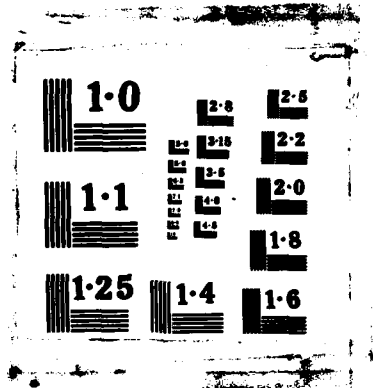




# ASSIF



UNCLASSIFIED

12

ARL/STREC-TR-414

AD-004-020



**DEPARTMENT OF DEFENCE  
DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION  
AERONAUTICAL RESEARCH LABORATORIES  
MELBOURNE, VICTORIA**

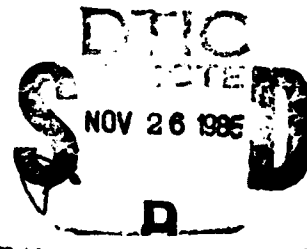
AD-A161 541

Structures Technical Memorandum 414

**INTERACTIVE FLUTTER SOLVER AT ARL**

by

S. GALEA



Approved for Public Release

THE UNITED STATES NATIONAL  
TECHNICAL INFORMATION SERVICE  
IS AUTHORIZED TO  
REPRODUCE AND SELL THIS REPORT

(C) COMMONWEALTH OF AUSTRALIA 1985

JUNE 1985

UNCLASSIFIED 11 21-85 034

DTIC FILE COPY

DEPARTMENT OF DEFENCE  
DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION  
AERONAUTICAL RESEARCH LABORATORIES

Structures Technical Memorandum 414

**INTERACTIVE FLUTTER SOLVER AT ARL**

by

S. GALEA

**SUMMARY**

An interactive direct flutter solving routine has been installed on ARL's ELXSI computer. Interactive graphics routines, using DI-3000 graphics, software, have been incorporated to give the user a progressive picture of the solution. Estimates of subcritical response data may also be determined using this method. A description of the method and the associated software is presented here. Also included is a two-dimensional flutter problem, using quasi-steady aerodynamics, which is solved by this direct flutter solver.



© COMMONWEALTH OF AUSTRALIA 1985

---

POSTAL ADDRESS: Director, Aeronautical Research Laboratories,  
P.O. Box 4331, Melbourne, Victoria, 3001, Australia.

CONTENTS

Page No.

1	INTRODUCTION	1
2.	THE FLUTTER EQUATION	2
3.	METHOD OF SOLUTION	2
3.1	Refining Initial Estimates	2
3.2	Scaling	4
3.3	Predicting Initial Estimates	4
4.	PROGRAM	4
5.	EXAMPLE	5
6.	DISCUSSION	7
7.	CONCLUSION	8

REFERENCES

APPENDIX A Flutter Solver Routines

APPENDIX B User-Supplied Routines

APPENDIX C Two Degree-of-Freedom Example

TABLES

FIGURES

DISTRIBUTION LIST

DOCUMENT CONTROL DATA

Accession For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification .....	
By .....	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	



## 1. INTRODUCTION

The classical methods for the solution of the flutter equations are the k method ("American Method") and the p-k method ("British Method"), see [1,2]. Both these methods treat the flutter equations, which are a set of n second-order linear homogeneous differential equations, as an eigenvalue problem and solve for the eigenvalue,  $\chi$ . Here the imaginary component of  $\chi$  gives the frequency of the response while the real component provides the decay rate. Flutter occurs when the oscillation is non-convergent, i.e., when the decay rate is zero.

The flight speed at which this occurs is called the flutter speed. The k method assumes that the aircraft's aeroelastic response is sinusoidal and hence the solution is correct only at the flight flutter speed. No such restriction is made in the p-k method. The latter method does assume, however, that the aerodynamic forces due to constant amplitude oscillating lifting surfaces are equal to aerodynamic forces generated by lifting surfaces with slowly increasing or decreasing amplitudes of oscillation. This assumption is accurate for low values of decay rate, either positive or negative. The p-k method is therefore more accurate in the subcritical and super critical regime than the k method.

In flutter problems the prime concern is to calculate the flutter speed, in which case the k and the p-k methods will provide accurate results. Another concern is the aircraft's subcritical flight characteristics, whereby if in flight or wind tunnel tests the flutter speed is not attained then these characteristics can be used for comparison with the appropriate mathematical model. In this case the p-k method will give more accurate results than the k method.

A new method of solving the flutter equation is described in References 3 and 4. The present paper describes a computer program based on this method which provides estimates of subcritical response data. Unlike classical methods, this direct flutter solver allows the user to establish quickly relationships between any two parameters which satisfy the flutter equations.

## 2. THE FLUTTER EQUATION

The aeroelastic motion of an aircraft can be modelled by the flutter equations which may be given in non-dimensional form as:

$$[A] \ddot{q} + [B + D] \dot{q} + [C + E] q = 0 \quad (1)$$

where A, D, and E are, respectively, the structural inertia, damping and stiffness matrices, B and C are, respectively, the aerodynamic damping and stiffness matrices, and q is the generalised co-ordinate vector. The aerodynamic matrices are functions of the non-dimensionalised frequency v, Mach number M, velocity U, air density  $\rho$ , and aircraft planform.

Here  $v = \frac{\omega l}{U}$

where  $\omega$  is the frequency of oscillation, l is the reference length and U the flight velocity. In this case  $\dot{q} = \partial q / \partial \tau$  where  $\tau$  is the non-dimensional time given by  $\tau = tU/l$ . On substituting  $q(\tau) = q_0 e^{\chi \tau}$  and dividing by  $e^{\chi \tau}$  equation 1 becomes -

$$\{ \chi^2 [A] + \chi [B + D] + [C + E] \} q_0 = 0 \quad (2)$$

Here  $\chi = \delta + iv$ ,  $\delta = v\Delta$  and  $\Delta$  is the decay rate of the response. Equation 2 may be rewritten as -

$$[S] q_0 = 0 \quad (3)$$

where S, the system matrix, is a function of U, M, v, air density ( $\rho$ ), stiffness, damping, mass distribution and so on.

## 3. METHOD OF SOLUTION

### 3.1 Refining Initial Estimates

Consider n real parameters  $x = x_1, x_2, \dots, x_n$  which specify the model; then eqn (3) can be expressed as:

$$[S(x)] q_0 = 0 \quad (4)$$

For a non-trivial solution the system matrix must be singular; thus the determinant, which is normally complex, must be zero.

Following the notation in Reference [4], if two of the system's parameters,  $x_j$  and  $x_k$ , are not fixed then the equation that must be satisfied is

$$| S(x) | = F(x_j, x_k) - \bar{F}(x_j, x_k) + i \hat{F}(x_j, x_k) = 0 \quad (5)$$

Gaussian elimination is used to evaluate the determinant of the system matrix  $S$ . For large degree of freedom systems the slowest step in evaluating  $F(x_j, x_k)$  is the evaluation of the system matrix  $S$ .

To minimize the number of function evaluations, the secant method is used to solve for the two variables,  $x_j$  and  $x_k$ , so as to satisfy equation (5). This is described more fully in Reference 4.

The secant method requires three points at any time to evaluate a refined estimate. For the first iteration, the three points are given by  $\bar{F}(x_j^0, x_k^0)$ ,  $\bar{F}(x_j^0 + \delta x_j, x_k^0)$  and  $\bar{F}(x_j^0, x_k^0 + \delta x_k)$  in the  $\bar{F}, x_j, x_k$  co-ordinate system and  $\hat{F}(x_j^0, x_k^0)$ ,  $\hat{F}(x_j^0 + \delta x_j, x_k^0)$  and  $\hat{F}(x_j^0, x_k^0 + \delta x_k)$  in the  $\hat{F}, x_j, x_k$  co-ordinate system. The initial estimate is denoted by the superscript zero. These points then determine a plane in each co-ordinate system. The refined estimate  $(x_j^1, x_k^1)$  is obtained by determining the point of intersection of the above two planes with the plane  $\bar{F} - \hat{F} = 0$ .

The solution is said to have converged at the  $n$ th iteration when -

$$\log_{10} \left[ \left( \left| \frac{x_j^n - x_j^{n-1}}{x_j^n} \right| + \left| \frac{x_k^n - x_k^{n-1}}{x_k^n} \right| \right) \times 100 \right] < 0.0002 \quad (6)$$

If convergence does not occur at the  $n$ th iteration then one of the points  $(x_j^{n-3}, x_k^{n-3})$ ,  $(x_j^{n-1}, x_k^{n-2})$  or  $(x_j^{n-1}, x_k^{n-1})$ , whichever is furthest from the refined estimate  $(x_j^n, x_k^n)$ , is discarded and so the two surviving points plus  $(x_j^n, x_k^n)$  are used to obtain a further refined estimate  $(x_j^{n+1}, x_k^{n+1})$ .

### 3.2 Scaling

In evaluating the determinant of  $S$ , computational problems may arise either when its diagonal elements are very large leading to overflow, or when they are very small so that significant figures may be lost. In order to overcome these problems each row of the determinant is divided by its diagonal element.

### 3.3 Predicting Initial Estimates

The idea of this direct method of solution is to solve for two dependent parameters  $x_j$  and  $x_k$  as one varies a particular independent parameter  $x_i$ . That is, to observe the effect on  $x_j$  and  $x_k$  as  $x_i$  varies. Assume a particular independent parameter, denoted by  $x_{i,p}$ , has the solution  $(x_{j,p}, x_{k,p})$ . The procedure for predicting the new initial estimate is shown in Figure 1. To march along the  $x_i$  axis, a small increment is chosen such that  $x_{i,p+1} = x_{i,p} + \delta x_i$ . Thus the initial estimate  $(x_{j,p+1}, x_{k,p+1})$  is taken as  $(x_{j,p}, x_{k,p})$ , causing rapid convergence to the solution  $(x_{j,p+1}, x_{k,p+1})$ . The estimate for the new point  $x_{i,p+2}$  (i.e.  $(x_{j,p+2}^0, x_{k,p+2}^0)$ ) is obtained by linear extrapolation using the previous two solutions.

The step  $\delta x_i$  is small to ensure rapid convergence. Once convergence has been attained an arc of a circle is fitted through the three points  $x_{i,p}$ ,  $x_{i,p+1}$ , and  $x_{i,p+2}$ , to obtain an estimate for the dependent parameters at  $x_{i,p+3}$ . Fitting an arc of a circle allows one to rise large  $\delta x_i$  and still obtain accurate initial estimates  $(x_{j,p+3}^0, x_{k,p+3}^0)$  which are essential to obtain rapid convergence to the solution  $(x_{j,p+3}, x_{k,p+3})$ .

## 4. PROGRAM

Appendix A contains descriptions of the main subroutines used to solve the flutter equation. The user-supplied main program PREPFLSO and subroutine SYSCAL are described in Appendix B. A block diagram of all subroutines used in program VAFLUTSOLV is shown in Figure 2.

## 5. EXAMPLE

A simple two-degree-of-freedom system is considered here. This example (taken from Dowell et al, Reference [5]) was used to validate the program.

The typical section, is shown in Figure 3. If the angle of attack,  $\alpha$ , and the vertical displacement,  $h$ , are the generalized coordinates then the equations of motion about the elastic axis for the  $h$  and the  $\alpha$  components, respectively, become

$$m \ddot{h} + S_{\alpha} \ddot{\alpha} + K_h h + L = 0 \quad (7a)$$

$$\text{and } I_{\alpha} \ddot{\alpha} + S_{\alpha} \ddot{h} + K_{\alpha} \alpha - M_y = 0 \quad (7b)$$

Here	$m$	=	mass of section,
	$K_h$	=	bending stiffness,
	$K_{\alpha}$	=	torsional stiffness,
	$L$	=	resultant lift force per unit span
	$I_{\alpha}$	=	moment of inertia of the section about the e.a.,
	$M_y$	=	moment about elastic axis due to $L$ ,
	$e$	=	distance from e.a. to aerodynamic centre,
	$S_{\alpha}$	=	$m x_{\alpha}$ , and
	$x_{\alpha}$	=	distance from e.a to c.g.

Using quasi-steady aerodynamic theory the resultant section lift force,  $L$ , is given by

$$q c \frac{\partial C_L}{\partial \alpha} \left[ \alpha + \frac{\dot{h}}{U} \right] \quad (8)$$

where	$q$	=	$1/2 \rho U^2$
	$\rho$	=	density of air,
	$U$	=	flight velocity,
	$c$	=	wing chord, and
	$C_L$	=	lift coefficient.

6.

Using  $I_a = m c^2 \bar{r}_a^{-2}$ ,  $S_a = m \bar{x}_a c$  and  $M_y = L e$  and substituting eqn (8) into eqn (7), eqn (7) may be written as

$$m \ddot{h} + m \bar{x}_a c \ddot{\alpha} + K_h h + \frac{1}{2} \rho U^2 c \frac{\partial C_L}{\partial \alpha} \left[ \alpha + \frac{\dot{h}}{U} \right] = 0$$

and

$$m c^2 \bar{r}_a^{-2} \ddot{\alpha} + m \bar{x}_a c \ddot{h} + K_a \alpha - \frac{1}{2} \rho U^2 c e \frac{\partial C_L}{\partial \alpha} \left[ \alpha + \frac{\dot{h}}{U} \right] = 0 \quad (9)$$

Let

$$\begin{aligned} \bar{U} &= U/c\omega_a \\ \bar{h} &= h/c \\ \bar{r} &= U/c t \\ \bar{h}'' &= (U^2/c) \ddot{h}'' \\ \bar{\alpha}'' &= (U/c)^2 \ddot{\alpha}'' \\ \bar{e}_h^2 &= K_h/m \quad \text{and} \\ \bar{e}_a^2 &= K_a/I_a \end{aligned}$$

and by multiplying eqn (9) by  $(1/2 \rho U^2 c^2)^{-1}$ , eqn (9) can be non-dimensionalised to give, in matrix form,

$$\begin{aligned} \frac{2m}{\rho c} \begin{bmatrix} 1 & \bar{x}_a \\ \bar{x}_a & \bar{r}_a^2 \end{bmatrix} \begin{bmatrix} \bar{h}'' \\ \bar{\alpha}'' \end{bmatrix} + \frac{\partial C_L}{\partial \alpha} \begin{bmatrix} 1 & 0 \\ -\bar{e} & 0 \end{bmatrix} \begin{bmatrix} \bar{h}' \\ \bar{\alpha}' \end{bmatrix} \\ + \begin{bmatrix} \frac{2 K_h}{\rho \bar{U}^2 c \omega_a} & \frac{\partial C_L}{\partial \alpha} \\ 0 & \frac{2 K_a}{\rho \bar{U}^2 c \omega_a} - \bar{e} \frac{\partial C_L}{\partial \alpha} \end{bmatrix} \begin{bmatrix} \bar{h} \\ \bar{\alpha} \end{bmatrix} = 0 \end{aligned} \quad (10)$$

Note that the overbar denotes non-dimensional quantities.

Assuming the solution is oscillatory, then

$$\begin{aligned} \bar{h} &= \bar{h}_0 e^{iXt} \\ \bar{\alpha} &= \bar{\alpha}_0 e^{iXt} \end{aligned} \quad (11)$$

where  $\chi = (\Delta v + i v)$   
 $v = \omega c/U$   
 $\Delta = \text{decay rate (fraction of critical)}$

Alternatively,

$$\chi = \left( \frac{\Delta \omega}{\omega_\alpha} + i \frac{\omega}{\omega_\alpha} \right) \frac{1}{\bar{U}}$$

ie  $\chi = \bar{\chi} / \bar{U}$  (12)

Using eqns (11) and (12) and rearranging, eqn (10) becomes,

$$\left\{ \pi \beta \bar{\chi}^{-2} \begin{bmatrix} 1 & \bar{\chi}_\alpha \\ \bar{\chi}_\alpha & \bar{r}_\alpha^2 \end{bmatrix} + \pi \beta \left( \frac{\omega_h}{\omega_\alpha} \right)^2 \begin{bmatrix} 1 & 0 \\ 0 & \left( \frac{r_\alpha \omega_\alpha}{\omega_h} \right)^2 \end{bmatrix} + \bar{U} \frac{\partial C_L}{\partial \alpha} \begin{bmatrix} \bar{\chi} & \bar{U} \\ -\bar{\chi} e & -\bar{U} e \end{bmatrix} \right\} \begin{bmatrix} \bar{h}_0 \\ \bar{a}_0 \end{bmatrix} = 0 \quad (13)$$

where  $\beta = \frac{2m}{\pi \rho c}$ .

The above eqn is of the form

$$[ S(x) ] \underline{q}_0 = 0 \quad (14)$$

and so can be solved by our method.

Appendix C gives a list of the parameters, their corresponding values and listings of the user-supplied routines. Figure 4 contains plots of  $\omega/\omega_\alpha$  and  $\Delta\omega/\omega_\alpha$  versus  $\bar{U}$ . Comparisons with results published in Dowell et al. (Reference [5]) are also shown in Figure 4, and it can be seen that a good agreement has been obtained.

## 6. DISCUSSION

The following is a list of improvements that could be incorporated into the present program.

8.

1. A reasonable initial estimate is required so that the solution will converge quickly. Consequently one could incorporate a classical eigenvalue solver to obtain initial solutions of damping and frequency at  $V = 0$ .
2. An interpolation procedure to determine the flutter speed accurately.
3. Use of a conic or cubic spline extrapolation to estimate the next  $x_j$  and  $x_k$ . Presently only an arc of a circle is used.

There are a number of options contained in the original program (described in Reference [4]) that have, as of yet, not been included in the version at ARL. These options are:-

1. Display of error function (see eqn (6)) against number of iterations.
2. Stability test, i.e., if for a particular solution of  $x_j$  Vs  $x_i$ , say, the domain of stability is not particularly clear. This test determines which domain is stable.

## 7. CONCLUSION

A direct method of solving the flutter equations has been programmed at ARL. The program is interactive. Graphics routines, using DI-3000 software, have been incorporated to give the user an immediate picture of the solution's progress. The user may evaluate the relationship between any two parameters for which the flutter equation is solved, the only prerequisite being that these are variables in the subroutine that calculates the system matrix (i.e. subroutine SYSCAL). This method also provides estimates of subcritical response data.

## REFERENCES

1. Jocelyn Lawrence, A. and Jackson P.  
"Comparison of Different Methods of Assessing the Free Oscillatory Characteristics of Aeroelastic Systems",  
RAE CP No. 1084, 1970.
2. Hassig, H.J.  
"An Approximate True Damping Solution of the Flutter Equation by Determinant Iteration",  
Journal of Aircraft, Vol. 8, No. 11, 1971. p833.
3. Herszberg, I.  
"Body Freedom Flutter of Mk IIIB Jindivik with Mk 7/8 Pods and W.R.E. Autopilot",  
G.A.F. Project Report B3B 9/72, 1972.
4. Herszberg, I.  
"Direct Solution of Flutter Equations with Interactive Graphics Procedure",  
27th Annual Conference on Aviation and Astronautics Tel Aviv, February, 1985.
5. Dowell, E.H.; Curtiss, Jr. H.C.; Scanlan, R.H.; and Sisto, F.  
"A Modern Course in Aeroelasticity",  
Sijthoff and Noordhoff, 1980.

## APPENDIX A

### Flutter Solver Routines

Table 1 gives a list of the routines used in VAFLUTSOLV. A more detailed discussion of each routine, except for the user supplied routines (see Appendix B), is given below.

#### Subroutine INIT

This subroutine prompts the user on -

- (a) the choice of independent and dependent parameters to be used initially, and
- (b) the initial estimates of the dependent parameters for a particular independent parameter.

If convergence has not been achieved in 20 iterations (this check is carried out in subroutine F2V) then the user has the option of entering another estimate or continuing with the present estimate.

#### Subroutine OUTP

This subroutine outputs the present values of the independent and dependent parameters.

#### Subroutine CHAND (OPTIONS, IOPT, ICOM)

When called, this subroutine displays a list of options contained in the character array OPTIONS. Generally the integer variable IOPT specifies the number of options available and ICOM is the integer returned to the caller routine (i.e. ICOM is the option selected). Other variations are, if on entry

IOPT - 0 then the DI 3000 graphics software is initialized,  
IOPT - -1 then the screen is cleared, and

(A.2)

IOPT = -2 then the screen is set up for interactive plotting.

If on calling CMAND, ICOM = -2, then options are set up but no command is returned to the caller routine

Subroutine TPTEXT

This routine provides a list of parameters with corresponding values.

Subroutine DLABEL (XMIN, YMIN, XMAX, YMAX, LABX, LABY, SIZE)

When called, this subroutine selects suitable scaling for the x and y axes. The graph axes, with appropriate labelling, are drawn and stored in the integer array, PICT1, using routines from GRAFMAKER graphics software (which is based on DI 3000 graphics software).

The first four dummy arguments, listed above, give the lower and upper limits of the variables to be plotted (here x and y denote the horizontal and vertical axes, respectively) Character variables LABX and LABY contain the x and y label names, respectively. The desired maximum dimension of the plot is given by the real variable SIZE, where the screen is assumed to be 10 units square (i.e.  $SIZE \leq 10$ ).

Subroutine DISP

On calling subroutine DISP a plot of the first dependent parameter versus the independent parameter is displayed. Subroutine CMAND is then called; thus, the following menu is displayed:

VRBLS	1
SELECT	2
CONT	3
DELETE	4
STABLITY	5
STEP SZ	6
SKP PT	7
PLOT	8
CHG VBL	9

(A.3)

Once an option is selected, subroutine DISP then executes the command. A description of each option is given below:-

(1) VRBLS - step (a) Displays a plot of the second dependent parameter versus the independent parameter,

step (b) Displays the menu

```
SHW IVZ  1
CHG VBL  2
CONT     3
```

where SHW IVZ again displays a plot of the first dependent parameter versus the independent parameter, CHG VBL allows the user to select new independent and dependent parameters and CONT causes the main menu to be displayed again.

(2) SELECT Allows the user to select new starting values. This is to be used if the solution has not converged or if the dependent and independent parameters have been changed.

(3) CONT Returns the user to the caller routine (DRIVNF).

(4) DELETE Deletes the previous entry.

(5) STABILITY Not operative.

(6) STEP SZ Displays the menu,

```
INCRSE  1
DECRSE  -1
SAME    0
```

By entering 1, -1, or 0 the user may double, halve or not alter the step size, respectively.

(7) SKP PT Not operative.

- (8) PLOT Stores plots, presently displayed on the screen, into a metafile (Refer to the DI-3000 manual). The name of the metafile is specified by the user after the screen displays the prompt:

...ENTER FILE NAME.

- (9) CHG VBL Allows the user to select new independent and dependent parameters.

Subroutine VORT (X0, Y0, X1, Y1, X2, Y2, ICL)

Given co-ordinates of a circle centre (X0, Y0) and two points (X1, Y1), (X2, Y2) on the circle, subroutine VORT determines the direction from 1 to 2. Subroutine VORT returns ICL = 1 if the direction is clockwise or ICL = -1 if it is anticlockwise.

Subroutine ANG (X, Y, TH)

Given the co-ordinates (X, Y), subroutine ANG calculates the angle in radians where  $0 < TH < 2\pi$ .

Subroutine CIRCLE (X, Y, XC, YC, RAD)

This subroutine establishes the centre of a circle (XC, YC) that passes through three points contained in the dummy arrays X(3) and Y(3). The radius of the circle, RAD, is also found.

Subroutine NF

This subroutine factorises the dependent parameters, using the real array FACTR (see PREPFLSO), and then calls subroutine F2V.

Subroutine FG (XX, Y, F, G)

Given dependent parameters XX and Y this subroutine calls subroutine SYSCAL to calculate the system matrix S. Subroutine CMINV is then called to obtain the complex determinant of S (i.e.  $|S| = F + i G$ ).

(A.5)

Subroutine CMINV (A, N, ND, B, L, D, ERROR)

If L is negative, zero or positive then the N x N matrix A is inverted, by Gauss-Jordan Pivotal elimination, and A is replaced by its inverse.

If L is positive then the N x L matrix B is manipulated by the same procedure as used to invert A and is then replaced by the resulting matrix.

Subroutine F2V (XX, YY, EPS, NC, IR, FG)

This subroutine calls subroutine FG to determine the determinant |S| given the two dependent parameter estimates (XX, YY). These estimates are then refined by using the secant method. Convergence of the estimates is checked by using eqn (6). If either convergence is attained or the maximum number of iterations is exceeded then the user is returned to NF and hence to DRIVNF.

Subroutine DRIVNF

Subroutine DRIVNF is the main subroutine of program VAFLUTSOLV. This subroutine calls INIT to obtain the initial estimates from the user. That is, for a particular independent parameter value ( $x_{1,p}$ ), estimates  $x_{j,p}^0$  and  $x_{k,p}^0$  are supplied by the user. Subroutine NF is then called (at least three times) to evaluate solutions for the first two increments in the independent parameter (i.e. solutions are obtained at  $x_{1,p}$ ,  $x_{1,p+1}$ , and  $x_{1,p+2}$ ) in a manner similar to that discussed in Section 3. Subroutine CIRCLE and VORT are then called to determine the initial estimates for the next point ( $x_{1,p+4}$ ). Once again NF is called to solve for the dependent parameters ( $x_{j,p+4}$ ,  $x_{k,p+4}$ ). If the solution has failed to converge (see Subroutine F2V) then the step size of the independent parameter is halved, and the above procedure is repeated. If the initial estimate, at half the step size, does not converge then the step size is halved again. Rapidly varying functions of  $x_1$  usually cause estimates, when fitting an arc of a circle, to be inaccurate and hence convergence is hard to attain.

If convergence has not occurred after carrying out the above procedure then the program steps backward, in small steps, to obtain two

(A.6)

other solutions just upstream of the last convergent point (i.e.  $x_{i,p+3}$  in this case). With these three points a new estimate  $(x_{j,p+4}^{\circ}, x_{k,p+4}^{\circ})$  is evaluated, again fitting an arc of a circle, and the program proceeds as usual.

Interactive graphics displays are obtained by calling Subroutine DISP.

## APPENDIX B

### User Supplied Routines

#### Program PREPFLSO

```
COMMON/VR/   VVRBL, FACTR, IVRBL, IV1, IV2, NM
COMMON/LAB   VRBL
```

```
CHARACTER *10 VRBL (12)
DIMENSION VVRBL (12), FACTR (12)
DATA (VVRBL(I), I = 1,12)/      /
DATA (VRBL(I), I = 1,12)/      /
CALL DRIVNF
END
```

This program sets up the number of parameters, the parameter values, the independent parameter and the number of degrees-of-freedom of the problem. The variables that need to be defined are:-

- VVRBL - Array containing the values of the parameters (allowed up to 12 parameters).
- FACTR - Array of values used to factorize the parameters (set  $FACTR(I) = 1/VVRBL(I)$ ).
- IVRBL - Index of the independent parameter.
- IV1,IV2 - Indices of the dependent parameters.
- NM - Number of degrees-of-freedom of the system. ( $NM \leq 10$ ).
- VRBL - Character array containing a descriptor of the corresponding parameter, up to 10 characters are allowed (i.e.  $VRBL(I)$  contains descriptor of parameter  $VVRBL(I)$ ).

(B.2)

Subroutine SYSCAL

```
COMMON/VR/VVRBL, FACTR, IVRBL, IV1, IV2, NM  
COMMON/SYS/SYSM  
DIMENSION VVRBL(12), FACTR(12)  
COMPLEX SYSM(10,10)
```

Given the array of parameters, VVRBL, this routine will calculate the system matrix S.

SYSM - Complex array which contains the system matrix S. The order of matrix S is NM x NM, in this case  $NM \leq 10$ .

APPENDIX C

Two Degree-of-Freedom Example

Table 2 gives a list of values of the system parameters used to solve eqn 13. Listings of the user supplied routines are given below.

PROGRAM PREPFLSO

```

C=====
C
C      This sets up a two degree of freedom flutter
C      problem. (i.e. eqn.14) (see DOWELL et al., p80)
C
C      This program starts VAFLUTSOLV by:
C      (1) specifying the no. of D.O.F. of the system.
C      (2) specifying the parameters of the system and
C          their initial values.
C      (3) specifying the initial independent parameter.
C

```

```

COMMON/VR/VVRBL, FACTR, IVRBL, IV1, IV2, NM
COMMON/LAB/VRBL
CHARACTER*10 VRBL(12)
DIMENSION VVRBL(12), FACTR(12)
DATA (VVRBL(I), I=1, 12)/
1 10., 0.5, 0.5, 0.20, 0.4, 1.0, 0.0, 0.5, 6.28318, 0.0,
2 0.0, 0.0/
DATA (VRBL(I), I=1, 10)/
: ' BETA      ', ' FREQ. RAT. ',
: ' RAD. GYRA ', ' X MASS   ',
: ' R LIFT   ', ' VELOCITY ',
: ' DEC. RATIO', ' FREQ    ',
: ' DC1/DA   ', ' FREQ    '//
DATA PI/3.14159265/

```

C	INDEX	VARIABLE	VALUE
C	1	BETA	10.
C	2	FREQ. RATIO	0.5
C	3	RAD. GYRAT.	0.5
C	4	MASS MOM. ARM	0.20
C	5	LIFT MOM. ARM	0.4
C	6	NON. DIM. VEL.	0.001
C	7	NON. DIM. DECAY RA.	0.001
C	8	NON. DIM. FREQ1	0.5
C	9	LIFT CURVE SL.	2PI

```

C** D.O.F. OF THE SYSTEM **
NM=2
C** SET UP FACTORIZING VALUES **
DO 10 I=1, 12
  FACTR(I)=0.0
DO 20 I=1, 10
  IF(VVRBL(I).EQ.0.0) THEN
    FACTR(I)=1.0
  ELSE
    FACTR(I)=1./VVRBL(I)
  ENDIF
20 CONTINUE

C** SET UP INITIAL INDEP VARBLE **
IVRBL=6

CALL DRIVNF

STOP
END

```

SUBROUTINE SYSCAL

C=====

C

C        This routine sets up the system matrix S.

C

COMMON/VR/VVRBL, FACTR, IVRBL, IV1, IV2, NM

COMMON/SYS/ SYM

DIMENSION VVRBL(12), FACTR(12)

COMPLEX SYM(10,10), M(2,2), S(2,2), A(2,2), CHI, II

DATA PI/3.14159265/

II=CMPLX(0.0,1.0)

C\*\* MASS/INERTIA MATRIX \*\*

M(1,1)=1.0

M(1,2)=0.20

M(2,1)=M(1,2)

M(2,2)=0.25

CHI=VVRBL(7)+(II\*VVRBL(8))

C\*\* STIFFNESS MATRIX \*\*

S(1,1)=1.0

S(1,2)=0.0

S(2,1)=0.0

S(2,2)=(VVRBL(3)\*\*2.)\*(1./(VVRBL(2)\*\*2.))

C\*\* AERO. MATRIX \*\*

A(1,1)=CHI

A(1,2)=VVRBL(6)

A(2,1)=-CHI\*0.4

A(2,2)=-0.4\*VVRBL(6)

C\*\* CALC. SYSTEM MATRIX \*\*

DO 10 I=1,2

DO 10 J=1,2

M(I,J)=31.4\*(CHI\*\*2)\*M(I,J)

S(I,J)=7.85\*S(I,J)

A(I,J)=6.28\*A(I,J)\*(VVRBL(6))

10 SYM(I,J)=M(I,J)+S(I,J)+A(I,J)

RETURN

END

TABLE 1

Summary of Flutter Solver Routines

Subroutines in VAFLUTSOLV File	Description
INIT	Input data, start program
OUTP	Output data
CMAND	Graphics input/output data
DISP	Graphics, Output data
DLABEL	Graphics, Output data
TPTEXT	Graphics, Output data
DRIVNF	Main solving routine
VORT	Called by DRIVNF
ANG	Called by DRIVNF
CIRCLE	Called by DRIVNF
NF	Called by DRIVNF, calls F2V
FG	Calls SYSCAL to determine S
F2V	Refines estimates $x_j, x_k$
CMINV	Input A determine $A^{-1},  A $
<u>Other Routines</u>	
SYSCAL	User supplied routine to determine S
PREPFLSO	User supplied routine to set up problem

(NOTE: These files are contained in sub-directory FLUTSOLVER)

TABLE 2

List of Parameter Values used to Solve EQN (14)

Parameter	Description (VRBL)	Index of VVRBL	Parameter Value or Initial Guess (VVRBL)	Factorising Value (FACTR)
$\beta$	Mass Parameter	1	10	0.1
$\omega_h/\omega_\alpha$	Freq. Ratio	2	0.5	2.0
$\bar{r}_\alpha$	Radius of Gyration	3	0.5	2.0
$\bar{x}_\alpha$	Mass Moment Arm	4	0.2	5.0
$\bar{e}$	Lift Moment Arm	5	0.4	2.5
$\bar{U}$	Non-dim Velocity	6	0.001	1000
$\omega/\omega_\alpha$	Non-dim. Freq.	7	0.5	2.0
$\Delta\omega/\omega_\alpha$	Decay Rate	8	0.001	1000
$\partial C_L/\partial\alpha$	Lift Curve Slope	9	6.283	1.66

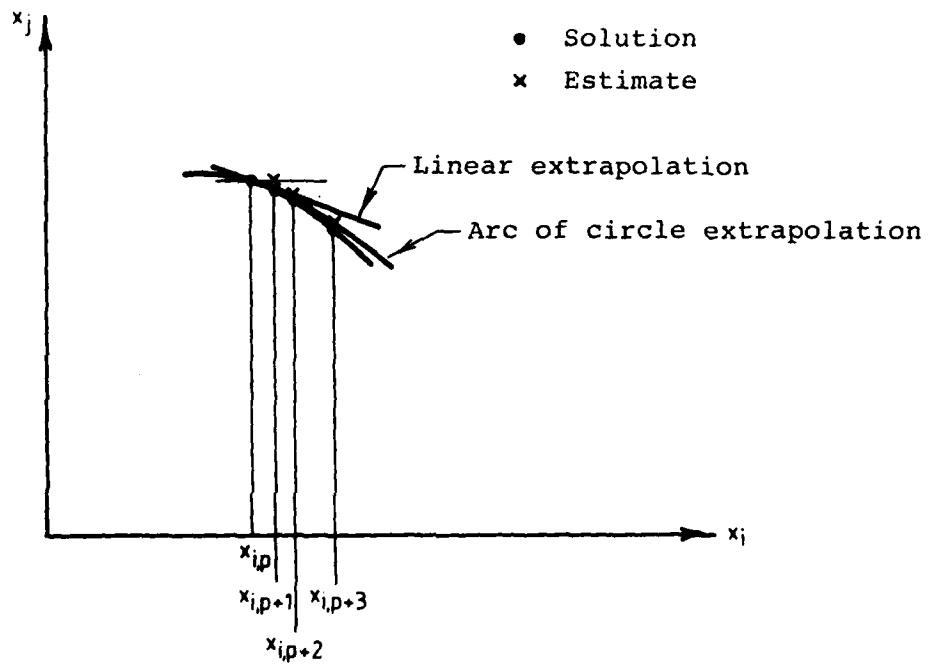


FIG. 1. METHOD FOR PREDICTING THE NEW INITIAL ESTIMATE OF  $x_j$  (SIMILAR PROCEDURE FOR  $x_k$ )

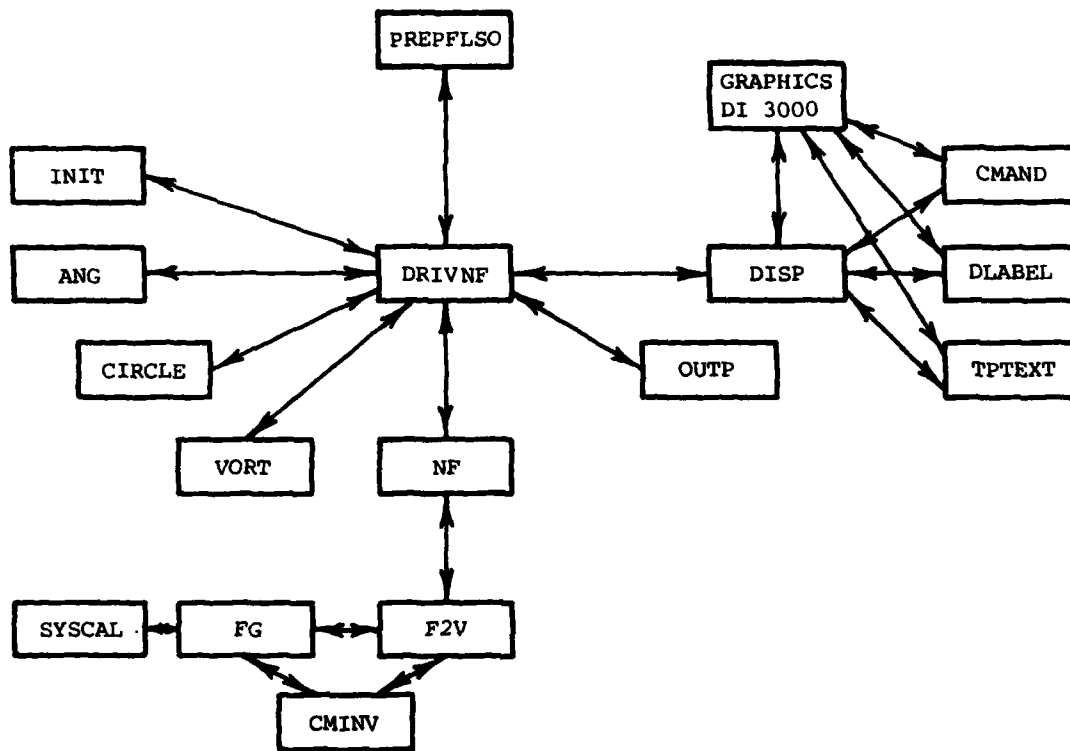


FIG. 2. BLOCK DIAGRAM OF THE SUBROUTINES USED TO SOLVE THE FLUTTER EQUATIONS

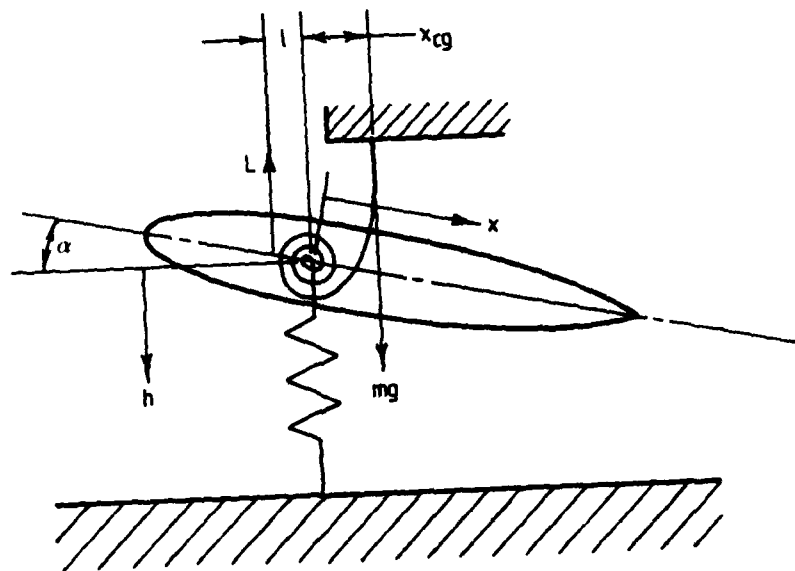


FIG. 3 GEOMETRY OF TYPICAL AIRFOIL SECTION

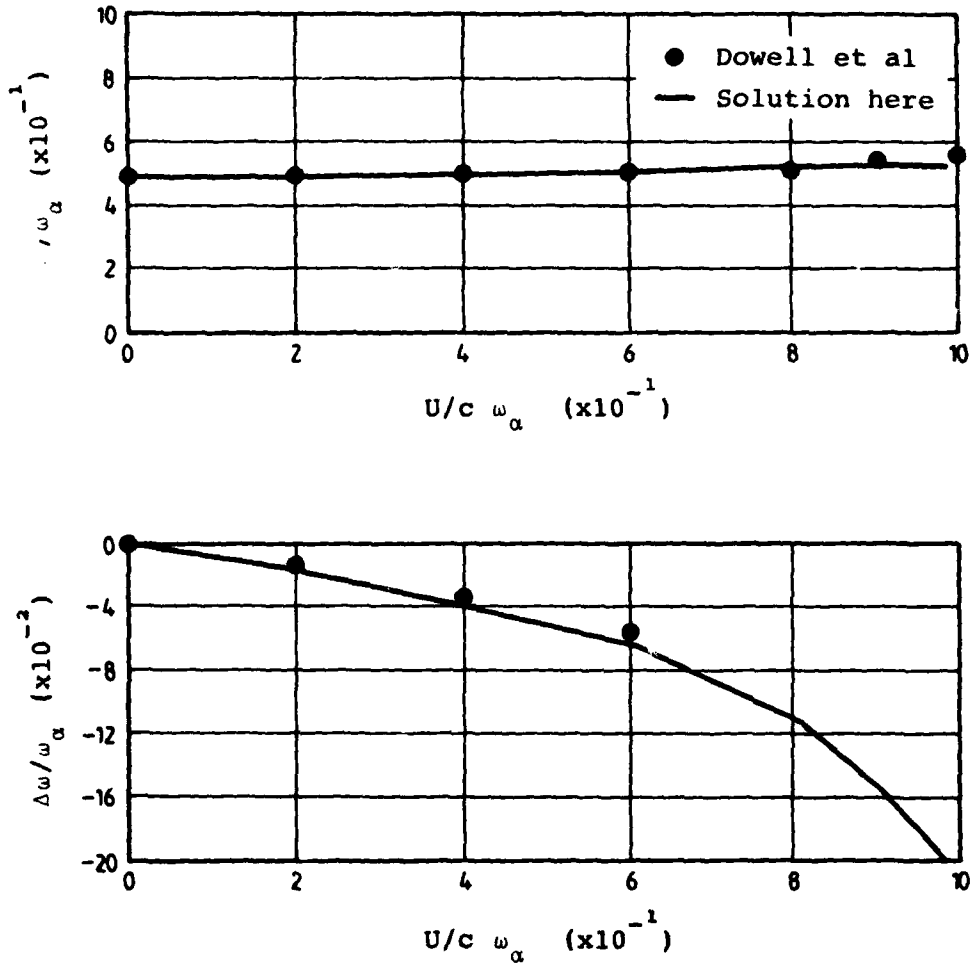


FIG. 4(a) VARIATION OF NON-DIMENSIONAL FREQUENCY AND DAMPING WITH NON-DIMENSIONAL VELOCITY FOR THE FIRST NATURAL MODE

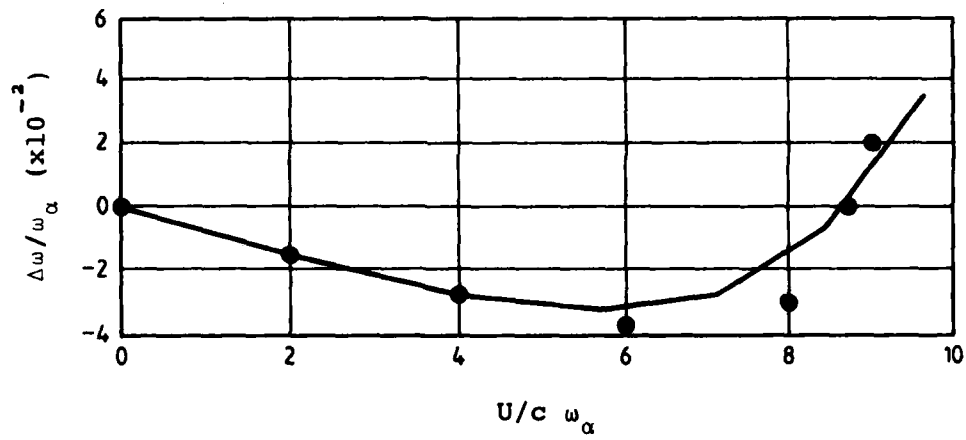
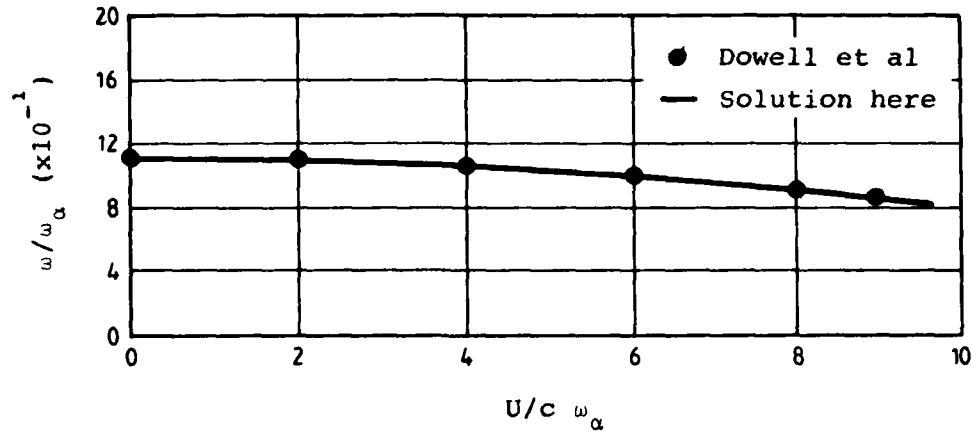


FIG. 4 (b) VARIATION OF NON-DIMENSIONAL FREQUENCY AND DAMPING WITH NON-DIMENSIONAL VELOCITY FOR THE SECOND NATURAL MODE

**DISTRIBUTION LIST**

**AUSTRALIA**

**DEPARTMENT OF DEFENCE**

Defence Central

Chief Defence Scientist )  
Deputy Chief Defence Scientist )  
Superintendent, Science and Program ) 1 copy  
Administration )  
Controller, External Relations, Projects )  
and Analytical Studies )  
Defence Science Adviser (UK) (Doc Data sheet only)  
Counsellor, Defence Science (USA) (Doc Data sheet only)  
Defence Central Library  
Document Exchange Centre, DISB (18 copies)  
Joint Intelligence Organisation  
Librarian H Block, Victoria Barracks, Melbourne  
Director General - Army Development (NSO) (4 copies)

Aeronautical Research Laboratories

Director  
Library  
Divisional File - Structures  
Author: S.C. Galea  
R. Jones  
P. Farrell  
T. Ryall  
B. Emslie  
A. Goldman  
W. Waldman

Defence Research Centre

Library

Air Force Office

Air Force Scientific Adviser  
Aircraft Research and Development Unit  
Scientific Flight Group  
Library  
Technical Division Library  
Director General Aircraft Engineering - Air Force  
RAAF Academy, Point Cook

Central Studies Establishment

Information Centre

Government Aircraft Factories

Manager  
Library

**DEPARTMENT OF AVIATION**

Library  
Flight Standards Division

**UNIVERSITIES AND COLLEGES**

Adelaide	Barr Smith Library Professor of Mechanical Engineering
Flinders	Library
La Trobe	Library
Melbourne	Engineering Library
Monash	Hargrave Library
Newcastle	Library
Sydney	Engineering Library
NSW	Physical Sciences Library Professor R.A.A. Bryant, Mechanical Engineering
Queensland	Library
Tasmania	Engineering Library
Western Australia	Library
RMIT	Associate Professor J.A.Cole Mechanical Engineering, Library I. Herzberg, Aero. Engineering Dept.

SPARES (10 copies)  
TOTAL (74 copies)

## DOCUMENT CONTROL DATA

1. a. AR No AR-004-028	1. b. Establishment No ARL-STRUC-TM-414	2. Document Date JUNE, 1985	3. Task No DST 82/053
4. Title INTERACTIVE FLUTTER SOLVER AT ARL		5. Security a. document: UNCLASSIFIED b. title      c. abstract U              U	6. No Pages 8 7. No Refs 5
8. Author(s) S.C. GALEA		9. Downgrading Instructions	
10. Corporate Author and Address Aeronautical Research Laboratories P.O. Box 4331 Melbourne, Vic. 3001		11. Authority (as appropriate) a. Sponsor b. Security c. Downgrading d. Approval	
12. Secondary Distribution (of this document) Approved for Public Release			
Overseas enquirers outside stated limitations should be referred through ASDIS, Defence Information Services Branch, Department of Defence, Campbell Park, CANBERRA ACT 2601			
13. a. This document may be ANNOUNCED in catalogues and awareness services available to ... No Limitations			
13. b. Citation for other purposes (ie casual announcements) may be (select) unrestricted (or) as per 13 a.			
14. Descriptors → Aeroelasticity; Flutter Differential equations Computer programs <i>Resonance</i>		15. COSATI Group 01020 09020	
16. Abstract An interactive direct flutter solving routine has been installed on ARL's ELXSI computer. Interactive graphics routines, using DI-3000 graphics, software, have been incorporated to give the user a progressive picture of the solution. Estimates of subcritical response data may also be determined using this method. A description of the method and the associated software is presented here. Also included is a two-dimensional flutter problem, using quasi-steady aerodynamics, which is solved by this direct flutter solver. <i>Keywords:</i>			

This page is to be used to record information which is required by the Establishment for its own use but which will not be added to the DISTIS data base unless specifically requested.

16. Abstract (Cont'd)		
17. Imprint  Aeronautical Research Laboratories, Melbourne		
18. Document Series and Number  Structures Technical Memorandum 414	19. Cost Code  23-6925	20. Type of Report and Period Covered
21. Computer Programs Used		
22. Establishment File Ref(s)		

ATE  
LMED  
-8