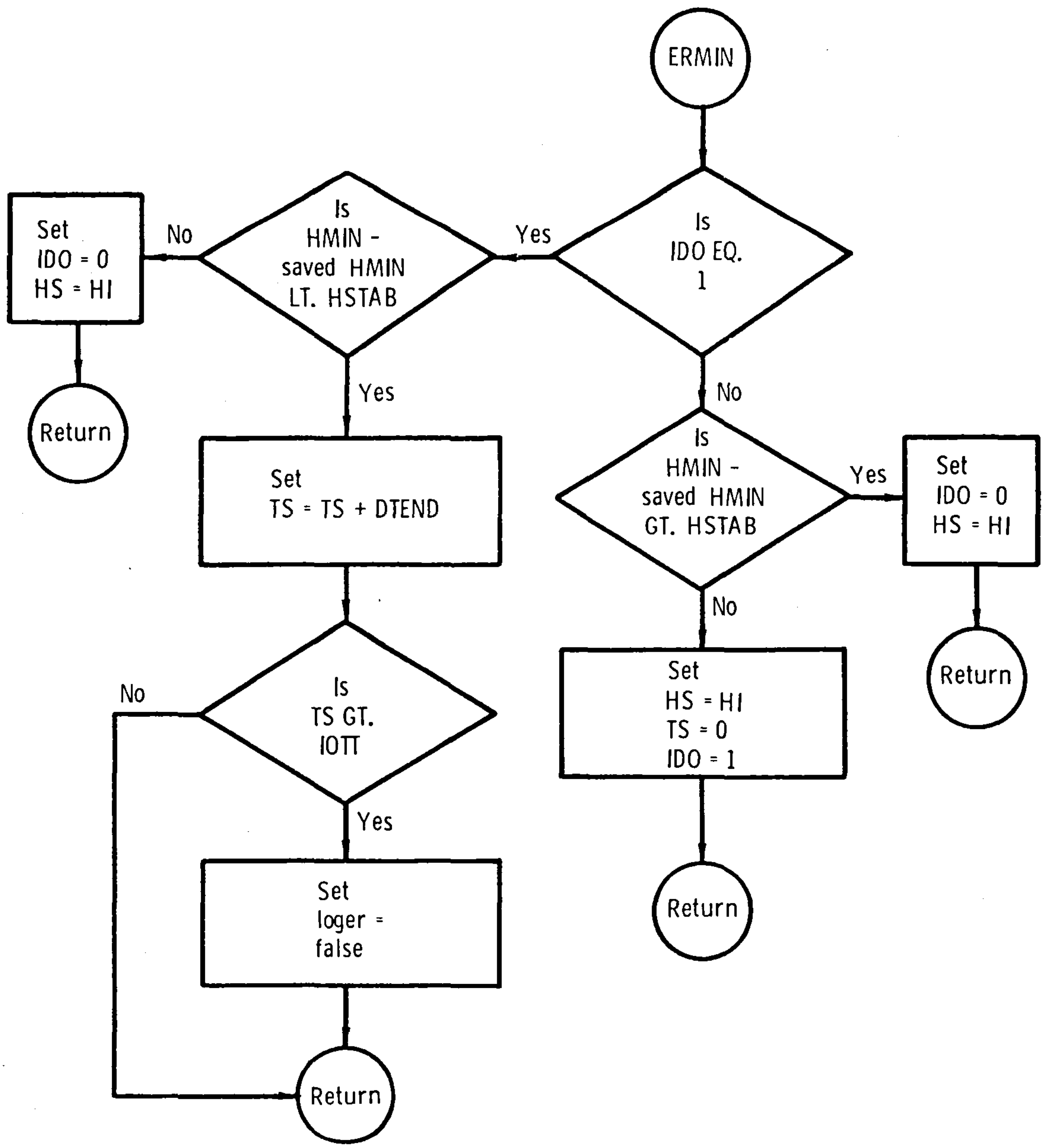




2
Call DEGAR move ahead one time step
Is gear output to be saved
Yes
Store output in arrays
No
Call ERTST test for seal or gear package failure
Is there a failure
Yes
Is output to be saved
No
1
No
Yes
Write output at current time
Write output into file 15
1
Increment tend by DTEND
Yes
No
Is T ≥ TSTOP
Yes
Is output to be saved
No
1


ERTST
Compute
γ, ψ, ε, δ
Compute
HMIN,
HMIN1, and
HMIN2
Is
HMIN >
HMSTOP
No
Set loger = False
Touch = True
Return
Yes
Call
ERMIN
Is
IER within
bounds
No
Set
loger =
false
Return
Yes
Return


FQUAD
Compute
ψ, $\dot{\psi}$, γ, $\dot{\gamma}$
Compute
integral
functions
and constants
Calculate the
double inte-
grals for
M_1, M_2, & F_z
Compute M_x,
M_y, F_c, & F_z
Return


ERMIN
Is
IDO EQ.
1
Yes
Is
HMIN -
saved HMIN
LT. HSTAB
No
Set
IDO = 0
HS = HI
Return
Yes
Set
TS = TS + DTEND
Is
TS GT.
IOTT
No
Yes
Set
loger =
false
Return
No
Is
HMIN -
saved HMIN
GT. HSTAB
Yes
Set
IDO = 0
HS = HI
Return
No
Set
HS = HI
TS = 0
IDO = 1
Return

APPENDIX E

SAMPLE PROBLEM

A computer output listing is shown as an example of a representative run. The computer run generated output for twenty revolutions but only output from the first, second, nineteenth, and the twentieth revolutions is shown. Input data of the two namelists, INPUT1 and INPUT2 is shown preceeding the listed output and this input is defined in the report section, USING THE PROGRAM.

In this case, the Adams predictor corrector method was used (METH = 1), an initial step size of 0.001 was tried (DELH = 0.001), functional iteration was used (MITER = 0), and the input relative error bound, TOL, was set to 0.0001. The initial values T = 0 (TZRO = 0) and $\alpha_x = 0$, $\alpha_y = 0.1$, $Z = 0$, $\dot{\alpha}_x = 0$, $\dot{\alpha}_y = 0$, and $\dot{Z} = 0$ (YIN = 0, 0.1,0,0,0,0) were set. TSTOP was set equal to 125.66 radians, or twenty revolutions of the rotor. Also, the inputs $R_i = 0.9$, $K \frac{C_o}{r_o} = 0.0025$, $\beta = 10$, $P_i = 0$, $P_o = 0.25$, $M_e = 0$, $m = 0.08$, $r_g = 1$, and RSP = 1 were used.

The execution of this case required two minutes of computer time on the Univac 1100/42.

```
$INPUT1
READ1   =  F
METH    =              +1
MITER   =              +0
TOL     =      .10000000E-03
INDEX   =              +1
DELH    =      .10000000E-02
TZRO    =      .00000000E+00
YIN     =      .00000000E+00,    .10000000E+00,    .00000000E+00,    .00000000E+00,
               .00000000E+00,    .00000000E+00
TSTOP   =      .12566371E+03
DTEND   =      .12566371E+00
NR      =              +5
NTH     =              +8
HMSTOP  =      .10000000E-20
HSTAB   =      .10000000E-02
NPLT    =              +1
PUNCH   =  F

$END
$INPUT2
RI      =      .90000000E+00
FKROC   =      .25000000E-02
BETA    =      .10000000E+02
PI      =      .00000000E+00
PO      =      .25000000E+00
FME     =      .00000000E+00
FM      =      .80000000E-01
RG      =      .10000000E+01
RSP     =      .10000000E+01

$END
```

CZ	GAM	PSI	CZD	GAMD	PSID	CAX	CAY	HMIN	TWRT
.0000	.1000+00	.1571+01	.0000	.0000	.0000	.0000	.1000+00	.9130+00	.0000
.1153-03	.9986-01	-.1569+01	.1881-02	-.2279-02	.2177-01	-.1335-03	.9986-01	.9101+00	.2000-01
.4605-03	.9944-01	-.1565+01	.3654-02	-.4422-02	.4282-01	-.5332-03	.9944-01	.9105+00	.4000-01
.1021-02	.9876-01	-.1559+01	.5318-02	-.6421-02	.6338-01	-.1183-02	.9876-01	.9112+00	.6000-01
.1784-02	.9784-01	-.1550+01	.6873-02	-.8268-02	.8367-01	-.2072-02	.9782-01	.9121+00	.8000-01
.2735-02	.9670-01	-.1538+01	.8320-02	-.9957-02	.1038+00	-.3180-02	.9665-01	.9132+00	.1000+00
.3871-02	.9535-01	-.1524+01	.9658-02	-.1148-01	.1241+00	-.4508-02	.9524-01	.9145+00	.1200+00
.5150-02	.9383-01	-.1507+01	.1089-01	-.1284-01	.1445+00	-.5999-02	.9363-01	.9160+00	.1400+00
.6590-02	.9214-01	-.1487+01	.1202-01	-.1403-01	.1653+00	-.7676-02	.9182-01	.9176+00	.1600+00
.8171-02	.9030-01	-.1465+01	.1305-01	-.1504-01	.1866+00	-.9516-02	.8980-01	.9194+00	.1800+00
.9878-02	.8835-01	-.1440+01	.1399-01	-.1587-01	.2085+00	-.1149-01	.8760-01	.9213+00	.2000+00
.1169-01	.8631-01	-.1413+01	.1482-01	-.1653-01	.2310+00	-.1359-01	.8523-01	.9232+00	.2200+00
.1361-01	.8420-01	-.1382+01	.1556-01	-.1701-01	.2541+00	-.1579-01	.8271-01	.9252+00	.2400+00
.1560-01	.8204-01	-.1349+01	.1622-01	-.1732-01	.2780+00	-.1807-01	.8003-01	.9273+00	.2600+00
.1768-01	.7985-01	-.1312+01	.1678-01	-.1745-01	.3027+00	-.2042-01	.7720-01	.9294+00	.2800+00
.1982-01	.7766-01	-.1273+01	.1727-01	-.1739-01	.3282+00	-.2282-01	.7423-01	.9315+00	.3000+00
.2202-01	.7549-01	-.1230+01	.1767-01	-.1716-01	.3543+00	-.2525-01	.7114-01	.9335+00	.3200+00
.2426-01	.7336-01	-.1183+01	.1800-01	-.1675-01	.3810+00	-.2771-01	.6792-01	.9355+00	.3400+00
.2654-01	.7129-01	-.1134+01	.1825-01	-.1616-01	.4080+00	-.3017-01	.6459-01	.9375+00	.3600+00
.2885-01	.6930-01	-.1081+01	.1843-01	-.1541-01	.4351+00	-.3261-01	.6115-01	.9394+00	.3800+00
.3117-01	.6742-01	-.1024+01	.1855-01	-.1449-01	.4619+00	-.3503-01	.5761-01	.9412+00	.4000+00
.3351-01	.6567-01	-.9648+00	.1860-01	-.1341-01	.4880+00	-.3741-01	.5398-01	.9428+00	.4200+00
.3585-01	.6406-01	-.9019+00	.1860-01	-.1219-01	.5128+00	-.3973-01	.5025-01	.9443+00	.4400+00
.3819-01	.6261-01	-.8359+00	.1853-01	-.1084-01	.5358+00	-.4198-01	.4645-01	.9457+00	.4600+00
.4051-01	.6133-01	-.7673+00	.1841-01	-.9382-02	.5564+00	-.4415-01	.4257-01	.9470+00	.4800+00
.4281-01	.6024-01	-.6962+00	.1824-01	-.7845-02	.5739+00	-.4623-01	.3863-01	.9480+00	.5000+00
.4509-01	.5936-01	-.6231+00	.1802-01	-.6254-02	.5878+00	-.4820-01	.3464-01	.9489+00	.5200+00
.4734-01	.5867-01	-.5485+00	.1776-01	-.4641-02	.5978+00	-.5006-01	.3059-01	.9496+00	.5400+00
.4956-01	.5819-01	-.4730+00	.1745-01	-.3037-02	.6037+00	-.5180-01	.2651-01	.9501+00	.5600+00
.5173-01	.5791-01	-.3969+00	.1710-01	-.1473-02	.6053+00	-.5341-01	.2239-01	.9504+00	.5800+00
.5386-01	.5783-01	-.3209+00	.1671-01	.2301-04	.6029+00	-.5487-01	.1824-01	.9506+00	.6000+00
.5594-01	.5793-01	-.2455+00	.1629-01	.1424-02	.5969+00	-.5619-01	.1408-01	.9506+00	.6200+00
.5797-01	.5820-01	-.1710+00	.1584-01	.2710-02	.5879+00	-.5735-01	.9904-02	.9505+00	.6400+00
.5993-01	.5862-01	-.9784-01	.1536-01	.3866-02	.5763+00	-.5834-01	.5726-02	.9502+00	.6600+00
.6183-01	.5917-01	-.2627-01	.1485-01	.4882-02	.5629+00	-.5915-01	.1555-02	.9498+00	.6800+00
.6367-01	.5985-01	.4350-01	.1432-01	.5755-02	.5482+00	-.5979-01	-.2603-02	.9494+00	.7000+00
.6543-01	.6063-01	.1114+00	.1376-01	.6485-02	.5327+00	-.6026-01	-.6737-02	.9488+00	.7200+00
.6713-01	.6150-01	.1772+00	.1319-01	.7074-02	.5170+00	-.6053-01	-.1084-01	.9481+00	.7400+00
.6876-01	.6242-01	.2411+00	.1260-01	.7530-02	.5015+00	-.6062-01	-.1491-01	.9474+00	.7600+00
.7030-01	.6339-01	.3031+00	.1199-01	.7863-02	.4865+00	-.6050-01	-.1892-01	.9467+00	.7800+00
.7177-01	.6440-01	.3633+00	.1137-01	.8080-02	.4723+00	-.6020-01	-.2289-01	.9459+00	.8000+00
.7317-01	.6543-01	.4217+00	.1074-01	.8192-02	.4589+00	-.5970-01	-.2678-01	.9451+00	.8200+00
.7448-01	.6647-01	.4785+00	.1011-01	.8209-02	.4465+00	-.5901-01	-.3061-01	.9443+00	.8400+00
.7572-01	.6751-01	.5338+00	.9467-02	.8144-02	.4352+00	-.5812-01	-.3435-01	.9435+00	.8600+00
.7687-01	.6853-01	.5877+00	.8822-02	.8006-02	.4251+00	-.5703-01	-.3800-01	.9427+00	.8800+00
.7794-01	.6953-01	.6405+00	.8176-02	.7806-02	.4160+00	-.5575-01	-.4155-01	.9420+00	.9000+00
.7893-01	.7050-01	.6922+00	.7530-02	.7553-02	.4080+00	-.5427-01	-.4499-01	.9412+00	.9200+00
.7984-01	.7143-01	.7429+00	.6885-02	.7254-02	.4010+00	-.5261-01	-.4832-01	.9405+00	.9400+00
.8067-01	.7233-01	.7928+00	.6245-02	.6918-02	.3950+00	-.5076-01	-.5152-01	.9398+00	.9600+00
.8142-01	.7318-01	.8420+00	.5610-02	.6553-02	.3899+00	-.4874-01	-.5460-01	.9391+00	.9800+00
.8208-01	.7359-01	.8907+00	.4983-02	.6166-02	.3857+00	-.4653-01	-.5753-01	.9385+00	.1000+01
.8267-01	.7474-01	.9389+00	.4365-02	.5764-02	.3823+00	-.4415-01	-.6031-01	.9379+00	.1020+01
.8318-01	.7544-01	.9867+00	.3757-02	.5351-02	.3797+00	-.4160-01	-.6293-01	.9373+00	.1040+01
.8362-01	.7609-01	.1034+01	.3161-02	.4934-02	.3778+00	-.3890-01	-.6540-01	.9368+00	.1060+01
.8399-01	.7669-01	.1082+01	.2578-02	.4516-02	.3766+00	-.3604-01	-.6770-01	.9363+00	.1080+01
.8428-01	.7724-01	.1129+01	.2010-02	.4103-02	.3759+00	-.3305-01	-.6982-01	.9359+00	.1100+01

.8450-01	.7774-01	.1176+01	.1458-02	.3699-02	.3758+00	-.2991-01	-.7175-01	.9355+00	.1120+01
.8465-01	.7818-01	.1223+01	.9240-03	.3306-02	.3762+00	-.2664-01	-.7350-01	.9351+00	.1140+01
.8473-01	.7857-01	.1270+01	.4074-03	.2929-02	.3771+00	-.2325-01	-.7505-01	.9348+00	.1160+01
.8475-01	.7892-01	.1318+01	-.9053-04	.2569-02	.3783+00	-.1976-01	-.7641-01	.9345+00	.1180+01
.8471-01	.7923-01	.1365+01	-.5687-03	.2230-02	.3800+00	-.1617-01	-.7757-01	.9343+00	.1200+01
.8461-01	.7950-01	.1413+01	-.1026-02	.1913-02	.3820+00	-.1248-01	-.7851-01	.9340+00	.1220+01
.8446-01	.7972-01	.1461+01	-.1462-02	.1620-02	.3842+00	-.8720-02	-.7925-01	.9338+00	.1240+01
.8425-01	.7991-01	.1510+01	-.1874-02	.1352-02	.3867+00	-.4887-02	-.7976-01	.9337+00	.1260+01
.8399-01	.8007-01	.1558+01	-.2265-02	.1111-02	.3894+00	-.9984-03	-.8007-01	.9335+00	.1280+01
.8368-01	.8020-01	-.1534+01	-.2633-02	.8979-03	.3923+00	.2934-02	-.8015-01	.9334+00	.1300+01
.8333-01	.8031-01	-.1485+01	-.2978-02	.7121-03	.3953+00	.6898-02	-.8002-01	.9333+00	.1320+01
.8294-01	.8040-01	-.1435+01	-.3298-02	.5543-03	.3984+00	.1088-01	-.7966-01	.9332+00	.1340+01
.8250-01	.8046-01	-.1385+01	-.3595-02	.4242-03	.4015+00	.1488-01	-.7908-01	.9331+00	.1360+01
.8204-01	.8051-01	-.1334+01	-.3866-02	.3213-03	.4047+00	.1887-01	-.7827-01	.9330+00	.1380+01
.8155-01	.8055-01	-.1283+01	-.4114-02	.2453-03	.4078+00	.2285-01	-.7724-01	.9330+00	.1400+01
.8100-01	.8058-01	-.1232+01	-.4338-02	.1954-03	.4110+00	.2680-01	-.7599-01	.9329+00	.1420+01
.8044-01	.8061-01	-.1180+01	-.4538-02	.1702-03	.4140+00	.3071-01	-.7453-01	.9329+00	.1440+01
.7986-01	.8063-01	-.1128+01	-.4714-02	.1684-03	.4170+00	.3457-01	-.7284-01	.9328+00	.1460+01
.7926-01	.8066-01	-.1075+01	-.4867-02	.1885-03	.4199+00	.3836-01	-.7095-01	.9327+00	.1480+01
.7864-01	.8069-01	-.1022+01	-.4997-02	.2292-03	.4226+00	.4208-01	-.6885-01	.9327+00	.1500+01
.7800-01	.8072-01	-.9690+00	-.5104-02	.2886-03	.4251+00	.4570-01	-.6654-01	.9326+00	.1520+01
.7736-01	.8077-01	-.9154+00	-.5189-02	.3647-03	.4275+00	.4922-01	-.6403-01	.9325+00	.1540+01
.7670-01	.8082-01	-.8616+00	-.5252-02	.4552-03	.4296+00	.5263-01	-.6133-01	.9324+00	.1560+01
.7604-01	.8089-01	-.8075+00	-.5294-02	.5585-03	.4316+00	.5592-01	-.5844-01	.9323+00	.1580+01
.7537-01	.8097-01	-.7532+00	-.5316-02	.6725-03	.4333+00	.5907-01	-.5538-01	.9322+00	.1600+01
.7470-01	.8106-01	-.6986+00	-.5319-02	.7951-03	.4349+00	.6207-01	-.5214-01	.9321+00	.1620+01
.7403-01	.8118-01	-.6439+00	-.5303-02	.9240-03	.4362+00	.6492-01	-.4873-01	.9320+00	.1640+01
.7337-01	.8130-01	-.5890+00	-.5268-02	.1057-02	.4372+00	.6760-01	-.4517-01	.9318+00	.1660+01
.7271-01	.8145-01	-.5341+00	-.5216-02	.1193-02	.4381+00	.7010-01	-.4146-01	.9317+00	.1680+01
.7205-01	.8161-01	-.4790+00	-.5148-02	.1330-02	.4387+00	.7243-01	-.3761-01	.9315+00	.1700+01
.7141-01	.8179-01	-.4238+00	-.5065-02	.1466-02	.4392+00	.7455-01	-.3364-01	.9313+00	.1720+01
.7078-01	.8199-01	-.3687+00	-.4967-02	.1599-02	.4395+00	.7648-01	-.2954-01	.9311+00	.1740+01
.7016-01	.8220-01	-.3135+00	-.4855-02	.1729-02	.4395+00	.7820-01	-.2535-01	.9309+00	.1760+01
.6956-01	.8243-01	-.2582+00	-.4731-02	.1853-02	.4395+00	.7969-01	-.2105-01	.9306+00	.1780+01
.6897-01	.8267-01	-.2030+00	-.4594-02	.1972-02	.4392+00	.8097-01	-.1667-01	.9304+00	.1800+01
.6840-01	.8293-01	-.1479+00	-.4448-02	.2083-02	.4389+00	.8202-01	-.1222-01	.9301+00	.1820+01
.6785-01	.8320-01	-.9278-01	-.4291-02	.2187-02	.4384+00	.8284-01	-.7708-02	.9299+00	.1840+01
.6732-01	.8348-01	-.3772-01	-.4126-02	.2282-02	.4379+00	.8342-01	-.3148-02	.9296+00	.1860+01
.6682-01	.8378-01	.1728-01	-.3952-02	.2369-02	.4372+00	.8376-01	.1448-02	.9293+00	.1880+01
.6633-01	.8408-01	.7220-01	-.3772-02	.2446-02	.4365+00	.8386-01	.6066-02	.9290+00	.1900+01
.6587-01	.8439-01	.1270+00	-.3586-02	.2515-02	.4358+00	.8371-01	.1069-01	.9287+00	.1920+01
.6543-01	.8471-01	.1818+00	-.3395-02	.2575-02	.4350+00	.8332-01	.1531-01	.9284+00	.1940+01
.6502-01	.8504-01	.2364+00	-.3199-02	.2626-02	.4342+00	.8267-01	.1992-01	.9281+00	.1960+01
.6463-01	.8537-01	.2909+00	-.3001-02	.2669-02	.4334+00	.8179-01	.2449-01	.9278+00	.1980+01
.6426-01	.8571-01	.3453+00	-.2800-02	.2704-02	.4325+00	.8065-01	.2901-01	.9275+00	.2000+01

.5320+00	.1490+01	-.8256+00	.3100-01	.5260-01	.4265+00	.1011+01	-.1095+01	.1245+00	.1800+02
.5360+00	.1497+01	-.7720+00	.3130-01	.5294-01	.4265+00	.1073+01	-.1044+01	.1229+00	.1802+02
.5399+00	.1504+01	-.7184+00	.3162-01	.5326-01	.4265+00	.1132+01	-.9897+00	.1212+00	.1804+02
.5439+00	.1510+01	-.6649+00	.3194-01	.5359-01	.4265+00	.1189+01	-.9318+00	.1196+00	.1806+02
.5480+00	.1517+01	-.6113+00	.3226-01	.5394-01	.4265+00	.1242+01	-.8707+00	.1180+00	.1808+02
.5521+00	.1524+01	-.5577+00	.3257-01	.5431-01	.4265+00	.1293+01	-.8064+00	.1163+00	.1810+02
.5562+00	.1531+01	-.5041+00	.3290-01	.5465-01	.4265+00	.1340+01	-.7393+00	.1147+00	.1812+02
.5603+00	.1538+01	-.4505+00	.3325-01	.5496-01	.4265+00	.1384+01	-.6695+00	.1131+00	.1814+02
.5645+00	.1545+01	-.3969+00	.3363-01	.5524-01	.4265+00	.1425+01	-.5971+00	.1115+00	.1816+02
.5688+00	.1552+01	-.3433+00	.3397-01	.5560-01	.4265+00	.1461+01	-.5222+00	.1099+00	.1818+02
.5731+00	.1559+01	-.2897+00	.3433-01	.5594-01	.4265+00	.1494+01	-.4452+00	.1083+00	.1820+02
.5774+00	.1566+01	-.2361+00	.3465-01	.5634-01	.4265+00	.1522+01	-.3662+00	.1067+00	.1822+02
.5818+00	.1573+01	-.1825+00	.3502-01	.5669-01	.4266+00	.1547+01	-.2854+00	.1052+00	.1824+02
.5862+00	.1580+01	-.1289+00	.3535-01	.5712-01	.4266+00	.1567+01	-.2031+00	.1036+00	.1826+02
.5907+00	.1587+01	-.7535-01	.3569-01	.5751-01	.4266+00	.1583+01	-.1195+00	.1021+00	.1828+02
.5952+00	.1594+01	-.2176-01	.3606-01	.5789-01	.4266+00	.1594+01	-.3469-01	.1005+00	.1830+02
.5998+00	.1602+01	.3182-01	.3644-01	.5827-01	.4266+00	.1601+01	.5095-01	.9900-01	.1832+02
.6044+00	.1609+01	.8540-01	.3682-01	.5865-01	.4266+00	.1603+01	.1372+00	.9747-01	.1834+02
.6090+00	.1616+01	.1390+00	.3720-01	.5905-01	.4266+00	.1601+01	.2239+00	.9594-01	.1836+02
.6137+00	.1624+01	.1926+00	.3758-01	.5946-01	.4266+00	.1594+01	.3109+00	.9440-01	.1838+02
.6185+00	.1631+01	.2463+00	.3796-01	.5988-01	.4266+00	.1582+01	.3977+00	.9287-01	.1840+02
.6235+00	.1639+01	.2999+00	.3835-01	.6031-01	.4266+00	.1566+01	.4841+00	.9137-01	.1842+02
.6281+00	.1646+01	.3535+00	.3874-01	.6073-01	.4267+00	.1545+01	.5700+00	.8986-01	.1844+02
.6330+00	.1654+01	.4071+00	.3915-01	.6114-01	.4267+00	.1519+01	.6550+00	.8836-01	.1846+02
.6379+00	.1662+01	.4607+00	.3958-01	.6153-01	.4267+00	.1489+01	.7388+00	.8687-01	.1848+02
.6429+00	.1670+01	.5143+00	.4002-01	.6192-01	.4267+00	.1454+01	.8214+00	.8538-01	.1850+02
.6480+00	.1677+01	.5679+00	.4047-01	.6230-01	.4267+00	.1414+01	.9323+00	.8390-01	.1852+02
.6531+00	.1685+01	.6216+00	.4093-01	.6269-01	.4267+00	.1370+01	.9814+00	.8242-01	.1854+02
.6582+00	.1693+01	.6752+00	.4139-01	.6309-01	.4267+00	.1322+01	.1058+01	.8096-01	.1856+02
.6634+00	.1701+01	.7288+00	.4184-01	.6352-01	.4267+00	.1269+01	.1133+01	.7950-01	.1858+02
.6687+00	.1709+01	.7824+00	.4228-01	.6399-01	.4267+00	.1212+01	.1205+01	.7807-01	.1860+02
.6740+00	.1717+01	.8360+00	.4271-01	.6449-01	.4268+00	.1151+01	.1274+01	.7665-01	.1862+02
.6794+00	.1726+01	.8897+00	.4313-01	.6502-01	.4268+00	.1087+01	.1341+01	.7524-01	.1864+02
.6849+00	.1734+01	.9433+00	.4356-01	.6557-01	.4268+00	.1018+01	.1404+01	.7385-01	.1866+02
.6904+00	.1742+01	.9969+00	.4402-01	.6607-01	.4268+00	.9458+00	.1463+01	.7246-01	.1868+02
.6960+00	.1750+01	.1051+01	.4447-01	.6660-01	.4268+00	.8701+00	.1519+01	.7108-01	.1870+02
.7016+00	.1759+01	.1104+01	.4498-01	.6704-01	.4268+00	.7912+00	.1571+01	.6970-01	.1872+02
.7072+00	.1767+01	.1158+01	.4548-01	.6753-01	.4268+00	.7093+00	.1619+01	.6833-01	.1874+02
.7130+00	.1776+01	.1211+01	.4598-01	.6801-01	.4268+00	.6245+00	.1662+01	.6697-01	.1876+02
.7188+00	.1784+01	.1265+01	.4646-01	.6856-01	.4268+00	.5372+00	.1702+01	.6565-01	.1878+02
.7247+00	.1793+01	.1319+01	.4693-01	.6913-01	.4268+00	.4474+00	.1736+01	.6433-01	.1880+02
.7306+00	.1802+01	.1372+01	.4745-01	.6964-01	.4268+00	.3555+00	.1766+01	.6301-01	.1882+02
.7366+00	.1810+01	.1426+01	.4800-01	.7011-01	.4268+00	.2616+00	.1791+01	.6169-01	.1884+02
.7426+00	.1819+01	.1479+01	.4855-01	.7061-01	.4268+00	.1661+00	.1812+01	.6039-01	.1886+02
.7488+00	.1828+01	.1533+01	.4908-01	.7115-01	.4268+00	.6911-01	.1827+01	.5909-01	.1888+02
.7550+00	.1837+01	-.1555+01	.4959-01	.7174-01	.4268+00	-.2895-01	.1837+01	.5783-01	.1890+02
.7612+00	.1846+01	-.1501+01	.5012-01	.7232-01	.4268+00	-.1280+00	.1842+01	.5656-01	.1892+02
.7676+00	.1855+01	-.1448+01	.5066-01	.7291-01	.4268+00	-.2277+00	.1841+01	.5529-01	.1894+02
.7740+00	.1865+01	-.1394+01	.5121-01	.7348-01	.4268+00	-.3277+00	.1836+01	.5403-01	.1896+02
.7804+00	.1874+01	-.1340+01	.5180-01	.7402-01	.4268+00	-.4277+00	.1824+01	.5278-01	.1898+02
.7870+00	.1883+01	-.1287+01	.5235-01	.7464-01	.4268+00	-.5275+00	.1808+01	.5156-01	.1900+02
.7936+00	.1893+01	-.1233+01	.5290-01	.7528-01	.4268+00	-.6267+00	.1786+01	.5035-01	.1902+02
.8003+00	.1902+01	-.1180+01	.5348-01	.7588-01	.4268+00	-.7252+00	.1758+01	.4914-01	.1904+02
.8070+00	.1912+01	-.1126+01	.5409-01	.7645-01	.4268+00	-.8225+00	.1726+01	.4794-01	.1906+02
.8139+00	.1921+01	-.1072+01	.5472-01	.7702-01	.4268+00	-.9184+00	.1687+01	.4674-01	.1908+02
.8208+00	.1931+01	-.1019+01	.5536-01	.7758-01	.4268+00	-.1013+01	.1644+01	.4554-01	.1910+02
.8278+00	.1941+01	-.9651+00	.5601-01	.7815-01	.4268+00	-.1105+01	.1595+01	.4437-01	.1912+02

.8348+00	.1951+01	-.9114+00	.5666-01	.7873-01	.4268+00	-.1195+01	.1542+01	.4321-01	.1914+02
.8420+00	.1961+01	-.8578+00	.5732-01	.7934-01	.4269+00	-.1282+01	.1483+01	.4206-01	.1916+02
.8492+00	.1971+01	-.8042+00	.5796-01	.7998-01	.4269+00	-.1367+01	.1419+01	.4093-01	.1918+02
.8565+00	.1981+01	-.7505+00	.5861-01	.8066-01	.4269+00	-.1449+01	.1351+01	.3980-01	.1920+02
.8639+00	.1991+01	-.6969+00	.5924-01	.8136-01	.4269+00	-.1527+01	.1278+01	.3869-01	.1922+02
.8714+00	.2001+01	-.6433+00	.5990-01	.8206-01	.4269+00	-.1601+01	.1200+01	.3760-01	.1924+02
.8790+00	.2012+01	-.5896+00	.6062-01	.8268-01	.4269+00	-.1672+01	.1119+01	.3650-01	.1926+02
.8866+00	.2022+01	-.5360+00	.6129-01	.8339-01	.4269+00	-.1738+01	.1033+01	.3543-01	.1928+02
.8944+00	.2033+01	-.4823+00	.6202-01	.8404-01	.4269+00	-.1801+01	.9428+00	.3437-01	.1930+02
.9022+00	.2043+01	-.4287+00	.6272-01	.8477-01	.4269+00	-.1858+01	.8493+00	.3332-01	.1932+02
.9101+00	.2054+01	-.3751+00	.6342-01	.8552-01	.4269+00	-.1911+01	.7524+00	.3230-01	.1934+02
.9182+00	.2065+01	-.3214+00	.6419-01	.8618-01	.4269+00	-.1959+01	.6523+00	.3127-01	.1936+02
.9263+00	.2076+01	-.2678+00	.6493-01	.8692-01	.4269+00	-.2002+01	.5492+00	.3027-01	.1938+02
.9345+00	.2086+01	-.2142+00	.6567-01	.8767-01	.4269+00	-.2039+01	.4435+00	.2928-01	.1940+02
.9428+00	.2098+01	-.1605+00	.6643-01	.8844-01	.4269+00	-.2071+01	.3353+00	.2830-01	.1942+02
.9512+00	.2109+01	-.1069+00	.6720-01	.8922-01	.4269+00	-.2097+01	.2250+00	.2734-01	.1944+02
.9597+00	.2120+01	-.5328-01	.6800-01	.8995-01	.4269+00	-.2117+01	.1129+00	.2638-01	.1946+02
.9682+00	.2131+01	.3571-03	.6878-01	.9076-01	.4269+00	-.2131+01	-.7611-03	.2544-01	.1948+02
.9769+00	.2143+01	.5399-01	.6962-01	.9151-01	.4269+00	-.2140+01	-.1156+00	.2452-01	.1950+02
.9857+00	.2154+01	.1076+00	.7045-01	.9229-01	.4269+00	-.2142+01	-.2314+00	.2360-01	.1952+02
.9946+00	.2166+01	.1612+00	.7124-01	.9316-01	.4269+00	-.2138+01	-.3477+00	.2272-01	.1954+02
.1004+01	.2178+01	.2149+00	.7208-01	.9401-01	.4269+00	-.2128+01	-.4643+00	.2183-01	.1956+02
.1013+01	.2190+01	.2685+00	.7294-01	.9483-01	.4268+00	-.2111+01	-.5808+00	.2096-01	.1958+02
.1022+01	.2201+01	.3221+00	.7386-01	.9558-01	.4268+00	-.2088+01	-.6970+00	.2009-01	.1960+02
.1031+01	.2214+01	.3758+00	.7474-01	.9643-01	.4268+00	-.2059+01	-.8123+00	.1925-01	.1962+02
.1041+01	.2226+01	.4294+00	.7562-01	.9732-01	.4268+00	-.2024+01	-.9266+00	.1842-01	.1964+02
.1050+01	.2238+01	.4830+00	.7652-01	.9822-01	.4268+00	-.1982+01	-.1039+01	.1760-01	.1966+02
.1060+01	.2250+01	.5367+00	.7744-01	.9911-01	.4268+00	-.1934+01	-.1151+01	.1680-01	.1968+02
.1070+01	.2263+01	.5903+00	.7838-01	.1000+00	.4268+00	-.1880+01	-.1260+01	.1601-01	.1970+02
.1080+01	.2276+01	.6439+00	.7933-01	.1009+00	.4268+00	-.1820+01	-.1366+01	.1524-01	.1972+02
.1090+01	.2288+01	.6976+00	.8031-01	.1018+00	.4268+00	-.1754+01	-.1470+01	.1448-01	.1974+02
.1100+01	.2301+01	.7512+00	.8124-01	.1028+00	.4268+00	-.1682+01	-.1571+01	.1374-01	.1976+02
.1110+01	.2314+01	.8048+00	.8228-01	.1037+00	.4267+00	-.1604+01	-.1668+01	.1301-01	.1978+02
.1121+01	.2327+01	.8584+00	.8325-01	.1047+00	.4267+00	-.1521+01	-.1761+01	.1230-01	.1980+02
.1131+01	.2340+01	.9121+00	.8430-01	.1056+00	.4267+00	-.1433+01	-.1851+01	.1160-01	.1982+02
.1142+01	.2354+01	.9657+00	.8537-01	.1065+00	.4267+00	-.1339+01	-.1936+01	.1092-01	.1984+02
.1153+01	.2367+01	.1019+01	.8643-01	.1075+00	.4267+00	-.1240+01	-.2016+01	.1025-01	.1986+02
.1163+01	.2381+01	.1073+01	.8749-01	.1085+00	.4266+00	-.1137+01	-.2092+01	.9605-02	.1988+02
.1175+01	.2394+01	.1127+01	.8856-01	.1096+00	.4266+00	-.1029+01	-.2162+01	.8980-02	.1990+02
.1186+01	.2408+01	.1180+01	.8967-01	.1106+00	.4266+00	-.9171+00	-.2227+01	.8369-02	.1992+02
.1197+01	.2422+01	.1234+01	.9081-01	.1116+00	.4265+00	-.8011+00	-.2286+01	.7776-02	.1994+02
.1209+01	.2436+01	.1287+01	.9197-01	.1125+00	.4265+00	-.6815+00	-.2339+01	.7198-02	.1996+02
.1220+01	.2450+01	.1341+01	.9316-01	.1135+00	.4264+00	-.5585+00	-.2386+01	.6637-02	.1998+02
.1232+01	.2465+01	.1394+01	.9435-01	.1145+00	.4264+00	-.4324+00	-.2427+01	.6097-02	.2000+02

REFERENCES

1. Etsion, I: Dynamic Analysis of Noncontacting Face Seals. NASA TM-79294, 1980.

2. Gear, C. William: Numerical Initial Value Problems in Ordinary Differential Equations. Pretice-Hall, Inc., 1971.

3. Scarborough, J. B.: Numerical Mathematical Analysis. Third ed. John Hopkins Press, 1955, pp. 145-150.

4. Goldstein, Charles M.: Numerical Integration by Gaussian Quadrature. NASA TM X-52088, 1965, pp. 9-10.

E-650

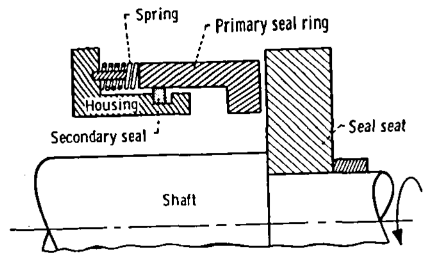


Figure 1. - Schematic of a radial face seal.

CS-80-3200

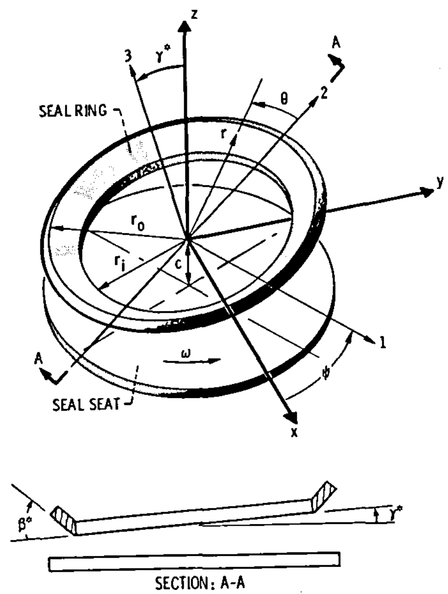


Figure 2. - Seal model and coordinates systems.

CS-80-3199

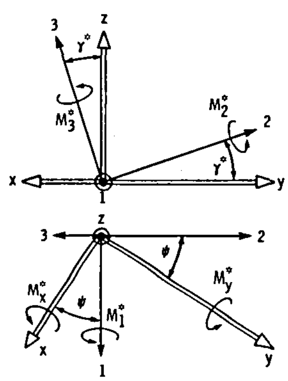


Figure 3. - Orientation of rotating cooridinate system 123 in inertial reference xyz.

CS-80-3198

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM-81636		

4. Title and Subtitle	
COMPUTER PROGRAM DOCUMENTATION FOR THE DYNAMIC ANALYSIS OF A NONCONTACTING MECHANICAL FACE SEAL	5. Report Date: November 1980
	6. Performing Organization Code
7. Author(s): B. M. Auer and I. Etsion	8. Performing Organization Report No.: E-650
	10. Work Unit No.
9. Performing Organization Name and Address: National Aeronautics and Space Administration, Lewis Research Center, Cleveland, Ohio 44135	11. Contract or Grant No.
	13. Type of Report and Period Covered: Technical Memorandum
12. Sponsoring Agency Name and Address: National Aeronautics and Space Administration, Washington, D.C. 20546	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

A computer program is presented which achieves a numerical solution for the equations of motion of a noncontacting mechanical face seal. The flexibly-mounted primary seal ring motion was expressed by a set of second order differential equations for three degrees of freedom. These equations were reduced to a set of first order equations and the GEAR software package was used to solve the set of first order equations. Program input includes seal design parameters and seal operating conditions. Output from the program includes velocities and displacements of the seal ring about the axis of an inertial reference system. One example problem is described.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Computer program Software Seal dynamics Mechanical face seals	Unclassified - unlimited STAR Category 37

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified		

* For sale by the National Technical Information Service, Springfield, Virginia 22161

National Aeronautics and
Space Administration

Washington, D.C.
20546

Official Business
Penalty for Private Use, $300

SPECIAL FOURTH CLASS MAIL
BOOK

Postage and Fees Paid
National Aeronautics and
Space Administration
NASA-451

U.S.MAIL

NASA

POSTMASTER: If Undeliverable (Section 158
Postal Manual) Do Not Return