N7331952

NASA TECHNICAL MEMORANDUM

NASA TM X-2740

NASA TM X-2740

(NASA-TM-X-2740) COMPUTER PROGRAMS FOR CALCULATING AND PLOTTING THE STABILITY CHARACTERISTICS OF A BALLOON TETHERED IN A WIND (NASA) ~~147~~ 150 p HC $4.50 CSCL 01B

N73-31952

Unclas

H1/02 18639

RECEIVED NASA STI FACILITY INPUT BRANCH OCT 1973

COMPUTER PROGRAMS FOR CALCULATING AND PLOTTING THE STABILITY CHARACTERISTICS OF A BALLOON TETHERED IN A WIND

by Robert M. Bennett, Samuel R. Bland, and L. Tracy Redd

Langley Research Center
Hampton, Va. 23665

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • SEPTEMBER 1973

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM X-2740		

4. Title and Subtitle COMPUTER PROGRAMS FOR CALCULATING AND PLOTTING THE STABILITY CHARACTERISTICS OF A BALLOON TETHERED IN A WIND	5. Report Date September 1973
	6. Performing Organization Code
7. Author(s) Robert M. Bennett, Samuel R. Bland, and L. Tracy Redd	8. Performing Organization Report No. L-8542
9. Performing Organization Name and Address NASA Langley Research Center Hampton, Va. 23665	10. Work Unit No. 501-22-04-01
	11. Contract or Grant No.
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546	13. Type of Report and Period Covered Technical Memorandum
	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

Descriptions are presented for six related computer programs for calculating the stability characteristics of a balloon tethered in a steady wind. Equilibrium conditions, characteristic roots, and modal ratios are calculated for a range of discrete values of velocity for a fixed tether-line length. Separate programs are used (1) to calculate longitudinal stability characteristics, (2) to calculate lateral stability characteristics, (3) to plot the characteristic roots versus velocity, (4) to plot the characteristic roots in root-locus form, (5) to plot the longitudinal modes of motion, and (6) to plot the lateral modes of motion. The basic equations, program listings, and the input and output data for sample cases are presented, with a brief discussion of the overall operation and limitations. The programs are based on a linearized, stability-derivative type of analysis, including balloon aerodynamics, apparent mass, buoyancy effects, and static forces which result from the tether line.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Aerostat Eigenvalue Quadratic determinant Stability Computer graphics	Unclassified – Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	147	$3.00

*For sale by the National Technical Information Service, Springfield, Virginia 22151

CONTENTS

	Page
SUMMARY	1
INTRODUCTION	1
SYMBOLS	2
GENERAL DESCRIPTION OF PROGRAMS	5
LONGITUDINAL STABILITY PROGRAM	6
Longitudinal Equations of Motion	8
Definition of Program Variables	14
Input Required for Longitudinal Stability Program	18
Limitations and Diagnostic Messages	22
Listing of Input Data Cards for Sample Case	23
Listing of Longitudinal Stability Program	24
Printing of Files Containing Calculated Results for Sample Case	40
LATERAL STABILITY PROGRAM	53
Lateral Equations of Motion	55
Input Required for Lateral Stability Program	57
Listing of Input Data Cards for Sample Case	61
Listing of Lateral Stability Program	62
Printing of Files Containing Calculated Results for Sample Case	70
PROGRAM FOR PLOTTING FREQUENCIES AND DECAY RATES VERSUS WIND VELOCITY	85
General Description	85
Definitions of Program Variables	85
Listing of Program	89
PROGRAM FOR PLOTTING ROOTS IN ROOT-LOCUS FORM WITH WIND VELOCITY AS A PARAMETER	91
General Description	91
Listing of Program	93
PROGRAM FOR PLOTTING LONGITUDINAL MODES OF MOTION	95
Definition of Program Variables	96
Description of Input Data	97
Listing of Input Data Cards for Sample Case	98
Listing of Program	99

	Page
PROGRAM FOR PLOTTING LATERAL MODES OF MOTION	102
Listing of Input Data Cards for Sample Case	103
Listing of Program	104
APPENDIX – DESCRIPTIONS OF SELECTED SUBROUTINES	107
Basic Subroutines	107
Plotting Subroutines	132
REFERENCES	145

COMPUTER PROGRAMS FOR CALCULATING AND PLOTTING THE STABILITY CHARACTERISTICS OF A BALLOON TETHERED IN A WIND

By Robert M. Bennett, Samuel R. Bland,
and L. Tracy Redd
Langley Research Center

SUMMARY

Descriptions are presented for six related computer programs for calculating the stability characteristics of a balloon tethered in a steady wind. Equilibrium conditions, characteristic roots, and modal ratios are calculated for a range of discrete values of velocity for a fixed tether-line length. Separate programs are used (1) to calculate longitudinal stability characteristics, (2) to calculate lateral stability characteristics, (3) to plot the characteristic roots versus velocity, (4) to plot the characteristic roots in root-locus form, (5) to plot the longitudinal modes of motion, and (6) to plot the lateral modes of motion. The basic equations, program listings, and the input and output data for sample cases are presented, with a brief discussion of the overall operation and limitations. The programs are based on a linearized, stability-derivative type of analysis, including balloon aerodynamics, apparent mass, buoyancy effects, and static forces which result from the tether line.

INTRODUCTION

Tethered balloons are used for many purposes such as supporting antennas or carrying measuring instruments aloft. Dynamic instabilities during high-wind conditions often limit the operation or utility of these devices. Although some limited early stability work (refs. 1 to 2, for example) and more recently some work including cable dynamics (refs. 3 to 6, for example) has been done, a systematic study of the factors involved in the stability of tethered balloons is apparently lacking. The Langley Research Center has undertaken a research study to develop methods for stability analysis. Portions of this study are given in references 7 to 10.

The purpose of this report is to list and describe the operation and use of six related computer programs for calculating the stability characteristics of a balloon tethered in a steady wind and for plotting the results. The analysis on which the programs are based is given in reference 8. It is essentially a linearized, stability-derivative type of analysis of dynamic motions in either the longitudinal plane or in the lateral plane. Buoyancy forces, aerodynamic apparent masses, and the static spring forces resulting from the tether cable are included, in addition to the usual static and dynamic aerodynamic terms.

A listing of each program is given along with a listing of input and output data for a sample case. The overall operation of the programs and some of their limitations are discussed. Usage descriptions of several of the basic subroutines are given in the appendix.

SYMBOLS

In addition to the symbol definitions given here, symbols relating to the input data are more specifically defined in the sections of this report which describe the input data required by the programs.

A	matrix of coefficients of acceleration terms in equations of motion
B	matrix of coefficients of velocity or rate terms in equations of motion
C	matrix of coefficients of displacement terms in equations of motion
C_D	drag coefficient, $\frac{D}{\rho V^2 S/2}$
C_L	lift coefficient, $\frac{L}{\rho V^2 S/2}$
C_l	rolling-moment coefficient, $\frac{\text{Rolling moment}}{\rho V^2 S\bar{c}/2}$
C_m	pitching-moment coefficient, $\frac{M}{\rho V^2 S\bar{c}/2}$
C_n	yawing-moment coefficient, $\frac{\text{Yawing moment}}{\rho V^2 S\bar{c}/2}$
C_Y	side-force coefficient, $\frac{\text{Side force}}{\rho V^2 S/2}$
$\bar{c}$	reference length

D,L,M	drag force, lift force, and pitching moment, respectively (see fig. 2)
d_c	tether cable diameter (see fig. 2)
h_{br}	component of distance from reference point to center of buoyancy (see fig. 2)
h_{cg}	component of distance from reference point to center of mass (see fig. 2)
h_{sr}	component of distance from reference point to center of mass of balloon structure (see fig. 2)
I_y	pitching moment of inertia about balloon center of mass
k	tether derivatives defined by equations (16) and (18) and equations (22) and (23)
l_{br}	component of distance from reference point to center of buoyancy (see fig. 2)
l_{cg}	component of distance from reference point to center of mass (see fig. 2)
l_{sr}	component of distance from reference point to center of mass of balloon structure (see fig. 2)
l_{tr}	component of distance from reference point to attachment point of tether line (see fig. 2)
m_T	mass of balloon structure and contained gas
$m_{x,a},m_{y,a},m_{z,a}$	aerodynamic apparent mass in body-reference X-axis, Y-axis, and Z-axis directions, respectively
n	number of degrees of freedom or quantity defined by equation (12a)
p	perturbation roll rate
$\bar{p}$	quantity defined by equation (12b)
q	perturbation pitch rate
q_i	generalized coordinate
$\bar{q}$	quantity defined by equation (12c)

r perturbation yaw rate

S reference area

T_0, T_1 tension of tether cable at lower and upper ends, respectively (see fig. 2)

t time

t_{tr} component of distance from reference point to attachment point of tether line (see fig. 2)

u perturbation velocity along stability X-axis

V velocity

W_s structural weight of balloon (see fig. 2)

w_c weight per unit length of tether cable

x,y,z coordinate displacements in body-fixed stability-axis system with origin at center of mass

x_1, z_1 coordinates of balloon center of mass (see fig. 2)

α perturbation angle of attack

α_t trim angle of attack

β angle of sideslip

γ_0, γ_1 angles between the horizontal and tether cable, respectively (see fig. 2)

θ, ϕ, ψ angular perturbations about the X-, Y-, and Z-axis, respectively

λ characteristic root

$Im(\lambda)$ frequency

$Re(\lambda)$ decay rate

$\bar{\lambda}$ variable defined by equation (12d)

ρ air density

τ variable defined by equation (12e)

Subscript:

R reference point

Dots over variables indicate differentiation with respect to time.

GENERAL DESCRIPTION OF PROGRAMS

A linearized, stability-derivative type of analysis, such as the one considered here, generally results in a system of simultaneous, linear, ordinary, second-order differential equations with constant coefficients. In order to examine the stability of such a system, a solution of the form $q_i = \bar{q}_i e^{\lambda t}$ for exponentially varying motion is assumed, where q_i is a generalized coordinate and $\bar{q}_i$ is a complex constant. The resulting stability determinant is of order $n \times n$, where n is the number of degrees of freedom, and has elements that are quadratic in λ. Thus the determinant has $2n$ characteristic roots or eigenvalues. For solution of the stability determinant, the use of a standard eigenvalue computer subroutine requires a transformation to a $2n \times 2n$ determinant with λ on the principal diagonal only (see description of subroutine QUADET in the appendix). The sign of the real part of λ, $\mathrm{Re}(\lambda)$, signifies growth or decay of a mode of motion of the system, with $\mathrm{Re}(\lambda) > 0$ indicating a growing motion (instability). Additional insight about the modes of motion of the dynamic system can also be obtained by substituting each characteristic value into the stability determinant and solving for the associated modal ratios or eigenvector elements.

The six programs described herein calculate the characteristic roots and modal ratios for a range of discrete values of velocity for a fixed tether-line length and plot the results. These programs are:

(1) Program STABLTY for longitudinal stability calculations

(2) Program STBLTY2 for lateral stability calculations

(3) Program VPLOT for plotting frequencies $\mathrm{Re}(\lambda)$ and decay rates $\mathrm{Im}(\lambda)$ versus wind velocity

(4) Program RTLOCUS for plotting roots in root-locus form with wind velocity as a parameter

(5) Program CALBALM for plotting longitudinal modes of motion

(6) Program CALBLM2 for plotting lateral modes of motion

A block diagram illustrating the relationship of these programs and their use is given in figure 1. Although the programs can be operated in any consistent system of units, the constants and labels are generally given in the SI unit system. The pertinent cards are labeled SI UNITS in the comments field (cols. 73 to 80).

The programs are written basically in FORTRAN language for the CDC 6000 series machines with the Langley Research Center version of the SCOPE 3.0 operating system and the RUN compiler. Some of the system library subroutines used by these programs are in the COMPASS language. A FORTRAN simulator for one of the more essential subroutines (MASCNT) is included in the appendix to facilitate use on other systems.

The programs were designed for use through a low-speed terminal system with the program stored on a data cell system at the central computer complex. Efficient usage of the low-speed terminal requires keeping the INPUT and OUTPUT files to minimum length. Thus, the results of the calculations are written onto disk files and routed after execution (fig. 1). In addition, many of the programs have their data for execution included in DATA statements and only the necessary case data or changes to the nominal case are read. Although written in a form suitable for construction of a single program with several levels of overlays, the programs have been left separate and are used sequentially with multiple executions and disk communication between programs. The zero-level overlay is used, however, to reduce the field length for loading. In this form the largest program (lateral version of STABLTY) loads and executes with a field length of 31000_8. Typical execution time for STABLTY is about 45 seconds of central processor unit time for one case of 100 velocity increments.

The four plotting programs described are written for a CalComp Electro-Mechanical Plotter, using the Langley Research Center (LRC) plotting system computer software. Relatively high-quality, hard-copy plots are produced by this system. The basic plotting subroutines are not given, but writeups are included in the appendix in order to facilitate program conversion for other systems.

LONGITUDINAL STABILITY PROGRAM

The longitudinal stability program has been adapted from an essentially general program for calculating the eigenvalues of a stability determinant with elements that are quadratic in the eigenvalue. The main program primarily calls working subroutines and handles a portion of the input and output. It contains one main loop for incrementing and varying the wind velocity. The coefficient matrices are generated by calling subroutine

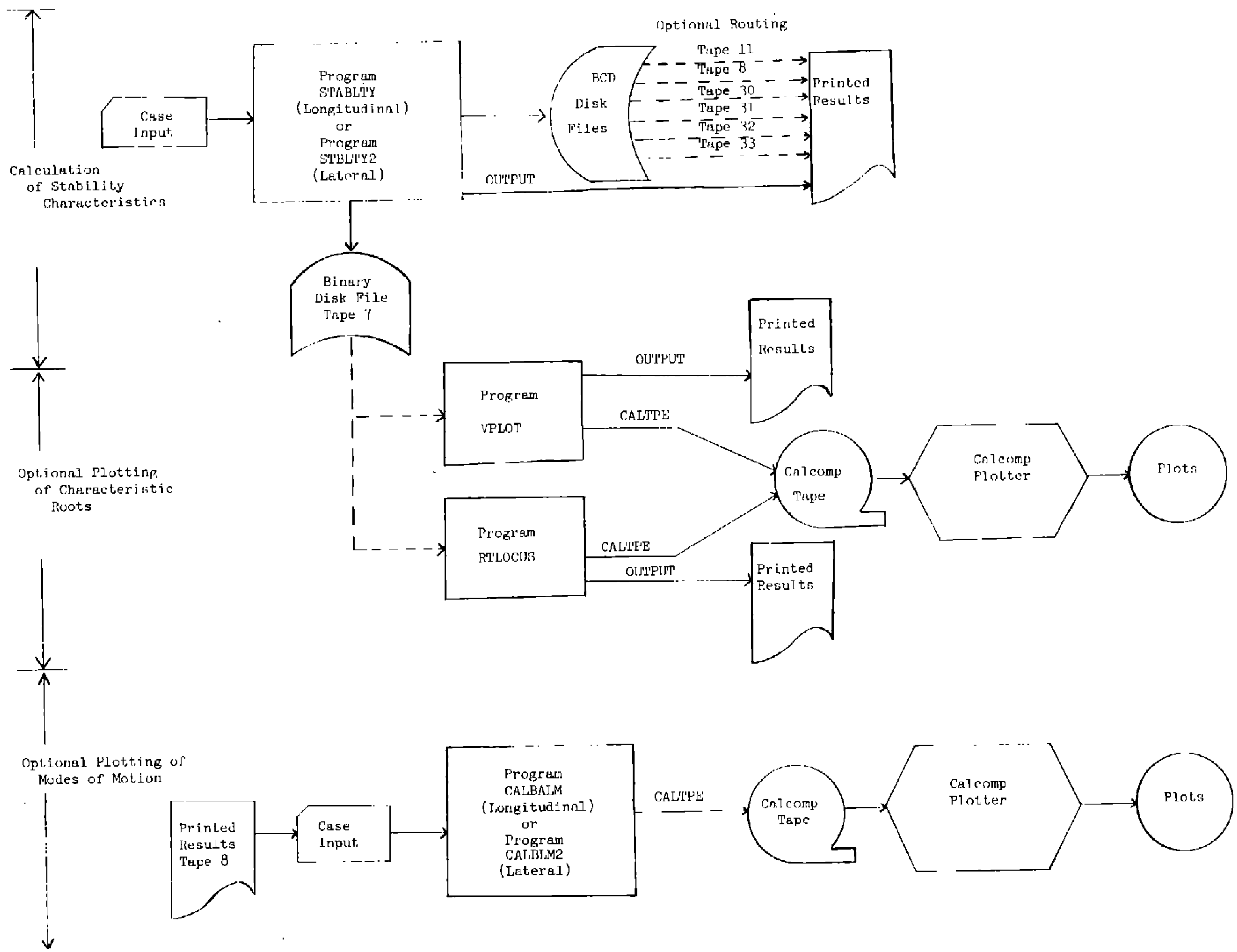


Figure 1.- Block diagram of programs for calculating and plotting stability characteristics of a tethered balloon.

INICOEF which contains the entry point VCOEF. Calculations that are independent of the wind velocity are performed by INICOEF, including the reading of the NAMELIST for input data. The velocity-dependent calculations are performed within the entry point VCOEF. For each value of velocity, the balloon equilibrium conditions such as tether-line angles, height, downstream distance (fig. 2), and aerodynamic trim conditions are calculated. The static aerodynamic coefficients C_D, C_L, and C_m are calculated by function subprograms that are to be written by the user to describe the aerodynamics of the balloon configuration to be analyzed. These functions are not restricted to linear functions and are written for a reference point (fig. 2). The program transfers the coefficients to the center of mass for use in the calculations, thereby facilitating parameter studies.

The eigenvalues are sorted in the order of ascending frequencies $\mathrm{Im}(\lambda)$ within the main program. Since the present system has real coefficient matrices, complex roots or eigenvalues occur only as complex pairs. For each such pair, the root with positive frequency contains all the needed information. Thus the conjugate root with a negative value of frequency is generally not printed on the files for output.

A symbol cycling technique is used that has been found to be helpful to relate printed and plotted results. A single symbol (plus signs in the plotting programs herein) is used for plotting all points except for every tenth increment of velocity. For every tenth velocity increment, the results are plotted using the standard NASA symbol sequence of circle, square, etc., and the name of the symbol is printed on the printed results. The indexing parameters for the symbol cycling are set up in the main program and are written on tape 7 for later use in plotting, and the symbol name is written on tapes 8 and 11 (blanks on tapes 8 and 11 if plus signs are to be used). This technique was taken from an unpublished flutter program written by Robert N. Desmarais of the Langley Research Center.

Longitudinal Equations of Motion

The equations of longitudinal motion written about the center of mass are (see ref. 9):

x-force

$$\ddot{x} + \frac{\rho VS}{2m_x}\left(2C_D + C_{D_u}\right)\dot{x} + \frac{k_{xx}}{m_x}x + \frac{\rho VS}{2m_x}\left(C_{D_\alpha} - C_L\right)\dot{z} + \frac{k_{xz}}{m_x}z$$

$$+ \left(\frac{k_{x\theta}}{m_x} + \frac{\rho V^2 SC_{D_\alpha}}{2m_x}\right)\theta = 0 \qquad (1)$$

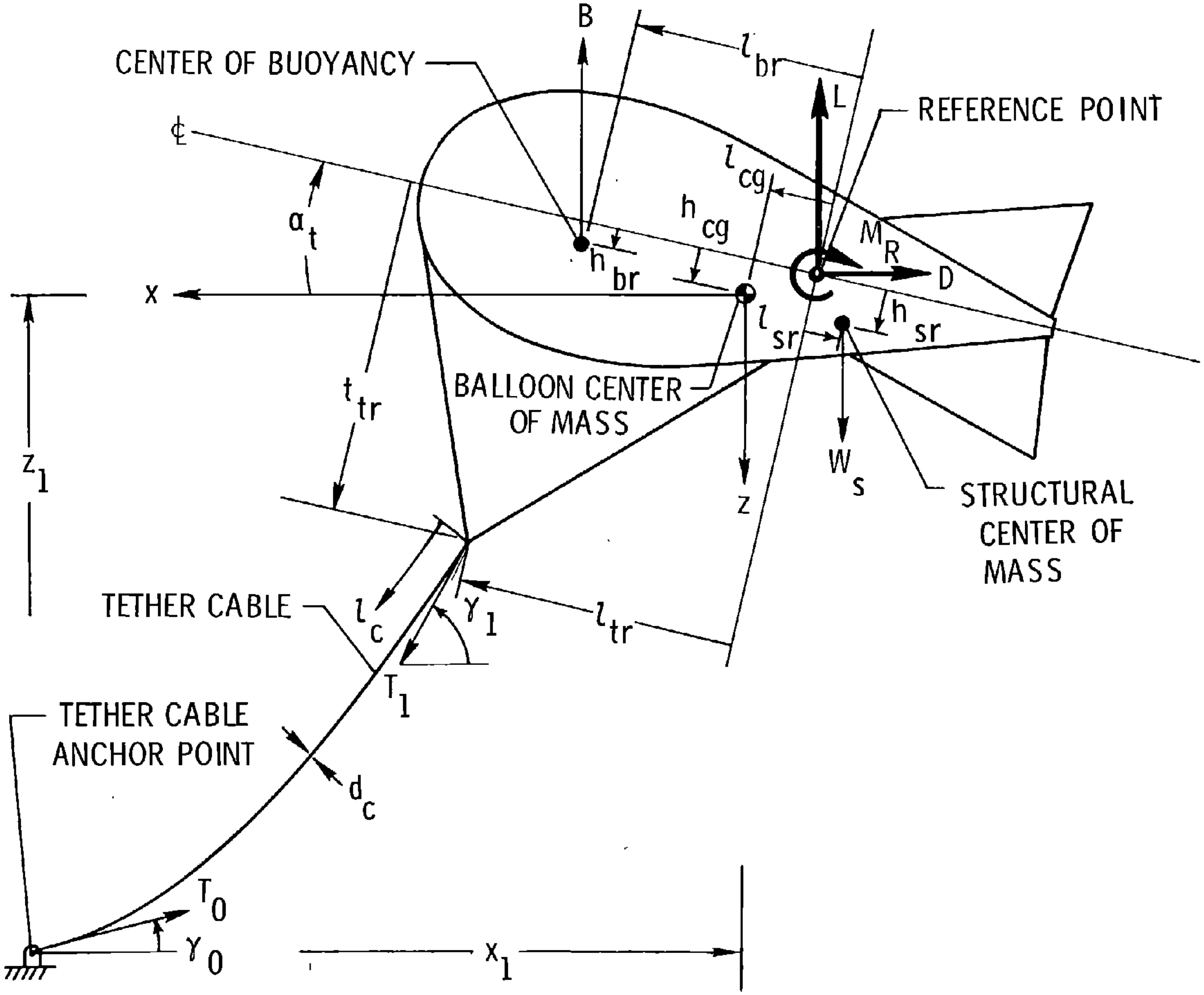


Figure 2.- Sketch of the balloon identifying pertinent dimensional relationships.
(All arrows are pointing in the positive direction.)

z-force

$$\frac{\rho VS}{2m_z}\left(2C_L + C_{L_u}\right)\dot{x} + \frac{k_{zx}}{m_z}x + \ddot{z} + \frac{\rho VS}{2m_z}\left(C_{L_\alpha} + C_D\right)\dot{z} + \frac{k_{zz}}{m_z}z$$

$$+ \frac{\rho VS\bar{c}}{4m_z}\left(C_{L_{\dot{\alpha}}} + C_{L_q}\right)\dot{\theta} + \left(\frac{k_{z\theta}}{m_z} + \frac{\rho V^2SC_{L_\alpha}}{2m_z}\right)\theta = 0 \tag{2}$$

Pitching moment

$$-\frac{\rho VS\bar{c}}{2I_y}\left(2C_m + C_{m_u}\right)\dot{x} + \frac{k_{\theta x}}{I_y}x - \frac{\rho S\bar{c}^2}{4I_y}C_{m_{\dot{\alpha}}}\ddot{z} - \frac{\rho VS\bar{c}}{2I_y}C_{m_\alpha}\dot{z} + \frac{k_{\theta z}}{I_y}z$$

$$+ \ddot{\theta} - \frac{\rho VS\bar{c}^2}{4I_y}\left(C_{m_{\dot{\alpha}}} + C_{m_q}\right)\dot{\theta} + \left(\frac{M_{s_1} + k_{\theta\theta}}{I_y} - \frac{\rho V^2S\bar{c}}{2I_y}C_{m_\alpha}\right)\theta = 0 \tag{3}$$

where the mass and buoyancy terms in equations (1) to (3) are given by

$$m_x = m_T + m_{x,a}\cos^2\alpha_t + m_{z,a}\sin^2\alpha_t \tag{4}$$

$$m_z = m_T + m_{x,a}\sin^2\alpha_t + m_{z,a}\cos^2\alpha_t + \frac{\rho S\bar{c}}{4}C_{L_{\dot{\alpha}}} \tag{5}$$

and

$$M_{s_1} = \left[\left(l_{br} - l_{cg}\right)B + \left(l_{sr} + l_{cg}\right)W_s\right]\sin\alpha_t$$

$$+ \left[\left(h_{cg} - h_{br}\right)B + \left(h_{sr} - h_{cg}\right)W_s\right]\cos\alpha_t \tag{6}$$

The coefficients of equations (1) to (6) are evaluated for the steady equilibrium or trim conditions and thus the values of α_t, z_1, T_0, γ_0, etc. (fig. 2) are required. The value of α_t is calculated by Newton iteration of the following equation (in subroutine TRIM) which is implicit in α_t and results from combining the lift, drag, and pitching-moment trim equations (ref. 9):

$$h_{k_1}\left(C_L\frac{\rho V^2S}{2} + B - W_s\right) - M_{s_2} - \left(h_{k_2}C_D + \bar{c}C_m\right)\frac{\rho V^2S}{2} = 0 \tag{7}$$

where h_{k_1}, h_{k_2}, and M_{s_2} are functions of α_t given by

$$h_{k1} = \left(l_{tr} - l_{cg}\right)\cos \alpha_t + \left(t_{tr} - h_{cg}\right)\sin \alpha_t \tag{8a}$$

$$h_{k2} = \left(t_{tr} - h_{cg}\right)\cos \alpha_t - \left(l_{tr} - l_{cg}\right)\sin \alpha_t \tag{8b}$$

and

$$\begin{aligned} M_{s_2} = &\left[\left(l_{br} - l_{cg}\right)B + \left(l_{sr} + l_{cg}\right)W_s\right]\cos \alpha_t \\ &- \left[\left(h_{cg} - h_{br}\right)B + \left(h_{sr} - h_{cg}\right)W_s\right]\sin \alpha_t \end{aligned} \tag{9}$$

and C_D, C_L, and C_m are also functions of α_t. It might be noted that the increments ΔC_D, ΔC_L, and ΔC_m, which are used in the program for parametric studies, are included in the trim calculation as constants but are not considered as functions of α_t.

The aerodynamic coefficients in equations (1) to (3) and equation (7) are referred to the center of mass. The coefficients are given in the program about a reference point (fig. 2) and are transferred to the center of mass by the following relations in subroutines TRIM and DERTRAN:

$$C_{L_q} = C_{L_{q,R}} + 2\frac{x_t}{\bar{c}} C_{L_\alpha} - 2\frac{z_t}{\bar{c}}\left(2C_L + C_{L_u}\right) \tag{10a}$$

$$C_m = C_{m,R} - \frac{x_t}{\bar{c}} C_L + \frac{z_t}{\bar{c}} C_D \tag{10b}$$

$$C_{m_u} = C_{m_{u,R}} - \frac{x_t}{\bar{c}} C_{L_u} + \frac{z_t}{\bar{c}} C_{D_u} \tag{10c}$$

$$C_{m_\alpha} = C_{m_{\alpha,R}} - \frac{x_t}{\bar{c}}\left(C_{L_\alpha} + C_D\right) - \frac{z_t}{\bar{c}}\left(C_L - C_{D_\alpha}\right) \tag{10d}$$

$$C_{m_{\dot\alpha}} = C_{m_{\dot\alpha,R}} - \frac{x_t}{\bar{c}} C_{L_{\dot\alpha}} \tag{10e}$$

$$C_{m_q} = C_{m_{q,R}} - \frac{x_t}{\bar{c}}\left(C_{L_{q,R}} - 2C_{m_\alpha}\right) - 2\frac{z_t}{\bar{c}}\left(2C_m + C_{m_u}\right) \tag{10f}$$

Calculation of the equilibrium tether conditions.- The equilibrium tether conditions are required for calculation of the tether derivatives as subsequently discussed. The equilibrium tether conditions are calculated as follows and are based on the analysis of reference 2. The values of T_1 and γ_1 (see fig. 2) are determined from the following equations, which are manipulations of the lift and drag trim equations:

$$T_1 = \left[\left(C_D \frac{\rho V^2 S}{2}\right)^2 + \left(B - W_s + C_L \frac{\rho V^2 S}{2}\right)^2\right]^{1/2} \tag{11a}$$

$$\gamma_1 = \cos^{-1}\left(C_D \frac{\rho V^2 S}{2T_1}\right) \tag{11b}$$

The velocity V and the cable parameters l_c, d_c, C_{D_c}, and w_c are specified. Let

$$n = C_{D_c} \frac{\rho V^2 d_c}{2} \tag{12a}$$

$$\bar{p} = \frac{w_c}{2n} \tag{12b}$$

$$\bar{q} = \sqrt{1 + \bar{p}^2} \tag{12c}$$

From the analysis of the tether, a variable $\bar{\lambda}$ is also defined as

$$\bar{\lambda}\left(\gamma_u\right) = \int_0^{\gamma_u} \left[\frac{\tau(\gamma)}{\sin^2\gamma + 2\bar{p} \cos \gamma}\right] d\gamma \tag{12d}$$

where

$$\tau(\gamma) = \left(\frac{\bar{q} + \bar{p} - \cos \gamma}{\bar{q} - \bar{p} + \cos \gamma}\right)^{\bar{p}/\bar{q}} \tag{12e}$$

γ is the cable angle, and γ_u is the value of γ associated with the upper limit of the integral in equation (12d). Since γ_1 is known from equation (11b), $\bar{\lambda}_1 = \bar{\lambda}\left(\gamma_1\right)$ is calculated (subroutine TETHER) from equation (12d), using numerical integration (subroutine ROMBERG), and $\tau_1 = \tau\left(\gamma_1\right)$ is calculated from equation (12e). The related value of $\bar{\lambda}_0$ is given by (ref. 2):

$$\bar{\lambda}_0 = \bar{\lambda}_1 - \frac{n\tau_1 l_c}{T_1} \tag{13}$$

where $\bar{\lambda}_0 = \bar{\lambda}(\gamma_0)$. Thus, the left-hand side of equation (12d) is known from equation (13) and can be evaluated for $\gamma_u = \gamma_0$. This is done in subroutine TETHER using Newton iteration (subroutine NEWINT) and numerical integration (subroutine ROMBERG). The values of T_0 and z_1 are given by (ref. 2):

$$T_0 = \frac{T_1 \tau_0}{\tau_1} \tag{14a}$$

$$z_1 = \frac{T_1 - T_0}{w_c} \tag{14b}$$

with $\tau_0 = \tau(\gamma_0)$. The value of x_1 is given by (ref. 2):

$$x_1 = \frac{T_1}{n\tau_1} \int_{\gamma_0}^{\gamma_1} \left[\frac{\tau(\gamma)}{\sin^2\gamma + 2\bar{p}\cos\gamma}\right] \cos\gamma \, d\gamma \tag{15}$$

which is also integrated numerically using subroutine ROMBERG (in subroutine TETHER).

Calculation of the tether derivatives. - The tether derivatives or spring constants k_{xx}, k_{xz}, k_{zx}, and k_{zz} required for equations (1) to (3) are also calculated using the analysis of reference 2. These derivatives are expressed in terms of the equilibrium tether conditions as follows:

$$k_{xx} = \frac{1}{\delta}\left[T_1 \cos\gamma_1 \left(\sin\gamma_1 - \sin\gamma_0\right) + n\left(z_1 - l_1 \sin\gamma_0\right)\sin^3\gamma_1\right] \tag{16a}$$

$$k_{xz} = \frac{1}{\delta}\left[T_1 \cos\gamma_1 \left(\cos\gamma_0 - \cos\gamma_1\right) + n\left(l_1 \cos\gamma_0 - x_1\right)\sin^3\gamma_1\right] \tag{16b}$$

$$k_{zx} = \frac{1}{\delta}\left[T_1 \sin\gamma_1 \left(\sin\gamma_1 - \sin\gamma_0\right) - \left(w_c + n\sin^2\gamma_1 \cos\gamma_1\right)\left(z_1 - l_1 \sin\gamma_0\right)\right] \tag{16c}$$

$$k_{zz} = \frac{1}{\delta}\left[T_1 \sin\gamma_1 \left(\cos\gamma_0 - \cos\gamma_1\right) - \left(w_c + n\sin^2\gamma_1 \cos\gamma_1\right)\left(l_1 \cos\gamma_0 - x_1\right)\right] \tag{16d}$$

where

$$\delta = x_1\left(\sin \gamma_1 - \sin \gamma_0\right) + z_1\left(\cos \gamma_0 - \cos \gamma_1\right) - l_1 \sin\left(\gamma_1 - \gamma_0\right) \tag{17}$$

The tether derivatives related to pitch angle θ and referred to the center of mass are

$$k_{\theta x} = h_{k_2}k_{xx} - h_{k_1}k_{zx} \tag{18a}$$

$$k_{\theta z} = h_{k_2}k_{xz} - h_{k_1}k_{zz} \tag{18b}$$

$$k_{x\theta} = h_{k_2}k_{xx} - h_{k_1}k_{xz} \tag{18c}$$

$$k_{z\theta} = h_{k_2}k_{zx} - h_{k_1}k_{zz} \tag{18d}$$

$$k_{\theta\theta} = k_{\theta\theta_D} + k_{\theta\theta_{T_1}} \tag{18e}$$

$$k_{\theta\theta_D} = h_{k_2}^2 k_{xx} - h_{k_2}h_{k_1}\left(k_{xz} + k_{zx}\right) + h_{k_1}^2 k_{zz} \tag{18f}$$

$$k_{\theta\theta_{T_1}} = h_{k_2}\left(T_1 \sin \gamma_1\right) + h_{k_1}\left(T_1 \cos \gamma_1\right) \tag{18g}$$

where h_{k_1} and h_{k_2} are defined by equations (8).

Definition of Program Variables

Some of the principal FORTRAN variable names are given and defined in the following sections. Where there are corresponding mathematical symbols, these symbols are also listed.

Variables in the main program.- The variables used in the main program are given as follows:

FORTRAN variable name	Mathematical symbol	Definition
A, B, C		$n \times n$ coefficient matrices of the equations of motion, i.e., mass, rate, and displacement matrices, respectively; rows 1, 2, and 3 contain the coefficients of the x-, z-, and θ-equations, respectively
CNOA, CNOEI		Turing's condition number of the matrices A and EIDET, respectively, defined in terms of the norm as, for example: CNOA = $\|A\| * \|A^{-1}\| / n$, n = Order of A
CROT, CRTSQ		a complex eigenvalue and its square, respectively
DELV	ΔV	increment in velocity
EICOEF		complex coefficient array for eigenvector calculations; here, 2×3 and normalized by θ in degrees
EIDET		stability or eigenvalue determinant in expanded form, $2n \times 2n$
ID		ten-word alphanumeric array containing case identification card and date and time for processing of case
IK2		index for cycling symbols
NEGR		number of complex eigenvalues with negative frequencies plus one
NMP		number of modes processed; here set to 3
NTWO		2 *NMP; order of linear eigenvalue determinant
NVEL		number of velocity increments
ROOTI	$Im(\lambda)$	array containing imaginary portion of eigenvalues (modal frequencies) for a given velocity

FORTRAN variable name	Mathematical symbol	Definition
ROOTR	$Re(\lambda)$	array containing real portion of eigenvalues for a given velocity in the same sequence as ROOTI
SYMBOL		eleven-word Hollerith array containing names of plotting symbols for symbol cycling
VEL	V	velocity
VMIN		minimum velocity

Variables in subroutine INICOEF.- Many of the variables used in INICOEF are listed in the NAMELIST and are defined in the section entitled "Input Required for Longitudinal Stability Program." Also, some have the same usage as in the main program. Other principal variables are given as follows:

FORTRAN variable name	Mathematical symbol	Definition
ALPHA	α_t	trim angle of attack in radians
ALPHAD	α_t	trim angle of attack in degrees
CD	C_D	drag coefficient at trim
CDA	C_{D_α}	$\partial C_D/\partial\alpha$
CDRAG	n	cable drag per unit length
CL	C_L	lift coefficient for trim
CLA	C_{L_α}	lift-curve slope
CLQ	C_{L_q}	lift pitching-rate derivative about center of mass, $\partial C_L \Big/ \partial \frac{q\bar{c}}{2V}$
CM	C_m	pitching-moment coefficient about center of mass at trim

FORTRAN variable name	Mathematical symbol	Definition
CMA	C_{m_α}	pitching-moment derivative about center of mass, $\partial C_m / \partial \alpha$
CMAD	$C_{m_{\dot{\alpha}}}$	moment α-rate stability derivative about center of mass, $\partial C_m \Big/ \partial \frac{\dot{\alpha}\bar{c}}{2V}$
CMQ	C_{m_q}	moment pitching-rate stability derivative about center of mass, $\partial C_m \Big/ \partial \frac{q\bar{c}}{2V}$
GAMO	γ_0	cable angle at ground measured from horizontal
GAM1	γ_1	cable angle at tether point on bridle measured from horizontal
Q		dynamic pressure, $\rho V^2/2$
SKTT	$k_{\theta\theta}$	total tether pitch spring in body-axis system for pitch about center of mass
SKTTD	$k_{\theta\theta_D}$	portion of SKTT due to displacement of tether point for pitch about center of mass
SKTTT	$k_{\theta\theta_T}$	portion of SKTT due to rotation of balloon relative to steady tension vector at tether point
SKTZ	$k_{\theta z}$	tether pitching moment due to z-displacement of balloon
SKTX	$k_{\theta x}$	tether pitching moment due to x-displacement of balloon
SKXT	$k_{x\theta}$	tether x-force due to pitching displacement
SKXX	k_{xx}	tether x-spring constant at tether point
SKXZ	k_{xz}	tether x-force due to z-displacement

FORTRAN variable name	Mathematical symbol	Definition
SKZT	$k_{z\theta}$	tether z-force due to pitching displacement
SKZX	k_{zx}	tether z-force due to x-displacement
SKZZ	k_{zz}	tether z-spring constant at tether point
TO	T_0	tether cable tension at ground (see fig. 2)
T1	T_1	tether cable tension at bridle (see fig. 2)
XOC	$x_t/\bar{c}$	x-distance in stability-axis system from reference point to center of mass
X1	x_1	balloon horizontal displacement, positive in direction of wind
ZOC	$z_t/\bar{c}$	z-distance in stability-axis system from reference point to center of mass
Z1	z_1	balloon altitude
UNCRT		complex eigenvalue obtained by factoring diagonal quadratic element of stability determinant

It may also be noted that the tether subroutines called by VCOEF are written with FORTRAN variable names closely paralleling the mathematical notation of reference 2.

Input Required for Longitudinal Stability Program

The user-written function subroutines FCD, FCL, and FCMR describe the longitudinal static aerodynamic coefficients about the reference point for the configuration. In the present usage, the curve fits (ref. 8) to the measured coefficients as functions of angle of attack are used. The static coefficients C_D, C_L, and $C_{m,R}$ are associated with the function variable names. Angle of attack is passed as a formal parameter and the derivatives C_{D_α}, C_{L_α}, and $C_{m_{\alpha,R}}$ are returned as formal parameters of the functions FCD, FCL, and FCMR, respectively. These functions must be replaced with functions appropriate for the configuration to be analyzed and are thus considered part of the input data.

For each case, one card of 80 characters of case identification is read in an 8A10 format, and a NAMELIST called LONGDTA is read. The FORTRAN variable names, their equivalent mathematical symbols, and their definitions are given as follows in the order the variables are listed in the NAMELIST, which is also the order for printing. All variables are preset in the program with DATA statements to the values for the reference configuration of the LRC balloon, and only changes need to be read with the NAMELIST. Thus, the program can be executed using no changes in the parameters in the NAMELIST.

FORTRAN variable name	Mathematical symbol	Definition
CDINS	$C_{D_{ins}}$	constant increment of C_D (allows for C_D of instrument package of balloon)
CLAD	$C_{L_{\dot{\alpha}}}$	lift α-rate stability derivative, $\partial C_L / \partial \frac{\dot{\alpha}\bar{c}}{2V}$
CLQR	$C_{L_{q,R}}$	lift pitching-rate stability derivative about reference point, $\partial C_L / \partial \frac{q\bar{c}}{2V}$
CMADR	$C_{m_{\dot{\alpha},R}}$	moment α-rate stability derivative about reference point, $\partial C_{m,R} / \partial \frac{\dot{\alpha}\bar{c}}{2V}$
CMQR	$C_{m_{q,R}}$	moment pitching-rate stability derivative about reference point, $\partial C_{m,R} / \partial \frac{q\bar{c}}{2V}$
DELCD	ΔC_D	constant increments in coefficients about center of mass which are used for parametric studies
DELCDA	ΔC_{D_α}	
DELCL	ΔC_L	
DELCLA	ΔC_{L_α}	
DELCM	ΔC_m	
DELCMA	ΔC_{m_α}	

FORTRAN variable name	Mathematical symbol	Definition
CDU	C_{D_u}	rate of change of drag coefficient with velocity, $\partial C_D / \partial \frac{u}{V}$
CLU	C_{L_u}	rate of change of lift coefficient with velocity, $\partial C_L / \partial \frac{u}{V}$
CMUR	$C_{m_{u,R}}$	rate of chage of moment about reference point with velocity, $\partial C_{m,R} / \partial \frac{u}{V}$
S	S	reference area, (Volume of balloon)$^{2/3}$
CBAR	$\bar{c}$	reference length, balloon body length used here
YYOI	I_y	pitching inertia about balloon center of mass (including aerodynamic apparent inertia)
TMASS	m_T	mass of balloon structure and contained gas
AXMASS	$m_{x,a}$	aerodynamic apparent mass in body-reference X-axis direction, $\alpha_t = 0$
AZMASS	$m_{z,a}$	aerodynamic apparent mass in body-reference Z-axis direction, $\alpha_t = 0$
WTS	W_s	structural weight of balloon
BUOY	B	net buoyancy force
BHR	h_{br}	component of distance from reference point to center of buoyancy, positive for center of buoyancy below reference point (see fig. 2)
BLR	l_{br}	component of distance from reference point to center of buoyancy, positive for center of buoyancy forward of reference point (see fig. 2)

FORTRAN variable name	Mathematical symbol	Definition
SHR	h_{sr}	component of distance from reference point to center of mass of balloon structure, positive for center of mass below reference point (see fig. 2)
SLR	l_{sr}	component of distance from reference point to center of mass of balloon structure, positive for center of mass aft of reference point (see fig. 2)
CGH	h_{cg}	component of distance from reference point to center of mass, positive for center of mass below reference point (see fig. 2)
CGL	l_{cg}	component of distance from reference point to center of mass, positive for center of mass forward of reference point (see fig. 2)
TLR	l_{tr}	component of distance from reference point to attachment point of tether line, positive for attachment point forward of reference point (see fig. 2)
TTR	t_{tr}	component of distance from reference point to attachment point of tether line, positive for attachment point below reference point (see fig. 2)
CLC	l_c	length of tether cable
CDIAM	d_c	diameter of tether cable
CDC	C_{D_c}	drag coefficient of tether cable based on diameter, i.e., drag of cable per unit length is $C_{D_c} d_c \rho V^2/2$
WC	w_c	weight per unit length of tether cable
RHO	ρ	ambient air density
VMIN	V_{min}	minimum wind velocity

FORTRAN variable name	Mathematical symbol	Definition
DELV	ΔV	wind-velocity increment
NVEL		number of velocity calculations

Limitations and Diagnostic Messages

The following comments are given to indicate some of the factors that are not treated in the program and to indicate some potential troublesome factors in program operation:

(1) The balloon must lift the tether cable off the ground. No diagnostic messages are given, but the listing of tether conditions will indicate zero cable angle GAMO as the constraint of $0 \leqq \text{GAMO} \leqq \pi/2$ is applied in the program.

(2) The balloon must be able to trim. If the trim angle has not converged to a tolerance of ERR (10^{-6}) in ITCMAX (100) Newton iterations of the trim equation, the message

ITERATION FOR TRIM DID NOT CONVERGE IN ________ ITERATIONS,

ALPHA = ________, TLPHA = ________.

is written on tape 11. The value of TLPHA is used for subsequent calculations.

(3) The z-component of the cable tension acting on the bridle must be directed down for trim. If this condition is not satisfied, GAM1 is erroneous. However, the balloon would normally not be lifting the tether cable off the ground when this limitation would apply.

(4) The bridle is treated as rigid and no consideration is given to the possibility that the bridle lines may go slack.

(5) Cable drag and weight cannot be zero or negative, as these conditions lead to overflows or to a negative number to a real power, both fatal errors. This condition also indicates that the zero wind velocity limit cannot be reached.

(6) The balloon drag must be positive. If there is a tendency for trim angle to diverge with velocity, care must be exercised in fitting C_D versus α such that C_D is always positive.

(7) The minimum velocity that can be treated is about 0.5 m/sec. Loss of significant figures in some of the tether springs and tether conditions may occur at very low velocities. For example, the vertical spring $k_{zz} \rightarrow \infty$ as $V \rightarrow 0$ such that the eigenvalue problem becomes poorly conditioned.

(8) The conditioning of the eigenvalue matrix EIDET is checked. If the Turing's condition number of EIDET exceeds 10^6, indicating an estimated loss of 6 or more of 14 significant figures, the diagnostic message

CONDITION NUMBER OF EIGENVALUE MATRIX = ________.

is written on tape 11 and calculations proceed.

(9) The conditioning of the mass matrix A is checked. The mass matrix here is normally well conditioned. If Turing's condition number of A exceeds 10^4, the diagnostic message

CONDITION NUMBER OF A-MATRIX = ________.

is written on tape 11 and calculations proceed.

(10) The density ρ is input and is considered as a constant both for the cable and for the balloon; thus, the altitude range may be restricted.

(11) The program transfers the stability derivatives from the reference point to the center of mass. However, the center of mass must be computed for input consistent with the structural weight center of mass, the included gas of the balloon, and the aerodynamic apparent masses. The aerodynamic apparent inertia must also be transferred external to this program for input for shifts in center of mass.

(12) It may be noted that the computer running time is closely related to the error tolerances EPS in the iteration and integration procedures used in the tether routines. Here, EPS is generally set to 10^{-8}.

Listing of Input Data Cards for Sample Case

```
********************************* COLUMN NUMBER *************************************
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890

LONGITUDINAL STABILITY OF TETHERED BALLOON - LRC BALLOON-REFERENCE CONFIGURATION
 $LONGDTA VMIN=1., NVEL=51, DELV=1.$
```

```
      OVERLAY(STABLTY,0,0)
      PROGRAM STABLTY(INPUT=1,OUTPUT=1,TAPE5=INPUT,TAPE7,TAPE8=1,
     + TAPE11=1,TAPE30=1,TAPE31=1,TAPE32=1,TAPE33=1)
C***********************************************************************
C*                                                                     *
C* PROGRAM A2864.1 - LONGITUDINAL STABILITY OF TETHERED BALLOON        *
C*                                                                     *
C* PROGRAM READS IDENTIFICATION CARD AND NAMELIST FROM INPUT FILE, AND *
C*   WRITES ONLY THE ID ARRAY FOR EACH CASE ON THE OUTPUT FILE         *
C* ALL FILES ARE BCD AND ARE SET TO MINIMUM BUFFER SIZE, EXCEPT TAPE7  *
C*   WHICH IS BINARY AND USES STANDARD BUFFER SIZE                     *
C* FILE ASSIGNMENTS ARE - TAPE7=PLOTTING PROGRAM INPUT, TAPE8=EIGEN-   *
C*   VECTORS, TAPE11=EIGENVALUES, TAPE30=AERODYNAMIC COEFFICIENTS,     *
C*   TAPE31=TETHER SPRINGS, TAPE32=TETHER CONDITIONS, AND TAPE33=      *
C*   UNCOUPLED ROOTS                                                   *
C*                                                                     *
C***********************************************************************
      COMMON/IROW/IROW(300)/ICOL/ICOL(300)
      DIMENSION A(3,3),B(3,3),C(3,3),SYMBOL(11),ID(10)
      DIMENSION EIDET(6,6),SAVE(6,7),ROOTR(6),ROOTI(6),INDEX(6)
     + ,IRUN(6),P(6),IPIV(3),INDX(3,2)
      COMPLEX EICOEF(3,3),CROT,CRTSQ,CDET
      DATA SYMBOL/110HCIRCLE    SQUARE    DIAMOND   TRIANGLE  RT TRNGL
     +QUADRANT  DOG HOUSE FAN       LNG DMND  HOUSE             /
      DATA RADEG,DELV,NVEL,VMIN/.017453292519943296,.5,104,.5/        SI UNITS
  108 FORMAT(1H1////X10A10///)
  107 FORMAT(12X8G13.5)
  106 FORMAT(/* VELOCITY=*G13.5,2XA10)
  105 FORMAT(50X*EIGENVECTORS*//14X*COMPLEX ROOT-REAL,IMAG*4X
     + *X/THETA,M/DEG-REAL,IMAG*3X*Z/THETA,M/DEG-REAL,IMAG*6X         SI UNITS
     + *THETA,DEG-REAL,IMAG*/)
  104 FORMAT(8X,A10,6G16.6)
  103 FORMAT(/2X7G16.6)
  102 FORMAT(/* CONDITION NUMBER OF EIGENVALUE MATRIX=*E10.2/)
  101 FORMAT(/* CONDITION NUMBER OF A-MATRIX=*E10.2/)
  100 FORMAT(//* VELOCITY,(REAL(ROOT(I)),I=1,NPOS)*/* SYMBOL,(IMAG(ROOT(
     +I)),I=1,NPOS)*/)
      A11= 8H(X10A10) $ A10= 6H(8A10)
C
C  INITIALIZATION SECTION - READ IDENTIFICATION CARD, CALL DAYTIM FOR
C    DATE AND TIME, AND WRITE ID ARRAY ON BCD TAPES 8,11,30,31,32,33,
C    AND BINARY TAPE 7 WITH RECOUT.  DO NON-VELOCITY-DEPENDENT
C    CALCULATIONS WITH A CALL TO INICOEF
C  SEE SUBROUTINE WRITEUP FOR DESCRIPTION OF RECOUT
C
      NMP=3 $ NTWO=NMP+NMP $ NPL1=NTWO+1
      REWIND 30 $ REWIND 31 $ REWIND 32 $ REWIND 33
      REWIND 7 $ REWIND 8 $ REWIND 11
    1 READ A10,(ID(I),I=1,8) $ IF(EOF,5)999,2
    2 CALL DAYTIM(ID(9)) $ PRINT A11,ID $ WRITE(11,108)ID
      WRITE(30,108)ID $ WRITE(31,108)ID $ WRITE(32,108)ID
      WRITE(8,108)ID $ WRITE(8,105) $ WRITE(33,108)ID
      CALL RECOUT(7,2,0,ID,1,10,1)
      CALL INICOEF(A,B,C,NMP,VEL,VMIN,DELV,NVEL) $ WRITE(11,100)
      CALL RECOUT(7,1,0,NVEL)
C
C  90-LOOP IS VELOCITY VARIATION LOOP
C
      DO 90 IV=1,NVEL $ VEL=VMIN+(IV-1.)*DELV
```

```
C   SET UP COEFFICIENT MATRICES FOR QUADRATIC STABILITY DETERMINANT
C     WITH CALL TO ENTRY VCOEF OF SUBROUTINE INICOEF
C
      CALL VCOEF(A,B,C,NMP,VEL,VMIN,DELV,NVEL)
C
C   EXPAND QUADRATIC N X N STABILITY DETERMINANT INTO 2N X 2N STANDARD
C     EIGENVALUE FORM AND CHECK CONDITIONING OF MASS MATRIX A
C
      CALL QUADET(A,B,C, 3, 6,NMP,10,EIDET,CNOA)
      IF(CNOA.GT.1.E+4)WRITE(11,101)CNOA
C
C   EIGENVALUES FOR 2N SYSTEM AND CHECK CONDITIONING OF 2N X 2N MATRIX
C     WITH CALL TO MATRIX FOR INVERSE AND TURING CONDITION NUMBER
C
      CALL REIG(EIDET,NTWO,NTWO,0,ROOTR,ROOTI,EIVEC, 6,INDEX,IRUN,P,
     + NPL1,SAVE)
      CALL MATRIX(10,NTWO,NTWO,0,EIDET, 6,DETEI,K8,CNOEI)
      IF(CNOEI.GT.1.E+6)WRITE(11,102)CNOEI
C
C   ROOT SORTING - SORT COMPLEX ROOTS IN ORDER OF INCREASING MAGNITUDE OF
C     FREQUENCY AND DETERMINE THE NUMBER OF COMPLEX ROOTS WITH POSITIVE
C     VALUE OF FREQUENCY (IMAGINARY PART)
C
      NEGR=1 $ DO 50 NRT=1,NTWO $ NI=NTWO-NRT $ DO 48 J=1,NI
      IF(ROOTI(J)-ROOTI(J+1))48,46,46
   46 TRI=ROOTI(J) $ TRR=ROOTR(J) $ ROOTI(J)=ROOTI(J+1)
      ROOTR(J)=ROOTR(J+1) $ ROOTI(J+1)=TRI $ ROOTR(J+1)=TRR
   48 CONTINUE
   50 CONTINUE $ DO 52 NR=1,NTWO
      IF(ROOTI(NR).LT.-1.E-12)NEGR=NEGR+1
   52 CONTINUE
C
C   WRITE ROOTS ON TAPE 11
C
      IK1=IV/10  $  IK2=11  $  IF(IV.EQ.10*IK1) IK2=1+MOD(IK1-1,10)
      WRITE(11,103)VEL,(ROOTR(N),N=NEGR,NTWO)
      WRITE(11,104)SYMBOL(IK2),(ROOTI(N),N=NEGR,NTWO)
C
C   WRITE RESULTS ON BINARY TAPE 7 FOR INPUT TO PLOTTING PROGRAMS
C
      CALL RECOUT(7,1,0,VEL,IK2,NEGR,NTWO)
      CALL RECOUT(7,2,0,ROOTR,NEGR,NTWO,1)
      CALL RECOUT(7,2,0,ROOTI,NEGR,NTWO,1)
C
C   SETUP COEFFICIENT MATRICES FOR EIGENVECTOR (MODAL RATIOS) BY
C     DIVIDING BY THETA AND CALLING CXINV - RESULTS ON TAPE8
C
      WRITE(8,106)VEL,SYMBOL(IK2)
      DO 70 NE=NEGR,NTWO $ CROT=CMPLX(ROOTR(NE),ROOTI(NE))
      CRTSQ=CROT*CROT $ DO 60 IC=1,2 $ DO 60 IR=1,3
   60 EICOEF(IC,IR)=A(IC,IR)*CRTSQ+B(IC,IR)*CROT+C(IC,IR)
      DO 64 I=1,2
   64 EICOEF(I,3)=-RADEG*EICOEF(I,3)
      CALL CXINV(EICOEF,2,EICOEF(1,3),1,CDET,IPIV,INDX,3,ISC)
      EICOEF(3,3)=(1.,0.)
   70 WRITE(8,107)CROT,(EICOEF(I,3),I=1,3)
C
   90 CONTINUE $ GO TO 1
  999 ENDFILE 7 $ REWIND 7 $ ENDFILE 8 $ REWIND 8 $ REWIND 11
      REWIND 30 $ REWIND 31 $ REWIND 32 $ REWIND 33
      END             PROGRAM STABLTY
```

```
      SUBROUTINE INICOEF(A,B,C,NMAX,VEL,VMIN,DELV,NVEL)
C
C   SUBROUTINE CALCULATES COEFFICIENT MATRICES FOR QUADRATIC STABILITY
C     DETERMINANT
C
      EQUIVALENCE(EQURT(1),UNCRT(1))
      DIMENSION A(NMAX,1),B(NMAX,1),C(NMAX,1),EQURT(1)
      COMPLEX UNCRT(6),CRAD,CSQRT
C
C   INPUT PARAMETERS ARE READ FROM THE INPUT FILE WITH A NAMELIST READ
C     OF THE NAMELIST LONGDTA AND ARE WRITTEN ON TAPE 11 WITH A NAMELIST
C     WRITE STATEMENT
C
      NAMELIST/LONGDTA/CDINS,CLAD,CLQR,CMADR,CMQR,DELCD,DELCDA,DELCL,
     + DELCLA,DELCM,DELCMA,CDU,CLU,CMUR,S,CBAR,YYOI,TMASS,AXMASS,
     + AZMASS,WTS,BUOY,BHR,BLR,SHR,SLR,CGH,CGL,TLR,TTR,CLC,CDIAM,CDC,WC,
     + RHC,VMIN,DELV,NVEL
      COMMON/LCNGDLC/CDINS,DELCD,DELCL,DELCM
C
C   PARAMETERS FOR LRC BALLOON - REFERENCE CONFIGURATION - IN SI UNITS
C
      DATA CDINS,DELCD,DELCDA,DELCL,DELCLA,DELCM,DELCMA,CDU,CLU,CMUR/
     + .010, 9*0./
      DATA DEGRAD/57.2957795130823/
      DATA CMADR,CMQR,CLAD,CLQR/-.026,-.189,.089,.685/
      DATA S,CBAR,YYOI/7.04,7.64,171./                                  SI UNITS
      DATA TMASS,AXMASS,AZMASS,WTS,BUOY/14.2,5.11,23.9,108.,190./       SI UNITS
      DATA BHR,BLR,SLR,TLR,TTR/0.,2.15,-.66,3.44,3.82/                  SI UNITS
      DATA SHR,CGH,CGL/.36,.109,1.10/                                   SI UNITS
      DATA CLC,CDIAM,CDC,WC/61.,.0141,1.17,.343/                        SI UNITS
      DATA RHO/1.225/                                                   SI UNITS
C
C   VELOCITY INDEPENDENT CALCULATIONS
C
      WRITE(30,101) $ WRITE(31,102) $ WRITE(32,103) $ WRITE(33,104)
      READ LONGDTA $ WRITE(11,LONGDTA)
      SL=SLR+CGL $ SH=SHR-CGH $ BL=BLR-CGL
      TL=TLR-CGL $ TT=TTR-CGH $ BH=CGH-BHR
      CBAR2=.5*CBAR $ RI=1./YYOI $ ROSCI=RHO*S*CBAR2*RI
      DELMZA=.5*CBAR2*S*RHO*CLAD
      A(1,2)=A(1,3)=A(2,1)=A(2,3)=A(3,1)=0.
      A(1,1)=A(2,2)=A(3,3)=1. $ RETURN
C
C   ENTRY POINT VCOEF FOR VELOCITY-DEPENDENT CALCULATIONS
C
      ENTRY VCOEF
      Q=.5*RHO*VEL*VEL
C
C   TRIM ANGLE OF ATTACK AND AERODYNAMIC COEFFICIENTS ABOUT THE CENTER
C     OF MASS
C
      CALL TRIM(S,CBAR,WTS,BUOY,BL,BH,SL,SH,TL,TT,CGH,CGL,Q,
     + CL,CM,CD,CLA,CMA,CDA,ALPHA,XOC,ZOC,SINA,COSA)
      ALPHAD=DEGRAD*ALPHA
      CDA=CDA+DELCDA $ CLA=CLA+DELCLA $ CMA=CMA+DELCMA
```

```
C
C  TRANSFER DYNAMIC STABILITY DERIVATIVES FROM REFERENCE POINT (MOMENT
C    CENTER) TO CENTER OF MASS AND WRITE AERODYNAMIC COEFFICIENTS ON
C    TAPE 30
C
      CALL DERTRN(XOC,ZOC,CD,CDU,CL,CLA,CLU,CLAD,CLQR,CM,CMA,CMADR,CMQR,
     + CMUR,CLQ,CMAD,CMQ,CMU)
      WRITE(30,100)VEL,ALPHAD,CD,CL,CM,CDA,CLA,CMA,CLQ,CMAD,CMQ


C
C  CALCULATE EQUILIBRIUM CABLE CONDITIONS AND WRITE RESULTS ON TAPE 32
C
      DRAG=CD*Q*S $ BLIFT=CL*Q*S $ CDRAG=CDC*CDIAM*Q
      T1=SCRT(DRAG*DRAG+(BLIFT-WTS+BUOY)**2)
      COSG1=DRAG/T1 $ GAM1=ACOS(COSG1) $ SING1=SIN(GAM1)
      CALL TETHER(CDRAG,WC,CLC,T1,GAM1,TO,GAMO,X1,Z1)
      GAM1D=DEGRAD*GAM1 $ GAMOD=DEGRAD*GAMO
      COSGO=COS(GAMO) $ SINGO=SIN(GAMO)
      WRITE(32,100)VEL,X1,Z1,GAMOD,TO,GAM1D,T1,CDRAG
C
C  CALCULATE CABLE SPRINGS FROM DERIVATIVES OF NEUMARK AND TRANSFER TO
C    STABILITY AXES - WRITE RESULTS ON TAPE 31
C
      SNG2=SING1*SING1 $ SNG3=SING1*SNG2 $ TSG1=T1*SING1 $ TCG1=T1*COSG1
      RDEL=1./(X1*(SING1-SINGO)-Z1*(COSG1-COSGO)-CLC*SIN(GAM1-GAMO))
      SKXX=RDEL*(TCG1*(SING1-SINGO)+CDRAG*SNG3*(Z1-CLC*SINGO))
      SKXZ=RDEL*(TCG1*(COSGO-COSG1)-CDRAG*SNG3*(X1-CLC*COSGO))
      SKZX=RDEL*(TSG1*(SING1-SINGO)-(WC+CDRAG*SNG2*COSG1)*(Z1-CLC*SINGO)
     + ) $ SKZZ=RDEL*(TSG1*(COSGO-COSG1)+(WC+CDRAG*SNG2*COSG1)*(X1-
     + CLC*COSGO))
      HK1=TL*COSA+TT*SINA $ HK2=TT*COSA-TL*SINA
      SKXT=HK2*SKXX-HK1*SKXZ $ SKTX=HK2*SKXX-HK1*SKZX
      SKZT=HK2*SKZX-HK1*SKZZ $ SKTZ=HK2*SKXZ-HK1*SKZZ
      SKTTD=HK2*(HK2*SKXX-HK1*(SKXZ+SKZX))+HK1*HK1*SKZZ
      SKTTT=HK2*TSG1+HK1*TCG1 $ SKTT=SKTTD+SKTTT
      WRITE(31,100)VEL,SKXX,SKXZ,SKXT,SKZX,SKZZ,SKZT,SKTX,SKTZ,SKTTD,
     + SKTTT
C
C  CALCULATE WEIGHT-BOUYANCY MOMENT TERM AND MASS TERMS INCLUDING
C    APPARENT MASS ROTATION TO STABILITY AXES
C
      SM1=(BL*BUOY+SL*WTS)*SINA+(BH*BUOY+SH*WTS)*COSA
      AMZ=AZMASS*COSA*COSA+AXMASS*SINA*SINA
      AMX=AXMASS*COSA*COSA+AZMASS*SINA*SINA
      RMX=1./(TMASS+AMX) $ RMZ=1./(TMASS+AMZ+DELMZA) $ ROVSCI=VEL*ROSCI
      ROVSMX=.5*RHO*VEL*S*RMX $ ROVSMZ=.5*RHO*VEL*S*RMZ
C
C  CALCULATE COEFFICIENT MATRICES A , B, AND C
C
      A(3,2)=-CBAR*ROSCI*CMAD
      B(1,1)=ROVSMX*(2.*CD+CDU) $ B(1,2)=ROVSMX*(CDA-CL) $ B(1,3)=0.
      B(2,1)=ROVSMZ*(2.*CL+CLU) $ B(2,2)=ROVSMZ*(CLA+CD)
      B(2,3)=CBAR2*ROVSMZ*(CLAD+CLQ) $ B(3,1)=-ROVSCI*(2.*CM+CMU)
      B(3,2)=-ROVSCI*CMA $ B(3,3)=-CBAR2*ROVSCI*(CMAD+CMQ)
      C(1,1)=SKXX*RMX $ C(1,2)=SKXZ*RMX $ C(1,3)=SKXT*RMX+VEL*ROVSMX*CDA
      C(2,1)=SKZX*RMZ $ C(2,2)=SKZZ*RMZ $ C(2,3)=SKZT*RMZ+VEL*ROVSMZ*CLA
      C(3,1)=SKTX*RI  $ C(3,2)=SKTZ*RI
      C(3,3)=RI*(SM1+SKTT)-VEL*ROVSCI*CMA
```

```
C
C  CALCULATE UNCOUPLED ROOTS BY FACTORING DIAGONAL QUADRATIC TERMS AND
C    WRITE RESULTS ON TAPE 33
C
      DO 1 M=1,3 $ CRAD=.25*B(M,M)*B(M,M)-C(M,M) $ CRAD=CSQRT(CRAD)
      M2=2*M $ M1=M2-1 $ UNCRT(M1)=-.5*B(M,M)+CRAD
    1 UNCRT(M2)=-.5*B(M,M)-CRAD
      WRITE(33,105)VEL,(EQURT(I),I=1,11,2),(EQURT(I),I=2,12,2) $ RETURN
C
  100 FORMAT(2X11G11.4)
  101 FORMAT(/20X*AERODYNAMIC COEFFICIENTS*/* VELOCITY*5X*ALPHAD*9X*CD*
     + 9X*CL*9X*CM*8X*CDA*8X*CLA*8X*CMA*7X*CLQ*8X*CMAD*8X*CMQ*)


  102 FORMAT(/20X*TETHER SPRINGS*/* VELOCITY*7X*SKXX*7X*SKXZ*7X*SKXT*7X
     + *SKZX*7X*SKZZ*7X*SKZT*7X*SKTX*7X*SKTZ*5X*SKTT,D*4X*SKTT,T1*)
  103 FORMAT(/20X*TETHER CONDITIONS*/* VELOCITY*9X*X1*9X*Z1*7X*GAM0*9X
     + *T0*7X*GAM1*9X*T1*3X*CAB DRAG*)
  104 FORMAT(/20X*UNCOUPLED ROOTS*/7X*VEL*8X*RLX1*8X*RLX2*8X*RLZ1*8X
     + *RLZ2*8X*RLT1*8X*RLT2*/18X*IMX1*8X*IMX2*8X*IMZ1*8X*IMZ2*8X
     + *IMT1*8X*IMT2*)
  105 FORMAT(/2X7G12.4/14X6G12.4)
      END            SUBROUTINE INICOEF

      SUBROUTINE TRIM(S,CBAR,WTS,BUOY,BL,BH,SL,SH,TL,TT,CGH,CGL,Q,
     + CL,CM,CD,CLA,CMA,CDA,ALPHA,F,G,SA,CA)
C
C  SUBROUTINE COMPUTES THE STATIC TRIM ANGLE-OF-ATTACK  ALPHA  USING
C    NEWTON ITERATION OF THE TRIM EQUATION
C  THE ALPHA DEPENDENT DERIVATIVES CD, CDA, CL, CLA, CM, AND CMA ARE
C    ALSO TRANSFERRED TO THE CENTER OF MASS AND RETURNED
C  IF CONVERGENCE IS NOT OBTAINED IN ITCMAX ITERATIONS, MESSAGE IS
C    WRITTEN ON TAPE 11
C
      COMMON/LONGDLC/CDINS,DELCD,DELCL,DELCM
      ERR=1.E-6 $ TLPHA=.05 $ QS=Q*S $ ITCMAX=100 $ ITC=0
      D=BUOY*BL+WTS*SL $ E=BUOY*BH+WTS*SH
    1 ALPHA=TLPHA $ CL=FCL(ALPHA,CLA) $ CD=FCD(ALPHA,CDA)
      CMR=FCMR(ALPHA,CMAR) $ CA=COS(ALPHA) $ SA=SIN(ALPHA)
      A=TL*CA+TT*SA $ B=TT*CA-TL*SA
      F=(CGL*CA+CGH*SA)/CBAR $ G=(CGH*CA-CGL*SA)/CBAR
      CM=CMR-F*CL+G*CD $ CMA=CMAR-F*(CLA+CD)-G*(CL-CDA)
      CL=CL+DELCL  $  CM=CM+DELCM  $  C=BUOY-WTS+QS*CL
      CD=CD+DELCD+CDINS
      FUN=A*C-(B*CD+CM*CBAR)*QS-D*CA+E*SA
      DFUN=B*C+D*SA+E*CA+(A*(CLA+CD)-B*CDA-CMA*CBAR)*QS
      TLPHA=ALPHA-FUN/DFUN $ ITC=ITC+1 $ IF(ITC.GT.ITCMAX)GO TO 2
      IF(ABS(ALPHA-TLPHA).GT.ERR)1,4
    2 WRITE(11,3)ITC,ALPHA,TLPHA
    3 FORMAT(/*  ITERATION FOR TRIM DID NOT CONVERGE IN*I6* ITERATIONS.
     + ALPHA=*G12.6* TALPHA=*G12.6/)
    4 RETURN
      END            SUBROUTINE TRIM
```

```
      FUNCTION FCD(A,CDA)
C
C   FCD IS A FUNCTION TO BE WRITTEN BY THE USER TO CALCULATE CD AND CDA
C     AS FUNCTIONS OF ANGLE OF ATTACK FOR THE CONFIGURATION TO BE ANALYZED
C
C   CURVE FIT OF CD AND CDA VS ANGLE OF ATTACK IN RADIANS FOR LRC
C     BALLOON - REFERENCE CONFIGURATION
C
      X=A-.023 $ X5=X**5 $ CDA=1117.2*X5
      FCD=.0487+186.2*X*X5  $  RETURN  $  END


      FUNCTION FCL(A,CLA)
C
C   FCL IS A FUNCTION TO BE WRITTEN BY THE USER TO CALCULATE CL AND CLA
C     AS FUNCTIONS OF ANGLE OF ATTACK FOR THE CONFIGURATION TO BE ANALYZED
C
C   CURVE FIT OF CL AND CLA VS ANGLE OF ATTACK IN RADIANS FOR LRC
C     BALLOON - REFERENCE CONFIGURATION
C
      X=A-.023 $ X2=X*X $ CLA=.82-X2*(15.069-557.0*X2)
      FCL=X*(.82-X2*(5.023-111.4*X2))  $  RETURN  $  END


      FUNCTION FCMR(A,CMAR)
C
C   FCMR IS A FUNCTION TO BE WRITTEN BY THE USER TO CALCULATE CMR AND CMAR
C     AS FUNCTIONS OF ANGLE OF ATTACK FOR THE CONFIGURATION TO BE ANALYZED
C
C   CURVE FIT OF CM AND CMA ABOUT REFERENCE POINT VS ANGLE OF ATTACK IN
C     RADIANS FOR LRC BALLOON - REFERENCE CONFIGURATION
C
      CMAR=.1435 $ FCMR=-.0106+.1435*A  $  RETURN  $  END


      SUBROUTINE DERTRN(X,Z,CD,CDU,CL,CLA,CLU,CLAD,CLQR,CM,CMA,CMADR,
     + CMQR,CMUR,CLQ,CMAD,CMQ,CMU)
C
C   TRANSFERS CLQ, CMAD, CMQ, AND CMU FROM REFERENCE POINT (MOMENT CENTER)
C     TO CENTER OF MASS LOCATED X/C FORWARD AND Z/C DOWN FROM REFERENCE
C     POINT
C
      CLQ=CLQR+2.*(X*CLA-Z*(2.*CL+CLU))
      CMU=CMUR+Z*CDU-X*CLU $ CMAD=CMADR-X*CLAD
      CMQ=CMQR-X*(CLQR-2.*CMA)-2.*Z*(2.*CM+CMU) $ RETURN $ END DERTRN
```

```
      SUBROUTINE TETHER(CDRAG,WC,CL1,T1,GAMMA1,TO,GAMMAO,X1,Z1)
C
C  SUBROUTINE IS BASED ON THE ANALYSIS IN -
C    NEUMARK, S.-  EQUILIBRIUM CONFIGURATIONS OF FLYING CABLES OF
C      CAPTIVE BALLOONS, AND CABLE DERIVATIVES FOR STABILITY
C      CALCULATIONS.  BRITISH R. AND M. NO. 3333, 1963.
C  THE TETHER PARAMETERS CDRAG, WC, CL1, T1, AND GAMMA1 ARE INPUT
C  THE TETHER PARAMETERS TO, GAMMAO, X1, AND Z1 ARE OUTPUT
C
      EXTERNAL FLAM,FSIG $ COMMON/PQ/P,Q
      P=.5*WC/CDRAG $ Q=SQRT(1+P*P) $ EPS=1.E-8
      CALL ROMBERG(RLAM1,0.,GAMMA1,FLAM,EPS) $ TAU1=TAU(GAMMA1)
      RLAMO=RLAM1-CDRAG*TAU1*CL1/T1 $ CALL NEWINT(RLAMO,FLAM,1.,GAMMAO)
      CALL ROMBERG(DSIG,GAMMAO,GAMMA1,FSIG,EPS)
      X1=T1*DSIG/(CDRAG*TAU1) $ TAUO=TAU(GAMMAO)
      TO=T1*TAUO/TAU1 $ Z1=(T1-TO)/WC $ RETURN $ END


      FUNCTION FLAM(T) $ COMMON/PQ/P,Q $ CT=COS(T)
      FLAM=((Q+P-CT)/(Q-P+CT))**(P/Q)/(1-CT*CT+2*P*CT) $ RETURN $ END


      FUNCTION FSIG(T) $ COMMON/PQ/P,Q $ CT=COS(T)
      FSIG=((Q+P-CT)/(Q-P+CT))**(P/Q)/(1-CT*CT+2*P*CT)*CT $ RETURN $ END


      FUNCTION TAU(T) $ COMMON/PQ/P,Q $ CT=COS(T)
      TAU=((Q+P-CT)/(Q-P+CT))**(P/Q) $ RETURN $ END


      SUBROUTINE NEWINT(C,F,XO,X) $ EPS=1.E-8 $ XT=XO $ I=0
C
C  SUBROUTINE COMPUTES THE UPPER LIMIT  X  OF THE DEFINITE INTEGRAL
C    FROM  0 TO X  OF THE FUNCTION  F  FOR WHICH THE VALUE OF THE
C    INTEGRAL  C  IS KNOWN
C  NEWTON ITERATION IS USED WITH THE VALUE OF THE INTEGRAL DETERMINED
C    BY SUBROUTINE ROMBERG
C
    1 X=XT $ CALL ROMBERG(S,0.,X,F,EPS) $ XT=X+(C-S)/F(X) $ I=I+1
      PI2=1.570796326 $ IF(XT.GT.PI2) XT=PI2 $ IF(XT.LT.0.) XT=0.
      IF(I.LT.2) GO TO 1 $ IF(ABS(X-XT).GT.EPS) GO TO 1
      X=XT  $  RETURN
      END            SUBROUTINE NEWINT
```

```
      SUBROUTINE ROMBERG(SUM,A,B,FUN,EPS) $ DIMENSION Q(20)
***********************************************************************
*  SUBROUTINE FOR ROMBERG QUADRATURE - SEE WRITEUP FOR DESCRIPTION    *
***********************************************************************
      SUM=0 $ IF(A.EQ.B) RETURN
      H=B-A $ FA=FUN(A) $ FB=FUN(B) $ FM=AMAX1(ABS(FA),ABS(FB))
      T=.5*H*(FA+FB) $ NX=1 $ DO 5 N=1,19 $ H=.5*H $ SUM=0 $ I=0
    1 I=I+1 $ FX=FUN(A+H*(I+I-1)) $ IF(ABS(FX).GT.FM) FM=ABS(FX)
      SUM=SUM+FX $ IF(I.LT.NX) GO TO 1
      T=.5*T+H*SUM $ Q(N)=.666666666666667*(T+H*SUM)
      IF(N.LT.2) GO TO 4 $ F=4. $ DO 2 J=2,N $ I=N+1-J $ F=4.*F
    2 Q(I)=Q(I+1)+(Q(I+1)-Q(I))/(F-1.) $ IF(N.LT.3) GO TO 3
      X=ABS(Q(1)-QX2)+ABS(QX2-QX1) $ IF(X.LE.2.*EPS*FM*ABS(B-A)) GO TO 6
    3 QX1=QX2
    4 QX2=Q(1)
    5 NX=NX+NX
    6 SUM=Q(1) $ RETURN
      END            SUBROUTINE ROMBERG


      SUBROUTINE QUADET(A,B,C,NMAX,NMAX2,N,IOP,EIGDET,CNO)
C
C  SEE SUBROUTINE WRITEUP FOR DESCRIPTION
C
      COMMON/IROW/IROW(300)/ICOL/ICOL(300)
      DIMENSION KR(7),A(NMAX,1),B(NMAX,1),C(NMAX,1),EIGDET(NMAX2,1)
      AN=0. $ AINV=0. $ RINDF=01777000000000001777
      KR(1)=IOP $ KR(4)=0  $ KR(2)=KR(5)=N $ KR(3)=3*N $NSQ=N*N
      NS1=NMAX2*NMAX2-NSQ  $ NS2=NS1-NSQ $ NS3=NS2-NSQ $ NS3P1=NS3+1
      DO 1 J=1,N $ DO 1 I=1,N $ IE=I+N*J-N $ EIGDET(NS3+IE)=A(I,J)
      EIGDET(NS2+IE)=B(I,J)$ EIGDET(NS1+IE)=C(I,J)
    1 AN=AN+A(I,J)*A(I,J)  $ CALL MASCNT(KR,EIGDET(NS3P1),DET,CC)
      IF(DET.NE.0.)GO TO 2 $ CNO=-1. $ RETURN
    2 IF(IOP.NE.10)GO TO 4 $ DO 3 J=1,NSQ
    3 AINV=AINV+EIGDET(NS3+J)*EIGDET(NS3+J) $ CNO=SQRT(AN*AINV/NSQ)
    4 DO 5 J=1,N $ DO 5 I=1,N $ IE=I+N*J-N
    5 EIGDET(I,J)=-EIGDET(NS2+IE)$ DO 6 J=1,N $ DO 6 I=1,N $ IE=I+N*J-N
    6 EIGDET(I,J+N)=-EIGDET(NS1+IE) $ DO 7 J=1,NMAX2 $ DO 7 I=1,N
    7 EIGDET(I+N,J)=0. $ DO 8 J=1,N
    8 EIGDET(N+J,J)=1. $ IF(N.EQ.NMAX)RETURN
      N2P1=N+N+1 $ DO 9 I=N2P1,NMAX2 $ DO 9 J=1,NMAX2 $EIGDET(J,I)=RINDF
    9 EIGDET(I,J)=RINDF  $  RETURN
      END            SUBROUTINE QUADET


      SUBROUTINE MATRIX(I,M,N,K,A,KA,B,KB,C,KC)
C
C  SEE SUBROUTINE WRITEUP FOR DESCRIPTION
C
      COMMON/IROW/IROW(300)/ICOL/ICOL(300) $ DIMENSION A(1),B(1),KR(7)
      KR(1)=I $ IF(I.NE.10)GO TO 2 $ S=0 $ DO 1 L=1,M $ DO 1 J=1,M
    1 S=S+A(L+J*KA-KA)*A(L+J*KA-KA) $ T=0
    2 KR(2)=M $ KR(3)=N $ KR(4)=K $ KR(5)=KA $ KR(6)=KB $ KR(7)=KC
      CALL MASCNT(KR,A,B,C) $ IF(I.NE.10)GO TO 4 $ DO3 L=1,M $DO3 J=1,M
    3 T=T+A(L+J*KA-KA)*A(L+J*KA-KA) $ C=SQRT(S*T)/M
    4 RETURN
      END            SUBROUTINE MATRIX
```

```
      SUBROUTINE REIG (A,M,NVAL,NVEC,ROOTR,ROOTI,VEC,MAX,IDX,IRN,P,NP,
     1 SAVE)
C        PROGRAM TO CALL QR TRANSFORMATION, VARIABLE DIMENSION
C        MAXIMUM ITER IS 50
C
C     THIS VERSION OF REIG HAS BEEN ALTERED SUCH THAT EIGENVECTORS ARE
C     NOT CALCULATED.  SUBROUTINE VECTOR HAS BEEN DELETED AND VEC DOES
C     NOT HAVE TO BE DIMENSIONED
C
C  SEE SUBROUTINE WRITEUP FOR DESCRIPTION
C
      DIMENSION A(MAX,MAX),ROOTR(MAX),ROOTI(MAX),VEC(MAX,MAX),IDX(MAX),
     1 IRN(MAX),P(MAX),SAVE(MAX,NP)
      REAL IDX,IRN
C
      N = M
C        SAVE ORIGINAL MATRIX,  RESTORE AT 200
      DO 5 I=1,M
      DO 5 J=1,M
    5 SAVE(J,I) = A(J,I)
C        REDUCE MATRIX TO HESSENBERG FORM
      CALL HESSEN (A,M,MAX)
      ZERC = 0.0
      JJ = 1
  177 XNN = 0.0
      XN2 = 0.0
      AA = 0.0
      B = 0.0
      C = 0.0
      DD = 0.0
      R = 0.0
      SIG = 0.0
      ITER=0
      IF (N-2) 13,14,12
   13 ROOTR(1) = A(1,1)
      ROOTI(1) = 0.0
      GO TO 200
   14 JJ= -1
   12 X = (A(N-1,N-1) - A(N,N))**2
      S = 4.0*A(N,N-1)*A(N-1,N)
      ITER = ITER + 1
      IF (X .EQ. 0.0) GO TO 15
      IF (ABS(S/X) .GT. 1.0E-8) GO TO 15
      IF (ABS(A(N-1,N-1)) - ABS(A(N,N))) 32,32,31
   31 E = A(N-1,N-1)
      G = A(N,N)
      GO TO 33
   32 G = A(N-1,N-1)
      E = A(N,N)
   33 F = C.0
      H = 0.0
      GO TO 24
   15 S = X + S
      X = A(N-1,N-1) + A(N,N)
      SQ = SQRT(ABS(S))
      IF (S) 18,19,19
   19 F = 0.0
      H = 0.0
      IF (X) 21,21,22
```

```
 21 E = (X-SQ)/2.0
    G = (X+SQ)/2.0
    GO TO 24
 22 G = (X-SQ)/2.0
    E = (X+SQ)/2.0
    GO TO 24
 18 F = SQ/2.0
    E = X/2.0
    G = E
    H = -F
 24 IF (JJ .LT. 0) GO TO 28
    D = 1.0E-10*(ABS(G) + F)
    IF (ABS(A(N-1,N-2)) .GT. D) GO TO 26
 28 ROOTR(N) = E
    ROOTI(N) = F
    ROOTR(N-1) = G
    ROOTI(N-1) = H
    N = N-2
    IF (JJ) 200,177,177
 26 IF(ABS(A(N,N-1)).GT. 1.0E-10*ABS(A(N,N))) GO TO 50
 29 ROOTR(N) = A(N,N)
    ROOTI(N) = 0.0
    N = N-1
    GO TO 177
 50 IF (ABS(ABS(XNN/A(N,N-1))-1.0)-1.0E-6) 63,63,62
 62 IF (ABS(ABS(XN2/A(N-1,N-2))-1.0)-1.0E-6) 63,63,700
 63 VQ = ABS(A(N,N-1)) - ABS(A(N-1,N-2))
    IF (ITER-15) 53,164,64
164 IF (VQ) 165,165,166
165 R = A(N-1,N-2)**2
    SIG = 2.0*A(N-1,N-2)
    GO TO 60
166 R = A(N,N-1)**2
    SIG = 2.0*A(N,N-1)
    GO TO 60
 64 IF (VQ) 29,29,28
700 IF (ITER .GT. 50) GO TO 63
    IF (ITER .GT. 5 ) GO TO 53
    Z1 = ((E-AA)**2+(F-B)**2)/(E*E+F*F)
    Z2 = ((G-C)**2+(H-DD)**2)/(G*G+H*H)
    IF (Z1-0.25) 51,51,52
 51 IF (Z2-0.25) 53,53,54
 53 R=E*G-F*H
    SIG =E+G
    GO TO 60
 54 R=E*E
    SIG =E+E
    GO TO 60
 52 IF (Z2-0.25) 55,55,601
 55 R=G*G
    SIG=G+G
    GO TO 60
601 R = 0.0
    SIG = 0.0
 60 XNN = A(N,N-1)
    XN2 = A(N-1,N-2)
    CALL QRT (A,N,R,SIG,D,MAX)
    AA = E
    B = F
    C = G
    DD = H
    GO TO 12
```

```
C         RESTORE MATRIX                                                REIG1140
  200 DO 210 J=1,M                                                      REIG1150
      DO 210 I=1,M                                                      REIG1160
  210 A(I,J) = SAVE(I,J)                                                REIG1170
C         TEST FOR COMPLEX ROOTS                                        REIG1180
      N = 0                                                             REIG1190
      NC = M+1                                                          REIG1200
      DO 225 I=1,M                                                      REIG1210
      IF (ROOTR(I) .EQ. 0.) GO TO 212                                   REIG1220
      IF (ABS(ROOTI(I)/ROOTR(I)) .GE. 1.E-12) GO TO 215                 REIG1230
  212 IF (ABS(ROOTI(I)) .LT. 1.E-12) GO TO 218                          REIG1240
C         INDEX FOR COMPLEX (END OF ARRAYS)                             REIG1250
  215 NC = NC-1                                                         REIG1260
      JM = NC                                                           REIG1270
      GO TO 220                                                         REIG1280
C         INDEX FOR REAL ROOTS SAME, N = NO. REAL ROOTS                 REIG1290
  218 N = N+1                                                           REIG1300
      JM = N                                                            REIG1310
  220 IRN(JM) = ROOTR(I)                                                REIG1320
  225 IDX(JM) = ROOTI(I)                                                REIG1330
C         REAL ROOTS IN DESCENDING ORDER BY MAGNITUDE                   REIG1340
      DO 240 I=1,M                                                      REIG1350
      IF (I .GE. N) GO TO 235                                           REIG1360
      K = I+1                                                           REIG1370
      DO 230 J=K,N                                                      REIG1380
      IF (ABS(IRN(I)) .GE. ABS(IRN(J))) GO TO 230                       REIG1390
      SIG = IRN(J)                                                      REIG1400
      IRN(J) = IRN(I)                                                   REIG1410
      IRN(I) = SIG                                                      REIG1420
  230 CONTINUE                                                          REIG1430
  235 ROOTR(I) = IRN(I)                                                 REIG1440
      ROOTI(I) = IDX(I)                                                 REIG1450
C         STORE ZERO IN VECTOR                                          REIG1460
C         STORE ZERO IN VECTOR DELETED FROM THIS VERSION
C     DO 240 J=1,M                                                      REIG1470
C 240 VEC(J,I) = 0.0                                                    REIG1480
  240 CONTINUE
      IF (NVEC .LT. N) N = NVEC                                         REIG1490
      IF (N .LE. 0) GO TO 250                                           REIG1500
      DO 245 I = 1,N                                                    REIG1510
      K = N+1-I                                                         REIG1520
      IF (ABS(ROOTR(K)/ROOTR(1)) .GT.1.E-14) GO TO 248                  REIG1530
  245 ROOTR(K) =0.                                                      REIG1540
  248 CONTINUE                                                          REIG1541
C         CALL ROUTINE FOR N VECTORS                                    REIG1550
C         CALL ROUTINE FOR N VECTORS DELETED FROM THIS VERSION
C     CALL VECTOR (IDX,IRN,ROOTR,A,VEC,M,SAVE,P,NP,MAX,N)               REIG1560
  250 RETURN                                                            REIG1570
       END                                                              REIG1580
```

```
      SUBROUTINE QRT(A,N,R,SIG,D,MAX)                                   *QRT0001
      DIMENSION  A(MAX,MAX),PSI(2),G(3)                                 *QRT0002
      N1 = N - 1                                                        *QRT0003
      IA = N - 2                                                        *QRT0004
      IP = IA                                                           *QRT0005
      IF(N-3) 101,10,60                                                 *QRT0006
60    DO 12 J = 3,N1                                                    *QRT0007
      J1 = N - J                                                        *QRT0008
      IF(ABS(A(J1+1,J1))-D)     10,10,11                                *QRT0009
   11 DEN = A(J1+1,J1+1)*(A(J1+1,J1+1)-SIG)+A(J1+1,J1+2)*A(J1+2,J1+1)+R *QRT0010
      IF(DEN) 61,12,61                                                  *QRT0011
61    IF(ABS(A(J1+1,J1)*A(J1+2,J1+1)*(ABS(A(J1+1,J1+1)+A(J1+2,J1+2)    *QRT0012
     1-SIG)+ABS(A(J1+3,J1+2)))/DEN)-D)  10,10,12                        *QRT0013
12    IP=J1                                                             *QRT0014
10    DO  14  J=1,IP                                                    *QRT0015
      J1=IP-J+1                                                         *QRT0016
      IF(ABS(A(J1+1,J1))-D)     13,13,14                                *QRT0017
14    IQ=J1                                                             *QRT0018
13    DO  100  I=IP,N1                                                  *QRT0019
      IF(I-IP)    16,15,16                                              *QRT0020
15    G(1)=A(IP,IP)*(A(IP,IP)-SIG)+A(IP,IP+1)*A(IP+1,IP)+R              *QRT0021
      G(2)=A(IP+1,IP)*(A(IP,IP)+A(IP+1,IP+1)-SIG)                       *QRT0022
      G(3)=A(IP+1,IP)*A(IP+2,IP+1)                                      *QRT0023
      A(IP+2,IP)=0.0                                                    *QRT0024
      GO TO 19                                                          *QRT0025
16    G(1)=A(I,I-1)                                                     *QRT0026
      G(2)=A(I+1,I-1)                                                   *QRT0027
      IF(I-IA)    17,17,18                                              *QRT0028
17    G(3)=A(I+2,I-1)                                                   *QRT0029
      GO TO 19                                                          *QRT0030
18    G(3)=0.0                                                          *QRT0031
   19 XK = SIGN(SQRT(G(1)**2 + G(2)**2 + G(3)**2), G(1))                *QRT0032
22    IF(XK)    23,24,23                                                *QRT0033
23    AL=G(1)/XK+1.0                                                    *QRT0034
      PSI(1)=G(2)/(G(1)+XK)                                             *QRT0035
      PSI(2)=G(3)/(G(1)+XK)                                             *QRT0036
      GO TO 25                                                          *QRT0037
24    AL=2.0                                                            *QRT0038
      PSI(1)=0.0                                                        *QRT0039
      PSI(2)=0.0                                                        *QRT0040
25    IF(I-IQ)    26,27,26                                              *QRT0041
26    IF(I-IP)     29,28,29                                             *QRT0042
28    A(I,I-1)=-A(I,I-1)                                                *QRT0043
      GO TO 27                                                          *QRT0044
29    A(I,I-1)=-XK                                                      *QRT0045
27    DO 30  J=I,N                                                      *QRT0046
      IF(I-IA)  31,31,32                                                *QRT0047
31    C=PSI(2)*A(I+2,J)                                                 *QRT0048
      GO TO 33                                                          *QRT0049
32    C=0.0                                                             *QRT0050
33    E=AL*(A(I,J)+PSI(1)*A(I+1,J)+C)                                   *QRT0051
      A(I,J)=A(I,J)-E                                                   *QRT0052
      A(I+1,J)=A(I+1,J)-PSI(1)*E                                        *QRT0053
      IF(I-IA)    34,34,30                                              *QRT0054
34    A(I+2,J)=A(I+2,J)-PSI(2)*E                                        *QRT0055
30    CONTINUE                                                          *QRT0056
      IF(I-IA)    35,35,36                                              *QRT0057
35    L=I+2                                                             *QRT0058
      GO TO  37                                                         *QRT0059
36    L=N                                                               *QRT0060
37    DO  40  J=IQ,L                                                    *QRT0061
```

```
      IF(I-IA)  38,38,39                                                 *QRT0062
 38   C=PSI(2)*A(J,I+2)                                                  *QRT0063
      GO TO  41                                                          *QRT0064
 39   C=0.0                                                              *QRT0065
 41   E=AL*(A(J,I)+PSI(1)*A(J,I+1)+C)                                    *QRT0066
      A(J,I)=A(J,I)-E                                                    *QRT0067
      A(J,I+1)=A(J,I+1)-PSI(1)*E                                         *QRT0068
      IF(I-IA)  42,42,40                                                 *QRT0069
 42   A(J,I+2)=A(J,I+2)-PSI(2)*E                                         *QRT0070
 40   CONTINUE                                                           *QRT0071
      IF(I-N+3)   43,43,100                                              *QRT0072
 43   E=AL*PSI(2)*A(I+3,I+2)                                             *QRT0073
      A(I+3,I)=-E                                                        *QRT0074
      A(I+3,I+1)=-PSI(1)*E                                               *QRT0075
      A(I+3,I+2)=A(I+3,I+2)-PSI(2)*E                                     *QRT0076
 100  CONTINUE                                                           *QRT0077
 101  RETURN                                                             *QRT0078
       END                                                               *QRT0079


      SUBROUTINE HESSEN(A,M,MAX)                                         HESS0002
C     SUBROUTINE TO PUT MATRIX IN UPPER HESSENBERG FORM.                 HESS0001
      DIMENSION A(MAX,MAX), B(99)                                        HESS0003
      DOUBLE PRECISION SUM                                               HESS0004
      IF (M - 2) 30,30,32                                                HESS0005
   32 DO 40 LC = 3,M                                                     HESS0006
      N = M - LC + 3                                                     HESS0007
      N1 = N - 1                                                         HESS0008
      N2 = N - 2                                                         HESS0009
      NI = N1                                                            HESS0010
      DIV = ABS(A(N,N-1))                                                HESS0011
      DO 2  J = 1,N2                                                     HESS0012
      IF(ABS(A(N,J))- DIV) 2,2,1                                         HESS0013
    1 NI = J                                                             HESS0014
      DIV = ABS(A(N,J))                                                  HESS0015
    2 CONTINUE                                                           HESS0016
      IF(DIV) 3,40,3                                                     HESS0017
    3 IF(NI - N1) 4, 7,4                                                 HESS0018
    4 DO 5 J = 1,N                                                       HESS0019
      DIV = A(J,NI)                                                      HESS0020
      A(J,NI) = A(J,N1)                                                  HESS0021
    5 A(J,N1) = DIV                                                      HESS0022
      DO 6  J = 1,M                                                      HESS0023
      DIV = A(NI,J)                                                      HESS0024
      A(NI,J) = A(N1,J)                                                  HESS0025
    6 A(N1,J) = DIV                                                      HESS0026
    7 DO 26 K = 1, N1                                                    HESS0027
   26 B(K) = A(N,K)/A(N,N-1)                                             HESS0028
      DO 45 J = 1,M                                                      HESS0029
      SUM = 0.0                                                          HESS0030
      IF (J - N1) 46,43,43                                               HESS0031
   46 IF(B(J))  41,43,41                                                 HESS0032
   41 A(N,J) = 0.0                                                       HESS0033
      DO 42 K = 1,N1                                                     HESS0034
      A(K,J) = A(K,J) - A(K,N1)*B(J)                                     HESS0035
   42 SUM = SUM + A(K,J)*B(K)                                            HESS0036
      GO TO 45                                                           HESS0037
   43 DO 44 K = 1,N1                                                     HESS0038
   44 SUM = SUM + A(K,J)*B(K)                                            HESS0039
   45 A(N1,J) = SUM                                                      HESS0040
   40 CONTINUE                                                           HESS0041
   30 RETURN                                                             HESS0042
      END                                                                HESS0043
```

```
      SUBROUTINE CXINV(A,N,B,M,DET,IPIV,INDX,MAX,ISCALE)
C
C        COMPLEX MATRIX INVERSION WITH SOLUTION OF LINEAR EQUATIONS
C
C        CAVM = CABS(A(MAX)),   CAVA = CABS(A(I,J))
C        CADM = CABS(DETERM),   CAPV = CABS(PIVOT)
C
      COMPLEX A(MAX,N), B(MAX,M), SWAP, DET, PIV, PIVI, CO, C1
      DIMENSION IPIV(N), INDX(MAX,2)
C
C        CONSTANTS, INITIALIZATION
C
      CO = (0.0,0.0)
      C1 = (1.0,0.0)
      ISCALE = 0
      RL = 10.0**100
      RS = 1.0/RL
      DET = C1
      CADM = 1.0
      DO 20 J=1,N
   20 IPIV(J) = 0
      DO 500 I=1,N
C
C        SEARCH FOR PIVOT ELEMENT
C
      CAVM = 0.0
      DO 105 J=1,N
      IF (IPIV(J) .EQ. 1) GO TO 105
      DO 100 K=1,N
      IF (IPIV(K) - 1) 50,100,750
   50 CONTINUE
      CAVA = CABS(A(J,K))
      IF (CAVM .GE. CAVA) GO TO 100
      IROW = J
      ICOL = K
      CAVM = CAVA
  100 CONTINUE
  105 CONTINUE
      IF (CAVM .EQ. 0.0) GO TO 720
      IPIV(ICOL) = IPIV(ICOL) + 1
C
C        INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL
C
      IF (IROW .EQ. ICOL) GO TO 230
      DET = -DET
      DO 200 L=1,N
      SWAP = A(IROW,L)
      A(IROW,L) = A(ICOL,L)
      A(ICOL,L) = SWAP
  200 CONTINUE
      IF (M .LE. 0) GO TO 230
      DO 220 L=1,M
      SWAP = B(IROW,L)
      B(IROW,L) = B(ICOL,L)
      B(ICOL,L) = SWAP
  220 CONTINUE
  230 CONTINUE
      INDX(I,1) = IROW
      INDX(I,2) = ICOL
      PIV = A(ICOL,ICOL)
```

```
      CAPV = CABS(PIV)
      IF (CAPV .EQ. 0.0) GO TO 720
C
C        SCALE DETERMINANT
C
      PIVI = PIV
      CADM = CABS(DET)
      IF (CADM  .LT. RL ) GO TO 260
      DET = DET/RL
      CADM = CABS(DET)
      ISCALE = ISCALE + 1
      IF (CADM .LT. RL) GO TO 290
      DET = DET/RL
      ISCALE = ISCALE + 1
      GO TO 290
  260 CONTINUE
      IF (CADM .GT. RS) GO TO 290
      DET = DET*RL
      CADM = CABS(DET)
      ISCALE = ISCALE - 1
      IF (CADM .GT. RS) GO TO 290
      DET = DET*RL
      ISCALE = ISCALE - 1
  290 CONTINUE
      CAPV = CABS(PIVI)
      IF (CAPV .LT. RL) GO TO 320
      PIVI = PIVI/RL
      CAPV = CABS(PIVI)
      ISCALE = ISCALE + 1
      IF (CAPV .LT. RL) GO TO 340
      PIVI = PIVI/RL
      ISCALE = ISCALE + 1
      GO TO 340
  320 CONTINUE
      IF (CAPV .GT. RS) GO TO 340
      PIVI = PIVI*RL
      CAPV = CABS(PIVI)
      ISCALE = ISCALE - 1
      IF (CAPV .GT. RS) GO TO 340
      PIVI = PIVI*RL
      ISCALE = ISCALE - 1
  340 CONTINUE
      DET = DET * PIVI
C
C        CIVIDE PIVOT ROW BY PIVOT ELEMENT
C
      A(ICCL,ICOL) = C1
      DO 350 L=1,N
  350 A(ICOL,L) = A(ICOL,L)/PIV
      IF (M .LE. 0) GO TO 380
      DO 370 L=1,M
  370 B(ICCL,L) = B(ICOL,L)/PIV
C
C        REDUCE NON-PIVOT ROWS
C
  380 CONTINUE
      DO 500 L1=1,N
      IF (L1 .EQ. ICOL) GO TO 500
      SWAP = A(L1,ICOL)
      A(L1,ICOL) = C0
      DO 400 L=1,N
```

```
  400 A(L1,L) = A(L1,L) - A(ICOL,L)*SWAP
      IF (M .LE. 0) GO TO 500
      DO 450 L=1,M
  450 B(L1,L) = B(L1,L) - B(ICOL,L)*SWAP
  500 CONTINUE
C
C           INTERCHANGE COLUMNS
C
      DO 700 I=1,N
      L = N+1-I
      IF (INDX(L,1) .EQ. INDX(L,2))GO TO 700
      IROW = INCX(L,1)
      ICOL = INDX(L,2)
      DO 690 K=1,N
      SWAP = A(K,IROW)
      A(K,IROW) = A(K,ICOL)
      A(K,ICOL) = SWAP
  690 CCNTINUE
  700 CONTINUE
      GO TC 750
  720 DET = CO
      ISCALE = 0
  750 RETLRN
      END
```

Printing of Files Containing Calculated Results for Sample Case

A printing of each BCD file for the sample case is given. The headings essentially give the quantities printed using the FORTRAN variable names as previously given. The files and their contents are as follows:

File name	Content
OUTPUT	case identification, date, and time only
TAPE11	printing of input data NAMELIST and the NPOS dimensional (1/sec) characteristic roots with $\mathrm{Im}(\lambda) \geqq 0$ (NPOS = NTWO - NEGR + 1); velocity is printed followed by $\mathrm{Re}(\lambda)$, SYMBOL, and $\mathrm{Im}(\lambda)$
TAPE8	each characteristic root (with $\mathrm{Im}(\lambda) \geqq 0$) and the corresponding eigenvector is printed for every value of velocity
TAPE30	velocity, trim angle of attack, and the aerodynamic coefficients about the center of mass
TAPE31	dimensional tether derivatives or spring constants about the center of mass
TAPE32	dimensional balloon position and tether conditions (note that CAB DRAG = n)
TAPE33	the uncoupled roots of the x-, z-, and θ-equations are calculated by factoring the diagonal quadratic factors of the stability determinant and are associated with the FORTRAN variable name UNCRT; both roots are printed (even if $\mathrm{Im}(\lambda) < 0$); the headings RLX1 and IMX1 denote the real and imaginary parts of the uncoupled roots associated with the x-equation, etc., where the letters X, Z, and T in the headings indicate x-, z-, and θ-equations, respectively

Listing of OUTPUT for sample case (identification card, and date and time only).-

LONGITUDINAL STABILITY OF TETHERED BALLOON - LRC BALLOON-REFERENCE CONFIGURATION 11/20/72 09.56.50.

Listing of tape 11 for sample case (principal file for characteristic roots).-

```
$LONGOTA
CDINS   =  0.1E-01,
CLAD    =  0.89E-01,
CLQR    =  0.685E+00,
CMADR   = -0.26E-01,
CMQR    = -0.189E+00,
DELCD   =  0.0,
DELCDA  =  0.0,
DELCL   =  0.0,
DELCLA  =  0.0,
DELCM   =  0.0,
DELCMA  =  0.0,
CDU     =  0.0,
CLU     =  0.0,
CMUR    =  0.0,
S       =  0.704E+01,
CBAR    =  0.764E+01,
YYDI    =  0.171E+03,
TMASS   =  0.142E+02,
AXMASS  =  0.511E+01,
AZMASS  =  0.239E+02,
WTS     =  0.108E+03,
BUOY    =  0.19E+03,
BHR     =  0.0,
BLR     =  0.215E+01,
SHR     =  0.38E+00,
SLR     = -0.66E+00,
CGH     =  0.109E+00,
CGL     =  0.11E+01,
TLR     =  0.344E+01,
TTR     =  0.382E+01,
CLC     =  0.61E+02,
CDIAM   =  0.141E-01,
CDC     =  0.117E+01,
WC      =  0.343E+00,
RHO     =  0.1225E+01,
VMIN    =  0.1E+01,
DELV    =  0.1E+01,
NVEL    =  51,
$END
```

VELOCITY,(REAL(ROOT(I)),I=1,NPOS)
SYMBOL,(IMAG(ROOT(I)),I=1,NPOS)

1.00000	-1.269233E-02	-.107410	-4.111921E-02	
	.238376	.854283	94.1124	
2.00000	-2.519945E-02	-.216827	-7.993167E-02	
	.239258	.887273	23.7948	
3.00000	-3.815002E-02	-.329181	-.114988	
	.241742	.937637	10.8043	
4.00000	-5.281444E-02	-.444374	-.145607	
	.246732	.999484	6.29160	
5.00000	-7.054334E-02	-.562925	-.170315	
	.254893	1.06604	4.23316	
6.00000	-9.246417E-02	-.688306	-.184629	
	.266691	1.12903	3.13920	
7.00000	-.119482	-.825481	-.182608	
	.282599	1.17387	2.50507	
8.00000	-.152479	-.965847	-.171834	
	.303351	1.17858	2.13308	
9.00000	-.192646	-1.08371	-.176679	
	.330302	1.13870	1.92081	
10.0000	-.241991	-1.16736	-.206730	
CIRCLE	.366181	1.06449	1.79782	
11.0000	-.304131	-1.21527	-.259838	
	.417378	.950124	1.72567	
12.0000	-.383470	-1.22296	-.336016	
	.503816	.753418	1.68766	
13.0000	-1.23563	-.432001	-.438046	
	.299847	.679145	1.68394	
14.0000	-.806029	-1.90890	-.354069	-.557382
	0.	0.	.833818	1.72691
15.0000	-.717823	-2.28222	-.264867	-.667258
	0.	0.	.908994	1.81445
16.0000	-.694594	-2.58532	-.197329	-.758069
	0.	0.	.951976	1.92471
17.0000	-.695379	-2.85708	-.146998	-.835324
	0.	0.	.984365	2.04516
18.0000	-.708337	-3.11064	-.107909	-.904329
	0.	0.	1.01347	2.17151
19.0000	-.728525	-3.35222	-7.625469E-02	-.968251
	0.	0.	1.04164	2.30225
20.0000	-.753431	-3.58529	-4.972450E-02	-1.02892
SQUARE	0.	0.	1.07045	2.43669
21.0000	-.781609	-3.81199	-2.686880E-02	-1.08745
	0.	0.	1.09967	2.57437
22.0000	-.812151	-4.03376	-6.740318E-03	-1.14452
	0.	0.	1.12960	2.71495
23.0000	-.844451	-4.25159	1.130491E-02	-1.20057
	0.	0.	1.16027	2.85814
24.0000	-.878089	-4.46621	2.771996E-02	-1.25592
	0.	0.	1.19165	3.00367
25.0000	-.912761	-4.67817	4.283317E-02	-1.31075
	0.	0.	1.22369	3.15128
26.0000	-.948242	-4.88792	5.688874E-02	-1.36523
	0.	0.	1.25634	3.30075

27.0000	-.984363	-5.09577	7.007230E-02	-1.41944
	0.	0.	1.28954	3.45187
28.0000	-1.02099	-5.30202	8.252779E-02	-1.47346
	0.	0.	1.32324	3.60444
29.0000	-1.05803	-5.50688	9.436869E-02	-1.52734
	0.	0.	1.35739	3.75828
30.0000	-1.09540	-5.71054	.105686	-1.58112
DIAMOND	0.	0.	1.39196	3.91325
31.0000	-1.13304	-5.91315	.116553	-1.63484
	0.	0.	1.42691	4.06919
32.0000	-1.17089	-6.11485	.127030	-1.68850
	0.	0.	1.46220	4.22598
33.0000	-1.20892	-6.31574	.137167	-1.74213
	0.	0.	1.49780	4.38350
34.0000	-1.24709	-6.51593	.147007	-1.79574
	0.	0.	1.53368	4.54166
35.0000	-1.28538	-6.71549	.156584	-1.84933
	0.	0.	1.56982	4.70035
36.0000	-1.32376	-6.91450	.165928	-1.90291
	0.	0.	1.60621	4.85950
37.0000	-1.36222	-7.11301	.175064	-1.95649
	0.	0.	1.64281	5.01904
38.0000	-1.40074	-7.31108	.184016	-2.01007
	0.	0.	1.67962	5.17891
39.0000	-1.43931	-7.50876	.192802	-2.06364
	0.	0.	1.71661	5.33904
40.0000	-1.47792	-7.70609	.201438	-2.11722
TRIANGLE	0.	0.	1.75378	5.49939
41.0000	-1.51656	-7.90310	.209939	-2.17080
	0.	0.	1.79111	5.65992
42.0000	-1.55523	-8.09983	.218317	-2.22438
	0.	0.	1.82858	5.82059
43.0000	-1.59391	-8.29630	.226584	-2.27797
	0.	0.	1.86620	5.98136
44.0000	-1.63261	-8.49254	.234750	-2.33156
	0.	0.	1.90394	6.14221
45.0000	-1.67132	-8.68857	.242822	-2.38515
	0.	0.	1.94181	6.30311
46.0000	-1.71003	-8.88442	.250810	-2.43874
	0.	0.	1.97979	6.46404
47.0000	-1.74875	-9.08009	.258720	-2.49234
	0.	0.	2.01788	6.62497
48.0000	-1.78747	-9.27561	.266558	-2.54594
	0.	0.	2.05606	6.78590
49.0000	-1.82619	-9.47098	.274330	-2.59954
	0.	0.	2.09434	6.94680
50.0000	-1.86491	-9.66623	.282040	-2.65315
RT TRNGL	0.	0.	2.13270	7.10767
51.0000	-1.90362	-9.86136	.289694	-2.70675
	0.	0.	2.17115	7.26849

Listing of tape 8 for sample case (modal ratios or eigenvectors and characteristic roots).-

LONGITUDINAL STABILITY OF TETHERED BALLOON - LRC BALLOON-REFERENCE CONFIGURATION 11/20/72 09.56.50.

EIGENVECTORS

	COMPLEX ROOT-REAL	IMAG	X/THETA,M/DEG-REAL	IMAG	Z/THETA,M/DEG-REAL	IMAG	THETA,DEG-REAL	IMAG
VELOCITY=	1.0000							
	-1.26423E-02	.23838	-1.5448	.14418	6.27094E-02	-1.20755E-03	1.0000	0.
	-.10741	.85429	5.86335E-03	-2.26595E-03	4.97133E-02	1.87209E-05	1.0000	0.
	-4.11192E-02	94.112	-4.08401E-04	4.18936E-06	-2.45945E-02	1.13327E-04	1.0000	0.
VELOCITY=	2.0000							
	-2.51995E-02	.23926	-1.5489	.23647	5.90583E-02	-7.72946E-03	1.0000	0.
	-.21683	.86727	6.99266E-03	-5.25280E-03	4.60560E-02	1.63166E-04	1.0000	0.
	-7.99317E-02	23.795	-1.64100E-03	5.61810E-05	-2.52525E-02	9.03961E-04	1.0000	0.
VELOCITY=	3.0000							
	-3.81500E-02	.24174	-1.5725	.25923	.15826	-1.83395E-02	1.0000	0.
	-.32918	.93764	7.37272E-03	-8.55390E-03	4.57917E-02	5.37688E-04	1.0000	0.
	-.11499	10.804	-3.68046E-03	3.12446E-04	-2.60716E-02	3.02937E-03	1.0000	0.
VELOCITY=	4.0000							
	-5.28144E-02	.24673	-1.5974	.21878	.23721	-2.61859E-02	1.0000	0.
	-.44437	.99948	6.55074E-03	-1.13370E-02	4.34355E-02	1.06148E-03	1.0000	0.
	-.14561	6.2916	-6.37539E-03	1.12358E-03	-2.65554E-02	7.08613E-03	1.0000	0.
VELOCITY=	5.0000							
	-7.05433E-02	.25489	-1.5684	.13743	.32551	-2.44904E-02	1.0000	0.
	-.56292	1.0660	4.60497E-03	-1.31340E-02	4.12372E-02	1.38689E-03	1.0000	0.
	-.17032	4.2332	-9.30120E-03	3.07268E-03	-2.59752E-02	1.35780E-02	1.0000	0.
VELOCITY=	6.0000							
	-9.24642E-02	.26669	-1.5226	5.06467E-02	.40636	-1.35495E-02	1.0000	0.
	-.68831	1.1290	1.53592E-03	-1.38392E-02	3.90377E-02	8.75131E-04	1.0000	0.
	-.18463	3.1392	-1.16548E-02	6.98890E-03	-2.35629E-02	2.30076E-02	1.0000	0.
VELOCITY=	7.0000							
	-.11943	.28260	-1.4053	-1.05304E-02	.46543	-1.62598E-03	1.0000	0.
	-.82548	1.1739	-2.93894E-03	-1.35798E-02	3.66239E-02	-1.50006E-03	1.0000	0.
	-.18261	2.5051	-1.22588E-02	1.40914E-02	-1.69841E-02	3.61419E-02	1.0000	0.
VELOCITY=	8.0000							
	-.15248	.30335	-1.2582	-3.53007E-02	.49694	2.06101E-03	1.0000	0.
	-.96565	1.1786	-9.00565E-03	-1.26506E-02	3.45074E-02	-6.69443E-03	1.0000	0.
	-.17183	2.1331	-9.83434E-03	2.62502E-02	-1.30417E-02	5.37915E-02	1.0000	0.
VELOCITY=	9.0000							
	-.19265	.33030	-1.0999	-2.98148E-02	.50118	-6.89017E-03	1.0000	0.
	-1.0837	1.1387	-1.64413E-02	-1.07123E-02	3.32430E-02	-1.45352E-02	1.0000	0.
	-.17668	1.9208	-2.72283E-03	4.57690E-02	-6.70130E-03	7.61096E-02	1.0000	0.
VELOCITY=	10.000	CIRCLE						
	-.24199	.36618	-.93841	-6.19789E-03	.47926	-2.78004E-02	1.0000	0.
	-1.1674	1.0645	-2.52881E-02	-6.09529E-03	3.20354E-02	-2.49962E-02	1.0000	0.
	-.20673	1.7978	1.38480E-02	7.58449E-02	9.49311E-04	.10455	1.0000	0.
VELOCITY=	11.000							
	-.30413	.41738	-.77120	2.21511E-02	.42898	-5.58109E-02	1.0000	0.
	-1.2153	.95012	-3.55593E-02	5.15459E-03	2.68800E-02	-3.99044E-02	1.0000	0.
	-.25984	1.7257	5.53235E-02	.12079	1.51544E-02	.14489	1.0000	0.
VELOCITY=	12.000							
	-.38347	.50382	-.58687	3.14933E-02	.34181	-7.57858E-02	1.0000	0.
	-1.2230	.75342	-4.16676E-02	3.53035E-02	1.63240E-02	-6.56029E-02	1.0000	0.
	-.33602	1.6379	.17345	.16257	6.32657E-02	.20517	1.0000	0.

VELOCITY=	13.000							
	-1.2356	.29985	4.51310E-02	8.20167E-02	-7.76739E-02	-8.58335E-02	1.0000	0.
	-.43200	.67914	-.41369	-4.16970E-02	.23329	-3.81841E-02	1.0000	0.
	-.43805	1.6839	.35173	1.87030E-03	.20140	.19495	1.0000	0.
VELOCITY=	14.000							
	-.80603	0.	1.3383	0.	-1.2739	0.	1.0000	0.
	-1.9089	0.	1.50376E-02	0.	-3.34929E-02	0.	1.0000	0.
	-.35407	.83382	-.35015	-.12989	.18012	2.85098E-02	1.0000	0.
	-.55738	1.7269	.22511	-.17368	.21227	7.53582E-02	1.0000	0.
VELOCITY=	15.000							
	-.71782	0.	-16.511	0.	15.529	0.	1.0000	0.
	-2.2822	0.	5.85532E-03	0.	-2.05655E-02	0.	1.0000	0.
	-.26487	.90899	-.31905	-.17154	.15219	7.88962E-02	1.0000	0.
	-.66726	1.8144	.13159	-.16579	.17234	4.30847E-02	1.0000	0.
VELOCITY=	16.000							
	-.69459	0.	-2.8222	0.	2.7113	0.	1.0000	0.
	-2.5653	0.	2.29792E-03	0.	-1.51079E-02	0.	1.0000	0.
	-.19733	.95198	-.29249	-.19338	.13420	.11744	1.0000	0.
	-.75807	1.9247	9.67209E-02	-.14569	.15050	3.66613E-02	1.0000	0.
VELOCITY=	17.000							
	-.69538	0.	-1.9239	0.	1.9005	0.	1.0000	0.
	-2.8571	0.	3.62448E-04	0.	-1.20466E-02	0.	1.0000	0.
	-.14700	.98437	-.26891	-.20691	.12105	.14737	1.0000	0.
	-.83532	2.0452	8.10235E-02	-.13225	.13791	3.51906E-02	1.0000	0.
VELOCITY=	18.000							
	-.70834	0.	-1.5782	0.	1.6038	0.	1.0000	0.
	-3.1106	0.	-6.77885E-04	0.	-1.00952E-02	0.	1.0000	0.
	-.10791	1.0135	-.24840	-.21600	.11051	.17116	1.0000	0.
	-.90433	2.1715	7.26327E-02	-.12309	.12961	3.49751E-02	1.0000	0.
VELOCITY=	19.000							
	-.72852	0.	-1.3869	0.	1.4479	0.	1.0000	0.
	-3.3522	0.	-1.75685E-03	0.	-8.75710E-03	0.	1.0000	0.
	-7.62547E-02	1.0418	-.23068	-.22233	.10162	.19050	1.0000	0.
	-.96825	2.3023	6.76208E-02	-.11663	.12356	3.51384E-02	1.0000	0.
VELOCITY=	20.000	SQUARE						
	-.75343	0.	-1.2617	0.	1.3506	0.	1.0000	0.
	-3.5853	0.	-2.42408E-03	0.	-7.79584E-03	0.	1.0000	0.
	-4.97245E-02	1.0705	-.21539	-.22678	9.38913E-02	.20651	1.0000	0.
	-1.0289	2.4367	6.44029E-02	-.11184	.11884	3.54113E-02	1.0000	0.
VELOCITY=	21.000							
	-.78161	0.	-1.1716	0.	1.2833	0.	1.0000	0.
	-3.8120	0.	-2.95640E-03	0.	-7.08316E-03	0.	1.0000	0.
	-2.68668E-02	1.0997	-.20215	-.22993	8.70715E-02	.21998	1.0000	0.
	-1.0874	2.5744	6.22336E-02	-.10812	.11496	3.56973E-02	1.0000	0.
VELOCITY=	22.000							
	-.81215	0.	-1.1028	0.	1.2336	0.	1.0000	0.
	-4.0338	0.	-3.39710E-03	0.	-6.54293E-03	0.	1.0000	0.
	-6.74032E-03	1.1296	-.19064	-.23214	8.09958E-02	.23145	1.0000	0.
	-1.1445	2.7149	6.07200E-02	-.10514	.11167	3.59602E-02	1.0000	0.
VELOCITY=	23.000							
	-.84445	0.	-1.0481	0.	1.1952	0.	1.0000	0.
	-4.2516	0.	-3.77233E-03	0.	-6.12687E-03	0.	1.0000	0.
	1.13049E-02	1.1603	-.18058	-.23367	7.55502E-02	.24131	1.0000	0.
	-1.2006	2.8581	5.96375E-02	-.10268	.10881	3.61873E-02	1.0000	0.
VELOCITY=	24.000							
	-.87809	0.	-1.0034	0.	1.1644	0.	1.0000	0.
	-4.4662	0.	-4.09875E-03	0.	-5.60278E-03	0.	1.0000	0.
	2.77200E-02	1.1917	-.17176	-.23471	7.06487E-02	.24986	1.0000	0.
	-1.2559	3.0037	5.88491E-02	-.10060	.10628	3.63759E-02	1.0000	0.
VELOCITY=	25.000							
	-.91276	0.	-.96608	0.	1.1391	0.	1.0000	0.
	-4.6782	0.	-4.38746E-03	0.	-5.54832E-03	0.	1.0000	0.
	4.28332E-02	1.2237	-.16398	-.23538	6.62223E-02	.25734	1.0000	0.
	-1.3108	3.1513	5.82674E-02	-9.88138E-02	.10401	3.65278E-02	1.0000	0.

VELOCITY=	26.000							
	-.94824	0.	-.93440	0.	1.1180	0.	1.0000	0.
	-4.8879	0.	-4.64613E-03	0.	-5.34751E-03	0.	1.0000	0.
	5.66887E-02	1.2563	-.15709	-.23578	6.22139E-02	.26392	1.0000	0.
	-1.3652	3.3008	5.78342E-02	-9.72522E-02	.10196	3.66467E-02	1.0000	0.
VELOCITY=	27.000							
	-.96436	0.	-.90716	0.	1.1000	0.	1.0000	0.
	-5.0958	0.	-4.88025E-03	0.	-5.18860E-03	0.	1.0000	0.
	7.00723E-02	1.2895	-.15096	-.23597	5.85749E-02	.26974	1.0000	0.
	-1.4194	3.4519	5.75095E-02	-9.58715E-02	.10009	3.67366E-02	1.0000	0.
VELOCITY=	28.000							
	-1.0210	0.	-.88350	0.	1.0845	0.	1.0000	0.
	-5.3020	0.	-5.09385E-03	0.	-5.06279E-03	0.	1.0000	0.
	8.25278E-02	1.3232	-.14549	-.23602	5.52639E-02	.27493	1.0000	0.
	-1.4735	3.6044	5.72655E-02	-9.46383E-02	9.83791E-02	3.68018E-02	1.0000	0.
VELOCITY=	29.000							
	-1.0580	0.	-.86275	0.	1.0710	0.	1.0000	0.
	-5.5069	0.	-5.28997E-03	0.	-4.96336E-03	0.	1.0000	0.
	9.43687E-02	1.3574	-.14060	-.23597	5.22446E-02	.27957	1.0000	0.
	-1.5273	3.7583	5.70820E-02	-9.35280E-02	9.66079E-02	3.68460E-02	1.0000	0.
VELOCITY=	30.000	DIAMOND						
	-1.0954	0.	-.84443	0.	1.0591	0.	1.0000	0.
	-5.7105	0.	-5.47097E-03	0.	-4.88510E-03	0.	1.0000	0.
	.10569	1.3920	-.13619	-.23583	4.94857E-02	.28373	1.0000	0.
	-1.5611	3.9132	5.69445E-02	-9.25213E-02	9.53592E-02	3.68726E-02	1.0000	0.
VELOCITY=	31.000							
	-1.1330	0.	-.82814	0.	1.0487	0.	1.0000	0.
	-5.9131	0.	-5.63871E-03	0.	-4.82395E-03	0.	1.0000	0.
	.11655	1.4269	-.13222	-.23563	4.69595E-02	.28749	1.0000	0.
	-1.6348	4.0692	5.68422E-02	-9.16035E-02	9.40198E-02	3.68848E-02	1.0000	0.
VELOCITY=	32.000							
	-1.1709	0.	-.81358	0.	1.0393	0.	1.0000	0.
	-6.1148	0.	-5.79471E-03	0.	-4.77667E-03	0.	1.0000	0.
	.12703	1.4622	-.12863	-.23540	4.46417E-02	.29090	1.0000	0.
	-1.6885	4.2260	5.67669E-02	-9.07627E-02	9.27785E-02	3.68650E-02	1.0000	0.
VELOCITY=	33.000							
	-1.2089	0.	-.80049	0.	1.0310	0.	1.0000	0.
	-6.3157	0.	-5.94020E-03	0.	-4.74070E-03	0.	1.0000	0.
	.13717	1.4978	-.12537	-.23513	4.25112E-02	.29399	1.0000	0.
	-1.7421	4.3835	5.67126E-02	-8.99893E-02	9.16265E-02	3.68756E-02	1.0000	0.
VELOCITY=	34.000							
	-1.2471	0.	-.78869	0.	1.0235	0.	1.0000	0.
	-6.5159	0.	-6.07622E-03	0.	-4.71396E-03	0.	1.0000	0.
	.14701	1.5337	-.12240	-.23485	4.05491E-02	.29681	1.0000	0.
	-1.7957	4.5417	5.66746E-02	-8.92754E-02	9.05547E-02	3.68583E-02	1.0000	0.
VELOCITY=	35.000							
	-1.2654	0.	-.77799	0.	1.0167	0.	1.0000	0.
	-6.7155	0.	-6.20365E-03	0.	-4.69477E-03	0.	1.0000	0.
	.15658	1.5698	-.11969	-.23456	3.87388E-02	.29939	1.0000	0.
	-1.8493	4.7003	5.66492E-02	-8.86145E-02	8.95562E-02	3.68347E-02	1.0000	0.
VELOCITY=	36.000							
	-1.3238	0.	-.76826	0.	1.0105	0.	1.0000	0.
	-6.9145	0.	-6.32325E-03	0.	-4.68177E-03	0.	1.0000	0.
	.16593	1.6062	-.11722	-.23426	3.70655E-02	.30175	1.0000	0.
	-1.9029	4.8595	5.66338E-02	-8.80009E-02	8.86247E-02	3.68062E-02	1.0000	0.
VELOCITY=	37.000							
	-1.3622	0.	-.75937	0.	1.0049	0.	1.0000	0.
	-7.1130	0.	-6.43568E-03	0.	-4.67364E-03	0.	1.0000	0.
	.17506	1.6428	-.11495	-.23396	3.55162E-02	.30392	1.0000	0.
	-1.9565	5.0190	5.66260E-02	-8.74300E-02	8.77544E-02	3.67739E-02	1.0000	0.
VELOCITY=	38.000							
	-1.4007	0.	-.75124	0.	.99971	0.	1.0000	0.
	-7.3111	0.	-6.54152E-03	0.	-4.67007E-03	0.	1.0000	0.
	.18402	1.6796	-.11286	-.23367	3.40794E-02	.30593	1.0000	0.
	-2.0101	5.1789	5.66243E-02	-8.68976E-02	8.69403E-02	3.67386E-02	1.0000	0.

VELOCITY=	39.000							
	-1.4393	0.	-.74378	0.	.99500	0.	1.0000	0.
	-7.5088	0.	-6.64128E-03	0.	-4.66969E-03	0.	1.0000	0.
	.19230	1.7166	-.11093	-.23337	3.27447E-02	.30776	1.0000	0.
	-2.0636	5.3390	5.66273E-02	-8.64002E-02	8.61778E-02	3.67012E-02	1.0000	0.
VELOCITY=	40.000	TRIANGLE						
	-1.4779	0.	-.73691	0.	.99066	0.	1.0000	0.
	-7.7061	0.	-6.73542E-03	0.	-4.67208E-03	0.	1.0000	0.
	.20144	1.7538	-.10916	-.23309	3.15029E-02	.30949	1.0000	0.
	-2.1172	5.4994	5.66338E-02	-8.59347E-02	8.54628E-02	3.66622E-02	1.0000	0.
VELOCITY=	41.000							
	-1.5166	0.	-.73057	0.	.98665	0.	1.0000	0.
	-7.9031	0.	-6.82435E-03	0.	-4.67673E-03	0.	1.0000	0.
	.20994	1.7911	-.10752	-.23281	3.03457E-02	.31108	1.0000	0.
	-2.1708	5.6599	5.66432E-02	-8.54983E-02	8.47915E-02	3.66222E-02	1.0000	0.
VELOCITY=	42.000							
	-1.5552	0.	-.72471	0.	.98297	0.	1.0000	0.
	-8.0998	0.	-6.90845E-03	0.	-4.68320E-03	0.	1.0000	0.
	.21632	1.8286	-.10599	-.23254	2.92659E-02	.31256	1.0000	0.
	-2.2244	5.8206	5.66547E-02	-8.50886E-02	8.41607E-02	3.65816E-02	1.0000	0.
VELOCITY=	43.000							
	-1.5939	0.	-.71928	0.	.97955	0.	1.0000	0.
	-8.2963	0.	-6.98805E-03	0.	-4.69114E-03	0.	1.0000	0.
	.22658	1.8662	-.10458	-.23228	2.82568E-02	.31394	1.0000	0.
	-2.2780	5.9814	5.66677E-02	-8.47035E-02	8.35672E-02	3.65408E-02	1.0000	0.
VELOCITY=	44.000							
	-1.6326	0.	-.71423	0.	.97637	0.	1.0000	0.
	-8.4925	0.	-7.06346E-03	0.	-4.70025E-03	0.	1.0000	0.
	.23475	1.9039	-.10326	-.23202	2.73125E-02	.31522	1.0000	0.
	-2.3316	6.1422	5.66819E-02	-8.43410E-02	8.30063E-02	3.65000E-02	1.0000	0.
VELOCITY=	45.000							
	-1.6713	0.	-.70953	0.	.97342	0.	1.0000	0.
	-8.6886	0.	-7.13496E-03	0.	-4.71027E-03	0.	1.0000	0.
	.24262	1.9418	-.10204	-.23178	2.64276E-02	.31642	1.0000	0.
	-2.3851	6.3031	5.66969E-02	-8.39993E-02	8.24815E-02	3.64594E-02	1.0000	0.
VELOCITY=	46.000							
	-1.7100	0.	-.70515	0.	.97067	0.	1.0000	0.
	-8.8844	0.	-7.20281E-03	0.	-4.72101E-03	0.	1.0000	0.
	.25081	1.9798	-.10090	-.23154	2.55974E-02	.31754	1.0000	0.
	-2.4387	6.4640	5.67125E-02	-8.36770E-02	8.19844E-02	3.64193E-02	1.0000	0.
VELOCITY=	47.000							
	-1.7488	0.	-.70107	0.	.96810	0.	1.0000	0.
	-9.0801	0.	-7.26724E-03	0.	-4.73228E-03	0.	1.0000	0.
	.25872	2.0179	-9.96275E-02	-.23132	2.48175E-02	.31860	1.0000	0.
	-2.4923	6.6250	5.67284E-02	-8.33725E-02	8.15148E-02	3.63797E-02	1.0000	0.
VELOCITY=	48.000							
	-1.7875	0.	-.69724	0.	.96570	0.	1.0000	0.
	-9.2756	0.	-7.32847E-03	0.	-4.74395E-03	0.	1.0000	0.
	.26656	2.0561	-9.88272E-02	-.23110	2.40840E-02	.31958	1.0000	0.
	-2.5459	6.7859	5.67445E-02	-8.30846E-02	8.10710E-02	3.63409E-02	1.0000	0.
VELOCITY=	49.000							
	-1.8262	0.	-.69366	0.	.96345	0.	1.0000	0.
	-9.4710	0.	-7.38671E-03	0.	-4.75589E-03	0.	1.0000	0.
	.27433	2.0943	-9.78896E-02	-.23089	2.33933E-02	.32051	1.0000	0.
	-2.5995	6.9468	5.67606E-02	-8.28121E-02	8.06510E-02	3.63028E-02	1.0000	0.
VELOCITY=	50.000	RT TRNGL						
	-1.8649	0.	-.69030	0.	.96134	0.	1.0000	0.
	-9.6662	0.	-7.44213E-03	0.	-4.76801E-03	0.	1.0000	0.
	.28204	2.1327	-9.70095E-02	-.23069	2.27422E-02	.32139	1.0000	0.
	-2.6531	7.1077	5.67766E-02	-8.25540E-02	8.02533E-02	3.62656E-02	1.0000	0.
VELOCITY=	51.000							
	-1.9036	0.	-.68714	0.	.95935	0.	1.0000	0.
	-9.8614	0.	-7.49491E-03	0.	-4.78022E-03	0.	1.0000	0.
	.28969	2.1712	-9.61823E-02	-.23050	2.21278E-02	.32221	1.0000	0.
	-2.7068	7.2685	5.67925E-02	-8.23092E-02	7.98764E-02	3.62292E-02	1.0000	0.

Listing of tape 30 for sample case (trim angle of attack and aerodynamic coefficients).-

LONGITUDINAL STABILITY OF TETHERED BALLOON - LRC BALLOON-REFERENCE CONFIGURATION 11/20/72 09.56.50.

AERODYNAMIC COEFFICIENTS

VELOCITY	ALPHAD	CD	CL	CM	CDA	CLA	CMA	CLQ	CMAD	CMQ
1.000	8.767	5.9599E-02	9.9706E-02	-3.4372E-03	4.1492E-02	.7244	3.2132E-02	.8974	-3.8858E-02	-.2788
2.000	8.550	5.9453E-02	9.6972E-02	-3.5596E-03	3.5798E-02	.7213	3.2572E-02	.8963	-3.8861E-02	-.2787
3.000	8.250	5.9284E-02	9.3205E-02	-3.7310E-03	2.0968E-02	.7188	3.2908E-02	.8952	-3.8864E-02	-.2786
4.000	7.923	5.9137E-02	8.9106E-02	-3.9191E-03	2.2750E-02	.7181	3.2957E-02	.8947	-3.8867E-02	-.2786
5.000	7.609	5.9026E-02	8.5161E-02	-4.0996E-03	1.7825E-02	.7193	3.2733E-02	.8947	-3.8869E-02	-.2787
6.000	7.327	5.8948E-02	8.1624E-02	-4.2594E-03	1.4161E-02	.7216	3.2340E-02	.8951	-3.8871E-02	-.2788
7.000	7.086	5.8894E-02	7.8580E-02	-4.3946E-03	1.1554E-02	.7245	3.1880E-02	.8957	-3.8873E-02	-.2789
8.000	6.884	5.8856E-02	7.6016E-02	-4.5064E-03	9.6647E-03	.7274	3.1419E-02	.8964	-3.8874E-02	-.2791
9.000	6.716	5.8830E-02	7.3882E-02	-4.5978E-03	8.2928E-03	.7301	3.0991E-02	.8970	-3.8875E-02	-.2792
10.00	6.577	5.8811E-02	7.2110E-02	-4.6724E-03	7.2803E-03	.7326	3.0610E-02	.8976	-3.8875E-02	-.2793
11.00	6.462	5.8796E-02	7.0640E-02	-4.7335E-03	6.5194E-03	.7347	3.0279E-02	.8982	-3.8876E-02	-.2794
12.00	6.367	5.8787E-02	6.9415E-02	-4.7836E-03	5.9366E-03	.7366	2.9993E-02	.8986	-3.8876E-02	-.2795
13.00	6.287	5.8779E-02	6.8389E-02	-4.8252E-03	5.4829E-03	.7382	2.9748E-02	.8990	-3.8876E-02	-.2795
14.00	6.220	5.8773E-02	6.7526E-02	-4.8598E-03	5.1234E-03	.7395	2.9538E-02	.8994	-3.8876E-02	-.2796
15.00	6.164	5.8768E-02	6.6795E-02	-4.8889E-03	4.8344E-03	.7407	2.9358E-02	.8997	-3.8876E-02	-.2796
16.00	6.115	5.8764E-02	6.6172E-02	-4.9135E-03	4.5986E-03	.7417	2.9203E-02	.8999	-3.8876E-02	-.2797
17.00	6.074	5.8761E-02	6.5638E-02	-4.9345E-03	4.4045E-03	.7426	2.9069E-02	.9002	-3.8877E-02	-.2797
18.00	6.039	5.8758E-02	6.5177E-02	-4.9525E-03	4.2425E-03	.7434	2.8952E-02	.9004	-3.8877E-02	-.2797
19.00	6.008	5.8756E-02	6.4777E-02	-4.9680E-03	4.1060E-03	.7440	2.8851E-02	.9005	-3.8877E-02	-.2798
20.00	5.981	5.8754E-02	6.4429E-02	-4.9815E-03	3.9899E-03	.7446	2.8762E-02	.9007	-3.8877E-02	-.2798
21.00	5.958	5.8753E-02	6.4124E-02	-4.9933E-03	3.8905E-03	.7451	2.8684E-02	.9008	-3.8877E-02	-.2798
22.00	5.937	5.8751E-02	6.3855E-02	-5.0036E-03	3.8046E-03	.7456	2.8615E-02	.9009	-3.8877E-02	-.2798
23.00	5.919	5.8750E-02	6.3617E-02	-5.0127E-03	3.7300E-03	.7460	2.8554E-02	.9010	-3.8877E-02	-.2799
24.00	5.902	5.8749E-02	6.3406E-02	-5.0208E-03	3.6647E-03	.7464	2.8499E-02	.9011	-3.8877E-02	-.2799
25.00	5.888	5.8748E-02	6.3217E-02	-5.0280E-03	3.6072E-03	.7467	2.8451E-02	.9012	-3.8877E-02	-.2799
26.00	5.875	5.8747E-02	6.3049E-02	-5.0344E-03	3.5564E-03	.7470	2.8407E-02	.9013	-3.8877E-02	-.2799
27.00	5.863	5.8746E-02	6.2897E-02	-5.0402E-03	3.5113E-03	.7472	2.8368E-02	.9014	-3.8877E-02	-.2799
28.00	5.853	5.8746E-02	6.2760E-02	-5.0454E-03	3.4710E-03	.7475	2.8332E-02	.9014	-3.8877E-02	-.2799
29.00	5.843	5.8745E-02	6.2636E-02	-5.0500E-03	3.4348E-03	.7477	2.8300E-02	.9015	-3.8877E-02	-.2799
30.00	5.835	5.8745E-02	6.2524E-02	-5.0543E-03	3.4023E-03	.7479	2.8271E-02	.9015	-3.8877E-02	-.2799
31.00	5.827	5.8744E-02	6.2422E-02	-5.0581E-03	3.3730E-03	.7480	2.8244E-02	.9016	-3.8877E-02	-.2799
32.00	5.820	5.8744E-02	6.2329E-02	-5.0617E-03	3.3464E-03	.7482	2.8220E-02	.9016	-3.8877E-02	-.2800
33.00	5.813	5.8743E-02	6.2244E-02	-5.0649E-03	3.3222E-03	.7483	2.8198E-02	.9016	-3.8877E-02	-.2800
34.00	5.807	5.8743E-02	6.2166E-02	-5.0678E-03	3.3001E-03	.7485	2.8177E-02	.9017	-3.8877E-02	-.2800
35.00	5.802	5.8743E-02	6.2094E-02	-5.0705E-03	3.2800E-03	.7486	2.8159E-02	.9017	-3.8877E-02	-.2800
36.00	5.797	5.8742E-02	6.2028E-02	-5.0730E-03	3.2615E-03	.7487	2.8141E-02	.9017	-3.8877E-02	-.2800
37.00	5.792	5.8742E-02	6.1966E-02	-5.0753E-03	3.2445E-03	.7488	2.8125E-02	.9018	-3.8877E-02	-.2800
38.00	5.788	5.8742E-02	6.1910E-02	-5.0774E-03	3.2289E-03	.7489	2.8111E-02	.9018	-3.8877E-02	-.2800
39.00	5.784	5.8742E-02	6.1858E-02	-5.0794E-03	3.2144E-03	.7490	2.8097E-02	.9018	-3.8877E-02	-.2800
40.00	5.780	5.8742E-02	6.1809E-02	-5.0812E-03	3.2011E-03	.7491	2.8084E-02	.9018	-3.8877E-02	-.2800
41.00	5.777	5.8741E-02	6.1764E-02	-5.0829E-03	3.1887E-03	.7492	2.8073E-02	.9019	-3.8877E-02	-.2800
42.00	5.773	5.8741E-02	6.1722E-02	-5.0845E-03	3.1772E-03	.7492	2.8062E-02	.9019	-3.8877E-02	-.2800
43.00	5.770	5.8741E-02	6.1682E-02	-5.0860E-03	3.1665E-03	.7493	2.8051E-02	.9019	-3.8877E-02	-.2800
44.00	5.768	5.8741E-02	6.1646E-02	-5.0873E-03	3.1565E-03	.7494	2.8042E-02	.9019	-3.8877E-02	-.2800
45.00	5.765	5.8741E-02	6.1611E-02	-5.0886E-03	3.1472E-03	.7494	2.8033E-02	.9019	-3.8877E-02	-.2800
46.00	5.762	5.8741E-02	6.1579E-02	-5.0898E-03	3.1385E-03	.7495	2.8024E-02	.9019	-3.8877E-02	-.2800
47.00	5.760	5.8740E-02	6.1549E-02	-5.0909E-03	3.1304E-03	.7495	2.8016E-02	.9020	-3.8877E-02	-.2800
48.00	5.758	5.8740E-02	6.1521E-02	-5.0920E-03	3.1227E-03	.7496	2.8009E-02	.9020	-3.8877E-02	-.2800
49.00	5.756	5.8740E-02	6.1494E-02	-5.0930E-03	3.1156E-03	.7496	2.8002E-02	.9020	-3.8877E-02	-.2800
50.00	5.754	5.8740E-02	6.1469E-02	-5.0939E-03	3.1088E-03	.7497	2.7995E-02	.9020	-3.8877E-02	-.2800
51.00	5.752	5.8740E-02	6.1445E-02	-5.0948E-03	3.1025E-03	.7497	2.7989E-02	.9020	-3.8877E-02	-.2800

Listing of tape 31 for sample case (tether spring constants).-

LONGITUDINAL STABILITY OF TETHERED BALLOON - LRC BALLOON-REFERENCE CONFIGURATION 11/20/72 09.56.50.

	TETHER SPRINGS									
VELOCITY	SKXX	SKXZ	SKXT	SKZX	SKZZ	SKZT	SKTX	SKTZ	SKTT,D	SKTT,TL
1.000	9.239	961.7	-2737	961.7	1.1465E+05	-3.2680E+05	-2737	-3.2680E+05	9.3156E+05	273.7
2.000	9.445	250.9	-687.6	250.8	7627	-2.1023E+04	-687.4	-2.1023E+04	5.7963E+04	280.9
3.000	9.801	119.4	-307.5	119.3	1664	-4342	-307.2	-4341	1.1340E+04	292.2
4.000	10.32	73.76	-174.1	73.58	601.5	-1455	-173.6	-1454	3533	307.1
5.000	11.05	53.09	-112.0	52.80	290.5	-638.6	-111.2	-637.7	1418	325.0
6.000	12.01	42.44	-77.95	42.01	170.2	-333.4	-76.75	-332.0	568.0	345.8
7.000	13.26	36.68	-56.97	36.07	114.7	-196.4	-55.28	-194.3	352.6	369.2
8.000	14.83	33.68	-42.72	32.84	85.93	-126.1	-40.40	-123.2	203.6	395.5
9.000	16.78	32.40	-32.11	31.28	69.85	-85.94	-29.01	-82.10	127.5	424.5
10.00	19.13	32.31	-23.44	30.84	60.47	-60.83	-19.40	-55.80	87.12	456.3
11.00	21.94	33.13	-15.74	31.24	54.95	-43.76	-10.56	-37.30	66.14	491.1
12.00	25.23	34.66	-8.370	32.28	51.84	-31.23	-1.855	-23.06	56.77	528.8
13.00	29.03	36.81	-.9242	33.86	50.34	-21.33	7.148	-11.19	55.10	569.6
14.00	33.39	39.52	6.894	35.91	49.97	-12.98	16.75	-.5587	59.09	613.3
15.00	38.33	42.75	15.30	38.39	50.46	-5.507	27.19	9.492	67.62	660.1
16.00	43.68	46.48	24.47	41.28	51.62	1.497	38.62	19.39	80.10	710.0
17.00	50.07	50.69	34.51	44.56	53.34	8.314	51.19	29.43	96.20	763.0
18.00	56.91	55.37	45.55	48.22	55.54	15.14	65.00	39.80	115.7	819.1
19.00	64.43	60.53	57.65	52.25	58.16	22.10	80.13	50.63	138.6	878.4
20.00	72.65	66.15	70.90	56.66	61.16	29.31	96.66	62.03	164.8	940.7
21.00	81.57	72.24	85.33	61.44	64.52	36.83	114.6	74.07	194.3	1006
22.00	91.21	78.78	101.0	66.58	68.20	44.73	134.1	86.81	227.1	1075
23.00	101.6	85.79	117.9	72.10	72.21	53.03	155.1	100.3	263.2	1147
24.00	112.7	93.26	136.2	77.98	76.51	61.78	177.6	114.5	302.5	1222
25.00	124.5	101.2	155.7	84.22	81.11	71.01	201.7	129.5	345.2	1300
26.00	137.1	109.6	176.6	90.83	85.99	80.71	227.3	145.4	391.1	1381
27.00	150.4	118.4	198.8	97.81	91.16	90.93	254.6	162.0	440.3	1466
28.00	164.5	127.7	222.4	105.1	96.59	101.7	283.4	179.5	492.8	1553
29.00	179.3	137.4	247.3	112.8	102.3	112.9	313.8	197.8	548.6	1644
30.00	194.8	147.6	273.6	120.9	106.3	124.7	345.8	216.9	607.6	1738
31.00	211.1	158.2	301.2	129.3	114.5	137.0	379.4	236.8	669.8	1835
32.00	228.1	169.3	330.1	138.1	121.0	149.8	414.5	257.6	735.2	1935
33.00	245.8	180.8	360.4	147.2	127.7	163.2	451.2	279.2	803.9	2039
34.00	264.2	192.7	392.0	156.6	134.7	177.1	489.5	301.7	875.6	2146
35.00	283.4	205.1	424.9	166.5	141.9	191.5	529.3	325.0	950.5	2255
36.00	303.3	217.9	459.1	176.6	149.4	206.5	570.7	349.1	1029	2368
37.00	323.9	231.1	494.7	187.1	157.2	222.0	613.6	374.0	1110	2485
38.00	345.2	244.8	531.5	198.0	165.1	238.0	658.0	399.7	1194	2604
39.00	367.2	258.9	569.6	209.2	173.4	254.6	703.9	426.3	1281	2727
40.00	389.9	273.4	609.0	220.7	181.8	271.6	751.3	453.7	1371	2852
41.00	413.3	288.3	649.7	232.6	190.5	289.2	800.3	481.8	1464	2981
42.00	437.4	303.6	691.7	244.8	199.5	307.4	850.7	510.8	1561	3113
43.00	462.2	319.4	734.9	257.3	208.6	326.0	902.6	540.6	1660	3249
44.00	487.6	335.6	779.3	270.2	218.0	345.2	955.9	571.2	1762	3387
45.00	513.8	352.2	825.0	283.4	227.7	364.8	1011	602.6	1867	3529
46.00	540.6	369.2	871.9	297.0	237.6	385.0	1067	634.8	1975	3674
47.00	568.1	386.6	920.1	310.8	247.7	405.7	1125	667.7	2085	3822
48.00	596.3	404.4	969.5	325.0	258.1	426.9	1184	701.5	2199	3973
49.00	625.1	422.7	1020	339.6	268.6	448.6	1245	736.0	2316	4127
50.00	654.6	441.3	1072	354.4	279.5	470.8	1307	771.3	2435	4285
51.00	684.8	460.4	1125	369.6	290.5	493.6	1370	807.4	2557	4445

Listing of tape 32 for sample case (equilibrium tether conditions).-

LONGITUDINAL STABILITY OF TETHERED BALLOON - LRC BALLOON-REFERENCE CONFIGURATION 11/20/72 09.56.50.

	TETHER CONDITIONS						
VELOCITY	X1	Z1	GAM0	T0	GAM1	T1	CAB DRAG
1.000	.4953	61.00	89.19	61.51	89.62	82.43	1.0104E-02
2.000	1.943	60.96	86.82	62.77	89.30	83.68	4.0418E-02
3.000	4.236	60.63	83.08	64.78	88.46	85.65	9.0940E-02
4.000	7.218	60.51	78.28	67.49	87.55	88.24	.1617
5.000	10.59	59.92	72.78	70.85	86.01	91.40	.2526
6.000	14.44	59.02	66.58	74.87	84.46	95.11	.3638
7.000	18.26	57.83	61.25	79.55	82.61	99.38	.4951
8.000	21.95	56.38	55.88	84.91	81.04	104.3	.6467
9.000	25.39	54.76	51.04	90.96	79.21	109.7	.8185
10.00	28.52	53.10	46.81	97.59	77.36	115.9	1.010
11.00	31.25	51.41	43.19	105.1	75.53	122.8	1.223
12.00	33.73	49.77	40.12	113.2	73.73	130.3	1.455
13.00	35.84	48.21	37.53	122.1	72.00	138.6	1.708
14.00	37.67	46.77	35.36	131.6	70.34	147.7	1.980
15.00	39.25	45.44	33.54	141.9	68.77	157.5	2.273
16.00	40.61	44.22	32.00	152.9	67.30	168.1	2.587
17.00	41.79	43.12	30.70	164.6	65.91	179.4	2.920
18.00	42.81	42.13	29.60	177.1	64.62	191.5	3.274
19.00	43.70	41.23	28.66	190.3	63.42	204.4	3.648
20.00	44.45	40.42	27.85	204.2	62.31	218.1	4.042
21.00	45.16	39.69	27.16	218.9	61.28	232.5	4.456
22.00	45.75	39.03	26.55	234.4	60.33	247.7	4.891
23.00	46.28	38.43	26.03	250.5	59.46	263.7	5.345
24.00	46.75	37.89	25.57	267.4	58.65	280.4	5.820
25.00	47.17	37.40	25.16	285.1	57.90	297.9	6.315
26.00	47.54	36.95	24.80	303.5	57.21	316.2	6.831
27.00	47.88	36.55	24.49	322.6	56.57	335.2	7.366
28.00	48.18	36.18	24.20	342.5	55.98	354.9	7.922
29.00	48.45	35.84	23.95	363.1	55.43	375.4	8.498
30.00	48.69	35.53	23.72	384.5	54.92	396.7	9.094
31.00	48.91	35.24	23.52	406.6	54.45	418.7	9.710
32.00	49.12	34.98	23.33	429.5	54.02	441.5	10.35
33.00	49.30	34.74	23.17	453.0	53.61	464.9	11.00
34.00	49.47	34.52	23.01	477.3	53.23	489.2	11.68
35.00	49.63	34.31	22.87	502.4	52.88	514.2	12.38
36.00	49.77	34.12	22.75	528.2	52.55	539.9	13.10
37.00	49.90	33.95	22.63	554.7	52.25	566.4	13.83
38.00	50.02	33.78	22.52	582.0	51.96	593.6	14.59
39.00	50.13	33.63	22.42	610.0	51.69	621.5	15.37
40.00	50.24	33.49	22.33	638.7	51.44	650.2	16.17
41.00	50.33	33.35	22.25	668.2	51.21	679.6	16.99
42.00	50.42	33.23	22.17	698.4	50.99	709.8	17.82
43.00	50.51	33.11	22.10	729.3	50.78	740.7	18.68
44.00	50.58	33.01	22.03	761.0	50.58	772.3	19.56
45.00	50.66	32.90	21.97	793.4	50.40	804.6	20.46
46.00	50.73	32.81	21.91	826.5	50.23	837.7	21.38
47.00	50.79	32.72	21.85	860.3	50.06	871.6	22.32
48.00	50.85	32.63	21.80	894.9	49.91	906.1	23.28
49.00	50.91	32.55	21.75	930.3	49.76	941.4	24.26
50.00	50.96	32.48	21.71	966.3	49.62	977.5	25.26
51.00	51.01	32.40	21.67	1003	49.49	1014	26.28

Listing of tape 33 for sample case (uncoupled characteristic roots).-

LONGITUDINAL STABILITY OF TETHERED BALLOON - LRC BALLOON-REFERENCE CONFIGURATION 11/20/72 09.56.50.

	UNCOUPLED ROOTS					
VEL	RLX1 IMX1	RLX2 IMX2	RLZ1 IMZ1	RLZ2 IMZ2	RLT1 IMT1	RLT2 IMT2
1.000	-1.3015E-02 .6839	-1.3015E-02 -.6839	-4.3199E-02 54.13	-4.3199E-02 -54.13	-.1169 73.82	-.1169 -73.82
2.000	-2.5993E-02 .6915	-2.5993E-02 -.6915	-8.5992E-02 13.96	-8.5992E-02 -13.96	-.2337 18.47	-.2337 -18.47
3.000	-3.8935E-02 .7043	-3.8935E-02 -.7043	-.1284 6.516	-.1284 -6.516	-.3504 8.267	-.3504 -8.267
4.000	-5.1863E-02 .7227	-5.1863E-02 -.7227	-.1710 3.913	-.1710 -3.913	-.4673 4.757	-.4673 -4.757
5.000	-6.4799E-02 .7472	-6.4799E-02 -.7472	-.2138 2.713	-.2138 -2.713	-.5842 3.190	-.5842 -3.190
6.000	-7.7749E-02 .7787	-7.7749E-02 -.7787	-.2572 2.066	-.2572 -2.066	-.7013 2.385	-.7013 -2.385
7.000	-9.0715E-02 .8175	-9.0715E-02 -.8175	-.3010 1.682	-.3010 -1.682	-.8186 1.928	-.8186 -1.928
8.000	-.1037 .8642	-.1037 -.8642	-.3451 1.438	-.3451 -1.438	-.9359 1.644	-.9359 -1.644
9.000	-.1167 .9187	-.1167 -.9187	-.3895 1.275	-.3895 -1.275	-1.053 1.447	-1.053 -1.447
10.00	-.1297 .9806	-.1297 -.9806	-.4339 1.162	-.4339 -1.162	-1.171 1.294	-1.171 -1.294
11.00	-.1427 1.050	-.1427 -1.050	-.4785 1.081	-.4785 -1.081	-1.288 1.160	-1.288 -1.160
12.00	-.1557 1.125	-.1557 -1.125	-.5231 1.022	-.5231 -1.022	-1.406 1.033	-1.406 -1.033
13.00	-.1687 1.207	-.1687 -1.207	-.5678 .9783	-.5678 -.9783	-1.523 .9018	-1.523 -.9018
14.00	-.1817 1.295	-.1817 -1.295	-.6124 .9461	-.6124 -.9461	-1.641 .7566	-1.641 -.7566
15.00	-.1947 1.388	-.1947 -1.388	-.6571 .9223	-.6571 -.9223	-1.758 .5816	-1.758 -.5816
16.00	-.2077 1.485	-.2077 -1.485	-.7017 .9052	-.7017 -.9052	-1.875 .3277	-1.875 -.3277
17.00	-.2207 1.586	-.2207 -1.586	-.7463 .8936	-.7463 -.8936	-1.643 0.	-2.342 0.
18.00	-.2337 1.692	-.2337 -1.692	-.7909 .8864	-.7909 -.8864	-1.518 0.	-2.703 0.
19.00	-.2467 1.800	-.2467 -1.800	-.8355 .8829	-.8355 -.8829	-1.466 0.	-2.990 0.
20.00	-.2597 1.912	-.2597 -1.912	-.8801 .8828	-.8801 -.8828	-1.445 0.	-3.245 0.
21.00	-.2727 2.026	-.2727 -2.026	-.9246 .8854	-.9246 -.8854	-1.443 0.	-3.483 0.
22.00	-.2857 2.143	-.2857 -2.143	-.9692 .8907	-.9692 -.8907	-1.453 0.	-3.708 0.
23.00	-.2986 2.262	-.2986 -2.262	-1.014 .8981	-1.014 -.8981	-1.472 0.	-3.923 0.
24.00	-.3116 2.383	-.3116 -2.383	-1.058 .9076	-1.058 -.9076	-1.496 0.	-4.132 0.
25.00	-.3246 2.506	-.3246 -2.506	-1.103 .9189	-1.103 -.9189	-1.530 0.	-4.334 0.

26.00	-.3376	-.3376	-1.147	-1.147	-1.567	-4.532
	2.629	-2.629	.9318	-.9318	0.	0.
27.00	-.3506	-.3506	-1.192	-1.192	-1.608	-4.726
	2.755	-2.755	.9462	-.9462	0.	0.
28.00	-.3636	-.3636	-1.236	-1.236	-1.652	-4.917
	2.881	-2.881	.9619	-.9619	0.	0.
29.00	-.3766	-.3766	-1.281	-1.281	-1.699	-5.105
	3.008	-3.008	.9789	-.9789	0.	0.
30.00	-.3896	-.3896	-1.325	-1.325	-1.749	-5.290
	3.136	-3.136	.9969	-.9969	0.	0.
31.00	-.4026	-.4026	-1.370	-1.370	-1.801	-5.472
	3.265	-3.265	1.016	-1.016	0.	0.
32.00	-.4156	-.4156	-1.414	-1.414	-1.855	-5.653
	3.394	-3.394	1.036	-1.036	0.	0.
33.00	-.4286	-.4286	-1.458	-1.458	-1.911	-5.832
	3.524	-3.524	1.057	-1.057	0.	0.
34.00	-.4416	-.4416	-1.503	-1.503	-1.969	-6.009
	3.654	-3.654	1.078	-1.078	0.	0.
35.00	-.4546	-.4546	-1.547	-1.547	-2.027	-6.185
	3.785	-3.785	1.100	-1.100	0.	0.
36.00	-.4676	-.4676	-1.592	-1.592	-2.087	-6.360
	3.916	-3.916	1.123	-1.123	0.	0.
37.00	-.4806	-.4806	-1.636	-1.636	-2.149	-6.534
	4.047	-4.047	1.147	-1.147	0.	0.
38.00	-.4936	-.4936	-1.681	-1.681	-2.211	-6.706
	4.178	-4.178	1.170	-1.170	0.	0.
39.00	-.5066	-.5066	-1.725	-1.725	-2.274	-6.878
	4.310	-4.310	1.195	-1.195	0.	0.
40.00	-.5196	-.5196	-1.769	-1.769	-2.338	-7.049
	4.441	-4.441	1.219	-1.219	0.	0.
41.00	-.5326	-.5326	-1.814	-1.814	-2.403	-7.219
	4.573	-4.573	1.245	-1.245	0.	0.
42.00	-.5455	-.5455	-1.858	-1.858	-2.468	-7.388
	4.705	-4.705	1.270	-1.270	0.	0.
43.00	-.5585	-.5585	-1.902	-1.902	-2.534	-7.557
	4.836	-4.836	1.296	-1.296	0.	0.
44.00	-.5715	-.5715	-1.947	-1.947	-2.600	-7.725
	4.968	-4.968	1.322	-1.322	0.	0.
45.00	-.5845	-.5845	-1.991	-1.991	-2.667	-7.893
	5.100	-5.100	1.348	-1.348	0.	0.
46.00	-.5975	-.5975	-2.036	-2.036	-2.735	-8.060
	5.231	-5.231	1.375	-1.375	0.	0.
47.00	-.6105	-.6105	-2.080	-2.080	-2.802	-8.227
	5.363	-5.363	1.401	-1.401	0.	0.
48.00	-.6235	-.6235	-2.124	-2.124	-2.871	-8.394
	5.495	-5.495	1.428	-1.428	0.	0.
49.00	-.6365	-.6365	-2.169	-2.169	-2.939	-8.560
	5.626	-5.626	1.456	-1.456	0.	0.
50.00	-.6495	-.6495	-2.213	-2.213	-3.008	-8.726
	5.758	-5.758	1.483	-1.483	0.	0.
51.00	-.6625	-.6625	-2.258	-2.258	-3.077	-8.892
	5.889	-5.889	1.510	-1.510	0.	0.

LATERAL STABILITY PROGRAM

The main program for the lateral stability program is the same as for the main program for the longitudinal stability program with the exception of the formats for labeling. In this case, however, rows 1, 2, and 3 of the matrices A, B, and C correspond to the coefficients of the y-, ϕ-, and ψ-equations, respectively. The general organization of subroutine INICOEF with entry point VCOEF is also similar to the organization of the longitudinal program. The only lateral tether spring is in the y-direction of the earth-axis system at the tether point (ref. 2) which is calculated in function subroutine YSUBY. The related springs about the balloon center of mass are then calculated from the y-spring. The FORTRAN variable names for the tether springs and their definitions are:

FORTRAN variable name	Mathematical symbol	Definition
SKPP	$k_{\phi\phi}$	spring constant for roll displacement
SKPS	$k_{\phi\psi}$	rolling moment due to yawing displacement
SKPY	$k_{\phi y}$	rolling moment due to y-displacement
SKSP	$k_{\psi\phi}$	yawing moment due to rolling displacement
SKSY	$k_{\psi y}$	yawing moment due to y-displacement
SKSS	$k_{\psi\psi}$	spring constant for yaw displacement
SKYP	$k_{y\phi}$	y-force due to roll displacement
SKYS	$k_{y\psi}$	y-force due to yaw displacement
SKYY	k_{yy}	spring constant for y-displacement

Trim or equilibrium conditions are calculated by the same procedures and subroutines as used in the longitudinal program. The lateral stability derivatives are defined about the reference point by user-written function subprograms and are transferred to the center of mass. The variable names for the derivatives about the center of mass are as follows:

FORTRAN variable name	Mathematical symbol	Definition
CLB	C_{l_β}	$\partial C_l / \partial \beta$
CLBD	$C_{l_{\dot{\beta}}}$	$\partial C_l \Big/ \partial \frac{\dot{\beta}\bar{c}}{2V}$
CLP	C_{l_p}	$\partial C_l \Big/ \partial \frac{p\bar{c}}{2V}$
CLR	C_{l_r}	$\partial C_l \Big/ \partial \frac{r\bar{c}}{2V}$
CNB	C_{n_β}	$\partial C_n / \partial \beta$
CNBD	$C_{n_{\dot{\beta}}}$	$\partial C_n \Big/ \partial \frac{\dot{\beta}\bar{c}}{2V}$
CNP	C_{n_p}	$\partial C_n \Big/ \partial \frac{p\bar{c}}{2V}$
CNR	C_{n_r}	$\partial C_n \Big/ \partial \frac{r\bar{c}}{2V}$
CYB	C_{Y_β}	$\partial C_Y / \partial \beta$
CYBD	$C_{Y_{\dot{\beta}}}$	$\partial C_Y \Big/ \partial \frac{\dot{\beta}\bar{c}}{2V}$
CYR	C_{Y_r}	$\partial C_Y \Big/ \partial \frac{r\bar{c}}{2V}$
CYP	C_{Y_p}	$\partial C_Y \Big/ \partial \frac{p\bar{c}}{2V}$

The limitations and diagnostics for the lateral program are essentially those given for the longitudinal program. In addition, the body reference axis for $\alpha_t = \theta = 0$ is assumed to be the principal axis for which $I_{XZ} = 0$ (where I_{XZ} is product of inertia).

Lateral Equations of Motion

The equations of lateral motion written about the center of mass are (see ref. 9):

y-force

$$\ddot{y} - \frac{\rho VS}{2m_y} C_{Y_\beta} \dot{y} + \frac{k_{yy}}{m_y} y - \frac{\rho VS\bar{c}}{4m_y} C_{Y_P} \dot{\phi} + \left(\frac{k_{y\phi}}{m_y} - \frac{\rho V^2 SC_L}{2m_y}\right)\phi$$

$$+ \left[\frac{\rho VS\bar{c}}{4m_y}\left(C_{Y_{\dot{\beta}}} - C_{Y_r}\right)\right]\dot{\psi} + \left[\frac{\rho V^2 S\left(C_{Y_\beta} + C_D\right)}{2m_y} + \frac{k_{y\psi}}{m_y}\right]\psi = 0 \tag{19}$$

Rolling moment

$$-\frac{\rho S\bar{c}^2}{4I_x} C_{l_{\dot{\beta}}} \ddot{y} - \frac{\rho VS\bar{c}}{2I_x} C_{l_\beta} \dot{y} + \frac{k_{\phi y}}{I_x} y + \ddot{\phi} - \frac{\rho VS\bar{c}^2}{4I_x} C_{l_p} \dot{\phi}$$

$$+ \frac{k_{\phi\phi} + h_{k_2} T_1 \sin\gamma_1 + M_{S_1}}{I_x}\phi - \frac{I_{xz}}{I_x}\ddot{\psi}$$

$$+ \left[\frac{\rho VS\bar{c}^2}{4I_x}\left(C_{l_{\dot{\beta}}} - C_{l_r}\right)\right]\dot{\psi} + \left[\frac{k_{\phi\psi}}{I_x} + \frac{\rho V^2 S\left(\bar{c}C_{l_\beta} - h_{k_2} C_D\right)}{2I_x}\right]\psi = 0 \tag{20}$$

Yawing moment

$$-\frac{\rho S\bar{c}^2}{4I_z} C_{n_{\dot{\beta}}} \ddot{y} - \frac{\rho VS\bar{c}}{2I_z} C_{n_\beta} \dot{y} + \frac{k_{\psi y}}{I_z} y - \frac{I_{xz}}{I_z}\ddot{\phi} - \frac{\rho VS\bar{c}^2}{4I_z} C_{n_p} \dot{\phi}$$

$$+ \frac{k_{\psi\phi} + M_{S_2} - h_{k_1} T_1 \sin\gamma_1}{I_z}\phi + \ddot{\psi} + \left[\frac{\rho VS\bar{c}^2}{4I_z}\left(C_{n_{\dot{\beta}}} - C_{n_r}\right)\right]\dot{\psi}$$

$$+ \left[\frac{k_{\psi\psi}}{I_z} + \frac{\rho V^2 S\left(\bar{c}C_{n_\beta} + h_{k_1} C_D\right)}{2I_z}\right]\psi = 0 \tag{21}$$

The definition of M_{S_1} is given by equation (6), M_{S_2} by equation (9), and

$$m_y = m_T + m_{y,a} - \frac{\rho S\bar{c}}{4} C_{Y_{\dot{\beta}}}$$

$$I_x = I_{xx}\cos^2\alpha_t + I_{zz}\sin^2\alpha_t$$

$$I_z = I_{zz} \cos^2\alpha_t + I_{xx} \sin^2\alpha_t$$

$$I_{xz} = -\frac{I_{zz} - I_{xx}}{2} \sin 2\alpha_t = \left(I_{xx} - I_{zz}\right) \sin \alpha_t \cos \alpha_t$$

The tether derivatives about the center of gravity are

$$k_{y\phi} = -h_{k_2} k_{yy} \tag{22a}$$

$$k_{y\psi} = h_{k_1} k_{yy} \tag{22b}$$

$$k_{\phi y} = k_{y\phi} \tag{22c}$$

$$k_{\phi\phi} = h_{k_2}^2 k_{yy} \tag{22d}$$

$$k_{\phi\psi} = k_{\psi\phi} \tag{22e}$$

$$k_{\psi y} = k_{y\psi} \tag{22f}$$

$$k_{\psi\phi} = -h_{k_1} h_{k_2} k_{yy} \tag{22g}$$

$$k_{\psi\psi} = h_{k_1}^2 k_{yy} \tag{22h}$$

and h_{k_1} and h_{k_2} are given by equations (8).

The spring constant k_{yy} is based on the analysis of reference 2 and is given by

$$k_{yy} = \frac{n\left(\tau_1 \sin^2\gamma_1 + 2\bar{p} \cos \gamma_1\right)^{1/2}}{\int_{\gamma_0}^{\gamma_1} \frac{\tau(\gamma)}{\sin^2\gamma + 2\bar{p} \cos \gamma} d\gamma} \tag{23}$$

The above expression is evaluated by function subprogram YSUBY which calls subroutine ROMBERG to evaluate the integral numerically.

The lateral stability derivatives are also given about the reference point and are transferred to the center of mass by the following relations:

$$C_{Y_r} = C_{Y_{\gamma,R}} - \frac{2x_t}{\bar{c}} C_{Y_\beta}$$

$$C_{Y_p} = C_{Y_{p,R}} + \frac{2z_t}{\bar{c}} C_{Y_\beta}$$

$$C_{n_\beta} = C_{n_{\beta,R}} - \frac{x_t}{\bar{c}} C_{Y_\beta}$$

$$C_{n_{\dot{\beta}}} = C_{n_{\dot{\beta},R}} - \frac{x_t}{\bar{c}} C_{Y_{\dot{\beta}}}$$

$$C_{n_r} = C_{n_{r,R}} - \frac{x_t}{\bar{c}} \left(C_{Y_{r,R}} + 2C_{n_\beta}\right)$$

$$C_{n_p} = C_{n_{p,R}} - \frac{x_t}{\bar{c}} C_{Y_{p,R}} + \frac{2z_t}{\bar{c}} C_{n_\beta}$$

$$C_{l_\beta} = C_{l_{\beta,R}} + \frac{z_t}{\bar{c}} C_{Y_\beta}$$

$$C_{l_{\dot{\beta}}} = C_{l_{\dot{\beta},R}} + \frac{z_t}{\bar{c}} C_{Y_{\dot{\beta}}}$$

$$C_{l_p} = C_{l_{p,R}} + \frac{z_t}{\bar{c}} \left(C_{Y_{p,R}} + 2C_{l_\beta}\right)$$

$$C_{l_r} = C_{l_{r,R}} + \frac{z_t}{\bar{c}} C_{Y_{r,R}} - \frac{2x_t}{\bar{c}} C_{l_\beta}$$

Input Required for Lateral Stability Program

The user-written function subprograms FCD, FCL, and FCMR for the longitudinal static aerodynamic coefficients as described for the longitudinal program are also required for the lateral stability program as the lateral program also calculates longitudinal trim conditions. In addition, the 12 lateral stability derivatives for the configuration are described as user-written function subroutines with trim angle of attack as a formal parameter. Each function is written about the reference point and is transferred to the center of mass by DERTRN. It might also be noted that the definitions of the lateral derivatives are conventional (ref. 11) except that they are based on the reference length $\bar{c}$ and the reference area S.

For each case, one card of 80 characters of case identification is read in an 8A10 format, and a namelist called LATDATA is read. The FORTRAN variable names, their equivalent mathematical symbols, and their definitions are listed in the NAMELIST which is also the order for printing. All variables are preset in the program with DATA statements to values for the reference configuration of the LRC balloon and only changes need to be read with the NAMELIST.

FORTRAN variable name	Mathematical symbol	Definition
CDINS	$C_{D_{ins}}$	constant increment of C_D (allows for C_D of instrument package of balloon)
DELCD	ΔC_D	constant increments in coefficients about center of mass which are used for parametric studies
DELCL	ΔC_L	
DELCM	ΔC_m	
DELCLB	ΔC_{l_β}	constant increments in lateral stability derivatives about center of mass which are used for parametric studies
DELCLBD	$\Delta C_{l_{\dot\beta}}$	
DELCLP	ΔC_{l_p}	
DELCLR	ΔC_{l_r}	
DELCNB	ΔC_{n_β}	
DELCNBD	$\Delta C_{n_{\dot\beta}}$	
DELCNP	ΔC_{n_p}	
DELCNR	ΔC_{n_r}	
DELCYB	ΔC_{Y_β}	
DELCYBD	$\Delta C_{Y_{\dot\beta}}$	
DELCYP	ΔC_{Y_p}	
DELCYR	ΔC_{Y_r}	

FORTRAN variable name	Mathematical symbol	Definition
RATIOKY		factor multiplying calculated y-spring, k_{yy} (SKYY); used for parametric studies
S	S	reference area, (Volume of balloon)$^{2/3}$
CBAR	$\bar{c}$	reference length, balloon body length used here
XXOI	I_{xx}	rolling inertia about axis through balloon center of mass parallel to body reference X-axis including aerodynamic apparent inertia, $\alpha_t = 0$
ZZOI	I_{zz}	yawing inertia about axis through balloon center of mass parallel to body reference Z-axis including aerodynamic apparent inertia, $\alpha_t = 0$
TMASS	m_T	mass of balloon structure and contained gas
AYMASS	$m_{y,a}$	aerodynamic apparent mass in body-reference y-axis direction, $\alpha_t = 0$
WTS	W_s	structural weight of balloon
BUOY	B	net buoyancy force
BHR	h_{br}	component of distance from reference point to center of buoyancy, positive for center of buoyancy below reference point (see fig. 2)
BLR	l_{br}	component of distance from reference point to center of buoyancy, positive for center of buoyancy forward of reference point (see fig. 2)
SHR	h_{sr}	component of distance from reference point to center of mass of balloon structure, positive for center of mass below reference point (see fig. 2)

FORTRAN variable name	Mathematical symbol	Definition
SLR	l_{sr}	component of distance from reference point to center of mass gravity of balloon structure, positive for center of mass aft of reference point (see fig. 2)
CGH	h_{cg}	component of distance from reference point to center of mass, positive for center of mass below reference point (see fig. 2)
CGL	l_{cg}	component of distance from reference point to center of mass, positive for center of mass forward of reference point (see fig. 2)
TLR	l_{tr}	component of distance from reference point to attachment point of tether line, positive for attachment point forward of reference point (see fig. 2)
TTR	t_{tr}	component of distance from reference point to attachment point of tether line, positive for attachment point below reference point (see fig. 2)
CLC	l_c	length of tether cable
CDIAM	d_c	diameter of tether cable
CDC	C_{D_c}	drag coefficient of tether cable based on diameter, i.e., drag of cable per unit length is $C_{D_c} d_c \rho V^2/2$
WC	w_c	weight per unit length of tether cable
RHO	ρ	ambient air density
VMIN	V_{min}	minimum wind velocity
DELV	ΔV	wind-velocity increment
NVEL		number of velocity calculations

Listing of Input Data Cards for Sample Case

```
******************************** COLUMN NUMBER ************************************
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890

LATERAL STABILITY OF TETHERED BALLOON - LRC BALLOON - REFERENCE CONFIGURATION
 $LATDATA VMIN=1., NVEL=51, DELV=1.$
```

```
      OVERLAY(STBLTY2,0,0)
      PROGRAM STBLTY2(INPUT=1,OUTPUT=1,TAPE5=INPUT,TAPE7,TAPE8=1,
     + TAPE11=1,TAPE30=1,TAPE31=1,TAPE32=1,TAPE33=1)
C***********************************************************************
C*                                                                     *
C* PROGRAM A2864.2 - LATERAL STABILITY OF TETHERED BALLOON             *
C*                                                                     *
C* THE FOLLOWING SUBROUTINES ARE CALLED BY THIS PROGRAM, BUT ARE LISTED *
C*   ONLY WITH THE LONGITUDINAL PROGRAM - ROMBERG,  QUADET,  MATRIX,    *
C*   REIG,  HESSEN,  AND  QRT.                                         *
C*                                                                     *
C* PROGRAM READS IDENTIFICATION CARD AND NAMELIST FROM INPUT FILE, AND *
C*   WRITES ONLY THE ID ARRAY FOR EACH CASE ON THE OUTPUT FILE         *
C* ALL FILES ARE BCD AND ARE SET TO MINIMUM BUFFER SIZE, EXCEPT TAPE7  *
C*   WHICH IS BINARY AND USES STANDARD BUFFER SIZE                     *
C* FILE ASSIGNMENTS ARE - TAPE7=PLOTTING PROGRAM INPUT, TAPE8=EIGEN-   *
C*   VECTORS, TAPE11=EIGENVALUES, TAPE30=AERODYNAMIC COEFFICIENTS,     *
C*   TAPE31=TETHER SPRINGS, TAPE32=TETHER CONDITIONS, AND TAPE33=      *
C*   UNCOUPLED ROOTS                                                   *
C*                                                                     *
C***********************************************************************
      COMMON/IROW/IROW(300)/ICOL/ICOL(300)
      DIMENSION A(3,3),B(3,3),C(3,3),SYMBOL(11),ID(10)
      DIMENSION EIDET(6,6),SAVE(6,7),ROOTR(6),ROOTI(6),INDEX(6)
     + ,IRUN(6),P(6),IPIV(3),INDX(3,2)
      COMPLEX EICOEF(3,3),CROT,CRTSQ,CDET
      DATA SYMBOL/110HCIRCLE    SQUARE    DIAMOND   TRIANGLE  RT TRNGL
     +QUADRANT  DOG HOUSE FAN       LNG DMND  HOUSE               /
      DATA RADEG,DELV,NVEL,VMIN/.01745329251994329б,.5,104,.5/       SI UNITS
  108 FORMAT(1H1///X10A10///)
  107 FORMAT(12X8G13.5)
  106 FORMAT(/* VELOCITY=*G13.5,2XA10)
  105 FORMAT(50X*EIGENVECTORS*//14X*COMPLEX ROOT-REAL,IMAG*5X
     + *Y/PSI, M/DEG-REAL,IMAG*3X*PHI/PSI,DEG/DEG-REAL,IMAG*6X       SI UNITS
     + *PSI,DEG-REAL,IMAG*/)
  104 FORMAT(8X,A10,6G16.6)
  103 FORMAT(/2X7G16.6)
  102 FORMAT(/* CONDITION NUMBER OF EIGENVALUE MATRIX=*E10.2/)
  101 FORMAT(/* CONDITION NUMBER OF A-MATRIX=*E10.2/)
  100 FORMAT(//* VELOCITY,(REAL(ROOT(I)),I=1,NPOS)*/* SYMBOL,(IMAG(ROOT(
     +I)),I=1,NPOS)*/)
      A11= 8H(X10A10) $ A10= 6H(8A10)
C
C  INITIALIZATION SECTION - READ IDENTIFICATION CARD, CALL DAYTIM FOR
C    DATE AND TIME, AND WRITE ID ARRAY ON BCD TAPES 8,11,30,31,32,33,
C    AND BINARY TAPE 7 WITH RECOUT.  DO NON-VELOCITY-DEPENDENT
C    CALCULATIONS WITH A CALL TO INICOEF
C  SEE SUBROUTINE WRITEUP FOR DESCRIPTION OF RECOUT
C
      NMP=3 $ NTWO=NMP+NMP $ NPL1=NTWO+1
      REWIND 30 $ REWIND 31 $ REWIND 32 $ REWIND 33
      REWIND 7 $ REWIND 8 $ REWIND 11
    1 READ A10,(ID(I),I=1,8) $ IF(EOF,5)999,2
    2 CALL DAYTIM(ID(9)) $ PRINT A11,ID $ WRITE(11,108)ID
      WRITE(30,108)ID $ WRITE(31,108)ID $ WRITE(32,108)ID
      WRITE(8,108)ID $ WRITE(8,105) $ WRITE(33,108)ID
      CALL RECOUT(7,2,0,ID,1,10,1)
      CALL INICOEF(A,B,C,NMP,VEL,VMIN,DELV,NVEL) $ WRITE(11,100)
      CALL RECOUT(7,1,0,NVEL)
C
C  90-LOOP IS VELOCITY VARIATION LOOP
C
      DO 90 IV=1,NVEL $ VEL=VMIN+(IV-1.)*DELV
```

```
C
C  SET UP COEFFICIENT MATRICES FOR QUADRATIC STABILITY DETERMINANT
C    WITH CALL TO ENTRY VCOEF OF SUBROUTINE INICOEF
C
      CALL VCOEF(A,B,C,NMP,VEL,VMIN,DELV,NVEL)
C
C  EXPAND QUADRATIC N X N STABILITY DETERMINANT INTO 2N X 2N STANDARD
C    EIGENVALUE FORM AND CHECK CONDITIONING OF MASS MATRIX A
C
      CALL QUADET(A,B,C, 3, 6,NMP,10,EIDET,CNOA)
      IF(CNOA.GT.1.E+4)WRITE(11,101)CNOA
C
C  EIGENVALUES FOR 2N SYSTEM AND CHECK CONDITIONING OF 2N X 2N MATRIX
C    WITH CALL TO MATRIX FOR INVERSE AND TURING CONDITION NUMBER
C
      CALL REIG(EIDET,NTWO,NTWO,0,ROOTR,ROOTI,EIVEC, 6,INDEX,IRUN,P,
     + NPL1,SAVE)
      CALL MATRIX(10,NTWO,NTWO,0,EIDET, 6,DETEI,KB,CNOEI)
      IF(CNOEI.GT.1.E+6)WRITE(11,102)CNOEI
C
C  ROOT SORTING - SORT COMPLEX ROOTS IN ORDER OF INCREASING MAGNITUDE OF
C    FREQUENCY AND DETERMINE THE NUMBER OF COMPLEX ROOTS WITH POSITIVE
C    VALUE OF FREQUENCY (IMAGINARY PART)
C
      NEGR=1 $ DO 50 NRT=1,NTWO $ NI=NTWO-NRT $ DO 48 J=1,NI
      IF(ROOTI(J)-ROOTI(J+1))48,46,46
   46 TRI=ROOTI(J) $ TRR=ROOTR(J) $ ROOTI(J)=ROOTI(J+1)
      ROOTR(J)=ROOTR(J+1) $ ROOTI(J+1)=TRI $ ROOTR(J+1)=TRR
   48 CONTINUE
   50 CONTINUE $ DO 52 NR=1,NTWO
      IF(ROOTI(NR).LT.-1.E-12)NEGR=NEGR+1
   52 CONTINUE
C
C  WRITE ROOTS ON TAPE 11
C
      IK1=IV/10  $  IK2=11  $  IF(IV.EQ.10*IK1) IK2=1+MOD(IK1-1,10)
      WRITE(11,103)VEL,(ROOTR(N),N=NEGR,NTWO)
      WRITE(11,104)SYMBOL(IK2),(ROOTI(N),N=NEGR,NTWO)
C
C  WRITE RESULTS ON BINARY TAPE 7 FOR INPUT TO PLOTTING PROGRAMS
C
      CALL RECOUT(7,1,0,VEL,IK2,NEGR,NTWO)
      CALL RECOUT(7,2,0,ROOTR,NEGR,NTWO,1)
      CALL RECOUT(7,2,0,ROOTI,NEGR,NTWO,1)
C
C  SETUP COEFFICIENT MATRICES FOR EIGENVECTOR (MODAL RATIOS) BY
C    DIVIDING BY PSI AND CALLING CXINV - RESULTS ON TAPE8
C
      WRITE(8,106)VEL,SYMBOL(IK2)
      DO 70 NE=NEGR,NTWO $ CROT=CMPLX(ROOTR(NE),ROOTI(NE))
      CRTSQ=CROT*CROT $ DO 60 IC=1,2 $ DO 60 IR=1,3
   60 EICOEF(IC,IR)=A(IC,IR)*CRTSQ+B(IC,IR)*CROT+C(IC,IR)
      DO 64 I=1,2
   64 EICOEF(I,3)=-RADEG*EICOEF(I,3)
      CALL CXINV(EICOEF,2,EICOEF(1,3),1,CDET,IPIV,INDX,3,ISC)
      EICOEF(3,3)=(1.,0.)
   70 WRITE(8,107)CROT,(EICOEF(I,3),I=1,3)
C
   90 CONTINUE $ GO TO 1
  999 ENDFILE 7 $ REWIND 7 $ ENDFILE 8 $ REWIND 8 $ REWIND 11
      REWIND 30 $ REWIND 31 $ REWIND 32 $ REWIND 33
      END             PROGRAM STBLTY2
```

```
      SUBROUTINE INICOEF(A,B,C,NMAX,VEL,VMIN,DELV,NVEL)
C
C  SUBROUTINE CALCULATES COEFFICIENT MATRICES FOR QUADRATIC STABILITY
C    DETERMINANT
C
      EQUIVALENCE(EQURT(1),UNCRT(1))
      DIMENSION A(NMAX,1),B(NMAX,1),C(NMAX,1),EQURT(1)
      COMPLEX UNCRT(6),CRAD,CSQRT
C
C  INPUT PARAMETERS ARE READ FROM THE INPUT FILE WITH A NAMELIST READ
C    OF THE NAMELIST LATDATA AND ARE WRITTEN ON TAPE 11 WITH A NAMELIST
C    WRITE STATEMENT
C
      NAMELIST/LATDATA/CDINS,DELCD,DELCL,DELCM,DELCLB,DELCLBD,DELCLP,
     + DELCLR,DELCNB,DELCNBD,DELCNP,DELCNR,DELCYB,DELCYBD,DELCYP,DELCYR,
     + RATIOKY,S,CBAR,XXOI,ZZOI,TMASS,AYMASS,WTS,BUOY,BHR,BLR,SHR,SLR,
     + CGH,CGL,TLR,TTR,CLC,CDIAM,CDC,WC,RHO,VMIN,DELV,NVEL
      COMMON/LONGDLC/CDINS,DELCD,DELCL,DELCM
C
C  PARAMETERS FOR LRC BALLOON - REFERENCE CONFIGURATION - IN SI UNITS
C
      DATA CDINS,DELCD,DELCL,DELCM,DELCLB,DELCLBD,DELCLP,DELCLR,DELCNB,
     + DELCNBD,DELCNP,DELCNR,DELCYB,DELCYBD,DELCYP,DELCYR/.01,15*0./
      DATA DEGRAD/57.2957795130823/,RATIOKY/1./
      DATA S,CBAR,XXOI,ZZOI/7.04,7.64,16.1,164./                        SI UNITS
      DATA TMASS,AYMASS,WTS,BUOY/14.2,23.9,108.,190./                   SI UNITS
      DATA BHR,BLR,SLR,TLR,TTR/0.,2.15,-.66,3.44,3.82/                  SI UNITS
      DATA SHR,CGH,CGL/.38,.109,1.10/                                   SI UNITS
      DATA CLC,CDIAM,CDC,WC/61.,.0141,1.17,.343/                        SI UNITS
      DATA RHO/1.225/                                                   SI UNITS
C
C  VELOCITY INDEPENDENT CALCULATIONS
C
      WRITE(30,101) $ WRITE(31,102) $ WRITE(32,103) $ WRITE(33,104)
      READ LATDATA $ WRITE(11,LATDATA)
      SL=SLR+CGL $ SH=SHR-CGH $ BL=BLR-CGL
      TL=TLR-CGL $ TT=TTR-CGH $ BH=CGH-BHR
      CBAR2=.5*CBAR $ ROSC2=RHO*S*CBAR2 $ ROSCS4=CBAR2*ROSC2
      A(1,1)=A(2,2)=A(3,3)=1. $ A(1,2)=A(1,3)=0. $ RETURN
C
C  ENTRY POINT VCOEF FOR VELOCITY-DEPENDENT CALCULATIONS
C
      ENTRY VCOEF
      Q=.5*RHO*VEL*VEL $ VCON=.5*VEL*ROSC2
C
C  TRIM ANGLE OF ATTACK AND AERODYNAMIC COEFFICIENTS ABOUT THE CENTER
C    OF MASS
C
      CALL TRIM(S,CBAR,WTS,BUOY,BL,BH,SL,SH,TL,TT,CGH,CGL,Q,
     + CL,CM,CD,CLA,CMA,CDA,ALPHA,XOC,ZOC,SINA,COSA)
      ALPHAD=DEGRAD*ALPHA
C
C  TRANSFER DYNAMIC STABILITY DERIVATIVES FROM REFERENCE POINT (MOMENT
C    CENTER) TO CENTER OF MASS
C
      CALL DERTRN(XOC,ZOC,ALPHA,CLB,CLBD,CLP,CLR,CNB,CNBD,CNP,CNR,CYB,
     + CYBD,CYR,CYP)
```

```
C
C   INCREMENT LATERAL STABILITY DERIVATIVES FOR TREND STUDIES AND WRITE
C     AERODYNAMIC COEFFICIENTS ON TAPE 30
C
      CLB=CLB+DELCLB   $  CLBD=CLBD+DELCLBD   $  CLP=CLP+DELCLP
      CNB=CNB+DELCNB   $  CNBD=CNBD+DELCNBD   $  CNP=CNP+DELCNP
      CYB=CYB+DELCYB   $  CYBD=CYBD+DELCYBD   $  CYP=CYP+DELCYP
      CLR=CLR+DELCLR   $  CNR=CNR+DELCNR   $  CYR=CYR+DELCYR
      WRITE(30,106)VEL,ALPHAD,CD,CL,CLB,CLBD,CLP,CLR,CNB,CNBD,CNP,CNR,
     + CYB,CYBD,CYP,CYR


C
C   CALCULATE EQUILIBRIUM CABLE CONDITIONS AND WRITE RESULTS ON TAPE 32
C
      DRAG=CD*Q*S $ BLIFT=CL*Q*S $ CDRAG=CDC*CDIAM*Q
      T1=SQRT(DRAG*DRAG+(BLIFT-WTS+BUOY)**2)
      COSG1=DRAG/T1 $ GAM1=ACOS(COSG1) $ TSG1=T1*SIN(GAM1)
      CALL TETHER(CDRAG,WC,CLC,T1,GAM1,TO,GAMO,X1,Z1)
      GAM1D=DEGRAD*GAM1 $ GAMOD=DEGRAD*GAMO
      WRITE(32,100)VEL,X1,Z1,GAMOD,TO,GAM1D,T1,CDRAG
C
C   CALCULATE WEIGHT-BOUYANCY MOMENT TERM AND MASS TERMS INCLUDING
C     APPARENT MASS ROTATION TO STABILITY AXES
C
      HK1=TL*COSA+TT*SINA $ HK2=TT*COSA-TL*SINA
      RMYT=1./(TMASS+AYMASS-.5*ROSC2*CYBD)
      RIX=1./(XXOI*COSA*COSA+ZZOI*SINA*SINA) $ XZI=(XXOI-ZZOI)*SINA*COSA
      RIZ=1./(ZZOI*COSA*COSA+XXOI*SINA*SINA)
      SM1=(BL*BUOY+SL*WTS)*SINA+(BH*BUOY+SH*WTS)*COSA
      SM2=(BL*BUOY+SL*WTS)*COSA-(BH*BUOY+SH*WTS)*SINA
C
C   CALCULATE CABLE SPRINGS FROM DERIVATIVES OF NEUMARK AND TRANSFER TO
C     STABILITY AXES - WRITE RESULTS ON TAPE 31
C
      SKYY=RATIOKY*YSLBY(CDRAG,GAMC,GAM1)
      SKYP=SKPY=-HK2*SKYY $ SKPS=SKSP=HK1*SKYP
      SKYS=SKSY=HK1*SKYY $ SKPP=-HK2*SKYP $ SKSS=HK1*SKYS
      WRITE(31,100)VEL,SKYY,SKYP,SKYS,SKPP,SKPS,SKSS
C
C   CALCULATE COEFFICIENT MATRICES A , B, AND C
C
      A(2,1)=-RIX*ROSCS4*CLBD $ A(2,3)=-XZI*RIX
      A(3,1)=-RIZ*ROSCS4*CNBD $ A(3,2)=-XZI*RIZ
      B(1,1)=-VCON/CBAR2*RMYT*CYB $  B(1,2)=-VCON*RMYT*CYP
      B(1,3)=-VCON*RMYT*(CYR-CYBD) $ B(2,1)=-2.*VCON*RIX*CLB
      B(2,2)=-VCON*CBAR*RIX*CLP $ B(2,3)=-VCON*CBAR*RIX*(CLR-CLBD)
      B(3,1)=-2.*VCON*RIZ*CNB $ B(3,2)=-VCON*CBAR*RIZ*CNP
      B(3,3)=-VCON*CBAR*RIZ*(CNR-CNBD)
      C(1,1)=SKYY*RMYT $ C(1,2)=RMYT*(SKYP-Q*S*CL)
      C(1,3)=RMYT*(SKYS+Q*S*(CYB+CD)) $ C(2,1)=RIX*SKPY
      C(2,2)=RIX*(SKPP+HK2*TSG1+SM1)
      C(2,3)=RIX*(SKPS+Q*S*(CBAR*CLB-HK2*CD)) $ C(3,1)=SKSY*RIZ
      C(3,2)=RIZ*(SKSP-HK1*TSG1+SM2)
      C(3,3)=RIZ*(SKSS+Q*S*(CBAR*CNB+HK1*CD))
C
C   CALCULATE UNCOUPLED ROOTS BY FACTORING DIAGONAL QUADRATIC TERMS AND
C     WRITE RESULTS ON TAPE 33
C
      DO 1 M=1,3 $ CRAD=.25*B(M,M)*B(M,M)-C(M,M) $ CRAD=CSQRT(CRAD)
      M2=2*M $ M1=M2-1 $ UNCRT(M1)=-.5*B(M,M)+CRAD
    1 UNCRT(M2)=-.5*B(M,M)-CRAD
      WRITE(33,105)VEL,(EQURT(I),I=1,11,2),(EQURT(I),I=2,12,2) $ RETURN
```

```
C
  100 FORMAT(2X11G11.4)
  101 FORMAT(/20X*AERODYNAMIC COEFFICIENTS*//* VELOCITY*5X*ALPHAD*9X*CD*
     + 9X*CL*8X*CLB*8X*CLBD*8X*CLP*8X*CLR*8X*CNB*7X*CNBD*8X*CNP*/
     + 30X*CNR*8X*CYB*7X*CYBD*8X*CYP*8X*CYR*/)
  102 FORMAT(/20X*TETHER SPRINGS*/* VELOCITY*7X*SKYY*7X*SKYP*7X*SKYS*7X
     + *SKPP*7X*SKPS*7X*SKSS*)

  103 FORMAT(/20X*TETHER CONDITIONS*/* VELOCITY*9X*X1*9X*Z1*7X*GAMO*9X
     + *TO*7X*GAM1*9X*T1*3X*CAB DRAG*)
  104 FORMAT(/20X*UNCOUPLED ROOTS*/7X*VEL*8X*RLY1*8X*RLY2*8X*RLP1*8X
     + *RLP2*8X*RLS1*8X*RLS2*/18X*IMY1*8X*IMY2*8X*IMP1*8X*IMP2*8X
     + *IMS1*8X*IMS2*)
  105 FORMAT(/2X7G12.4/14X6G12.4)
  106 FORMAT(/2X11G11.4/24X9G11.4)
      END              SUBROUTINE INICOEF


      SUBROUTINE TRIM(S,CBAR,WTS,BUOY,BL,BH,SL,SH,TL,TT,CGH,CGL,Q,
     + CL,CM,CD,CLA,CMA,CDA,ALPHA,F,G,SA,CA)
C
C  SUBROUTINE COMPUTES THE STATIC TRIM ANGLE-OF-ATTACK  ALPHA  USING
C    NEWTON ITERATION OF THE TRIM EQUATION
C  THE ALPHA DEPENDENT DERIVATIVES CD, CDA, CL, CLA, CM, AND CMA ARE
C    ALSO TRANSFERRED TO THE CENTER OF MASS AND RETURNED
C  IF CONVERGENCE IS NOT OBTAINED IN ITCMAX ITERATIONS, MESSAGE IS
C    WRITTEN ON TAPE 11
C
      COMMON/LONGDLC/CDINS,DELCD,DELCL,DELCM
      ERR=1.E-6 $ TLPHA=.05 $ QS=Q*S $ ITCMAX=100 $ ITC=0
      D=BUOY*BL+WTS*SL $ E=BUOY*BH+WTS*SH
    1 ALPHA=TLPHA $ CL=FCL(ALPHA,CLA) $ CD=FCD(ALPHA,CDA)
      CMR=FCMR(ALPHA,CMAR) $ CA=COS(ALPHA) $ SA=SIN(ALPHA)
      A=TL*CA+TT*SA $ B=TT*CA-TL*SA
      F=(CGL*CA+CGH*SA)/CBAR $ G=(CGH*CA-CGL*SA)/CBAR
      CM=CMR-F*CL+G*CD $ CMA=CMAR-F*(CLA+CD)-G*(CL-CDA)
      CL=CL+DELCL   $  CM=CM+DELCM   $  C=BUOY-WTS+QS*CL
      CD=CD+DELCD+CDINS
      FUN=A*C-(B*CD+CM*CBAR)*QS-D*CA+E*SA
      DFUN=B*C+D*SA+E*CA+(A*(CLA+CD)-B*CDA-CMA*CBAR)*QS
      TLPHA=ALPHA-FUN/DFUN $ ITC=ITC+1 $ IF(ITC.GT.ITCMAX)GO TO 2
      IF(ABS(ALPHA-TLPHA).GT.ERR)1,4
    2 WRITE(11,3)ITC,ALPHA,TLPHA
    3 FORMAT(/*  ITERATION FOR TRIM DID NOT CONVERGE IN*I6* ITERATIONS.
     + ALPHA=*G12.6* TALPHA=*G12.6/)
    4 RETURN
      END              SUBROUTINE TRIM
```

```
      FUNCTION FCD(A,CDA)
C
C  FCD IS A FUNCTION TO BE WRITTEN BY THE USER TO CALCULATE CD AND CDA
C    AS FUNCTIONS OF ANGLE OF ATTACK FOR THE CONFIGURATION TO BE ANALYZED
C
C  CURVE FIT OF CD AND CDA VS ANGLE OF ATTACK IN RADIANS FOR LRC
C    BALLOON - REFERENCE CONFIGURATION
C
      X=A-.023 $ X5=X**5 $ CDA=1117.2*X5
      FCD=.0487+186.2*X*X5  $  RETURN  $  END


      FUNCTION FCL(A,CLA)
C
C  FCL IS A FUNCTION TO BE WRITTEN BY THE USER TO CALCULATE CL AND CLA
C    AS FUNCTIONS OF ANGLE OF ATTACK FOR THE CONFIGURATION TO BE ANALYZED
C
C  CURVE FIT OF CL AND CLA VS ANGLE OF ATTACK IN RADIANS FOR LRC
C    BALLOON - REFERENCE CONFIGURATION
C
      X=A-.023 $ X2=X*X $ CLA=.82-X2*(15.069-557.0*X2)
      FCL=X*(.82-X2*(5.023-111.4*X2))  $  RETURN  $  END


      FUNCTION FCMR(A,CMAR)
C
C  FCMR IS A FUNCTION TO BE WRITTEN BY THE USER TO CALCULATE CMR AND CMAR
C    AS FUNCTIONS OF ANGLE OF ATTACK FOR THE CONFIGURATION TO BE ANALYZED
C
C  CURVE FIT OF CM AND CMA ABOUT REFERENCE POINT VS ANGLE OF ATTACK IN
C    RADIANS FOR LRC BALLOON - REFERENCE CONFIGURATION
C
      CMAR=.1435 $ FCMR=-.0106+.1435*A  $  RETURN  $  END


      FUNCTION FCLBO(A)
C
C  FUNCTIONS FCLBO, ETC, ARE FUNCTIONS TO BE WRITTEN BY THE USER TO
C    CALCULATE THE LATERAL STABILITY DERIVATIVES AS FUNCTIONS OF ANGLE
C    OF ATTACK FOR THE CONFIGURATION TO BE ANALYZED.  FUNCTIONS ARE
C    REFERENCED ONLY BY SUBROUTINE DERTRN.
C
C  FUNCTIONS FCLBO, ETC, GIVE THE LATERAL STABILITY DERIVATIVES VS ANGLE
C    OF ATTACK IN RADIANS FOR THE LRC BALLOON - REFERENCE CONFIGURATION.
C
      FCLBO=-.1435*SIN(A)  $  RETURN  $  END


      FUNCTION FCLBDO(A)
      FCLBDO=0.  $  RETURN  $  END


      FUNCTION FCLPO(A)
      FCLPO=-.0237  $  RETURN  $  END


      FUNCTION FCLRO(A)
      FCLRO=-.178*SIN(A)  $  RETURN  $  END
```

```
      FUNCTION FCNBO(A)
      FCNBO=-.1435  $  RETURN  $  END


      FUNCTION FCNBDO(A)
      FCNBDO=.026  $  RETURN  $  END


      FUNCTION FCNPO(A)
      FCNPO=-.0641*SIN(2.*A)  $  RETURN  $  END


      FUNCTION FCNRO(A)
      FCNRO=-.189  $  RETURN
      END


      FUNCTION FCYB(A)
      FCYB=-.82  $  RETURN  $  END


      FUNCTION FCYBD(A)
      FCYBD=-.089  $  RETURN  $  END


      FUNCTION FCYPO(A)
      FCYPO=.494*SIN(A)  $  RETURN  $  END


      FUNCTION FCYRO(A)
      FCYRO=.685  $  RETURN  $  END


      SUBROUTINE DERTRN(X,Z,A,CLB,CLBD,CLP,CLR,CNB,CNBD,CNP,CNR,CYB,
     + CYBD,CYR,CYP)
C
C  TRANSFERS ALL 12 LATERAL DERIVATIVES FROM REFERENCE POINT (MOMENT
C    CENTER) TO CENTER OF MASS LOCATED X/C FORWARD AND Z/C DOWN FROM
C    REFERENCE POINT
C
      CYB=FCYB(A) $ CYBD=FCYBD(A) $ CYPO=FCYPO(A) $ CYRO=FCYRO(A)
      CYP=CYPO+2.*Z*CYB $ CYR=CYRO-2.*X*CYB $ CNB=FCNBO(A)-X*CYB
      CNBD=FCNBDO(A)-X*CYBD $ CNP=FCNPO(A)-X*CYPO+2.*Z*CNB
      CNR=FCNRO(A)-X*(CYRO+2.*CNB) $ CLB=FCLBO(A)+Z*CYB
      CLBD=FCLBDO(A)+Z*CYBD $ CLP=FCLPO(A)+Z*(CYPO+2.*CLB)
      CLR=FCLRO(A)+Z*CYRO-2.*X*CLB $ RETURN
      END            SUBROUTINE DERTRN
```

```
      SUBROUTINE TETHER(CDRAG,WC,CL1,T1,GAMMA1,TO,GAMMAO,X1,Z1)
C
C   SUBROUTINE IS BASED ON THE ANALYSIS IN -
C     NEUMARK, S.-  EQUILIBRIUM CONFIGURATIONS OF FLYING CABLES OF
C       CAPTIVE BALLOONS, AND CABLE DERIVATIVES FOR STABILITY
C       CALCULATIONS.  BRITISH R. AND M. NO. 3333, 1963.
C   THE TETHER PARAMETERS CDRAG, WC, CL1, T1, AND GAMMA1 ARE INPUT
C   THE TETHER PARAMETERS TO, GAMMAO, X1, AND Z1 ARE OUTPUT
C
      EXTERNAL FLAM,FSIG $ COMMON/PC/P,Q
      P=.5*WC/CDRAG $ C=SQRT(1+P*P) $ EPS=1.E-8
      CALL ROMBERG(RLAM1,0.,GAMMA1,FLAM,EPS) $ TAU1=TAU(GAMMA1)
      RLAMO=RLAM1-CDRAG*TAU1*CL1/T1 $ CALL NEWINT(RLAMO,FLAM,1.,GAMMAO)
      CALL ROMBERG(DSIG,GAMMAO,GAMMA1,FSIG,EPS)
      X1=T1*DSIG/(CDRAG*TAU1) $ TAUO=TAU(GAMMAO)
      TO=T1*TAUO/TAU1 $ Z1=(T1-TO)/WC $ RETURN $ END


      FUNCTION FLAM(T) $ COMMON/PC/P,Q $ CT=COS(T)
      FLAM=((Q+P-CT)/(Q-P+CT))**(P/Q)/(1-CT*CT+2*P*CT) $ RETURN $ END


      FUNCTION FSIG(T) $ COMMON/PQ/P,Q $ CT=COS(T)
      FSIG=((Q+P-CT)/(Q-P+CT))**(P/Q)/(1-CT*CT+2*P*CT)*CT $ RETURN $ END


      FUNCTION TAU(T) $ COMMON/PQ/P,Q $ CT=COS(T)
      TAU=((Q+P-CT)/(Q-P+CT))**(P/Q) $ RETURN $ END


      FUNCTION YSUBY(CDRAG,GAMMAO,GAMMA1) $ COMMON/PC/P,Q
      EXTERNAL FTHE $ EPS=1.E-8 $ CT=COS(GAMMA1)
      CALL ROMBERG(DTHE,GAMMAO,GAMMA1,FTHE,EPS)
      YSUBY=CDRAG*SQRT(TAU(GAMMA1)*(1-CT*CT+2*P*CT))/DTHE $ RETURN $ END


      FUNCTION FTHE(T) $ COMMON/PC/P,Q $ CT=COS(T)
      FTHE=SQRT(((Q+P-CT)/(Q-P+CT))**(P/Q)/(1-CT*CT+2*P*CT)) $  RETURN
      END             SUBPROGRAM FTHE


      SUBROUTINE NEWINT(C,F,XO,X) $ EPS=1.E-8 $ XT=XO $ I=0
C
C   SUBROUTINE COMPUTES THE UPPER LIMIT  X  OF THE DEFINITE INTEGRAL
C     FROM  0 TO X  OF THE FUNCTION  F  FOR WHICH THE VALUE OF THE
C     INTEGRAL  C  IS KNOWN
C   NEWTON ITERATION IS USED WITH THE VALUE OF THE INTEGRAL DETERMINED
C     BY SUBROUTINE ROMBERG
C
    1 X=XT $ CALL ROMBERG(S,0.,X,F,EPS) $ XT=X+(C-S)/F(X) $ I=I+1
      PI2=1.570796326 $ IF(XT.GT.PI2) XT=PI2 $ IF(XT.LT.0.) XT=0.
      IF(I.LT.2) GO TO 1 $ IF(ABS(X-XT).GT.EPS) GO TO 1
      X=XT $ RETURN $ END
```

Printing of Files Containing Calculated Results for Sample Case

A printing of each BCD file for the sample case is given. The headings essentially give the quantities printed using the FORTRAN variable names previously given. The lateral program uses the same files as does the longitudinal program. The information written on tapes 30 and 32 by the lateral program is identical to that written by the longitudinal program. Tapes 8, 11, 31, and 33 are also closely paralleled to the corresponding longitudinal files. Note that the headings PSI and PHI are used to denote ψ and ϕ. Also,the letters P, S, and Y are used to denote ϕ, ψ, and y in the headings on tapes 31 and 33.

Listing of OUTPUT for sample case (identification card, and date and time only).-

```
LATERAL STABILITY OF TETHERED BALLOON - LRC BALLOON - REFERENCE CONFIGURATION     11/17/72  00.33.53.
```

Listing of tape 11 for sample case (principal file for characteristic roots).

```
$LATDATA
CDINS    =  0.1E-01,
DELCD    =  0.0,
DELCL    =  0.0,
DELCM    =  0.0,
DELCLB   =  0.0,
DELCLBD  =  0.0,
DELCLP   =  0.0,
DELCLR   =  0.0,
DELCNB   =  0.0,
DELCNBD  =  0.0,
DELCNP   =  0.0,
DELCNR   =  0.0,
DELCYB   =  0.0,
DELCYBD  =  0.0,
DELCYP   =  0.0,
DELCYR   =  0.0,
RATIOKY  =  0.1E+01,
S        =  0.704E+01,
CBAR     =  0.764E+01,
XXOI     =  0.161E+02,
ZZOI     =  0.164E+03,
TMASS    =  0.142E+02,
AYMASS   =  0.239E+02,
WTS      =  0.108E+03,
BUOY     =  0.19E+03,
BHR      =  0.0,
BLR      =  0.215E+01,
SHR      =  0.38E+00,
SLR      = -0.66E+00,
CGH      =  0.109E+03,
CGL      =  0.11E+01,
TLR      =  0.344E+01,
TTR      =  0.382E+01,
CLC      =  0.61E+02,
CDIAM    =  0.141E-01,
CDC      =  0.117E+01,
WC       =  0.343E+00,
RHC      =  0.1225E+01,
VMIN     =  0.1E+01,
DELV     =  0.1E+01,
NVEL     =  51,
$END
```

```
VELOCITY,(REAL(ROOT(I)),I=1,NPOS)
SYMBCL,(IMAG(ROOT(I)),I=1,NPOS)
```

1.CC000	-.135178	-2.813337E-02	-8.429729E-02	
	6.892606E-02	.265843	4.78413	
2.C0000	-.266458	-.374386	-5.864670E-03	-.169685
	0.	0.	.257232	4.81727
3.00000	-.306809	-.670208	-2.078607E-04	-.256765
	0.	0.	.269696	4.86837
4.CC000	-.360604	-.934484	-2.988420E-03	-.345582
	0.	0.	.282354	4.93369
5.00000	-.414394	-1.19071	-9.198426E-03	-.435867
	0.	0.	.293780	5.01052
6.CC000	-.467727	-1.44349	-1.681126E-02	-.527282
	0.	0.	.304180	5.09736
7.00000	-.520878	-1.69461	-2.482685E-02	-.619547
	0.	0.	.313688	5.19349
8.00000	-.574105	-1.94497	-3.270870E-02	-.712468
	0.	0.	.323148	5.29862
9.00000	-.627562	-2.19504	-4.017218E-02	-.805916
	0.	0.	.332093	5.41256
1C.0000	-.681313	-2.44507	-4.708165E-02	-.899806
CIRCLE	0.	0.	.340765	5.53512
11.0000	-.735369	-2.69520	-5.339067E-02	-.994079
	0.	0.	.349153	5.66603
12.0000	-.789713	-2.94550	-5.910478E-02	-1.08869
	0.	0.	.357222	5.80497
13.CC00	-.844317	-3.19599	-6.425813E-02	-1.18359
	0.	0.	.364933	5.95156
14.C000	-.899152	-3.44668	-6.889910E-02	-1.27876
	C.	0.	.372251	6.10540
15.0000	-.954191	-3.69756	-7.308160E-02	-1.37415
	0.	0.	.379153	6.26605
16.C000	-1.00941	-3.94859	-7.686002E-02	-1.46974
	0.	0.	.385629	6.43309
17.CCC0	-1.06480	-4.19976	-8.028631E-02	-1.56550
	0.	0.	.391677	6.60608
18.0000	-1.12033	-4.45104	-8.340854E-02	-1.66140
	0.	0.	.397303	6.78463
19.CCC0	-1.17601	-4.70241	-8.627016E-02	-1.75742
	0.	0.	.402521	6.96832
20.C000	-1.23181	-4.95386	-8.890988E-02	-1.85353
SQUARE	0.	0.	.407349	7.15680
21.0000	-1.28772	-5.20536	-9.136179E-02	-1.94973
	0.	0.	.411807	7.34972
22.C000	-1.34375	-5.45691	-9.365563E-02	-2.04598
	C.	0.	.415916	7.54673
23.0000	-1.39989	-5.70848	-9.581719E-02	-2.14229
	0.	0.	.419699	7.74755
24.0000	-1.45612	-5.96006	-9.786871E-02	-2.23863
	0.	0.	.423179	7.95189
25.0C00	-1.51244	-6.21166	-9.982928E-02	-2.33499
	C.	0.	.426376	8.15948

26.0000	-1.56885	-6.46325	-.101715	-2.43138
	0.	0.	.429313	8.37010
27.0000	-1.62534	-6.71484	-.103540	-2.52777
	0.	0.	.432008	8.58352
28.0000	-1.68191	-6.96642	-.105317	-2.62417
	0.	0.	.434480	8.79954
29.0000	-1.73856	-7.21798	-.107054	-2.72057
	0.	0.	.436746	9.01798
30.0000	-1.79527	-7.46953	-.108761	-2.81696
DIAMOND	0.	0.	.438821	9.23866
31.0000	-1.85204	-7.72106	-.110445	-2.91335
	0.	0.	.440721	9.46142
32.0000	-1.90888	-7.97257	-.112112	-3.00972
	0.	0.	.442458	9.68613
33.0000	-1.96578	-8.22406	-.113766	-3.10608
	0.	0.	.444045	9.91265
34.0000	-2.02272	-8.47552	-.115413	-3.20243
	0.	0.	.445494	10.1409
35.0000	-2.07973	-8.72696	-.117055	-3.29876
	0.	0.	.446814	10.3706
36.0000	-2.13677	-8.97838	-.118696	-3.39508
	0.	0.	.448014	10.6019
37.0000	-2.19387	-9.22978	-.120337	-3.49137
	0.	0.	.449104	10.8345
38.0000	-2.25101	-9.48115	-.121982	-3.58765
	0.	0.	.450091	11.0685
39.0000	-2.30818	-9.73250	-.123632	-3.68391
	0.	0.	.450982	11.3036
40.0000	-2.36540	-9.98382	-.125288	-3.78015
TRIANGLE	0.	0.	.451784	11.5399
41.0000	-2.42265	-10.2351	-.126950	-3.87638
	0.	0.	.452502	11.7773
42.0000	-2.47994	-10.4864	-.128621	-3.97258
	0.	0.	.453142	12.0156
43.0000	-2.53726	-10.7377	-.130301	-4.06877
	0.	0.	.453709	12.2549
44.0000	-2.59461	-10.9889	-.131990	-4.16494
	0.	0.	.454208	12.4951
45.0000	-2.65199	-11.2401	-.133688	-4.26109
	0.	0.	.454642	12.7362
46.0000	-2.70940	-11.4913	-.135397	-4.35723
	0.	0.	.455015	12.9780
47.0000	-2.76683	-11.7425	-.137115	-4.45335
	0.	0.	.455331	13.2206
48.0000	-2.82429	-11.9937	-.138844	-4.54945
	0.	0.	.455593	13.4639
49.0000	-2.88177	-12.2449	-.140583	-4.64554
	0.	0.	.455804	13.7078
50.0000	-2.93928	-12.4960	-.142332	-4.74161
RT TRNGL	0.	0.	.455967	13.9525
51.0000	-2.99680	-12.7471	-.144092	-4.83766
	0.	0.	.456083	14.1977

Listing of tape 8 for sample case (modal ratios or eigenvectors and characteristic roots).-

LATERAL STABILITY OF TETHERED BALLOON - LRC BALLOON - REFERENCE CONFIGURATION 11/17/72 00.33.53.

EIGENVECTORS

	COMPLEX ROOT-REAL,IMAG		Y/PSI, M/DEG-REAL,IMAG		PHI/PSI,DEG/DEG-REAL,IMAG		PSI,DEG-REAL,IMAG	
VELOCITY=	1.0000							
	-.13518	6.89261E-02	-3.36829E-02	2.93396E-03	2.38689E-04	4.22723E-05	1.0000	0.
	-2.81334E-02	.26584	1.23671E-02	-4.20748E-02	7.79867E-04	-4.75541E-04	1.0000	0.
	-8.42973E-02	4.7841	6.00912E-04	-6.15610E-04	-.12250	1.98372E-03	1.0000	0.
VELOCITY=	2.0000							
	-.26646	0.	8.86002E-03	0.	8.90943E-04	0.	1.0000	0.
	-.37439	0.	-1.11113E-02	0.	6.27148E-04	0.	1.0000	0.
	-5.86467E-03	.25723	-1.24897E-03	-9.95209E-02	7.61205E-04	-1.13124E-03	1.0000	0.
	-.16968	4.8173	6.43908E-04	-1.24825E-03	-.12511	3.99629E-03	1.0000	0.
VELOCITY=	3.0000							
	-.30681	0.	.11976	0.	2.56077E-03	0.	1.0000	0.
	-.67021	0.	-1.06071E-02	0.	8.09115E-04	0.	1.0000	0.
	-2.07861E-04	.26970	-1.18541E-02	-.15509	8.54964E-04	-1.77141E-03	1.0000	0.
	-.25676	4.8684	7.02602E-04	-1.91073E-03	-.12881	6.04351E-03	1.0000	0.
VELOCITY=	4.0000							
	-.36060	0.	.36950	0.	6.30787E-03	0.	1.0000	0.
	-.93448	0.	-1.00568E-02	0.	1.05533E-03	0.	1.0000	0.
	-2.98842E-03	.26235	-2.41561E-02	-.20809	9.93299E-04	-2.40028E-03	1.0000	0.
	-.34558	4.9337	7.62738E-04	-2.60910E-03	-.13289	8.09989E-03	1.0000	0.
VELOCITY=	5.0000							
	-.41439	0.	1.2058	0.	1.87768E-02	0.	1.0000	0.
	-1.1907	0.	-9.78014E-03	0.	1.35051E-03	0.	1.0000	0.
	-9.19843E-03	.29378	-3.82481E-02	-.25873	1.16252E-03	-3.03290E-03	1.0000	0.
	-.43587	5.0105	8.10678E-04	-3.34236E-03	-.13675	1.01169E-02	1.0000	0.
VELOCITY=	6.0000							
	-.46773	0.	-19.884	0.	-.29479	0.	1.0000	0.
	-1.4435	0.	-9.62444E-03	0.	1.68677E-03	0.	1.0000	0.
	-1.68113E-02	.30418	-5.36271E-02	-.30684	1.35542E-03	-3.68033E-03	1.0000	0.
	-.52728	5.0974	8.35310E-04	-4.10384E-03	-.14000	1.20391E-02	1.0000	0.
VELOCITY=	7.0000							
	-.52088	0.	-1.8665	0.	-2.68670E-02	0.	1.0000	0.
	-1.6946	0.	-9.52921E-03	0.	2.05496E-03	0.	1.0000	0.
	-2.48269E-02	.31389	-6.96787E-02	-.35231	1.56591E-03	-4.35182E-03	1.0000	0.
	-.61955	5.1935	8.28839E-04	-4.88316E-03	-.14247	1.38189E-02	1.0000	0.
VELOCITY=	8.0000							
	-.57411	0.	-1.2025	0.	-1.69993E-02	0.	1.0000	0.
	-1.9450	0.	-9.46765E-03	0.	2.44563E-03	0.	1.0000	0.
	-3.27087E-02	.32315	-8.58344E-02	-.39517	1.78808E-03	-5.05451E-03	1.0000	0.
	-.71247	5.2986	7.86895E-04	-5.66836E-03	-.14413	1.54243E-02	1.0000	0.
VELOCITY=	9.0000							
	-.62756	0.	-.97593	0.	-1.36520E-02	0.	1.0000	0.
	-2.1950	0.	-9.42631E-03	0.	2.84973E-03	0.	1.0000	0.
	-4.01722E-02	.33209	-.10165	-.43561	2.01609E-03	-5.79304E-03	1.0000	0.
	-.80592	5.4126	7.08233E-04	-6.44747E-03	-.14506	1.68400E-02	1.0000	0.
VELOCITY=	10.000	CIRCLE						
	-.68131	0.	-.86398	0.	-1.20237E-02	0.	1.0000	0.
	-2.4451	0.	-9.39777E-03	0.	3.25911E-03	0.	1.0000	0.
	-4.70817E-02	.34077	-.11686	-.47394	2.24432E-03	-6.56955E-03	1.0000	0.
	-.89981	5.5351	5.94197E-04	-7.20968E-03	-.14539	1.80656E-02	1.0000	0.
VELOCITY=	11.000							
	-.73537	0.	-.79835	0.	-1.10947E-02	0.	1.0000	0.
	-2.6952	0.	-9.37767E-03	0.	3.66681E-03	0.	1.0000	0.
	-5.33907E-02	.34915	-.13130	-.51050	2.46761E-03	-7.38407E-03	1.0000	0.
	-.99408	5.6660	4.48060E-04	-7.94601E-03	-.14524	1.91106E-02	1.0000	0.
VELOCITY=	12.000							
	-.78971	0.	-.75577	0.	-1.05155E-02	0.	1.0000	0.
	-2.9455	0.	-9.36328E-03	0.	4.06716E-03	0.	1.0000	0.
	-5.91048E-02	.35722	-.14495	-.54566	2.68142E-03	-8.23508E-03	1.0000	0.
	-1.0887	5.8050	2.74343E-04	-8.64959E-03	-.14475	1.99909E-02	1.0000	0.

VELOCITY= 13.000

-.84432	0.	-.72621	0.	-1.01339E-02	0.	1.0000	0.
-3.1960	0.	-9.35287E-03	0.	4.45582E-03	0.	1.0000	0.
-6.42581E-02	.36493	-.15785	-.57974	2.88193E-03	-9.12001E-03	1.0000	0.
-1.1836	5.9516	7.81978E-05	-9.31564E-03	-.14401	2.07254E-02	1.0000	0.

VELOCITY= 14.000

-.89915	0.	-.70467	0.	-9.87266E-03	0.	1.0000	0.
-3.4467	0.	-9.34526E-03	0.	4.82959E-03	0.	1.0000	0.
-6.88991E-02	.37225	-.17007	-.61300	3.06604E-03	-1.00357E-02	1.0000	0.
-1.2788	6.1054	-1.35098E-04	-9.94129E-03	-.14311	2.13334E-02	1.0000	0.

VELOCITY= 15.000

-.95419	0.	-.68838	0.	-9.68897E-03	0.	1.0000	0.
-3.6976	0.	-9.33967E-03	0.	5.18632E-03	0.	1.0000	0.
-7.30816E-02	.37915	-.18171	-.64566	3.23135E-03	-1.09789E-02	1.0000	0.
-1.3742	6.2660	-3.60505E-04	-1.05253E-02	-.14212	2.18337E-02	1.0000	0.

VELOCITY= 16.000

-1.0094	0.	-.67569	0.	-9.55706E-03	0.	1.0000	0.
-3.9486	0.	-9.33554E-03	0.	5.52470E-03	0.	1.0000	0.
-7.68600E-02	.38563	-.19286	-.67791	3.37610E-03	-1.19459E-02	1.0000	0.
-1.4697	6.4331	-5.93457E-04	-1.10676E-02	-.14108	2.22431E-02	1.0000	0.

VELOCITY= 17.000

-1.0648	0.	-.66558	0.	-9.46072E-03	0.	1.0000	0.
-4.1998	0.	-9.33249E-03	0.	5.84409E-03	0.	1.0000	0.
-8.02863E-02	.39168	-.20363	-.70987	3.49906E-03	-1.29337E-02	1.0000	0.
-1.5655	6.6061	-8.29986E-04	-1.15692E-02	-.14002	2.25768E-02	1.0000	0.

VELOCITY= 18.000

-1.1203	0.	-.65737	0.	-9.38938E-03	0.	1.0000	0.
-4.4510	0.	-9.33023E-03	0.	6.14440E-03	0.	1.0000	0.
-8.34085E-02	.39730	-.21410	-.74164	3.59950E-03	-1.39390E-02	1.0000	0.
-1.6614	6.7846	-1.06676E-03	-1.20318E-02	-.13898	2.28476E-02	1.0000	0.

VELOCITY= 19.000

-1.1760	0.	-.65060	0.	-9.33590E-03	0.	1.0000	0.
-4.7024	0.	-9.32857E-03	0.	6.42590E-03	0.	1.0000	0.
-8.62702E-02	.40252	-.22435	-.77331	3.67705E-03	-1.49591E-02	1.0000	0.
-1.7574	6.9683	-1.30109E-03	-1.24575E-02	-.13797	2.30666E-02	1.0000	0.

VELOCITY= 20.000 SQUARE

-1.2318	0.	-.64494	0.	-9.29538E-03	0.	1.0000	0.
-4.9539	0.	-9.32735E-03	0.	6.68917E-03	0.	1.0000	0.
-8.89099E-02	.40735	-.23445	-.80491	3.73161E-03	-1.59913E-02	1.0000	0.
-1.8535	7.1568	-1.53085E-03	-1.28485E-02	-.13700	2.32431E-02	1.0000	0.

VELOCITY= 21.000

-1.2877	0.	-.64014	0.	-9.26436E-03	0.	1.0000	0.
-5.2054	0.	-9.32647E-03	0.	6.93495E-03	0.	1.0000	0.
-9.13618E-02	.41181	-.24446	-.83650	3.76337E-03	-1.70336E-02	1.0000	0.
-1.9497	7.3497	-1.75442E-03	-1.32074E-02	-.13607	2.33846E-02	1.0000	0.

VELOCITY= 22.000

-1.3438	0.	-.63605	0.	-9.24042E-03	0.	1.0000	0.
-5.4569	0.	-9.32584E-03	0.	7.16412E-03	0.	1.0000	0.
-9.36556E-02	.41592	-.25443	-.86810	3.77263E-03	-1.80837E-02	1.0000	0.
-2.0460	7.5467	-1.97065E-03	-1.35367E-02	-.13519	2.34976E-02	1.0000	0.

VELOCITY= 23.000

-1.3999	0.	-.63251	0.	-9.22176E-03	0.	1.0000	0.
-5.7085	0.	-9.32541E-03	0.	7.37762E-03	0.	1.0000	0.
-9.58172E-02	.41970	-.26441	-.89972	3.75986E-03	-1.91401E-02	1.0000	0.
-2.1423	7.7475	-2.17872E-03	-1.38388E-02	-.13437	2.35873E-02	1.0000	0.

VELOCITY= 24.000

-1.4561	0.	-.62945	0.	-9.20712E-03	0.	1.0000	0.
-5.9601	0.	-9.32512E-03	0.	7.57644E-03	0.	1.0000	0.
-9.78687E-02	.42318	-.27444	-.93138	3.72559E-03	-2.02012E-02	1.0000	0.
-2.2386	7.9519	-2.37813E-03	-1.41159E-02	-.13359	2.36580E-02	1.0000	0.

VELOCITY= 25.000

-1.5124	0.	-.62676	0.	-9.19553E-03	0.	1.0000	0.
-6.2117	0.	-9.32495E-03	0.	7.76154E-03	0.	1.0000	0.
-9.98293E-02	.42638	-.28454	-.96308	3.67040E-03	-2.12656E-02	1.0000	0.
-2.3350	8.1595	-2.56862E-03	-1.43703E-02	-.13287	2.37132E-02	1.0000	0.

VELOCITY=	26.000							
	-1.5689	0.	-.62440	0.	-9.18629E-03	0.	1.0000	0.
	-6.4633	0.	-9.32486E-03	0.	7.93387E-03	0.	1.0000	0.
	-.10172	.42931	-.29476	-.99483	3.59491E-03	-2.23324E-02	1.0000	0.
	-2.4314	8.3701	-2.75009E-03	-1.46041E-02	-.13219	2.37557E-02	1.0000	0.
VELOCITY=	27.000							
	-1.6253	0.	-.62231	0.	-9.17887E-03	0.	1.0000	0.
	-6.7148	0.	-9.32483E-03	0.	8.09435E-03	0.	1.0000	0.
	-.10354	.43201	-.30510	-1.0266	3.49975E-03	-2.34004E-02	1.0000	0.
	-2.5278	8.5835	-2.92262E-03	-1.48189E-02	-.13155	2.37880E-02	1.0000	0.
VELOCITY=	28.000							
	-1.6819	0.	-.62045	0.	-9.17288E-03	0.	1.0000	0.
	-6.9664	0.	-9.32486E-03	0.	8.24385E-03	0.	1.0000	0.
	-.10532	.43448	-.31560	-1.0584	3.38553E-03	-2.44689E-02	1.0000	0.
	-2.6242	8.7995	-3.08637E-03	-1.50167E-02	-.13096	2.38119E-02	1.0000	0.
VELOCITY=	29.000							
	-1.7386	0.	-.61879	0.	-9.16800E-03	0.	1.0000	0.
	-7.2180	0.	-9.32492E-03	0.	8.38318E-03	0.	1.0000	0.
	-.10705	.43675	-.32627	-1.0903	3.25287E-03	-2.55372E-02	1.0000	0.
	-2.7206	9.0180	-3.24159E-03	-1.51989E-02	-.13041	2.38290E-02	1.0000	0.
VELOCITY=	30.000	DIAMOND						
	-1.7953	0.	-.61730	0.	-9.16400E-03	0.	1.0000	0.
	-7.4695	0.	-9.32500E-03	0.	8.51311E-03	0.	1.0000	0.
	-.10876	.43882	-.33713	-1.1222	3.10235E-03	-2.66047E-02	1.0000	0.
	-2.8170	9.2387	-3.38858E-03	-1.53670E-02	-.12989	2.38406E-02	1.0000	0.
VELOCITY=	31.000							
	-1.8520	0.	-.61595	0.	-9.16070E-03	0.	1.0000	0.
	-7.7211	0.	-9.32511E-03	0.	8.63437E-03	0.	1.0000	0.
	-.11044	.44072	-.34820	-1.1541	2.93453E-03	-2.76708E-02	1.0000	0.
	-2.9133	9.4614	-3.52768E-03	-1.55223E-02	-.12941	2.38478E-02	1.0000	0.
VELOCITY=	32.000							
	-1.9089	0.	-.61474	0.	-9.15797E-03	0.	1.0000	0.
	-7.9726	0.	-9.32523E-03	0.	8.74761E-03	0.	1.0000	0.
	-.11211	.44246	-.35947	-1.1861	2.74995E-03	-2.87352E-02	1.0000	0.
	-3.0097	9.6861	-3.65925E-03	-1.56658E-02	-.12896	2.38513E-02	1.0000	0.
VELOCITY=	33.000							
	-1.9658	0.	-.61363	0.	-9.15568E-03	0.	1.0000	0.
	-8.2241	0.	-9.32536E-03	0.	8.85344E-03	0.	1.0000	0.
	-.11377	.44405	-.37098	-1.2181	2.54909E-03	-2.97973E-02	1.0000	0.
	-3.1061	9.9127	-3.78366E-03	-1.57988E-02	-.12854	2.38520E-02	1.0000	0.
VELOCITY=	34.000							
	-2.0227	0.	-.61263	0.	-9.15377E-03	0.	1.0000	0.
	-8.4755	0.	-9.32550E-03	0.	8.95244E-03	0.	1.0000	0.
	-.11541	.44549	-.38272	-1.2500	2.33245E-03	-3.08570E-02	1.0000	0.
	-3.2024	10.141	-3.90128E-03	-1.59220E-02	-.12815	2.38504E-02	1.0000	0.
VELOCITY=	35.000							
	-2.0797	0.	-.61171	0.	-9.15215E-03	0.	1.0000	0.
	-8.7270	0.	-9.32564E-03	0.	9.04512E-03	0.	1.0000	0.
	-.11705	.44681	-.39470	-1.2820	2.10045E-03	-3.19139E-02	1.0000	0.
	-3.2988	10.371	-4.01249E-03	-1.60365E-02	-.12778	2.38469E-02	1.0000	0.
VELOCITY=	36.000							
	-2.1368	0.	-.61087	0.	-9.15077E-03	0.	1.0000	0.
	-8.9764	0.	-9.32578E-03	0.	9.13197E-03	0.	1.0000	0.
	-.11870	.44801	-.40694	-1.3140	1.85351E-03	-3.29678E-02	1.0000	0.
	-3.3951	10.602	-4.11764E-03	-1.61428E-02	-.12744	2.38420E-02	1.0000	0.
VELOCITY=	37.000							
	-2.1939	0.	-.61010	0.	-9.14960E-03	0.	1.0000	0.
	-9.2298	0.	-9.32592E-03	0.	9.21342E-03	0.	1.0000	0.
	-.12034	.44910	-.41943	-1.3459	1.59201E-03	-3.40185E-02	1.0000	0.
	-3.4914	10.835	-4.21709E-03	-1.62419E-02	-.12712	2.38359E-02	1.0000	0.
VELOCITY=	38.000							
	-2.2510	0.	-.60940	0.	-9.14859E-03	0.	1.0000	0.
	-9.4811	0.	-9.32606E-03	0.	9.28987E-03	0.	1.0000	0.
	-.12198	.45009	-.43219	-1.3779	1.31631E-03	-3.50659E-02	1.0000	0.
	-3.5877	11.068	-4.31116E-03	-1.63341E-02	-.12682	2.38290E-02	1.0000	0.

VELOCITY=	39.000							
	-2.3082	0.	-.60874	0.	-9.14772E-03	0.	1.0000	0.
	-9.7325	0.	-9.32620E-03	0.	9.36170E-03	0.	1.0000	0.
	-.12363	.45098	-.44521	-1.4098	1.02674E-03	-3.61097E-02	1.0000	0.
	-3.6839	11.304	-4.40018E-03	-1.64203E-02	-.12653	2.38214E-02	1.0000	0.
VELOCITY=	40.000	TRIANGLE						
	-2.3654	0.	-.60814	0.	-9.14697E-03	0.	1.0000	0.
	-9.9838	0.	-9.32633E-03	0.	9.42925E-03	0.	1.0000	0.
	-.12529	.45178	-.45851	-1.4417	7.23609E-04	-3.71498E-02	1.0000	0.
	-3.7802	11.540	-4.48444E-03	-1.65007E-02	-.12627	2.38133E-02	1.0000	0.
VELOCITY=	41.000							
	-2.4227	0.	-.60758	0.	-9.14631E-03	0.	1.0000	0.
	-10.235	0.	-9.32646E-03	0.	9.49283E-03	0.	1.0000	0.
	-.12695	.45250	-.47208	-1.4736	4.07192E-04	-3.81861E-02	1.0000	0.
	-3.8764	11.777	-4.56425E-03	-1.65760E-02	-.12602	2.38048E-02	1.0000	0.
VELOCITY=	42.000							
	-2.4799	0.	-.60706	0.	-9.14573E-03	0.	1.0000	0.
	-10.486	0.	-9.32659E-03	0.	9.55273E-03	0.	1.0000	0.
	-.12862	.45314	-.48594	-1.5054	7.77566E-05	-3.92186E-02	1.0000	0.
	-3.9726	12.016	-4.63587E-03	-1.66455E-02	-.12578	2.37961E-02	1.0000	0.
VELOCITY=	43.000							
	-2.5373	0.	-.60658	0.	-9.14523E-03	0.	1.0000	0.
	-10.738	0.	-9.32671E-03	0.	9.60921E-03	0.	1.0000	0.
	-.13030	.45371	-.50008	-1.5372	-2.64455E-04	-4.02471E-02	1.0000	0.
	-4.0688	12.255	-4.71155E-03	-1.67126E-02	-.12556	2.37873E-02	1.0000	0.
VELOCITY=	44.000							
	-2.5946	0.	-.60613	0.	-9.14478E-03	0.	1.0000	0.
	-10.989	0.	-9.32683E-03	0.	9.66251E-03	0.	1.0000	0.
	-.13199	.45421	-.51451	-1.5683	-6.19217E-04	-4.12715E-02	1.0000	0.
	-4.1649	12.495	-4.77954E-03	-1.67747E-02	-.12535	2.37784E-02	1.0000	0.
VELOCITY=	45.000							
	-2.6520	0.	-.60571	0.	-9.14438E-03	0.	1.0000	0.
	-11.240	0.	-9.32695E-03	0.	9.71285E-03	0.	1.0000	0.
	-.13369	.45464	-.52922	-1.6006	-9.86322E-04	-4.22918E-02	1.0000	0.
	-4.2611	12.736	-4.84406E-03	-1.68330E-02	-.12515	2.37696E-02	1.0000	0.
VELOCITY=	46.000							
	-2.7094	0.	-.60532	0.	-9.14403E-03	0.	1.0000	0.
	-11.491	0.	-9.32706E-03	0.	9.76044E-03	0.	1.0000	0.
	-.13540	.45502	-.54423	-1.6322	-1.36558E-03	-4.33079E-02	1.0000	0.
	-4.3572	12.978	-4.90532E-03	-1.68878E-02	-.12496	2.37608E-02	1.0000	0.
VELOCITY=	47.000							
	-2.7668	0.	-.60495	0.	-9.14371E-03	0.	1.0000	0.
	-11.743	0.	-9.32717E-03	0.	9.80547E-03	0.	1.0000	0.
	-.13712	.45533	-.55954	-1.6638	-1.75681E-03	-4.43197E-02	1.0000	0.
	-4.4533	13.221	-4.96352E-03	-1.69395E-02	-.12479	2.37520E-02	1.0000	0.
VELOCITY=	48.000							
	-2.8243	0.	-.60461	0.	-9.14342E-03	0.	1.0000	0.
	-11.994	0.	-9.32727E-03	0.	9.84810E-03	0.	1.0000	0.
	-.13884	.45559	-.57514	-1.6953	-2.15984E-03	-4.53272E-02	1.0000	0.
	-4.5494	13.464	-5.01884E-03	-1.69883E-02	-.12462	2.37434E-02	1.0000	0.
VELOCITY=	49.000							
	-2.8818	0.	-.60429	0.	-9.14316E-03	0.	1.0000	0.
	-12.245	0.	-9.32737E-03	0.	9.88850E-03	0.	1.0000	0.
	-.14058	.45580	-.59104	-1.7267	-2.57453E-03	-4.63303E-02	1.0000	0.
	-4.6455	13.708	-5.07145E-03	-1.70342E-02	-.12446	2.37350E-02	1.0000	0.
VELOCITY=	50.000	RT TRNGL						
	-2.9393	0.	-.60398	0.	-9.14293E-03	0.	1.0000	0.
	-12.496	0.	-9.32747E-03	0.	9.92682E-03	0.	1.0000	0.
	-.14233	.45597	-.60723	-1.7581	-3.00073E-03	-4.73290E-02	1.0000	0.
	-4.7416	13.952	-5.12152E-03	-1.70777E-02	-.12431	2.37267E-02	1.0000	0.
VELOCITY=	51.000							
	-2.9968	0.	-.60370	0.	-9.14272E-03	0.	1.0000	0.
	-12.747	0.	-9.32756E-03	0.	9.96318E-03	0.	1.0000	0.
	-.14409	.45608	-.62373	-1.7893	-3.43831E-03	-4.83231E-02	1.0000	0.
	-4.8377	14.198	-5.16920E-03	-1.71187E-02	-.12417	2.37186E-02	1.0000	0.

Listing of tape 30 for sample case (trim angle of attack and aerodynamic coefficients).-

LATERAL STABILITY OF TETHERED BALLOON - LRC BALLOON - REFERENCE CONFIGURATION 11/17/72 00.33.53.

AERODYNAMIC COEFFICIENTS

VELOCITY	ALPHAD	CD	CL	CLB	CLBD	CLP	CLR	CNB	CNBD	CNP	CNR	CYB	CYBD	CYP	CYR
1.000	8.767	5.9599E-02	9.9706E-02	-1.5439E-02	6.9808E-04	-2.4048E-02	-2.8041E-02	-2.5033E-02	3.8858E-02	-2.9795E-02	-.2807	-.8200	-8.9000E-02	8.8154E-02	.9219
2.000	8.550	5.9453E-02	9.6972E-02	-1.5351E-02	6.4945E-04	-2.4012E-02	-2.7026E-02	-2.5010E-02	3.8861E-02	-2.9096E-02	-.2808	-.8200	-8.9000E-02	8.5411E-02	.9220
3.000	8.250	5.9284E-02	9.3205E-02	-1.5228E-02	5.8215E-04	-2.3964E-02	-2.5621E-02	-2.4980E-02	3.8864E-02	-2.8125E-02	-.2808	-.8200	-8.9000E-02	8.1614E-02	.9220
4.000	7.923	5.9137E-02	8.9106E-02	-1.5094E-02	5.0871E-04	-2.3917E-02	-2.4097E-02	-2.4952E-02	3.8867E-02	-2.7062E-02	-.2808	-.8200	-8.9000E-02	7.7469E-02	.9221
5.000	7.609	5.9026E-02	8.5161E-02	-1.4964E-02	4.3807E-04	-2.3875E-02	-2.2612E-02	-2.4928E-02	3.8869E-02	-2.6037E-02	-.2808	-.8200	-8.9000E-02	7.3480E-02	.9221
6.000	7.327	5.8948E-02	8.1624E-02	-1.4848E-02	3.7488E-04	-2.3840E-02	-2.1292E-02	-2.4909E-02	3.8871E-02	-2.5119E-02	-.2809	-.8200	-8.9000E-02	6.9911E-02	.9222
7.000	7.086	5.8894E-02	7.8580E-02	-1.4748E-02	3.2067E-04	-2.3813E-02	-2.0160E-02	-2.4896E-02	3.8873E-02	-2.4329E-02	-.2809	-.8200	-8.9000E-02	6.6848E-02	.9222
8.000	6.884	5.8856E-02	7.6016E-02	-1.4663E-02	2.7522E-04	-2.3792E-02	-1.9210E-02	-2.4886E-02	3.8874E-02	-2.3665E-02	-.2809	-.8200	-8.9000E-02	6.4279E-02	.9222
9.000	6.716	5.8830E-02	7.3882E-02	-1.4593E-02	2.3750E-04	-2.3776E-02	-1.8422E-02	-2.4879E-02	3.8875E-02	-2.3114E-02	-.2809	-.8200	-8.9000E-02	6.2148E-02	.9222
10.00	6.577	5.8811E-02	7.2110E-02	-1.4536E-02	2.0632E-04	-2.3764E-02	-1.7771E-02	-2.4874E-02	3.8875E-02	-2.2658E-02	-.2809	-.8200	-8.9000E-02	6.0385E-02	.9223
11.00	6.462	5.8798E-02	7.0640E-02	-1.4488E-02	1.8051E-04	-2.3754E-02	-1.7231E-02	-2.4871E-02	3.8876E-02	-2.2280E-02	-.2809	-.8200	-8.9000E-02	5.8925E-02	.9223
12.00	6.367	5.8787E-02	6.9415E-02	-1.4448E-02	1.5907E-04	-2.3746E-02	-1.6783E-02	-2.4868E-02	3.8876E-02	-2.1965E-02	-.2809	-.8200	-8.9000E-02	5.7713E-02	.9223
13.00	6.287	5.8779E-02	6.8389E-02	-1.4414E-02	1.4116E-04	-2.3740E-02	-1.6409E-02	-2.4866E-02	3.8876E-02	-2.1703E-02	-.2809	-.8200	-8.9000E-02	5.6700E-02	.9223
14.00	6.220	5.8773E-02	6.7526E-02	-1.4386E-02	1.2612E-04	-2.3735E-02	-1.6094E-02	-2.4865E-02	3.8876E-02	-2.1482E-02	-.2809	-.8200	-8.9000E-02	5.5849E-02	.9223

15.00	6.164	5.8768E-02	6.6795E-02	-1.4362E-02	1.1340E-04	-2.3731E-02	-1.5828E-02	-2.4864E-02	3.8876E-02	-2.1295E-02
		-.2809	-.8200	-8.9000E-02	5.5129E-02	.9223				
16.00	6.115	5.8764E-02	6.6172E-02	-1.4342E-02	1.0257E-04	-2.3728E-02	-1.5602E-02	-2.4863E-02	3.8876E-02	-2.1136E-02
		-.2809	-.8200	-8.9000E-02	5.4517E-02	.9223				
17.00	6.074	5.8761E-02	6.5638E-02	-1.4325E-02	9.3306E-05	-2.3725E-02	-1.5408E-02	-2.4862E-02	3.8877E-02	-2.1000E-02
		-.2809	-.8200	-8.9000E-02	5.3992E-02	.9223				
18.00	6.039	5.8758E-02	6.5177E-02	-1.4310E-02	8.5322E-05	-2.3722E-02	-1.5241E-02	-2.4862E-02	3.8877E-02	-2.0883E-02
		-.2809	-.8200	-8.9000E-02	5.3541E-02	.9223				
19.00	6.008	5.8756E-02	6.4777E-02	-1.4297E-02	7.8404E-05	-2.3720E-02	-1.5097E-02	-2.4861E-02	3.8877E-02	-2.0781E-02
		-.2809	-.8200	-8.9000E-02	5.3149E-02	.9223				
20.00	5.981	5.8754E-02	6.4429E-02	-1.4286E-02	7.2378E-05	-2.3719E-02	-1.4971E-02	-2.4861E-02	3.8877E-02	-2.0693E-02
		-.2809	-.8200	-8.9000E-02	5.2808E-02	.9223				
21.00	5.958	5.8753E-02	6.4124E-02	-1.4276E-02	6.7101E-05	-2.3717E-02	-1.4861E-02	-2.4861E-02	3.8877E-02	-2.0615E-02
		-.2809	-.8200	-8.9000E-02	5.2510E-02	.9223				
22.00	5.937	5.8751E-02	6.3855E-02	-1.4267E-02	6.2456E-05	-2.3716E-02	-1.4763E-02	-2.4860E-02	3.8877E-02	-2.0547E-02
		-.2809	-.8200	-8.9000E-02	5.2247E-02	.9223				
23.00	5.919	5.8750E-02	6.3617E-02	-1.4260E-02	5.8350E-05	-2.3715E-02	-1.4678E-02	-2.4860E-02	3.8877E-02	-2.0486E-02
		-.2809	-.8200	-8.9000E-02	5.2015E-02	.9223				
24.00	5.902	5.8749E-02	6.3406E-02	-1.4253E-02	5.4704E-05	-2.3714E-02	-1.4601E-02	-2.4860E-02	3.8877E-02	-2.0433E-02
		-.2809	-.8200	-8.9000E-02	5.1808E-02	.9223				
25.00	5.888	5.8748E-02	6.3217E-02	-1.4247E-02	5.1453E-05	-2.3713E-02	-1.4533E-02	-2.4860E-02	3.8877E-02	-2.0385E-02
		-.2809	-.8200	-8.9000E-02	5.1624E-02	.9223				
26.00	5.875	5.8747E-02	6.3049E-02	-1.4241E-02	4.8544E-05	-2.3712E-02	-1.4473E-02	-2.4860E-02	3.8877E-02	-2.0342E-02
		-.2809	-.8200	-8.9000E-02	5.1460E-02	.9223				
27.00	5.863	5.8746E-02	6.2897E-02	-1.4236E-02	4.5930E-05	-2.3711E-02	-1.4418E-02	-2.4860E-02	3.8877E-02	-2.0304E-02
		-.2809	-.8200	-8.9000E-02	5.1312E-02	.9223				
28.00	5.853	5.8746E-02	6.2760E-02	-1.4232E-02	4.3574E-05	-2.3711E-02	-1.4369E-02	-2.4860E-02	3.8877E-02	-2.0269E-02
		-.2809	-.8200	-8.9000E-02	5.1178E-02	.9223				
29.00	5.843	5.8745E-02	6.2636E-02	-1.4228E-02	4.1443E-05	-2.3710E-02	-1.4324E-02	-2.4860E-02	3.8877E-02	-2.0238E-02
		-.2809	-.8200	-8.9000E-02	5.1058E-02	.9223				
30.00	5.835	5.8745E-02	6.2524E-02	-1.4224E-02	3.9510E-05	-2.3710E-02	-1.4284E-02	-2.4860E-02	3.8877E-02	-2.0209E-02
		-.2809	-.8200	-8.9000E-02	5.0949E-02	.9223				
31.00	5.827	5.8744E-02	6.2422E-02	-1.4221E-02	3.7752E-05	-2.3709E-02	-1.4247E-02	-2.4859E-02	3.8877E-02	-2.0184E-02
		-.2809	-.8200	-8.9000E-02	5.0849E-02	.9223				
32.00	5.820	5.8744E-02	6.2329E-02	-1.4218E-02	3.6148E-05	-2.3709E-02	-1.4213E-02	-2.4859E-02	3.8877E-02	-2.0160E-02
		-.2809	-.8200	-8.9000E-02	5.0758E-02	.9223				
33.00	5.813	5.8743E-02	6.2244E-02	-1.4215E-02	3.4682E-05	-2.3708E-02	-1.4183E-02	-2.4859E-02	3.8877E-02	-2.0138E-02
		-.2809	-.8200	-8.9000E-02	5.0675E-02	.9223				

34.00	5.807	5.8743E-02	6.2166E-02	-1.4213E-02	3.3337E-05	-2.3708E-02	-1.4155E-02	-2.4859E-02	3.8877E-02	-2.0119E-02
		-.2809	-.8200	-8.9000E-02	5.0599E-02	.9223				
35.00	5.802	5.8743E-02	6.2094E-02	-1.4210E-02	3.2101E-05	-2.3708E-02	-1.4129E-02	-2.4859E-02	3.8877E-02	-2.0100E-02
		-.2809	-.8200	-8.9000E-02	5.0529E-02	.9223				
36.00	5.797	5.8742E-02	6.2028E-02	-1.4208E-02	3.0963E-05	-2.3707E-02	-1.4105E-02	-2.4859E-02	3.8877E-02	-2.0084E-02
		-.2809	-.8200	-8.9000E-02	5.0465E-02	.9223				
37.00	5.792	5.8742E-02	6.1966E-02	-1.4206E-02	2.9912E-05	-2.3707E-02	-1.4083E-02	-2.4859E-02	3.8877E-02	-2.0068E-02
		-.2809	-.8200	-8.9000E-02	5.0405E-02	.9223				
38.00	5.788	5.8742E-02	6.1910E-02	-1.4205E-02	2.8941E-05	-2.3707E-02	-1.4063E-02	-2.4859E-02	3.8877E-02	-2.0054E-02
		-.2809	-.8200	-8.9000E-02	5.0350E-02	.9223				
39.00	5.784	5.8742E-02	6.1858E-02	-1.4203E-02	2.8041E-05	-2.3707E-02	-1.4044E-02	-2.4859E-02	3.8877E-02	-2.0041E-02
		-.2809	-.8200	-8.9000E-02	5.0299E-02	.9223				
40.00	5.780	5.8742E-02	6.1809E-02	-1.4201E-02	2.7205E-05	-2.3707E-02	-1.4026E-02	-2.4859E-02	3.8877E-02	-2.0028E-02
		-.2809	-.8200	-8.9000E-02	5.0252E-02	.9223				
41.00	5.777	5.8741E-02	6.1764E-02	-1.4200E-02	2.6428E-05	-2.3706E-02	-1.4010E-02	-2.4859E-02	3.8877E-02	-2.0017E-02
		-.2809	-.8200	-8.9000E-02	5.0208E-02	.9223				
42.00	5.773	5.8741E-02	6.1722E-02	-1.4198E-02	2.5704E-05	-2.3706E-02	-1.3995E-02	-2.4859E-02	3.8877E-02	-2.0006E-02
		-.2809	-.8200	-8.9000E-02	5.0167E-02	.9223				
43.00	5.770	5.8741E-02	6.1682E-02	-1.4197E-02	2.5029E-05	-2.3706E-02	-1.3981E-02	-2.4859E-02	3.8877E-02	-1.9996E-02
		-.2809	-.8200	-8.9000E-02	5.0129E-02	.9223				
44.00	5.768	5.8741E-02	6.1646E-02	-1.4196E-02	2.4398E-05	-2.3706E-02	-1.3968E-02	-2.4859E-02	3.8877E-02	-1.9987E-02
		-.2809	-.8200	-8.9000E-02	5.0093E-02	.9223				
45.00	5.765	5.8741E-02	6.1611E-02	-1.4195E-02	2.3808E-05	-2.3706E-02	-1.3955E-02	-2.4859E-02	3.8877E-02	-1.9978E-02
		-.2809	-.8200	-8.9000E-02	5.0060E-02	.9223				
46.00	5.762	5.8741E-02	6.1579E-02	-1.4194E-02	2.3255E-05	-2.3706E-02	-1.3944E-02	-2.4859E-02	3.8877E-02	-1.9970E-02
		-.2809	-.8200	-8.9000E-02	5.0029E-02	.9223				
47.00	5.760	5.8740E-02	6.1549E-02	-1.4193E-02	2.2736E-05	-2.3705E-02	-1.3933E-02	-2.4859E-02	3.8877E-02	-1.9963E-02
		-.2809	-.8200	-8.9000E-02	4.9999E-02	.9223				
48.00	5.758	5.8740E-02	6.1521E-02	-1.4192E-02	2.2248E-05	-2.3705E-02	-1.3923E-02	-2.4859E-02	3.8877E-02	-1.9955E-02
		-.2809	-.8200	-8.9000E-02	4.9972E-02	.9223				
49.00	5.756	5.8740E-02	6.1494E-02	-1.4191E-02	2.1790E-05	-2.3705E-02	-1.3913E-02	-2.4859E-02	3.8877E-02	-1.9949E-02
		-.2809	-.8200	-8.9000E-02	4.9946E-02	.9223				
50.00	5.754	5.8740E-02	6.1469E-02	-1.4190E-02	2.1358E-05	-2.3705E-02	-1.3904E-02	-2.4859E-02	3.8877E-02	-1.9942E-02
		-.2809	-.8200	-8.9000E-02	4.9921E-02	.9223				
51.00	5.752	5.8740E-02	6.1445E-02	-1.4190E-02	2.0950E-05	-2.3705E-02	-1.3896E-02	-2.4859E-02	3.8877E-02	-1.9936E-02
		-.2809	-.8200	-8.9000E-02	4.9898E-02	.9223				

Listing of tape 31 for sample case (tether spring constants).-

LATERAL STABILITY OF TETHERED BALLCCN - LRC BALLOON - REFERENCE CCNFIGURATICN 11/17/72 00.33.53.

TETHER SPRINGS

VELOCITY	SKYY	SKYP	SKYS	SKPP	SKPS	SKSS
1.000	1.171	-3.879	3.372	12.84	-11.16	9.705
2.000	1.193	-3.962	3.418	13.16	-11.35	9.795
3.000	1.227	-4.096	3.496	13.67	-11.67	9.958
4.000	1.276	-4.279	3.610	14.35	-12.10	10.21
5.000	1.339	-4.512	3.765	15.20	-12.68	10.58
6.000	1.419	-4.799	3.965	16.23	-13.41	11.08
7.000	1.516	-5.146	4.215	17.47	-14.31	11.72
8.000	1.632	-5.556	4.518	18.91	-15.38	12.51
9.000	1.768	-6.032	4.876	20.58	-16.64	13.45
10.00	1.923	-6.574	5.287	22.47	-18.08	14.54
11.00	2.097	-7.182	5.753	24.59	-19.70	15.78
12.00	2.291	-7.855	6.271	26.93	-21.50	17.16
13.00	2.503	-8.592	6.840	29.49	-23.48	18.69
14.00	2.734	-9.393	7.460	32.27	-25.63	20.35
15.00	2.983	-10.26	8.129	35.27	-27.95	22.15
16.00	3.250	-11.18	8.847	38.47	-30.44	24.08
17.00	3.535	-12.17	9.613	41.89	-33.09	26.14
18.00	3.837	-13.22	10.43	45.52	-35.91	28.33
19.00	4.156	-14.32	11.29	49.35	-38.89	30.65
20.00	4.493	-15.49	12.19	53.39	-42.04	33.10
21.00	4.848	-16.71	13.15	57.63	-45.34	35.67
22.00	5.219	-18.00	14.15	62.09	-48.80	38.37
23.00	5.608	-19.35	15.20	66.74	-52.43	41.19
24.00	6.013	-20.75	16.29	71.60	-56.22	44.14
25.00	6.436	-22.21	17.43	76.67	-60.16	47.21
26.00	6.876	-23.74	18.62	81.93	-64.26	50.41
27.00	7.333	-25.32	19.85	87.41	-68.53	53.73
28.00	7.807	-26.96	21.13	93.08	-72.95	57.17
29.00	8.298	-28.66	22.45	98.97	-77.54	60.75
30.00	8.806	-30.42	23.82	105.1	-82.28	64.44
31.00	9.331	-32.23	25.24	111.3	-87.18	68.26
32.00	9.874	-34.11	26.70	117.8	-92.24	72.21
33.00	10.43	-36.05	28.21	124.5	-97.46	76.27
34.00	11.01	-38.04	29.76	131.4	-102.8	80.47
35.00	11.60	-40.09	31.37	138.5	-108.4	84.79
36.00	12.21	-42.21	33.01	145.9	-114.1	89.23
37.00	12.84	-44.38	34.71	153.4	-119.9	93.80
38.00	13.49	-46.61	36.45	161.1	-126.0	98.49
39.00	14.15	-48.90	38.23	169.0	-132.1	103.3
40.00	14.83	-51.25	40.06	177.2	-138.5	108.2
41.00	15.52	-53.66	41.94	185.5	-145.0	113.3
42.00	16.24	-56.13	43.87	194.0	-151.6	118.5
43.00	16.97	-58.66	45.84	202.8	-158.4	123.8
44.00	17.72	-61.24	47.85	211.7	-165.4	129.3
45.00	18.48	-63.89	49.91	220.9	-172.6	134.8
46.00	19.26	-66.60	52.02	230.2	-179.9	140.5
47.00	20.06	-69.36	54.18	239.8	-187.3	146.3
48.00	20.88	-72.18	56.38	249.6	-194.9	152.3
49.00	21.71	-75.07	58.63	259.6	-202.7	158.3
50.00	22.56	-78.01	60.92	269.7	-210.7	164.5
51.00	23.43	-81.01	63.26	280.1	-218.7	170.8

Listing of tape 32 for sample case (equilibrium tether conditions).-

LATERAL STABILITY OF TETHERED BALLOON - LRC BALLOON - REFERENCE CONFIGURATION 11/17/72 00.33.53.

VELOCITY	TETHER CONDITIONS X1	Z1	GAMO	TO	GAM1	T1	CAB DRAG
1.000	.4953	61.00	89.19	61.51	89.82	82.43	1.0104E-02
2.000	1.943	60.96	96.82	62.77	89.30	83.68	4.0418E-02
3.000	4.236	60.83	83.08	64.78	88.46	85.65	9.0940E-02
4.000	7.218	60.51	78.28	67.49	87.35	88.24	.1617
5.000	10.69	59.92	72.78	70.85	86.01	91.40	.2526
6.000	14.44	59.02	66.98	74.37	84.48	95.11	.3638
7.000	18.26	57.83	61.25	79.55	82.81	99.38	.4951
8.000	21.95	56.38	55.88	84.91	81.04	104.3	.6467
9.000	25.39	54.78	51.04	90.96	79.21	109.7	.8185
10.00	28.52	53.10	46.81	97.69	77.36	115.9	1.010
11.00	31.29	51.41	43.19	105.1	75.53	122.8	1.223
12.00	33.73	49.77	40.12	113.2	73.73	130.3	1.455
13.00	35.84	48.21	37.53	122.1	72.00	138.6	1.708
14.00	37.67	46.77	35.36	131.6	70.34	147.7	1.980
15.00	39.25	45.44	33.54	141.9	68.77	157.5	2.273
16.00	40.61	44.22	32.00	152.9	67.30	168.1	2.587
17.00	41.79	43.12	30.70	164.6	65.91	179.4	2.920
18.00	42.81	42.13	29.60	177.1	64.62	191.5	3.274
19.00	43.70	41.23	28.66	190.3	63.42	204.4	3.648
20.00	44.48	40.42	27.85	204.2	62.31	218.1	4.042
21.00	45.16	39.69	27.16	218.9	61.28	232.5	4.456
22.00	45.75	39.03	26.55	234.4	60.33	247.7	4.891
23.00	46.28	38.43	26.03	250.5	59.46	263.7	5.345
24.00	46.75	37.89	25.57	267.4	58.65	280.4	5.820
25.00	47.17	37.40	25.16	285.1	57.90	297.9	6.315
26.00	47.54	36.95	24.80	303.5	57.21	316.2	6.831
27.00	47.88	36.55	24.49	322.6	56.57	335.2	7.366
28.00	48.18	36.18	24.20	342.5	55.98	354.9	7.922
29.00	48.45	35.84	23.95	363.1	55.43	375.4	8.498
30.00	48.69	35.53	23.72	384.5	54.92	396.7	9.094
31.00	48.91	35.24	23.52	406.6	54.45	418.7	9.710
32.00	49.12	34.98	23.33	429.5	54.02	441.5	10.35
33.00	49.30	34.74	23.17	453.0	53.61	464.9	11.00
34.00	49.47	34.52	23.01	477.3	53.23	489.2	11.68
35.00	49.63	34.31	22.87	502.4	52.88	514.2	12.38
36.00	49.77	34.12	22.75	528.2	52.55	539.9	13.10
37.00	49.90	33.95	22.63	554.7	52.25	566.4	13.83
38.00	50.02	33.78	22.52	582.0	51.96	593.6	14.59
39.00	50.13	33.63	22.42	610.0	51.69	621.5	15.37
40.00	50.24	33.49	22.33	638.7	51.44	650.2	16.17
41.00	50.33	33.35	22.25	668.2	51.21	679.6	16.99
42.00	50.42	33.23	22.17	698.4	50.99	709.8	17.82
43.00	50.51	33.11	22.10	729.3	50.78	740.7	18.68
44.00	50.58	33.01	22.03	761.0	50.58	772.3	19.56
45.00	50.66	32.90	21.97	793.4	50.40	804.6	20.46
46.00	50.73	32.81	21.91	826.5	50.23	837.7	21.38
47.00	50.79	32.72	21.85	860.3	50.06	871.6	22.32
48.00	50.85	32.63	21.80	894.9	49.91	906.1	23.28
49.00	50.91	32.55	21.75	930.3	49.76	941.4	24.26
50.00	50.96	32.48	21.71	966.3	49.62	977.5	25.26
51.00	51.01	32.40	21.67	1003	49.49	1014	26.28

Listing of tape 33 for sample case (uncoupled characteristic roots).-

LATERAL STABILITY OF TETHERED BALLOON - LRC BALLOON - REFERENCE CONFIGURATION 11/17/72 00.33.53.

UNCOUPLED ROOTS

VEL	RLY1 IMY1	RLY2 IMY2	RLP1 IMP1	RLP2 IMP2	RLS1 IMS1	RLS2 IMS2
1.000	-4.4683E-02	-4.4683E-02	-7.7458E-02	-7.7458E-02	-.1252	-.1252
	.1662	-.1662	4.368	-4.368	.2103	-.2103
2.000	-8.9366E-02	-8.9366E-02	-.1560	-.1560	-.1877	-.3127
	.1489	-.1489	4.411	-4.411	0.	0.
3.000	-.1340	-.1340	-.2363	-.2363	-8.5105E-02	-.6647
	.1143	-.1143	4.474	-4.474	0.	0.
4.000	-.1787	-.1787	-.3183	-.3183	-5.6692E-02	-.9416
	1.7479E-02	-1.7479E-02	4.550	-4.550	0.	0.
5.000	-9.6676E-02	-.3502	-.4018	-.4018	-4.0784E-02	-1.206
	0.	0.	4.635	-4.635	0.	0.
6.000	-7.8326E-02	-.4579	-.4864	-.4864	-3.0034E-02	-1.464
	0.	0.	4.725	-4.725	0.	0.
7.000	-6.8835E-02	-.5567	-.5716	-.5716	-2.2012E-02	-1.719
	0.	0.	4.819	-4.819	0.	0.
8.000	-6.3315E-02	-.6516	-.6572	-.6572	-1.5637E-02	-1.973
	0.	0.	4.918	-4.918	0.	0.
9.000	-6.0035E-02	-.7443	-.7430	-.7430	-1.0340E-02	-2.226
	0.	0.	5.021	-5.021	0.	0.
10.00	-5.8170E-02	-.8355	-.8288	-.8288	-5.7874E-03	-2.477
	0.	0.	5.129	-5.129	0.	0.
11.00	-5.7260E-02	-.9258	-.9148	-.9148	-1.7686E-03	-2.729
	0.	0.	5.243	-5.243	0.	0.
12.00	-5.7026E-02	-1.015	-1.001	-1.001	1.8564E-03	-2.979
	0.	0.	5.361	-5.361	0.	0.
13.00	-5.7285E-02	-1.104	-1.086	-1.086	5.1838E-03	-3.230
	0.	0.	5.485	-5.485	0.	0.
14.00	-5.7914E-02	-1.193	-1.172	-1.172	8.2816E-03	-3.480
	0.	0.	5.613	-5.613	0.	0.
15.00	-5.8827E-02	-1.282	-1.258	-1.258	1.1199E-02	-3.731
	0.	0.	5.747	-5.747	0.	0.
16.00	-5.9963E-02	-1.370	-1.344	-1.344	1.3972E-02	-3.981
	0.	0.	5.886	-5.886	0.	0.
17.00	-6.1276E-02	-1.458	-1.429	-1.429	1.6629E-02	-4.231
	0.	0.	6.029	-6.029	0.	0.
18.00	-6.2732E-02	-1.546	-1.515	-1.515	1.9189E-02	-4.481
	0.	0.	6.176	-6.176	0.	0.
19.00	-6.4305E-02	-1.634	-1.600	-1.600	2.1669E-02	-4.731
	0.	0.	6.328	-6.328	0.	0.
20.00	-6.5976E-02	-1.721	-1.686	-1.686	2.4083E-02	-4.980
	0.	0.	6.483	-6.483	0.	0.
21.00	-6.7729E-02	-1.809	-1.771	-1.771	2.6438E-02	-5.230
	0.	0.	6.642	-6.642	0.	0.
22.00	-6.9552E-02	-1.896	-1.857	-1.857	2.8745E-02	-5.480
	0.	0.	6.805	-6.805	0.	0.
23.00	-7.1435E-02	-1.984	-1.942	-1.942	3.1010E-02	-5.730
	0.	0.	6.971	-6.971	0.	0.
24.00	-7.3370E-02	-2.071	-2.027	-2.027	3.3237E-02	-5.979
	0.	0.	7.140	-7.140	0.	0.

25.00	-7.5349E-02	-2.159	-2.113	-2.113	3.5432E-02	-6.229
	0.	0.	7.312	-7.312	0.	0.
26.00	-7.7369E-02	-2.246	-2.198	-2.198	3.7597E-02	-6.479
	0.	0.	7.486	-7.486	0.	0.
27.00	-7.9423E-02	-2.333	-2.283	-2.283	3.9738E-02	-6.728
	0.	0.	7.663	-7.663	0.	0.
28.00	-8.1509E-02	-2.421	-2.368	-2.368	4.1855E-02	-6.978
	0.	0.	7.843	-7.843	0.	0.
29.00	-8.3623E-02	-2.508	-2.454	-2.454	4.3952E-02	-7.227
	0.	0.	8.024	-8.024	0.	0.
30.00	-8.5761E-02	-2.595	-2.539	-2.539	4.6031E-02	-7.477
	0.	0.	8.208	-8.208	0.	0.
31.00	-8.7922E-02	-2.682	-2.624	-2.624	4.8092E-02	-7.727
	0.	0.	8.393	-8.393	0.	0.
32.00	-9.0104E-02	-2.770	-2.709	-2.709	5.0139E-02	-7.976
	0.	0.	8.580	-8.580	0.	0.
33.00	-9.2304E-02	-2.857	-2.794	-2.794	5.2171E-02	-8.226
	0.	0.	8.769	-8.769	0.	0.
34.00	-9.4521E-02	-2.944	-2.880	-2.880	5.4191E-02	-8.475
	0.	0.	8.960	-8.960	0.	0.
35.00	-9.6754E-02	-3.031	-2.965	-2.965	5.6200E-02	-8.725
	0.	0.	9.152	-9.152	0.	0.
36.00	-9.9001E-02	-3.118	-3.050	-3.050	5.8197E-02	-8.974
	0.	0.	9.346	-9.346	0.	0.
37.00	-.1013	-3.205	-3.135	-3.135	6.0185E-02	-9.224
	0.	0.	9.540	-9.540	0.	0.
38.00	-.1035	-3.292	-3.220	-3.220	6.2164E-02	-9.473
	0.	0.	9.737	-9.737	0.	0.
39.00	-.1058	-3.379	-3.305	-3.305	6.4134E-02	-9.723
	0.	0.	9.934	-9.934	0.	0.
40.00	-.1081	-3.467	-3.390	-3.390	6.6097E-02	-9.972
	0.	0.	10.13	-10.13	0.	0.
41.00	-.1104	-3.554	-3.475	-3.475	6.8052E-02	-10.22
	0.	0.	10.33	-10.33	0.	0.
42.00	-.1127	-3.641	-3.560	-3.560	7.0000E-02	-10.47
	0.	0.	10.53	-10.53	0.	0.
43.00	-.1150	-3.728	-3.645	-3.645	7.1942E-02	-10.72
	0.	0.	10.73	-10.73	0.	0.
44.00	-.1174	-3.815	-3.730	-3.730	7.3879E-02	-10.97
	0.	0.	10.94	-10.94	0.	0.
45.00	-.1197	-3.902	-3.815	-3.815	7.5810E-02	-11.22
	0.	0.	11.14	-11.14	0.	0.
46.00	-.1221	-3.989	-3.901	-3.901	7.7735E-02	-11.47
	0.	0.	11.34	-11.34	0.	0.
47.00	-.1244	-4.076	-3.986	-3.986	7.9656E-02	-11.72
	0.	0.	11.55	-11.55	0.	0.
48.00	-.1268	-4.163	-4.071	-4.071	8.1572E-02	-11.97
	0.	0.	11.75	-11.75	0.	0.
49.00	-.1291	-4.250	-4.156	-4.156	8.3484E-02	-12.22
	0.	0.	11.96	-11.96	0.	0.
50.00	-.1315	-4.337	-4.241	-4.241	8.5392E-02	-12.47
	0.	0.	12.17	-12.17	0.	0.
51.00	-.1339	-4.424	-4.326	-4.326	8.7296E-02	-12.72
	0.	0.	12.37	-12.37	0.	0.

PROGRAM FOR PLOTTING FREQUENCIES AND DECAY RATES VERSUS WIND VELOCITY

General Description

The dynamic characteristics of a tethered balloon may vary considerably with wind velocity. Plots of the frequencies $\mathrm{Im}(\lambda)$ and of the decay rates $\mathrm{Re}(\lambda)$ versus wind velocity are helpful in assessing trends. Program VPLOT is used to plot both the frequencies and the decay rates versus wind velocity for the longitudinal and lateral cases. A sample plot is given in figure 3.

The parameters for the scales of the plot are set with data statements within the program and must be changed internally if desired. The number of frequencies and number of decay rates off scale are counted and written on the plot (fig. 3). All of the data for the program are read from binary tape 7 written by program STABLTY. The file INPUT is thus deleted from the file assignments for the program because no data are read from it. The only printing from the program is the identification array and the number of velocity increments to be processed, both of which are read from tape 7. The principal plotting routines are described in the appendix. The version of the program given here requires 30-inch-wide paper for the plotter.

Definitions of Program Variables

Some of the principal FORTRAN variable names are given and defined in the following sections. The variables associated with scaling and axes are:

FORTRAN variable name	Definition
DAXL, FAXL, VAXL	length of plot axes for decay rate, frequency, and velocity, respectively, in.
DLAB, FLAB, VLAB	arrays of labeling information for decay rate, frequency, and velocity axes, respectively
DDEL, FDEL, VDEL	scale factors for decay rate, frequency, and velocity (change in units per in. of plot)
DMIN, FMIN, VMIN	minimum values of decay rate, frequency, and velocity to be plotted
DMAX, FMAX	maximum values of decay rate and frequency to be plotted

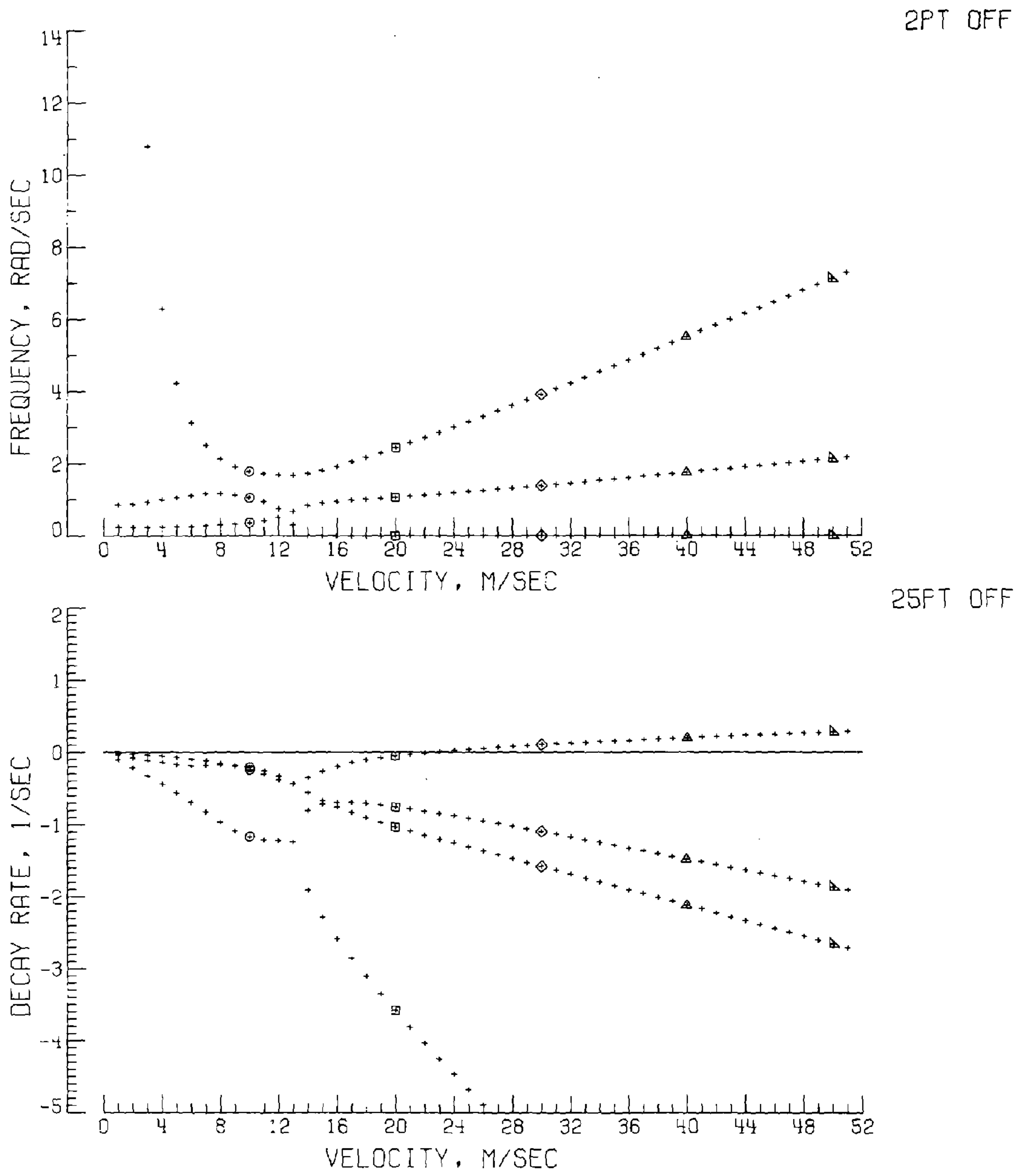


Figure 3.- Example of plot from program VPLOT.

FORTRAN variable name	Definition
DTICL, FTICL, VTICL	distance between large tick marks in inches on plot for the damping, frequency, and velocity axes, respectively
DZERO	distance in inches from zero decay rate level to plot origin as set by CALPLT
FFHH	vertical distance in inches from plot origin to end of frequency axis
FH	vertical distance in inches from plot origin to zero frequency level
HGT	height of lettering for labeling, in.
TICSND, TICSNF, TICSNV	number of small tick marks per inch on plot for the damping, frequency, and velocity axes, respectively

The variables read from tape 7 are:

FORTRAN variable name	Definition
ID	ten-word array containing case identification and date and time of processing of case by STABLTY
IK2	index for cycling symbols
NEGR	one plus the number of complex roots with negative frequencies
NROOT	number of roots for plotting for a given velocity
NTWO	order of linear eigenvalue problem
NVEL	number of velocity increments
ROOTI	array containing imaginary portion of eigenvalues (modal frequencies for a given velocity)

FORTRAN variable name	Definition
ROOTR	array containing real portion of eigenvalues (decay rates) for a given velocity, in same sequence as ROOTI
VEL	velocity (first in velocity units, then scaled to in. on plot)

Other variables are:

FORTRAN variable name	Definition
ISYM	index for plotting symbols
QLSYM	symbol quality parameter (+1. for high quality, -1. for low quality)
NDOFF, NFOFF	number of decay rates or frequencies off scale (encoded to Hollerith as DOFF and FOFF, respectively, for labeling plot)
RIN, RRN	real and imaginary parts of an eigenvalue (scaled to in. from plot origin as FREQ and DECAY, respectively)

```
      OVERLAY(VPLOT,0,0)
      PROGRAM VPLOT(OUTPUT=1,TAPE7)
C***********************************************************************
C*                                                                     *
C* PROGRAM A2864.3 - ROOTS VS VELOCITY PLOTTING PROGRAM                *
C* PLOTS REAL(ROOT) VS VELOCITY AND IMAG(ROOT) VS VELOCITY             *
C* SEE SUBROUTINE WRITEUPS FOR DESCRIPTION OF CALPLT,NOTATE,AXES,      *
C*   PNTPLT, OR DASHLN                                                 *
C*                                                                     *
C***********************************************************************
      DIMENSION ID(10),ROOTR(50),ROOTI(50),VLAB(5),DLAB(5),FLAB(5)
C
C  SCALING PARAMETERS AND LABELS FOR AXES
C
      DATA DLAB/50H                 DECAY RATE, 1/SEC                /
      DATA FLAB/50H                FREQUENCY, RAD/SEC                /
      DATA VLAB/50H                 VELOCITY, M/SEC                  /  SI UNITS
      DATA VAXL,VMIN,VDEL,VTICL,TICSNV/10.4,0.,5.,.8,5./
      DATA DAXL,DMIN,DDEL,DTICL,TICSND/7.,-5.,1.,1.,10./
      DATA FAXL,FMIN,FDEL,FTICL,TICSNF/7.,0.,2.,1.,2./,HGT/.250/
C
C  INITITALIZE PLOTTING ROUTINES, SET PLOT ORGIN, AND CALCULATE SCALING
C    PARAMETERS - CALL TO LEROY SLCWS PLOTTER FOR USING INK
C
      CALL CALCOMP  $  CALL CALPLT(3.,3.,-3)
      QLSYM=1.  $   IF(QLSYM.GT.0.) CALL LEROY
      RVDEL=1./VDEL $ RDDEL=1./DDEL $ RFDEL=1./FDEL
      FMAX=FMIN+FAXL*FDEL $ DMAX=DMIN+DAXL*DDEL $ DZERO=-DMIN*RDDEL
      FH=(DMAX-DMIN)*RDDEL+1.
C
C  READ ID ARRAY AND NUMBER OF VELOCITY POINTS FROM BINARY TAPE7
C   SEE SUBROUTINE WRITEUP FOR DESCRIPTION OF RECIN
C
    1 CALL RECIN(7,2,IEOF,ID,1,10,1) $ IF(EOF,7)999,2
    2 CALL RECIN(7,1,IEOF,NVEL)
      PRINT 100,ID,NVEL $ NDOFF=0 $ NFOFF=0
C
C  WRITE ID ARRAY BELOW HORIZONTAL AXIS OF PLOT AND DRAW X-Y AXES WITH
C    WITH TIC-MARK GRIDS, NUMBERS AND LABELS
C
      CALL NOTATE(0.,-4.*HGT,.5*HGT,ID,0.,100)
      CALL AXES(0.,0.,0.,VAXL,VMIN,VDEL,VTICL,TICSNV,VLAB,HGT,-50)
      CALL AXES(0.,FH,0.,VAXL,VMIN,VDEL,VTICL,TICSNV,VLAB,HGT,-50)
      CALL AXES(-.5,0.,90.,DAXL,DMIN,DDEL,DTICL,TICSND,DLAB,HGT, 50)
      CALL AXES(-.5,FH,90.,FAXL,FMIN,FDEL,FTICL,TICSNF,FLAB,HGT, 50)
C
C  DRAW LINE FOR ZERO DECAY RATE
C
      IF(DZERO.GT.0.)CALL DASHLN(0.,DZERO,VAXL,DZERO,VAXL)
C
C  90-LOOP IS VELOCITY LOOP
C
      DO 90 IV=1,NVEL $ CALL RECIN(7,1,4,VEL,IK2,NEGR,NTWO)
      NROOT=NTWO-NEGR+1
      CALL RECIN(7,2,NROOT,ROOTR,1,NROOT,1)
      CALL RECIN(7,2,NROOT,ROOTI,1,NROOT,1)
      VEL=(VEL-VMIN)*RVDEL
```

```
C
C   20-LOOP IS ROOT PLOTTING LOOP FOR VELOCITY VEL
C
       DO 20 N=1,NROUT $ RRN=ROOTR(N) $ RIN=ROOTI(N)
C
C   SCALE FOR PLOTTING AND SET UP SYMBOL CYCLING
C
       DECAY=(RRN-DMIN)*RDDEL $ FREQ=(RIN-FMIN)*RFDEL+FH
       IF(IK2.EQ.11)GO TO 10  $ ISYM=QLSYM*(IK2+10)
C
C   PLOT SYMBOL IF DECAY RATE IS ON PLOT - IF OFF-SCALE, INCREMENT NDOFF
C     - REPEAT FOR FREQUENCY, NFOFF
C
       IF(RRN.GE.DMIN.AND.RRN.LE.DMAX)4,5
     4 CALL PNTPLT(VEL,DECAY,ISYM,1) $ GO TO 6
     5 NDOFF=NDOFF+1
     6 IF(RIN.GE.FMIN.AND.RIN.LE.FMAX)7,8
     7 CALL PNTPLT(VEL,FREQ,ISYM,1) $ GO TO 20
     8 NFOFF=NFOFF+1 $ GO TO 20
C
C   PLOT PLUS SIGN IF DECAY RATE IS ON PLOT - IF OFF-SCALE, INCREMENT
C     NDOFF - REPEAT FOR FREQUENCY, NFOFF
C
    10 IF(RRN.GE.DMIN.AND.RRN.LE.DMAX)12,13
    12 CALL NOTATE(VEL,DECAY,.07,3,0.,-1) $ GO TO 14
    13 NDOFF=NDOFF+1
    14 IF(RIN.GE.FMIN.AND.RIN.LE.FMAX)15,16
    15 CALL NOTATE(VEL,FREQ,.07,3,0.,-1) $ GO TO 20
    16 NFOFF=NFOFF+1
    20 CONTINUE
    90 CONTINUE
C
C   ENCODE NUMBER OF POINTS OFF PLOT AND WRITE ON PLOT
C
       ENCODE(10,101,DOFF)NDOFF  $ ENCODE(10,101,FOFF)NFOFF
       CALL NOTATE(VAXL,DAXL,HGT,DOFF,0.,10) $ FFHH=FH+FAXL
       CALL NOTATE(VAXL,FFHH,HGT,FOFF,0.,10)
C
C   SHIFT ORIGIN AND CHECK FOR NEXT CASE
C
       CALL CALPLT((VAXL+6.),0.,-3) $ GO TO 1
   999 CALL CALPLT(0.,0.,999) $ REWIND 7
   101 FORMAT(I4,6HPT OFF)
   100 FORMAT(/2X10A10/* NVEL=*I4)
       END              PROGRAM VPLOT
```

PROGRAM FOR PLOTTING ROOTS IN ROOT-LOCUS FORM WITH WIND VELOCITY AS A PARAMETER

General Description

One form of plotting characteristic roots often used in parametric stability investigations is the plotting of $\mathrm{Im}(\lambda)$ versus $\mathrm{Re}(\lambda)$; that is, frequency versus decay rate. Curves are formed by the roots as a parameter is varied. Root-locus diagrams are often used because the qualitative variation of the roots with the parameter can be sketched with no computation, providing the parameter enters the stability determinant in a simple manner. Wind velocity is used as the parameter and enters the stability determinant in a complicated fashion. Thus, the root-locus plots generated by program RTLOCUS is used only as a form of plotting the calculated results in order to assist in interpreting such trends as the splitting of a complex pair into real roots. One feature of this type of plot is that radial lines from the origin form lines of constant damping ratio. A sample plot is given in figure 4.

The organization and operation of program RTLOCUS is quite similar to that of program VPLOT. The definitions of the FORTRAN variables are also essentially the same. The program uses the same tape 7 as VPLOT and is normally executed in series with VPLOT. The version of RTLOCUS given here requires 30-inch-wide paper.

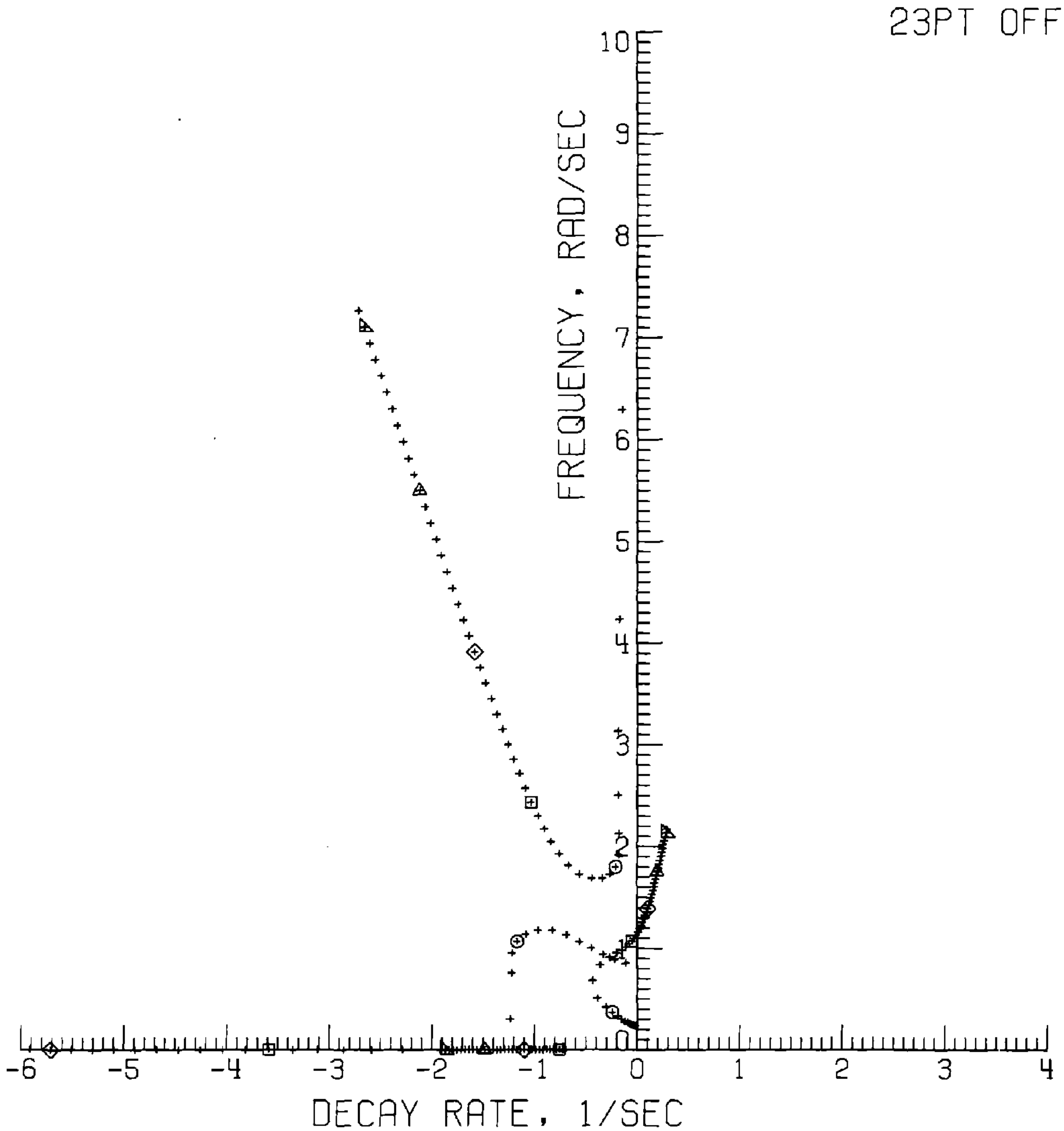


Figure 4.- Example of plot from program RTLOCUS.

```
      OVERLAY(RTLOCUS,0,0)
      PROGRAM RTLOCLS(OUTPUT=1,TAPE7)
C*********************************************************************
C*                                                                   *
C* PROGRAM A2864.4 - ROUT LOCUS PLOTTING PROGRAM                     *
C* PLOTS IMAG(ROOT) VS REAL(ROOT) WITH VELOCITY AS A PARAMETER       *
C* SEE SUBROUTINE WRITEUPS FOR DESCRIPTION OF CALPLT,NOTATE,AXES, OP *
C*   PNTPLT                                                          *
C*                                                                   *
C*********************************************************************
      DIMENSION ID(10),POOTR(50),ROOTI(50),DLAB(5),FLAB(5)
C
C  SCALING PARAMETERS AND LABELS FOR AXES
C
      DATA DLAB/50H                    DECAY RATE, 1/SEC              /
      DATA DAXL,DMIN,DDEL,DTICL,TICSND/10.,-6.,1.,1.,10./
      DATA FAXL,FMIN,FDEL,FTICL,TICSNF/10.,0.,1.,1.,10./
      DATA FLAB/50H                              FREQUENCY, RAD/SEC    /
      DATA HGT,FH/.250,0./
C
C  INITITALIZE PLOTTING ROUTINES, SET PLOT ORGIN, AND CALCULATE SCALING
C    PARAMETERS - CALL TO LFROY SLCWS PLOTTER FOR USING INK
C
      CALL CALCUMP  $  CALL CALPLT(1.,5.*HGT,-3)
      QLSYM=1.  $  IF(QLSYM.GT.0.) CALL LEROY
      RDDEL=1./DDEL $ RFDEL=1./FDEL
      FMAX=FMIN+FAXL*FDEL $ DMAX=DMIN+DAXL*DDEL $ DZERO=-DMIN*RDDEL
C
C  READ ID ARRAY AND NUMBER OF VELOCITY POINTS FROM BINARY TAPE7
C   SEE SUBROUTINE WRITEUP FOR DESCRIPTION OF RECIN
C
    1 CALL RECIN(7,2,IEOF,ID,1,10,1) $ IF(EOF,7)999,2
    2 CALL RECIN(7,1,IEOF,NVEL)
      PRINT 100,ID,NVEL $ NOFF=0
C
C  WRITE ID ARRAY BELOW HORIZONTAL AXIS OF PLOT AND DRAW X-Y AXES WITH
C    WITH TIC-MARK GRIDS, NUMBERS AND LABELS
C
      CALL NOTATE(0.,-4.*HGT,.5*HGT,ID,0.,100)
      CALL AXES(0.,0.,0.,DAXL,DMIN,DDEL,DTICL,TICSND,DLAB,HGT,-50)
      CALL AXES(DZERO,0.,90.,FAXL,FMIN,FDEL,FTICL,TICSNF,FLAB,HGT,50)
C
C  90-LOOP IS VELOCITY LOOP
C
      DO 90 IV=1,NVEL $ CALL RECIN(7,1,4,VEL,IK2,NEGR,NTWO)
      NROOT=NTWO-NEGR+1
      CALL RECIN(7,2,NROOT,ROOTR,1,NROOT,1)
      CALL RECIN(7,2,NROOT,ROOTI,1,NROOT,1)
C
C  20-LOOP IS ROOT PLOTTING LOOP FOR VELOCITY VEL
C
      DO 20 N=1,NROOT $ RRN=ROOTR(N) $ RIN=ROOTI(N)
C
C  SCALE FOR PLOTTING AND SET UP SYMBOL CYCLING
C
      DECAY=(RRN-DMIN)*RDDEL $ FREQ=(RIN-FMIN)*RFDEL+FH
      IF(IK2.EQ.11)GO TO 10  $ ISYM=QLSYM*(IK2+10)
```

```
C
C   IF ON PLOT, PLOT SYMBOL -  COUNT OFF SCALE POINTS
C
       IF((RRN.GE.DMIN.AND.RRN.LE.DMAX) .AND.
     1    (RIN.GE.FMIN.AND.RIN.LE.FMAX))8,9
    8 CALL PNTPLT(DECAY,FREQ,ISYM,1) $  GO TO 20
    9 NOFF=NOFF+1 $ GO TO 20
C
C   IF ON PLOT, PLOT PLUS SIGN - COUNT OFF SCALE POINTS
C
   10 IF((RRN.GE.DMIN.AND.RRN.LE.DMAX) .AND.
     1    (RIN.GE.FMIN.AND.RIN.LE.FMAX))14,15
   14 CALL NOTATE(DECAY,FREQ,.07,3,0.,-1) $ GO TO 20
   15 NOFF=NOFF+1
   20 CONTINUE
   90 CONTINUE
C
C   ENCODE NUMBER OF POINTS OFF PLOT AND WRITE ON PLOT
C
      ENCODE(10,101,POFF)NOFF
      CALL NOTATE(DAXL,FAXL,HGT,POFF,0.,10)
C
C   SHIFT ORIGIN AND CHECK FOR NEXT CASE
C
      CALL CALPLT(15.,0.,-3) $ GO TO 1
  999 CALL CALPLT(0.,0.,999) $ REWIND 7
  101 FORMAT(I4,6HPT OFF)
  100 FORMAT(/2X10A10/* NVEL=*I4)
      END             PROGRAM RTLOCUS

      SUBROUTINE NOTATE(X,Y,HT,BCD,THETA,N)
      CALL CFTRAN(X,Y,HT,BCD,THETA,N)  $  RETURN
      END             SUBROUTINE NOTATE
```

PROGRAM FOR PLOTTING LONGITUDINAL MODES OF MOTION

As a means of illustrating the longitudinal modes of motion, the outline of the tethered balloon and the center of mass are displaced in proportion to the eigenvector and are drawn for a sequence of time intervals by program CALBALM. The balloon is viewed from the side. It is displaced proportional to the pitch-angle amplitude, which is an input to the program, and is selected initially such that the balloon remains on the plot. The center-of-mass position is shown as a plotted point for each time sample, and the balloon outline can be deleted for some time samples. Shifting of the plotting frame between time intervals is also optional. A sample plot is given in figure 5. It might be noted that this is not a transient response problem in the usual sense. In general, all modes of a dynamic system participate in transient motions to a degree depending upon the initial conditions or excitation. The purpose of program CALBALM is to illustrate the character of a single mode.

Program CALBALM is highly specialized as it is based on the shape of a particular balloon given in feet by DATA statements within the program (and the same is true for program CALBLM2 subsequently described). However, it can be modified for other purposes with minor reprograming efforts. For example, a slightly modified version has been used for making computer-generated movies of the modes of motion.

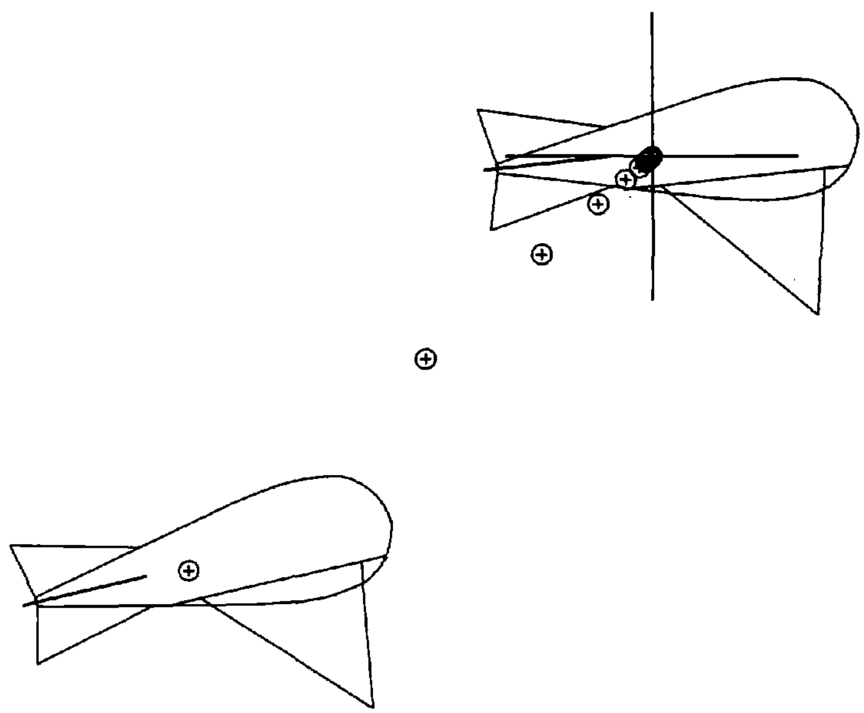


Figure 5.- Example of plot from program CALBALM.

Definition of Program Variables

The principal FORTRAN variables for the program are given and defined except for the variables required for input data which are described subsequently. Note that the coordinate y refers to the plotting coordinate direction which is actually -z for the stability-axis system.

FORTRAN variable name	Definition
AXESL	length of X- and Y-axes of plotting frame for a given time or times, in.
DAY	two-word array containing date and time of processing of case
DT	time increment between frames, 0.0625 * TIME (where TIME is time scale factor)
HGT	height of title (identification and date and time) written underneath initial frame
NB	number of points used to describe balloon half profile
NF	number of points in array used to describe fins
NLB	number of points in array used to describe load band
NT	number of points in array used to describe tether bridle
XAXE, YAXE	arrays containing coordinates for drawing axes for equilibrium position of balloon
XBAL, YBAL, XBTM, YBTM	x- and y-coordinates of balloon profile, given as station coordinates in feet from nose with balloon facing to the left (rotated and translated coordinates including motion are stored in XBTM and YBTM for plotting)
XCG, YCG	x- and y-coordinates of balloon center of mass in coordinate system consistent with XBAL and YBAL

FORTRAN variable name	Definition
XFIN, YFIN, XFTM, YFTM	fin coordinate array in coordinate system consistent with XBAL and YBAL (rotated and translated coordinates are stored in XFTM and YFTM)
XLBAND, YLBAND, XLBDTM, YLBDTM	load band coordinate array in coordinate system consistent with XBAL and YBAL, including motion stored in XLBDTM and YLBDTM
XMIN, YMIN, XMAX, YMAX	minimum and maximum values of x and y for limits of plotting frame, in units relating to full-scale balloon (here ft)
XMG, YMG	x- and y-coordinates of center of mass, including motion scaled to inches on plot
XTET, YTET, XTTM, YTTM	tether bridle x- and y-arrays in coordinate system consistent with XBAL and YBAL, including motion stored in XTTM and YTTM

Description of Input Data

The input data consist of four cards per case. The program can process multiple cases.

Card 1 (8A10 format).- Eighty columns of identification information read into ID array.

Card 2 (8F10.0 format).- The eight 10-column fields contain the following variables in sequence:

ROOTR	real part of eigenvalue for mode (decay rate), 1/sec
ROOTI	imaginary part of eigenvalue for mode (frequency), 1/sec
XREAL	real part of x-component of eigenvector normalized by θ, m/deg
XIMAG	imaginary part of x-component of eigenvector normalized by θ, m/deg
ZREAL	real part of z-component of eigenvector normalized by θ, m/deg

ZIMAG imaginary part of z-component of eigenvector normalized by θ, m/deg

TREAL real part of θ-component of eigenvector, normally 1.0

TIMAG imaginary part of θ-component of eigenvector, normally 0.0

Card 3 (8F10.0 format).- The first four 10-column fields contain the following variables in sequence:

AMPL initial amplitude of pitching motion for mode, deg

TRIMA trim angle of attack, deg

TIME time scale factor, $DT = \Delta t = 0.0625 * TIME$, seconds between frames

XSHIFT distance between origins of frames in inches on plot if frames are shifted between plot time intervals

Card 4 (20I4 format).- The first four 4-column fields contain the following integer variables in sequence:

NFRM number of time increments

ISHIFT frame origin is shifted between plot time intervals only if = 0

MODBAL parameter determining interval for drawing full balloon outline (center-of-mass point is plotted MODBAL times as often as balloon outline)

INIMOD initial time frame for plotting ballon outline (= 0, outline plotted first frame; = 1, second frame, etc.)

Listing of Input Data Cards for Sample Case

```
******************************** COLUMN NUMBER ***********************************
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890

LRC BALLOON - REFERENCE CONFIGURATION - MODE 1A AT V=20 M/S, ROOT=-.763+0.*I
   -.76353         0.   -1.3920         0.    1.2549         0.         1.         0.
        7.     5.980       15.       10.
  10   1   9   0
```

```
      OVERLAY(CALBALM,0,0)
      PROGRAM CALBALM(OUTPUT=1,INPUT=1,TAPE5=INPUT)
C**********************************************************************
C*                                                                    *
C* PROGRAM A2864.5 - LONGITUDINAL EIGENVECTOR PLOTTING PROGRAM        *
C* PLOTS C G POSITION AND/OR BALLOON OUTLINE FOR A SINGLE MODE OF     *
C*   MOTION FOR SELECTED INCREMENTS OF TIME                           *
C*  SEE SUBROUTINE WRITEUPS FOR DESCRIPTION OF CALPLT,NOTATE,LINE, OR *
C*   PNTPLT - LEROY SLOWS PLOTTER FOR DRAWING WITH INK                *
C*                                                                    *
C**********************************************************************
      DIMENSION XBAL(46),XFIN(12),XLBAND(2),XTET(3),XAXE(8)
     +  ,DAY(2),YBAL(46),YFIN(12),YLBAND(2),YTET(3),YAXE(8)
     +   ,ID(8),XBTM(48),XFTM(14),XLBDTM(4),XTTM(5)
     +          ,YBTM(48),YFTM(14),YLBDTM(4),YTTM(5)
      DATA AI/6H(8A10)/,F10/8H(8F10.0)/,AO/10H(/X10A10/)/
      DATA I4,AOW/6H(20I4),7H(13A10)/
      DATA HGT/.125/,AXESL/10./,ISHIFT/0/
      DATA DEGRAD/.0174532925199433/
      DATA FTPERM/3.280839895/                                        SI UNITS
C
C  DATA FOR LRC BALLOON IN FEET
C  BALLOON DATA IS FOR NOSE FACING LEFT - TURNED AROUND IN CALCULATIONS
C
      DATA NB,NF,NLB,NT/23,12,2,3/
      DATA (XBAL(I),I=1,23)/0.,.01,.07,.2,.33,.5,.67,1.,1.33,1.67,2.,
     + 2.33,2.67,3.,3.33,4.,4.67,5.33,6.67,8.,9.33,10.51,25./
      DATA (YBAL(I),I=1,23)/0.,.12,.45,.95,1.33,1.72,2.03,2.52,2.88,
     + 3.18,3.42,3.63,3.77,3.87,3.95,4.06,4.14,4.18,4.13,3.99,3.78,
     + 3.57,.33/
      DATA XFIN/17.22,25.93,2*25.,2*17.22,2*25.93,2*25.,25.93,17.22/
      DATA YFIN/2.07,4.25,.3,2*.03,2*-.03,2*.03,-.33,-4.25,-2.07/
      DATA XLBAND/.88,15.97/,YLBAND/2*-2.33/
      DATA XTET/2.6,4.13,13.94/,YTET/-2.33,-12.51,-2.33/
      DATA XMC/15.42/,YMC/0./,XCG/14.32/,YCG/-.29/
      DATA XMIN,XMAX,YMIN,YMAX/-60.,50.,-60.,50./
C
C  REFLECT BALLOON HALF-PROFILE, INITIALIZE PLOTTING ROUTINES, COMPUTE
C    SCALING PARAMETERS, AND SET ORIGIN FOR INITIAL FRAME
C
      NB1=NB+1$ NB2=NB+NB$ DO 1 I=NB1,NB2$ IB=NB2+1-I$ XBAL(I)=XBAL(IB)
    1 YBAL(I)=-YBAL(IB)
      SCALE=(XMAX-XMIN)/AXESL $ RSCL=1./SCALE $ ZER=.75
      XAXE(1)=-10. $ XAXE(2)=10. $ XAXE(3)=XAXE(7)=XMIN
      XAXE(5)=XAXE(6)=0. $ XAXE(4)=XAXE(8)=SCALE
      YAXE(1)=YAXE(2)=0. $ YAXE(3)=YAXE(7)=YMIN
      YAXE(4)=YAXE(8)=SCALE $ YAXE(5)=-10. $ YAXE(6)=10.
      CALL CALCOMP  $  CALL CALPLT(ZER,ZER,-3)  $  CALL LEROY
C
C  READ INDENTIFICATION ARRAY, CALL FOR DATE AND TIME, AND PRINT AND
C    WRITE AT THE BOTTOM OF THE FIRST FRAME
C
   10 READ AI,ID $ IF(EOF,5)999,11
   11 CALL DAYTIM(DAY) $ PRINT AO,ID,DAY
      CALL NOTATE(0.,-.5,HGT,ID,0.,80)
C
C  READ EIGENVECTOR-MOTION DATA, TRIM PARAMETERS, AND FRAME DATA
C
      READ F10,ROOTR,ROOTI,XREAL,XIMAG,ZREAL,ZIMAG,TREAL,TIMAG
      READ F10,AMPL,TRIMA,TIME,XSHIFT$ READ I4,NFRM,ISHIFT,MODBAL,INIMOD
      INIMOD=INIMOD+MODBAL
```

```
C
C   CONVERT DISPLACEMENT TO FEET, SCALE TIME, AND COMPUTE MAGNITUDES AND
C     PHASES OF MODAL COMPONENTS
C
        XREAL=FTPERM*XREAL  $ XIMAG=FTPERM*XIMAG
        ZREAL=FTPERM*ZREAL  $ ZIMAG=FTPERM*ZIMAG
        DT=.0625*TIME
        DTW=DT*ROOTI  $ DTR=DT*ROOTR
        XABS=SQRT(XREAL**2+XIMAG**2)  $ PHIX=ATAN2(XIMAG,XREAL)
        ZABS=SQRT(ZREAL**2+ZIMAG**2)  $ PHIZ=ATAN2(ZIMAG,ZREAL)
        CPX=COS(PHIX) $ SPX=SIN(PHIX) $ CPZ=COS(PHIZ) $ SPZ=SIN(PHIZ)
C
C   50-LOOP IS TIME-FRAME LOOP
C
        DO 50 NTS=1,NFRM $ N=NTS-1
C
C   CALCULATE MOTION FOR TIME FRAME
C
        ST=SIN(N*DTW) $ CT=COS(N*DTW) $ AET=AMPL*EXP(N*DTR) $ THEM=AET*CT
        XM=AET*XABS*(CT*CPX-ST*SPX) $ ZM=AET*ZABS*(CT*CPZ-ST*SPZ)
        TH=DEGRAD*(THEM+TRIMA) $ CTH=COS(TH) $ STH=SIN(TH)
        IF(ISHIFT.EQ.0)GO TO 18
C
C   IF C G POINT ONLY IS DRAWN FOR THIS TIME FRAME GO TO 40
C
        IF(MOD(N+INIMOD,MODBAL).NE.0)GO TO 40
C
C   20-LOOP PREPARES ARRAYS FOR BALLOON OUTLINE FOR PLOTTING
C     25-LOOP - FINS, 30-LOOP - LOAD BAND, AND 35-LOOP - TETHER BRIDLE
C   POINTS OFF-SCALE ARE SET TO OUTER LIMITS OF FRAME
C
   18 DO 20 I=1,NB2   $ XTEM=XBAL(I)-XCG $ YTEM=YBAL(I)-YCG
        YBTM(I)=YTEM*CTH-XTEM*STH-ZM $ XBTM(I)=-(XTEM*CTH+YTEM*STH-XM)
        IF(XBTM(I).LT.XMIN)XBTM(I)=XMIN $ IF(XBTM(I).GT.XMAX)XBTM(I)=XMAX
        IF(YBTM(I).LT.YMIN)YBTM(I)=YMIN $ IF(YBTM(I).GT.YMAX)YBTM(I)=YMAX
   20 CONTINUE $ DO 25 I=1,NF $ XTEM=XFIN(I)-XCG $ YTEM=YFIN(I)-YCG
        YFTM(I)=YTEM*CTH-XTEM*STH-ZM $ XFTM(I)=-(XTEM*CTH+YTEM*STH-XM)
        IF(XFTM(I).LT.XMIN)XFTM(I)=XMIN $ IF(XFTM(I).GT.XMAX)XFTM(I)=XMAX
        IF(YFTM(I).LT.YMIN)YFTM(I)=YMIN $ IF(YFTM(I).GT.YMAX)YFTM(I)=YMAX
   25 CONTINUE $ DO 30 I=1,NLB $ XTEM=XLBAND(I)-XCG $ YTEM=YLBAND(I)-YCG
        YLBDTM(I)=YTEM*CTH-XTEM*STH-ZM $ XLBDTM(I)=-(XTEM*CTH+YTEM*STH-XM)
        IF(XLBDTM(I).LT.XMIN)XLBDTM(I)=XMIN
        IF(XLBDTM(I).GT.XMAX)XLBDTM(I)=XMAX
        IF(YLBDTM(I).LT.YMIN)YLBDTM(I)=YMIN
        IF(YLBDTM(I).GT.YMAX)YLBDTM(I)=YMAX
   30 CONTINUE $ DO 35 I=1,NT $ XTEM=XTET(I)-XCG $ YTEM=YTET(I)-YCG
        YTTM(I)=YTEM*CTH-XTEM*STH-ZM $ XTTM(I)=-(XTEM*CTH+YTEM*STH-XM)
        IF(XTTM(I).LT.XMIN)XTTM(I)=XMIN $ IF(XTTM(I).GT.XMAX)XTTM(I)=XMAX
        IF(YTTM(I).LT.YMIN)YTTM(I)=YMIN $ IF(YTTM(I).GT.YMAX)YTTM(I)=YMAX
   35 CONTINUE
C
C   SET ADJUSTED MINIMUM AND SCALE FACTOR, AND PLOT BALLOON OUTLINE BY
C     CONNECTING POINTS
C
        XBTM(NB2+1)=XFTM(NF+1)=XLBDTM(NLB+1)=XTTM(NT+1)=XMIN
        YBTM(NB2+1)=YFTM(NF+1)=YLBDTM(NLB+1)=YTTM(NT+1)=YMIN
        XBTM(NB2+2)=XFTM(NF+2)=XLBDTM(NLB+2)=XTTM(NT+2)=SCALE
        YBTM(NB2+2)=YFTM(NF+2)=YLBDTM(NLB+2)=YTTM(NT+2)=SCALE
        CALL LINE(XBTM,YBTM,NB2,1,0,1,.1)
        CALL LINE(XFTM,YFTM,NF,1,0,1,.1)
        CALL LINE(XLBDTM,YLBDTM,NLB,1,0,1,.1)
        CALL LINE(XTTM,YTTM,NT,1,0,1,.1)
```

```
C
C   DRAW X AND Z AXES FOR EQUILIBRIUM BALLOON POSITION
C
      CALL LINE(XAXE,YAXE,2,1,0,1,.1)
      CALL LINE(XAXE(5),YAXE(5),2,1,0,1,.1)
C
C   PLOT C G POSITION FOR TIME FRAME
C
   40 XMG=-(XMIN-XM)*RSCL$YMG=-(YMIN+ZM)*RSCL$ CALL PNTPLT(XMG,YMG,11,1)
C
C   SHIFT PLOT ORIGIN IF ISHIFT IS 0
C
      IF(ISHIFT.EQ.0)CALL CALPLT(XSHIFT,0.,-3)
   50 CONTINUE
C
C   SHIFT PLOT ORIGIN AND CHECK FOR NEXT CASE
C
      CALL CALPLT((AXESL+XSHIFT),0.,-3) $ GO TO 10
C
  999 CALL CALPLT(0.,0.,999)
      END              PROGRAM CALBALM
```

PROGRAM FOR PLOTTING LATERAL MODES OF MOTION

Program CALBLM2 plots the lateral modes of motion in similar fashion to that described for the longitudinal program. The organization of this program is basically the same as that for the longitudinal program. However, the additional complication of treating the rolling motion about the stability axis requires the z-coordinates of the components to be given in addition to the x- and y-coordinates. The apparent shape of the balloon must also be altered as a result of the trim pitch angle. Roll displacements are treated as linearized displacements and the hidden portion of the top fin is deleted in approximate fashion. The eigenvector is normalized by yaw angle ψ. An example of a plot generated by this program is given in figure 6.

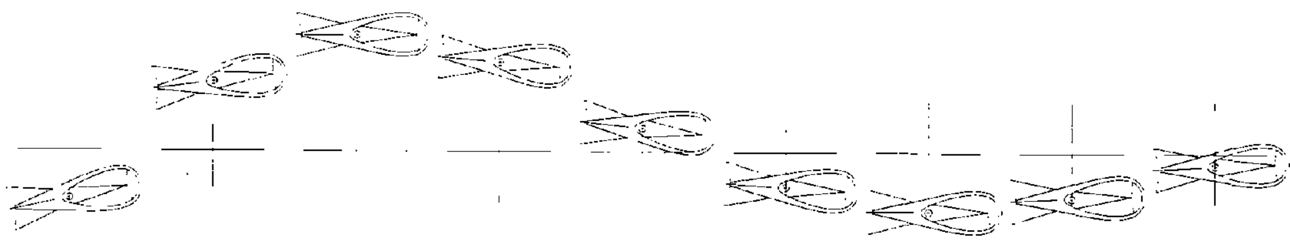


Figure 6.- Example of plot from program CALBLM2.

The four input-data cards are essentially the same as for the longitudinal program with the following exceptions: For Card 2, the third through eighth 10-column fields contain

YREAL	real part of y-component of eigenvector normalized by ψ, m/deg
YIMAG	imaginary part of y-component of eigenvector normalized by ψ, m/deg
PREAL	real part of ϕ-component of eigenvector normalized by ψ, nondimensional
PIMAG	imaginary part of ϕ-component of eigenvector normalized by ψ, nondimensional
SREAL	real part of ψ-component of eigenvector, normally 1.0
SIMAG	imaginary part of ψ-component of eigenvector, normally 0.0

For Card 3, the variable AMPL refers to the initial yaw angle ψ, in degrees.

Listing of Input Data Cards for Sample Case

```
******************************* COLUMN NUMBER ***********************************
00000000011111111112222222222333333333344444444445555555555666666666677777777778
12345678901234567890123456789012345678901234567890123456789012345678901234567890

LRC BALLOON - REFERENCE CONFIGURATION - MODE 2 AT V=20 M/S, ROOT=-.089+.407*I
  -.088919     .40736    -.23446    -.80488    .003732   -.015992          1.         0.
       12.      5.980     30.848        2.5
   9    0    1    0
```

```
      OVERLAY(CALBLM2,0,0)
      PROGRAM CALBLM2(OUTPUT=1,INPUT=1,TAPE5=INPUT)
C***********************************************************************
C*                                                                     *
C* PROGRAM A2864.6 - LATERAL EIGENVECTOR PLOTTING PROGRAM              *
C* PLOTS C G POSITION AND/OR BALLOON OUTLINE FOR A SINGLE MODE OF      *
C*   MOTION FOR SELECTED INCREMENTS OF TIME                            *
C*  SEE SUBROUTINE WRITEUPS FOR DESCRIPTION OF CALPLT,NOTATE,LINE, OR  *
C*   PNTPLT - LEROY SLOWS PLOTTER FOR DRAWING WITH INK                 *
C*                                                                     *
C***********************************************************************
      DIMENSION XBAL(46),XFIN(16),XLBAND(46),XTET(6),XAXE(8)
     +  ,DAY(2),YBAL(46),YFIN(16),YLBAND(46),YTET(6),YAXE(8)
     +         ,ZBAL(46),ZFIN(16),ZLBAND(46),ZTET(6),ZAXE(4)
     +   ,IC(8),XBTM(48),XFTM(24),XLBDTM(48),XTTM(10),XTRIM(114)
     +   ,IN(2),YBTM(48),YFTM(24),YLBDTM(48),YTTM(10),ZTRIM(114)
      DATA AI/6H(8A10)/,F10/8H(8F10.0)/,AO/10H(/X10A10/)/
      DATA I4,AOW/6H(20I4),7H(13A10)/
      DATA HGT/.125/,AXESL/10./,ISHIFT/0/
      DATA DEGRAD/.0174532925199433/
      DATA FTPERM/3.280839895/                                        SI UNITS
C
C  DATA FOR LRC BALLOON IN FEET
C  BALLOON DATA IS FOR NOSE FACING LEFT - TURNED AROUND IN CALCULATIONS
C
      DATA NB,NF,NLB,NT/23,16,23,6/
      DATA (XBAL(I),I=1,23)/0.,.01,.07,.2,.33,.5,.67,1.,1.33,1.67,2.,
     + 2.33,2.67,3.,3.33,4.,4.67,5.33,6.67,8.,9.33,10.51,25./
      DATA (YBAL(I),I=1,23)/0.,.12,.45,.95,1.33,1.72,2.03,2.52,2.88,
     + 3.18,3.42,3.63,3.77,3.87,3.95,4.06,4.14,4.18,4.13,3.99,3.78,
     + 3.57,.33/
      DATA ZBAL/46*0./
      DATA XFIN/17.22,25.93,25.,2*17.22,25.93,25.,2*17.22,25.93,25.,
     + 2*17.22,25.93,25.,17.22/
      DATA YFIN/4*0.,2.07,4.25,.33,2.08,4*0.,-2.07,-4.25,-.33,-2.07/
      DATA ZFIN/-2.07,-4.25,-.33,-2.07,4*0.,2.07,4.25,.33,2.07,4*0./
      DATA (YLBAND(I),I=1,23)/0.,.25,.95,1.69,2.16,2.5,2.78,2.96,3.09,
     + 3.19,3.32,3.42,3.47,3.41,3.23,2.97,2.7,2.25,1.53,1.07,.63,.27,0./
      DATA (XLBAND(I),I=1,23)/.89,.92,1.,1.33,1.67,2.,2.33,2.67,3.,
     + 3.33,4.,4.67,5.33,6.67,8.,9.33,10.51,12.,14.,15.,15.73,15.98,
     + 16.06/
      DATA ZLBAND/46*2.333333333333/
      DATA XTET/2.64,4.13,2.64,13.79,4.13,13.79/
      DATA YTET/2.93,0.,-2.93,1.62,0.,-1.62/
      DATA ZTET/2.33,12.51,2*2.33,12.51,2.33/
      DATA XMC/15.42/,YMC/0./,XCG/14.32/,YCG/0./,ZCG/.29/
      DATA XMIN,XMAX,YMIN,YMAX/-60.,50.,-60.,50./
C
C  REFLECT BALLOON HALF-PROFILE, INITIALIZE PLOTTING ROUTINES, COMPUTE
C    SCALING PARAMETERS, AND SET ORIGIN FOR INITIAL FRAME
C
      NLB1=NLB+1 $ NLB2=NLB+NLB $ DO 101 I=NLB1,NLB2
      IB=NLB2+1-I $ XLBAND(I)=XLBAND(IB)
  101 YLBAND(I)=-YLBAND(IB)
      NTOT=2*NB+NF+2*NLB+NT $ NT2=NT/2 $ NF4=NF/4
      NB1=NB+1$ NB2=NB+NB$ DO 1 I=NB1,NB2$ IB=NB2+1-I$ XBAL(I)=XBAL(IB)
    1 YBAL(I)=-YBAL(IB)
      SCALE=(XMAX-XMIN)/AXESL $ RSCL=1./SCALE $ ZER=.25
      XAXE(1)=-10. $ XAXE(2)=10. $ XAXE(3)=XAXE(7)=XMIN
      XAXE(5)=XAXE(6)=0. $ XAXE(4)=XAXE(8)=SCALE
      YAXE(1)=YAXE(2)=0. $ YAXE(3)=YAXE(7)=YMIN
      YAXE(4)=YAXE(8)=SCALE $ YAXE(5)=-10. $ YAXE(6)=10.
      CALL CALCOMP  $  CALL CALPLT(ZER,ZER,-3)  $  CALL LEROY
```

```
C
C   READ INCENTIFICATICN ARRAY, CALL FOR DATE AND TIME, AND PRINT AND
C     WRITE AT THE BOTTOM OF THE FIRST FRAME
C
   10 REAC AI,ID $ IF(EOF,5)999,11
   11 CALL CAYTIM(DAY) $ PRINT AO,ID,DAY
      CALL NOTATE(0.,0.,HGT,ID,0.,80)
C
C   READ EIGENVECTOR-MOTION DATA, TRIM PARAMETERS, AND FRAME DATA
C
      REAC F10,ROOTR,ROOTI,YREAL,YIMAG,PREAL,PIMAG,SREAL,SIMAG
      REAC F10,AMPL,TRIMA,TIME,XSHIFT$ READ I4,NFRM,ISHIFT,MODBAL,INIMOD
      INIMOD=INIMOC+MCDBAL
C
C   CONVERT DISPLACEMENT TO FEET, SCALE TIME, AND COMPUTE MAGNITUDES AND
C     PHASES OF MODAL COMPONENTS
C
      DT=.C625*TIME $ AMPL=DEGRAC*AMPL
      DTW=DT*ROOTI $ DTR=DT*ROOTR
      TH=DEGRAD*TRIMA $ CTH=COS(TH) $ STH=SIN(TH)
      YABS=SQRT(YREAL**2+YIMAG**2)/DEGRAD $ PHIY=ATAN2(YIMAG,YREAL)
      YABS=FTPERM*YABS
      PABS=SQRT(PREAL**2+PIMAG**2) $ PHIP=ATAN2(PIMAG,PREAL)
      CPY=COS(PHIY) $ SPY=SIN(PHIY) $ CPP=COS(PHIP) $ SPP=SIN(PHIP)
C
C   ROTATE BALLOON IN PITCH TO TRIM ANGLE
C
      DO 15 I=1,NTOT $ XTEM=XBAL(I)-XCG $ ZTEM=ZBAL(I)-ZCG
      XTRIM(I)=-XTEM*CTH+ZTEM*STH
   15 ZTRIM(I)=XTEM*STH+ZTEM*CTH
      NB21=NB2+1
      ZTRIM(NB21)=ZTRIM(NB21+3)=ZTRIM(NB21)+(XTRIM(NB)-XTRIM(NB21))
     + /(XTRIM(NB21+1)-XTRIM(NB21))*(ZTRIM(NB2+2)-ZTRIM(NB21))
      XTRIM(NB21)=XTRIM(NB21+3)=XTRIM(NB)
C
C   50-LOOP IS TIME-FRAME LOOP
C
      DO 50 NTS=1,NFRM $ N=NTS-1
C
C   CALCULATE MOTICN FCR TIME FRAME
C
      ST=SIN(N*DTW) $ CT=COS(N*DTW) $ AET=AMPL*EXP(N*DTR) $ SM=AET*CT
      YM=-AET*YABS*(CT*CPY-ST*SPY) $ PM=-AET*PABS*(CT*CPP-ST*SPP)
      CS=COS(SM) $ SS=SIN(SM) $ IF(ISHIFT.EQ.0)GO TO 18
C
C   IF C G POINT ONLY IS DRAWN FOR THIS TIME FRAME GO TO 40
C
      IF(MOD(N+INIMOD,MODBAL).NE.0)GO TO 40
C
C   20-LOOP PREPARES ARRAYS FOR BALLCON OUTLINE FOR PLOTTING
C     25-LOOP - FINS, 30-LOOP - LOAD BAND, AND 35-LOOP - TETHER BRIDLE
C   POINTS OFF-SCALE ARE SET TO OUTER LIMITS OF FRAME
C   SET ADJUSTED MINIMUM AND SCALE FACTOR, AND PLOT BALLOON OUTLINE BY
C     CONNECTING POINTS
C
   18 DO 20 I=1,NB2 $ XTEM=XTRIM(I) $ YTEM=YBAL(I)-ZTRIM(I)*PM
      XBTM(I)=XTEM*CS+YTEM*SS $ YBTM(I)=XTEM*SS-YTEM*CS-YM
      IF(XBTM(I).LT.XMIN)XBTM(I)=XMIN $ IF(XBTM(I).GT.XMAX)XBTM(I)=XMAX
      IF(YBTM(I).LT.YMIN)YBTM(I)=YMIN $ IF(YBTM(I).GT.YMAX)YBTM(I)=YMAX
   20 CONTINUE
      XBTM(NB2+1)=XMIN $ XBTM(NB2+2)=SCALE
      YBTM(NB2+1)=YMIN $ YBTM(NB2+2)=SCALE
      CALL LINE(XBTM,YBTM,NB2,1,0,1,.1)
```

```
      DO 28 IF=1,4 $ K=1+(IF-1)*NF4
      DO 25 J=1,NF4 $ I=J+K+2*(IF-1)-1 $ II=NB2+J+K-1
      XTEM=XTRIM(II) $ YTEM=YBAL(II)-ZTRIM(II)*PM
      XFTM(I)=XTEM*CS+YTEM*SS $ YFTM(I)=XTEM*SS-YTEM*CS-YM
      IF(XFTM(I).LT.XMIN)XFTM(I)=XMIN $ IF(XFTM(I).GT.XMAX)XFTM(I)=XMAX
      IF(YFTM(I).LT.YMIN)YFTM(I)=YMIN $ IF(YFTM(I).GT.YMAX)YFTM(I)=YMAX
   25 CONTINUE
      XFTM(I+1)=XMIN $ XFTM(I+2)=SCALE
      YFTM(I+1)=YMIN $ YFTM(I+2)=SCALE $ KI=K+2*(IF-1)
   28 CALL LINE(XFTM(KI),YFTM(KI),NF4,1,0,1,.1)
      DO 30 I=1,NLB2 $ II=I+NB2+NF
      XTEM=XTRIM(II) $ YTEM=YBAL(II)-ZTRIM(II)*PM
      XLBDTM(I)=XTEM*CS+YTEM*SS $ YLBDTM(I)=XTEM*SS-YTEM*CS-YM
      IF(XLBDTM(I).LT.XMIN)XLBDTM(I)=XMIN
      IF(YLBDTM(I).LT.YMIN)YLBDTM(I)=YMIN
      IF(XLBDTM(I).GT.XMAX)XLBDTM(I)=XMAX
      IF(YLBDTM(I).GT.YMAX)YLBDTM(I)=YMAX
   30 CONTINUE
      XLBDTM(NLB2+1)=XMIN $ XLBDTM(NLB2+2)=SCALE
      YLBDTM(NLB2+1)=YMIN $ YLBDTM(NLB2+2)=SCALE
      CALL LINE(XLBDTM,YLBDTM,NLB2,1,0,1,.1)
      DO 38 IT=1,2 $ K=1+(IT-1)*NT2
      DO 35 J=1,NT2 $ I=J+K+2*(IT-1)-1 $ II=NB2+NF+NLB2+J+K-1
      XTEM=XTRIM(II) $ YTEM=YBAL(II)-ZTRIM(II)*PM
      XTTM(I)=XTEM*CS+YTEM*SS $ YTTM(I)=XTEM*SS-YTEM*CS-YM
      IF(XTTM(I).LT.XMIN)XTTM(I)=XMIN $ IF(XTTM(I).GT.XMAX)XTTM(I)=XMAX
      IF(YTTM(I).LT.YMIN)YTTM(I)=YMIN $ IF(YTTM(I).GT.YMAX)YTTM(I)=YMAX
   35 CONTINUE
      XTTM(I+1)=XMIN $ XTTM(I+2)=SCALE
      YTTM(I+1)=YMIN $ YTTM(I+2)=SCALE $ KI=K+2*(IT-1)
   38 CALL LINE(XTTM(KI),YTTM(KI),NT2,1,0,1,.1)
C
C     DRAW X AND Y AXES FOR EQUILIBRIUM BALLOON POSITION
C
      CALL LINE(XAXE,YAXE,2,1,0,1,.1)
      CALL LINE(XAXE(5),YAXE(5),2,1,0,1,.1)
C
C     PLOT C G POSITION FOR TIME FRAME
C
   40 XMG=-XMIN*RSCL $ YMG=-(YMIN+YM)*RSCL $ CALL PNTPLT(XMG,YMG,11,1)
C
C     SHIFT PLOT ORIGIN IF ISHIFT IS 0
C
      IF(ISHIFT.EQ.0)CALL CALPLT(XSHIFT,0.,-3)
   50 CONTINUE
C
C     SHIFT PLOT ORIGIN AND CHECK FOR NEXT CASE
C
      CALL CALPLT((AXESL+XSHIFT),0.,-3) $ GO TO 10
C
  999 CALL CALPLT(0.,0.,999)
      END              PROGRAM CALBLM2
```

Langley Research Center,

National Aeronautics and Space Administration,

Hampton, Va., April 9, 1973.

APPENDIX

DESCRIPTIONS OF SELECTED SUBROUTINES

Basic Subroutines

Usage descriptions are given for several of the basic subroutines called by the programs of this report. Subroutines QUADET and ROMBERG were written by the authors. The versions of REIG and MATRIX given herein are modified versions of the LRC computer system library. The subprograms CXINV, DAYTIM, RECIN, and RECOUT are LRC computer system library subroutines. Note that listings of QUADET, ROMBERG, REIG, MATRIX, and CXINV are given in the listing of the longitudinal program STABLTY. The listings of the RECIN, RECOUT, and DAYTIM are not given, but the usage descriptions are given to facilitate replacement with equivalent routines if necessary.

In addition to the above subprograms, a FORTRAN subroutine to simulate the COMPASS subroutine MASCNT is described and listed. The subroutine MATINV which is called by MASCNT (simulator) is also described and listed.

The subroutines are described in the following order:

	Page
QUADET	108
ROMBERG	111
REIG	113
MATRIX	115
CXINV	117
RECIN	119
RECOUT	121
DAYTIM	124
MASCNT	126
MATINV	128

Subroutine QUADET

LANGUAGE: FORTRAN and COMPASS

PURPOSE: To convert an $n \times n$ matrix equation, $[A]\lambda^2 + [B]\lambda + [C] = 0$, to a $2n \times 2n$ matrix equation of standard eigenvalue form, $[E] - \lambda[I] = 0$.

USE: CALL QUADET (A, B, C, NMAX, NMAX2, N, IOP, EIGDET, CNO)

A — A two-dimensional input array containing the coefficients of λ^2. The matrix is not destroyed.

B — A two-dimensional input array containing the coefficients of λ. The matrix is not destroyed.

C — A two-dimensional input array containing constant coefficients. The matrix is not destroyed.

NMAX — Column length (number of rows) of A as dimensioned in the calling program.

NMAX2 — Column length (number of rows) of EIGDET as dimensioned in the calling program.

N — The order of A, B, and C.

IOP — An integer, 10 or 11 supplied by user to select option for calculating the condition number of A:
IOP = 10 Condition number of A is calculated.
IOP = 11 Condition number of A is not calculated.

EIGDET — The two-dimensional eigenvalue output array.

CNO — CNO is the Turing condition number of A. If 0.0 is returned from MASCNT for the determinant of A (singular A), CNO = -1.

RESTRICTIONS: Arrays A, B, C, and EIGDET are used with variable dimensions in the subroutine. The maximum size in the calling program must be A (NMAX, N), B (NMAX, N), C (NMAX, N), EIGDET (NMAX2, NMAX2); NMAX2 $\geqq$ 2 *NMAX. A must be nonsingular. Restricted to real arrays.

METHOD:

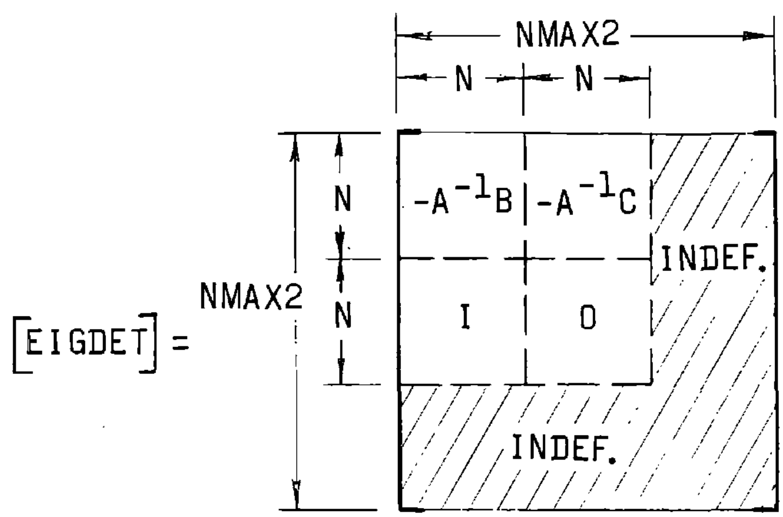
The $n \times n$ system $[A]\lambda^2 + [B]\lambda + [C] = 0$ is equivalent to the $2n \times 2n$ system (refs. (a) and (b)). $[E] - \lambda[I] = 0$, where

$$[E] = \left[\begin{array}{c|c} -A^{-1}B & -A^{-1}C \\ \hline I & 0 \end{array}\right]$$

QUADET performs these matrix manipulations within the storage allocated for EIGDET. The matrix solutions are obtained from a matrix routine written in COMPASS (ref. (c)). The arrays A, B, C, and EIGDET are treated as one-dimensional arrays. A, B, C are transferred into locations (NMAX2 **2-3 * N * N) through (NMAX2 **2) of EIGDET. MASCNT is called and $A^{-1}B$ is returned in the location of B and $A^{-1}C$ is returned in the location of C in EIGDET. If IOP = 10, A^{-1} is returned in A and Turing's condition number defined as

$$C_A = \|A\| * \|A^{-1}\| / N$$

is calculated. The matrices $[A^{-1}B]$, $[A^{-1}C]$, $[I]$, and $[0]$ are then transferred to the upper left $2n \times 2n$ portion of EIGDET and the remainder of EIGDET filled with indefinites (ϕ 1777 0000 0000 0000 1777) such that EIGDET is returned as



ACCURACY: The number of figures lost in the operations $A^{-1}B$ and $A^{-1}C$ can be estimated as $\log_{10}$(CNO) where CNO is condition number of A.

REFERENCES:

(a) Pipes, Louis A.; and Hovanessian, Shahen A.: Matrix-Computer Methods in Engineering. John Wiley & Sons, Inc., c.1969, pp. 265-267.

(b) Frazer, R. A.; Duncan, W. J.; and Collar, A. R.: Elementary Matrices. Cambridge Univ. Press, 1960, p. 289.

(c) Anon.: Control Data 6000 Series Computer Systems Matrix Algebra Subroutines Reference Manual. Publ. No. 60135200, Control Data Corp., June 1966.

(d) Marcus, Marvin: Basic Theorems in Matrix Theory. Nat. Bur. Stand. Appl. Math. Ser. 57, U.S. Dep. Com., Jan. 22, 1960, p. 21. (Reprinted 1964.)

STORAGE:

QUADET	333_8	
MASCNT	431_8	
LABELED COMMON	1130_8	- /IROW/IROW(300) /ICOL/ICOL(300)

SUBPROGRAM USED: MASCNT

Subroutine ROMBERG

LANGUAGE: FORTRAN

PURPOSE: To integrate the function FUN(X) between the limits A and B.

USE: CALL ROMBERG (SUM, A, B, FUN, EPS)

SUM	The computed value of the integral.
A	The lower limit of integration.
B	The upper limit of integration.
FUN	The name of the integrand function.
EPS	Relative error criterion.

An example of the usage follows. A segment of the program to evaluate $\int_a^b \cos(t^2 + p)\, dt$ might be:

```
EXTERNAL FUN
COMMON PHI
PHI=.33
A=.1
B=.275
EPS=1.E-6
CALL ROMBERG (SUM, A, B, FUN, EPS)
.
.
.
END
```

The function subprogram would be:

```
FUNCTION FUN(T)
COMMON PHI
FUN = COS(T * T + PHI)
RETURN
END
```

RESTRICTIONS: A function subprogram with a single argument must be written by the user to evaluate the integrand FUN(X). Since ROMBERG requires that its integrand be a function of one argument only, any variable parameters of the integrand must be passed to the function subprogram through COMMON. The name of the function must appear in an EXTERNAL statement in the calling program. (See example under USE.)

METHOD: This subroutine is taken with minor modifications from p. 199 of the reference. The method is described on pp. 166 to 170 of the reference. ROMBERG integration is a so-called automatic method in that the routine normally returns a value of the integral within the prescribed accuracy (see ACCURACY, below).

ACCURACY: Normally, iteration proceeds until

$$\left|SUM_i - SUM_{i-1}\right| < EPS * \left|B-A\right| * \max_{X \epsilon [A,B]} \left|FUN(X)\right|$$

The subroutine is dimensioned such that if the accuracy criterion is not satisfied after 19 steps (262 144 integrand evaluations) the best estimate of the integral at that point is returned and no error message is given.

REFERENCE: Davis, Philip J.; and Rabinowitz, Philip: Numerical Integration. Blaisdell Pub. Co., c.1967.

STORAGE: 254_8 locations

Subroutine REIG

LANGUAGE: FORTRAN

PURPOSE: To find the eigenvalues of a real matrix.

USE: CALL REIG (A, N, NVAL, NVEC, RTR, RTI, VEC, NMAX, INDEX, IRUN, P, NPLUS, SAVE)

A	A two-dimensional array containing the input matrix.
N	The order of A; $1 \leqq N \leqq NMAX$.
NVAL	The number of eigenvalues desired.
NVEC	Dummy parameter; not used.
RTR	A one-dimensional array in which the real parts of the eigenvalues are stored. The real eigenvalues are stored first, and are sorted by magnitude in decreasing order.
RTI	A one-dimensional array in which the imaginary parts of the eigenvalues are stored.
VEC	Dummy parameter; not used.
NMAX	The maximum order of A as stated in the dimension statement of the calling program.
INDEX	A one-dimensional array of temporary storage.
IRUN	A one-dimensional array of temporary storage.
P	A one-dimensional array of temporary storage.
NPLUS	The order of A plus one; i.e., NPLUS = N + 1.
SAVE	A two-dimensional array of temporary storage.

RESTRICTIONS: The following arrays must be dimensioned in the calling program as indicated: A (NMAX, NMAX), RTR (NMAX), RTI (NMAX), INDEX (NMAX), IRUN (NMAX), P (NMAX), SAVE (NMAX, NMAX+1). N is limited to 100. The input matrix is not destroyed.

METHOD: The original matrix is transformed to upper Hessenberg form. Then the eigenvalues are found using the QR transform of J. G. F. Francis (ref. (a)).

ACCURACY: Accuracy depends on the conditioning of A (ref. (c)).

REFERENCES:

(a) Francis, J. G. F.: The QR Transformation – A Unitary Analogue to the LR Transformation Comput. J., vol. 4.
Pt. 1 – Oct. 1961, pp. 265-271.
Pt. 2 – Jan. 1962, pp. 332-345.

(b) Wilkinson, J. H.: Stability of the Reduction of a Matrix to Almost Triangular and Triangular Forms by Elementary Similarity Transformations. J. Assoc. Comput. Mach., vol. 6, 1959, pp. 336-359.

(c) Marcus, Marvin: Basic Theorems in Matrix Theory. Nat. Bur. Stand. Appl. Math. Ser. 57, U.S. Dep. Com., Jan. 22, 1960. (Reprinted 1964.)

STORAGE: REIG 2167_8 locations including QRT and HESSEN.

SUBPROGRAMS USED: The following subprograms are used by REIG:

QRT	540_8 locations
HESSEN	431_8 locations

Subroutine MATRIX (Modified)

LANGUAGE: FORTRAN AND COMPASS

PURPOSE: Shortened version of CDC subroutine MATRIX (CDC publication No. 60135200) for a comprehensive group of matrix operations. This version deletes the real symmetric eigenvalue/eigenvector options to conserve storage. In addition, Turing's condition number is calculated for option 10.

USE: CALL MATRIX (I, M, N, K, A, KA, B, KB, C, KC)

I – option code	Matrix operation
0	Transpose A into B ($B \neq A$)
1	Move A into B
2	Symmetric product $A^T * A = B$, B packed
3	Deleted
4	Pack symmetric matrix A into B
5	Unpack symmetric matrix A into B
10	Solve D X = E with D^{-1} returned, A upon call = $[D \mid E]$ A upon return = $[D^{-1} \mid X]$
11	Same as 10 except calculation of D^{-1} deleted
20	Multiply $A*B = C$; $C = A$ or $= B$ is permissible.
21	Add $A + B = C$
22	Subtract $A - B = C$
23	Transpose multiply $A^T * B = C$
24	Scalar multiply $A * B = C$; A = Scalar

M — Number of rows of the matrix A.

N — Number of columns of the matrix A.

K — Unused, = 0, except for I = 10, 11, 20, and 23.
For I = 10 and 11, K = pivoting parameter
K = 0, full search each pass
K = 1, search Jth row on Jth pass
K = 2, no pivoting, use diagonal elements
For I = 20 and 23, K = number of columns of matrices B and C.

A — Matrix A.

KA	Column size of matrix A.
B	Matrix B, or the determinant of D for I = 10.
KB	Column size of matrix B.
C	Matrix C, or Turing's condition number for I = 10.
KC	Column size of matrix C.

RESTRICTIONS: For description of restrictions, see reference. However, it may be noted that maximum number of rows or columns is 300. Maximum dimensions in calling program must be consistent with operations. For options 0 and 1, $KA \geqq M$, $KB \geqq N$; for options 10 and 11, $N \geqq M$, $KA \geqq M$; for options 21, 22, and 23, $KA \geqq M$, $KB \geqq M$; for options 21 and 22, $KC \geqq M$; for option 23, $KA \geqq M$. Note that for options 10 and 11, matrix A must contain D and E of $DX = E$ in adjacent columns. Option 3 has been deleted for this version.

METHOD: The CDC subroutine MATRIX is a FORTRAN subroutine that computes the eigenvalues and eigenvectors of a real symmetric matrix for option 3. For all other options, the subprogram only sets up the call to a COMPASS subroutine, MASCNT, to perform the matrix operations. In this version the lengthy FORTRAN eigenvalue/eigenvector section has been deleted to conserve storage, and the calculation of Turing's condition number has been added for the matrix-inversion option. See reference for further discussion of methods.

ACCURACY: Applicability depends on matrix operation. For matrix inversion and simultaneous linear equations, the loss of significant figures can be estimated from $\log_{10}(CNO)$ where CNO is Turing's condition number defined as:

$$CNO = \|A\| * \|A^{-1}\| / N$$

REFERENCE: Anon.: Control Data 6000 Series Computer Systems Matrix Algebra Subroutines Reference Manual. Publ. No. 60135200, Control Data Corp., June 1966.

STORAGE:

MATRIX (Mod.)	132_8
MASCNT	431_8
LABELED COMMON	1130_8 - /IROW/IROW(300) /ICOL/ICOL(300)

Subroutine CXINV

LANGUAGE: FORTRAN

PURPOSE: To solve the complex matrix equation $AX = B$ where A is a square complex coefficient matrix and B is a complex matrix of constant vectors. The solution to a set of simultaneous equations, the matrix inverse, and the determinant may be obtained.

USE: CALL CXINV (A, N, B, M, DETERM, IPIVOT, INDEX, MAX, ISCALE)

A — A two-dimensional complex array of the coefficients. On return to the calling program, A contains the matrix inverse.

N — The order of A; $1 \leq N \leq MAX$

B — A two-dimensional complex array of the constant vectors B. On return to the calling program, B contains the X values.

M — The number of column vectors in B. If $M = 0$, there is no solution of the simultaneous equations; however, there must be an entry for B in the call statement.

DETERM — Gives the complex value of the determinant by the formula:

$$DET(A) = \left(10^{100}\right)^{ISCALE} (DETERM)$$

IPIVOT — A one-dimensional integer array of temporary storage.

INDEX — A two-dimensional integer array of temporary storage.

MAX — The maximum order of A as stated in the dimension statement of the calling program.

ISCALE — A scale factor computed by the subroutine to keep the results of computation within the floating point word size of the computer.

RESTRICTIONS: The calling program must dimension arrays as indicated:
A (MAX, MAX), B (MAX, M), IPIVOT (MAX), INDEX (MAX, 2).
It must also type A, B, DETERM as COMPLEX.

The input matrices A and B are destroyed. On return to the calling program the inverse of A is in A and X is in B.

METHOD: Jordan's method is used to reduce the matrix A to the identity matrix I through a succession of elementary transformations: $l_n, l_{n-1}, \ldots, l_1 * A = I$. If these transformations are simultaneously applied to I and to the matrix B of constant vectors, the results are A^{-1} and X where $AX = B$.

Each transformation is selected so that the largest element is used in the pivotal position.

ACCURACY: Total pivotal strategy is used to minimize the rounding errors; however, the accuracy of the final results depends upon how well-conditioned the original matrix is.

REFERENCE: Fox, L.: An Introduction to Numerical Linear Algebra. Oxford Univ. Press, 1965.

STORAGE: 721_8 locations

SUBPROGRAMS USED: Library: CABS

Subroutine RECIN

LANGUAGE: COMPASS

PURPOSE: To read binary records written by the subroutine RECOUT.

USE:

1. Type 1 – Individual elements (not arrays)
   CALL RECIN (LUN; IT; ICOUNT; L1, L2, . . ., LN) where

   LUN = logical unit number

   IT = type = 1

   ICOUNT = location reserved by the user. RECIN will store the following information in this location: 0 = end-of-file; nonzero = number of words actually in the logical record. If the end-of-file flag was written by a call to RECOUT with IEOF = 1, then end-of-file testing must be done by testing ICOUNT for 0. If the end-of-file was written by an END FILE statement, then testing for end-of-file must be done by the IF (EOF, LUN) statement.

   L1, L2, . . ., LN = individual list elements.

2. Type 2 – Arrays
   CALL RECIN (LUN, IT, ICOUNT, ARRAY, IFIRST, ILAST, INC) where

   LUN = logical unit number

   IT = type = 2

   ICOUNT = 0 = end-of-file; nonzero = number of words actually in the logical record (See ICOUNT under type 1)

   ARRAY = array name

   IFIRST = first subscript

   ILAST = last subscript

   INC = increment

EXAMPLES:

1. CALL RECIN (1, 1, K, A, B, ARRAY(1), ARRAY (2))
   Read a record from logical unit 1 into A, B, ARRAY(1) and Array (2). Note that if the record contained only 3 words, K would equal 3 and ARRAY(2) would be unaltered.

2. CALL RECIN (1, 2, K, ARRAY, 1, 39, 2)
   Read 20 words from logical unit 1 into
   ARRAY(1), ARRAY(3), . . ., ARRAY(39).

RESTRICTIONS: If RECIN is used on a file, the only other FORTRAN statements which may be used on that file are REWIND and IF (EOF, i).

The buffer size must be at least 2001_8.

RECIN must be used to read files written by RECOUT and only by RECOUT.

METHOD: RECIN reads into a central memory buffer physical records written by RECOUT, then passes to the user the requested logical record via a list giving the elements of the desired logical record. RECIN is analogous to a FORTRAN binary READ statement.

ACCURACY: Not applicable.

REFERENCE: None.

STORAGE: 315_8 locations

OTHER CODING INFORMATION: Day file diagnostics and their meaning:

(1) UNASSIGNED FILE MEDIUM FILE TAPEnn – No FET exists for this file. Every file has a file environment table that contains information describing the file to the system. This error would probably be caused by the file not being defined in the PROGRAM card or the user accidentally over-writing portions of his program.

(2) BAD TYPE – The IT parameter was not 1 or 2.

(3) UNCHECKED END FILE – The program attempted to read past EOF without testing for EOF.

(4) READ/WRITE SEQUENCE ERROR – An attempt was made to read after writing.

Subroutine RECOUT

LANGUAGE: COMPASS

PURPOSE: To write short binary records on a disk or tape in an optimum manner to increase peripheral processor and central processor efficiency. These records are to be read by RECIN.

USE: RECOUT may be used for either tape or disk files.

1. Type 1 – Individual elements (not arrays)
   CALL RECOUT (LUN; IT; IEOF; L1, L2, . . ., LN) where

   LUN = logical unit number

   IT = type = 1

   IEOF = 1 if an end-of-file flag is desired, otherwise it must be zero. There are two methods by which the user may end his file. One method is to call RECOUT with IEOF = 1 when the last data record is written. This will cause an end-of-file flag (a short length record of less than 512_{10}CM words) to be written. RECIN is programed to sense this and will set ICOUNT = 0 when sensed. If this method is used, the user must set IEOF = 1 when outputting his last data record since RECOUT should not be called with an empty list. For all other calls to RECOUT, IEOF must be set = 0. The other method of ending the file is to use the END FILE statement. This is the most convenient way of ending the file.

   L1, L2, . . ., LN = individual list elements.

2. Type 2 – Arrays
   CALL RECOUT (LUN, IT, IEOF, ARRAY, IFIRST, ILAST, INC) where

   LUN = logical unit number

   IT = type = 2

   IEOF = 1 if an end-of-file desired
   = 0 no end-of-file
   See explanation under type 1.

   ARRAY = array name

IFIRST = first subscript

ILAST = last subscript

INC = increment

EXAMPLES:

(1) CALL RECOUT (1, 1, 0, A, B, ARRAY(1), ARRAY(2))
Write a record on logical unit 1 containing A, B, ARRAY(1), ARRAY(2).

(2) CALL RECOUT (1, 2, 0, ARRAY, 1, 20, 1)
Write a record containing ARRAY(1) through ARRAY(20). This is equivalent to WRITE(1) (ARRAY(I), I = 1, 20).

RESTRICTIONS:

If RECOUT is used on a file, the only other FORTRAN statements which may be used on that file are REWIND and END FILE.

The buffer size must be at least 2001_8. A normal FORTRAN buffer is this size.

FILES written with RECOUT must be read with RECIN.

If the list to be written in a logical record is larger than 511_{10}CM words, then RECOUT offers no advantage and should not be used.

If the programer wishes to write a file containing multifiles using RECOUT, then he must end each file by setting IEOF = 1 and not by using the END FILE statement. Consequently, he should then test for end-of-file in RECIN by testing ICOUNT for zero.

METHOD:

Under the CDC SCOPE 3.0 operating system each binary write commanded by the FORTRAN statement WRITE (LUN) . . . causes one or more physical records to be output to either a disk or tape file. If the logical record size written by the programer is small and the number of records processed is large, then excessive usage of I/0 routines and equipment results. To decrease this I/0 time, RECOUT blocks binary data into an optimum record size (512_{10}CM words) in a central memory buffer before transmitting it to the actual disk or tape file.

ACCURACY:

Not applicable.

REFERENCE:

None.

STORAGE: 340_8 locations

OTHER CODING INFORMATION: Day file diagnostics and their meaning:

(1) UNASSIGNED FILE MEDIUM FILE TAPEnn – No FET exists for the file. Every file has a file environment table that contains information describing the file to the system. This error would probably be caused by the file not being defined in the PROGRAM card or the user accidentally overwriting portions of his program.

(2) BAD TYPE – The IT parameter was not 1 or 2.

(3) BUFFER TOO SMALL – The buffer size was less than 2001_8.

(4) BAD PARAM COUNT – The number of parameters in the call was illegal.

(5) WRITE/READ SEQUENCE ERROR – A write request was made after a read request.

Subroutine DAYTIM

LANGUAGE: COMPASS

PURPOSE: The purpose of this subroutine is to provide the current date and time of day to a central memory program.

USE: The subroutine may be called from a FORTRAN program using the following sequence:

CALL DAYTIM (RESULT, JUDATE)

RESULT = A single subscripted array dimensioned by two (2). The current date will be returned in RESULT(1) and the current time of day will be returned in RESULT(2).

JUDATE = An optional parameter. JUDATE need not be supplied. If it is, the Julian date will be returned in the memory cell named JUDATE. If JUDATE is not specified, then the Julian date will not be returned.

The date, the time, and the Julian date will be in display code in the following formats:

```
DATE:      b    MM/DD/YY
                        Year
                        Day
                        Month
                        Blank

TIME:      b    HH.MM.SS
                        Seconds
                        Minutes
                        Hours
                        Blank

JULIAN    bbb   YY  DDD  bb
  DATE:
                        Blanks
                        Day
                        Year
                        Blanks
```

Each value may be printed using an A10 format in a FORTRAN program.

RESTRICTIONS: Whenever the 6000 system is deadstarted, the operators key in the date and the current time of day. It is the date and the current time of day based on the initial date and time of day keyed in by the operator that are returned to the calling program.

METHOD: The subroutine uses the macros supplied by CDC with Scope 3.0 to obtain the current date, time of day, and Julian date.

STORAGE: 31_8

Subroutine MASCNT (Simulator)

LANGUAGE: FORTRAN

PURPOSE: Simulate the COMPASS language subroutine MASCNT with a FORTRAN subroutine for use with options 10 and 11 of FORTRAN subroutine MATRIX (options 10 and 11 are for matrix inverse and solutions of linear equations, DX = E).

USE: CALL MASCNT (KR, A, B, C)

KR An integer array that contains seven of the formal parameters of subroutine MATRIX. The usage for this version of MASCNT is:

	MATRIX parameter	Definition
KR(1)	I	Unused option parameter; I of 10 of MATRIX is assumed
KR(2)	M	Number of rows of A
KR(3)	N	Number of columns of A
KR(4)	K	Unused pivoting parameter; full pivoting used by MATINV
KR(5)	KA	Maximum number of rows of A as dimensioned in calling program
KR(6)	KB	Unused
KR(7)	KC	Unused

A A single two-dimensional array containing the two-dimensional matrix of coefficients and the two-dimensional matrix of column vectors. For DX = E, the first M columns of A must contain D and the adjacent (M + 1) to N columns must contain E. Upon return to the calling program, the locations of E contain the solution vectors. D is destroyed and the inverse of D is returned in D.

B The determinant of D.

C Unused.

RESTRICTIONS: The array A must be dimensioned in the calling program as: A (KA, MAX), MAX ≧ N. The original matrix D is destroyed. Although this version of MASCNT can be used with MATRIX option 11, the inverse of D is always obtained. N ≦ 150.

METHOD: Subprogram sets up a call to FORTRAN subroutine MATINV to perform the matrix operations (see description of MATINV). Common blocks IROW and ICOL are used for the temporary storage required by MATINV.

ACCURACY: See description of subroutine MATINV.

REFERENCE: Anon.: Control Data 6000 Series Computer Systems Matrix Algebra Subroutines Reference Manual. Publ. No. 60135200, Control Data Corp., June 1966.

STORAGE:

MASCNT (Sim.)	66_8	
MATINV	542_8	
LABELED COMMON	1130_8	- /IROW/IROW(300) /ICOL/ICOL(300)

SUBPROGRAM USED: MATINV

```
      SUBROUTINE MASCNT(KR,A,B,C)
C
C  FORTRAN SIMULATOR FOR CDC MATRIX PACKAGE COMPASS LANGUAGE SUBROUTINE
C  MATRIX OPTIONS 10 AND 11 ONLY
C
      COMMON/IROW/IROW(300)/ICOL/ICOL(300)
      DIMENSION KR(1),A(1)
      NS=1+KR(2)*KR(5)
      M=KR(3)-KR(2)
      CALL MATINV(A,KR(2),A(NS),M,B,IROW,ICOL,KR(5),IS)
      B=10**(100*IS)*B
      RETURN
      END SUBROUTINE MASCNT
```

Subroutine MATINV

LANGUAGE: FORTRAN

PURPOSE: MATINV solves the matrix equation $AX = B$ where A is a square coefficient matrix and B is a matrix of constant vectors. The solution to a set of simultaneous equations, the matrix inverse, and the determinant may be obtained. If the user does not want the inverse, use SIMEQ for savings in time and storage. For the determinant only, use DETEV.

USE: CALL MATINV (A, N, B, M, DETERM, IPIVOT, INDEX, NMAX, ISCALE)

A	A two-dimensional array of the coefficients. On return to the calling program, A^{-1} is stored in A.
N	The order of A; $1 \leqq N \leqq NMAX$
B	A two-dimensional array of the constant vectors B. On return to calling program, X is stored in B.
M	The number of column vectors in B. $M = 0$ signals that the subroutine is used solely for inversion; however, in the call statement an entry corresponding to B must still be present.
DETERM	Gives the value of the determinant by the following formula: $$DET(A) = \left(10^{100}\right)^{ISCALE} (DETERM)$$
IPIVOT	A one-dimensional array of temporary storage used by the routine.
INDEX	A two-dimensional array of temporary storage used by the routine.
NMAX	The maximum order of A as stated in the dimension statement of the calling program.
ISCALE	A scale factor computed by the subroutine to keep the results of computation within the floating point word size of the computer.

RESTRICTIONS: Arrays A, B, IPIVOT, and INDEX are dimensioned with variable dimensions in the subroutine. The maximum size of these arrays must be specified in a dimension statement of the calling program as: A (NMAX, NMAX), B (NMAX, M), IPIVOT (NMAX), INDEX (NMAX, 2). The original matrices, A and B, are destroyed. They must be saved by the user if there is further need for them.

The determinant is set to zero for a singular matrix.

METHOD: Jordan's method is used to reduce a matrix A to the identity matrix I through a succession of elementary transformations: $l_n, l_{n-1}, \ldots, l_1 * A = I$. If these transformations are simultaneously applied to I and to a matrix B of constant vectors, the results are A^{-1} and X where $AX = B$. Each transformation is selected so that the largest element is used in the pivotal position.

ACCURACY: Total pivotal strategy is used to minimize the rounding errors; however, the accuracy of the final results depends upon how well-conditioned the original matrix is.

REFERENCE: Fox, L.: An Introduction to Numerical Linear Algebra. Oxford Univ. Press, 1965.

STORAGE: 542_8 locations.

```
      SUBROUTINE MATINV(A,N,B,M,DETERM,IPIVOT,INDEX,NMAX,ISCALE)        MATI0010
C                                                                       MATI0020
C     MATRIX INVERSION WITH ACCOMPANYING SOLUTION OF LINEAR EQUATIONS   MATI0030
C                                                                       MATI0040
      DIMENSION IPIVOT(N),A(NMAX,N),B(NMAX,M),INDEX(NMAX,2)             MATI0050
      EQUIVALENCE (IROW,JROW), (ICOLUM,JCOLUM), (AMAX, T, SWAP)         MATI0060
C                                                                       MATI0070
C     INITIALIZATION                                                    MATI0080
C                                                                       MATI0090
    5 ISCALE=0                                                          MATI0100
    6 R1=10.0**100                                                      MATI0110
    7 R2=1.0/R1                                                         MATI0120
   10 DETERM=1.0                                                        MATI0130
   15 DO 20 J=1,N                                                       MATI0140
   20 IPIVOT(J)=0                                                       MATI0150
   30 DO 550 I=1,N                                                      MATI0160
C                                                                       MATI0170
C     SEARCH FOR PIVOT ELEMENT                                          MATI0180
C                                                                       MATI0190
   40 AMAX=0.0                                                          MATI0200
   45 DO 105 J=1,N                                                      MATI0210
   50 IF (IPIVOT(J)-1) 60, 105, 60                                      MATI0220
   60 DO 100 K=1,N                                                      MATI0230
   70 IF (IPIVOT(K)-1) 80, 100, 740                                     MATI0240
   80 IF (ABS(AMAX)-ABS(A(J,K)))85,100,100                              MATI0250
   85 IROW=J                                                            MATI0260
   90 ICOLUM=K                                                          MATI0270
   95 AMAX=A(J,K)                                                       MATI0280
  100 CONTINUE                                                          MATI0290
  105 CONTINUE                                                          MATI0300
      IF (AMAX) 110,106,110                                             MATI0310
  106 DETERM=0.0                                                        MATI0320
      ISCALE=0                                                          MATI0330
      GO TO 740                                                         MATI0340
  110 IPIVOT(ICOLUM)=IPIVOT(ICOLUM)+1                                   MATI0350
C                                                                       MATI0360
C     INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL                 MATI0370
C                                                                       MATI0380
  130 IF (IROW-ICOLUM) 140, 260, 140                                    MATI0390
  140 DETERM=-DETERM                                                    MATI0400
  150 DO 200 L=1,N                                                      MATI0410
  160 SWAP=A(IROW,L)                                                    MATI0420
  170 A(IROW,L)=A(ICOLUM,L)                                             MATI0430
  200 A(ICOLUM,L)=SWAP                                                  MATI0440
  205 IF(M) 260, 260, 210                                               MATI0450
  210 DO 250 L=1, M                                                     MATI0460
  220 SWAP=B(IROW,L)                                                    MATI0470
  230 B(IROW,L)=B(ICOLUM,L)                                             MATI0480
  250 B(ICOLUM,L)=SWAP                                                  MATI0490
  260 INDEX(I,1)=IROW                                                   MATI0500
  270 INDEX(I,2)=ICOLUM                                                 MATI0510
  310 PIVOT=A(ICOLUM,ICOLUM)                                            MATI0520
C                                                                       MATI0530
C     SCALE THE DETERMINANT                                             MATI0540
C                                                                       MATI0550
 1000 PIVOTI=PIVOT                                                      MATI0560
 1005 IF(ABS(DETERM)-R1)1030,1010,1010                                  MATI0570
 1010 DETERM=DETERM/R1                                                  MATI0580
      ISCALE=ISCALE+1                                                   MATI0590
      IF(ABS(DETERM)-R1)1060,1020,1020                                  MATI0600
 1020 DETERM=DETERM/R1                                                  MATI0610
```

```
      ISCALE=ISCALE+1
      GO TO 1060
 1030 IF(ABS(DETERM)-R2)1040,1040,1060
 1040 DETERM=DETERM*R1
      ISCALE=ISCALE-1
      IF(ABS(DETERM)-R2)1050,1050,1060
 1050 DETERM=DETERM*R1
      ISCALE=ISCALE-1
 1060 IF(ABS(PIVOTI)-R1)1090,1070,1070
 1070 PIVCTI=PIVOTI/R1
      ISCALE=ISCALE+1
      IF(ABS(PIVOTI)-R1)320,1080,1080
 1080 PIVCTI=PIVOTI/R1
      ISCALE=ISCALE+1
      GO TC 320
 1090 IF(ABS(PIVOTI)-R2)2000,2000,320
 2000 PIVCTI=PIVOTI*R1
      ISCALE=ISCALE-1
      IF(ABS(PIVOTI)-R2)2010,2010,320
 2010 PIVCTI=PIVOTI*R1
      ISCALE=ISCALE-1
  320 DETERM=DETERM*PIVOTI
C
C     DIVIDE PIVOT ROW BY PIVOT ELEMENT
C
  330 A(ICCLUM,ICOLUM)=1.0
  340 DO 350 L=1,N
  350 A(ICCLUM,L)=A(ICOLUM,L)/PIVOT
  355 IF(M) 380, 380, 360
  360 DO 370 L=1,M
  370 B(ICOLUM,L)=B(ICOLUM,L)/PIVCT
C
C     REDUCE NON-PIVOT ROWS
C
  380 DO 550 L1=1,N
  390 IF(L1-ICCLUM) 400, 550, 400
  400 T=A(L1,ICOLUM)
  420 A(L1,ICULUM)=0.0
  430 DO 450 L=1,N
  450 A(L1,L)=A(L1,L)-A(ICOLUM,L)*T
  455 IF(M) 550, 550, 460
  460 DO 500 L=1,M
  500 B(L1,L)=B(L1,L)-B(ICOLUM,L)*T
  550 CONTINUE
C
C     INTERCHANGE COLUMNS
C
  600 DO 710 I=1,N
  610 L=N+1-I
  620 IF (INDEX(L,1)-INDEX(L,2)) 630, 710, 630
  630 JROW=INDEX(L,1)
  640 JCOLUM=INDEX(L,2)
  650 DO 705 K=1,N
  660 SWAP=A(K,JROW)
  670 A(K,JROW)=A(K,JCOLUM)
  700 A(K,JCCLUM)=SWAP
  705 CONTINUE
  710 CONTINUE
  740 RETURN
      END
```

Plotting Subroutines

Usage descriptions are given for the plotting subroutines called by the plotting programs of this report. These subroutines are LRC computer system library subroutines and listings are not given. They are described in the following order:

	Page
CALCOMP	133
AXES	134
CALPLT	136
DASHLN	138
LEROY/BALLPT	139
LINE	140
NOTATE	141
PNTPLT	143

Subroutine CALCOMP

LANGUAGE:	FORTRAN
PURPOSE:	This is the normal mode processor. The necessary parameters and linkage are set up to output a tape for the CalComp Model 780/763 Electro-Mechanical Plotter.
USE:	CALL CALCOMP
RESTRICTIONS:	This call must be given before the first call to a plotting routine.
STORAGE:	CALCOMP 5467_8 total for all subprograms used.
SUBPROGRAMS USED:	PLOTSW, PLT763, BLCR, STRC, TAPWRI, ENCOD1, STRCALL, CREATEF, BOUNDCK, TRUNCL, and LOCATE
OTHER CODING INFORMATION:	The following is a list of messages, the circumstances under which they will appear, and the action taken:

NO PLOTTING DEVICE SPECIFIED	This message is printed in the output file and the job is ended in subroutine PLOTSW. This condition occurs if there is no initialization CALL CALCOMP in the program prior to using subroutines which generate plotting output.
PLOTTING COMMENCED	This message is printed in the dayfile when the first pen movement is encountered as a result of a program call to the plotting subroutines.
THE LAST CALCOMP BLOCK ADDRESS WAS xxx DATA PLOTTED = yyyyyy	These messages are printed in the dayfile when the plotting is completed. xxx is the value of the last block address on the CalComp plotter tape. This block address is at the end of the last valid data. yyyyyy is the approximate number of data points. Plotting is completed either as a result of a CALL CALPLT(x,y,999) or when the job is ended due to an error recognized by the CalComp subroutines.

Subroutine AXES

LANGUAGE: FORTRAN

PURPOSE: To draw a line with tick marks at specified intervals, annotate the value of the variable at tick marks, and provide an axis identification label.

USE: CALL AXES (X, Y, THETA, DIST, ORIGIN, DV, TMAJ, TMIN, BCD, HGT, N) where

X,Y are the coordinates in floating point inches of the starting point of the axis with reference to the plotting area origin as established by CALPLT.

THETA is the angle of rotation measured counterclockwise from the X-axis in floating point degrees. NOTE: Normally, THETA is 0° for an X-axis and 90° for a Y-axis.

DIST is the length of the axis in floating point inches. Should be a multiple of TMAJ.

ORIGIN is the functional value to be assigned to the origin (i.e., the value of the first scale), in floating point.

DV is the adjusted scale factor for the array to be plotted (change in value per in.). NOTE: Values of ORIGIN and DV which will produce a reasonable scale may be calculated using the subroutine ASCALE.

TMAJ is the distance in floating point inches for major tick marks (0.25 in. high). Numbers are placed on the axis at the major tick marks in accordance with the values of ORIGIN and DV. The numbers written along the axis are adjusted to be between 1000.00 and 0.01 in magnitude. Immediately after the last number on the axis is placed the caption, $\times 10^{exp}$, where exp is the required exponent.
If the values are integer multiples, the decimal point and decimal places are eliminated. A negative TMAJ will cause the actual value to be written instead of the adjusted value.

TMIN is the divisions per inch in floating point for minor tick marks (0.125 in. high). To eliminate minor tick marks the following may be used:

TMIN = 0.

BCD is the character label for the axis (see NOTATE routine).

HGT is the height of the full-size characters in the BCD title. Numbers at the tick marks will be (0.75 *HGT) high. HGT is in floating point inches.

If HGT = 0., all annotation will be eliminated.

N is an integer specifying the number of characters in BCD title. A negative N places the annotation on the clockwise side of the axis and a positive N places the annotation on the counterclockwise side of the axis. N = 0 is not allowed. If it is desired to have no label, then the BCD parameter should be 1HΔ and N = +1 or -1.

RESTRICTIONS: Only perpendicular axes are recommended.

STORAGE: 1016_8

SUBPROGRAMS USED: CALPLT, NOTATE, NUMBER, ROUND, SIN, COS

Subroutine CALPLT

LANGUAGE: FORTRAN

PURPOSE: To move the plotter pen to a new location with pen up or down and to signal the end of a job segment by incrementing the block address number.

USE: CALL CALPLT (X, Y, IPEN)

X,Y	The floating-point values for pen movement
IPEN = 2	Pen down
IPEN = 3	Pen up

Negative IPEN will assign (X = 0, Y = 0) as the location of the pen after moving the X,Y (creating a new reference point) and will increase the block number by one. (The block number is the number that appears in the display at the top of the tape drive on the plotter and identifies the portion of the output tape that is being plotted. The block address 001 is written automatically as a result of the initialization processor call.) Each block address generally implies a separate page or plot.

IPEN = 999 Writes a terminating block address of 999 for peripheral handling of the plotter tape and all further processing is skipped. X and Y may be any values since they are ignored.

RESTRICTIONS: All X- and Y-coordinates must be expressed as floating-point values in inches (actual page dimensions) in deflection from the origin.

(A CALL TO CALPLT WITH EITHER NEGATIVE IPEN (USUALLY -3) OR A TERMINATING BLOCK ADDRESS (IPEN = 999) MUST BE GIVEN AS THE LAST PLOTTING INSTRUCTION BEFORE ENDING A PROGRAM WHICH USES ANY OF THE PLOTTER SUBROUTINES: THIS IS TO BE SURE THAT ALL PLOTTER INSTRUCTIONS ARE WRITTEN ON THE PLOTTER TAPE.)

METHOD: The main subroutine in the CalComp software package is the CALPLT subroutine. All other special-purpose subroutines eventually call CALPLT either directly or indirectly. Subroutine CALPLT moves the pen in a straight line between the present pen position and another pen location to which the programer wishes the pen to be moved.

In order to cause such instructions to be written, the programer specifies the coordinates of the point to which the pen is to be moved and whether the pen is to be moved in a raised or lowered position. This movement is accomplished by the FORTRAN instruction

CALL CALPLT (X, Y, IPEN)

Also, the subroutine provides "sequence numbers" on the tape, making it possible to afford identification of job segments. The block address 001 is written on the first call to CALPLT. Thereafter, if the programer defines a new origin or wishes to divide the job into several segments, he need only set the argument IPEN negative. The CALPLT routine then moves the pen to (X,Y); stores this location as (0,0), that is, a new origin; and increases the block address by one.

STORAGE: CALPLT 251_8.

SUBPROGRAMS USED: PLOTSW, STRCALL, and LOCATE

Subroutine DASHLN

LANGUAGE: FORTRAN

PURPOSE: To draw a dashed line between two points.

USE: CALL DASHLN (X0, Y0, X1, Y1, D) where

X0,Y0, and X1,Y1	are coordinates in floating point inches of the end points of a line.
D	is the length in floating point inches of each dash.

RESTRICTIONS:

METHOD: No matter what the slope of the line, the dash length will remain the requested length. The first dash of a line segment is set at one-half the requested dash length so that whenever line segments are connected, the dash at the meeting of the line segments will not be twice as large as the requested dash length. The last dash of a line segment will derive its length from that portion at the end of a line segment which is less than the requested dash length. The subroutine will draw a dash the length of the line segment if a dash length is requested that is equal to or larger than the line segment. If the end points of the line segment are the same, the subroutine will return to the calling program.

STORAGE: 267_8

SUBPROGRAMS USED: CALPLT, SQRT

Subroutine LEROY/BALLPT

LANGUAGE: FORTRAN

PURPOSE: The parameters necessary to accommodate plotting with the liquid ink pen are set up by CALL LEROY. Once set, this mode will remain in effect as long as CALCOMP is in use or until a call to BALLPT is given.

The parameters for plotting with the ballpoint pen are reset by CALL BALLPT. This mode is automatically in effect with CALCOMP unless there has been a call to LEROY.

USE: CALL BALLPT
CALL LEROY

RESTRICTIONS: The CALL LEROY should be used only with CALCOMP. In addition to reducing the speed of the plotter for all plotting movements, the number of plot vectors in any annotation is considerably increased.

The CALL LEROY must be made prior to any plotting calls, but after the CALL CALCOMP.

STORAGE: LEROY/BALLPT 25_8

SUBPROGRAMS USED: None.

Subroutine LINE

LANGUAGE: FORTRAN

PURPOSE: To draw a continuous line through and/or draw a symbol at each successive data point (stored in an array).

USE: CALL LINE (XARRAY, YARRAY, N, K, J, L, S) where

XARRAY and YARRAY	are the names of arrays containing the X values and Y values, respectively, to be plotted. Values must be in floating point.
N	is the number of points to be plotted.
K	is the interleave factor of a mixed array (normally = 1).
J	is positive for line and symbol plot, negative for symbol only plot. The magnitude specifies the alternate number of data points at which to plot a symbol. = 0 for line plot. = 1 for symbol for every data point. = 2 for symbol for every other data point, etc.
L	is an integer describing symbol to be used, see NOTATE routine for list.
S	is the desired symbol height in floating point (see NOTATE routine).

RESTRICTIONS: LINE expects the adjusted minimums and scale factors as described in ASCALE since the routine automatically sets an origin and scales the data in the array.

STORAGE: 352_8

SUBPROGRAMS USED: CALPLT, NOTATE, WHERE

Subroutine NOTATE

LANGUAGE: FORTRAN

PURPOSE: To draw alphanumeric information for annotation and labeling.

USE: CALL NOTATE (X, Y, HEIGHT, BCD, THETA, N) where

X,Y are the floating point page coordinates of the first character. The coordinates of the lower left-hand corner of the characters are specified.

HEIGHT specifies character size and spacing in floating point inches for a full-size character. The smallest possible character is 0.07 inch high. The width of a character will be (4/7) *HEIGHT and the space between characters is (2/7) *HEIGHT.

The ith character is plotted at:

$$x_i = X + (i-1)(6/7)(\text{HEIGHT})(\text{Cos } \theta)$$

$$y_i = Y + (i-1)(6/7)(\text{HEIGHT})(\text{Sin } \theta)$$

$$1 \leqq i \leqq N$$

BCD is the string of characters to be drawn and is usually written in the form: nHXXXX--- (the same way an alpha message is written using FORTRAN format statements). Instead of specifying alpha information as above, one may give the beginning storage location of an array containing alphanumeric information.

THETA is the angle in floating point degrees at which the information is to be drawn. Zero degrees will print horizontally reading from left to right, 90° will print the line vertically reading from bottom to top, 180° will print the line horizontally reading from right to left (i.e., upside down), and 270° will print vertically reading from top to bottom.

N is the number of characters, including blanks, in the label.

METHOD: The character height is a variable entry parameter to the subroutine NOTATE. However, the width-to-height ratio is fixed at 4/7. This is because the characters are defined by a series of bi-octal offset pairs for a 4×7 matrix. The reference origin for the offset pairs which define each character is the lower left-hand corner of the matrix. The X and Y values which are entry parameters to NOTATE define the location of the lower left-hand corner of the first character to be plotted for this entry to NOTATE. Subsequent characters to be plotted are spaced from the previous character origin by 6/7 of the specified character height.

OTHER CODING INFORMATION: Only the alphanumeric option used in this report is described by this usage description.

STORAGE: 1252_8

SUBPROGRAMS USED: CALPLT, CNTRLN, DECODE, DECOD2, SIN, COS

Subroutine PNTPLT

LANGUAGE: FORTRAN

PURPOSE: To draw NASA standard plot symbols centered on a given coordinate value.

USE: CALL PNTPLT (A, B, NO, IS) where

A is the X coordinate for the centered symbol in floating point inches.

B is the Y coordinate for the centered symbol in floating point inches.

NO is an integer specifying the symbol to be used.
= 21 for a point •
= 22 for a plus sign +

IS is an integer value specifying the size symbol to be used.
= 1 small
= 2 medium
= 3 large
(See fig. A1.)

ACCURACY: A positive integer value for NO in the calling sequence will produce symbols of the same quality as in figure A1. A negative integer value will produce symbols of less quality but will result in a considerably faster computer run.

STORAGE: 506_8

SUBPROGRAMS USED: CALPLT, CIRCLE, CNTRLN

Figure A1.- NASA standard plot symbols.

REFERENCES

1. Bairstow, L.; Relf, E. F.; and Jones, R.: The Stability of Kite Balloons: Mathematical Investigation. R. & M. No. 208, Brit. A.R.C., 1915.

2. Neumark, S.: Equilibrium Configurations of Flying Cables of Captive Balloons, and Cable Derivatives for Stability Calculations. R. & M. No. 3333, Brit. A.R.C., 1963.

3. DeLaurier, James D.: A Stability Analysis of Cable-Body Systems Totally Immersed in a Fluid Stream. NASA CR-2021, 1972.

4. DeLaurier, James D.: A First Order Theory for Predicting the Stability of Cable Towed and Tethered Bodies Where the Cable Has a General Curvature and Tension Variation. VKI-TN-68, Von Karman Inst. Fluid Dyn., Dec. 1970.

5. DeLaurier, James D.: A Stability Analysis for Tethered Aerodynamically Shaped Balloons. J. Aircraft, vol. 9, no. 9, Sept. 1972, pp. 646-651.

6. Vorachek, Jerome J.; Burbick, James W.; and Doyle, George R., Jr.: Investigation of Dynamic Behavior of Tethered Balloon Systems. AFCRL-72-0113, U.S. Air Force, 1972.

7. Redd, L. Tracy: A Towing Technique for Determining the Aerodynamic Forces on Tethered Balloons. Paper presented at Sixth Air Force Cambridge Research Laboratories Scientific Balloon Symposium (Portsmouth, N.H.), June 1970.

8. Redd, L. Tracy; Bennett, Robert M.; and Bland, Samuel R.: Experimental and Analytical Determination of Stability Parameters for a Balloon Tethered in a Wind. NASA TN D-7222, 1973.

9. Redd, L. Tracy; Bland, Samuel R.; and Bennett, Robert M.: Stability Analysis and Trend Study of a Balloon Tethered in a Wind, With Experimental Comparisons. NASA TN D-7272, 1973.

10. Redd, L. Tracy; Bennett, Robert M.; and Bland, Samuel R.: Analytical and Experimental Investigation of the Stability of a Balloon Tethered in a Wind. Paper presented at Seventh Air Force Cambridge Laboratories Scientific Balloon Symposium (Portsmouth, N.H.), Sept. 1972.

11. Etkin, Bernard: Dynamics of Flight. John Wiley & Sons, Inc., c.1959.

☆U.S. GOVERNMENT PRINTING OFFICE: 1973-739-152/32

CAT. - 02