A FORTRAN PROGRAM FOR
DETERMINING AIRCRAFT STABILITY
AND CONTROL DERIVATIVES
FROM FLIGHT DATA

Richard E. Maine and Kenneth W. Iliff

Flight Research Center
Edwards, Calif. 93523

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • APRIL 1975
A FORTRAN program written in FORTRAN IV for the estimation of aircraft stability and control derivatives is presented. The program uses a maximum likelihood estimation method. Two associated programs for routine, related data handling are also included. The three programs form a package that can be used by relatively inexperienced personnel to process large amounts of data with a minimum of manpower. This package has been used to successfully analyze 1500 maneuvers on 20 aircraft. It is designed to be used without modification on as many types of computers as feasible. Program listings and sample check cases are included.
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>SYMBOLS</td>
<td>1</td>
</tr>
<tr>
<td>PARAMETER ESTIMATION</td>
<td>5</td>
</tr>
<tr>
<td>SOLUTION BY THE MODIFIED NEWTON-RAPHSON METHOD</td>
<td>7</td>
</tr>
<tr>
<td>INCLUSION OF A PRIORI INFORMATION</td>
<td>8</td>
</tr>
<tr>
<td>CONFIDENCE LEVELS</td>
<td>9</td>
</tr>
<tr>
<td>DESCRIPTION AND USE OF PROGRAMS</td>
<td>10</td>
</tr>
<tr>
<td>MMLE – MODIFIED MAXIMUM LIKELIHOOD ESTIMATION PROGRAM</td>
<td>11</td>
</tr>
<tr>
<td>Input Description</td>
<td>11</td>
</tr>
<tr>
<td>Title card</td>
<td>11</td>
</tr>
<tr>
<td>NAMELIST/INPUT/</td>
<td>11</td>
</tr>
<tr>
<td>Time cards</td>
<td>18</td>
</tr>
<tr>
<td>Matrix input</td>
<td>18</td>
</tr>
<tr>
<td>A matrix (4 by 4)</td>
<td>19</td>
</tr>
<tr>
<td>B matrix (4 by 5 to 4 by 8)</td>
<td>19</td>
</tr>
<tr>
<td>AA array (4 by 4)</td>
<td>20</td>
</tr>
<tr>
<td>BB array (4 by 5 to 4 by 8)</td>
<td>20</td>
</tr>
<tr>
<td>AR matrix (4 by 4)</td>
<td>20</td>
</tr>
<tr>
<td>BR matrix (4 by 5 to 4 by 8)</td>
<td>20</td>
</tr>
<tr>
<td>APRA matrix (4 by 4)</td>
<td>21</td>
</tr>
<tr>
<td>APRB matrix (4 by 5 to 4 by 8)</td>
<td>21</td>
</tr>
<tr>
<td>AP array (3 by 4)</td>
<td>21</td>
</tr>
<tr>
<td>BP array (3 by 5 to 3 by 8)</td>
<td>22</td>
</tr>
<tr>
<td>R matrix (4 by 4)</td>
<td>23</td>
</tr>
<tr>
<td>D1 matrix (5 by 5 to 7 by 7)</td>
<td>23</td>
</tr>
<tr>
<td>ENDCASE</td>
<td>23</td>
</tr>
<tr>
<td>Card input</td>
<td>23</td>
</tr>
<tr>
<td>Data file input</td>
<td>24</td>
</tr>
<tr>
<td>Output Description</td>
<td>24</td>
</tr>
<tr>
<td>Printed output</td>
<td>24</td>
</tr>
<tr>
<td>Plotted output</td>
<td>25</td>
</tr>
<tr>
<td>Punched card output</td>
<td>25</td>
</tr>
<tr>
<td>SETUP – PREPROCESSING PROGRAM</td>
<td>26</td>
</tr>
<tr>
<td>Input Description</td>
<td>26</td>
</tr>
<tr>
<td>Options</td>
<td>26</td>
</tr>
<tr>
<td>Vehicle characteristics</td>
<td>27</td>
</tr>
<tr>
<td>NAMELIST/WIND/</td>
<td>27</td>
</tr>
<tr>
<td>Vehicle name</td>
<td>28</td>
</tr>
<tr>
<td>Lateral–directional weighting matrix</td>
<td>29</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Longitudinal weighting matrix</td>
<td>29</td>
</tr>
<tr>
<td>Lateral-directional APRA and APRB matrices</td>
<td>29</td>
</tr>
<tr>
<td>Longitudinal APRA and APRB matrices</td>
<td>29</td>
</tr>
<tr>
<td>Predicted derivatives</td>
<td>29</td>
</tr>
<tr>
<td>Angle-of-attack breakpoints</td>
<td>30</td>
</tr>
<tr>
<td>Mach number breakpoints</td>
<td>30</td>
</tr>
<tr>
<td>Arbitrary parameter breakpoints</td>
<td>30</td>
</tr>
<tr>
<td>User-supplied data for subroutine COND1</td>
<td>30</td>
</tr>
<tr>
<td>Input tape data</td>
<td>30</td>
</tr>
<tr>
<td>Case specification</td>
<td>30</td>
</tr>
<tr>
<td>Time card</td>
<td>30</td>
</tr>
<tr>
<td>NAMELIST/COND/</td>
<td>30</td>
</tr>
<tr>
<td>End card</td>
<td>32</td>
</tr>
<tr>
<td>Output Description</td>
<td>32</td>
</tr>
<tr>
<td>SUMMARY — PLOTTING PROGRAM</td>
<td>32</td>
</tr>
<tr>
<td>Input Description</td>
<td>32</td>
</tr>
<tr>
<td>Title card</td>
<td>32</td>
</tr>
<tr>
<td>NAMELIST/WIND/</td>
<td>32</td>
</tr>
<tr>
<td>Predicted derivatives</td>
<td>34</td>
</tr>
<tr>
<td>Flight data</td>
<td>34</td>
</tr>
<tr>
<td>Plotting instructions</td>
<td>34</td>
</tr>
<tr>
<td>End card</td>
<td>35</td>
</tr>
<tr>
<td>Output description</td>
<td>35</td>
</tr>
<tr>
<td>CONCLUDING REMARKS</td>
<td>35</td>
</tr>
<tr>
<td>APPENDIX A — MMLE PROGRAM AND SUBROUTINES</td>
<td>36</td>
</tr>
<tr>
<td>MAIN MMLE PROGRAM</td>
<td>36</td>
</tr>
<tr>
<td>SUBROUTINE EDIT</td>
<td>40</td>
</tr>
<tr>
<td>SUBROUTINE DATA</td>
<td>47</td>
</tr>
<tr>
<td>SUBROUTINE AGIRL</td>
<td>53</td>
</tr>
<tr>
<td>SUBROUTINE OUTPUT</td>
<td>61</td>
</tr>
<tr>
<td>SUBROUTINE THPLOT</td>
<td>66</td>
</tr>
<tr>
<td>SUBROUTINE APRPLT</td>
<td>70</td>
</tr>
<tr>
<td>SUBROUTINE MATLD</td>
<td>72</td>
</tr>
<tr>
<td>SUBROUTINE MAK</td>
<td>73</td>
</tr>
<tr>
<td>SUBROUTINE DERIV</td>
<td>74</td>
</tr>
<tr>
<td>SUBROUTINE Cramer</td>
<td>75</td>
</tr>
<tr>
<td>SUBROUTINE Cramer</td>
<td>77</td>
</tr>
<tr>
<td>SUBROUTINE AMULT</td>
<td>78</td>
</tr>
<tr>
<td>SUBROUTINE DMULT</td>
<td>79</td>
</tr>
</tbody>
</table>
Determination of aircraft stability and control derivatives from flight data is of great importance in flight testing and control system design. Several methods have been used, but recent interest has turned toward maximum likelihood estimators. In addition to producing the "best" possible estimates as defined by some probabilistic criterion, these methods can be automated to a large extent.

Experience at the NASA Flight Research Center has shown that derivatives can be extracted with minimum effort by relatively inexperienced personnel using maximum likelihood estimators. Others have had some difficulty, perhaps partially due to inadequately designed programs. A production version of a maximum likelihood estimation program has been developed and used at the Flight Research Center to determine aircraft stability and control derivatives from large amounts of flight data. The program was designed to be compatible with as many types of computers as feasible and was structured to accommodate alterations easily. The program is applicable to many linear parameter estimation problems, although several of the features are intended specifically for aircraft stability and control applications. Reference 1 discusses an earlier program from which this maximum likelihood estimation program was conceptually derived.

This report presents the modified maximum likelihood estimation computer program used at the Flight Research Center for derivative extraction as well as associated programs for table lookup of initial estimates of the derivatives and for plotting results. Program listings and sample check cases for each program are included in the appendixes.

**SYMBOLS**

Parenthetical symbols are computer identifiers for data channels.

\[ A \]  
stability matrix, or axial force (appendix E)
\( a_n \) (AN) \hspace{1cm} \text{vertical acceleration, } g

\( a_x \) (AX) \hspace{1cm} \text{longitudinal acceleration, } g

\( a_y \) (AY) \hspace{1cm} \text{lateral acceleration, } g

\( B \) \hspace{1cm} \text{control matrix}

\( C_m \) \hspace{1cm} \text{dimensionless pitching-moment coefficient}

\( C_n \) \hspace{1cm} \text{dimensionless yawing-moment coefficient}

\( C_Z \) \hspace{1cm} \text{dimensionless normal-force coefficient}

\( c \) \hspace{1cm} \text{vector of unknowns}

\( c_0 \) \hspace{1cm} \text{a priori value of } c

\( D1 \) \hspace{1cm} \text{signal weighting matrix}

\( D2 \) \hspace{1cm} \text{a priori weighting matrix}

\( E[ ] \) \hspace{1cm} \text{expected value}

\( E_k \) \hspace{1cm} \text{relative error}

\( G \) \hspace{1cm} \text{observation matrix}

\( g \) \hspace{1cm} \text{acceleration of gravity, m/sec}^2 \ (\text{ft/sec}^2)

\( H \) \hspace{1cm} \text{observation matrix}

\( I \) \hspace{1cm} \text{identity matrix}

\( I_X \) \hspace{1cm} \text{moment of inertia about the longitudinal axis, kg-m}^2 \ (\text{slug-ft}^2)

\( I_{XZ} \) \hspace{1cm} \text{cross-product of inertia about the longitudinal and normal axes, kg-m}^2 \ (\text{slug-ft}^2)

\( I_Z \) \hspace{1cm} \text{moment of inertia about the normal axis, kg-m}^2 \ (\text{slug-ft}^2)

\( i \) \hspace{1cm} \text{time index}

\( J \) \hspace{1cm} \text{cost functional}

2
$L$, rolling moment divided by moment of inertia about longitudinal axis, rad/sec$^2$

$L_0, L_{02}, L_{03}, L_{04}$, rolling acceleration equation biases

$M$, pitching moment divided by moment of inertia about lateral axis, rad/sec$^2$

$M_0, M_{02}, M_{03}, M_{04}$, pitching acceleration equation biases

$N$, yawing moment divided by moment of inertia about normal axis, rad/sec$^2$, or number of time points

$N_0, N_{02}, N_{03}, N_{04}$, yawing acceleration equation biases

$p$ (P), roll rate, deg/sec or rad/sec

$q$ (Q), pitch rate, deg/sec or rad/sec

$q$, dynamic pressure, N/m$^2$ (lb/ft$^2$)

$R$, acceleration transformation matrix

$r$ (R), yaw rate, deg/sec or rad/sec

$S$, reference area, m$^2$ (ft$^2$)

$s$, auxiliary time variable, sec

$T$, total time, sec

$t$, time, sec

$\Delta t$, time interval between samples, sec

$u$, control vector

$V$, velocity, m/sec (ft/sec)

$v$, variable bias vector

$W$, aircraft weight, N (lb)

$X$, longitudinal force divided by mass, m/sec$^2$ (ft/sec$^2$)

$X_0, X_{02}, X_{03}, X_{04}$, longitudinal acceleration equation biases
\( x \)  
state vector  

\( Y \)  
side force divided by mass and velocity, rad/sec  

\( Y_0, Y_{02}, Y_{03}, Y_{04} \)  
side force equation biases  

\( y \)  
computed observation vector  

\( Z \)  
normal force divided by mass and velocity, rad/sec  

\( Z_0, Z_{02}, Z_{03}, Z_{04} \)  
ormal force equation biases  

\( z \)  
measured observation vector  

\( \alpha \, (A) \)  
angle of attack, deg or rad  

\( \beta \, (B) \)  
angle of sideslip, deg or rad  

\( \delta \)  
control, deg or rad  

\( \delta_a \, (DA) \)  
aileron position, deg or rad  

\( \delta_c, \delta_1, \delta_2 \)  
extra controls, deg or rad  

\( (DC, D1, D2) \)  

\( \delta_e \, (DE) \)  
elevator position, deg or rad  

\( \delta_r \, (DR) \)  
rudder position, deg or rad  

\( \eta \)  
noise vector  

\( \theta \, (THET) \)  
pitch attitude, deg or rad  

\( \dot{\theta}_0, \dot{\theta}_{02}, \dot{\theta}_{03}, \dot{\theta}_{04} \)  
biases in Euler pitch rate equation  

\( \tau \)  
revised time interval, sec  

\( \varphi \, (PHI) \)  
Euler roll attitude, deg or rad  

\( \dot{\varphi}_0, \dot{\varphi}_{02}, \dot{\varphi}_{03}, \dot{\varphi}_{04} \)  
biases in Euler roll rate equation
\[ \nabla_c \] \text{gradient with respect to } c \\
\[ \nabla^2_c \] \text{second gradient with respect to } c \text{ (Hessian matrix)} \\
\[ 0 \] null matrix \\
\text{Superscript:} \\
\[ * \] transpose \\
\text{Subscripts:} \\
p, q, r, V, a, \beta, \delta_a, \delta_c, \delta_e, \delta_p, \delta_1, \delta_2 \text{ partial derivatives with respect to the subscripted variable} \\
i, k \text{ } i^{th} \text{ and } k^{th} \text{ elements of vector or matrix} \\
L \text{ iteration number} \\
0 \text{ constant value} \\

A dot over a quantity denotes the time derivative of that quantity.

**PARAMETER ESTIMATION**

The problem considered is: Given a set of flight time histories of an aircraft's response variables, find the values of some unknown parameters in the system equations that best represent the actual aircraft response. An intuitive mathematical approach to this problem would be to minimize the difference between the flight response and the response computed from the system equations. This difference could be defined for each response variable as the integral of the error squared. These signal errors could then be multiplied by weighting factors and summed to obtain the total response error, thereby defining an integral squared error criterion.

A mathematically more precise formulation can be made in probabilistic terms. For each possible estimate of the unknown parameters, a probability that the aircraft response time histories take on the values actually observed can be defined. The estimates should be chosen so that this probability is maximized. This process is referred to as a maximum likelihood formulation of the problem. Maximum likelihood estimators have many desirable characteristics; for example, they yield asymptotically unbiased and consistent estimates. If the measurement noise is assumed to be Gaussian, white, stationary, and uncorrelated, this formulation is equivalent to a response error formulation, in which the weightings used are the inverse of the measurement noise covariance matrix.
To mathematically describe the maximum likelihood estimator it is first necessary to define the equations of motion for the aircraft system. These equations are:

\[ R \dot{x}(t) = Ax(t) + Bu(t) \]  
(1)

\[ y(t) = \begin{bmatrix} -\frac{1}{G} \end{bmatrix} x(t) + \begin{bmatrix} -\frac{\Phi}{H} \end{bmatrix} u(t) + \begin{bmatrix} -\frac{\Theta}{\nu} \end{bmatrix} \]  
(2)

\[ z(t) = y(t) + \eta(t) \]  
(3)

where

- \( x \) state vector
- \( u \) control vector
- \( v \) bias vector
- \( y \) computed observation vector
- \( z \) measured observation vector
- \( \eta \) noise vector

For the aircraft problem being considered, it is convenient to separate the equations of motion into longitudinal and lateral-directional sets. The linearized longitudinal equations are:

\[
\begin{bmatrix}
\frac{d}{dt} \dot{q} \\
\frac{d}{dt} \dot{\theta} \\
\theta
\end{bmatrix} = \begin{bmatrix} a \end{bmatrix} \begin{bmatrix} Z_a & 1 & Z_v & -\sin (\theta) \cos (\varphi) & \frac{g}{V} \\
M_a & M_q & M_V & 0 & q \\
0 & \cos (\varphi) & 0 & 0 & 0 \\
0 & 0 & \cos (\varphi) & 0 & 0 \end{bmatrix} \begin{bmatrix} Z_{\dot{\theta}} \\
Z_{\dot{\varphi}} \\
Z_{\dot{\alpha}} \\
Z_{\dot{\beta}} \end{bmatrix} + \begin{bmatrix} \delta_e \\
\delta_c \\
\delta_{\dot{\theta}} \\
\delta_{\dot{\varphi}} \end{bmatrix} \]  
(4)

\[ a_n = \frac{-V}{g} \left[ \dot{\alpha} - q + \sin (\theta) \cos (\varphi) \frac{g}{V} \right] + a_{n_{\text{bias}}} \]  
(5)

The linearized lateral-directional equations are:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & -\frac{L_2}{L_y} & 0 \\
0 & -\frac{L_2}{L_y} & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix} \begin{bmatrix} p \\
\dot{\beta} \\
\dot{\varphi} \\
\theta \end{bmatrix} = \begin{bmatrix} Y_{\beta} \\
Y_{\varphi} \\
Y_{\theta} \end{bmatrix} + \begin{bmatrix} \dot{Y}_{\beta} \\
\dot{Y}_{\varphi} \\
\dot{Y}_{\theta} \end{bmatrix} \begin{bmatrix} Y_{\beta} \\
Y_{\varphi} \\
Y_{\theta} \end{bmatrix} + \begin{bmatrix} \delta_x \\
\delta_y \\
\delta_{\dot{\beta}} \\
\delta_{\dot{\varphi}} \end{bmatrix} \]  
(6)

\[ a_y = \frac{V}{g} \left[ \beta - \sin (\alpha)p + \cos (\alpha)r - \cos (\varphi) \cos (\theta) \frac{g}{V} \varphi \right] + a_{y_{\text{bias}}} \]  
(7)
The unknown parameters are contained in the matrices $A$, $B$, $G$, and $H$ and in
the bias vector, $v$. For notational simplicity, the unknown parameters will be
regarded as forming a vector $c$. Then $A$, $B$, $G$, $H$, and $v$ are functions of $c$. There
is no provision for modeling state noise, that is, random or unknown inputs to the
system such as turbulence. (This problem is treated in reference 2.) Instead, it
is assumed that noise is introduced only in the measurement process. It is also
assumed that there is no noise in the control measurements.

The integral squared error criterion can now be expressed as finding the
vector of unknowns, $c$, that minimizes the cost functional:

$$ J = \frac{1}{T} \int_0^T \left[ z(t) - y(t) \right]^* D_1 \left[ z(t) - y(t) \right] dt $$

or as approximated in the discrete case:

$$ J = \frac{1}{(N - 1)} \sum_{i=1}^N \left( z_i - y_i \right)^* D_1 \left( z_i - y_i \right) $$

where $D_1$ is the symmetric, non-negative definite weighting matrix, $i$ is a time
index, and $N$ is the number of time points. The cost functional, $J$, can also be
called the index of performance or the fit error.

SOLUTION BY THE MODIFIED NEWTON-RAPHSON METHOD

Several algorithms for the minimization of nonlinear functionals exist that could
be used to minimize $J$. The modified Newton-Raphson method has proved to be the
most suitable for aircraft derivative determination, both in terms of computer time
and convergence properties.

The Newton-Raphson algorithm is an iterative method of functional minimization
which requires some initial estimate of $c$ and a means of computing the first and
second gradients of $J$ with respect to $c$. Revised estimates of $c$ are then obtained
from the equation

$$ c_L = c_{L-1} - \left( \nabla^2_c J \right)_L^{-1} \left( \nabla_c J \right)_L $$

where $L$ denotes the iteration number, $\nabla_c^2$ indicates the gradient with respect to $c$,
and $\nabla^2_c$ indicates the second gradient. The first and second gradients of $J$ are then

$$ \nabla_c J = \frac{2}{(N - 1)} \sum_{i=1}^N \left( z_i - y_i \right)^* D_1 \nabla_c \left( z_i - y_i \right) $$
\[ \nabla^2_{c_j} = \frac{2}{N-1} \sum_{i=1}^{N} \nabla_{c_i}(z_i - y_i)^* D_1 \nabla_{c_i}(z_i - y_i) + \frac{2}{(N-1)} \sum_{i=1}^{N} (z_i - y_i)^* D_1 \nabla^2_{c_i}(z_i - y_i) \] (12)

Computation of \( \nabla_{c_i}(z_i - y_i) \) is relatively straightforward, as described in reference 3. Computation of \( \nabla^2_{c_i}(z_i - y_i) \) is much more time consuming; however, Balakrishnan shows in reference 4 that the contribution of this term to the second gradient goes to zero as the process converges. Thus, if we neglect this term, the method is still an asymptotically unbiased estimator. The Newton-Raphson algorithm with this term neglected is referred to as the modified Newton-Raphson algorithm and provides the same result as obtained by quasilinearization.

Reference 1 describes a modification in the computation of the gradient that is used on the first iteration. This modification, analogous to linear least squares, helps to obtain convergence when the initial estimates are far from the minimum. With this modification it is often possible to start with estimates of zero for all the unknowns and still converge to the correct solution.

INCLUSION OF A PRIORI INFORMATION

Information from wind tunnel studies, previous flight tests, and other sources (referred to collectively as predicted derivatives) is often available on the values of some of the aircraft derivatives. It may be desirable to include this information in the program's algorithm. The use of this information is particularly important when there is a linear dependence or near dependence of the effect of several derivatives, for instance, in a maneuver in which the control motion is due largely or solely to a feedback of the states. The second gradient matrix then becomes ill-conditioned, resulting in poor convergence properties and unreliable estimates. In most instances a true minimum of the cost functional is still approached, despite the ill conditioning. The location of this minimum may not be important, however, because the linearly dependent derivatives could be altered greatly without significantly increasing the cost. In this instance the slight improvement in the fit obtained by altering the derivatives would not seem sufficient justification for altering them from the a priori values.

One solution to this problem would be to add to the cost functional a quadratic penalty function for departure from the a priori values. The cost functional, \( J \), would then be

\[ J = \frac{1}{(N-1)} \sum_{i=1}^{N} (z_i - y_i)^* D_1 (z_i - y_i) + (c - c_0)^* D_2 (c - c_0) \] (13)

where \( c_0 \) is the a priori estimate, and \( D_2 \) is a symmetric, non-negative definite
weighting matrix. The algorithm with this penalty function will be referred to as the modified maximum likelihood estimator. It is important in this formulation for the elements of $D_2$ to be small enough that, in general, $(c - c_0)^*D_2(c - c_0)$ is significantly less than \( \frac{1}{(N - 1)} \sum_{i=1}^{N} (z_i - y_i)^* D_1(z_i - y_i) \). Thus the estimates of those parameters that are well defined by the response data will not be altered.

The first and second gradients of $J$ now become

\[ \nabla_c^2 J = \frac{2}{(N - 1)} \sum_{i=1}^{N} (z_i - y_i)^* D_1 \nabla_c(z_i - y_i) + 2(c - c_0)^* D_2 \]  \( \text{(14)} \)

\[ \nabla_c^2 J = \frac{2}{(N - 1)} \sum_{i=1}^{N} \nabla_c(z_i - y_i)^* D_1 \nabla_c(z_i - y_i) + 2D_2 \]  \( \text{(15)} \)

where the second term of equation (12) has been neglected.

When this feature is used, convergence is generally improved. With small enough values of $D_2$, the estimates of the derivatives are not affected when the maneuver is well conditioned, but poorly conditioned maneuvers may converge and reveal some information instead of diverging.

CONFIDENCE LEVELS

One advantage of using a maximum likelihood estimator to determine aircraft stability and control derivatives is that an objective measure of the validity of the estimates is obtainable. With some other methods the main criterion of the validity of an estimate is the engineer's subjective judgment.

If the noise obeys the stated assumptions and $D_1$ is, in fact, the inverse of the noise covariance matrix, the Cramèr-Rao inequality (ref. 3) gives a lower bound on the covariance matrix of the estimates as follows:

\[ E \left[ (c - c_0)(c - c_0)^* \right] \geq \left[ \sum_{i=1}^{N} \nabla_c(z_i - y_i)^* D_1 \nabla_c(z_i - y_i) \right]^{-1} \]  \( \text{(16)} \)

The right side of this inequality is recognized as $(\nabla_c^2 J)^{-1}$ evaluated without the term for a priori. This expression is available in the minimization algorithm (eq. (12)), so these confidence levels (sometimes referred to in the literature as
is determined from the A matrix: LONG if $A(1,2) > +0.5$, LATR otherwise. This element is usually +1 in a longitudinal case or sin (ALPHA) in a lateral-directional case.

Items (2) to (11) are related to the input time histories. The signals which are input from the time histories fall into three classes: observations, controls, and extra. The observations form a vector, $z$, seven words long; the controls form a vector, $u$, four words long; and the extra signals form a vector four words long of quantities not actually used in the estimation process but useful in evaluating the quality of the maneuver. For a longitudinal case,

$$z = \begin{bmatrix} a & q & V & \theta & a_n & \dot{q} & a_x \end{bmatrix}^*$$  \hspace{1cm} (17)

$$u = \begin{bmatrix} \delta_e & \delta_c & \delta_1 & \delta_2 \end{bmatrix}^*$$  \hspace{1cm} (18)

Extra = $[\varphi \text{ Altitude Mach number } \overline{q}]^*$  \hspace{1cm} (19)

and for a lateral-directional case,

$$z = \begin{bmatrix} \beta & p & r & \varphi & a_y & \dot{\rho} & r \end{bmatrix}^*$$  \hspace{1cm} (20)

$$u = \begin{bmatrix} \delta_a & \delta_r & \delta_1 & \delta_2 \end{bmatrix}^*$$  \hspace{1cm} (21)

Extra = $[a \text{ V Mach number } \overline{q}]^*$  \hspace{1cm} (22)

(2) CARD, TAPE- (logical) – input source for time histories. Set either CARD = T or TAPE = T. Only one of the two variables can be set to true in the NAMELIST. Default condition is TAPE = T.

(3) SPS - sample rate of input time histories (samples per second). If SPS is not set, a default value is computed from the times shown on the time histories. The times of the first two data points are subtracted and the difference rounded to the nearest 5 milliseconds. The reciprocal of this value is then used as the default value for SPS.

(4) THIN- (integer) – thinning factor for input data. If THIN = 1, every point on the file is used; if THIN = 2, every second point is used, and so forth. SPS is the sampling rate of the data before this thinning. Default value of 1 is used.

(5) NCASE – number of disjoint maneuvers to be used in obtaining one set of estimates. If two or more maneuvers were performed at approximately the same flight condition, they may be processed together to obtain a single set of estimates. Each interval will be weighted by the number of time points in the interval. Default value of 1 is used.

(6) SCALE- (seven-word vector) – scale factor for observations. The observations are multiplied by corresponding elements of SCALE when read in to compensate for any scaling errors or sign changes. Default sets all elements of the vector to 1.0.
weighting matrix. The algorithm with this penalty function will be referred to as
the modified maximum likelihood estimator. It is important in this formulation for
the elements of \( D_2 \) to be small enough that, in general, \( (c - c_0)^* D_2 (c - c_0) \) is
significantly less than \( \frac{1}{(N-1)} \sum_{i=1}^{N} (z_i - y_i)^* D_1 (z_i - y_i) \). Thus the estimates of
those parameters that are well defined by the response data will not be altered.

The first and second gradients of \( J \) now become

\[
\nabla_c J = \frac{2}{(N-1)} \sum_{i=1}^{N} (z_i - y_i)^* D_1 \nabla_c (z_i - y_i) + 2(c - c_0)^* D_2
\]

(14)

\[
\nabla_c^2 J = \frac{2}{(N-1)} \sum_{i=1}^{N} \nabla_c (z_i - y_i)^* D_1 \nabla_c (z_i - y_i) + 2 D_2
\]

(15)

where the second term of equation (12) has been neglected.

When this feature is used, convergence is generally improved. With small
enough values of \( D_2 \), the estimates of the derivatives are not affected when the
maneuver is well conditioned, but poorly conditioned maneuvers may converge and
reveal some information instead of diverging.

CONFIDENCE LEVELS

One advantage of using a maximum likelihood estimator to determine aircraft
stability and control derivatives is that an objective measure of the validity of the
estimates is obtainable. With some other methods the main criterion of the validity
of an estimate is the engineer's subjective judgment.

If the noise obeys the stated assumptions and \( D_1 \) is, in fact, the inverse of the
noise covariance matrix, the Cramèr-Rao inequality (ref. 3) gives a lower bound
on the covariance matrix of the estimates as follows:

\[
E\left[ (c - c_0)(c - c_0)^* \right] \geq \left[ \sum_{i=1}^{N} \nabla_c (z_i - y_i)^* D_1 \nabla_c (z_i - y_i) \right]^{-1}
\]

(16)

The right side of this inequality is recognized as \( (\nabla_c^2 J)^{-1} \) evaluated without the
term for a priori. This expression is available in the minimization algorithm
(eq. (12)), so these confidence levels (sometimes referred to in the literature as
uncertainty levels) may be obtained with little additional effort. They can be useful in assessing the validity of the estimates obtained even when the noise characteristics are different from those assumed.

DESCRIPTION AND USE OF PROGRAMS

A basic computer program and two associated programs form a package that has been used at the NASA Flight Research Center to successfully analyze 1500 maneuvers from 20 aircraft. The basic program, referred to as the modified maximum likelihood estimation program, or MMLE, is designed to obtain maximum likelihood estimates from flight data. The associated programs, SETUP and SUMARY, although not directly related to the mathematical aspects of parameter estimation, have proved useful in extracting aircraft derivatives. The programs are designed to be used easily with the longitudinal and lateral equations of motion (eqs. (4) to (7)) by applying appropriate default values. For the options in the programs, the values designated as defaults are used only if no other values are specified. Each program is discussed in detail in the following sections.

In these programs a general matrix storage convention that permits flexibility and error checking is used. Each matrix is dimensioned with a fixed number of rows, MAX. The last row of the matrix, however, contains information about the matrix, instead of containing matrix elements. The first number in the last row is the number of rows of the matrix that are used; the second number is the number of columns used; and the third element is the matrix name in A format. For example, a 19 by 4 matrix called XJI could be stored in an array dimensioned 35 by 8 as:

This convention permits a variable-size matrix to be stored in an array of fixed dimension. The matrix manipulation subroutines can also check matrix compatibility by examining the last row before performing operations.

The programs use a standard matrix input format which facilitates data checking. The first card of any matrix to be input is a header card containing the name of the matrix, left-justified, in columns 1 to 4, the number of rows in the matrix, right-justified, in columns 9 to 10, and the number of columns in the matrix, right-justified, in columns 11 to 20. The body of the matrix follows, one row to a card, in an 8F10 format.

Additionally, the abbreviation T is used to denote true and F to denote false. NAMELIST variables follow the FORTRAN convention for type (names beginning with I, J, K, L, M, or N indicate integer variables; all other names indicate real variables), unless stated otherwise. Exceptions to this convention are given in parentheses after the NAMELIST variable.
The MMLE program can be run on most large modern computers with FORTRAN IV compilers. Approximately $31,000_{10}$ words of core storage are required. If overlay or segmentation is used, this requirement can be reduced to about $22,000_{10}$. Overlay and segmentation, however, are machine specific; directives for segmenting the MMLE program on the CDC OPERATING SYSTEM SCOPE 3,4 (ref. 5) are included in appendix A (p. 92) and can be used as a guide for implementation on other systems. Some form of automatic plotting equipment is desirable. The MMLE program plotting routines are written for a CalComp pen plotter (ref. 6). If other plotting equipment is used, it may be necessary to modify the plotting routines. The user must verify whether the routines supplied are compatible with the system being used.

From 4000 to 20,000 words of temporary disk storage are required, depending on the number of data points. This requirement is doubled if plots are made. A tape drive (two if plots are desired) may be substituted for disk storage.

Two types of input data are required for the MMLE program. The measured values contained in time histories of a flight maneuver must be available on cards, tape, or a disk file. These time histories are limited by dimensions in the plotting routines to 1000 time points per maneuver; these dimensions may be changed easily. In addition, the program must be provided information on the flight condition of the maneuver, values of pertinent characteristics of the aircraft, a set of starting estimates of the derivatives, and instructions controlling the activation of different program options.

Listings of the MMLE program and its subroutines are given in appendix A. A sample case is presented in appendix B.

Input Description

The inputs required for the MMLE program are described in this section. Each program option is explained immediately after the description of the input that controls the option.

**Title card.**—The title card contains any information needed to identify a particular set of data that is appropriate to include in the printed and plotted MMLE output. All 80 columns on this card may be used.

**NAMELIST/INPUT/**.— (See appropriate FORTRAN reference manuals for the format for specific machines.) The parameters included in the NAMELIST are as follows:

1. **LONG, LATR**—(logical) — type of aerodynamic mode to be analyzed. The mode type is indicated by LONG = T or LATR = T for longitudinal or lateral-directional, respectively. Only one type should be set. If neither is set, the type
is determined from the A matrix: LONG if \( A(1,2) > +0.5 \), LATR otherwise. This element is usually +1 in a longitudinal case or \( \sin (\alpha) \) in a lateral-directional case.

Items (2) to (11) are related to the input time histories. The signals which are input from the time histories fall into three classes: observations, controls, and extra. The observations form a vector, \( z \), seven words long; the controls form a vector, \( u \), four words long; and the extra signals form a vector four words long of quantities not actually used in the estimation process but useful in evaluating the quality of the maneuver. For a longitudinal case,

\[
z = \begin{bmatrix} a & q & V & \theta & a_n & \dot{q} & a_x \end{bmatrix}^* \tag{17}
\]

\[
u = \begin{bmatrix} \delta_e & \delta_c & \delta_1 & \delta_2 \end{bmatrix}^* \tag{18}
\]

\[
\text{Extra} = \begin{bmatrix} \varphi & \text{Altitude} & \text{Mach number} & \overline{q} \end{bmatrix}^* \tag{19}
\]

and for a lateral-directional case,

\[
z = \begin{bmatrix} \beta & p & r & \varphi & a_y & \dot{p} & \dot{r} \end{bmatrix}^* \tag{20}
\]

\[
u = \begin{bmatrix} \delta_a & \delta_r & \delta_1 & \delta_2 \end{bmatrix}^* \tag{21}
\]

\[
\text{Extra} = \begin{bmatrix} a & V & \text{Mach number} & \overline{q} \end{bmatrix}^* \tag{22}
\]

(2) CARD, TAPE- (logical) — input source for time histories. Set either CARD = T or TAPE = T. Only one of the two variables can be set to true in the NAMELIST. Default condition is TAPE = T.

(3) SPS — sample rate of input time histories (samples per second). If SPS is not set, a default value is computed from the times shown on the time histories. The times of the first two data points are subtracted and the difference rounded to the nearest 5 milliseconds. The reciprocal of this value is then used as the default value for SPS.

(4) THIN— (integer) — thinning factor for input data. If THIN = 1, every point on the file is used; if THIN = 2, every second point is used, and so forth. SPS is the sampling rate of the data before this thinning. Default value of 1 is used.

(5) NCASE — number of disjoint maneuvers to be used in obtaining one set of estimates. If two or more maneuvers were performed at approximately the same flight condition, they may be processed together to obtain a single set of estimates. Each interval will be weighted by the number of time points in the interval. Default value of 1 is used.

(6) SCALE— (seven-word vector) — scale factor for observations. The observations are multiplied by corresponding elements of SCALE when read in to compensate for any scaling errors or sign changes. Default sets all elements of the vector to 1.0.
(7) FIXED- (seven-word vector) — fixed biases for observations. The known biases are added to the corresponding observations after scaling (item (6)) but before any other operations with the data. Default sets all elements of the vector to 0.

(8) DC— (four-word vector) — known biases for controls. These biases are added to the corresponding controls before any operations with the controls. Default sets all elements of the vector to 0.

(9) NREC — number of data words in each record on input tape. This parameter has no meaning if card input is used. The total number of words in each record should be at least NREC + 4, because the first four words in the record contain the time (hours, minutes, seconds, milliseconds) and are not counted as data words. (See data file input section, p. 24.) NREC is limited by program dimensions to ≤100. Default value of 15 is used unless BOTH = T (item (11)); then the value of 25 is used instead.

(10) ORDER— (15-word integer vector) — location of desired signals on input tape. This parameter has no meaning if card input is used. The signals z, u, and extra are considered to form a single vector of signals, and ORDER describes a mapping of the data record from the tape onto this vector. The ith word in the resulting vector is set equal to the ORDER (i) data word in the tape record. (The first four words in the tape record contain the time and are not counted as data words.) The default is ORDER (i) = i for i = 1, 2, ..., 15, which implies that there is no reordering from the input tape to the program.

(11) BOTH— (logical) — special signal order with both longitudinal and lateral-directional data on the tape. This parameter has no meaning if card input is used. If BOTH = T, the input tape is assumed to contain all the data, both longitudinal and lateral-directional, in a specific order. This order is a, q, V, \( \dot{\theta} \), \( \ddot{a}_n \), \( \delta_e \), \( \delta_c \), \( \delta_1 \), \( \delta_2 \), \( \varphi \), altitude, Mach number, \( \bar{q} \), \( \beta \), p, r, \( a_{y} \), \( \dot{p} \), \( \dot{r} \), \( \delta_a \), \( \delta_r \), \( \delta_{1lateral} \), and \( \delta_{2lateral} \), where normally all angular measurements are in degrees, accelerations in g units, and velocities in feet per second. Also, if BOTH = T, NREC is overridden and set to 25; if the case is lateral-directional, the ORDER array is automatically set to \{16 17 18 12 19 20 21 22 23 24 25 1 3 14 15\}, which overrides any order that may have been read in. Thus if the tape has data in the proper order, BOTH may be set to T and the program will automatically pick off the appropriate signals for the type of case being analyzed. Default condition is F.

Items (12) to (18) specify the form of the plotted output.

(12) PLOTEM— (logical) — time history plots comparing measured and estimated response produced if PLOTEM = T. If PLOTEM = F, no plots are made. If the \textit{a priori} variation option (item (53)) is activated, the related derivative plots will be made instead. Default condition is T.

(13) PLTMAX — maximum error for plotting. If the error sum, \( J \), of the last or next to last iteration is greater than PLTMAX, time history plots are not made, even
if PLOTEM = T, to avoid exceeding reasonable plotter limits. Instead, the measured
time histories are printed to provide hints about the presumed problem. PLTMAX
may not be larger than ERRMAX (item (22)) or it will be set equal to ERRMAX by
the program. Default value of $1 \times 10^5$ is used.

(14) INCH- (logical) – plots scaled for inch grid paper if INCH = T; otherwise,
for centimeter grid paper. Default condition is F.

(15) ZMIN, ZMAX- (seven-word vectors) — minimum and maximum values on
vertical axis for plots comparing measured and estimated observations. The axes
are 4 centimeters long (2 inches if INCH = T). If corresponding elements of ZMIN
and ZMAX are equal for any signal, automatic scaling will be used on that signal.
Default values are all 0 (which implies that automatic scaling is used for the default,
since ZMIN = ZMAX).

(16) DCMIN, DCMAX- (eight-word vectors) – minimum and maximum values on
vertical axes for plots of controls and extra signals. The comments about ZMIN and
ZMAX (item (15)) apply. In addition, if automatic scaling is used for a signal and
there is no nonzero point on that signal, the plot of the signal will be omitted.
Default values of 0 are used.

(17) NC PLOT — number of controls and extra signals for plotting. Only the
first NC PLOT controls and extra signals will be plotted in addition to the observa-
tions. This option may be used to reduce plotting of data that may be extraneous
for some cases. The value of NC PLOT must be between 1 and 8, inclusive. Default
value of 8 is used.

(18) TIMES C — time scale for plots in seconds per centimeter (or seconds per
half inch if INCH = T). Default value of 1. is used.

(19) PRINT- (logical) – time histories based on measured data and final com-
puted time histories printed if PRINT = T. Default condition is F.

(20) TEST- (logical) – extra output printed each iteration if TEST = T to facili-
tate debugging. Extra output includes time histories (in radians), the transition
matrix (ref. 8) and its integral, and the first and second gradients of $J$. Default
condition is F.

(21) NOITER — number of iterations desired. NOITER = 0 is defined as a special
case for which the program computes the final time histories using initial estimates
of the unknown coefficients; that is, the parameter estimation step is omitted entirely.
The measured time histories are always printed when NOITER = 0, regardless of the
value of PRINT (item (19)). Default value of 6 is used.

(22) ERRMAX — maximum allowable error sum. If the error sum, $J$, at any time
becomes greater than ERRMAX, this is taken as an indication that the process is not
converging properly. Therefore, iteration will stop and the measured time histories
will be printed to provide clues to the reason for the problem. Default value of
$1 \times 10^{38}$ is used.
(23) BOUND – convergence bound. If the error sum, \( J \), in any iteration changes by less than \( \text{BOUND} \times \) times the error of the previous iteration, the process is assumed to have converged and iteration is stopped. Default value of 0.001 is used.

(24) PUNCH-(logical) – punched card output of nondimensional estimates. If PUNCH = T, the final estimates of the nondimensional derivatives are punched on cards along with the confidence levels obtained from the Cramér-Rao bound. Default condition is F.

(25) PUNCHD-(logical) – punched card output of dimensional estimates. If PUNCHD = T, the final dimensional A and B matrices are punched on cards. These cards can be used to restart the program from the final values. Default condition is F.

(26) NEAT – number of time reductions in computation of transition matrix, \( e^{A\Delta t} \), and its integral. In typical aircraft uses, a direct series evaluation of \( e^{A\Delta t} \) may become computationally unstable for sample rates less than about 10 samples per second. In such cases, the power series evaluation has been used to compute \( e^{A\tau} \) and its integral, with \( \tau = \frac{\Delta t}{2^{\text{NEAT}}} \). The desired transition matrices are then obtained after recursive applications of the formulas:

\[
\begin{align*}
e^{At} &= \left[ e^{(At)/2} \right]^2 \quad (23) \\
\int_0^t e^{As} ds &= \left[ e^{(At)/2} + I \right] \int_0^{t/2} e^{As} ds \quad (24)
\end{align*}
\]

This process provides improved computational stability without increased time or complexity. In general, NEAT should be large enough to make \( \tau \leq 0.05 \) second. NEAT = 0 implies direct series computation. Default value of 0 is used.

Items (27) to (48) are related to the geometry of the aircraft and the flight condition. Items (28) to (35) are required only if nondimensional derivatives are of interest. If these items are not entered, very large values of all nondimensional derivatives will be printed as a result of the default values to avoid accidental use of the meaningless nondimensional coefficients.

(27) METRIC-(logical) – unit designation for aircraft data. If METRIC = T, all units are standard SI (MKS) units (meter, kilogram, second); otherwise, U.S. Customary (EGS) units are assumed. Default condition is F. All input data units must be consistent with the system specified.

(28) GROSWT – aircraft gross weight (pounds or newtons). Default value of \( 1. \times 10^8 \) is used.
(29) IX-(real) – moment of inertia about the X-axis. This parameter is not needed for longitudinal cases (slug-ft^2 or kg-m^2). Default value of 1. × 10^9 is used.

(30) IY-(real) – moment of inertia about the Y-axis. This parameter is not needed for lateral-directional cases (slug-ft^2 or kg-m^2). Default value of 1. × 10^9 is used.

(31) IZ-(real) – moment of inertia about the Z-axis. This parameter is not needed for longitudinal cases (slug-ft^2 or kg-m^2). Default value of 1. × 10^9 is used.

(32) IXZ-(real) – cross-product of inertia between X- and Z-axes. This parameter is not needed for longitudinal cases (slug-ft^2 or kg-m^2). Default value of 0 is used.

(33) SPAN – wing span (ft or m). Default value of 0.001 is used.

(34) CBAR – reference chord (ft or m). Default value of 0.001 is used.

(35) S – reference wing area (ft^2 or m^2). Default value of 0.001 is used.

Items (36) to (42) concern instrument locations relative to the center of gravity. Angle-of-attack and angle-of-sideslip vane readings are corrected to the center of gravity by using the angular rates. The system model includes an arbitrary accelerometer location, so that accelerations need not be corrected to the center of gravity. The longitudinal axis locations are positive for instruments forward of the center of gravity, and the normal axis locations are positive for instruments below the center of gravity. All values are in feet or meters, and a default value of 0 is used.

(36) XB – location of angle-of-sideslip vane along the longitudinal axis.

(37) XALF – location of angle-of-attack vane along the longitudinal axis.

(38) ZB – location of angle-of-sideslip vane along the normal axis.

(39) XAY – location of \( a_y \) accelerometer along the longitudinal axis.

(40) ZAY – location of \( a_y \) accelerometer along the normal axis.

(41) XAN – location of \( a_n \) accelerometer along the longitudinal axis.

(42) ZAX – location of \( a_n \) accelerometer along the normal axis.

Items (43) to (46) are not used in the estimation process, but are useful for identifying the flight condition of the maneuver. They are passed to the SUMMARY program for plot identification purposes.

(43) CG – aircraft center of gravity in fraction of chord. Default value of 0.25 is used.

16
(44) MACH-(real) — average Mach number. If 0, this parameter will be obtained from the input time history. Default value of 0 is used.

(45) ALPHA — average angle of attack. If 999, this parameter will be obtained from the input time history. Default value of 999 is used.

(46) PARAM — any other parameter that might be used to distinguish between flight conditions. PARAM may be used as flap position or wing sweep. Default value of 0 is used.

(47) Q — average dynamic pressure. If 0, this parameter is obtained from the input time history (lb/ft$^2$ or N/m$^2$). Default value of 0 is used.

(48) V — average velocity. If 0, this parameter is obtained from the input time history (ft/sec or m/sec). Default value of 0 is used.

(49) VAR-(three-word logical vector) — option that controls variable bias. The fifth to seventh signals of the observation vector have an unknown bias that is included in the system model. (See p. 12 for the elements of the observation vector.) This bias is determined if the corresponding elements of VAR are T. The initial values of these variable biases are 0, except for the $a_n$ bias in a longitudinal case, which starts with a value of 1. The bias on a signal that has a D1 weighting of 0 cannot be determined; therefore, any attempt to determine a bias for an unweighted signal will be overridden in the program. Default sets all elements of the vector to T.

(50) ZERO-(four-word logical vector) — option that requires the program to determine variable initial condition. For each element of ZERO that is T, a variable increment to the initial condition is determined for the corresponding state. This increment is added to the measured initial condition to obtain the initial condition used for the computed data. If the variable initial condition is used in conjunction with NCASE > 1 (item (5)), the same increment from the measured value is used for each maneuver in the case. Default sets all elements of the vector to F.

(51) ND1, D1RLX, D1TOL — parameters that affect diagonal D1 determination option. This puts the program into a different mode of operation. A D1 weighting matrix (see matrix input section) should be determined for each airplane at the beginning of its flight program. This option automatically determines the diagonal elements of the D1 matrix based on a particular case and is activated if ND1 > 0. The program executes one run with the initial D1 matrix (described on p. 23) input, or its default, and then applies a simple iterative algorithm ND1 times to determine the proper D1 matrix. Each iteration of this algorithm involves another run through the estimation loop to obtain a set of weighted relative errors $E_k^2 = \frac{D1_{kk}}{t} \int_0^t \left[ z_k(t) - y_k(t) \right]^2 dt$.

The algorithm is designed to find a D1 matrix that results in the weighted error being approximately 1 on each signal being used (as indicated by a nonzero initial estimate of the corresponding D1 element). The motivation for this procedure is discussed in reference 3. The revised estimate of each diagonal element of the D1 matrix is then produced by multiplying the previous estimate by a factor that depends on the previous weighted error of that signal, $E_k$, and a relaxation factor,
DIRLX. If $E_k \geq 1$, the factor is $\frac{1}{(E_k - 1)\text{DIRLX} + 1}$; and if $E_k < 1$, the factor is $\left(\frac{1}{E_k} - 1\right)\text{DIRLX} + 1$. The variable $\text{D1TOL}$ will stop this process if the process has converged before $\text{ND1}$ iterations. If, after any iteration, none of the weighted errors are greater than $\text{D1TOL}$ or less than $\frac{1}{\text{D1TOL}}$, a final iteration will be run, and the process will be stopped. The parameter $\text{WMAPR}$ (item (52)) will be set to 0 if this option is used, regardless of the MMLE program's input value. If plotting was specified (item (12)), only the time history using the final D1 vector will be plotted. If both the D1 vector determination and the a priori variation (item (53)) are activated, the D1 vector will be determined first, and the a priori variation will use the final D1 matrix. Default values used are $\text{ND1} = 0$, $\text{DIRLX} = 1.2$, and $\text{D1TOL} = 1.4$.

Items (52) and (53) are related to the a priori feature.

(52) $\text{WMAPR}$ -- overall weighting factor for a priori information. Each element in the a priori weighting matrices $\text{APRA}$ and $\text{APRB}$ (see matrix input section) is multiplied by WMAPR before use. A value of 0 implies that the a priori feature is not used in the estimation process. Default value of 0 is used.

(53) $\text{NAPR}$, $\text{WFAC}$ -- parameters that control a priori variation option which puts the program into a different mode. If the a priori feature is used, a set of a priori weighting matrices should be selected at the beginning of the flight program for each aircraft analyzed. In determining the best weighting matrices to use, it is useful to run the same case with several values of WMAPR (item (52)). Reference 3 describes this process. The option to vary the value of WMAPR is activated if $\text{NAPR}$ is greater than 0. The program then runs the entire case a total of $\text{NAPR}$ times with different values of WMAPR. The first run is with $\text{WMAPR} = 0$, and the second run is with the value specified for WMAPR by item (52) (if 0 was specified, 0.001 is used instead). For each subsequent run, the value of WMAPR used is WFAC times the value used on the previous run. Time history plots are never produced when this option is used; instead, if $\text{PLOTEM} = T$ (item (12)), the final estimates of each of the derivatives are plotted versus WMAPR on a logarithmic scale. The a priori estimates, which may be considered as the estimates obtained as WMAPR approaches infinity, are also plotted to the right of the other estimates. These plots may then be used as described in references 3 and 8 to estimate the best values to use for the a priori weightings. For these plots to be correct, the NAMELIST variable PUNCH (item (24)) must equal F, because of the order in which the computations are performed. Default values of WFAC = 100. and $\text{NAPR} = 0$ are used.

Time cards. -- For each of the NCASE (NAMELIST item (5)) time segments to be included, one time card is required. The time cards contain the start and end times for the segment expressed as hours, minutes, seconds, and milliseconds in the format (2(312,13,1X)). The program starts the segment at the first time point greater than or equal to the start time and stops it at the last point less than or equal to the stop time.

Matrix input. -- Several input matrices are read next in a standard matrix input format. The matrices may be read in any order. Only the A and B matrices must be read in; the others may be read in if the default values are to be changed.
A matrix (4 by 4): The A matrix is the starting estimate of the stability matrix. For a longitudinal three-degree-of-freedom case it should be:

\[
\begin{bmatrix}
Z_a & 1 & Z \nu & -\sin(\theta) \cos(\varphi) \frac{q}{V} \\
M_a & M_q & M \nu & 0 \\
X_a & 0 & X \nu & -\cos(\theta) g \\
0 & 0 & 0 & 0
\end{bmatrix}
\]

In a two-degree-of-freedom case the third column should be set to 0. For a lateral-directional case the A matrix should be:

\[
\begin{bmatrix}
Y_\beta & \sin(\alpha) & -\cos(\alpha) & \cos(\varphi) \cos(\theta) \frac{q}{V} \\
L_\beta & L_\rho & L_\kappa & 0 \\
N_\beta & N_\rho & N_\kappa & 0 \\
0 & 1 & \cos(\varphi) \tan(\theta) & 0
\end{bmatrix}
\]

Average values of \( \alpha, \theta, \varphi, \) and \( V \) are used in these matrices.

B matrix (4 by 5 to 4 by 8): The B matrix is the starting estimate of the control matrix. The first four columns are for the control derivatives; the fifth column contains aerodynamic biases (treated as control derivatives, in which the control is defined as a constant value of 1 radian). Usually, only these five columns are required. If NCASE is greater than 1, independent aerodynamic biases may be determined for up to the first four maneuvers when necessitated by trim changes or other factors. In this event, the fifth column's aerodynamic biases are included in every maneuver, the sixth column's biases are included in all maneuvers after the first, the seventh column's biases are included in all maneuvers after the second, and the eighth column's biases are included in all maneuvers after the third. Thus the total aerodynamic bias on the first maneuver would be in column 5; for the bias on the second maneuver, columns 5 and 6 would be added; for the third maneuver, columns 5, 6, and 7 would be added; and for the fourth and all subsequent maneuvers, columns 5, 6, 7, and 8 would be added. For a lateral-directional case the B matrix should then be:

\[
\begin{bmatrix}
Y_\delta_a & Y_\delta_r & Y_\delta_1 & Y_\delta_2 & Y_0 & Y_{02} & Y_{03} & Y_{04} \\
L_\delta_a & L_\delta_r & L_\delta_1 & L_\delta_2 & L_0 & L_{02} & L_{03} & L_{04} \\
N_\delta_a & N_\delta_r & N_\delta_1 & N_\delta_2 & N_0 & N_{02} & N_{03} & N_{04} \\
0 & 0 & 0 & 0 & \varphi_0 & \varphi_{02} & \varphi_{03} & \varphi_{04}
\end{bmatrix}
\]
For a longitudinal case the B matrix would be:

\[
\begin{bmatrix}
Z_\theta e & Z_\theta c & Z_\theta 1 & Z_\theta 2 & Z_0 & Z_{02} & Z_{03} & Z_{04} \\
M_\theta e & M_\theta c & M_\theta 1 & M_\theta 2 & M_0 & M_{02} & M_{03} & M_{04} \\
X_\theta e & X_\theta c & X_\theta 1 & X_\theta 2 & X_0 & X_{02} & X_{03} & X_{04} \\
0 & 0 & 0 & 0 & \dot{\theta}_0 & \dot{\theta}_{02} & \dot{\theta}_{03} & \dot{\theta}_{04}
\end{bmatrix}
\]

AA array (4 by 4): The AA array defines which elements in the A matrix are to be determined by the program. Each element in the AA array should be either 1. or 0. A 1. implies that the corresponding element in the A matrix will be estimated by the program, whereas a 0. implies that it will be held fixed at the starting value. If not read in, the AA array has the following default:

Longitudinal –

\[
\begin{bmatrix}
1. & 0. & 0. & 0. \\
1. & 1. & 0. & 0. \\
0. & 0. & 0. & 0. \\
0. & 0. & 0. & 0.
\end{bmatrix}
\]

Lateral-directional –

\[
\begin{bmatrix}
1. & 1. & 0. & 0. \\
1. & 1. & 1. & 0. \\
0. & 0. & 0. & 0. \\
0. & 0. & 0. & 0.
\end{bmatrix}
\]

BB array (4 by 5 to 4 by 8): The BB array defines which elements in the B matrix are to be determined in the same manner as the AA array defines those in the A matrix. If not read in, the BB array defaults to:

Longitudinal –

\[
\begin{bmatrix}
1. & 0. & 0. & 0. & 1. \\
1. & 0. & 0. & 0. & 1. \\
0. & 0. & 0. & 0. & 0. \\
0. & 0. & 0. & 0. & 1.
\end{bmatrix}
\]

Lateral-directional –

\[
\begin{bmatrix}
1. & 1. & 0. & 0. & 1. \\
1. & 1. & 0. & 0. & 1. \\
1. & 1. & 0. & 0. & 1. \\
0. & 0. & 0. & 0. & 1.
\end{bmatrix}
\]

AR matrix (4 by 4): The AR matrix is the a priori stability matrix and contains the a priori value of the A matrix. If the a priori feature is used, the estimates are weighted toward the AR matrix values. In general, the a priori values and the starting values are the same, but it is possible to distinguish between them. If not read in, the AR matrix is set equal to the A matrix.

BR matrix (4 by 5 to 4 by 8): The BR matrix is the a priori control matrix and plays a role similar to that of the AR matrix. If not read in, it is set equal to the B matrix.
APRA matrix (4 by 4): The APRA matrix contains *a priori* weightings for the stability matrix and contains the weightings to be assigned to the elements of the AR matrix for the *a priori* option. The program multiplies each relevant element in the APRA matrix by the overall weighting factor, WMAPR (NAMELIST item (52)), and assigns it an appropriate diagonal location in the D2 matrix (eq. (13)). No provision is made for the input of off-diagonal elements of the D2 matrix, although they are provided for in the algorithm. If not read in, the APRA matrix defaults to:

Longitudinal -

\[
\begin{bmatrix}
13000 & 0 & 0 & 0 \\
15 & 800 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

Lateral-directional -

\[
\begin{bmatrix}
13000 & 13000 & 13000 & 0 \\
0.15 & 500 & 5 & 0 \\
15 & 800 & 800 & 0 \\
0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

APRB matrix (4 by 5 to 4 by 8): The APRB matrix contains *a priori* weightings for the control matrix and plays a role analogous to that of the APRA matrix. If not read in, the APRB matrix defaults to:

Longitudinal -

\[
\begin{bmatrix}
13000 & 13000 & 13000 & 13000 & 0 & 0 & 0 & 0 \\
15 & 15 & 15 & 15 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

Lateral-directional -

\[
\begin{bmatrix}
13000 & 13000 & 13000 & 13000 & 0 & 0 & 0 & 0 \\
0.15 & 0.15 & 0.15 & 0.15 & 0 & 0 & 0 & 0 \\
15 & 15 & 15 & 15 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

AP array (3 by 4): The AP array is used in the formation of the observation matrix G of equation (2). For the aircraft identification problem, the observations generally available are either elements of the state vector, their derivatives, or accelerations. If only states and their derivatives are available, the G matrix would be identical to the A matrix. When accelerations are also of interest, the G matrix may still be expressed as a simple function of the A matrix; this function
is specified by the AP array. For example, consider the two-degree-of-freedom longitudinal case:

\[
\dot{a} = \frac{V}{g} a + q - \frac{g}{V} \cos (\varphi) \sin (\theta) + z_e \delta_e + \left( z_0 + \frac{g}{V} \cos \varphi \cos \theta \right)
\]  

(25)

\[
\left(a_n - a_{n_{bias}}\right) = -\frac{V}{g} a + 0q - 0\theta + \left( -\frac{V}{g} \right) z_e \delta_e - \frac{V}{g} \left( z_0 + \frac{g}{V} \cos \varphi \cos \theta \right)
\]  

(26)

where

\[a_{n_{bias}} = -\cos \varphi \cos \theta + a_{n_{instrument \ bias}}\]

From this example it can be seen that \(a_n - a_{n_{bias}}\) can be computed like \(\dot{a}\) if appropriate terms are simply multiplied by constant values of \(-\frac{V}{g}\) or 0. Thus each element in the G matrix can be defined as the product of the corresponding element in \(R^{-1}A\) and a constant. These constants form the AP array. This formulation results in a considerable saving of computer time. It should be noted that the accelerometer offsets from the center of gravity (NAMELIST items (39) to (42)) add terms to the G matrix after the basic terms are computed from the AP array. If the AP array is read in, the BP array must also be read in. If not read in, the AP array defaults to the following standard forms:

**Longitudinal**

\[
\begin{bmatrix}
-\frac{V}{g} & 0 & 0 & 0 \\
1 & 1 & 1 & 1 \\
\frac{1}{g} & 0 & 0 & 0
\end{bmatrix}
\]

**Lateral-directional**

\[
\begin{bmatrix}
\frac{V}{g} & 0 & 0 & 0 \\
1 & 1 & 1 & 1 \\
\frac{1}{g} & 1 & 1 & 1
\end{bmatrix}
\]

**BP array (3 by 5 to 3 by 8):** The BP array plays a role analogous to that of the AP array. It defines the H matrix of equation (2) as a function of the B matrix. Each element in the H matrix is defined as the product of the corresponding elements in \(R^{-1}B\) and the BP array. As in the G matrix, accelerometer offsets from the center of gravity may cause additional terms to be added to the basic H matrix. If either the AP or the BP array is read in, both must be read in. The BP array defaults to:

**Longitudinal**

\[
\begin{bmatrix}
-\frac{V}{g} & -\frac{V}{g} & -\frac{V}{g} & -\frac{V}{g} & -\frac{V}{g} & -\frac{V}{g} & -\frac{V}{g} & -\frac{V}{g} \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\frac{1}{g} & \frac{1}{g} & \frac{1}{g} & \frac{1}{g} & \frac{1}{g} & \frac{1}{g} & \frac{1}{g} & \frac{1}{g}
\end{bmatrix}
\]
Lateral-directional --

$$\begin{bmatrix}
\frac{V}{g} & \frac{V}{g} & \frac{V}{g} & \frac{V}{g} \\
1. & 1. & 1. & 1. \\
1. & 1. & 1. & 1. \\
\end{bmatrix}$$

R matrix (4 by 4): The R matrix is an acceleration transformation matrix. If not read in, it defaults to the unit matrix for longitudinal cases, or for lateral-directional cases to:

$$\begin{bmatrix}
1. & 0. & 0. & 0. \\
0. & 1. & \frac{1}{T_X} & 0. \\
0. & \frac{1}{T_Z} & 1. & 0. \\
0. & 0. & 0. & 1. \\
\end{bmatrix}$$

D1 matrix (5 by 5 to 7 by 7): The D1 matrix is the signal weighting matrix. The diagonal elements are the weightings used for each response signal in the cost functional. The size of this matrix determines the number of signals used in the analysis; therefore, if \( \dot{p} \) and \( \dot{r} \) are not measured for a lateral-directional case, the D1 matrix should be 5 by 5. This reduction will save a significant amount of computer time. If the D1 matrix is diagonal, it should be entered as a vector containing the diagonal elements. The program will then recognize that the matrix is diagonal and take advantage of this in its computations. A vector is indicated by a header card with 0 for the number of columns. The vector is then entered on one card in an 8F10 format. If not read in, the D1 matrix is assumed to be diagonal with the following values:

Longitudinal –

$$\begin{bmatrix}
30000. & 200000. & 0. & 100000. & 2000. \\
\end{bmatrix}$$

Lateral –

$$\begin{bmatrix}
500000. & 1500. & 1000000. & 30000. & 5000. \\
\end{bmatrix}$$

ENDCASE. – The end of the matrix input is signaled by a card with ENDCASE starting in column 1. If no more cases follow, this card should have simply END instead of ENDCASE.

Card input. – If card input was specified, the input time histories are necessary here. For each time point there should be a record of two cards containing four time words (hours, minutes, seconds, milliseconds), seven observations, four controls, and four extra signals. The order of these quantities is given in NAMELIST item (11). The format is (3I2, I4, 7F10/8F10). Normally, the angular measurements are in degrees, the accelerations in \( g \) units, and the velocities in feet per second.
Data file input. — If tape input was specified, the time histories must be on an unformatted data file (either tape or disk). The device number of this file should be specified as 4 by the control cards. This file by default has 4 time words plus 15 data words per record, as in the card input. The length of the records on this file and the order of the parameters (except for the time words) may be changed by the use of the NREC and ORDER parameters (NAMELIST items (9) and (10)); alternatively, the file may be specified to be in the special BOTH form (NAMELIST item (11)). Normally, the angular measurements are in degrees, the accelerations in \( g \) units, and the velocities in feet per second.

Output Description

Three basic forms of output are available from the MMLE program: printed, plotted (time history or derivative plots), and punched card.

Printed output. — The three levels of printed output are controlled by the parameters PRINT and TEST (NAMELIST items (19) and (20)). The basic output is always printed. If PRINT = T, measured and final computed time histories are also printed. If, in addition, TEST = T, time histories in radians, the transition matrices, and the first and second gradients of \( J \) are printed every iteration. The TEST parameter is generally used only for program debugging.

Appendix B presents a listing for a sample case with only the basic output. The first page (p. 112) of the output listing summarizes the input options chosen, and the second page lists the matrices read in. The dimensional and nondimensional starting values are then summarized. An asterisk indicates the values held fixed; the other values are to be determined as unknowns in the program. Each iteration includes a printout of the revised A and B matrices, the integral squared error on each input signal, the weighted errors on each signal, and the total error sum. This iterative loop may terminate in three ways. If the error sum exceeds ERRMAX (NAMELIST item (22)) at any time, the iteration will stop immediately and the input time history will be printed (not included in appendix B). If the maximum number of iterations is reached or the process converges within the range defined by BOUND (NAMELIST item (23)), normal termination will occur. The message "ITERATION TERMINATED, ERROR WITHIN .00100 BOUND" indicates that the convergence bound caused termination in the sample case.

Confidence levels in dimensional and nondimensional form are listed next. These confidence levels are analogous to the standard deviation. Their magnitude indicates the relative confidence to be placed in estimates of the same coefficient from different maneuvers. A small confidence level for a particular derivative estimate indicates that the estimate of the derivative should be very good. Confidence levels are useful in fairing estimated derivative values.

The final page (p. 117) of the first case is a summary of the converged values. The final dimensional and nondimensional derivatives are printed in the same format as the starting values, followed by the final A and B matrices. The final integral squared errors, weighted errors, and total error sum are printed, followed by a summary of the convergence of the error sum.
If either the D1 determination option (NAMELIST item (51)) or the *a priori* variation option (NAMELIST item (53)) is activated, the program prints an appropriate message at this point and begins its second pass through the estimation loop. The output resumes from the top of the third page. This output pattern would be repeated as many times as specified by the option. If more cases follow, the same output pattern is repeated for each case.

*Plotted output.*—If plotting is invoked (NAMELIST item (12)), time history plots like those in appendix B will be produced. On the observation signals, the solid lines represent the flight data and the dotted lines are the computed fits. When plotting is invoked and the *a priori* variation option (NAMELIST item (53)) is being used, time history plots are not produced, but, instead, the derivative plots discussed under that option (not included in appendix B).

*Punched card output.*—If PUNCH = T (NAMELIST item (24)), the nondimensional A and B matrices and confidence levels are punched on cards. These cards are preceded by a header card which contains the characters LATR or LONG followed by the first 35 characters of the title card and the values of MACH, ALPHA, PARAM, and CG. These cards are in the exact format required for the SUMARY plotting program. If the case is longitudinal, a computed $\delta_e$ appears in the matrix location for $C_{m0}$, and $C_Z$ appears in the location for $C_{Z0}$. These quantities are of more interest in this form, although the confidence levels are not readily available. (The confidence levels punched are those for the original $C_{m0}$ and $C_{Z0}$.) The equations used to compute these parameters are:

$$\delta_{e_{trim}} = \frac{(C_m + C_{m0})}{C_{m0}}$$

$$C_Z = C_{Z0} + C_{Z0} \alpha + C_{Z0} \delta_e \delta_{e_{trim}} - \cos (\theta) \frac{W}{qS}$$

These equations are valid only for a two-degree-of-freedom case with no lateral-directional cross-coupling terms.

The final dimensional A and B matrices may be output on punched cards if PUNCHD = T (NAMELIST item (25)) is specified. These matrices may be used if it is desired to restart a case from the final values and run additional iterations. If the *a priori* feature is used in the restart, the original A and B matrices should be relabeled AR and BR and inserted (see discussion of AR and BR matrices, pp. 20-21) because the *a priori* values would be different from the new starting values. Any variable bias from the original run should also be subtracted from the data using FIXED (NAMELIST item (7)) in order to start at the same values as the final iteration of the previous run.
One of the most time-consuming portions of the analysis of aircraft stability and control derivatives is the preparation of input data for the derivative estimation program. The preprocessing program, SETUP, automates much of this work and is a key element in the routine processing of a large number of cases. It can produce, at the user's option, the data file and the punched input deck for the MMLE program. Listings of the program and its subroutines are presented in appendix C. A sample case is included in appendix D.

The SETUP program reads a set of predicted derivatives to be interpolated and dimensionalized for the given flight condition. The flight condition may be specified by the user, or if appropriate data were recorded on a flight tape, the program can obtain the flight condition automatically, given only the start and stop times for the case.

When the program is used in the most automated manner, the only inputs required for each case are the start and stop times, the type of case (longitudinal or lateral-directional), and an indication of which controls were used for the particular maneuver if more than one control is relevant. Using the program in this manner requires some preparation, but only at the beginning of the flight program rather than for each case. This distinction is important when several hundred cases are being analyzed, as has been done on several aircraft.

In preparation for the most automated use of the SETUP program, the user must write four small FORTRAN subroutines. Subroutine TAPEIN reads a flight data tape, finds the time interval requested, and places the data and times from the data into two arrays. The sample included in appendix D reads an unformatted tape with time in the first four words. Subroutine RDSET provides any initialization needed for TAPEIN; in the sample case it reads the number of channels on the input tape and the channel numbers of the data needed. Subroutine COND obtains the flight condition if it is to be computed automatically instead of read in manually. The averages of each of the data channels read in are available for use in this subroutine, and the subroutine can compute the required parameters from these averages. The sample obtains ALPHA, THETA, PHI, DETRIM, Q, V, and MACH from the data channel averages. The subroutine will also compute Q and V from knots indicated airspeed and altitude, if preferred. Weight, inertia, and center of gravity are not computed in the example subroutine, although they may be computed in user-supplied versions. Subroutine COND1 reads in any data needed in subroutine COND, for instance, tables of inertia versus fuel weight. This subroutine, as given in appendix C, is a null subroutine.

Input Description

The input data and the case specifications are described in the following sections.

Options.— The options to be used are specified by the following cards. All the options begin in column 1. The cards may appear in any order (except for the START card, as noted). Only the first four characters of each card are checked.
WRITE TAPE — instructs the program to write a data file for the MMLE program. This option automatically invokes the READ TAPE option.

PUNCH DECK — instructs the program to punch a data deck for the MMLE program.

READ TAPE — instructs the program to read an input tape. This option might be specified if input tape data are needed to determine the flight condition for the punched data card deck. This instruction is redundant if WRITE TAPE was specified.

START — signals the end of the options and the start of processing. This card must be the last card in the options section.

Vehicle characteristics.— The input segment that starts here and ends at, but does not include, User-supplied data (p. 30) is required if PUNCH DECK was specified in the preceding options. If PUNCH DECK was not specified, this segment must not be included.

NAMELIST/WIND/: The following parameters may be input in NAMELIST format:

1. NABP — number of angle-of-attack breakpoints for predicted derivatives. Default value of 1 is used.

2. NMBP — number of Mach number breakpoints for predicted derivatives. Default value of 1 is used.

3. NBP — number of sets of predicted derivatives. Each set is identified subsequently as either lateral-directional or longitudinal and as having a particular value of the extra identifying parameter PARAM (used if the data are to be separated by some other criterion, such as wing sweep or flap position). Thus if there is one longitudinal and one lateral-directional data set and no additional distinction is made, NBP = 2. Dimensions in the program restrict the value of NBP to less than or equal to 8. Default value of 1 is used.

4. LATR, LONG—(eight-word logical vectors) — parameters that specify dynamic modes of the predicted derivatives. The type of each set of predicted derivatives should be specified by setting the corresponding element of either LATR or LONG to true. Only one of the variables can be set to true in the NAMELIST. Default type for each set is longitudinal.

5. NCLA, NCLO — number of coefficients per lateral-directional and longitudinal data set, respectively.

6. CGLA, CGLO — reference center of gravity for lateral-directional and longitudinal predicted derivatives in fraction of reference chord. Default value of 0.25 is used.

7. MZLA, MZLO — number of signals for the MMLE program to analyze in lateral-directional and longitudinal cases (that is, the length of the D1 vector; see D1 matrix description, p. 23). The values must be between 5 and 7, inclusive. Default value of 5 is used.
(8) WMLA, WMLO -- overall a priori weighting for lateral-directional and longitudinal cases (WMAPR in MMLE program). If WMLA or WMLO are not entered, the SETUP program will not read the appropriate APRA and APRB matrices discussed subsequently and will use a weighting of 0. If WMLA or WMLO are set to 0, the corresponding APRA and APRB matrices will be read by the SETUP program and punched with the MMLE program card deck, although the weighting on the matrices will still be 0. If WMLA or WMLO is set to a positive value, the APRA and APRB matrices will be read and punched normally. If WMLA or WMLO is set to a negative value, the APRA and APRB matrices will not be read and the absolute value of WMLA or WMLO will be passed to the MMLE program (using the MMLE program's defaults for the APRA and APRB matrices). In all these cases, the lateral-directional usage and longitudinal usage are independent.

(9) DEG, RAD-(logical) -- parameters that specify degrees or radians for units of predicted derivatives by setting either DEG = T or RAD = T. Only one of the two variables can be set to true in the NAMELIST. The rotary derivatives are per radian regardless of this option. Default condition is DEG = T.

(10) METRIC-(logical) - parameter that specifies SI (MKS) units if true and U.S. Customary (EGS) units if false. All input data units must be consistent with the system specified. Default condition is F.

(11) BODY, STAB- (logical) -- parameters that specify axis system of longitudinal predicted derivatives as body or stability. Only one of the two variables can be set to true in the NAMELIST. (Lateral-directional data are in the body axes system independent of this option.) Default condition is STAB = T.

(12) S -- value of reference wing area (ft² or m²).

(13) SPAN -- value of reference wing span (ft or m).

(14) CBAR -- value of reference wing chord (ft or m).

(15) SPS -- samples per second for data file. If not specified, 0 is passed to the MMLE program which then, by default, determines SPS from the times on the data file.

(16) PUNCH-(logical) -- option passed to the MMLE program to control its PUNCH (MMLE NAMELIST item (24)) option to punch cards with final estimates of the nondimensional derivatives and confidence levels. Default condition is F.

(17) XB, XALF, ZB, XAY, ZAY, XAN, ZAX -- instrument locations relative to the center of gravity. The meaning of each of these parameters is the same as that given in items (36) to (42) of the MMLE NAMELIST except that, as used here, these values refer to the reference center of gravity for the predicted derivatives instead of the flight center of gravity. If 0 is entered, it is assumed that the signals have been corrected to the flight center of gravity, and no additional correction terms will be used. Default value of 0 is used.

Vehicle name: The vehicle name is specified by up to eight characters, starting in column 1. These eight characters will be used on the title card punched out for
the MMLE deck and will be included on the first line of the output from the SETUP program.

Lateral-directional weighting matrix: The lateral-directional D1 matrix is read in as a vector on one card in a 7F10 format. This vector is omitted if no lateral-directional predicted derivatives are read in. If every element is 0, the default in the MMLE program will be used.

Longitudinal weighting matrix: The longitudinal D1 matrix is read in as a vector. The comments for the lateral-directional D1 vector apply.

Lateral-directional APRA and APRB matrices: The APRA and APRB matrices for lateral-directional cases are in standard matrix input format. As mentioned above, these matrices are omitted if the WMLA parameter was not read in or was set to a negative value.

Longitudinal APRA and APRB matrices: The APRA and APRB matrices for longitudinal cases are in standard matrix format. The matrices are omitted if the WMLO parameter was not specified or was negative.

Predicted derivatives: NBP sets of predicted derivatives are required in the order specified in item (4) of SETUP NAMELIST/WIND/. Each set consists of data for NCLA or NCLO coefficients, depending on whether the set is lateral-directional or longitudinal. The data for each coefficient may be read as a function of Mach number and angle of attack, or as a function of Mach number only in the following forms.

The data for each coefficient begin with a header card containing the coefficient name in the first four columns and either a 1 or a 2 in column 10; a 1 indicates that the coefficient is a function of Mach number only, a 2 indicates that it is a function of Mach number and angle of attack. The only acceptable coefficient names are: lateral-directional – CYB, CLB, CNB, CLP, CNP, CLR, CNR, CYDA, CLDA, CNDA, CYDR, CLDR, CNDR, CYD1, CLD1, CND1, CYD2, CLD2, CND2; longitudinal (body axes) – CNA, CMA, CAA, CMQ, CNV, CMV, CAV, CNDE, CMDE, CADE, CNDC, CMDC, CADC, CND1, CMD1, CAD1, CMD2, CAD2, CN, CA; longitudinal (stability axes) – CLA, CMA, CDA, CMQ, CLV, CMV, CDV, CLDE, CMDE, CDD1, CLDC, CMDC, CDDC, CLD1, CMD1, CDD1, CLD2, CMD2, CDD2, CL, CD. The first two characters of each name indicate the force or moment coefficients (for lateral-directional, CY = side force, CL = rolling moment, and CN = yawing moment; for longitudinal, CL = lift force, CD = drag force, CN = normal force, CA = longitudinal force (positive direction is rearward)), and the remaining characters indicate the quantity with respect to which the derivative is taken. (A ~ angle of attack, B ~ angle of sideslip, P, Q, R ~ angular rates, V ~ velocity, DE, DC, DA, DR, D1, and D2 ~ controls.)

If the coefficient is a function of Mach number and angle of attack, the data for each Mach number appear on a separate card, with each card containing the values of the coefficient for the NABP angle-of-attack breakpoints. These cards are in an 8F10 format, and the card entries may be continued on additional cards if needed.
If the coefficient is a function of Mach number only, the values for the NMBP Mach number breakpoints appear on one card in an 8F10 format. As before, this card may be continued if needed.

Angle-of-attack breakpoints: A card containing the NABP values of the angle-of-attack breakpoints in an 8F10 format is necessary. If NABP = 1, this card may be blank.

Mach number breakpoints: A card containing the NMBP values of the Mach number breakpoints in an 8F10 format is necessary. If NMBP = 1, this card may be blank.

Arbitrary parameter breakpoints: A card containing the NBP values of PARAM to distinguish the predicted derivative data sets is necessary. If no distinction other than longitudinal and lateral-directional is used, this card may be blank. The card is in an 8F10 format.

*User-supplied data for subroutine COND1.*—Any input required for subroutine COND1 goes in the input data at this point. With the subroutine supplied, there is none.

*Input tape data.*—The input tape data section should be omitted if the READ TAPE option is not active. Any input required by subroutine RDSET is made here. The subroutine supplied requires a card with the number of data words per record of the input tape; this card is in an I5 format. This is followed by three cards containing the channel numbers of the 40 channels to be used; each of these cards is in a 16I5 format. A 0 indicates a signal not used. The first 25 signals will be put on the MMLE program tape if a tape is written. (The signals should be in the BOTH order defined by item (11) in the MMLE NAMELIST.) The last 15 of the 40 channels are reserved for use in subroutine COND, should they be needed. These last 15 channels are typically used for fuel weight, flap position, or any other quantities useful in identifying the flight condition and vehicle configuration. The SETUP program automatically averages the values of all 40 data channels over the requested time interval and passes these averages to subroutine COND through a labeled common block.

*Case specification.*—The case specification is repeated as many times as necessary, once for each case to be analyzed.

Time card: The start time and end time for the case in hours, minutes, seconds, and milliseconds are required. The format is 2(312,13,IX).

*NAMElIST/COND/*: The following parameters may be read in NAMElIST/COND*/:

(1) LONG, LATR—(logical) — type of case to be analyzed. Set either LONG or LATR to true. Only one of the two variables can be set to true in the NAMElIST.

(2) CASE—(integer) — case number. Default value of 0 is used.
(3) DELTA- (four-word logical vector) — option that specifies which controls were used in the maneuver. A value of T for any element of DELTA indicates that the corresponding control was used. If all four locations are F (default condition), the MMLE program default is used; this default is $\delta_e$ for longitudinal cases, $\delta_q$ and $\delta_r$ for lateral-directional cases. If DELTA is omitted in a case but has been specified in a previous case of the same type (longitudinal or lateral-directional), it will assume the values of the previous case.

(4) FLT- (integer) — flight number. This identification is needed only on the first case.

All the following items may be set in subroutine COND instead of reading them in at this point. The subroutine supplied will set ALPHA, THETA, PHI, DETRIM, Q, V, and MACH if the READ TAPE option is active.

(5) ALPHA — average angle of attack.

(6) THETA — average pitch attitude. Default value of 0 is used.

(7) PHI — average roll attitude. Default value of 0 is used.

(8) Q — average dynamic pressure.

(9) V — average velocity.

(10) MACH— (real) — average Mach number.

(11) PARAM — extra identifying parameter. If nonzero, the predicted derivative data with the same value of PARAM will be used for the derivatives. If there is only one longitudinal data set or one lateral-directional data set, or a longitudinal and a lateral-directional data set, PARAM need not be specified. Default value of 0 is used.

(12) W — aircraft weight (pounds or newtons).

(13) IX, IY, IZ— (real) — moments of inertia (slug-ft$^2$ or kg-m$^2$).

(14) IXZ— (real) — cross-product of inertia (slug-ft$^2$ or kg-m$^2$). Default value of 0 is used.

(15) CG — center of gravity in fraction of chord. Default is the predicted derivative reference value.

(16) DETRIM — trimmed value of $\delta_e$. Default value of 0 is used.

Items (17) and (18) are simply for convenience if $\bar{q}$ and $V$ are not readily available. The subroutine COND supplied may compute $\bar{q}$ and $V$ from the values of indicated airspeed and altitude, using an approximation to the standard atmosphere.
(17) KIAS-(real) -- knots indicated airspeed. If KIAS is nonzero, $\ddot{q}$ and $V$ will be computed. Default value of 0 is used.

(18) ALT -- altitude (ft or m). Default value of 0 is used.

End card: The last card in the data deck contains a -1 in the first two columns to indicate the end of the data.

Output Description

The two primary outputs of the SETUP program are the MMLE program data tape and the punched card deck. These outputs are described in the MMLE Input Description section. A permanent disk file may be substituted for the data tape, without modifying the program. The punched card deck from SETUP will be ready to run through the MMLE program with the addition of control cards and the substitution of an END card for the last ENDCASE card at the end of the deck.

The printed output includes the predicted derivatives. For each case the data channel averages as passed to subroutine COND are printed if an input tape was read. All matrices punched in the MMLE program card deck are also printed for easy reference. A sample case is presented in appendix D.

SUMARY -- PLOTTING PROGRAM

Data presentation can be a time-consuming portion of the derivative estimation process when a large number of maneuvers are involved. It is still common to laboriously plot derivatives and wind-tunnel data by hand, a procedure which can easily take longer than the entire estimation process. To efficiently utilize available manpower, graphs or data listings should be automatically produced. The SUMARY program produces plots of estimated derivatives and confidence levels as a function of angle of attack and, if desired, provides predicted derivative values for comparison. The program is presented as a prepared package that may be modified to meet users' specific data presentation requirements. Listings of the program and its subroutines are presented in appendix E. A sample case is given in appendix F.

The SUMARY program reads a set of predicted and flight-determined derivatives, and plots specific groups of the data as instructed. Several groups may appear on one plot, indicated by different symbols. The same predicted derivative card deck used for the SETUP program may be used in the SUMARY program, or predicted derivatives may be omitted. The flight-determined derivatives are punched out by the MMLE program in the exact format required for the SUMARY program.

Input Description

Title card. -- The title card contains any information needed to identify a particular set of data that is appropriate to include in the printed output. All 80 columns on this card may be used.
NAMELIST/WIND/. – Parameters in NAMELIST/WIND/ are as follows:

(1) NABP – number of angle-of-attack breakpoints for predicted derivatives. Default value of 1 is used.

(2) NMBP – number of Mach number breakpoints for predicted derivatives. Default value of 1 is used.

(3) NBP – number of sets of predicted derivatives. The definition of a set of predicted derivatives is the same as that in the SETUP program. Default value of 1 is used.

(4) LONG, LATR-(eight-word logical vectors) – types of each set of predicted derivatives. The type is specified by setting corresponding element of either LONG or LATR to true. Only one of the two variables can be set to true in the NAMELIST. Default type for each set is longitudinal.

(5) NCLA, NCLO – number of coefficients in lateral-directional and longitudinal data sets, respectively. Default value of 0 is used.

(6) CGLA, CGLO – reference centers of gravity for lateral-directional and longitudinal predicted derivatives in fraction of chord. Default value of 0.25 is used.

(7) SHIFT-(logical) – parameter that corrects data for center-of-gravity location. If true, the flight \( C_{m} \) and \( C_{n} \) will be corrected to the predicted derivative reference center of gravity. Default condition is F.

(8) DEG, RAD-(logical) – options that specify degrees or radians for units of predicted derivatives. Only one of the two variables can be set to true in the NAMELIST. Rotary derivatives are per radian regardless of this option. Default units are degrees.

(9) BODY, STAB-(logical) – options that specify body or stability axes for input of predicted derivatives. If STAB = T, longitudinal predicted derivatives are converted from stability to body axes. If BODY = T, no conversion is made. Only one of the two variables can be set to true in the NAMELIST. Default condition is STAB = T.

(10) PRINT-(logical) – option that prints out predicted derivatives, if true. Default condition is F.

(11) WTPLOT-(logical) – option that plots predicted derivatives, if true. Default condition is T.

(12) CBAR, SPAN – aircraft reference chord and span, respectively. These quantities are needed only if SHIFT = T and there are lateral-directional data. Default values of CBAR = 0 and SPAN = 10^5° are used.

(13) AMIN, AMAX – minimum and maximum for values on angle-of-attack axis. Default values of AMIN = 0 and AMAX = 12. are used.

33
(14) ASCALE – scale for angle-of-attack axis in degrees per centimeter. Default value of 1. is used.

(15) YLEN – length of ordinate axis in centimeters. Default value of 10. is used.

(16) XDIST – X-distance between plots in centimeters. Default value of 10. is used.

(17) CRFACT – factor by which confidence levels are multiplied before plotting. If equal to 0, no confidence levels are plotted. Default value of 1. is used.

(18) NPARAM – variable which distinguishes the two modes of data organization to be used. If NPARAM = 0, flight data points are sorted by Mach number to the nearest Mach number breakpoint. Plots are then produced with the different Mach numbers indicated by different symbols. If NPARAM > 0, Mach number is ignored and the data are sorted by the value of PARAM, the extra identifying parameter, to the nearest PARAM breakpoint. Plots are then produced with different symbols distinguishing these groups. The lowest Mach number of the predicted derivatives is plotted if more than one Mach number breakpoint is specified. Only one predicted derivative curve is plotted. In this case there should be only one set of lateral-directional and one set of longitudinal predicted derivatives; if there is more than one set, only the first will be plotted. Default value of 0 is used.

Predicted derivatives. – The NBP sets of predicted derivatives are necessary in exactly the same format required for the SETUP program, including the cards with angle of attack, Mach number, and PARAM breakpoints.

Flight data. – The flight data desired are required at this point in the form punched on cards by the MMLE program if PUNCH = T (p. 25).

(1) Header card – TYPE, TITLE, MACH, ALPHA, PARAM, CG in format A4,1X,A35,4F10. TYPE is either LONG or LATR.

(2) A, B, AC, BC matrices in nondimensional form. The AC and BC matrices contain the confidence levels. The fifth column of the B matrix in a longitudinal case should contain $C_Z$ in the first row and $\delta_{trim}$ in the second row if they are desired for plotting. These quantities replace the logically expected, but more difficult to interpret, quantities (perturbation $C_{Z0}$ and $C_{m0}$) from which they are derived.

Plotting instructions. – The end of the flight data and the beginning of the plotting instructions are signaled by a card with PLOT in the first four columns. Then, for each set of plots desired, the following instruction cards are needed:

(1) TYPE, PARM, TOL – TYPE is either LATR or LONG. PARM should equal one of the PARAM breakpoints of the predicted derivatives. The program will then select the corresponding set of predicted derivatives to be used. Flight data points with this same value of PARAM (±TOL) will be selected for plotting. For instance, if PARM = 35. and TOL = 2., a flight point with PARM = 36. will be plotted, but
a flight point with \( \text{PARAM} = 38 \) will be rejected. In the special case, \( \text{PARM} = 0 \), the first set of predicted derivatives of the correct type (LATR, LONG) is used together with all the flight data. The format of this card is A4,F6,F10.

(2) Up to six cards specifying the derivatives to be plotted and the scales to use. Four plot instructions are included on a card (less may be on the last card). Each plot instruction is of the form \( \text{DERIV}, \text{SMIN}, \text{SMAX} \); \( \text{DERIV} \) is the derivative name, and \( \text{SMIN} \) and \( \text{SMAX} \) are the minimum and maximum values for the ordinate. The valid derivative names are the same as those in the SETUP program for lateral-directional data; for longitudinal data, all the body axis derivative names except \( \text{CA} \) are valid and the additional name of \( \text{DE} \) may be used to plot \( \delta \) versus \( \alpha_{\text{trim}} \). If \( \text{SMIN} = \text{SMAX} \) (in particular, if left blank), automatic scaling will be used for the plot. The format of these cards is 4(A4,F6,F10).

End card.—The end of the plotting instructions is signaled by a card with END starting in column 1.

Output Description

The printed output from the SUMARY program includes the header cards for all flight points read in and a summary of the plotting instructions. The predicted derivatives are printed if PRINT is set to T. In addition, informative messages are provided if no predicted derivatives or flight data are available at a requested condition.

Plots are scaled for centimeter grid paper. Confidence levels are indicated by vertical bars. Predicted derivative data are identified by small symbols that correspond to those in the figure legend, at the beginning and end of each curve. A sample is shown in appendix F.

CONCLUDING REMARKS

A digital computer program written in FORTRAN IV has been successfully applied by relatively inexperienced personnel to aircraft linear parameter estimation problems with measurement noise but no state noise. This maximum likelihood estimation program includes an option for using \( \text{a priori} \) information and provides estimates of the derivatives and confidence levels. A program to automate the setup work and a program to plot the results have also been written. The three programs form a package which has been used to successfully analyze 1500 maneuvers on 20 aircraft.

Flight Research Center
National Aeronautics and Space Administration
Edwards, Calif., January 22, 1975
APPENDIX A

MMLE PROGRAM AND SUBROUTINES

Listings of the main program and the subroutines used in the MMLE program are presented. The listings are preceded by a brief description, a flow chart, when needed for clarification, and programing notes which explain some of the conventions used and point out items needed to understand the operation of the program.

MAIN MMLE PROGRAM

Description: The main MMLE program activates the three operating modes of the program (basic mode, D1 determination mode, and a priori variation mode).

Flow chart:
Programing notes: The PROGRAM card is required on CDC 6000/7000 systems. On an IBM 360/370 system the following DD cards, or equivalent information, are necessary to perform the same function as the PROGRAM card:

//GO.FT02F001 DD SYSOUT=B,SPACE=(TRK,10,RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3520)
//GO.FT03F001 DD SYSOUT=A,SPACE=(TRK,50,RLSE),
// DCB=(RECFM=FBA,LRECL=133,BLKSIZE=3458)
//GO.FT04F001 DD DUMMY

(Substitute the appropriate DD card for the input file if a tape or disk input is used.)

//GO.FT08F001 DD UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(10,2)),
// DCB=(RECFM=VSB,LRECL=92,BLKSIZE=924),DSN='PLOTTER DATA'
//GO.FT07F001 DD UNIT=SYSDA,DISP=(NEW,DELETE),SPACE=(CYL,(10,2)),
// DCB=(RECFM=VSB,LRECL=84,BLKSIZE=844),DSN='INTERNAL'
//GO.PLOTTAPE DD DUMMY

(Substitute the appropriate DD card for the plotter file as used on the particular system. The file name will be either PLOTTAPE or FT13F001, depending on the plotter software used.)

//GO.FT01F001 DD*

This routine alters PRINT and PLOTEM to suppress any extraneous output during intermediate steps of the D1 determination and the a priori variation option.

Important variables –

ND1, NAPR – control the D1 determination and a priori variation options as described in MMLE NAMELIST input.

D2 – vector of final weighted relative errors returned from the estimation process.

STORE – storage for final coefficient values during a priori variation. It is used to plot these values.
APPENDIX A — Continued

Program listing:

```
PROGRAM MMLE(INPUT,PUNCH,OUTPUT,TAPE7,TAPE8,TAPE9,
- TAPE15,TAPE2=PUNCH,TAPE3=OUTPUT) MAIN 0

C C
COMMON /ALLDIM/ MAX,MIX
COMMON /BUFY/ BUFFER,Y,THT
COMMON /MATRIX/ A,E,AAA,BB,AP,BP,D1,D1
COMMON /COM/ NCASE,MAX,NPTS,NPI,SPS,PRI,T,LON,T,LAT,PLT,HOLD,ND1,
- D1LX,NAV,T,T,MAPR,ERRSUM,LAST,RATIO
DIMENSION NPTS(15),N1S(17),AMOLD(5,W),
- DMOLD(5,8),D2(7),STP,STORE(IW,27)
- BUFFER(IW,27),RT(5,5)
COMMON /PLTM/PLT,LAT,PLT,AA(5,4),EE(5,4),STP,PRINT,PRINT,
- FIRST,LAST
FIRST=.TRUE.
LAST=.FALSE.
REWIND 5
REWIND 7
CALL EDIT
IF(ND1.EQ.Q.AND.NaN.RT.LT.1) GO TO 10
PLT=PLTM
IF(NAV.NE.1) PLTM=.FALSE.
PRINT=PRINT
HOLD=MAPR
MAX=5
CALL AMAKE(AMOLD,4)
CALL AMAKE(BMOLD,0)
C*********** BASE RUN
10 CALL DATA(.TRUE.)
CALL AGIPL
IF(ND1.EQ.W) GO TO 15
PLT=.FALSE.
PRINT=.FALSE.
CALL OUTPUT(D2)
IF (ND1.EQ.W) GO TO 100
C*********** D1 DETERMINATION (IF DESIRED)
TOL1=1./0TOL
GO 20 I=1,ND1
STP=TRUE.
GO 30 J=1,MIX
IF(D1(J,.NE.,W) GO TO 22
D1(J)=1.
GO TO 30
22 IF(D2(J).GT.0.TOL,O1),D2(J),LT,TOL) STF=.FALSE.
IF(D2(J).LT.1.) GO TO 25
D2(J)=0.
GO TO 27
25 D1(J)=1./D2(J)
GO TO 27
27 D1(J)=D1(J)
D2(J)=D2(J)
GO TO 27
30 CONTINUE
IF(LT,ND1.AND..NOT.STP) GO TO 40
PLT=PLT
PRINT=PRINT
40 WRITE(3,EO0)
```

APPENDIX A — Continued

MAX=8
CALL ASPIT(D1)
MAX=5
CALL AMAKE(A,AHOLD)
CALL AMAKE(B,BHOLD)
CALL DATA(.FALSE.,)
CALL AGIRL
CALL OUTPUT(02)
IF(93P) GO TO 95
94 CONTINUE
95 WRITE(3,2002)
MAX=8
CALL ASPIT(D1)
160 IF(INAPR.GT.O) GO TO 165
IF(.NOT.PLOTEM) GO TO 200
CALL THPLOI(FIRST)
FIRST=.FALSE.
GO TO 242
C-------------------------------------------- APRIORI VARIATION (IF DESIRED)
165 PRINT=.FALSE.
IF(WHOLD.EQ.4.) WMPR=.001
WMPR=WMPR
I=1
116 I=I+1
JKMM=1
DO 120 J=1,3
DO 120 K=1,4
IF(BB(J,K)) GO TO 120
120 CONTINUE
DO 130 K=1,3
If(AA(J,K)) GO TO 130
130 CONTINUE
STORE(1,JKMM)=B(J,K)
136 CONTINUE
STORE(1,JKMM+1)=ERRSUM
MAX=8
CALL AMAKE(A,AHOLD)
CALL AMAKE(B,BHOLD)
IF(I-NAPR) GO TO 140
140 WRITE(3,2001)WMPR
CALL DATA(.FALSE.,)
CALL AGIRL
CALL OUTPUT(02)
WMPR=WMPR*WFACT
GO TO 116
166 IF(.NOT.PLOT) GO TO 200
CALL APPLISTORE,AA,BE,NAPR,WMPR,WFC,WFACT,LONF,FIRL,LASt,RATI0
FIRST=.FALSE.
200 IF(.NOT.LASt) GO TO 5
IF(.NOT.FIRST) CALL PLOTQ(0.0,.999)
2000 FORMAT(15H000,000,000)
2001 FORMAT(12H0WMPR NOW =,E10.2)
2002 FORMAT(10H0FINAL 01)
STOP
END
APPENDIX A — Continued

SUBROUTINE EDIT

Description: Subroutine EDIT initializes the program, sets defaults, and reads input options and matrices.

Programming notes: If used with a system that does not support the NAMELIST, some other form of input might be used.

Subroutine MATLD, called at card 1820, sets appropriate elements of ABC to -99999 when reading a matrix. These elements are then tested after all the matrix input has been made to determine what matrix defaults are needed.

The R matrix is inverted at card 2460, since \( R^{-1} \) is the form needed by the rest of the program.

From card 2530 on, the AA and BB matrices are being converted to logical variables and the number of the different types of unknown coefficients to be determined is found. An element in AA or BB is set to false if that element in A or B is to be determined. This may be contrary to the expected convention.
Subroutine listing:

SUBROUTINE EDIT
SETS DEFAULT VALUES AND READS PROGRAM OPTIONS FROM CARDS

COMMON /ALLOIM/ MAX, MAX
COMMON /OPTLOT/ ZMAX, ZMIN, DMAX, DOMIN, TIMES, NCLOT
COMMON /SOCIRL/ JK, GM, DOMAX, ERMAX, ZEROS, RX, DOWD, APRT
- VARP, ZER0, APRT, JK, V, DIA
COMMON /MATRIX/ AA, AB, AP, AB, DILR
COMMON /COM/ NCASE, NPTS, ITPS, PRINT, XLNT, NPLOT, LONLAB
- SCALE, NREC, ORDER, METER, I, IV, IX, IY, IZ, IXZ, O, V, GROSX, T

END
APPENDIX A -- Continued

C***************************************** DEFAULTS
NCPLT=8
V=0.
Q=0.
MACH=0.
PUNCH=.FALSE.
NEAR=8
NP RE=15
METRIC=.FALSE.
MDN=.FALSE.
DO 1 I=1,15
14 ORDER(I)=I
DO 10 I=1,15
10 DCMAX(I)=0.
DO 10 I=1,12
13 ARC(I)=LAB(I)
CORRECT=.FALSE.
CARD=.FALSE.
TAPEx=.TRUE.
ZEROIN=.FALSE.
BIAS=.FALSE.
PLIT=.TRUE.
TEST=.FALSE.
LONG=.FALSE.
LAT=.FALSE.
DIAG=.TRUE.
PLTMAX=1.E+05
RDMAX=1.E+20
PUNCH=.FALSE.
PARAM = 0.
CC = .25
X0=0.
Z0=0.
XAY=0.
ZAY=.0.
THIN=1
D1A(I,1)=5.
D1A(2,1)=5.
MAX=5
CALL AZOT (01)
MAX=5
MTX=5
RT5,1=4.
RT5,2=4.
R(5,3)=ABC(I,2)
CALL AZOT(P1)
DO 136 I=1,?
13 E DC(I)=0.
NC(I)= 0.
X0=0.
ALPHA=.99.
PRINT=.FALSE.
DO 137 I=1,?
ZMIN(I)=0.
ZMAX(I)=0.

EDIT 570
EDIT 580
EDIT 590
EDIT 600
EDIT 610
EDIT 620
EDIT 630
EDIT 640
EDIT 650
EDIT 660
EDIT 670
EDIT 680
EDIT 690
EDIT 700
EDIT 710
EDIT 720
EDIT 730
EDIT 740
EDIT 750
EDIT 760
EDIT 770
EDIT 780
EDIT 790
EDIT 800
EDIT 810
EDIT 820
EDIT 830
EDIT 840
EDIT 850
EDIT 860
EDIT 870
EDIT 880
EDIT 890
EDIT 900
EDIT 910
EDIT 920
EDIT 930
EDIT 940
EDIT 950
EDIT 960
EDIT 970
EDIT 980
EDIT 990
EDIT1000
EDIT1010
EDIT1020
EDIT1030
EDIT1040
EDIT1050
EDIT1060
EDIT1070
EDIT1080
EDIT1090
EDIT1100
EDIT1110
EDIT1120
EDIT1130
APPENDIX A — Continued

FIXED(I)=0.
137 SCALE(I)=1.
XAN=2.
ZAY=0.
BOUND=.001
SPS=0.
WMAPR=0.
NOITER=6
NCASE=1
NI=35
TIMESC=1.
OOUND:.O01
SPS=O.
WMAPR=O.
NOITER=6
NCASE=1
NI=35
TIMESC=1.
OOUND:.O01
SPS=O.
WMAPR=O.
NOITER=6
NCASE=1
NI=35
TIMESC=1.
OOUND:.O01
SPS=O.
WMAPR=O.
NOITER=6
NCASE=1
NI=35
TIMESC=1.
OOUND:.O01
SPS=O.
WMAPR=O.
NOITER=6
NCASE=1
NI=35
TIMESC=1.
OOUND:.O01
SPS=O.
WMAPR=O.
NOITER=6
NCASE=1
NI=35
TIMESC=1.
OOUND:.O01
SPS=O.
WMAPR=O.
NOITER=6
NCASE=1
NI=35
TIMESC=1.
OOUND:.O01
SPS=O.
WMAPR=O.
NOITER=6
NCASE=1
NI=35
TIMESC=1.
OOUND:.O01
SPS=O.
WMAPR=O.
NOITER=6
NCASE=1
NI=35
TIMESC=1.
OOUND:.O01
SPS=O.
WMAPR=O.
NOITER=6
NCASE=1
NI=35
TIMESC=1.
OOUND:.O01
SPS=O.
WMAPR=O.
NOITER=6
NCASE=1
NI=35
TIMESC=1.
OOUND:.O01
SPS=O.
WMAPR=O.
NOITER=6
NCASE=1
NI=35
TIMESC=1.
OOUND:.O01
SPS=O.
WMAPR=O.
NOITER=6
NCASE=1
NI=35
TIMESC=1.
OOUND:.O01
SPS=O.
WMAPR=O.
NOITER=6
NCASE=1
NI=35
TIMESC=1.
OOUND:.O01
SPS=O.
WMAPR=O.
NOITER=6
NCASE=1
NI=35
TIMESC=1.
OOUND:.O01
SPS=O.
WMAPR=O.
NOITER=6
NCASE=1
NI=35
TIMESC=1.
OOUND:.O01
SPS=O.
WMAPR=O.
NOITER=6
NCASE=1
NI=35
TIMESC=1.
OOUND:.O01
SPS=O.
WMAPR=O.
NOITER=6
NCASE=1
NI=35
TIMESC=1.
OOUND:.O01
SPS=O.
WMAPR=O.
NOITER=6
NCASE=1
NI=35
TIMESC=1.
OOUND:.O01
SPS=O.
WMAPR=O.
NOITER=6
NCASE=1
NI=35
TIMESC=1.
OOUND:.O01
SPS=O.
WMAPR=O.
NOITER=6
NCASE=1
NI=35
TIMESC=1.
OOUND:.O01
SPS=O.
WMAPR=O.
NOITER=6
NCASE=1
NI=35
TIMESC=1.
OOUND:.O01
SPS=O.
WMAPR=O.
NOITER=6
NCASE=1
NI=35
TIMESC=1.
OOUND:.O01
SPS=O.
WMAPR=O.
NOITER=6
NCASE=1
NI=35
TIMESC=1.
OOUND:.O01
SPS=O.
WMAPR=O.
NOITER=6
NCASE=1
NI=35
TIMESC=1.
OOUND:.O01
SPS=O.
WMAPR=O.
NOITER=6
NCASE=1
NI=35
TIMESC=1.
OOUND:.O01
SPS=O.
WMAPR=O.
NOITER=6
NCASE=1
NI=35
TIMESC=1.
OOUND:.O01
SPS=O.
WMAPR=O.
NOITER=6
NCASE=1
NI=35
TIMESC=1.
OOUND:.O01
SPS=O.
WMAPR=O.
NOITER=6
NCASE=1
NI=35
TIMESC=1.
OOUND:.O01
SPS=O.
WMAPR=O.
NOITER=6
NCASE=1
NI=35
TIMESC=1.
OOUND:.O01
SPS=O.
WMAPR=O.
APPENDIX A – Continued

WRITE (3,1913) LABELS,VAR,ZERO,FIXED,DC,SCALE
IF (PLT0MX) WRITE (3,2001) ZMIN,DCMIN,MAX,SCMAX
SPS=SPS/ATHIN
DO 130 I=1,NCASE
READ (1,1800) ST,ET
STC(I)=ST+I*60*3600
ETC(I)=ET+I*60*3600
DO 105 I=1,NCASE
READ (1,1000) ST,ET
STC(I)=ST+(ST(3)+60*ST(ZI_3600)ST(III
ETC(I)=ET+W+I*60*ET(3)+60*ET(3)+60*ET(I)
GO TO 106
106 WRITE (3,20021) ST,ET
C****************************************************************************** REAC MATRICES
WRITE (3,2003) TITLE,JULIAN
WRITE (3,2004) MAX,8
101 CALL MATLD (MATRX,ABC,ILD)
IF (IABS (ILO).EQ.9999) GO TO 108
IF (ILD.EQ.0) DIAG=.FALSE.
IF (ILO.EQ.1) CALL MAK(A,MATRX,5)
IF (ILD.EQ.2) CALL MAK(B,MATRX,5)
IF (ILD.EQ.3) CALL MAK(AA,MATRX,5)
IF (ILD.EQ.4) CALL MAK(AR,MATRX,5)
IF (ILD.EQ.5) CALL MAK(APA,MATRX,5)
IF (ILD.EQ.6) CALL MAK(APB,MATRX,5)
IF (ILD.EQ.7) CALL MAK(AR,MATRX,5)
IF (ILD.EQ.8) CALL MAK(ARP,MATRX,5)
IF (ILO.EQ.9) CALL MAK(O,B,MATRX,5)
IF (ILD.EQ.10) CALL HAK(AP,MATRX,5)
IF (ILD.EQ.11) CALL MAK(BP,MATRX,5)
IF (ILD.EQ.12) CALL MAK(A,B,MATRX,5)
GO TO 101
108 MAKE=
*IF (ILO.EQ.11) .NE. -99999. .AND. ABC(11).EQ.-99999. .APPRO=.TRUE.
IF (ABC(1).GT. 0.9) GO TO 113
LATR=.TRUE.
GO TO 117
LONG=.TRUE.
DO 118 I=1,12
LABELS(I)=LAB(i)
WRITE (3,2007)
117 IF (ILO.EQ.9999) LAST*.TRUE.
IF (ABC(5).NE.-99999.) CALL MADE(AR,A)
IF (ABC(6).NE.-99999.) CALL MADE(BR,B)
IF (LONG) GO TO 121
DO 118 I=1,15
118 D(I)=D(I)+LAT(I)
119 IF (ABC(7).NE.-99999.) CALL MADE(APA,APPLAT)
IF (ABC(8).NE.-99999.) CALL MADE(APB,BPRLAT)
IF (ABC(9).NE.-99999.) CALL MADE(AA,ALAT)
IF (ABC(10).NE.-99999.) CALL MADE(BB,BRRLAT)
IF (XZ.INE.3.) .OR. (XZ.INE.3.) .OR. (XZ.INE.3.) ICOREC=.TRUE.
IF (ABC(12).EQ.-99999) GO TO 123
R(3,2)=-IXZ/I
R(3,2)=-IXZ/I
R(3,2)=IXZ/I
IF (TEST) CALL ASPITR(9)
123 IF (.NOT.BOT?) GO TO 122
*END
APPENDIX A – Continued

```plaintext
DO 7 I=1,3
7 ORDER(I)=I+15
DO 9 I=1,11
9 ORDER(I)=I+1
ORDER(I)=3
GO TO 122
121 IF(ABC(7).NE.-99999.) CALL AMAKE(1APRA,APRLON)
IF(ABC(3).NE.-99999.) CALL AMAKE(4APRLON)
IF(ABC(4).NE.-99999.) CALL AMAKE(2APRLON)
IF(ABC(9).EQ.-99999.) GO TO 122
DO 124 I=1,5
124 ORDER(I)=I+15

C******************************************************************************
C******************************************************************************
C******************************************************************************
C******************************************************************************

122 CALL INVIR(MAX,2)
MX=MAX+2
MU=2MAX+2
MXP1=MX+1
MYP1=MY+1
MAXP1=MAX+1
MXP2=MAX+2
MYP2=MY+2
M(X)=M(X+1)
DO 150 I=MXP1,PZ
150 IF(DT(I,I).EQ.0.) VAR(I-MX)=.FALSE.
JKMM=0
DO 120 I=1,MY
DO 110 J=1,MX
IF(AA(I,J)) JKMM=JKMM+1
110 IF(BB(I,J)) JKMM=JKMM+1
120 CONTINUE
JKM=JKMK+I
DO 125 I=1,MY
125 IF(JRNO(I)) JKMK=JKMK+1
JKV=JKMK
DO 126 I=1,MS
126 IF(JRNO(I)) JKMK=JKMK+1
RETURN

100 FORMAT(13X,3I3,H)
2000 FORMAT(28X)
2001 FORMAT(15H PLOT LIMITS/5X,7HMINIMUM,15F8.2/5X,7HMAXIMUM,15F8.2)
2002 FORMAT(15H MANEUVER,H4.12H START TIME,H4.12H STOP TIME,H4.15)
2003 FORMAT(15H INPUT MATRICES #)
```

APPENDIX A -- Continued

208 FORMAT(1H1,27X,9I4,1X,9I4,1X,4X,14X,NEWTON-RAPHSON,      EDIT2850
        28H DIGITAL DERIVATIVE MATCHING/60X,10H APR 1974)        EDIT2860
208 FORMAT(2H40H WILL BE DETERMINED USING,11H, EDIT2870
-28H PASSES, RELAXATION FACTOR =F5.2,13H TOLERANCE =F5.2)       EDIT2880
207 FORMAT(4S THE A MATRIX INDICATES CASE IS LONGITUDINAL,      EDIT2890
-28H LABELS ABOVE ARE WRONG, APPROPRIATE CORRECTIONS NOW MADE.) EDIT2900
208 FORMAT(12HNO MORE WILL BE RUN WITH,13H25H VALUES. FIRST 0., THEAEDIT2910
-9H THEREAFTER MULTIPLYING BY,EB.2) EDIT2920
209 FORMAT(50H INFUT DATA {T INDICATES TRUE OR YES, F INDICATES, EDIT2930
-12H FALSE OR NO) EDIT2940
257X,SA4,SH CASEIi6H DATA SOURCE, EDIT2950
56H SAMPLES/SECOND ON SOURCE FILE IF 0, DETERMINED FROM TIM,EDIT2970
229X,26HDIVIDED BY THINNING FACTOR, EDIT2980
2H DATA WCPOS PER RECORD. SPECIAL SIGNAL ORDER DEFAULT? , EDIT3000
2010 FORMAT(/I6H OPRAM OPTIONS/2&HO APRIORI WEIGHTING =E8.2,      EDIT3010
-13X,13X,22H TIME HALVINGS IN EAT./EDIT3020
-5X,12H ITERATIONS =I3,32H (ITERATION WILL STOP IF ERROR,      EDIT3030
-36HUM CHANGES BY LESS THAN A FACTOR OF,E4.2,1H/5X, EDIT3040
-49H CASE WILL BE STOPPED IF ERROR SUM IS GREATER THAN,E9.2) EDIT3050
211 FORMAT(/7MOOUTPUT/12HO PLOTS? ,LL1.25H (NO PLOTS UNLESS FINAL, EDIT3060
-22H ERROR SUM IS LESS THAN,E9.3,1H/10X, EDIT3070
-52H NUMBER OF CONTROLS AND EXTRA SIGNALS TO BE PLOTTED =I3,32H, EDIT3080
-10X,24H SECONDS PER CENTIMETER =F5.2/5X, EDIT3090
-50H PRINTED FLIGHT AND FINAL COMPUTED TIME HISTORIES 111/5X, EDIT3100
-57H EXTRA OUTPUT OF INTERMEDIATE STEPS FOR A DIAGNOSTIC AID? , EDIT3110
-11H 1/5X,5H PUNCHER NON-DIMENSIONAL DERIVATIVES AND CONFI,EDIT3120
-14H ONCE LEVELS,111/5X,5H PUNCHER COND DIMENSIONAL , EDIT3130
-20H MATRICES? LL1) EDIT3150
212 FORMAT(50HFLIGHT CONDITION AND VEHICLE CHARACTERISTICS (0 INDICA, EDIT3160
-55H ES VALUE OBTAINED FROM TIME HISTORY OR GAB, W OR MACH), EDIT3170
-44H MACH =F8.3,1H/4H ALPHA =F8.3,22M (IF 999. , OBTAINED, EDIT3180
-18H FROM TIME HISTORY)/5X,1H CENTER OF GRAVITY =F6.3,10X, EDIT3190
-2H OTHER IDENTIFYING PARAMETER =F10.3,5X,11H MING AREA =, EDIT3200
-57H NUMBER OF CONTROLS AND EXTRA SIGNALS =I3,32H, EDIT3210
-4MY =F10.1,4X,5H IX =F8.1/5X,4H WEIGHT =, EDIT3220
-91X,2X,2H INSTRUMENT OFFSETS FROM CG/ EDIT3250
-11H 40X,4H INSTRUMENT OFFSETS FROF CGI EDIT3260
-14X,5H MACH =F8.3,4H R苑 Tc=93,4H FLIGHT & #3,4X,1H/3Y, EDIT3270
-10X,4H DIRECTION OFFSETS (+ = INSTRUMENT IS BELOW CG/ EDIT3280
-14X,4H MACH =F8.3,4H AX,AY,AX,AY, EDIT3290
213 FORMAT(50H SIGNAL SCA LE AND R ACES/9M SIGNALS,7X,14A,4V, EDIT3300
-1Hh VAR EAS12X,17X,11H VAR I.G. ,6X,11H ,11H/EDIT3310
-12H FIXED RINS,11F,2/12H SCALE FACT,FI4.2) EDIT3320
END                                                 EDIT3330
APPENDIX A — Continued

SUBROUTINE DATA

Description: Subroutine DATA reads the input time histories, performs any scaling and biasing required, and completes the program initialization. Averages of several time histories are obtained for use as default values for input parameters not set.

Programming notes: Comment cards separate major subroutine sections. If this is an intermediate step in the D1 determination or the \textit{a priori} variation option, most of the subroutine is skipped since those sections were executed in the first step; this is true when the formal parameter IN is false.

Important variables –

- \textit{X} — vector containing one time point of the input time histories in degrees.
- \textit{Z}, \textit{DCR} — vectors containing the input observations and controls in radians.
- \textit{C} — matrix containing factors for nondimensionalizing derivatives.
- \textit{APR} — matrix containing any off-diagonal \textit{a priori} weightings. These weightings would be stored in the upper triangular portion of APR. There are no terms inserted here, but if such terms are desired, they may be inserted and the rest of the program will treat them properly. This matrix is referred to elsewhere in the program as \textit{SUM}, and the lower triangular portion and the diagonal will be used to store other information.
- \textit{APRD} — vector containing the diagonal \textit{a priori} weightings.
Subroutine listing:

**SUBROUTINE DATA(IN)**

**READS TIME HISTORIES, PERFORMS VARIOUS INITIALIZATION**

**COMMON /ALLOIM/ MAX, MEX**
**COMMON /COM/ NCASE, MZ, NPTS, NPTT, SP, PRINT, LATR, PLOTEM, DI, DITOL, DAIR, TEMP, WAC, WMAPR, ERM, LAST, LNG**
**COMMON /MATR/ XX, XX, XX, XX, XX, XX**
**COMMON /TOGIRL/ JKM, JMM, IERRMAX, ZERINO, TX, TX**
**COMMON /INFO/ HH, NOITER, W, MPP, HU, TST, XBT, TST, IPAR, IYX2, XT, WZM**
**COMMON /COM/ NCASE, MZ, NPTS, NPTT, SP, PRINT, LATR, PLOTEM, DI, DITOL, DAIR, TEMP, WAC, WMAPR, ERM, LAST, LNG**
**COMMON /MATR/ XX, XX, XX, XX, XX, XX**
**COMMON /TOGIRL/ JKM, JMM, IERRMAX, ZERINO, TX, TX**
**COMMON /INFO/ HH, NOITER, W, MPP, HU, TST, XBT, TST, IPAR, IYX2, XT, WZM**

**DATA**

DATA 0 0
DATA 10 0
DATA 20 0
DATA 30 0
DATA 40 0
DATA 50 0
DATA 60 0
DATA 70 0
DATA 80 0
DATA 90 0
DATA 100 0
DATA 110 0
DATA 120 0
DATA 130 0
DATA 140 0
DATA 150 0
DATA 160 0
DATA 170 0
DATA 180 0
DATA 190 0
DATA 200 0
DATA 210 0
DATA 220 0
DATA 230 0
DATA 240 0
DATA 250 0
DATA 260 0
DATA 270 0
DATA 280 0
DATA 290 0
DATA 300 0
DATA 310 0
DATA 320 0
DATA 330 0
DATA 340 0
DATA 350 0
DATA 360 0
DATA 370 0
DATA 380 0
DATA 390 0
DATA 400 0
DATA 410 0
DATA 420 0
DATA 430 0
DATA 440 0
DATA 450 0
DATA 460 0
DATA 470 0
DATA 480 0
DATA 490 0
DATA 500 0
DATA 510 0
DATA 520 0
DATA 530 0
DATA 540 0
DATA 550 0
DATA 560 0
**APPENDIX A -- Continued**

```plaintext
THETA=O.
VEL=V
DO 500 II=1,NCASE
DATA 590
ISTMS=STD(II)
DATA 600
IETMS=ETC(I)
DATA 610
ITHIN=TNIN-I
DATA 620
NPTS(I11)=O
DATA 630
IF(I11.LE.4) DCR(I11)=1.
DATA 640
GO TO 250
DATA 650
READ 1,1001 T X
DATA 660
READ 2,20 (W) T,(RECORD(I).I=t,NREC)
DATA 670
IF((T(4)+100*T(3)+60*T(2)+3600*T(1)).LT.ISTMS) GO TO 260
DATA 680
ITST=T(W)
DATA 690
IF(.NOT.TAPE) GO TO 300
DATA 700
DO 270 I=1,15
DATA 710
X(I)=RECORD(I) SCALE(I)
DATA 720
ITM=T+I*100
DATA 730
IF(ITM.GT.IETMS) GO TO 300
DATA 740
ITHIN=ITHIN+I
DATA 750
IF(MOO(ITHIN,THIN).NE.G) GO TO 385
DATA 760
NPTS(I11)=NPTS(I11)+1
DATA 770
MM=MM+I
DATA 780
NPTS(I).NE.20. VEL=EXT_A(2)
DATA 790
X(I)=X(I)-ZB
DATA 800
ALFA=ALFA+EXTRA(I)
DATA 810
AV=AV+EXTRA(2)
DATA 820
PHI=PHI+EXTRA(I)
DATA 830
THETA=THETA+X(I)
DATA 840
DO 365 I=I+1
DATA 850
DDC(I)=DDC(I)+DCAS(I)
DATA 860
DDCR(I)=DDCR(I)/RAD
DATA 870
DO 315 I=I+1
DATA 880
X(I)=X(I)*SCALE(I)+BIAS(I)
DATA 890
X(I)=X(I)+XALFA*SCALE(I)
DATA 900
ZI=ZI+ZBSCALE(I)
DATA 910
WRITE(I,71) ITIME,Z,DCR,EXT_A
DATA 920
IF(.NOT.PRNT) GO TO 375
DATA 930
IF(_OD(LINE.50).ED°O)
DATA 940
WRITE(3, 400) TITLE,JULIAN,GROSWT,AIX_AIZ,AIXZ,AIZ,VEL LABELS
DATA 950
WRITE(7,400) TITLE,JULIAN,GROSWT,AIX_AIZ,AIXZ,AIZ,VEL LABELS
DATA 960
WRITE(7,400) TITLE,JULIAN,GROSWT,AIX_AIZ,AIXZ,AIZ,VEL LABELS
DATA 970
WRITE(7,400) TITLE,JULIAN,GROSWT,AIX_AIZ,AIXZ,AIZ,VEL LABELS
DATA 980
WRITE(7,400) TITLE,JULIAN,GROSWT,AIX_AIZ,AIXZ,AIZ,VEL LABELS
DATA 990
WRITE(7,400) TITLE,JULIAN,GROSWT,AIX_AIZ,AIXZ,AIZ,VEL LABELS
DATA 1000
WRITE(7,400) TITLE,JULIAN,GROSWT,AIX_AIZ,AIXZ,AIZ,VEL LABELS
DATA 1010
WRITE(7,400) TITLE,JULIAN,GROSWT,AIX_AIZ,AIXZ,AIZ,VEL LABELS
DATA 1020
WRITE(7,400) TITLE,JULIAN,GROSWT,AIX_AIZ,AIXZ,AIZ,VEL LABELS
DATA 1030
WRITE(7,400) TITLE,JULIAN,GROSWT,AIX_AIZ,AIXZ,AIZ,VEL LABELS
DATA 1040
WRITE(7,400) TITLE,JULIAN,GROSWT,AIX_AIZ,AIXZ,AIZ,VEL LABELS
DATA 1050
WRITE(7,400) TITLE,JULIAN,GROSWT,AIX_AIZ,AIXZ,AIZ,VEL LABELS
DATA 1060
WRITE(7,400) TITLE,JULIAN,GROSWT,AIX_AIZ,AIXZ,AIZ,VEL LABELS
DATA 1070
WRITE(7,400) TITLE,JULIAN,GROSWT,AIX_AIZ,AIZ,VEL LABELS
DATA 1080
WRITE(7,400) TITLE,JULIAN,GROSWT,AIZ,VEL LABELS
DATA 1090
WRITE(7,400) TITLE,JULIAN,GROSWT,AIZ,VEL LABELS
DATA 1100
WRITE(7,400) TITLE,JULIAN,GROSWT,AIZ,VEL LABELS
DATA 1110
WRITE(7,400) TITLE,JULIAN,GROSWT,AIZ,VEL LABELS
DATA 1120
WRITE(7,400) TITLE,JULIAN,GROSWT,AIZ,VEL LABELS
DATA 1130
```

---

The source code appears to be a segment of a FORTRAN program for data processing, likely related to scientific computations or simulations. The code includes loops, conditional statements, and data manipulation, indicative of a complex algorithm designed to process input data and produce output results. The specific details of the application are not clear from the code alone, but it is structured to read and analyze data, perform computations, and write results to files. The presence of variables like `THETA`, `VEL`, `NPTS`, and `ISTMS` suggests a focus on tracking or calculating certain quantities over iterations. The loops and conditional blocks are designed to handle different cases, possibly for error checking or selecting appropriate data subsets for analysis. The final sections seem to be related to output formatting, possibly for reporting purposes.
APPENDIX A – Continued

WRITE(3,2010) T,X
LINE=LINE+1
375 IF(ITM.EQ.IETM) GO TO 430
165 IF(CARD) GO TO 400
READ (4) T,X,(RECORD(I),I=1,NREC)
390 X(I)=RECORD(ORDER(I))
GO TO 300
400 READ IT,1001 T,X
GO TO 300
430 IF(NPTS(I).LT.1) GO TO 435
WRITE(3,2000) IT
STOP
435 NPTT=NPTT+NPTS(I)
WRITE(3,2001) IT, NPTT
500 CONTINUE
ANPT=FLOAT(NPTT)
IF(MACH.EQ.0.) MACH=AMACN/ANPT
IF(ALPHA.EQ.999.) ALPHA=ALFA/ANPT
IF(V.EQ.0.) V=AV/ANPT
IF(Q.EQ.0.) Q=AQRAR/ANPT
VOG=V
AM=GROSWT_VOG/(Q_S)
2=2.*V
IF(LONG) GO TO 170
C ********************** LATERAL SETUP
XAN=-ZAY
ZAX=XAY
IPQ=3
IXY=1
PI=VOG
AP=1.
GSM=0.5*SPAN
GBSR=0.5*SPAN
C1,2=1.
C1,2=V2*AIX/GBSR
C1,2=V2*AIZ/GBSR
C1,2=0.
C1,2=-C1,2
C1,2=0.
C1,2=AIX/QS_RB
C1,2=AIZ/QS_RB
DO 150 J=3,7
150 C1,3=C1,3*PAD
GO TO 200
C ********************** LONGITUDINAL SETUP
170 SC1=0.5*CBAP/V
THETA=THETA/ANPT
Q5=Q5*(THETA/RAD)*COS(PHI/(FRAC*ANPT)**GROSWT/10*S)
100=2
IXY=3
PI=VOG
PI=1.*/
DATA 140
DATA 150
DATA 160
DATA 170
DATA 180
DATA 190
DATA 200
DATA 210
DATA 220
DATA 230
DATA 240
DATA 250
DATA 260
DATA 270
DATA 280
DATA 290
DATA 300
DATA 310
DATA 320
DATA 330
DATA 340
DATA 350
DATA 360
DATA 370
DATA 380
DATA 390
DATA 400
DATA 410
DATA 420
DATA 430
DATA 440
DATA 450
DATA 460
DATA 470
DATA 480
DATA 490
DATA 500
DATA 510
DATA 520
DATA 530
DATA 540
DATA 550
DATA 560
DATA 570
DATA 580
DATA 590
DATA 600
DATA 610
DATA 620
DATA 630
DATA 640
DATA 650
DATA 660
DATA 670
DATA 680
DATA 690
DATA 700
APPENDIX A – Continued

AP3=;
C(1,1)=AM/RAD
C(2,1)=1/(OSCI*1/0
C(1,1)=C(1,1)/V
C(2,2)=V/(OSCI*CBAR)
C(3,1)=1
C(3,2)=V/2
DO I=1,3
I(1,1)=RAD
I(I,2)=V
I(I,J)=I(I,II)
THETA=THETA*(1,1)
C SET E=0 IF DERIVATIVE IS FIXED, OTHERWISE E=0.9
DO 220 J=1,3
E(I,J)=STAR
IF(.NOT.AA(I,J) ) E(I,J)=BLANK
220 CONTINUE
DO 220 J=1,3
K=J-3
E(I,J)=STAR
IF(.NOT.BB(I,K) ) E(I,J)=BLANK
220 CONTINUE
C************************************************************ FORM AP AND BP IF NOT READ IN
IF (APAP) GO TO 129
DO 112 J=1,MU
AP(I,J)=P1
BP(3,J)=P3
112 AP(I,J)=P3
DO 129 J=1,MU
AP(I,J)=P1
129 CONTINUE
XBN=XBN/G
500 XAX/G
C************************************************************ STORE APRIORI WEIGHTINGS
C ALL T1 VALUES
DO 510 T=1,35
CALL AZT1(APR)
DO 510 T=1,35
APR(T)=0.
510 CONTINUE
X55(T=1.6
IF(MAPR.EQ.6.1) RETURN
K=0
DO 520 T=1,35
520 CONTINUE
DO 525 J=1,MX
IF(AK) GO TO 520
APPENDIX A — Continued

K=K+1
XTS(K)=A(I,J)-AR(I,J)
&PRD(K)=APR(A(I,J)*WMAPR
525 CONTINUE
DATA2290
DATA2300
DATA2310
1001 FORMAT(1X,2F10.4)
2000 FORMAT(14HTIME INTERVAL,13,1H NOT FOUND)
2001 FORMAT(55HINPUT TIME HISTORY WITH BIASES AND SCALE FACTORS APPLI,
- 35MED AND VANE CORRECTIONS ADDED FOLLOWS.),
2003 FORMAT(1H1,26X,20Ah,13X,A trehow =,F9.1,6H IX =,F9.1,6H IX =,
- F10.1,7H IX =,F7.1,6H IX =,F9.1,6H QBAR =,F7.2,5H V =,
- F8.2/5x,4HTIME,6x,14AH,5H)
2007 FORMAT(1H1,40X,35HTOTAL NUMBER OF POINTS FOR MANEUVER,13,2H =,16)
2010 FORMAT(1X,312,I3,2X,12F8.3,F8.1,F8.3,F8.2)
RETURN
DATA2390
DATA2400
DATA2410
END
DATA2420
APPENDIX A – Continued

SUBROUTINE AGIRL

Description: Subroutine AGIRL performs the parameter estimation. Almost all the routine is skipped if NOITER = 0.

Flow chart:
APPENDIX A — Continued

Programming notes: For derivation of the form of the first and second gradients, see reference 3.

Important variables --

SUM -- contains second gradient in lower triangular and diagonal locations, and off-diagonal \textit{a priori} weightings in upper triangular. Diagonal \textit{a priori} weightings are in APRD. The first gradient appears as an extra column in SUM (the JKM\textsuperscript{th} column). The SUM matrix is printed each iteration when TEST = T.

\[ X_{JI} = \nabla_c (z_i - y_i^*) \]

\[ RIA = R^{-1}A \]

\[ RIB = R^{-1}B \]

\[ PHI1 = e^{R^{-1}A \Delta t} \]

\[ APHI = \left( \int_0^{\Delta t} e^{R^{-1}A t} dt \right) R^{-1} \]

\[ BPHI = (APHI)(B) \]

AAP, BBP -- observation matrices formed from A and AP or B and BP, with any terms for accelerometer offset from the center of gravity added. (These matrices are referred to as G and H in the derivation.)

\[ RIAP - \text{array of partial derivatives of AAP with respect to A.} \]

\[ RIAP(I,J,K) = \frac{\partial AAP(I,K)}{\partial A(J,K)} \]

\[ RIBP - \text{array of partial derivatives of BBP with respect to B.} \]

\[ RIBP(I,J,K) = \frac{\partial BBP(I,K)}{\partial B(J,K)} \]

Z, U -- measured values of observations and controls.

XT1, XT2 -- computed values for observations.

XT3 -- variable initial conditions on the states.

XT4 -- variable bias on the observations other than states.

XT5 -- difference between the estimated coefficients and the \textit{a priori} values.

PB -- solution vector for the change in the estimates of the coefficients.

MX -- number of states.

MZ -- number of observations.
SUBROUTINE AGIRL

CORE SUBROUTINE - ITERATIVE LOOP

COMMON /ALLDIM/ MAX,MIX
COMMON /CASE/ NPTS,NPTT,SPS,PRINT,LONG,LAIR,PLTM,NOI.
COMMON /DIOLR/ JKM,JKMM,JKMMI,ERPMAX,ZEROIN,XTB,BCUND,
COMMON /DIMENS/ C,CC,E,WQS,TMETN
REAL XT3(4),XT4(3),XTBK35,APPPI351,LAEELS(15)
REAL U12(B_,U23(a),BPHI(5,8),EXTRA(4),AC(5,_,BC(5,8)
LOGICAL AAeBB,TEST,ZEROINeQASKN,CORECTeLATR,ERSTOp,PUNCHD
DIMENSION A( 5,@ ), B( 5,8 De SUM(35,35),PHIZ( 5,4
AA( 5,4 ),BB( 5,8 ),RI(5.W|, APHI( E,4
AP 4,4 ),BP( 4,8 )...
APPENDIX A -- Continued

MAX = 5
CALL ASPIT(A)
CALL ASPIT(B)
CALL AMULT(RI,A,PIA)
CALL AMULT(RI,E,RIB)

C COMPUTE A_AP AND B_BP
DO W5 I=I,J
DO W5 J=I,J
AAP(I,J) = AAP(I,J) + XAN * RIA(I,J)
BBP(I,J) = BBP(I,J) + XAN * RIB(I,J)
ENDIF
ENDIF
IF(L,NOT.CORECT) GO TO 50

DO W5 J=I,MX
AAP(I,J) = AAP(I,J) + XAN * RIA(I,J)
BBP(I,J) = BBP(I,J) + XAN * RIB(I,J)
ENDIF
ENDIF
RESTART 7

CALL ABEAT(RIA,PHI1,APHI,DUM1,SUM,NEAT)
CALL AMULT(APHI,DUM1)
CALL AMULT(APHI,B,BPHI)
IF(L,NOT,TEST) GO TO 51

CALL ASPIT(PHI1)
CALL ASPIT(APHI)

DO 51 J=I,MX
DO 51 J=I,MX
PHI1(I,J) = PHI1(J,I)
ENDIF
ENDIF

DO W5 I=I,J
DO W5 I=I,J
SUM(I,J) = 0.
DO 52 I=I,J
DO 52 I=I,J
D2(I) = 0.0
C VARIABLE BIAS
IF(L,NOT.BIASKN) GO TO 3
IBIAS = JKV
DO 16 I = 1,MZ
IF(L,NOT.BIASK(I)) GO TO 16
IBIAS = IBIAS + 1
DO 15 J = 1,MZ
15 XJI(IBIAS,I,J) = 0.
XJI(IBIAS+1,MX) = 1.
16 CONTINUE
WRITE(3,1010)(LABELS(I),I=MYP1,MZ)
WRITE(3,1020)(XJI(I),I=I,MZ)
C*********************** CASE LOOP

DO 26 LM = 1,NCASE
NM1 = NPS(LM)-1
XJI(NM1,1) = JKV
XJI(NM1,2) = MK
CALL AZOT(XJI)
READ (7) IT,XI1,(UXK,1),K=1,85,EXTRA
AGIR 570
AGIR 580
AGIR 590
AGIR 600
AGIR 610
AGIR 620
AGIR 630
AGIR 640
AGIR 650
AGIR 660
AGIR 670
AGIR 680
AGIR 690
AGIR 700
AGIR 710
AGIR 720
AGIR 730
AGIR 740
AGIR 750
AGIR 760
AGIR 770
AGIR 780
AGIR 790
AGIR 800
AGIR 810
AGIR 820
AGIR 830
AGIR 840
AGIR 850
AGIR 860
AGIR 870
AGIR 880
AGIR 890
AGIR 900
AGIR 910
AGIR 920
AGIR 930
AGIR 940
AGIR 950
AGIR 960
AGIR 970
AGIR 980
AGIR 990
AGIR 1000
AGIR 1010
AGIR 1020
AGIR 1030
AGIR 1040
AGIR 1050
AGIR 1060
AGIR 1070
AGIR 1080
AGIR 1090
AGIR 1100
AGIR 1110
AGIR 1120
AGIR 1130
APPENDIX A — Continued

C

READ (7) I, X, Z, (U(K, Z), K=1, N), EXTRA
C
VAR VALUES INITIAL CONDITION
IF (.NOT. ZEROIN) GO TO 6
IC = JKM
DO 4 I = 1, MX
IF (.NOT. ZEROII)) GO TO 6
IC = IC + 1
XJ(I, I) = 1.
XT(I) = XT(I) + XT3(I)
XT2(I) = XT2(I) + XT3(I)
4 CONTINUE
IF (LM.NE.1) GO TO 6
WRITE (3, 1001) (LABELS(I), I=1, MX)
WRITE (3, 108) (XT3(I, I), I=1, MX)
6 DO 8 I=1, MX
Z(I, 1) = XT(I)
8 Z(I, 2) = XT2(I)
IF (TEST) WRITE (3, 111) (XT(I), I=1, MX)
IF (TEST) WRITE (3, 111) (XT2(I), I=1, MX)

C

TIME LOOP
C

DO 10025 J = 2, NNM1
     READ (7) I, Z, (Z(K, 3), K=1, N), EXTRA
     IF (LL.EQ.1) GO TO 203
     DO ZO1 I = T, HX
         XT12(I) = .5'* (XT1(I) + XT2(I))
         XT6(I) = XT(I)
         GO TO 205
     201 XT6(I) = XT2(I)
     GO TO 205
     203 DO 204 I = 1, MX
         XT2(I) = .5'* (Z(I, 2) + Z(I, 1))
         Z(I, 1) = Z(I, 2)
     204 Z(I, 2) = Z(I, 1)
     205 CONTINUE
     DO 206 I = 1, MU
         U2(I) = .5'* (U(I, 1) + U(I, 2))
     206 U(I, 2) = U(I, 1)
     DO 210 I = 1, MX
         XJ(I, J) = 0.
     210 XJ(I, J) = XJ(I, J)
     DO 11 I = 1, MX
         XT(I) = XT2(I)
     11 XT2(I) = 0.
     CALL AMULT (XJ, PHI, XJID1)
     CALL AMUL (XJ, PHI, XJID2)
     JK = 0.
     DO 124 J = 1, MX
     124 JK = JK + 1.
     XT2(J) = XT2(J) + PHI(J, K)*U23(JK)
     IF (BB(J, K)) GO TO 12
     JK = JK + 1.
     DO 115 I = 1, MX
         JXI(JK, I) = RI(J, J) + PHI(J, K)*U23(JK)
     115 JXI(JK, I) = JXI(JK, I)
     12 CONTINUE
Appendix A – Continued

00 14 K = 1,MX
XTZ(J) = XTZ(J) + AP(T,J,K)*XT2(J)
IF (AA(J,K)) GO TO 14
00 15 J = J + 1
DO 17 I = 1,MX
XJ(J,K,MX) = AP(T,J,K)*XT2(J)
17 CONTINUE
DO 18 K = 1,MX
XTZ(L) = XTZ(L) + AP(T,L,K)*XT2(L)
DO 19 J = 1,MZ
XJ(J,K,MZ) = XJ(J,K,MZ) + XT2(J)
19 CONTINUE
DO 20 J = 1,MZ
XJ(J,K,MZ) = XJ(J,K,MZ) + XT2(J)
20 CONTINUE
IF (TA = TRUE) WRITE (3, 101) XT2(J), J = 1,MZ
IF (TEST) GO TO 62
MIX = 8
XJI(NI,K) = XJI(NI,K) + XT2(K)
IF (XII(NI) .EQ. 0) GO TO 28
DO 27 J = 1,I
27 SUM(J,K) = SUM(J,K) + XT2(J)
28 CONTINUE
C******************************************************************************
End of time and case loops
C******************************************************************************
C******************************************************************************
Solution of system
C******************************************************************************
DO 29 I = 1,JKM
XT5(I) = XT5(I) + PM(I)
SUM1 = SUM1 + SUM(J,KM)*APRO(I)
SUM2 = SUM2 + SUM(J,JM)*APRO(I)
IM = I - 1
IF (IM .LE. 0) GO TO 28
DO 27 J = 1,IM
27 SUM(J,J) = SUM(J,J) + SUM(J,J)
28 CONTINUE
APPENDIX A – Continued

IF (TEST) CALL APIT(SUM)  AGIR2280
 IF(NEQSTOP,0, (LLL,EO,NOITER)) CALL CRAPER(SUM,APRO,MU,MZ,ERPSUM)  AGIR2290
 CALL TOLVE(SM,PM)  AGIR2300
 IF(NEQTEST) WRITE(1,57) (PRI(I),I=1,JMPS)  AGIR2310

******************************************************************************

** UPDATE COEFFICIENTS **

IJ = I 
 DO 31 I = 1, MX
 DO 31 J = 1, MU
 IF (S3(I,J) ) GO TO 30

S3(I,J) = S3(I,J) + S3(I,J)

30 CONTINUE

DO 31 J = 1, MU
 IF (S4(I,J) ) GO TO 31

S4(I,J) = S4(I,J) + S4(I,J)

31 CONTINUE

IF (.NOT.ZEROIN) GO TO 35

DO 36 I = 1, MX
 IF (.NOT.ZERO(I)) GO TO 36

XT4(I) = XT4(I) + T4(I)

36 CONTINUE

35 IF (.NOT.BIASSN) GO TO 37
 DO 38 I = 1, MZN
 IF (.NOT.BIASS(I)) GO TO 38

XT4(I) = XT4(I) + T4(I)

38 CONTINUE

37 WRITE (1,102) ILL
tf (TEST) GO TO 38

32 CONTINUE

******************************************************************************

** END OF ITERATION LOOP **

38 WRITE(1,103) LROUND
 NOITER=LL

500 WRITE(1,2003)
 CALL ASPIRAC
 CALL ASPIRBC
 WRITE(1,2006)
 DO 508 I = 1,3
 DO 507 J = 1,5

507 Aeci,I,J)*AC(I,J)*C(I,J)
 DO 508 J = 1,5
 Aeci,J)*AC(I,J)*C(I,J)
 CALL ASPIRAC
 CALL ASPIRBC
 RETURN

510 WRITE(1,2001) ERRMAX
 NOITER=LL
 CRVDECDL+ERRMAX

101 FORMAT(15X,16 ITERATION NUMBER,1X,10H COMPLETED)
102 FORMAT(15H VARIANCE RIA ,6E12.4)
103 FORMAT(1H+,10X,20H NUMBER OF UKNOWNS ,4I3/1HG,20X)
 - 39 MATRICES PER Radian RIA S IN RADIAN, RIA S IN RADIAN

59
APPENDIX A — Continued

104 FORMAT(95X,2SH WEIGHTED ERROR SUM = ,E12.4)  
105 FORMAT(7H ERRORS/IX,11E12.4)  
106 FORMAT(16H WEIGHTED ERRORS/IX,11E12.4)  
107 FORMAT(12H RBI VECTOR =,10E12.4/(12X,10E12.4))  
109 FORMAT(15H VARIABLE IC ,4IE12.4)  
110 FORMAT(37H ITERATION TERMINATING, ERROR WITHIN ,F9.6, 8H BOUND.)  
111 FORMAT(1X,7E12.4)  
120 FORMAT(1H,36X,20A4,13X,A10/1M0,10X,1SH STARTING VALUES,5X, 
121   - 6HMAC = F6.3,5X,7H ALPHA = F7.2,5X,7H PARA = F5.2,5X, 
122   - WCG = F6.3)  
1201 FORMAT(15X,21AW.13X,AIQIINO,IOX_lBMSTARTING 
1202   VALUES,BX, 
1203   - 6HMAC = F6.3,5X,7H ALPHA = F7.2,5X,7H PARA = F5.2,5X, 
1204   - WCG = F6.3)  
2001 FORMAT(48H ITERATION TERMINATING, MAXIMUM ERROR OF,E10.2, 
2002   - 9H EXCEEDED/27H INPUT TIME HISTORY FOLLOWS)  
2003 FORMAT(48H CONFIDENCE LEVELS FOR NEXT TO LAST ITERATION/ 
2004   - 5X,13H (DIMENSIONAL))  
2005 FORMAT(22H (NON-DIMENSIONAL))  
600 RETURN  
END
APPENDIX A - Continued

SUBROUTINE OUTPUT

Description: Subroutine OUTPUT provides the final output in several forms. The time histories are computed with the final derivative estimates and may be printed or written on a file for plotting. Final derivative estimates are also printed and, if requested, punched on cards. An error exit section to print the input time history is entered if PLTMAX or ERRMAX was exceeded.

Programming notes: Time history data for plotting are written on unit 8. The time histories are always computed to determine the final error sum, even if neither printout nor plots are requested. Most variable names are similar to those in subroutine AGIRL. ERRVEC contains the error sum from each iteration in AGIRL for the convergence summary.
APPENDIX A — Continued

Subroutine listing:

```plaintext
SUBROUTINE OUTPUT(D2)
  OUTP 0
  OUTP 10
  OUTP 20
  OUTP 30
  OUTP 40
  OUTP 50
  OUTP 60
  OUTP 70
  OUTP 80
  OUTP 90
  OUTP 100
  OUTP 110
  OUTP 120
  OUTP 130
  OUTP 140
  OUTP 150
  OUTP 160
  OUTP 170
  OUTP 180
  OUTP 190
  OUTP 200
  OUTP 210
  OUTP 220
  OUTP 230
  OUTP 240
  OUTP 250
  OUTP 260
  OUTP 270
  OUTP 280
  OUTP 290
  OUTP 300
  OUTP 310
  OUTP 320
  OUTP 330
  OUTP 340
  OUTP 350
  OUTP 360
  OUTP 370
  OUTP 380
  OUTP 390
  OUTP 400
  OUTP 410
  OUTP 420
  OUTP 430
  OUTP 440
  OUTP 450
  OUTP 460
  OUTP 470
  OUTP 480
  OUTP 490
  OUTP 500
  OUTP 510
  OUTP 520
  OUTP 530
  OUTP 540
  OUTP 550
  OUTP 560
```
APPENDIX A — Continued

55 BBP(I,J) = AA(I,J) * R(I,J)
56 IF (.NOT. CORRECT) GO TO 69
50 61 J = I, M
51 AAP(I,J) = AAP(I,J) + P(I,J) * X
52 BBP(I,J) = BBP(I,J) + X * A(I,J)
61 AAP(I,J) = AAP(I,J) + X * A(I,J)
62 BBP(I,J) = BBP(I,J) + X * A(I,J)
69 CONTINUE

*******************************************************************************
CASE LOOP
DO 20 LM = 1, NCASE
149 NNMI = NPTS(LM) + 1
150 READ (7) IT, X1, U1, EXTRA
151 DO 75 I = 1, M
153 X(I) = X1(I) + X1(I) + X1(I)
154 IF (.NOT. OUTPUT) GO TO 95
155 Y(I) = X6(I) + X6(I) + X6(I)
75 CONTINUE

*******************************************************************************
TIME LOOP
DO 110 IP = 2, NNMI
110 READ (7) IT, Z, U2, EXTRA
115 I = I, M
118 J = J, M
125 J = J, M
130 K = K, M

*******************************************************************************
ITEM LOOP
DO 140 Q = 1, NITEM
140 WRITE (9) Y, Z, U3, EXTRA
145 WRITE (10) X6, X1, U1, EXTRA

*******************************************************************************
APPENDIX A -- Continued

140 XT2(L) = XT2(L) + XT2(K)*AAP(LHMXtK)
00 150 J=1,MZ
150 D2(J) = D2(J) + (Z(J) - XT2(J))**2
IF(.NOT.OUTPT) GO TO 195
00 170 I=1,MZ
170 Y(I)=XT2(I)/CALIR(I)
00 191 I=1,4
191 U(I)=U(I)**RAD
IF(PLOTLY) WRITE (8) Y,Z,U,EXTRA
IF(.NOT.PRINT) GO TO 195
IF(LINE.LT. 50) GO TO 190
190 LINE = LINE + 1
WRITE(3,11000) TITLE, JULIAN
WRITE(3,2004) (LABELS(I), I=1,MZ)
END LOOPS
WRITE(3,2002) CALL ASPIT(A)
CALL ASPIT(B)
C**************************************************************************
************** PUNCHED OUTPUT AS DESIRED
IF(.NOT.PUNCH) GO TO 300
CALL PLOP(A)
CALL PLOP(R)
300 IF (.NOT.PUNCH) GO TO 400
00 310 1=1,3
310 A (I,J) = A (I,J)*C(I,J)
00 320 I=1,6
320 A (I,J) = A (I,J)*C(I,J)
TYPE=ALAT
IF(.NOT.LONG) GO TO 330
TYPE=ALON
C GETRM AND CZ (GOOD ONLY FOR 2 DEGREE OF FREEDOM WITH NO EXTRA
C CONTROLS.)
R(1,2) = R(1,2)*MACH**2
R(1,3) = R(1,3)*MACH
400 IF (.NOT.BIASKN) GO TO 209
WRITE(3,1004) (LABELS(I), I=MXP1,MZ)
WRITE(3,1003) (BIS10D(I), I=MXP1,MZ)
209 DD = D2(I) + D2(I)**2
210 ERRSUM = ERRSUM + D2(I)
210 WRITE(3,1001) ERRSUM

64
APPENDIX A -- Continued

WRITE(3,100)
WRITE(3,105)(XTI(I),I=1,MZ;
WRITE(I,106)
WRITE(I,IO5)(O2(1),I=I,MZ)
IF(NOITER°NE.O) WRITEI3,10_)(ERRVEC(I),I=i. NOITER).ERRSUM
IF{ERRSUM.LT.PLTMAX .OR. .NOT.PLOTEM| RETURN
450 WRITE{I,IOOZIPLTMAX
WRITE(3,1OOO)TITLE.JULIAN
WRITE(I.2OOW)
(LABELS(1).I=I,MZI
PLOTEM=.FALSE.
REWIND
7
DO 500 I=i,NPTT
READ
(7)
IT, Z,U,EXT_A
DO
_60
J=1,?
k60
Z(J):Z(J)ICALIB(J)
00 k70
J=l,k
_70
U(J)=U(J)_RAD
50C
WRITE(3,113)IT.Z.U
I00
FORMAT(7H ERRORS)
105 FORMAT
(1X.11E12.W)
10E
¢ORMATII6H
108 FORMAT(iHO.6ZX.6HERRORS/(IX.13FlO.2))
113 FORMAT(2X,I12,11FIO,W)
1000 FORMAT(1H1,26X,2_AW, 13X, AlO/)
1001 FORMAT(55HiDATA
WILL
NOT
BE
PLOTTED BECAUSE
THE
ERROR SUM
EXCEEDS THE
MAXIMUM
PERMISSIBLE, Ei_.2/
OUTP1960
1002 FORMAT(15HVARIABLE
BIAS .4E12.4)
OUTP1990
1003 FORMAT(AW,1X,BAW, A3,4FIO.3) OUTP2000
- 24h THE
MAXIMUM PERMISSIBLE,Elg.2/
27hINPUT TIME HISTORY
OUTP1800
1000 FORMAT(1H1,26X,26HOUTPUT TIME HISTORY)
OUTP2010
RETURN
OUTP2020
END OUTP2030
Description: Subroutine THPLOT plots measured and computed time histories of observations and measured time histories of controls and extra signals.

Programming notes: The comment cards show how to decrease the run time in some instances at the cost of some storage. At present, two time histories at a time are read from the disk and plotted. Dimensions may be increased as indicated to permit more than two to be handled simultaneously, resulting in fewer disk accesses. With reasonably efficient disk units, the saving is not a significant portion of the program execution time. The limitation of 1000 points per maneuver arises from the dimensioning of X,XX,XXX and TIME as 1002. (The extra two locations are used for scaling information.) Program size may be reduced or the permissible maneuver length increased by changing this value. The special treatment of the title (plotting groups of four characters in a DO loop instead of using only one call to SYMBOL) is needed for compatibility with machines that use different word lengths.
SUBROUTINE THPLOTFIRST)

C PLOTS TIME HISTORIES

COMMON /BUFF/ BUFFER,YO,THGT
COMMON /LONG/ LENV,NCASE,NPTS,NPI,SPS,PRINT,ERROR,LAST,NO1,
         - ADF,DISP,NAPR,MCAS,WHAPF,ERSUM,RAF
COMMON /TPLOTT/ XMAX,DMIN,XCUB,XMIN,NPTS,TIMESC,NC
COMMON /HEAD/ LABELS,TITLE,JULIAN

DIMENSION OCMAX(_),DCMIN(_),XMIX7),XMIN7),NPTS15),TIME(1002),
         - XXX(1002,E),X(1002,F),XX(1002,T),Z(7),DC(8),ZZ(7),
         - 0LAlELS15),MBGO(30)

LOGICAL LONG,FIRST,LAST

EQUIVALENCE (X(I,1),XXX(I,1),XX(I,1),XXX(I,2),Z,DC(8),
         - ZZ(7),L_ALNS15),MBGO(30)

DATA M_COI3HDEG'HOIS'3HF/S',HDIS2HDEG,3HDeS,2_HWD/S2, THPL _50
         - 3HOES,HRT/S,THPL 160
         - HPSF,3HCEG,2_3HCIS_3HDEG,3HGeS,2_WHD/S2, THPL 170
         - 3HOES,HRT/S, THPL 180

C ***_*w_* FOR A OIRECT DECREASE IN RUN TIME
C ***_*w_* FOR A OIRECT DECREASE IN RUN TIME AT THE COST OF
C STORAGE, NCH MAY BE INCREASED (UP TO 7), THEN THE
C FOLLOWING DIMENSIONS AND EQUIVALENCE MUST BE
C CHANGED.

C *************** EQUIVALENCE (XX(1,1),XXX(1,NCCH))

NBUF=102W
TIMESC2=TIMESC2/1.
XO=5.
HGT=.01
TSI=SPS*TIMSC2
TIIME-=MAXI(TSI/20.,1.1)
REWIND 8

IF (.NOT.FIRST) GO TO 10
CALL PLOTS(HUFFER,MUF,13)
CALL FACTOR(RATIO)
YO=I_,
IF(RATIO.EQ.10) YO=9.5
THGT=.I_/RAT_C

10 Y75=YO+.375

IF (LONG) GO TO 50
NO 20 I=1,15
20 MBCO(I)=MBCO(I+1)
50 NO 20 I=1,NCASE
CALL PLOT(XO,O.,-3)
CALL SYMBOL(-I*5,YO,THGT,TITLE(1),270.,_)
DO 55 J=_,20
55 CALL SYMBOL(-I*5,999.,THGT,TITLE(J),270,,8_
IF(NCASE.EQ.I) GO TO 57
CALL SYMBOL(-2°,YO,THGT,FLOAT(I),270.,°i)
CALL PLTDAT{_.5,YO)

C =**_=_*_*_ FORM TIME VECTOR AND PLOT TIME AXIS
NOPTS=NPTS12)
NPI=NOPTS+1

C *************** FOR TIME VECTOR AND PLOT TIME AXIS

APPENDIX A – Continued
APPENDIX A -- Continued

C *************** PLOT STATE TIME HISTORIES
ICHAN=0
NCHAN=NCHAN
DO 120 K=1,NOPTS
IF(K .EQ. 0) GO TO 90
ICHANG=ICHANG+NCHAN
IF(ICHANG+NCHAN .LE.,2) GO TO 70
IF(ICHANG .GE.,2) GO TO 125
NCHAN=NCHAN+ICHANG
70 REWIND 5
IF(I .EQ. 1) GO TO 90
DO 100 J=1,NOPTS
REAO
103 L=T,NCHAN
X(J,L)=ZIL_ICHANC)
100 XX(J,L)=ZZ(L*ICCHAN)
DO 110 L=T,NCHAN
ICHAN=ICHAN+L
SCAL=(XMAX(ICHAN)-XMIN(ICHAN))**5
XMN=XMIN(ICHAN)
IF(SCAL NE.0) GO TO 105
CALL SCALES(X(I,L),2.,NOPTS,,FALSE.)
CALL SCALES(XX(_,L),2,NOPTS,)
SCAL=AMAXIIXX(NP2,L),X(NP2,L))
IF(SCAL EO.-Qqg.) GO TO 110
_XMN=X(NPt,L)
IF(XX(NP2,L),GT,X(NO2,L)
XMN=_X{NP_,L)
105 CALL SYH90L(XORG,1.,Y75,,%25,L_9ELS(ICHAN),O,)
CALL AXIS(XORG,YO,MQCD(ICHAN,4,2,,3,0,MN,SCAL)
X(NPI,L)=XMN-XORG4SCAL
XX[NPL,L]=X(NPL,L)
CALL LINES(X(i,L),TIME,NOPTS,t, 1.11
CALL LINES(XX(L,L),TIME,NOPTS,ITIMIN.,2,75)
XORG=XORG+2.5
110 CONTINUE
120 CONTINUE
C *************** PLOT CONTROL TIME HISTORIES
125 NCHAN=NCHAN*2
IF(INCH2,GT,NC) NCHAN=NC
ICHANG=ICHANG
DO 140 K=1,2
ICHANG=ICCHAN,NCHAN=ICHANG
IF(ICHANG .GE.,NC) GO TO 170
[I]CHANG=NCHAN,GT,NC) NCHAN=NC=ICHANG;
REWIND 5

CPlT 570
CPlT 580
CPlT 590
CPlT 600
CPlT 610
CPlT 620
CPlT 630
CPlT 640
CPlT 650
CPlT 660
CPlT 670
CPlT 680
CPlT 690
CPlT 700
CPlT 710
CPlT 720
CPlT 730
CPlT 740
CPlT 750
CPlT 760
CPlT 770
CPlT 780
CPlT 790
CPlT 800
CPlT 810
CPlT 820
CPlT 830
CPlT 840
CPlT 850
CPlT 860
CPlT 870
CPlT 880
CPlT 890
CPlT 900
CPlT 910
CPlT 920
CPlT 930
CPlT 940
CPlT 950
CPlT 960
CPlT 970
CPlT 980
CPlT 990
CPlT 990
CPlT 990
CPlT 1000
CPlT 1001
CPlT 1020
CPlT 1030
CPlT 1040
CPlT 1050
CPlT 1060
CPlT 1070
CPlT 1080
CPlT 1090
CPlT 1100
CPlT 1110
CPlT 1120
CPlT 1130

68
APPENDIX A — Continued

IF(J.EQ.1) GO TO 140
DO 130 J=1,NIP
DO 130 READ (8)
140 DO 155 J=1,NOPTS
READ (8) ZZ,Z,OC
DO 155 L=1,NCH2
J=L+ICHAN0
NZ=J+7
SCAL=(OCMAX(J)-OCMIN(J))*5
OCMN=OCMIN(J)
IF(SCAL.EQ.0) GO TO 155
CALL SCALED(XXX(1,L),L,NOPTS,.TRUE.,
IF(XXX(NP2,L),EQ.,-999.) GO TO 165
IF(XXX(NP2,L),GE.,0.) AND J.EQ.74 GO TO 153
XXX(NP2,L)=10.
XXX(NP1,L)=10.
153 CALL SYMBOL (XCRG+1.,Y75.,.125.,LABELS(M),3,4)
CALL AXIS(XORG,Y0,MBCO(M),4,Z2,.5,OCMN,SCAL)
XXX(NP1,L)=OCMN*XORG+SCAL
XXX(NP1,L)=SCAL
CALL LINES(XXX(1,L),TIME,NOPTS,1,0,1)
XORG=XORG+2.5
160 CONTINUE
170 NIP=NIP*NOPTS
XXX(NP2,L)=XORG+2.
200 CONTINUE
CALL PLOT(XO,0.,-3)
RETURN
END
APPENDIX A — Continued

SUBROUTINE APRPLT

Description: Subroutine APRPLT plots the variation of the derivatives with \textit{a priori} weighting. It may be used when the \textit{a priori} variation option is active. The information to be plotted is in the matrix STORE.

Subroutine listing:

```fortran
SUBROUTINE APRPLT(STORE,AA,RR,APPR,WHOLD,WFAC,LONG,FIRST,LAST,
                   NAPR,WHOLO,WFACT,LONG,FIRST,LAST,
                   REC,DEP,4,HDLAG1,STORE,NAPR,STORE,NAPR,STORE)

DESCRIPTION: Subroutine APRPLT plots the variation of the derivatives with
\textit{a priori} weighting. It may be used when the \textit{a priori} variation option is active.
The information to be plotted is in the matrix STORE.

Subroutine listing:

```
CALL LINES(STORE(1, JK), WMAPR, NPT, 1, 1)  
CALL PLOT(3.5, 0., -3)  
100 CONTINUE  
DO 200 J = 1, 3  
IF(AA(I, J)) GO TO 200  
JK = JK + 1  
CALL SCALES(STORE(1, JK), 3., NPT, .FALSE.)  
DEP = ALABLA(I, J)  
IF(LONG) DEP = ALABLO(I, J)  
CALL AXIS(3., 0., DEP, 4, 3, 0., STORE(NPT1, JK), STORE(NPT2, JK))  
CALL LINES(STORE(1, JK), WMAPR, NPT, 1, 1)  
CALL PLOT(3.5, 0., -3)  
ZOO CONTINUE  
WMAPR(P) = WMAPR(NPT1)  
WMAPR(NPT2) = WMAPR(NPT2)  
NPT2 = NPT1  
NPT1 = NPT  
NPT = NAPR  
JK = JK + 1  
CALL SCALES(STORE(1, JK), 3., NPT, .FALSE.)  
CALL AXIS(3., 0., DEP, 5, 5, 0., STORE(NPT1, JK), STORE(NPT2, JK))  
CALL LINES(STORE(1, JK), WMAPR, NPT, 1, 1)  
CALL PLOT(3.5, 0., -3)  
RETURN  
END
APPENDIX A — Continued

SUBROUTINE MATLD

Description: Subroutine MATLD reads matrices from cards and identifies the matrices.

Programming notes: ABC contains the names of the matrices that may be read in. The program compares the name read with elements of ABC to determine which matrix is being input. The characters END are taken as an indication that this is the last case; any other word not identifiable as a valid matrix name signals the end of a case, implying more cases to follow. The values of ILD and ABC indicate the status of the matrix input.

Subroutine listing:

```
SUBROUTINE MATLD(HATRX,ADC,ILO) MAIL 0
C
C LOADS IN MATRICES - DETERMINES WHICH MATRIX IS BEING READ
C
C MATLD - Passes status information back to REDIT
C
REAL MATRX(8,8),ABC(12) MAIL 50
DATA EN01,HENOI MATL 60
ILD:-ggg MATL 70
READ (1,1000) MATRX(8,3),II,JJ
IF(MATRX(8,3),EQ.VEND) RETURN
DO 10 I=1,12
IF(MATRX(8,3),NE.ABC(I)) GO TO 10
ABC(I)=99999.
ILD=I
GO TO 20
10 CONTINUE
ILD=-ILD
GO TO 25
C
DIAGONAL MATRIX
ILD=ILD
CALL AZOT(MATRX)
READ (1,1001) (MATRX(I,I),I=1,II)
RETURN
C
C FULL MATRIX
25 MATRX(8,2)=JJ
DO 30 I=1,II
30 READ (1,1001) (MATRX(I,J),J=1,JJ)
RETURN
C
1000 FORMAT(4X,I2,1T0)
1001 FORMAT(8F10.4)
END
```
APPENDIX A — Continued

SUBROUTINE MAK

Description: Subroutine MAK moves an input matrix from its temporary location in MATRIX to its proper location in X. Subroutines MATLD and EDIT have determined what the proper location is for each matrix.

Subroutine listing:

```fortran
SUBROUTINE MAK(X,MATRX,MAX)
REAL X(MAX,1),MATRX(8,8)
CALL ASPIT(MATRX)
X(MAX,1)=MATRX(8,1)
X(MAX,2)=MATRX(8,2)
II=MATRX(8,1)
JJ=MATRX(8,2)
DO 10 I=1,II
  DO 10 J=1,JJ
    X(I,J)=MATRX(I,J)
10 RETURN
END

MAK 0
MAK 10
MAK 20
MAK 30
MAK 40
MAK 50
MAK 60
MAK 70
MAK 80
MAK 90
MAK 100
MAK 110
MAK 120
```
APPENDIX A — Continued

SUBROUTINE DERIV

Description: Subroutine DERIV prints dimensional and nondimensional derivatives with labels. Arrays E and EE contain the characters " " or "*" to indicate, when printed, that a particular derivative is either varying or fixed, respectively.

Subroutine listing:
APPENDIX A – Continued

SUBROUTINE Cramer

Description: Subroutine Cramer computes confidence levels based on Cramer-Rao bounds.

Programming notes: The comment cards trace the steps of the subroutine. Note the manipulation of the SUM matrix required to store the second gradient (Hessian) with the a priori terms included, while also using the second gradient without the a priori terms for the confidence level computation.

Subroutine listing:
APPENDIX A — Continued

IF(.NOT.AA(I,4)) L=L+1
71 CONTINUE

C****************************************** RESTORE COMPLETE HESSIAN TO LOWER TRIANGULAR PART

SUM(JKMM1,JKMM1)=APRO(JKMM1)
DO 80 I=1,JK2
  SUM(I,I)=APRO(I)
  IP1=I+1
  DO 80 J=IP1,JKMM1
    SUM(J,I)=SUM(I,J)
80 RETURN
END

CRAM 570
CRAM 580
CRAM 590
CRAM 610
CRAM 620
CRAM 630
CRAM 640
CRAM 650
CRAM 660
CRAM 670
APPENDIX A — Continued

SUBROUTINE AEAT

Description: Subroutine AEAT computes $e^{A\Delta t}$ and $\int_0^{\Delta t} e^{At} dt$ using the Taylor series expansion.

Programming notes: The computational method used when $NEAT \neq 0$ is described in the NAMELIST option NEAT (item (26), p. 15). The two matrices desired are returned as PHI and APHI. A2 and A3 are temporary scratch storage.

Subroutine listing:

```fortran
SUBROUTINE AEAT(A,IT,PHI,APHI,A2,A3,NEAT)  AEAT 0
  COMMON /ALLOIM/ MAX,MIX  AEAT 10
  DIMENSION A(1),PHI(1),A2(1),APHI(1),A3(1)  AEAT 20
  MAX2 = MAX * 2  AEAT 30
  II=MAX(MAX)  AEAT 40
  PHI(MAX) = A(MAX)  AEAT 50
  PHI(MAX2) = A(MAX)  AEAT 60
  T=TT/(2.**NEAT)  AEAT 70
  CALL ASET(PHI)  AEAT 80
  CALL AMAK(E,APHI,PHI)  AEAT 90
  MI=-MAX  AEAT 100
  CALL AOOE(AI,PHI)  AEAT 110
  G = 1.0  AEAT 120
  DO 1 I=1,II  AEAT 130
    MI=MI+1  AEAT 140
    PHI(MI) = I  AEAT 150
  CONTINUE  AEAT 160
  CALL AMAK(E,PHI)  AEAT 170
  CALL AMULT(A2,APHI,P)  AEAT 180
  MI=-MAX  AEAT 190
  DO 2 I=1,II  AEAT 200
    HI=HI+MI  AEAT 210
    A2(MI+J) = PHI(MI)  AEAT 220
  CONTINUE  AEAT 230
  IF (NEAT.EQ.0) RETURN  AEAT 240
  CALL AMAK(E,PHI)  AEAT 250
  CALL AMULT(A2,PHI)  AEAT 260
  MI=-MAX  AEAT 270
  DO 3 I=1,II  AEAT 280
    MI=MI+1  AEAT 290
  CONTINUE  AEAT 300
  CALL AOOE(A2,APHI,P)  AEAT 310
  CALL AMAK(E,PHI)  AEAT 320
  CALL AMULT(A2,PHI)  AEAT 330
  MI=-MAX  AEAT 340
  DO 4 I=1,II  AEAT 350
    PHI(MI) = I  AEAT 360
  CONTINUE  AEAT 370
  RETURN  AEAT 380
END  AEAT 390
```
APPENDIX A — Continued

SUBROUTINE AMULT

Description: Subroutine AMULT computes \( C = A \times B \). The quantity \( C \) cannot be the same matrix as either \( A \) or \( B \).

Subroutine listing:

```fortran
SUBROUTINE AMULT(A, B, C)
COMMON /ALLOIM/ MAX, MIX
REAL A(I1,J1), B(J1), C(I1)
MAX2=MAX*X
MIX2=MIX*X
II=1,MAX1
JJ=A(MAX1)
C(MAX1)=A(MAX1)
K=K(MAX1)
C(MAX1)=B(K)
JE=JJ+1*MAX1
KE=1*K+1*MAX1
DO 20 I=1,II
JEND=JE-I
JEND=JE+I
L=1
DO 20 K=I,JEND,MAX1
C(K)=C(K0
J=N
DO 10 J=1,JEND,MAX1
C(K)=A(J)*B(J)+C(K)
10 J=J+1
20 L=L+MIX
RETURN
END
```

AMUL 0
AMUL 10
AMUL 20
AMUL 30
AMUL 40
AMUL 50
AMUL 60
AMUL 70
AMUL 80
AMUL 90
AMUL 100
AMUL 110
AMUL 120
AMUL 130
AMUL 140
AMUL 150
AMUL 160
AMUL 170
AMUL 180
AMUL 190
AMUL 200
AMUL 210
AMUL 220
AMUL 230
AMUL 240
APPENDIX A – Continued

SUBROUTINE DMULT

Description: Subroutine DMULT multiplies XJI by a diagonal matrix D1.

Subroutine listing:

```fortran
SUBROUTINE DMULT(XJI,DI,XJIDI,JKM,MZ)
    REAL XJI(3B,B),XJIDXK35,7),DI(_,7)
    DO 10 I=1,MAX
    DO 20 J=1,JKM
    10 XJIDI(J,I)=XJI(J,I)*DI(I,I)
END
```

SUBROUTINE SUMULT

Description: Subroutine SUMULT adds the term XJID1*XJI * to the SUM matrix. Only the lower triangular elements are accumulated because the result must always be symmetrical.

Subroutine listing:

```fortran
SUBROUTINE SUMULT(XJI, XJIOI,SUM,JKM,MZ)
    REAL XJI(35,B),XJIDi(35,7),SUM(35,35)
    DO 10 I=1,JKM
    DO 20 J=I,JKM
    10 SUM(I,J)=SUM(I,J)+XJIDI(I,K)*XJI(J,K)
END
```
APPENDIX A — Continued

SUBROUTINE PLOP

Description: Subroutine PLOP punches a matrix on cards.

Subroutine listing:

```fortran
SUBROUTINE PLOP(X)
COMMON /ALLOIM/ MAX,MIX
DIMENSION X(1)
102 FORMAT (8F10.6)
103 FORMAT(A4,4X,I2,12.E16)
MAX2=MAX+MAX
MAX3=MAX2+MAX
II=X(MAX)
JJ=X(MAX2)
WRITE(2,103)X(MAX3),II,JJ
KE=(JJ-II)*MAX
DO 1 I=II,II
KENO= I+KE
1 WRITE(2,102) X(K),K=I,KEND,MAX
RETURN
END
```

SUBROUTINE ASPIT

Description: Subroutine ASPIT prints a matrix.

Subroutine listing:

```fortran
SUBROUTINE ASPIT(W)
C WRITES OUT MATRICES
COMMON /ALLOIM/ MAX,MIX
DIMENSION X(1)
100 FORMAT(A4,33X,I3,W BY,I3)
101 FORMAT(12X,10E12.4)
MAX2 = MAX * 2
MAX3=MAX2+MAX
II=X(MAX)
JJ=X(MAX2)
WRITE(3,100)W(MAX3),II,JJ
KE=(JJ-II)*MAX
DO 1 I=II,II
KENO= I+KE
1 WRITE(3,101) X(K),K=I,KEND,MAX
RETURN
END
```
APPENDIX A – Continued

SUBROUTINE AADD

Description: Subroutine AADD adds scalar multiples of two matrices.
Z = g*X + h*Y with g = 1.

Subroutine listing:

```
SUBROUTINE AADD (G,X,H,Y,Z)
   COMMON /ALLDIM/ MAX,MIX
   INTEGER XI(1),Y(1),Z(1)
   MAX2 = MAX + 2
   JJ = X(MAX)
   JEND= (JJ-1)*MAX*1
   II=II-1
   DO 53 J=1,JEND,MAX
       KEND=J*II
       DO 54 K=J,MAX
           Z(K)=X(K)+H*Y(K)
       54 K=J,KEND
63 Z(MAX)=X(MAX)
   Z(MAX2)=X(MAX2)
RETURN
END
```

SUBROUTINE AZOT

Description: Subroutine AZOT sets all elements of a matrix to 0.

Subroutine listing:

```
SUBROUTINE AZOT(X)
   COMMON /ALLOIM/ MAX,MIX
   INTEGER XI(1)
   MAX2 = MAX * 2
   II=II-1
   JJ=JJ-1
   LEND=JJ*II
   DO 55 L=1,LEND,MAX
       KEND=L+II
       DO 56 K=L,KEND
           X(K)=0
56 K=L,KEND
55 L=1,LEND
RETURN
END
```
APPENDIX A — Continued

SUBROUTINE AMAKE

Description: Subroutine AMAKE moves the matrix Y into X.

Subroutine listing:

```fortran
SUBROUTINE AMAKE(X,Y)
COMMON /ALLDIM/ MAX,MIX
DIMENSION K(I), Y(I)
MAX2 = MAX * 2
(/IMAXLYMAX)-1,
JMAX=MAX2-1,
LEND=JIMAX+1
DO 1 L=1,LEND,MIX
KEND=L+I
00 1 X(L)=Y(L)
10 X(MAX)=Y(MAX)
20 AMAX()=Y(MAX2)
X(MAX2)=Y(MAX2)
RETURN
END
```

SUBROUTINE INV

Description: Subroutine INV inverts a general matrix in place.

Programming notes: Gauss elimination is used here; there is no pivoting, since this subroutine will be called only for a well-conditioned, near-diagonal matrix (the R matrix). See reference 9 for a discussion of this method.

Subroutine listing:

```fortran
SUBROUTINE INV(A,MAX)
C INVERTS A GENERAL MATRIX IN PLACE
C NO PIVOTING (DIAGONAL ELEMENTS MUST BE NON-ZERO)
DIMENSION A(MAX,1)
N=MAX,1
BIGA=A(K,K)
DO 59 I=1,N
IF(I.EQ.K) GO TO 50
A(I,K)=-A(I,K)/BIGA
50 CONTINUE
DO 60 I=1,N
IF(I.EQ.K) GO TO 60
A(I,J)=A(I,J)+A(I,K)*A(K,J)
60 CONTINUE
80 A(K,K)=1./BIGA
RETURN
END
```

82
Description: Subroutine SOLVE solves the system of linear equations, $Ax = b$, where $A$ is symmetrical. It uses Cholesky's matrix decomposition. (See programming notes for subroutine REDUCE.) Only the lower triangular and diagonal elements of $A$ are used.

Programming notes: The $b$ vector is assumed to be stored as the $N + 1$ column of $A$, where $N$ is the dimension of the system.

Subroutine listing:

```fortran
SUBROUTINE SOLVE(A,X)

REAL A(35,1), X(35)
CALL REDUCE(A)
N = A(1,1)
N1 = N-1
N2 = N-1

C*************** MULTIPLY (L)*(B)
D0 7 1=2,N
V(I) = A(I,1)
I1 = I-1
D0 7 J = 1, I1
C = A(I,J) / A(I,I)
X(I) = A(I,N2) / A(I,I)
X(I) = X(I) + A(J,I) * A(I,N2)
X(N) = A(N,N2)
RETURN
END
```

83
APPENDIX A – Continued

SUBROUTINE DIAGIN

Description: Subroutine DIAGIN obtains the diagonal elements of the inverse of a symmetric matrix. It uses Cholesky's decomposition of the matrix. (See subroutine REDUCE programing notes.)

Subroutine listing:

```
SUBROUTINE DIAGIN(A)

REAL A(35,1)
CALL REDUCE(A)
N=A(35,1)
DO 90 I=1,N
  A(I,I)=1./A(I,I)
  IP=I+1
  DO 90 J=IP,N
    A(I,J)=A(I,J)*A(J,J)"**2/A(J,J)
90   A(N,N)=1./A(N,N)
RETURN
END
```

84
APPENDIX A — Continued

SUBROUTINE REDUCE

Description: Subroutine REDUCE factors a symmetric matrix A by Cholesky's matrix decomposition.

Programming notes: The matrix is factored into $L^{-1}DL^{-1}^T$, where L is the lower diagonal with unity diagonal elements, and D is diagonal. The lower diagonal, L, is returned in the lower triangular locations of A, except for the diagonal locations, which contain D.

Subroutine listing:

```fortran
SUBROUTINE REDUCE(A)
C REDUCES SYMMETRIC MATRIX A STORED IN LOWER TRIANGULAR LOCATIONS
C TO THE FORM (LI)^* (DI) * (LI)^* WHERE L IS A LOWER TRIANGULAR MATRIX,
C WITH UNITY DIAGONAL TERMS. D IS A DIAGONAL MATRIX, I DENOTES INVERSE AND T TRANSPOSE
C
REAL A(35,1)
N=A(35,1)
NMI=N-1
DO 20 K=1,NMI
KP1=K+1
KM1=K-1
AKKI=1./A(K,K)
DO 20 I=KP1,N
AKKI=A(I,K)*AKKI
DO 10 J=I,N
A(I,J)=A(I,J)-AKKI*A(J,K)
A(I,K)=AKKI
IF(KM1.EQ.0) GO TO 20
DO 15 J=1,KM1
15 A(I,J)=A(I,J)-AKKI*A(J,K)
10 CONTINUE
C L IS NOW STORED IN LOWER TRIANGULAR PART OF A
C EXCEPT FOR DIAGONAL, WHICH CONTAINS 0
RETURN
END
```
APPENDIX A — Continued

SUBROUTINE SCALES

Description: Subroutine SCALES determines scales for plotting the vector X on an axis S inches long. If the formal parameter ZERO is true, 0 must be included in the scale.

Programming notes: The minimum value on the axis is returned in location X(N + 1), and the scale in units per inch is returned in location X(N + 2). The only scales permitted are 2, 4, and 10 units per inch times a multiple of 10. A-999. is returned to indicate that all values of X are the same.

Subroutine listing:

```fortran
SUBROUTINE SCALES(X, S, N, ZERO)
LOGICAL ZERO
REAL X(1), FAC(3)
DATA FAC/2., 4., 10./
XMAX=X(1)
XMIN=X(1)
IF (.NOT. ZERO) GO TO 10
XMAX=0.
XMIN=0.
10 DO 20 I=1, N
XMAX=MAX(XMAX, X(I))
20 XMIN=MIN(XMIN, X(I))
AXMAX=XMAX-XMIN
IF (AXMAX.GT.0.) GO TO 30
SCALE=-999.
GO TO 100
30 B=AXMAX/SCALE
J=IFIX(B) IF (LOGICAL (R))
FAC=10.**J
B=B/FAC
DO 50 I=1, J
SCALE=SCALE*FAC(I)
AMIN=AMIN+MOD(AMIN, SCALE)
IF (AMIN.GT.XMIN) AMIN=AMIN-SCALE
IF (XMAX-AMIN).LE.SCALE GO TO 100
50 CONTINUE
SCALE=SCALE/FAC(J)
AMIN=AMIN-SCALE
IF (AMIN.GT.XMIN) AMIN=AMIN-SCALE
100 X(N+1)=AMIN
X(N+2)=SCALE
RETURN
END
```
APPENDIX A — Continued

SUBROUTINE LINES

Description: Subroutine LINES plots solid or dashed lines or symbols of the X-axis versus the Y-axis.

Programming notes: The quantities X and Y are assumed to have scaling information in locations NPT + 1 and NPT + 2 as placed there by subroutine SCALES or other sources. Every ISKIP point of the data is used, and the sign of ISKIP determines whether the plot is to be made starting from the lower numbered locations in the arrays (positive sign) or the higher numbered locations (negative sign). If JSKIP = 0, a solid line is plotted; if positive, a solid line is plotted with symbols every JSKIP th point. If JSKIP = -1, only symbols are plotted. A dashed line may be plotted using JSKIP = -2. The quantity L indicates the symbol to be plotted if relevant, and HGT gives its height.

Subroutine listing:

```fortran
SUBROUTINE LINES(X,Y,NPT,ISKIP,JSKIP,L)
  REAL X(NPT),Y(NPT)
  LOGICAL SYMB
  COMMON /LINCOM/ HGT
  IF(ABS(HGT-.5).GE.5) HGT=0.7
  XMIN=X(NPT+1)
  YMIN=Y(NPT+1)
  OX=X(NPT+2)
  OY=Y(NPT+2)
  IS=IABS(ISKIP)
  N=(NPT-I)/IS
  NA=I
  IF(ISKIP.LT.3) NA=IS*(N-I)+I
  JMOD=MAX3(ABS(JSKIP),I)*IS
  SYMB=.TRUE.
  IF(JSKIP.EQ.-1) SYMB=.FALSE.
  IL=-2
  IF(JSKIP.LT.0) IL=-1
  CALL PLOT((X(I)-XMIN)/OX,(Y(I)-YMIN)/OY,3)
  GO TO 30
  IF(SYMB.AND.MOD(NA+1,JMOD).EQ.0) GO TO 30
  CALL PLOT((X(NA)-XMIN)/OX,(Y(NA)-YMIN)/OY,HGT,0,IL)
  NA=NA+ISKIP
  RETURN
END
```
APPENDIX A – Continued

SUBROUTINE PLTDAT

Description: Subroutine PLTDAT is used to identify plots. It is machine specific for the date and time software. The subroutine may be altered to reflect the form of plot identification desired (or a null subroutine may be used).

Subroutine listing:

```fortran
SUBROUTINE PLTDAT(X,Y)
C PLOTS DATE AND TIME FOR PLOT IDENTIFICATION
C MACHINE SPECIFIC FOR DATE AND TIME SOFTWARE
CALL SYMBOL(YY,1,DATE(JULIAN),1,1)
CALL SYMBOL(99,YY,1,TIME(SECOND),1,1)
RETURN
END

FUNCTION TIME

Description: FUNCTION TIME is a dummy function to substitute for the TIME function available on CDC 6000/7000 systems. If using such a system, FUNCTION TIME may be removed; for other systems it may be rewritten to properly access the system time, or it may be retained. It is called only from subroutine PLTDAT.

Function listing:

```fortran
FUNCTION TIME(ARG)
C DUMMY SUBROUTINE IF TIME NOT AVAILABLE
DATA BLNK/1H /
ARG=BLNK
TIME=BLNK
RETURN
END

FUNCTION DATE

Description: FUNCTION DATE is a dummy function to substitute for the DATE function available on CDC 6000/7000 systems. As with FUNCTION TIME, FUNCTION DATE should be removed if using such a system and should be either rewritten or retained when used on other systems. It is called from subroutine PLTDAT and EDIT.

Function listing:

```fortran
FUNCTION DATE(ARG)
C DUMMY SUBROUTINE IF DATE NOT AVAILABLE
DATA BLNK/1H /
ARG=BLNK
DATE=BLNK
RETURN
END
```
APPENDIX A — Continued

ASSEMBLER LANGUAGE SUBROUTINES

Since the program spends a large part of its time in matrix multiplication, the execution time may be reduced considerably by writing the two small matrix multiplication subroutines AMULT and SUMULT in efficient assembler language code. In the following listings these two subroutines are written in COMPASS for use on CDC systems. These particular subroutines should be usable on any 6000 or 7000 series CDC system with the FORTRAN EXTENDED compiler. (The RUN compiler has different subroutine linkage conventions.) The use of these subroutines in place of the FORTRAN routines will speed up the program by 20 percent to 25 percent. If extensive use on other systems is anticipated, it may be advisable to make assembler versions for them. Some FORTRAN optimizers may be efficient enough to negate the gain realized; the 20 percent to 25 percent improvement mentioned, however, is referenced to the highest level of optimization available with a CDC FORTRAN 4.0 compiler.
ASSEMBLY SUBROUTINE AMULT

Subroutine listing:

```assembly
IDENT AMULT
ENTRY AMULT
USE CODE
USE /ALLOC/2
MAX 1
MAM 1
ASS 1
SUB 1
USE CODE
AMULT EQ AMULT+4000000
SA4 MAX
SA6 MAX
SA2 X4 MAX
SA1 X5 MAX
SA1 A1-B1 B
SA1 A2-B1 C
SA4 E3+B1 MAX-1
SA6 A1+B1 MAM-1
SA4 Y1+B4 & (MAX+1)
SA5 X2+B6 Q(MAM+1)
X6 Yx
SA6 Y3+B4 =C(MAX +1)
UX4 A4+B4
SB5 X4+E1 II=1
UX5 B6+X5
LK5 B6, X5
SB7 X5 JJ
SA5 A5+B5 (MAM,2)
XK6 X5
SA6 A6+N2 =C(MAX+2)
UX6 P4, X5
LK6 P4, X0 KK
SB8 P1 J=C FOR FIRST K
SB4 A1 J=I+1
SK7 Y1+B4
SB6 P0 J=0
SK6 P0
SA4 Y7 A
SA5 X2+B6 A
FX4 X0+X5 APB
FX6 Yx+X4 +C
SK7 Y2+B2 STEP A
SB6 A6+B1 J=J+1
LT A6+B7 LOOPJ
NX6 Y6
SA6 X3+B4 =C
LT A4+B5+LCOPIK
SB4 P1 J=0
SK1 X0+B1 K=K+1
SK2 X2+B3 STEP B
SK3 X3+B2 STEP C
N7 X0, LOOPX
EQ AMULT
END
```
Subroutine listing:

```
IDENT SUMULT
ENTRY SUMULT
USE /ALLODIM/
RE 2
USE CODE
SUMULT EQ SUMULT+4090000A
SB  1  I  I
SA5 MAX +
SB2 X5
SA7 XI XJID
SA1 AI+1 XJID1
SA2 AI+1 SUM
SA4 A2+1
SA3 X4
SB4 X3 JKM
SA4 A4+B1
SB7 X4 M2*MAX
SB6 B6
SB3 X5+B1 MAX+1
SXB X7+B5 LOC(XJI(K,J))
SA6 X2+B5 SUMIK,I
BX X5
SB6 B6 J=I
LOOPJ SA4 X5+B6 XJI(K,J)
SA5 X1+B6 XJI(J,J)
FX5 X4*X5
FX6 X5+B6
SB6 B6+B2 J=J+1
NX6 X6
LT B6+B7+LOOPJ
SA6 X2+B5 XUM(I,K)
SB5 B5+B1 K=K+1
LT B5+B4+LOOPIK
SB5 B0 RESTART K
SB4 B4-B1 LOWER K LIMIT I=I+1
SX1 X1+B1 STEP LOC(XJI(I,J))
SX2 X2+B3 STEP SUM LOC TO DIAGONAL
SX7 X7+B1
LT B5+B4+LOOPIK
EQ SUMULT
END
```
Although the MMLE program does not require OVERLAY or SEGMENTATION to fit on most large computers, it is usually desirable to segment the program to decrease the load on the system. The following SEGMENTATION directives are used on the CDC OPERATING SYSTEM SCOPE 3.4 to reduce the loaded program size from 74,000\(^8\) words to 52,000\(^8\) words (including all buffers and system routines for input/output). The cost in execution time is negligible. The structure illustrated by these directives may be used as a guide for implementing the MMLE program on other systems.

```
PLTTREE TREE LINES-(THPLOT,APRPLT)
DATTREE TREE MATLD-(EDIT,DATA)
DOTREE TREE AEAT-(AGIRL,OUTPUT)
     TREE MMLE-(PLTTREE,ASPIT-(DATTREE,DOTREE))
LINES GLOBAL LINCOM
ASPIT GLOBAL TOGIRL,INFO,TODATA,ROUTH,DIMENS
     GLOBAL ALLDIM,BUF,MATRIX,COM,TOPLT,HEADNG
END
```
APPENDIX B

SAMPLE CHECK CASE FOR MMLE PROGRAM

This appendix presents a sample check case for the MMLE program. This listing is intended to aid the user in checking out the MMLE program; therefore, many of the available options have not been used.

INPUT CARDS

```
AIRCRAFT A CHECK CASE
INPUT CARDS
&INPUT CARO=TeQ=520,,V=4665., END
J a 8 0  9 07 5
. 605 10  500  .400
-9.906 0.19  .298  .7009  0.315  -5.024  .2036
+7.79  0.83  0.30  0.10  0.02
-1.59  0.45  .14  0.30  0.10
-1.95  -0.0284  -1.3642  3.0
+4  9
12.74  20.28
.3977  -2.445

01  7
2160  6.5  1365  115  12265  27  198
ENDCASE

J o 7 25  .729  2.001  .288  .7999  .315  -5.024  -2.036
-0.89  0.18  0.09  0.09  0.03
-8.26  0.08  .298  .7009  .015  -5.663  -3.395
+1.18  0.39  .04  0.10  0.03
-1.95  0.45  .14  0.30  0.10
-1.95  -0.0284  -1.3642  3.0
+4  9
12.74  20.28
.3977  -2.445
```
<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J 0 1 425</td>
<td>2.95E-01</td>
<td>-20.7974</td>
<td>-1.30E-01</td>
<td>2.50E-01</td>
<td>-2.48E-01</td>
<td>-53.0E-01</td>
<td>2.0E+01</td>
</tr>
<tr>
<td>J 0 1 45</td>
<td>2.43E-01</td>
<td>-21.8972</td>
<td>-1.23E-01</td>
<td>2.03E-01</td>
<td>-2.48E-01</td>
<td>-52.4E+01</td>
<td>2.0E+01</td>
</tr>
<tr>
<td>J 0 1 730</td>
<td>1.59E+00</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
<td>1.10E-02</td>
<td>1.59E+00</td>
<td>-2.40E+00</td>
<td>1.9E+01</td>
</tr>
<tr>
<td>J 0 1 475</td>
<td>2.42E-01</td>
<td>-23.9014</td>
<td>-1.15E-01</td>
<td>1.50E-01</td>
<td>-2.46E-01</td>
<td>-51.6E+01</td>
<td>1.9E+01</td>
</tr>
<tr>
<td>J 0 1 500</td>
<td>2.60E+00</td>
<td>-24.4013</td>
<td>-1.11E+00</td>
<td>7.99E+00</td>
<td>-2.44E+00</td>
<td>-49.9E+00</td>
<td>1.9E+01</td>
</tr>
<tr>
<td>J 0 1 500</td>
<td>2.00E+00</td>
<td>-26.9013</td>
<td>-1.10E+00</td>
<td>7.99E+00</td>
<td>-2.44E+00</td>
<td>-49.9E+00</td>
<td>1.9E+01</td>
</tr>
<tr>
<td>J 0 1 500</td>
<td>2.00E+00</td>
<td>-26.9013</td>
<td>-1.10E+00</td>
<td>7.99E+00</td>
<td>-2.44E+00</td>
<td>-49.9E+00</td>
<td>1.9E+01</td>
</tr>
<tr>
<td>J 0 1 500</td>
<td>2.00E+00</td>
<td>-26.9013</td>
<td>-1.10E+00</td>
<td>7.99E+00</td>
<td>-2.44E+00</td>
<td>-49.9E+00</td>
<td>1.9E+01</td>
</tr>
<tr>
<td>J 0 1 500</td>
<td>2.00E+00</td>
<td>-26.9013</td>
<td>-1.10E+00</td>
<td>7.99E+00</td>
<td>-2.44E+00</td>
<td>-49.9E+00</td>
<td>1.9E+01</td>
</tr>
<tr>
<td>J 0 1 500</td>
<td>2.00E+00</td>
<td>-26.9013</td>
<td>-1.10E+00</td>
<td>7.99E+00</td>
<td>-2.44E+00</td>
<td>-49.9E+00</td>
<td>1.9E+01</td>
</tr>
<tr>
<td>J 0 1 500</td>
<td>2.00E+00</td>
<td>-26.9013</td>
<td>-1.10E+00</td>
<td>7.99E+00</td>
<td>-2.44E+00</td>
<td>-49.9E+00</td>
<td>1.9E+01</td>
</tr>
<tr>
<td>J 0 1 500</td>
<td>2.00E+00</td>
<td>-26.9013</td>
<td>-1.10E+00</td>
<td>7.99E+00</td>
<td>-2.44E+00</td>
<td>-49.9E+00</td>
<td>1.9E+01</td>
</tr>
<tr>
<td>J 0 1 500</td>
<td>2.00E+00</td>
<td>-26.9013</td>
<td>-1.10E+00</td>
<td>7.99E+00</td>
<td>-2.44E+00</td>
<td>-49.9E+00</td>
<td>1.9E+01</td>
</tr>
<tr>
<td>J 0 1 500</td>
<td>2.00E+00</td>
<td>-26.9013</td>
<td>-1.10E+00</td>
<td>7.99E+00</td>
<td>-2.44E+00</td>
<td>-49.9E+00</td>
<td>1.9E+01</td>
</tr>
<tr>
<td>J 0 1 500</td>
<td>2.00E+00</td>
<td>-26.9013</td>
<td>-1.10E+00</td>
<td>7.99E+00</td>
<td>-2.44E+00</td>
<td>-49.9E+00</td>
<td>1.9E+01</td>
</tr>
<tr>
<td>J 0 1 500</td>
<td>2.00E+00</td>
<td>-26.9013</td>
<td>-1.10E+00</td>
<td>7.99E+00</td>
<td>-2.44E+00</td>
<td>-49.9E+00</td>
<td>1.9E+01</td>
</tr>
<tr>
<td>J 0.3 75</td>
<td>-6.30°</td>
<td>3.0023</td>
<td>-1.6694</td>
<td>-37.4534</td>
<td>.1730</td>
<td>37.4959</td>
<td>-1.9558</td>
</tr>
<tr>
<td>J 0.3 10</td>
<td>-6.00°</td>
<td>3.0066</td>
<td>-1.6798</td>
<td>-37.7134</td>
<td>.1700</td>
<td>36.9994</td>
<td>-1.8679</td>
</tr>
<tr>
<td>J 0.3 15</td>
<td>-6.00°</td>
<td>3.0096</td>
<td>-1.6798</td>
<td>-37.7134</td>
<td>.1700</td>
<td>36.9994</td>
<td>-1.8679</td>
</tr>
<tr>
<td>J 0.3 25</td>
<td>-5.60°</td>
<td>3.0107</td>
<td>-1.7478</td>
<td>-37.3533</td>
<td>.1650</td>
<td>35.9728</td>
<td>-1.7137</td>
</tr>
<tr>
<td>J 0.3 35</td>
<td>-6.00°</td>
<td>3.0107</td>
<td>-1.7478</td>
<td>-37.3533</td>
<td>.1650</td>
<td>35.9728</td>
<td>-1.7137</td>
</tr>
<tr>
<td>J 0.3 45</td>
<td>-6.00°</td>
<td>3.0107</td>
<td>-1.7478</td>
<td>-37.3533</td>
<td>.1650</td>
<td>35.9728</td>
<td>-1.7137</td>
</tr>
<tr>
<td>J 0.3 55</td>
<td>-6.00°</td>
<td>3.0107</td>
<td>-1.7478</td>
<td>-37.3533</td>
<td>.1650</td>
<td>35.9728</td>
<td>-1.7137</td>
</tr>
<tr>
<td>J 0.3 75</td>
<td>-6.00°</td>
<td>3.0107</td>
<td>-1.7478</td>
<td>-37.3533</td>
<td>.1650</td>
<td>35.9728</td>
<td>-1.7137</td>
</tr>
<tr>
<td>J 0.3 10</td>
<td>-6.00°</td>
<td>3.0107</td>
<td>-1.7478</td>
<td>-37.3533</td>
<td>.1650</td>
<td>35.9728</td>
<td>-1.7137</td>
</tr>
<tr>
<td>J 0.3 15</td>
<td>-6.00°</td>
<td>3.0107</td>
<td>-1.7478</td>
<td>-37.3533</td>
<td>.1650</td>
<td>35.9728</td>
<td>-1.7137</td>
</tr>
<tr>
<td>J 0.3 25</td>
<td>-5.60°</td>
<td>3.0107</td>
<td>-1.7478</td>
<td>-37.3533</td>
<td>.1650</td>
<td>35.9728</td>
<td>-1.7137</td>
</tr>
<tr>
<td>J 0.3 35</td>
<td>-6.00°</td>
<td>3.0107</td>
<td>-1.7478</td>
<td>-37.3533</td>
<td>.1650</td>
<td>35.9728</td>
<td>-1.7137</td>
</tr>
<tr>
<td>J 0.3 45</td>
<td>-6.00°</td>
<td>3.0107</td>
<td>-1.7478</td>
<td>-37.3533</td>
<td>.1650</td>
<td>35.9728</td>
<td>-1.7137</td>
</tr>
<tr>
<td>J 0.3 55</td>
<td>-6.00°</td>
<td>3.0107</td>
<td>-1.7478</td>
<td>-37.3533</td>
<td>.1650</td>
<td>35.9728</td>
<td>-1.7137</td>
</tr>
<tr>
<td>J 0.3 75</td>
<td>-6.00°</td>
<td>3.0107</td>
<td>-1.7478</td>
<td>-37.3533</td>
<td>.1650</td>
<td>35.9728</td>
<td>-1.7137</td>
</tr>
</tbody>
</table>

APPENDIX B – Continued
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>775</td>
<td>1,775</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>750</td>
<td>2,150</td>
<td>-3,798</td>
<td>-34.954</td>
<td>-1.167</td>
<td>-14.250</td>
<td>1.719</td>
</tr>
<tr>
<td>0</td>
<td>775</td>
<td>1,750</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>400</td>
<td>2,120</td>
<td>-4,297</td>
<td>-41.111</td>
<td>-1.150</td>
<td>-13.805</td>
<td>1.621</td>
</tr>
<tr>
<td>0</td>
<td>750</td>
<td>1,750</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>825</td>
<td>2,576</td>
<td>-4,463</td>
<td>-42.988</td>
<td>-1.156</td>
<td>-12.544</td>
<td>1.569</td>
</tr>
<tr>
<td>0</td>
<td>750</td>
<td>1,750</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>950</td>
<td>2,030</td>
<td>-5,002</td>
<td>-48.83</td>
<td>-1.146</td>
<td>-12.011</td>
<td>1.449</td>
</tr>
<tr>
<td>0</td>
<td>750</td>
<td>1,750</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>875</td>
<td>1,980</td>
<td>-5,502</td>
<td>-52.723</td>
<td>-1.141</td>
<td>-11.199</td>
<td>1.391</td>
</tr>
<tr>
<td>0</td>
<td>750</td>
<td>1,750</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>925</td>
<td>1,930</td>
<td>-5,869</td>
<td>-58.614</td>
<td>-1.134</td>
<td>-9.864</td>
<td>1.249</td>
</tr>
<tr>
<td>0</td>
<td>750</td>
<td>1,750</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>950</td>
<td>1,850</td>
<td>-6,247</td>
<td>-61.981</td>
<td>-1.124</td>
<td>-7.988</td>
<td>1.198</td>
</tr>
<tr>
<td>0</td>
<td>750</td>
<td>1,750</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>975</td>
<td>1,810</td>
<td>-6,396</td>
<td>-64.742</td>
<td>-1.124</td>
<td>-7.013</td>
<td>1.180</td>
</tr>
<tr>
<td>0</td>
<td>760</td>
<td>1,760</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>950</td>
<td>1,750</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>925</td>
<td>1,750</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>950</td>
<td>1,750</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>975</td>
<td>1,750</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>100</td>
<td>1,720</td>
<td>-6,303</td>
<td>-63.622</td>
<td>-1.170</td>
<td>-4.518</td>
<td>1.770</td>
</tr>
<tr>
<td>0</td>
<td>760</td>
<td>1,760</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>790</td>
<td>1,775</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>790</td>
<td>1,775</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>810</td>
<td>1,780</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>830</td>
<td>1,790</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>850</td>
<td>1,800</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>870</td>
<td>1,810</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>890</td>
<td>1,820</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>910</td>
<td>1,830</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>930</td>
<td>1,840</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>950</td>
<td>1,850</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>970</td>
<td>1,860</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>990</td>
<td>1,870</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>101</td>
<td>1,880</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>103</td>
<td>1,890</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>105</td>
<td>1,900</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>107</td>
<td>1,910</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>109</td>
<td>1,920</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>111</td>
<td>1,930</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>113</td>
<td>1,940</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>115</td>
<td>1,950</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>117</td>
<td>1,960</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>119</td>
<td>1,970</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>121</td>
<td>1,980</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>123</td>
<td>1,990</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>125</td>
<td>2,000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

APPENDIX B – Continued
<table>
<thead>
<tr>
<th>Command</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 5 55</td>
<td>2.700</td>
</tr>
<tr>
<td>0 0 5 60</td>
<td>2.068</td>
</tr>
<tr>
<td>0 0 5 65</td>
<td>1.170</td>
</tr>
</tbody>
</table>

**APPENDIX B — Continued**

<table>
<thead>
<tr>
<th>Command</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 24</td>
<td>2.700</td>
</tr>
<tr>
<td>3 25</td>
<td>2.068</td>
</tr>
<tr>
<td>3 30</td>
<td>1.170</td>
</tr>
</tbody>
</table>

**AIRCRAFT B CHECK CASE**

<table>
<thead>
<tr>
<th>Command</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPUT GROWTH= 2.737</td>
<td></td>
</tr>
<tr>
<td>O= 39.0</td>
<td></td>
</tr>
<tr>
<td>X= .152</td>
<td></td>
</tr>
<tr>
<td>PUNCH=T</td>
<td></td>
</tr>
<tr>
<td>TIMESC=.5</td>
<td></td>
</tr>
<tr>
<td>GROSWT=</td>
<td></td>
</tr>
<tr>
<td>GI= 0.0</td>
<td></td>
</tr>
<tr>
<td>XAN= -0.1</td>
<td></td>
</tr>
<tr>
<td>ZAX= .5</td>
<td></td>
</tr>
<tr>
<td>ZMAX=1000</td>
<td></td>
</tr>
<tr>
<td>CARD=</td>
<td></td>
</tr>
</tbody>
</table>

**CAIT**

<table>
<thead>
<tr>
<th>Command</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NMAPR= .10E+01</td>
<td></td>
</tr>
<tr>
<td>ALPHA= 7.86</td>
<td></td>
</tr>
<tr>
<td>MACH=.429</td>
<td></td>
</tr>
<tr>
<td>CG=.265</td>
<td></td>
</tr>
<tr>
<td>PARM= 5.0000</td>
<td></td>
</tr>
</tbody>
</table>

**LONGI. 5 | 85.5 | SPAN= 16.5 | |
| GMA= .585 | |
| SPS= 51 | |
| END | |

<table>
<thead>
<tr>
<th>Command</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>11365 750</td>
<td></td>
</tr>
<tr>
<td>11365 780</td>
<td></td>
</tr>
<tr>
<td>2.196</td>
<td></td>
</tr>
</tbody>
</table>

**END**

<table>
<thead>
<tr>
<th>Command</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>11365 750</td>
<td></td>
</tr>
<tr>
<td>11365 780</td>
<td></td>
</tr>
<tr>
<td>2.196</td>
<td></td>
</tr>
</tbody>
</table>

**END**
<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3,507</td>
<td>4.2457</td>
</tr>
<tr>
<td>3.014</td>
<td>0.298</td>
</tr>
<tr>
<td>2.519</td>
<td>0.234</td>
</tr>
<tr>
<td>2.014</td>
<td>0.168</td>
</tr>
<tr>
<td>1.514</td>
<td>0.105</td>
</tr>
<tr>
<td>1.014</td>
<td>0.043</td>
</tr>
<tr>
<td>0.519</td>
<td>0.003</td>
</tr>
</tbody>
</table>

**APPENDIX B – Continued**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.507</td>
<td>4.2457</td>
</tr>
<tr>
<td>3.014</td>
<td>0.298</td>
</tr>
<tr>
<td>2.519</td>
<td>0.234</td>
</tr>
<tr>
<td>2.014</td>
<td>0.168</td>
</tr>
<tr>
<td>1.514</td>
<td>0.105</td>
</tr>
<tr>
<td>1.014</td>
<td>0.043</td>
</tr>
<tr>
<td>0.519</td>
<td>0.003</td>
</tr>
</tbody>
</table>

101
<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>y'</th>
<th>y''</th>
<th>y'''</th>
<th>y''''</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.100</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>0.200</td>
<td>-0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>0.300</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>0.400</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>0.500</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>0.600</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>0.700</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>0.800</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>0.900</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

APPENDIX B — Continued
<table>
<thead>
<tr>
<th>Value</th>
<th>Value</th>
<th>Value</th>
<th>Value</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1364</td>
<td>835</td>
<td>0.1364</td>
<td>0.835</td>
<td>0.835</td>
</tr>
<tr>
<td>8.934</td>
<td>3.934</td>
<td>3.934</td>
<td>3.934</td>
<td>3.934</td>
</tr>
</tbody>
</table>

**APPENDIX B — Continued**
<table>
<thead>
<tr>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
<th>Value 4</th>
<th>Value 5</th>
<th>Value 6</th>
<th>Value 7</th>
<th>Value 8</th>
<th>Value 9</th>
<th>Value 10</th>
<th>Value 11</th>
<th>Value 12</th>
<th>Value 13</th>
<th>Value 14</th>
<th>Value 15</th>
<th>Value 16</th>
</tr>
</thead>
</table>

**APPENDIX B** – Continued
| 113641 465 | 7.3764 | -1.9779 | 412.8156 | -1.1875 | .6464 | 0.2003 | -6.1252 |
| 113641 485 | -1.4129 | -0.5114 | 0.2626 | 4.3314 | 6.954746555.3796 | 0.426 | 38.4442 |
| 113641 505 | -1.8586 | -0.5577 | 0.2626 | 4.3350 | 7.039745955.3796 | 0.426 | 38.4442 |
| 113641 525 | -2.3828 | -0.6062 | 0.2714 | 4.3399 | 7.166455555.3796 | 0.426 | 38.4442 |
| 113641 545 | -2.9356 | -0.6539 | 0.2802 | 4.3437 | 7.283454949.6626 | 0.426 | 38.4445 |
| 113641 565 | -3.4814 | -0.7031 | 0.3134 | 4.3473 | 7.399454449.6626 | 0.426 | 38.4445 |
| 113641 585 | -4.0261 | -0.7569 | 0.3410 | 4.3499 | 7.476454449.6626 | 0.426 | 38.4445 |
| 113641 605 | -4.5779 | -0.8052 | 0.3653 | 4.3525 | 7.544454449.6626 | 0.426 | 38.4445 |
| 113641 625 | -5.1227 | -0.8534 | 0.3835 | 4.3552 | 7.599454449.6626 | 0.426 | 38.4445 |
| 113641 645 | -5.6691 | -0.9014 | 0.4010 | 4.3578 | 7.652454449.6626 | 0.426 | 38.4445 |
| 113641 665 | -6.2121 | -0.9481 | 0.4177 | 4.3596 | 7.699454449.6626 | 0.426 | 38.4445 |
| 113641 685 | -6.7535 | -0.9936 | 0.4334 | 4.3614 | 7.742454449.6626 | 0.426 | 38.4445 |
| 113641 705 | -7.2946 | -1.0367 | 0.4479 | 4.3624 | 7.781454449.6626 | 0.426 | 38.4445 |
| 113641 725 | -7.8347 | -1.0762 | 0.4603 | 4.3634 | 7.816454449.6626 | 0.426 | 38.4445 |
| 113641 745 | -8.3738 | -1.1121 | 0.4708 | 4.3634 | 7.847454449.6626 | 0.426 | 38.4445 |
| 113641 765 | -8.9129 | -1.1457 | 0.4798 | 4.3634 | 7.874454449.6626 | 0.426 | 38.4445 |
| 113641 785 | -9.4510 | -1.1762 | 0.4878 | 4.3634 | 7.897454449.6626 | 0.426 | 38.4445 |
| 113641 805 | -10.0300 | -1.2052 | 0.5048 | 4.3634 | 7.917454449.6626 | 0.426 | 38.4445 |
| 113641 825 | -10.5689 | -1.2315 | 0.5201 | 4.3634 | 7.933454449.6626 | 0.426 | 38.4445 |
| 113641 845 | -11.0956 | -1.2552 | 0.5337 | 4.3634 | 7.944454449.6626 | 0.426 | 38.4445 |
| 113641 865 | -11.6124 | -1.2773 | 0.5457 | 4.3634 | 7.951454449.6626 | 0.426 | 38.4445 |
| 113641 885 | -12.1182 | -1.2970 | 0.5557 | 4.3634 | 7.954454449.6626 | 0.426 | 38.4445 |
| 113641 905 | -12.6027 | -1.3143 | 0.5640 | 4.3634 | 7.955454449.6626 | 0.426 | 38.4445 |
| 113641 925 | -13.0765 | -1.3297 | 0.5707 | 4.3634 | 7.955454449.6626 | 0.426 | 38.4445 |
| 113641 945 | -13.5392 | -1.3433 | 0.5762 | 4.3634 | 7.955454449.6626 | 0.426 | 38.4445 |
| 113641 965 | -13.9908 | -1.3547 | 0.5801 | 4.3634 | 7.955454449.6626 | 0.426 | 38.4445 |
| 113641 985 | -14.4314 | -1.3632 | 0.5825 | 4.3634 | 7.955454449.6626 | 0.426 | 38.4445 |
| 113641 1005 | -14.8596 | -1.3697 | 0.5840 | 4.3634 | 7.955454449.6626 | 0.426 | 38.4445 |

APPENDIX B — Continued
| Page 1347 | 125 | 8.4327 | -1.0222 | 414.1156 | -1.1871 | .7543 | 0.0000 | .0.005 | .0.62 |
| 1347 | 225 | 8.4327 | -1.0222 | 414.1156 | -1.1871 | .7543 | 0.0000 | .0.005 | .0.62 |
| 1347 | 425 | 8.4327 | -1.0222 | 414.1156 | -1.1871 | .7543 | 0.0000 | .0.005 | .0.62 |
| 1347 | 625 | 8.4327 | -1.0222 | 414.1156 | -1.1871 | .7543 | 0.0000 | .0.005 | .0.62 |

APPENDIX B – Continued
| 113642 785 | 8.9917 | -2.2720 | 410.5515 | -1.4213 | .7939 | 0.0003 | 0.6942 |
| 113642 805 | 8.9748 | -2.5647 | 410.5627 | -1.4256 | 0.7939 | 0.0000 | -0.6102 |
| 113642 825 | 8.9330 | -2.8323 | 410.5655 | -1.4519 | 0.8027 | 0.0000 | -0.6116 |
| 113642 845 | 8.8271 | -3.0946 | 410.7267 | -1.5287 | 0.8219 | 0.0000 | -0.6116 |
| 113642 865 | 8.634 | -3.3047 | 410.7374 | -1.6171 | 0.8772 | 0.0000 | -0.6116 |
| 113642 885 | 8.399 | -3.5391 | 410.7522 | -1.7255 | 0.9467 | 0.0000 | -0.6116 |
| 113642 905 | 8.190 | -3.790 | 410.7638 | -1.8665 | 0.9768 | 0.0000 | -0.6116 |
| 113642 925 | 8.007 | -4.050 | 410.7740 | -1.9947 | 0.9768 | 0.0000 | -0.6116 |
| 113642 945 | 7.839 | -4.360 | 410.7832 | -2.1206 | 0.9768 | 0.0000 | -0.6116 |
| 113642 965 | 7.600 | -4.680 | 410.7919 | -2.2459 | 0.9768 | 0.0000 | -0.6116 |
| 113642 985 | 7.378 | -5.060 | 410.7991 | -2.3806 | 0.9768 | 0.0000 | -0.6116 |
| 113643 005 | 7.160 | -5.460 | 410.8054 | -2.5267 | 0.9768 | 0.0000 | -0.6116 |
| 113643 025 | 6.932 | -5.880 | 410.8102 | -2.6739 | 0.9768 | 0.0000 | -0.6116 |
| 113643 045 | 6.664 | -6.320 | 410.8139 | -2.8221 | 0.9768 | 0.0000 | -0.6116 |
| 113643 065 | 6.356 | -6.800 | 410.8167 | -2.9713 | 0.9768 | 0.0000 | -0.6116 |
| 113643 085 | 6.007 | -7.320 | 410.8199 | -3.1267 | 0.9768 | 0.0000 | -0.6116 |
| 113643 105 | 5.608 | -7.880 | 410.8227 | -3.2852 | 0.9768 | 0.0000 | -0.6116 |
| 113643 125 | 5.170 | -8.440 | 410.8254 | -3.4513 | 0.9768 | 0.0000 | -0.6116 |
| 113643 145 | 4.711 | -9.000 | 410.8281 | -3.6256 | 0.9768 | 0.0000 | -0.6116 |
| 113643 165 | 4.228 | -9.600 | 410.8308 | -3.8079 | 0.9768 | 0.0000 | -0.6116 |
| 113643 185 | 3.715 | -10.240 | 410.8335 | -4.0000 | 0.9768 | 0.0000 | -0.6116 |
| 113643 205 | 3.184 | -10.880 | 410.8362 | -4.2000 | 0.9768 | 0.0000 | -0.6116 |

APPENDIX B – Continued
APPENDIX B — Continued
<table>
<thead>
<tr>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th>
<th>Column 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>11364</td>
<td>105</td>
<td>7.7835</td>
<td>-2.1930</td>
</tr>
<tr>
<td>11364</td>
<td>105</td>
<td>7.7835</td>
<td>-2.1930</td>
</tr>
<tr>
<td>11364</td>
<td>105</td>
<td>7.7835</td>
<td>-2.1930</td>
</tr>
<tr>
<td>11364</td>
<td>105</td>
<td>7.7835</td>
<td>-2.1930</td>
</tr>
<tr>
<td>11364</td>
<td>105</td>
<td>7.7835</td>
<td>-2.1930</td>
</tr>
<tr>
<td>11364</td>
<td>105</td>
<td>7.7835</td>
<td>-2.1930</td>
</tr>
<tr>
<td>11364</td>
<td>105</td>
<td>7.7835</td>
<td>-2.1930</td>
</tr>
<tr>
<td>11364</td>
<td>105</td>
<td>7.7835</td>
<td>-2.1930</td>
</tr>
<tr>
<td>11364</td>
<td>105</td>
<td>7.7835</td>
<td>-2.1930</td>
</tr>
<tr>
<td>11364</td>
<td>105</td>
<td>7.7835</td>
<td>-2.1930</td>
</tr>
<tr>
<td>11364</td>
<td>105</td>
<td>7.7835</td>
<td>-2.1930</td>
</tr>
<tr>
<td>11364</td>
<td>105</td>
<td>7.7835</td>
<td>-2.1930</td>
</tr>
<tr>
<td>11364</td>
<td>105</td>
<td>7.7835</td>
<td>-2.1930</td>
</tr>
<tr>
<td>11364</td>
<td>105</td>
<td>7.7835</td>
<td>-2.1930</td>
</tr>
<tr>
<td>11364</td>
<td>105</td>
<td>7.7835</td>
<td>-2.1930</td>
</tr>
<tr>
<td>11364</td>
<td>105</td>
<td>7.7835</td>
<td>-2.1930</td>
</tr>
<tr>
<td>11364</td>
<td>105</td>
<td>7.7835</td>
<td>-2.1930</td>
</tr>
<tr>
<td>11364</td>
<td>105</td>
<td>7.7835</td>
<td>-2.1930</td>
</tr>
<tr>
<td>11364</td>
<td>105</td>
<td>7.7835</td>
<td>-2.1930</td>
</tr>
<tr>
<td>11364</td>
<td>105</td>
<td>7.7835</td>
<td>-2.1930</td>
</tr>
<tr>
<td>11364</td>
<td>105</td>
<td>7.7835</td>
<td>-2.1930</td>
</tr>
</tbody>
</table>

APPENDIX B – Continued
<table>
<thead>
<tr>
<th>Temperature (°C)</th>
<th>Interval 1</th>
<th>Interval 2</th>
<th>Interval 3</th>
<th>Interval 4</th>
<th>Interval 5</th>
<th>Interval 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>113645 426</td>
<td>7.7170</td>
<td>-2.0894</td>
<td>420.4174</td>
<td>-5.9321</td>
<td>-0.002</td>
<td>-0.003</td>
</tr>
<tr>
<td>113645 446</td>
<td>7.6016</td>
<td>-2.5094</td>
<td>420.4575</td>
<td>-6.0286</td>
<td>-0.004</td>
<td>-0.005</td>
</tr>
<tr>
<td>113645 465</td>
<td>7.6638</td>
<td>-2.5717</td>
<td>420.5238</td>
<td>-6.0965</td>
<td>-0.005</td>
<td>-0.006</td>
</tr>
<tr>
<td>113645 486</td>
<td>7.5837</td>
<td>-2.7632</td>
<td>420.6526</td>
<td>-6.2809</td>
<td>-0.005</td>
<td>-0.006</td>
</tr>
<tr>
<td>113645 506</td>
<td>7.5255</td>
<td>-2.8894</td>
<td>420.7231</td>
<td>-6.4205</td>
<td>-0.005</td>
<td>-0.006</td>
</tr>
<tr>
<td>113645 526</td>
<td>7.4511</td>
<td>-3.508</td>
<td>420.8254</td>
<td>-6.6173</td>
<td>-0.006</td>
<td>-0.007</td>
</tr>
<tr>
<td>113645 546</td>
<td>7.3938</td>
<td>-3.5506</td>
<td>420.9321</td>
<td>-6.7681</td>
<td>-0.007</td>
<td>-0.008</td>
</tr>
<tr>
<td>113645 566</td>
<td>7.3638</td>
<td>-3.6555</td>
<td>421.0231</td>
<td>-6.9205</td>
<td>-0.008</td>
<td>-0.009</td>
</tr>
<tr>
<td>113645 586</td>
<td>7.3311</td>
<td>-3.7323</td>
<td>421.1265</td>
<td>-7.0732</td>
<td>-0.009</td>
<td>-0.010</td>
</tr>
<tr>
<td>113645 606</td>
<td>7.2955</td>
<td>-3.8107</td>
<td>421.2265</td>
<td>-7.2269</td>
<td>-0.010</td>
<td>-0.011</td>
</tr>
<tr>
<td>113645 626</td>
<td>7.2613</td>
<td>-3.8773</td>
<td>421.3265</td>
<td>-7.3806</td>
<td>-0.011</td>
<td>-0.012</td>
</tr>
<tr>
<td>113645 646</td>
<td>7.2251</td>
<td>-3.9331</td>
<td>421.4265</td>
<td>-7.5342</td>
<td>-0.012</td>
<td>-0.013</td>
</tr>
<tr>
<td>113645 666</td>
<td>7.1865</td>
<td>-3.9790</td>
<td>421.5265</td>
<td>-7.6878</td>
<td>-0.013</td>
<td>-0.014</td>
</tr>
<tr>
<td>113645 686</td>
<td>7.1463</td>
<td>-4.0150</td>
<td>421.6265</td>
<td>-7.8414</td>
<td>-0.014</td>
<td>-0.015</td>
</tr>
<tr>
<td>113645 706</td>
<td>7.1051</td>
<td>-4.0410</td>
<td>421.7265</td>
<td>-7.9950</td>
<td>-0.015</td>
<td>-0.016</td>
</tr>
<tr>
<td>113645 726</td>
<td>7.0629</td>
<td>-4.0610</td>
<td>421.8265</td>
<td>-8.1486</td>
<td>-0.016</td>
<td>-0.017</td>
</tr>
<tr>
<td>113645 746</td>
<td>7.0197</td>
<td>-4.0770</td>
<td>421.9265</td>
<td>-8.2022</td>
<td>-0.017</td>
<td>-0.018</td>
</tr>
<tr>
<td>113645 766</td>
<td>6.9755</td>
<td>-4.0910</td>
<td>421.9933</td>
<td>-8.2558</td>
<td>-0.018</td>
<td>-0.019</td>
</tr>
<tr>
<td>113645 786</td>
<td>6.9303</td>
<td>-4.1040</td>
<td>421.9933</td>
<td>-8.3094</td>
<td>-0.019</td>
<td>-0.020</td>
</tr>
<tr>
<td>113645 806</td>
<td>6.8841</td>
<td>-4.1150</td>
<td>421.9933</td>
<td>-8.3630</td>
<td>-0.020</td>
<td>-0.021</td>
</tr>
<tr>
<td>113645 826</td>
<td>6.8379</td>
<td>-4.1250</td>
<td>421.9933</td>
<td>-8.4166</td>
<td>-0.021</td>
<td>-0.022</td>
</tr>
<tr>
<td>113645 846</td>
<td>6.7907</td>
<td>-4.1340</td>
<td>421.9933</td>
<td>-8.4692</td>
<td>-0.022</td>
<td>-0.023</td>
</tr>
</tbody>
</table>
INPUT DATA (T INDICATES TRUE OR YES, F INDICATES FALSE OR NO)

LATERAL CASE
DATA SOURCE CARD? T DATA TAP? F
DATA RATE IS 0 SAMPLES/SECOND ON SOURCE FILE IF Q, DETERMINED FROM TIMES ON THE SOURCE FILE DIVIDED BY THINNING FACTOR OF 1
ON INPUT TAPE IS DATA WORDS PER RECORD, SPECIAL SIGNAL ORDER DEFAULT? F

PROGRAM OPTIONS
APRIORI WEIGHTING = 0, 0 TIME HALVINGS IN EAT.
ITERATIONS = 6 (ITERATION WILL STOP IF ERROR SUM CHANGES BY LESS THAN A FACTOR OF .10E-02)
CASE WILL BE STOPPED IF ERROR SUM IS GREATER THAN .10E+21

OUTPUT
PLOTS? T (NO PLOTS UNLESS FINAL ERROR SUM IS LESS THAN .10E+06)
NUMBER OF CONTROLS AND EXTRA SIGNALS TO BE PLOTTED = 8
SECONDS PER CENTIMETER = 1.0
PRINTED FLIGHT AND FINAL COMPUTED TIME HISTORIES? F
EXTRA OUTPUT OF INTERMEDIATE STEPS FOR A DIAGNOSTIC AID? F
PUNCHED FINAL NON-DIMENSIONAL DERIVATIVES AND CONFIDENCE LEVELS? F
PUNCHED FINAL DIMENSIONAL MATRIX? F

FLIGHT CONDITION AND VEHICLE CHARACTERISTICS (Q, INDICATES VALUE OBTAINED FROM TIME HISTORY ON QBAR, V OR MACH)
(MACH, ALPHA, CG AND PARAM ARE FOR REFERENCE ONLY, NOT USED IN PROGRAM)

METRIC UNITS? F
DYNAMIC PRESSURE = 520.0 VELICITY = 4685.0
MACH = 0.000 ALPHA = 999.000 (IF 999.000, OBTAINED FROM TIME HISTORY)
CENTER OF GRAVITY = 0.250 OTHER IDENTIFYING PARAMETER = 0.0
WING AREA = 0.0 SPAN = 0.0 CHORD = 0.0
IX = 0.0 IY = 0.0 IZ = 0.0
WEIGHT ********

INSTRUMENT OFFSETS FROM CG
X-DIRECTION OFFSETS (+ = INSTRUMENT IS FORWARD OF CG)
ALPHA 0.000 AN 0.000
MACH 0.000 AY 0.000
Z-DIRECTION OFFSETS (+ = INSTRUMENT IS BELOW CG)
METAL 0.000 AN 0.000

SIGNAL SCALING AND BIASES

SIGNALS
VAR RIAS BETA P R PMT AQ POST ROOT DA OR DCL DC2 ALFA V MACH QBAR
VAR 1.0
F F F
FIXED BIAS 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
SCALE FACT 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00
PLOT LIMITS
MINIMUM 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
MAXIMUM 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

MANEUVER 1 START TIME 0 0 0 0 STOP TIME 0 0 5 075
### INPUT MATRICES

<table>
<thead>
<tr>
<th></th>
<th>4 by 4</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>5 by 7</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-3000E-31</td>
<td>1.110E+00</td>
<td>-1.000E+01</td>
<td>6.900E-02</td>
<td></td>
<td>-1.679E+02</td>
<td>2.410E+00</td>
<td>-4.000E+00</td>
<td>0.</td>
<td>-1.550E+01</td>
<td>2.440E-02</td>
<td>-4.200E-01</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td></td>
<td>0.</td>
<td></td>
<td>-1.000E+01</td>
<td>0.</td>
<td>-0.</td>
<td>0.</td>
<td></td>
<td>-1.000E+01</td>
<td>0.</td>
<td></td>
<td>0.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>0.</td>
<td>1.400E+01</td>
<td>0.</td>
<td>-0.</td>
<td>-0.</td>
<td>-0.</td>
<td>1.270E+02</td>
<td>2.000E+02</td>
<td>-0.</td>
<td>-0.</td>
<td>-0.</td>
<td>3.570E+00</td>
<td>-2.440E+01</td>
<td>-0.</td>
</tr>
<tr>
<td></td>
<td>0.</td>
<td>-0.</td>
<td>0.</td>
<td>-0.</td>
<td>-0.</td>
<td>-0.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1</td>
<td>0.2160E+04</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.</td>
<td>6.500E+01</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>1.300E+00</td>
</tr>
<tr>
<td></td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>2.260E+02</td>
</tr>
<tr>
<td></td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>2.700E+01</td>
</tr>
<tr>
<td></td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>1.980E+03</td>
</tr>
</tbody>
</table>

**TOTAL NUMBER OF POINTS FOR MANEUVER 1 = 239**
### APPENDIX B -- Continued

**STARTING VALUES**

<table>
<thead>
<tr>
<th>MACH = 3.000</th>
<th>ALPHA = 0.000</th>
<th>PARAM = 0.6600</th>
<th>CG = 0.250</th>
</tr>
</thead>
</table>

**DIMENSIONAL DERIVATIVES / SEC / SEC**

<table>
<thead>
<tr>
<th>BETA</th>
<th>P</th>
<th>R</th>
<th>DA</th>
<th>DR</th>
<th>DC1</th>
<th>DC2</th>
<th>DELTA=0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y = 0.00000</td>
<td>1.11100</td>
<td>-1.09E+00</td>
<td>-0.00E+00</td>
<td>0.01E+00</td>
<td>-0.00E+00</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
</tr>
<tr>
<td>L = 1.56E+00</td>
<td>1.22E+00</td>
<td>0.00E+00</td>
<td>1.26E+00</td>
<td>26.00E+00</td>
<td>-0.00E+00</td>
<td>-0.00E+00</td>
<td>0.00E+00</td>
</tr>
<tr>
<td>N = 1.56E+00</td>
<td>-0.00E+00</td>
<td>-0.00E+00</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
<td>-0.00E+00</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
</tr>
</tbody>
</table>

**NON-DIMENSIONAL DERIVATIVES / DEG (ROTARY / RAD)**

<table>
<thead>
<tr>
<th>BETA</th>
<th>P</th>
<th>R</th>
<th>DA</th>
<th>DR</th>
<th>DC1</th>
<th>DC2</th>
<th>DELTA=0</th>
</tr>
</thead>
<tbody>
<tr>
<td>CV</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td></td>
</tr>
<tr>
<td>CL</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CN</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.00000</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(*) INDICATES DERIVATIVE HELD FIXED DURING MATCH

**NUMBER OF unknowns = 21**

**ENTERING ITERATION LOOP**

**DIMENSIONAL DERIVATIVE MATRICES PER RADIUS, RIGS IN RADIANS.**

**A**

<table>
<thead>
<tr>
<th>4 BY 4</th>
<th>4 BY 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.30E+01</td>
<td>1.11E+00</td>
</tr>
<tr>
<td>-0.16E+00</td>
<td>-0.21E+00</td>
</tr>
<tr>
<td>-0.15E+00</td>
<td>0.28E+00</td>
</tr>
<tr>
<td>0.00E+00</td>
<td>0.00E+00</td>
</tr>
</tbody>
</table>

**B**

<table>
<thead>
<tr>
<th>4 BY 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.14E+00</td>
</tr>
<tr>
<td>-0.12E+00</td>
</tr>
<tr>
<td>0.00E+00</td>
</tr>
</tbody>
</table>

**VARIALE RIAS**

<table>
<thead>
<tr>
<th>0.00E+00</th>
<th>0.00E+00</th>
</tr>
</thead>
</table>

**ERRORS**

<table>
<thead>
<tr>
<th>1.15E+03</th>
<th>1.95E+01</th>
<th>1.56E+03</th>
<th>4.19E+01</th>
<th>9.51E+01</th>
<th>1.45E+01</th>
<th>1.78E+03</th>
</tr>
</thead>
</table>

**WEIGHTED ERRORS**

<table>
<thead>
<tr>
<th>2.27E+00</th>
<th>1.23E+00</th>
<th>7.59E+00</th>
<th>5.60E+00</th>
<th>2.17E+00</th>
<th>3.94E+00</th>
<th>2.15E+01</th>
</tr>
</thead>
</table>

**WEIGHTED ERROR SUM = 6.85E+01**

**ITERATION NUMBER 1 COMPLETED**

**A**

<table>
<thead>
<tr>
<th>4 BY 4</th>
<th>4 BY 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.46E+01</td>
<td>0.12E+00</td>
</tr>
<tr>
<td>-0.32E+00</td>
<td>-0.14E+00</td>
</tr>
<tr>
<td>-0.12E+00</td>
<td>-0.32E+00</td>
</tr>
<tr>
<td>0.00E+00</td>
<td>0.00E+00</td>
</tr>
</tbody>
</table>

**B**

<table>
<thead>
<tr>
<th>4 BY 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.23E+00</td>
</tr>
<tr>
<td>-0.22E+00</td>
</tr>
<tr>
<td>0.00E+00</td>
</tr>
</tbody>
</table>

**VARIALE RIAS**

<table>
<thead>
<tr>
<th>0.52E+00</th>
<th>-0.24E+00</th>
<th>-0.47E+00</th>
</tr>
</thead>
</table>

**ERRORS**

<table>
<thead>
<tr>
<th>4.87E+05</th>
<th>1.32E+02</th>
<th>7.74E+05</th>
<th>1.86E+02</th>
<th>3.27E+03</th>
<th>3.59E+02</th>
<th>3.06E+04</th>
</tr>
</thead>
</table>

**WEIGHTED ERRORS**

<table>
<thead>
<tr>
<th>8.46E+02</th>
<th>3.56E+01</th>
<th>2.54E+00</th>
<th>0.00E+00</th>
<th>0.00E+00</th>
<th>0.00E+00</th>
</tr>
</thead>
</table>

**WEIGHTED ERROR SUM = 3.32E+00**

**ITERATION NUMBER 2 COMPLETED**
<table>
<thead>
<tr>
<th>Iteration</th>
<th>Termination Error</th>
<th>Confidence Levels (Dimensional)</th>
<th>Confidence Levels (Non-Dimensional)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.48E-03</td>
<td>84598E-03</td>
<td>21725E-03</td>
</tr>
<tr>
<td>2</td>
<td>1.88E-03</td>
<td>3.759E-02</td>
<td>9.93E-02</td>
</tr>
<tr>
<td>3</td>
<td>3.62E-03</td>
<td>1.301E-02</td>
<td>3.94E-02</td>
</tr>
<tr>
<td>4</td>
<td>7.64E-03</td>
<td>6.720E-01</td>
<td>2.24E-01</td>
</tr>
<tr>
<td>5</td>
<td>1.09E-02</td>
<td>3.720E-00</td>
<td>2.24E-01</td>
</tr>
</tbody>
</table>

Note: Iteration levels for next to last iteration bound.
### Aircraft Check Case

<table>
<thead>
<tr>
<th>Final Values</th>
<th>Mach = 0.000</th>
<th>Alpha = 0.00</th>
<th>Param = 0.0000</th>
<th>CG = 0.250</th>
</tr>
</thead>
</table>

#### Dimensional Derivatives / Sec / Sec**2 |

<table>
<thead>
<tr>
<th>Beta</th>
<th>P</th>
<th>R</th>
<th>OA</th>
<th>DR</th>
<th>DC1</th>
<th>DC2</th>
<th>Delta-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>-0.04673</td>
<td>0.12050</td>
<td>-1.000000*</td>
<td>0.02763</td>
<td>0.15943</td>
<td>-0.000000*</td>
<td>-0.000000*</td>
</tr>
<tr>
<td>L</td>
<td>-24.77029</td>
<td>-0.19515</td>
<td>2.463745</td>
<td>14.46976</td>
<td>17.08474</td>
<td>-0.000000*</td>
<td>-0.000000*</td>
</tr>
<tr>
<td>N</td>
<td>1.239892</td>
<td>0.05458</td>
<td>-1.91540</td>
<td>0.58618</td>
<td>-2.19490</td>
<td>-0.000000*</td>
<td>-0.000000*</td>
</tr>
</tbody>
</table>

#### Non-Dimensional Derivatives / Deg (Rotary /rad) |

<table>
<thead>
<tr>
<th>Beta</th>
<th>P</th>
<th>R</th>
<th>OA</th>
<th>DR</th>
<th>DC1</th>
<th>DC2</th>
<th>Delta-0</th>
</tr>
</thead>
<tbody>
<tr>
<td>CY**</td>
<td>3.000000</td>
<td>0.000000*</td>
<td>0.000000*</td>
<td>0.000000*</td>
<td>0.000000*</td>
<td>0.000000*</td>
<td>0.000000*</td>
</tr>
<tr>
<td>CL**</td>
<td>0.000000*</td>
<td>0.000000*</td>
<td>0.000000*</td>
<td>0.000000*</td>
<td>0.000000*</td>
<td>0.000000*</td>
<td>0.000000*</td>
</tr>
<tr>
<td>CN**</td>
<td>0.000000*</td>
<td>0.000000*</td>
<td>0.000000*</td>
<td>0.000000*</td>
<td>0.000000*</td>
<td>0.000000*</td>
<td>0.000000*</td>
</tr>
</tbody>
</table>

(* ) indicates derivative held fixed during match.

### Final Dimensional Matrices

#### A 4 BY 4

<table>
<thead>
<tr>
<th></th>
<th>-0.46703E-01</th>
<th>-1.026E+00</th>
<th>-1.000E+01</th>
<th>-6.406E-02</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.2432E+02</td>
<td>-1.019E+00</td>
<td>-2.406E+01</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>-1.200E+01</td>
<td>-4.093E-03</td>
<td>-1.516E+00</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>0.0</td>
<td>1.000E+01</td>
<td>0.0</td>
<td>-0.0</td>
<td></td>
</tr>
</tbody>
</table>

#### B 4 BY 5

<table>
<thead>
<tr>
<th></th>
<th>0.275E-02</th>
<th>1.594E-01</th>
<th>-0.0</th>
<th>-0.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1447E+02</td>
<td>1.797E+02</td>
<td>-0.0</td>
<td>-0.0</td>
<td></td>
</tr>
<tr>
<td>0.566E+02</td>
<td>3.125E+02</td>
<td>-0.0</td>
<td>-0.0</td>
<td></td>
</tr>
<tr>
<td>-0.0</td>
<td>-0.0</td>
<td>-0.0</td>
<td>-0.0</td>
<td></td>
</tr>
</tbody>
</table>

#### Degrees AY Root Root

<table>
<thead>
<tr>
<th>Variable Bias</th>
<th>0.517E+00</th>
<th>-1.179E+01</th>
<th>-2.562E+00</th>
</tr>
</thead>
</table>

#### Errors

<table>
<thead>
<tr>
<th>Error</th>
<th>Weighted Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>.449E-05</td>
<td>.813E-03</td>
</tr>
<tr>
<td>.203E-05</td>
<td>.627E-04</td>
</tr>
<tr>
<td>.354E-03</td>
<td>.432E-02</td>
</tr>
<tr>
<td>.283E-04</td>
<td>.561E-02</td>
</tr>
</tbody>
</table>

#### Weighted Error Sum = .589E-01
APPENDIX B -- Continued

AIRCRAFT B CHECK CASE
NEWTON-RAPHSON DIGITAL DERIVATIVE MATCHING
1 APR 1974

INPUT DATA (I INDICATES TRUE OR YES, F INDICATES FALSE OR NO)

LONGITUDINAL CASE
DATA SOURCE CARD? T TAPE? F
DATA RATE IS 50 SAMPLES/SECOND ON SOURCE FILE (IF 0, DETERMINED FROM TIMES ON THE SOURCE FILE)
DIVIDED BY THINNING FACTOR OF 1
ON INPUT TAPE, 25 DATA WORDS PER RECORD, SPECIAL SIGNAL ORDER DEFAULT? T

PROGRAM OPTIONS
APRIORI WEIGHTING = .10E+01
B TIME HALVINGS IN EACH
ITERATIONS = 4 (ITERATION WILL STOP IF ERROR SUM CHANGES BY LESS THAN A FACTOR OF .10E-02)
CASE WILL BE STOPPED IF ERROR SUM IS GREATER THAN .10E+01

OUTPUT
PLOTS? T (NO PLOTS UNLESS FINAL ERROR SUM IS LESS THAN .10E+06)
NUMBER OF CONTROLS AND EXTRA SIGNALS TO BE PLOTTED = 8
PRINTED FLIGHT AND FINAL COMPUTED TIME HISTORIES F
EXTRA OUTPUT OF INTERMEDIATE STEPS FOR A DIAGNOSTIC AID? F
PUNCHED FINAL NON-DIMENSIONAL DERIVATIVES AND CONFIDENCE LEVELS? T
PUNCHED FINAL DIMENSIONAL MATRICES F

FLIGHT CONDITION AND VEHICLE CHARACTERISTICS (I, INDICATES VALUE OBTAINED FROM TIME HISTORY ON QBAR, W OR MACH)

METRIC UNITS? F
DYNAMIC PRESSURE = 39.0
VELOCITY = 415.2
MACH = .429
CENTER OF GRAVITY = .260
WING AREA = 85.0
SPAN = 15.05
CHORD = 5.98
IY = 1932.0
IX = 2228.6
IX7 = 11.6
WEIGHT = 2570.0
INSTRUMENT OFFSETS FROM CG
X=DIRECTION OFFSETS (+ INSTRUMENT IS FORWARD OF CG)
Y = 0.000
Z = 0.000

RAY I.G.
ALPHA 0.000 AN 0.000
NOR 0.000 AV 0.000

Y = 0.000

SIGNAL SCALING AND BIASES
SIGNS ALFA G V THET AN QDOT AX DE DC DG1 DG2 PHI ALT MACH QBAR
VAR I.G. F F F F T T T
VAR I.C. F F F F T T T
FIXED BIASES 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
SCALE FACT 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00
PLOT LIMITS MINIMUM 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
MAXIMUM 11.36 38.750 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00

MANEUVER 1 START TIME 11 36 38 750 STOP TIME 11 36 45 840
### AIRCRAFT B CHECK CASE

**INPUT MATRICES**

**A**  
4 BY 4

-0.4204E+01  1.1025E+01  0.0000E+00  0.0000E+00  
-0.3794E+01  3.6325E+00  0.0000E+00  0.0000E+00  
0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00  
0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00  

**B**  
4 BY 5

-0.6449E+01  0.0000E+00  0.0000E+00  0.0000E+00  
0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00  
0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00  
0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00  
0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00  

**D1**  
5 BY 5

1.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00  
0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00  
0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00  
0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00  
0.0000E+00  0.0000E+00  0.0000E+00  1.0000E+00

**APRA**  
4 BY 4

5.0894E+05  0.0000E+00  0.0000E+00  0.0000E+00  
3.0000E+00  1.0000E+00  0.0000E+00  0.0000E+00  
0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00  
0.0000E+00  0.0000E+00  0.0000E+00  1.0000E+00

**APRB**  
4 BY 8

1.0000E+05  0.0000E+00  0.0000E+00  0.0000E+00  
1.0000E+05  0.0000E+00  0.0000E+00  0.0000E+00  
0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00  
0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00  

**TOTAL NUMBER OF POINTS FOR MANEUVER** \( I = 354 \)
**AIRCRAFT & CHECK CASE**

STARTING VALUES: MACH = .429  ALPHA = 7.86  PARAM = 5.0000  CG = .263

<table>
<thead>
<tr>
<th>DIMENSIONAL DERIVATIVES / SEC / SEC**2</th>
<th>( \alpha )</th>
<th>( \gamma )</th>
<th>( \delta_1 )</th>
<th>( \delta_2 )</th>
<th>( \delta_1^\prime )</th>
<th>( \delta_2^\prime )</th>
<th>( \delta_1^\prime )</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-2.3764E+00</td>
<td>-1.00000*</td>
<td>-0.00000*</td>
<td>-0.00000*</td>
<td>0.00000*</td>
<td>0.00000*</td>
<td>0.00000*</td>
</tr>
<tr>
<td>B</td>
<td>-3.7543E+00</td>
<td>-3.63210</td>
<td>-0.99030*</td>
<td>-2.99030*</td>
<td>0.00000*</td>
<td>0.00000*</td>
<td>0.00000*</td>
</tr>
<tr>
<td>C</td>
<td>-1.99030*</td>
<td>-0.99030*</td>
<td>-8.35192*</td>
<td>0.00000*</td>
<td>0.00000*</td>
<td>0.00000*</td>
<td>2.39365*</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NON-DIMENSIONAL DERIVATIVES / DEG (ROTARY / RAD)</th>
<th>( \alpha )</th>
<th>( \gamma )</th>
<th>( \delta_1 )</th>
<th>( \delta_2 )</th>
<th>( \delta_1^\prime )</th>
<th>( \delta_2^\prime )</th>
<th>( \delta_1^\prime )</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.070547</td>
<td>0.003000*</td>
<td>0.000000*</td>
<td>0.000000*</td>
<td>0.000000*</td>
<td>0.000000*</td>
<td>0.000000*</td>
</tr>
<tr>
<td>B</td>
<td>-0.006354</td>
<td>-2.39365*</td>
<td>-0.000000*</td>
<td>-0.000000*</td>
<td>0.000000*</td>
<td>0.000000*</td>
<td>0.000000*</td>
</tr>
<tr>
<td>C</td>
<td>-0.006354</td>
<td>-2.39365*</td>
<td>-0.000000*</td>
<td>-0.000000*</td>
<td>0.000000*</td>
<td>0.000000*</td>
<td>0.000000*</td>
</tr>
</tbody>
</table>

( * ) INDICATES DERIVATIVE HELD FIXED DURING MATCH

NUMBER OF Unknowns = 9

ENTERING ITERATION LOOP

DIMENSIONAL DERIVATIVE MATRICES PER RADIAN, BIASES IN RADIANS.

<table>
<thead>
<tr>
<th>A</th>
<th>( \delta_1 )</th>
<th>( \delta_2 )</th>
<th>( \delta_1^\prime )</th>
<th>( \delta_2^\prime )</th>
<th>( \delta_1^\prime )</th>
</tr>
</thead>
<tbody>
<tr>
<td>-42684E+00</td>
<td>0.1300E+01</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
</tr>
<tr>
<td>-3.7543E+00</td>
<td>-1.00000*</td>
<td>-0.00000*</td>
<td>0.00000*</td>
<td>0.00000*</td>
<td>2.39365*</td>
</tr>
<tr>
<td>-1.567E+02</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
</tr>
<tr>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B</th>
<th>( \delta_1 )</th>
<th>( \delta_2 )</th>
<th>( \delta_1^\prime )</th>
<th>( \delta_2^\prime )</th>
<th>( \delta_1^\prime )</th>
</tr>
</thead>
<tbody>
<tr>
<td>-6.4989E+01</td>
<td>-0.0000E+00</td>
<td>-0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td></td>
</tr>
<tr>
<td>-6.2561E+01</td>
<td>-0.0000E+00</td>
<td>-0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td></td>
</tr>
<tr>
<td>-1.567E+02</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td></td>
</tr>
<tr>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C</th>
<th>( \delta_1 )</th>
<th>( \delta_2 )</th>
<th>( \delta_1^\prime )</th>
<th>( \delta_2^\prime )</th>
<th>( \delta_1^\prime )</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1.567E+02</td>
<td>-0.0000E+00</td>
<td>-0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td></td>
</tr>
<tr>
<td>-6.2561E+01</td>
<td>-0.0000E+00</td>
<td>-0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td></td>
</tr>
<tr>
<td>-1.567E+02</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td></td>
</tr>
<tr>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

VARIABLE BIASES: .1500E+01

ERRORS: .4895E+04  .4733E-04  .8074E+02  .6080E-03  .5171E-02

WEIGHTED ERRORS: .1810E+01  .5484E+01  .1603E+03  .5171E+02

ITERATION NUMBER 1 COMPLETED

<table>
<thead>
<tr>
<th>A</th>
<th>( \delta_1 )</th>
<th>( \delta_2 )</th>
<th>( \delta_1^\prime )</th>
<th>( \delta_2^\prime )</th>
<th>( \delta_1^\prime )</th>
</tr>
</thead>
<tbody>
<tr>
<td>-4.9555E-30</td>
<td>1.0000E+01</td>
<td>-0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td></td>
</tr>
<tr>
<td>-3.229E+01</td>
<td>-3.651E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td></td>
</tr>
<tr>
<td>-1.567E+02</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td></td>
</tr>
<tr>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B</th>
<th>( \delta_1 )</th>
<th>( \delta_2 )</th>
<th>( \delta_1^\prime )</th>
<th>( \delta_2^\prime )</th>
<th>( \delta_1^\prime )</th>
</tr>
</thead>
<tbody>
<tr>
<td>-5.102E-03</td>
<td>0.0000E+00</td>
<td>-0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td></td>
</tr>
<tr>
<td>-6.2561E+00</td>
<td>-0.0000E+00</td>
<td>-0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td></td>
</tr>
<tr>
<td>-1.567E+02</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td></td>
</tr>
<tr>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C</th>
<th>( \delta_1 )</th>
<th>( \delta_2 )</th>
<th>( \delta_1^\prime )</th>
<th>( \delta_2^\prime )</th>
<th>( \delta_1^\prime )</th>
</tr>
</thead>
<tbody>
<tr>
<td>-5.102E-03</td>
<td>0.0000E+00</td>
<td>-0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td></td>
</tr>
<tr>
<td>-6.2561E+00</td>
<td>-0.0000E+00</td>
<td>-0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td></td>
</tr>
<tr>
<td>-1.567E+02</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td></td>
</tr>
<tr>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td>0.0000E+00</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

VARIABLE BIASES: .1012E+01

ERRORS: .8600E-05  .1238E-04  .2221E-02  .6795E-05  .2477E-03

WEIGHTED ERRORS: .6688E+02  .8665E+00  .2714E+01  .2477E+01

ITERATION NUMBER 2 COMPLETED

WEIGHTED ERROR SUM = .6688E+01
APPENDIX B -- Continued

**A**
\[
\begin{bmatrix}
-4.511E+00 & .100E+01 & -4.511E+00 & .221E+02 \\
-3.222E+01 & -4.853E+00 & -3.222E+01 & .3216E+02 \\
-1.567E+02 & 0. & -1.567E+02 & 0. \\
.9916E+00 & 0. & .9916E+00 & 0. \\
\end{bmatrix}
\]

**B**
\[
\begin{bmatrix}
-5.149E-01 & 0. & -5.149E-01 & 0. \\
-6.271E+01 & 0. & -6.271E+01 & 0. \\
-1.354E+01 & 0. & -1.354E+01 & 0. \\
-1.00E+01 & 0. & -1.00E+01 & 0. \\
\end{bmatrix}
\]

**VARIABLE DIAS**
1.013E+01

**ERRORS**
.5865E-05  .136E-04  .2252E+02  .6054E-05  .1961E-03

**WEIGHTED ERRORS**
.5865E+00  .9547E+00  0.  .2421E+01  .1961E+01

**WEIGHTED ERROR SUM**
.523E+01

**ITERATION NUMBER 3 COMPLETED**

**A**
\[
\begin{bmatrix}
-4.562E+00 & .100E+01 & -4.562E+00 & .221E+02 \\
-3.196E+01 & -4.956E+00 & -3.196E+01 & .3216E+02 \\
-1.567E+02 & 0. & -3.196E+02 & 0. \\
.9916E+00 & 0. & .9916E+00 & 0. \\
\end{bmatrix}
\]

**B**
\[
\begin{bmatrix}
-5.192E-01 & 0. & -5.192E-01 & 0. \\
-6.266E+01 & 0. & -6.266E+01 & 0. \\
-1.354E+01 & 0. & -1.354E+01 & 0. \\
-1.00E+01 & 0. & -1.00E+01 & 0. \\
\end{bmatrix}
\]

**VARIABLE DIAS**
1.812E+01

**ERRORS**
.5931E-05  .137E-04  .2251E+02  .6012E-05  .1941E-03

**WEIGHTED ERRORS**
.5931E+00  .9632E+00  0.  .2405E+01  .1941E+01

**WEIGHTED ERROR SUM**
.5902E+01

**ITERATION NUMBER 4 COMPLETED**

**A**
\[
\begin{bmatrix}
-4.563E+00 & .100E+01 & -4.999E+00 & .221E+02 \\
-3.193E+01 & -4.999E+00 & -3.193E+01 & .3216E+02 \\
-1.567E+02 & 0. & -3.193E+02 & 0. \\
.9916E+00 & 0. & .9916E+00 & 0. \\
\end{bmatrix}
\]

**B**
\[
\begin{bmatrix}
-5.195E-01 & 0. & -5.195E-01 & 0. \\
-6.264E+01 & 0. & -6.264E+01 & 0. \\
-1.354E+01 & 0. & -1.354E+01 & 0. \\
-1.00E+01 & 0. & -1.00E+01 & 0. \\
\end{bmatrix}
\]

**VARIABLE DIAS**
1.112E+01

**ERRORS**
.5946E-05  .138E-04  .2251E+02  .5987E-05  .1937E-03

**WEIGHTED ERRORS**
.5946E+00  .9722E+00  0.  .2395E+01  .1937E+01

**WEIGHTED ERROR SUM**
.5898E+01

**ITERATION NUMBER 5 COMPLETED**

**ITERATION TERMINATING, ERROR WITHIN .001000 ROUND.**
### CONFIDENCE LEVELS FOR NEXT TO LAST ITERATION

<table>
<thead>
<tr>
<th>AC (DIMENSIONAL)</th>
<th>0.799E-34</th>
<th>0.799E-34</th>
<th>0.799E-34</th>
<th>0.799E-34</th>
<th>0.799E-34</th>
<th>0.799E-34</th>
<th>0.799E-34</th>
<th>0.799E-34</th>
<th>0.799E-34</th>
</tr>
</thead>
<tbody>
<tr>
<td>BC</td>
<td>0.799E-34</td>
<td>0.799E-34</td>
<td>0.799E-34</td>
<td>0.799E-34</td>
<td>0.799E-34</td>
<td>0.799E-34</td>
<td>0.799E-34</td>
<td>0.799E-34</td>
<td>0.799E-34</td>
</tr>
<tr>
<td>AC (NON-DIMENSIONAL)</td>
<td>0.799E-34</td>
<td>0.799E-34</td>
<td>0.799E-34</td>
<td>0.799E-34</td>
<td>0.799E-34</td>
<td>0.799E-34</td>
<td>0.799E-34</td>
<td>0.799E-34</td>
<td>0.799E-34</td>
</tr>
<tr>
<td>BC</td>
<td>0.799E-34</td>
<td>0.799E-34</td>
<td>0.799E-34</td>
<td>0.799E-34</td>
<td>0.799E-34</td>
<td>0.799E-34</td>
<td>0.799E-34</td>
<td>0.799E-34</td>
<td>0.799E-34</td>
</tr>
</tbody>
</table>
### Appendix B

#### Aircraft B Check Case

<table>
<thead>
<tr>
<th>Final Values</th>
<th>Mach</th>
<th>Alpha</th>
<th>Param</th>
<th>CG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>.429</td>
<td>.786</td>
<td>5.6206</td>
<td>.260</td>
</tr>
</tbody>
</table>

#### Dimensional Derivatives / SEC / SEC**2

<table>
<thead>
<tr>
<th></th>
<th>V</th>
<th>DE</th>
<th>DC</th>
<th>DC1</th>
<th>DC2</th>
<th>Delta</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>-4.0026</td>
<td>-1.00000*</td>
<td>0.00000*</td>
<td>.51967</td>
<td>0.00000*</td>
<td>0.00000*</td>
</tr>
<tr>
<td>M</td>
<td>-3.192019</td>
<td>-5.00000*</td>
<td>0.00000*</td>
<td>-6.215162</td>
<td>0.00000*</td>
<td>0.00000*</td>
</tr>
<tr>
<td>a</td>
<td>-15.686030*</td>
<td>0.000000*</td>
<td>0.000000*</td>
<td>-4.359820</td>
<td>0.000000*</td>
<td>0.000000*</td>
</tr>
</tbody>
</table>

#### Non-Dimensional Derivatives / Deg (Rotary / Rad)

<table>
<thead>
<tr>
<th></th>
<th>V</th>
<th>DE</th>
<th>DC</th>
<th>DC1</th>
<th>DC2</th>
<th>Delta</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>.075962</td>
<td>0.000000*</td>
<td>0.000000*</td>
<td>.0000000*</td>
<td>0.000000*</td>
<td>0.000000*</td>
</tr>
<tr>
<td>M</td>
<td>.005345</td>
<td>-6.666683</td>
<td>0.000000*</td>
<td>.016481</td>
<td>0.000000*</td>
<td>0.000000*</td>
</tr>
<tr>
<td>a</td>
<td>.006333*</td>
<td>0.000000*</td>
<td>0.000000*</td>
<td>-0.003377</td>
<td>0.000000*</td>
<td>0.000000*</td>
</tr>
</tbody>
</table>

(*) indicates derivative held fixed during match.

#### Variable Bias

- Mach: 1.312e+01
- Alpha: 1.312e+01
- Param: 1.312e+01
- CG: 1.312e+01

#### Final Dimensional Matrices

<table>
<thead>
<tr>
<th></th>
<th>4 by 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-.450E+30</td>
</tr>
<tr>
<td></td>
<td>-3.192E+01</td>
</tr>
<tr>
<td></td>
<td>.056E+02</td>
</tr>
<tr>
<td></td>
<td>-.9910E00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>5 by 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>-.5197E-01</td>
</tr>
<tr>
<td></td>
<td>.6764E+01</td>
</tr>
<tr>
<td></td>
<td>.8545E+01</td>
</tr>
<tr>
<td></td>
<td>-.1000E+01</td>
</tr>
</tbody>
</table>

#### Degrees

<table>
<thead>
<tr>
<th></th>
<th>AN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable Bias</td>
<td>1.112e+01</td>
</tr>
</tbody>
</table>

#### Errors

<table>
<thead>
<tr>
<th></th>
<th>Weighted Error Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Errors</td>
<td>.5945E-05</td>
</tr>
<tr>
<td>Weighted Errors</td>
<td>.5945E+00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Errors</th>
<th>219.31</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.69</td>
<td>5.92</td>
</tr>
</tbody>
</table>
| 5.40   | 5.90   | 5.90
Two sample plots from the MMLE program are shown. The plots as produced by the automatic plotter are shown within the heavy lines. Explanatory material is included to aid the user in implementing the program. Each plot is presented in two parts to avoid loss of detail from a large reduction. The title on each plot corresponds to the title on the output listing.

\[ \delta_{r'}, \text{deg} \]

\[ \delta_{a'}, \text{deg} \]

\[ \dot{r}, \text{deg/sec}^2 \]

\[ \dot{p}, \text{deg/sec}^2 \]

\[ a_y, \text{g} \]
\( \varphi, \text{deg} \)

\( \dot{r}, \text{deg/sec} \)

\( \dot{p}, \text{deg/sec} \)

\( \beta, \text{deg} \)

Date and time of run

---

Computed

Flight

AIRCRAFT A CHECK CASE

081774 1200 ET
APPENDIX B – Concluded

\[\delta_e, \text{deg}\]

\[a_n, \text{g}\]

\[\theta, \text{deg}\]

\[V, \text{ft/sec}\]

\[q, \text{deg/sec}\]

\[\alpha, \text{deg}\]

Date and time of run

09/17/74 - 10:35

AIRCRAFT B CHECK CASE

--- Computed
--- Flight
APPENDIX C

SETUP PROGRAM AND SUBROUTINES

Listings of the main program and the subroutines used in the SETUP program are presented together with supplementary information.

MAIN PROGRAM SETUP

Description: The main SETUP program sets several defaults and then reads the option cards to determine whether it is to read an input tape, punch a card deck, write an output tape, or perform any combination of these operations. It then directs the execution of the assigned tasks for each case.

Programming notes: As in the MMLE program, the program statement is needed on CDC 6000/7000 systems. On an IBM 360/370 system, DD cards perform this function. Cards 590 to 730 are concerned solely with setting the default values for DELTA as defined in the input description (p. 30).
APPENDIX C – Continued

Program listing:

```
PROGRAM SETUP(INPUT=PUNCH,OUTPUT=TAPE4,TAPE15,TAPE1=INPUT,
               TAPE2=PUNCH,TAPE3=OUTPUT)
COMMON /ALLDIM/ MAX,MIX
COMMON /OPTION/ TAPE,DECK,READ
COMMON /FLCONO/ ALPHA,TThETA,QQ,MACH,IX,II,IZ,IXZ,PHI,KIAS,
               LAT,LON,PARAM,FLT,CASE,AVG,DELTA,ST,ET,DETRIM
INTEGER ST(4),ET(4),FLT,CASE
NAMELIST /COND/ LAT,LON,FLT,CASE,ALPHA,TThETA,QQ,MACH,IX,II,IZ,
NAMELIST /DEL/ LAT,LON,DELTA(4),LATDEL(4),LATDEL(4),DEL
LOGICAL TAPE,DECK,READ,LONG,PARAM,FLT,CASE

DATA WRTI,HWRITI,PNC,HPUNCI,ROIK,HREADI,START/4HSTAR/
MAX=5
REWIND 15
DO 10 I=1,4
  LATDEL(I)=.FALSE.
10  LATDEL(I)=.FALSE.
  TAPE=.FALSE.
  DECK=.FALSE.
  READ=.FALSE.
  DETRIK=.FALSE.
  PARAM=.FALSE.
  CG=999.
  ALT=0.
  KIAS=0.
  PHII=0.
  FLT=6
  CASE=0
  LONG=L.FALSE.
C  READ OPTIONS
C  20 READ (1,1000) OPTN
   IF(OPTN.EQ.START) GO TO 50
   IF(OPTN.EQ.WRTI) GO TO 40
   TAPE=.TRUE.
   WRITE(3,2001)
30  READ=.TRUE.
   WRITE(3,2000)
   GO TO 20
   IF(OPTN.EQ.RDI) GO TO 30
   IF(OPTN.EQ.PNC) GO TO 20
   DECK=.TRUE.
   WRITE(3,2002)
50  GO TO 20
   IF(DECK.IEQ.0) CALL SETIN
   IF(READ) CALL RDSET
C  CASE LOOP
C  100 READ (1,1001) ST,ET
    IF(ST(1).LT.61) GO TO 200
    LATR=.FALSE.
   GO TO 115
   IF(DECK) CALL SETIN
   IF(READ) CALL RDSET
D  DELTA(12)=.FALSE.
```
READ (1,COND)
IF(LAT.RH)=FALSE
DEL=DELTA(1), or DELTA(2), or DELTA(3), or DELTA(4)
IF(LONG) GO TO 190
IF(ONOT.ÖEL) GO TO 130
120 LATDEL(I)=DELTA(I)
GO TO 190
130 DO 140 I=1,4
140 DELTA(I)=LATDEL(I)
GO TO 190
150 IF(ONOT.DEL) GC TO 170
DO 160 I=1,4
160 LONDEL(I)=DELTA(I)
GO TO 190
170 DO 180 I=1,4
180 DELTA(I)=LONDEL(I)
190 WRITE(I,200)FLT, CASE, ST, ET, LONG
IF(READ) CALL TAPERD
IF(DECK) CALL PNCH
GO TO 100
100 FORMAT(A_)
1001 FORMAT(3(312, I3),I5)
1002 FORMAT(1800TAPE WILL BE READ)
1003 FORMAT(250HOME TAPE WILL BE WRITTEN)
1004 FORMAT(250HOME DECK WILL BE PUNCHED)
1005 FORMAT(11I4,2X,6HFLT, I3,5X,4HCASE, I4,5X,4HTIME, =I4,5X, TO,)
- =I4,5X,14MLONGITUDINAL ? (1)
200 STOP
END
APPENDIX C – Continued

SUBROUTINE SETIN

Description: Subroutine SETIN initializes all information needed to punch the MMLE program deck. It sets several defaults and reads in the values desired. It also calls WINDIN to input predicted derivatives and COND1 to make any other input required by the user.

Subroutine listing:

```
SUBROUTINE SETIN
COMMON /COM/ MZLA, MZLO, S, SPAN, CBAR, CGLA, CGLO, METRIC, DIL0,
        XALF, XB, ZB, XAN, ZAX, XAY, SPS, XA, XA0, DLA, DLO
COMMON /DATA/ MBP, NMBP, NABP, MBP, MABP, MBP(I), NABP(I), BPA(I)
REAL DATA(1600), ODL0(I), DLA(I), ABP(I), MBP(I), NABP(I), BPA(I), XCA(I), YCA(I)
LOGICAL METRIC, LONG(I), LATR(I), RAD, PUNCH, CORECT, BODY, STAB,
        LAT, LON, DLA, DLO
NAMELIST /WIND/ NABP, NMBP, MBP, SPAN, CBAR, CGLA, CGLO, DEG, RAD,
        METRIC, LONG, LATR, MZLO, MZLA, NCLO, NCLA, WMLC, WMLA, PUNCH, XALF,
        XAN, ZAX, XAY, ZAY, SPS, XA, ZA, BODY, STAB
DEFAULTS
SPS=0.
X=0.
Z=5.
XAY=0.
ZAY=0.
XAN=0.
ZAX=5.
XALF=0.
CORECT=FALSE.
CGLA=0.
CGLO=0.
MZLA=5.
MZLO=5.
PUNCH=FALSE.
METRIC=FALSE.
BODY=FALSE.
DLA=FALSE.
DLO=FALSE.
WMLA=-99999.
WMLLO=-99999.
RAD=FALSE.

DO 5 I=1,300
5 DATA(I)=0.
DO 11 I=1,8
11 NBP(I)=0.
LONG(I)=TRUE.
10 LATR(I)=FALSE.
NPB=1.
NABP=1.
MPB=1.
LONG=FALSE.
LAT=FALSE.
READ 41, WIND
IF (ABS(XX)+ABS(ZH)+ABS(XALF)+ABS(XA)+ABS(YA)+ABS(ZAY))
   NE 0.3 CORECT=TRUE.
   GO TO 31
IF (.NOT.LATR(I)) GO TO 33
LONG(I)=FALSE.
30 IF (LONG(I)) GO TO 35
LAT=TRUE.
GO TO 45
35 LONG=TRUE.
```
APPENDIX C — Continued

40 CONTINUE
READ (1,1000) VEH
IF(LAT) READ (1,1051) DILA
IF(LON) READ (1,1051) DIL0
IF(DILA(1)+DILA(2)+DILA(3)+DILA(4)+DILA(5),GT,0.) DILA=.TRUE.
IF(DIL0(1)+DIL0(2)+DIL0(3)+DIL0(4)+DIL0(5),GT,0.) DLO=.TRUE.
IF(WMLA.LT.0.) GO TO 50
CALL LOADI(APRALA)
CALL LOADI(APRLA)
50 IF(WMLO.LT.0.) GO TO 60
CALL LOADI(APRAL0)
CALL LOADI(APRLO)
60 WRITE (3,2000) VEH,CGLA,CGLO,RAD
CALL WINDIN(VEH,NBP,NMBP,NADP,BODY,.TRUE.,RAC)
CALL COND
100 FORMAT(244)
1001 FORMAT(7F15.4)
2000 FORMAT(1H4,2H4,5X,27H WIND TUNNEL DATA, REF GG =.F5.3,7H (LAT),
- .F5.3,23H (LONG) PER NACIAN7 .L1)
RETURN
END

SETI 579
SETI 580
SETI 590
SETI 600
SETI 610
SETI 620
SETI 630
SETI 640
SETI 650
SETI 660
SETI 670
SETI 680
SETI 690
SETI 700
SETI 710
SETI 720
SETI 730
SETI 740
SETI 750
SETI 760
SETI 770
APPENDIX C – Continued

SUBROUTINE WINDIN

Description: Subroutine WINDIN reads in predicted derivatives, converting longitudinal data from the stability axes to the body axes if required.

Flow chart:
APPENDIX C – Continued

Programming notes: The loop from cards 770 to 1000 is written in a more expanded form than necessary to improve its efficiency.

Subroutine listing:
APPENDIX C -- Continued

C WRITE (3,2004) (MBP(J),J=1,NMBP)
IF(PRINT) WRITE(3,2005)(BP(J),J=1,NBP)
IF(BODY) RETURN

CONVERT STABILITY TO BODY AXES
DO 210 I=1,Z1
NNABP(I)=NABP(I)
DO 300 K=1,NABP
SA=SIN(ABP(K)/57.2958)
CA=COS(ABP(K)/57.2958)
DO 3GO L=1,NBP
IF(.NOT.LONG(L)) GO TO 100
DO 2_0 J=1,NMBP
TEMP=DATA(L,J,NNABP(K))4CA+DATA(L,J,NNABP(K))2SA
DATA(L,J,NNABP(K))2SA=TEMP
END

READ BREAKPOINTS
READ (1,1001) (ABP(J),J=1,NABP)
READ (1,1001) (MBP(J),J=1,NMBP)
READ (1,1001) (BP(J),J=1,NBP)
IF(BODY) RETURN

READ (1,1001) (RPJ),J=1,NBP
IF(PRINT) WRITE(3,2005)(BP(J),J=1,NBP)
READ 570 1 1 211
DO 210 I=1,Z1
DO 300 K=1,NABP
DO 3GO L=1,NBP
IF(.NOT.LONG(L)) GO TO 100
DO 2_0 J=1,NMBP
TEMP=DATA(L,J,NNABP(K))4CA+DATA(L,J,NNABP(K))2SA
DATA(L,J,NNABP(K))2SA=TEMP
END

READ (1,1001) (ABP(J),J=1,NABP)
READ (1,1001) (MBP(J),J=1,NMBP)
READ (1,1001) (BP(J),J=1,NBP)
IF(BODY) RETURN

CONVERT STABILITY TO BODY AXES
DO 210 I=1,Z1
NNABP(I)=NABP(I)
DO 300 K=1,NABP
SA=SIN(ABP(K)/57.2958)
CA=COS(ABP(K)/57.2958)
DO 3GO L=1,NBP
IF(.NOT.LONG(L)) GO TO 100
DO 2_0 J=1,NMBP
TEMP=DATA(L,J,NNABP(K))4CA+DATA(L,J,NNABP(K))2SA
DATA(L,J,NNABP(K))2SA=TEMP
END

FORMAT(1X,10F13.5)
WIND570
WIND580
WIND590
WIND600
WIND610
WIND620
WIND630
WIND640
WIND650
WIND660
WIND670
WIND680
WIND690
WIND700
WIND710
WIND720
WIND730
WIND740
WIND750
WIND760
WIND770
WIND780
WIND790
WIND800
WIND810
WIND820
WIND830
WIND840
WIND850
WIND860
WIND870
WIND880
WIND890
WIND900
WIND910
WIND920
WIND930
WIND940
WIND950
WIND960
WIND970
WIND980
300 CONTINUE
1001 FORMAT(8F10.4)
1002 FORMAT(4X,2D15.5)
1003 FORMAT(4X,3D15.5)
2001 FORMAT(1X,4D15.5) NOT A VALID DERIVATIVE NAME FOR THIS TYPE CASE
2002 FORMAT(1X,5D13.5)
2003 FORMAT(1X,8H ALPHA BREAKPOINTS/5X,10F13.5)
2004 FORMAT(1X,18H MACK BREAKPOINTS/5X,10F13.5)
2005 FORMAT(1X,18H PARAM BREAKPOINTS/5X,10F13.5)
RETURN
APPENDIX C – Continued

SUBROUTINE TAPERD

Description: Subroutine TAPERD supervises the reading of the input tape and obtains averages of the channels read in. It also writes the output file if desired. It calls TAPEIN, the user-supplied input routine, to do the actual reading of the input tape.

Subroutine listing:

```fortran
SUBROUTINE TAPERD
COMMON /OPTION/ TAPE, DECK, READ
COMMON /ELCONDA/ ALTA, THETA, O, W, MACH, IX, IX, IX, IX, W, PHI, CG, KIAS,
REAL MACH, IX, IX, IX, IX, KIAS, AVG(W), DATA(W), I0, 100
INTEGER ST(W), ET(W), TIME(I, I), JST(I)
NFRAME=10
DO 1 I=1,40
1  AVG(I)=0.
NPT=3
20 CALL TAPEIN(DATA, TIME, NFRAME, ST, ET)
NFR=ABS(NFRAME)
DO 10 J=1,NFR
10  IF(TAPE) WRITE (6) (TIME(J, I), I=1, NPT)
DO 30 J=1,NPT
30  AVG(J)=AVG(J)+DATA(J, I)
            NPT=NPT+1
IF(NPT,LE,10) GO TO 100
DO 50 J=1,NPT
50  AVG(J)=AVG(J)/NPT
    NPT=NPT+1
DO 100 J=1,ST(W)
100 CONTINUE
DO 200 J=1,NPT
200  WRITE(*,2000) NPT, JST(J), TIME(J, I)
DO 210 I=1,44
210 CONTINUE
DO 300 I=1,44
300 CONTINUE
END
```

136
APPENDIX C – Continued

SUBROUTINE PNCH

Description: Subroutine PNCH dimensionalizes coefficients and punches the MMLE card deck.

Programming notes: Through card 540, this subroutine contains some computations and initializations used in all cases. Then cards 590 to 980 contain the lateral-directional dimensionalization and computations; cards 1030 to 1390 contain this information for the longitudinal cases. The remaining cards control the punching of the output deck.

Subroutine listing:
APPENDIX C  --  Continued

C LATERAL
C
WMAR=ABS(WMLA)
IF(.NOT.CORECT) GO TO 10
DCG=CGFLOT-CGLA
XBC=XB
YXY=CAY
IF(X<NE.X,0) XB=XBC*DCG*CBAR
IF(X>NE.X,0) XAYCX&YCYDCG*CBAR
WRITE(2,1001)XBC,ZM,XAY,CAY
10 TYPE=ALAT
RD(3,3)=1.
QSMV=QSMV*AO
QSB=QSB*SPAN*CAC
QSBIX=QSBIX/1X
QSBIV=QSBIV*BOV
QCG=QCG*CBAR/SPAN
A(1,1)=QSMV*(X(1)+DCG*X(1))
A(1,2)=SIN(ALPR)
A(1,3)=QSBIV*X(4)
A(4,2)=QSBIV*X(5)
A(4,3)=1.
A(1,3)=-QSBIV*ALPR
A(2,1)=QSBIV*X(6)
A(1,3)=QSBIV*X(7)
A(4,3)=CST/CT
A(4,4)=CST/CT
DO 2: I=1,6
J=I+5
R(1,J)=QSMV*X(J)
R(2,J)=QSBIV*X(J)
R(3,J)=QSBIV*X(J+2)+DCG*X(J+1)
IF(.NOT.DELTA(I)) GO TO 20
BB11,1=1.
BB12,1=1.
BB13,1=1.
20 CONTINUE
GO TO 260

C LONGITUDINAL
C
300 WHAR=ABS(WHLA)
IF(.NOT.CORECT) GO TO 11
DCG=CGFLOT-CGLO
XALFC=XALF
XANC=XAN
IF(XALFC<NE.X,0) XALFC=XALFC*DCG*CBAR
IF(XANC<NE.X,0) XANC=XANC*DCG*CBAR
WRITE(2,1003)XALF,XANC,ZRX
110 TYPE=ALON
WRITE(2,3010)
QSMV=QSMV*V
- 8H, SPAN=, F6.2, B  
- 2(3I2,13.1X) 
 305 FORMAT (2HOI, 7X, 1I/TF10.1) 
 330 FORMAT (7FENDCASE) 
END
APPENDIX C — Continued

SUBROUTINE INTERP

Description: Subroutine INTERP interpolates predicted derivative data tables to obtain the nondimensional derivatives for a particular flight condition.

Programming notes: The subroutine first brackets the Mach number and angle of attack of the flight condition between breakpoints of the predicted data; it also selects the correct set of predicted data depending on the value of PARAM. The interpolation is divided into four sections. The interpolation occurs in one of the four sections on the basis of how many Mach and angle-of-attack breakpoints are specified. If only one breakpoint is specified, the required code changes slightly, because there are not two points to interpolate between.

Subroutine listing:

```fortran
SUBROUTINE INTERP(DATA,NBP,NMBP,NABP, XI)
C INTERPOLATES WIND TUNNEL DATA
C
COMMON /FLCOND/ ALPHA,THETA,OM,DELTA,MACH,TH,THI,THII,PHA,PHI,CG,KIAS, INTE 40
- ALT,LONG,PARAM,PIT,CASE,AVG,DELTA,ST,ET,DELIM INTE 50
COMMON /WTDATA/ NCLA,NCLO,ABP,MBP,NCMAX,LONGWT INTE 60
REAL MACH,IX, IY,IZ, IXZ,KIAS, AVG,DATA(NBP,NMBP,NABP), XIC INTE 70
- DATA(NBP,NMBP,NABP), XIC INTE 80
INTEGER STKWI, ET, CASE, AVG, OELTA, LONGWT, LONG INTE 90
LOGICAL OELTA(4), LONG, LONGWT INTE 100
C FINO CORRECT SET OF DATA
L=I
DO 60 II=I,NBP
XOR=IX OR. (NOT.LONG.AND..NOT.LONGWTII)
60 IF((PARAM.EQ.BP{II).OR.PARAH_BPIIII.EQ.O.) .AND. XORF)
L=II
C BRACKET ALPHA
IF(NABP.EQ.I) GO TO 50
DO 20 W_ J=2,NABP
IF(ALPHA.GT. ABP(J)) GO TO 10
EAP=(ALPHA-ABP(J-1)/(ABP(J)-ABP(J-1))
IF(EAP.LT.O.) EAP=0.
GO TO 50
J=NABP
C BRACKET MACH NUMBER
IF(NMBP.EQ.I) GO TO 100
DO 25 I=2,NMBP
IF(MACH.GT. MBP(I)) GO TO 20
EMN=(MACH-MBP(I-1))/(MBP(I)-MBP(I-1))
IF(EMN.LT.O.) EMN=0.
GO TO 30
25 CONTINUE
I=NMBP
EMN=1
30 IM=1-1
IF(NMBP.EQ.I) GO TO 120
DO 90 K=1,NCMAX
JK=IK-1*NABP+J
JM=JK+1
PA=DATA(L,I,JK-1)*EMN+DATA(N,IM,JK)
PB=DATA(L,IM,JK-1)*EMN+DATA(N,IM,JK-1)
90 X(K)=(PA-PB)*EMN+POL
GO TO 200
100 IF(NMBP.EQ.I) GO TO 140
DO 100 I=1,NMBP
100 IF ONLY 1 MACH BREAKPOINT
C INTERPOLATE IF ONLY 1 ALPHA BREAKPOINT
120 DO 130 K=1,NCMAX
130 Y(K)={DATA(L,1,IK-1)*EMN+DATA(L,1,IK)}
GO TO 200
140 END
```
APPENDIX C – Continued

C IF ONLY 1 ALPHA AND I MACH BREAKPOINT
140 GO TO 150 K=1,NMAX
150 X(K)=DATA(L,1,K)
220 RETURN
END

SUBROUTINE PMAT

Description: Subroutine PMAT punches a matrix on cards in an 8F10.5 format.

Subroutine listing:

```
SUBROUTINE PMAT(A)
  COMMON /ALLOIM/ MAX, MIX
  REAL A(I)
  II=A(MAX)
  JJ=A(_MAX)
  WRITE(2,1001) (A(K),K=I,KEND,MAX)
  CALL ASPIT(A)
  FORMAT(AW,4X,12,110)
RETURN
END
```

SUBROUTINE PMAT1

Description: Subroutine PMAT1 punches a matrix on cards in an 8E10.3 format.

Programming notes: This subroutine is needed in addition to PMAT because the APRA and APRB matrices may contain large values but do not need as many significant figures as other matrices.

Subroutine listing:

```
SUBROUTINE PMAT1(A)
  COMMON /ALLOIM/ MAX, MIX
  REAL A(I)
  II=A(MAX)
  JJ=A(_MAX)
  WRITE(2,1001) (A(K),K=I,KEND,MAX)
  CALL ASPIT(A)
  FORMAT(AW,4X,12,110)
RETURN
END
```

142
APPENDIX C — Continued

SUBROUTINE RDSET

Description: Subroutine RDSET is user supplied; the subroutine listed here is a sample. This subroutine should do any initialization or input required before calling subroutine TAPEIN.

Subroutine listing:

```
SUBROUTINE RDSET
  COMMON /TAPODAT/ NWOR(ICHAN)
  INTEGER ICHAN(40)
  READ (1,1000) NWOR
  READ (1,1000) ICHAN
  WRITE(3,2000)NWOR,ICHAN
  1000 FORMAT(16I5)
  2000 FORMAT(20H0INPUT FILE CONTAINS,5,22M DATA WORDS PER RECORD/
               9M CHANNELS/(10X,20I5))
  RETURN
END
```
APPENDIX C — Continued

SUBROUTINE TAPEIN

Description: Subroutine TAPEIN is user supplied; the subroutine listed here is a sample. This subroutine should be written to read data in the form available for a particular flight program. The comment cards and sample program illustrate the conventions required for interface with the rest of the program.

Subroutine listing:

```fortran
SUBROUTINE TAPEIN(DATA,TIME,NFRAME,ST,ET)  TAPE 9
  COMMON /TAPOAT/,NWORD,ICHAN
  INTEGER ST(N),ET(N),TIME(N,100),ICHAN(N),IT(4)
  REAL DATA(N,100),RECORD(150)
  IST=ST(4)*1000+ST(3)*60+ST(2)+ST(1)
  IET=ET(4)*1000+ET(3)*60+ET(2)+ET(1)
  I=0
  READ (15),(ITM,I),J=1,NWORD
  IF(ITM.LT.IST) GO TO 10
  DO 30 J=1,4
  TIME(J)=ITM
  DO 40 J=1,40
  DATA(J,J)=0
  IF(ICHAN(J).EQ.0) GO TO 40
  DATA(J,J)=RECORD(ICHAN(J))
  40 CONTINUE
  IF(I.GE.NFRAME) RETURN
  IF(ITM.GE.IET) GO TO 100
  READ (15),(ITM,I),J=1,NWORD
  ITM=ITM+1000*IT(3)+60*IT(2)+60*IT(1)
  GO TO 20
  NFRAME=-I
  RETURN
END
```
APPENDIX C -- Continued

SUBROUTINE COND1

Description: Subroutine COND1 is user supplied, and is described by the comment cards.

Subroutine listing:

```fortran
SUBROUTINE COND1

C THIS SUBROUTINE SHOULD INCLUDE ANY INITIALIZATION NEEDED
C FOR SUBROUTINE COND TO DETERMINE THE FLIGHT CONDITION
C TYPICAL ITEMS INCLUDED HERE MIGHT BE TABLES OF INERTIAS AS A
C FUNCTION OF GROSS WEIGHT
C ANY DATA MAY BE PASSED TO SUBROUTINE COND THROUGH A LABELLED
C COMMON BLOCK TABLE/
C SUBROUTINE SUPPLIED IS A NULL SUBROUTINE
C RETURN
END
```

SUBROUTINE COND

Description: Subroutine COND is user supplied. It automatically obtains the flight condition from the channel averages computed by TAPERD. The subroutine listed illustrates the method of doing this.

Subroutine listing:

```fortran
SUBROUTINE COND

C THIS SUBROUTINE SHOULD SPECIFY THE FLIGHT CONDITION AND OTHER
C PARAMETERS NOT READ IN THROUGH NAMELIST /COND/
C AVG CONTAINS THE AVERAGE VALUES OF EACH CHANNEL READ OFF THE INPUT
C TAPE IF THERE WAS ONE READ
C THE USER MAY CHOOSE TO USE THESE AVERAGE VALUES FOR THE FLIGHT
C CONDITION INSTEAD OF READING IT IN
C FOR INSTANCE, IF ALPHA IS TO BE OBTAINED FROM THE CHANNEL
C STATEMENT COND 90
C ALPHA=AVG(1)
C WOULD BE INCLUDED HERE
C THE SEVERAL EXTRA CHANNELS AVAILABLE MAY BE USED TO OBTAIN
C FUEL WEIGHS OR OTHER QUANTITIES NEEDED TO COMPUTE THE INERTIAS
C THE SUBROUTINE SUPPLIED OBTAINS ALPHA,THETA,PHI,DETRIM,O,V,ANO
C FROM SIGNAL AVERAGES AND COMPUTES O AND V FROM ALTITUDE
C AND KIAS(NOT INDICATED AIRSPEED) IF THESE ARE MORE REASONABLE
C AVAILABLE (INDICATED BY A NON-ZERO VALUE OF KIAS)
C COMMON /OPTION/TAPE,DECK,READ
COMMON /COND/ ALPHA,THETA,PHI,DETRIM,O,V,ALPHA,AVG(40)
INTEGER ST(4),ET(4),OPT,CASE,PARAM,FLOAT,CASE,AVG,DELTA.E,ET,DETRIM
REAL MACM,IX,II,IZ,IXZ,KIAS,AVG(40)
LOGICAL DELTA,E,ET,DETRIM

IF (.NOT.READ) GO TO 20
ALPHA=AVG(1)
THETA=AVG(1)
PHI=AVG(1)
DETRIM=AVG(1)
O=AVG(1)
V=AVG(1)
MACM=AVG(1)
RETURN
IF(KIAS.EQ._.) RETURN
Q=(KIAS_0582)=_2
DALT=ALT_001
V=1.68*KIAS*EXP(-DALT*(.1375+.000975*DALT))
RETURN
END
```

145
APPENDIX C – Concluded

SUBROUTINE LOAD1

Description: Subroutine LOAD1 reads a matrix from cards.

Subroutine listing:

SUBROUTINE LOAD1(A)
COMMON /ALLDIM/ MAX, MIX
REAL A(I)
MAX3=1*MAX
READ (1,1000) A(MAX3),II,JJ
A(MAX3)=II
A(2*MAX3)=JJ
KE=(JJ-1)*MAX
DO 10 IC=1,II
  KEND=IC+KE
  READ (1,1001) (A(K),K=I,KEND,MAX)
10 FORMAT(A10,4)
RETURN
END

SUBROUTINES ASPIT, AMAKE, AND AZOT

Subroutines ASPIT, AMAKE, and AZOT are identical to those used in the MMLE program.
APPENDIX D

SAMPLE CASE FOR THE SETUP PROGRAM

This appendix presents a sample check case for the SETUP program.

INPUT CARDS

PUNCH DECK
START
$END
NH=3,NA=2,MC=8,LO=15,SPAN=15,CR=4,5=100,END
SAMPLE

.O
.C0 2
.1 .5
.O .4
.CD 2
.05 .1
.07 .12
.CLA 1
.07 .065
.COA 1
.01 .015
.CHA 1
-.005 -.006
.CLOE 1
.01 .01
.CMD 1
-.009 -.011
.CMD 1
-.009 -.011

10251000 11210000 SAMPLE CASE 1
$END
.IX=30,,.IY=20,JO=.FT=20,.,.W=2500,,.LONG=1,
.RE=1,CASE=1,ALPHA=45,MAC=.5,END
10252000 12522000 SAMPLE CASE 2
$END
.CASE=2,0=60,.,.W=900,.ALPHA=3,MAC=.5,END
-1
### Output Listing

**Sample Wind Tunnel Data**

<table>
<thead>
<tr>
<th>Sample</th>
<th>Wind Tunnel Data</th>
<th>Ref CG (.250 LAT), (.250 LONG)</th>
<th>Per Radian F</th>
</tr>
</thead>
<tbody>
<tr>
<td>CL</td>
<td>.108E+00, .500E+00</td>
<td>.400E+00</td>
<td></td>
</tr>
<tr>
<td>CD</td>
<td>.500E+00, .199E+00</td>
<td>.499E+00</td>
<td></td>
</tr>
<tr>
<td>CLA</td>
<td>.400E-01, .199E+00</td>
<td>.599E+00</td>
<td></td>
</tr>
<tr>
<td>CLAE</td>
<td>.400E-01, .199E+00</td>
<td>.599E+00</td>
<td></td>
</tr>
<tr>
<td>CDA</td>
<td>.400E+00</td>
<td>.500E+00</td>
<td></td>
</tr>
<tr>
<td>CMA</td>
<td>.400E-01, .199E+00</td>
<td>.599E+00</td>
<td></td>
</tr>
<tr>
<td>CLDE</td>
<td>.400E-01, .199E+00</td>
<td>.599E+00</td>
<td></td>
</tr>
<tr>
<td>CMDE</td>
<td>.400E-01, .199E+00</td>
<td>.599E+00</td>
<td></td>
</tr>
<tr>
<td>CMQ</td>
<td>.400E-01, .199E+00</td>
<td>.599E+00</td>
<td></td>
</tr>
<tr>
<td>Alpha Breakpoints</td>
<td>.1000</td>
<td>.7500</td>
<td></td>
</tr>
<tr>
<td>Mach Breakpoints</td>
<td>.4000</td>
<td>.3000</td>
<td></td>
</tr>
<tr>
<td>Param Breakpoints</td>
<td>.1000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FLIGHT L</td>
<td>CASE 1</td>
<td>TIME</td>
<td>10</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>------</td>
<td>----</td>
</tr>
<tr>
<td><strong>CG NOT SPECIFIED, DEFAULTED TO WIND TUNNEL</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ALPHA = 4.60, MACH = .500, V = 450.0, CG = .250, PARAM = 0.0000</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>-5841E+30</td>
<td>1000E+01</td>
<td>0.</td>
<td>-3.</td>
</tr>
<tr>
<td>-4338E+01</td>
<td>-5000E+00</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>-3970E+01</td>
<td>3.</td>
<td>0.</td>
<td>1321E+02</td>
</tr>
<tr>
<td>0.</td>
<td>1000E+01</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>B</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>-6192E-01</td>
<td>0.</td>
<td>-3.</td>
<td>4234E-01</td>
</tr>
<tr>
<td>-7907E+01</td>
<td>0.</td>
<td>0.</td>
<td>3056E+00</td>
</tr>
<tr>
<td>0.</td>
<td>0.</td>
<td>0.</td>
<td>0.</td>
</tr>
</tbody>
</table>
**APPENDIX D — Continued**

CG NOT SPECIFIED, DEFAULTED TO WIND TUNNEL.

**FLIGHT 1  CASE 2  TIME  10  25  12  0  TO  10  25  22  0  LONGITUDINAL**

**A**

\[
\begin{array}{cccc}
\text{A} & \text{B} & \text{C} & \text{D} \\
-6.374E+30 & -1.3E+31 & -0.6 & -0.6 \\
-5.270E+31 & -5.435E+30 & 0.6 & 0.6 \\
-5.955E+31 & 0.6 & -0.6 & -0.6 \\
0 & 1.003E+01 & 0.6 & 0.6 \\
\end{array}
\]

**B**

\[
\begin{array}{cccc}
\text{A} & \text{B} & \text{C} & \text{D} \\
-8.884E-01 & -0.6 & 0.6 & 0.6 \\
-9.659E+31 & 0.6 & 0.6 & 0.6 \\
-9.405E+30 & -0.6 & -0.6 & -0.6 \\
0 & 0.6 & 0.6 & 0.6 \\
\end{array}
\]

\[\alpha = 3.00 \quad \text{MACH} = .550 \quad Q = 60.0 \quad V = 500.0 \quad \text{CG} = .250 \quad \text{PARAM} = 0.0000\]
APPENDIX D – Concluded

PUNCHED CARD OUTPUT LISTING

SAMPLE FLIGHT 1 CASE 1 MACH=.500 ALPHA= 4.00 PARAM= 0.00
INPUT GROSWT= 2500. ,IX= 360. ,IY= 2000. ,IZ= 2000. ,IXZ= 10.0., Q= 50.0 ,V= 450.0 ,PUNCH=F ,TIMESC=.5 ,BOTH=T., ZMAX(3)=1000.,
WMAPR=0.,+ALPHA= 4.00 ,MACH=.500 ,CG= .250 ,PARAM= 0.3000.,
LONG=T, S= 100., ISMALL= 15.00 ,CBAR= 6.00 ,SPS= 0., TEND
10251 0 1021 0 0
A 4 4
-5.6867 1.00000 -6.00000 -6.00000
-4.38313 -5.06030 0.00000 0.99000
-3.97020 0.00000 -6.30000 -32.17200
0.00000 1.00000 0.00000 0.00000
B 4 4
-0.08192 -0.00000 -0.00000 -0.00000 34.204
-7.90687 0.00000 0.00000 0.00000 30.000
-4.50430 -0.00000 -0.00000 -0.00000 -5.90024
6.00000 0.00000 0.00000 0.00000 6.00000
ENDCASE

SAMPLE FLIGHT 1 CASE 2 MACH=.550 ALPHA= 3.00 PARAM= 0.00
INPUT GROSWT= 2500. ,IX= 360. ,IY= 2000. ,IZ= 2000. ,IXZ= 10.0., Q= 60.0 ,V= 500.0 ,PUNCH=F ,TIMESC=.5 ,BOTH=T., ZMAX(3)=1000.,
WMAPR=0.,+ALPHA= 3.00 ,MACH=.500 ,CG= .250 ,PARAM= 0.3000.,
LONG=T, S= 100., ISMALL= 15.00 ,CBAR= 6.02 ,SPS= 0., TEND
10251 2 0 10252 0
A 4 4
-5.59347 1.00000 -6.00000 -6.00000
-5.27667 -5.40000 0.00000 0.00000
-5.86510 0.00000 -6.30000 -32.17200
0.00000 1.00000 0.00000 0.00000
B 4 5
-0.08192 -6.00000 -0.00000 -0.00000 32.165
-9.50687 0.00000 0.00000 0.00000 -27.534
-5.40430 -8.00000 -0.00000 -0.00000 -7.16407
6.00000 0.00000 5.00000 0.00000 5.00000
ENDCASE
APPENDIX E

SUMMARY PROGRAM AND SUBROUTINES

Listings of the main program and the subroutines used in the SUMMARY program are presented together with supplemental information.

MAIN PROGRAM SUMMARY

Description: The main program SUMMARY sets defaults, reads the NAMELIST, and initializes variables.

Program listing:

```plaintext
PROGRAM SUMMARY(INPUT,OUTPUT,TAPE5,TAPE1=INPUT,TAPE3=OUTPUT) MAIN 0
C SUMMARY PLCT PROGRAM FOR MLE DATA
C COMMON /ALLDIM/ MAX,Mix
COMMON /ALMC/ HGT
COMMON /FDATA/ NCLA,NCLD,MP,MB,NP,NMAX,LONG
COMMON /CGCOD/ SHIFT,CGLA,CGLD,CGOE
COMMON /NBRP/ NMP,NBP,NP,NPARAM
COMMON /INS/ NLCT,WTPLCT
COMMON /SUMDAT/ YLOC,XSKIP,ACLE,ASCAL,SYSTEM,AIM,TARLT,FDATA,
- FDATAC,TITLE
COMMON /PSCL/ CFAC,TITLE,IMPI,ITY,TLEN,II,NPARAM
REAL TITLE(50),MBP(16),BP(16),DATA(500),FDATA(500),
- FDATAC(500),BUF(1224),FDATC(200),AMDF(12)
- TPLRT(2),MLB(2),PLAR(2)
LOGICAL PRINT,LONG(8),LATR,DEG,PO,BODY,STAB,SHIFT,WTPLCT
DATA MLAH/4HPARA,IMM/
NAMELIST /WIND/ N_P,NA_°,NMq,DNCLA,NCLO,RAD,DEG,BODY,STAB,
- LONG,LATR,PRINT,CGL,DGL,NPARAM,SHIFT,WTPLCT,CFAC,
- AMIN,AMAX,ASCAL,TLEN,YPO,TARLT,CGOE,
NBUF=L_7 MAX=4 READ (1,1000) TITLE WRITE (3,2000) TITLE
HGT=.97 SHIFT=.FALSE.
CBAR=0.
SPAN=1._90
NPARAM=6
NCLA=6
NCLD=6
MBP=1
CGLA=.25
CGLD=.25
NB=1
NMP=1
PRINT=.FALSE.
DO 5 I=1,500
  5 DATA(I)=0.
DO 10 I=1,3. LONG(I)=.FALSE.
  10 MLAH=I.
RAD=.FALSE.
STAB=.TRUE.
BODY=.FALSE.
WTPLCT=.FALSE.
CFAC=.FALSE.
AMIN=.FALSE.
ASCAL=.FALSE.
TLEN=10.
XPO=0.
READ (1,WIND)
NPARAM=PARAM
COMMON/CRAN/SPAN
```

152
APPENDIX E — Continued

YLENZ=YLENZ/2, MAIN 570
YLOC=0, MAIN 590
ASCAL2=ASCAL2*2, MAIN 590
ALEN=(AMAX-AMIN)/ASCAL2 MAIN 610
XSKIP=ALEN*XDIST/2, MAIN 610
YSTEP=YLENZ/2, MAIN 620
DO 22 I=1,6, MAIN 630
C IF(LATR(1).NE.FALSE.), MAIN 640
21 READ WIND TUNNEL DATA MAIN 650
CALL WINDIN(DATA,NBP,NMP,NABP,BODY,PRINT,RAD) MAIN 660
IF(SHIFT) WRITE(1,2001)GGLA,GGLO MAIN 670
IF(CRFACT,N.E.0.) WRITE(1,3002)CRFACT MAIN 680
INTS=1 MAIN 690
INTZ=1 MAIN 700
DO 15 I=1,NBP MAIN 710
15 IF(ABP(I).LT.AMIN) INTS=I+1 MAIN 720
IF(ABP(I).GT.AMAX) INTZ=I MAIN 730
NCMX=NCMX*2 MAIN 740
NO=NO+2 MAIN 750
TABLE1=MLAB(1) MAIN 760
TABLE2=MLAB(2) MAIN 770
IF(NPARAM,L.E.0) GO TO 25 MAIN 780
NO=NPARAM MAIN 790
TABLE1=PLAB(1) MAIN 800
TABLE2=PLAB(2) MAIN 810
25 ND2=ND/2 MAIN 820
C READ FLIGHT DATA MAIN 830
CALL FLIGHT(NCMP,ND,FDATA,FDATA) MAIN 840
CALL FACTOR(.T,87_02) MAIN 850
CALL PLOT(0,0,-3) MAIN 860
READ PLOTTING INSTRUCTIONS MAIN 870
CALL INSTR MAIN 880
IF(NPLOT.LT.0) GO TO 56 MAIN 890
MAKE PLOTS MAIN 900
DO 50 II=1,NPLOT MAIN 910
50 CALL SUMPLT(FDATA,FDATA,ALFS,NO2,DATA,NBP,NMP,NABP) MAIN 920
GO TO 30 MAIN 930
STOP MAIN 940
END MAIN 950

153
APPENDIX E — Continued

SUBROUTINE FLIGHT

Description: Subroutine FLIGHT reads and sorts flight data.

Programming notes: Data are stored in the arrays FDATA and FDATAC. The FDATA array contains derivative values, and the FDATAC array contains confidence levels. Note that the sign of the X and Z coefficients is changed for longitudinal data to agree with the more common N and A (axial) coefficients. The flight $C_{\alpha_m}$ and $C_{\beta_n}$ are shifted to the wind-tunnel reference center of gravity if $\text{SHIFT} = T$.

LONLOC and LATLOC give the positions of data in the A and B matrices considered as vectors.

Subroutine listing:
APPENDIX E — Continued

110 IND$=INDEX+NO2
    NCAS=NCASE(INDEX)+1
    IF(SHIFT#1#(3)EQ(CGLA-C0)*C0B#(1)
      GO TO 120
    IF(1.0E-4) GO TO 120
    FDATA(1,INDEX,NCAS)=6*(LATLOC(I))
    FDATAC(1,INDEX,NCAS)=AC(LATLOC(I))
    GO TO 115
120 FDATA(I,INDEX,NCAS)=6*(LATLOC(I))
    FDATAC(I,INDEX,NCAS)=AC(LATLOC(I))
    CONTINUE
130 CONTINUE
150 NCAS=INDEX+NCAS
    FDATA(2,INDEX,NCAS)=ALPHA
    FDATAC(2,INDEX,NCAS)=MACH
    FDATAC(2,INDEX,NCAS)=PARAM
    FDATAC(2,INDEX,NCAS)=C0
    GO TO 25
100 FORMAT(184,4F10.4)
200 FORMAT(184,5X,144,4F10.4)
END
APPENDIX E – Continued

SUBROUTINE INSTR

Description: Subroutine INSTR reads plotting instructions.

Programming notes: The instructions are passed to the rest of the program in the following form:

NPlot – number of coefficients to be plotted.

LATLON – 1 if lateral data, 2 if longitudinal data.

PARM, TOL – parameter value and tolerance.

LL – number of the predicted derivative data set corresponding to LATLON and PARM.

IDER – parameter numbers that correspond to the coefficients to be plotted.

YMIN, YMAX – minimum and maximum values for the ordinates.
APPENDIX E -- Continued

Subroutine listing:

```
SUBROUTINE INSTR
READS INSTRUCTIONS ON COEFFICIENTS TO PLOT, SCALES TO USE,
AND THE PARAMETER AND TOLERANCE FOR FLIGHT POINTS.
DO NOT OVERLAY THIS SUBROUTINE AS START, DERIV, SMIN AND SMAX MUST
BE PRESERVED.
COMMON /NBPS/ MMNP, NBP, NPARAM
COMMON /TDATA/ KOLK, NDLO, ABO, MAB, NC, NCMAX, LONG
COMMON /INGP/ NPLT, WPTPL
COMMON /SELECT/ PARAM, TOL, INEF, YMIN, YMAX, LATLON, DERIVL, LL, WPTPL
REAL DERIV1(1), SMIN1(1), SMAX1(1), DERIV2(1), YMIN1(1), YMAX1(1),
- DERIV3(1), SMIN2(1), SMAX2(1), DERIV4(1), YMIN2(1), YMAX2(1),
INTEGER IDER(1)
LOGICAL WPTPL, WPTPL, LONG1(18)
DATA END, ALON, ALAT, BLAK, STAR, MEND, MLONG, WLATR, 1H, 4HSTAR/
DATA DER(3), CYC, MCL, 3MCL, 3MCP, 3MC, 3MCYD, 4MC, 2MCDA,
- 4MCDB, 4MCYR, 4MDLR, 4MDW, 4MCYI, 4MCDD, 4MCDO1, 4MCDO2,
- 4MCDO3, 4MCDO4, 4MCDO5, 4MCDO6, 4MCDO7, 4MCDO8,
- 4MCDO9, 4MCDO10, 4MCDO11, 4MCDO12, 4MCDO13,
- 4MCDO14, 4MCDO15, 4MCDO16, 4MCDO17, 4MCDO18,
- 4MCDO19, 4MCDO20, 4MCDO21, 4MCDO22,
IF (START .NE. START) READ (1, 100) DERIV1, SMIN1, SMAX1
START = STAR
NPLOT = 0
IF (DERIV1(1) .EQ. 0) GOTO 125
LATLON = 1
IF (DERIV1(1) .EQ. 4) ALON = LATLON
NC = 19
IF (LATLON .EQ. 1) NC = 21
PARAM = SMIN1
TOL = SMAX1
READ (1, 101) DERIV1, PARAM, TOL
DO 25 I = 1, NC
25 WRITE (5, 102)(DERIV1(I), LATLON = 1)
IF (DERIV1(I) .EQ. 4) ALON = LATLON
IF (NC .LT. 21) GOTO 25
WRITE (6, 103) DERIV1
IF (DERIV1(I) .EQ. 0) GOTO 60
NPLOT = NPLOT + 1
WRITE (5, 104) SMIN1, SMMAX1
WRITE (6, 105) SMNMX1
DO 30 J = 1, NC
30 CONTINUE
WRITE (5, 106) DERIV1
STOP
60 CONTINUE
70 CONTINUE
80 READ (1, 101) DERIV1, SMIN1, SMAX1
90 WRITE (1, 103) DERIV1, LATLON, 1H, NPLT
100 PARAM = PARAM
WPTPL = WPTPL
IF (PARAM .LE. 3) PARAM = 0,
DO 110 II = 1, NRP
IF (LONG1(II) .NE. LATLON) GOTO 110
II = II
IF (PARAM .LE. 3) PARAM = 3
110 CONTINUE
```

APPENDIX E – Continued

WRITE(3,2003)     INSTR 570
WRITE(5,2004)     INSTR 585
WRITE(6,2005)     INSTR 590
WRITE(7,2006)     INSTR 600
WRITE(8,2007)     INSTR 610
WRITE(9,2008)     INSTR 620
WRITE(10,2009)    INSTR 630
WRITE(11,2010)    INSTR 640
WRITE(12,2011)    INSTR 650

1001 FORMAT(A6,6F6.0,F10.0)     INSTR 570
2001 FORMAT(27H COEFFICIENTS TO BE PLOTTED/1X,2LAE)     INSTR 6 0
2002 FORMAT(27H HO4,46H IS NOT A VALID DERIVATIVE NAME FOR THIS PLOT)     INSTR 610
2003 FORMAT(37H WIND TUNNEL DATA AVAILABLE)     INSTR 620
2004 FORMAT(37H PLTS,5K,4HPARAMS.FIC.4,5K,10HTOLFRANC.FIC.4)     INSTR 630
126 RETURN
END
APPENDIX E — Continued

SUBROUTINE SUMPLT

Description: Subroutine SUMPLT plots data for one derivative.

Programming notes: Most of the data manipulation has been done, and the data are ready to plot. Thus this subroutine does little except the actual plotting.

Subroutine listing:
APPENDIX E – Continued

\[ XN = \text{ALFS}(I,J) - \text{AMIN} / \text{ASCAL} \]
\[ YN = \text{FOAT}(I,J) - \text{YMNI} / \text{YSCALE} \]
\[ \text{HITE} = \text{PMATC}(I,J) / \text{YSCALE} \]
\[ \text{XNP} = XN + 0.03 \]
\[ \text{XNM} = XN - 0.03 \]
\[ \text{CALL PLOT}(XNP, YNN, 2) \]
\[ \text{CALL PLOT}(XN, YNH, 2) \]
\[ \text{CALL PLOT}(XNM, YNH, 2) \]
\[ \text{CALL PLOT}(XNP, YNH, 2) \]

\[ \text{IF}(\text{WTP}) \text{ GO TO 90} \]
\[ \text{IF}(. \text{NOT.} \text{WTP}) \text{ GO TO 90} \]
\[ \text{IF}(NPARAM.GT.0 .AND. I.LEQ.1) \text{ GO TO 83} \]
\[ \text{IF}(NPARAM.GT.0 .OR. NCI.GT.0) \text{ GO TO 90} \]
\[ \text{WTO} = \text{KWT1} + 1 \]
\[ \text{YNN} = \text{YMN} / \text{YSCALE} \]
\[ \text{CALL LINES}(\text{ARP}, \text{WTD}(I,I), \text{KWT}, \text{KWT1}, \text{ISI}) \]
\[ \text{IF}(\text{YN}, \text{YLEN} .GT. 0) \text{ GO TO 100} \]
\[ \text{CALL PLOT}(\text{YN}, \text{YN}, 2) \]

\[ 100 \text{ CONTINUE} \]
\[ \text{RETURN} \]
\[ \text{END} \]
Description: Subroutine PSCALE selects flight data points to be plotted on the basis of the criteria specified in subroutine INSTR. It places flight data and predicted derivatives for a single derivative into arrays for plotting and determines ordinate scales if needed.

Programming notes: Flight data are moved from arrays FDATA and FDATAC to arrays FDAT, FDATC, and ALFS. Array FDAT contains the derivative values, FDATC the confidence levels, and ALFS the angles of attack. Predicted derivatives are selected from array DATA and moved to array WTD.

Subroutine listing:

```plaintext
SUBROUTINE PSCALE(NCMX,N02,FDATA,FDATAC,FDAT,FDATC,ALFS,DATA,NBP,NMBP,KAP)
DETERMINES PLOT SCALES, SELECTS DATA TO BE PLOTTED
DATA TO BE PLOTTED IS SELECTED FROM ARRAYS FDATA AND FDATAC
AND PLACED INTO THE SMALLER ARRAYS FDAT, FDATC, AND ALFS

COMMON /CASES/ NCASE
COMMON /SCAL/ CFAC,WT2,YLEN2,NPARAM
COMMON /SELECT/ PARAM,FJ,JJ,JOER,YMIN,YMAX,WTLAT,IOER,YMIN,YMAX,DEL
COMMON /DATA/ N02,FDATAC,FDAT,ALFS,FDATC,DATA,N02,NMBP,WTLAT,CRF,WTP,CERT,WT2,WT
COMMON /FIN/ N02,FDATAC,FDAT,ALFS,FDATC,DATA,N02,WTLAT,CRF,WTP,CERT,WT2,WT
COMMON /CASES/ NCASE
COMMON /SCALE/ CFAC,WT2,YLEN2,NPARAM
COMMON /SELECT/ PARAM,FJ,JJ,JOER,YMIN,YMAX,WTLAT,IOER,YMIN,YMAX,DEL
COMMON /DATA/ N02,FDATAC,FDAT,ALFS,FDATC,DATA,N02,NMBP,WTLAT,CRF,WTP,CERT,WT2,WT
COMMON /FIN/ N02,FDATAC,FDAT,ALFS,FDATC,DATA,N02,WTLAT,CRF,WTP,CERT,WT2,WT

REAL FDATA(NCMX,N02),FDATAC(NCMX,N02),FDAT(N02,1),FDATC(N02,1),ALFS(2,N02,1),DATA(N02,NMBP,1)
INTEGER JOER(11),N02(16),N02(32)
LOGICAL WTPL,WTP,NOPLOT

HTP=WTPL
LONLAT=WT2+WT
IOER=IOER(II)
OFRIV=OF(RIV)
ZSC(1)=O.
ZSC(2)=O.
CRF=CRFACT
IF(FOATAC(JOER,JJL,1).LT.O.,AND.,ABS{PARAM-FOATAC(22,JJL,1)|.GT.TOL} GO TO 50
IF(FDATA(JJ,1).FL.O.,GO TO 50
NCJ=NCJ+1
FDAT(JJ,1)=FDATAIJDER,JJL,1
FOATC(JJ,1)=FDATAC(JOER,JJL,1,CRF
ALFS(JJ,1)=DATA(JJL,1,WT2,WT)
ZSC(1)=AMIN1(ZSC(1),FOAT(JJ,1),NCJ=FDATAC(JJ,1,CRF
ZSC(2)=AMAX1(ZSC(1),FOAT(JJ,1),NCJ=FDATAC(JJ,1,CRF
NCJ=NCJ+1
NOPLOT=NOPLOT .AND. (NCJ.EQ.9)
CONTINUE

50 CONTINUE

60 IF(N02,GT.80) GO TO 80
IF(NPARAM,GT.80 .AND. JJ.EQ.11) GO TO 70
IF(NPARAM,GT.80 .AND. JJ.EQ.11) GO TO 70
NCJ=NCJ+1
K1=K1+1
K2=K2+1
K3=K3+1
J=J+1
GO TO 50

70 IF(JJ.EQ.1) NP1=WT1
K1=K1+1
K2=K2+1
K3=K3+1
J=J+1
GO TO 50
```

161
APPENDIX E - Concluded

SUBROUTINES WINDIN, LOAD1, SCALES, LINES, PLTDAT, TIME, AND DATE

Subroutines WINDIN, LOAD1, SCALES, LINES, PLTDAT, TIME, and DATE are identical to those in the SETUP and MMLE program.
This appendix presents a sample case for the SUMARY program.

### INPUT CARDS

```plaintext
SAMPLE CASE FOR SUMARY
\[WINO NCOL=6,LONG1=1,T,WARP=4,BODY=T,\_MAX=24,ASCALE=2,CRFACT=10,\_END\]

<table>
<thead>
<tr>
<th>INPUT CARD</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>_WINO</td>
<td>6</td>
</tr>
<tr>
<td>NCOL</td>
<td>6</td>
</tr>
<tr>
<td>LONG1</td>
<td>1</td>
</tr>
<tr>
<td>T</td>
<td></td>
</tr>
<tr>
<td>WARP</td>
<td>4</td>
</tr>
<tr>
<td>BODY</td>
<td>T</td>
</tr>
<tr>
<td>_MAX</td>
<td>24</td>
</tr>
<tr>
<td>ASCALE</td>
<td>2</td>
</tr>
<tr>
<td>CRFACT</td>
<td>10</td>
</tr>
</tbody>
</table>

| CMDES     | 1     |
| CMDE2     | 2     |
| CMDES     | 3     |
| CMDE      | 3     |
| CMDE      | 3     |
| CMDE      | 3     |
| CMDE      | 3     |
| CMDE      | 3     |
| CMDE      | 3     |

| CMG       | 2     |
| CMG       | 2     |
| CMG       | 2     |
| CMG       | 2     |

| \_END     |       |

| AINCRAFTR | 1     |
| AFT       | 1     |
| CASE1     | 1     |
|           | 0.000 |
|           | 4.863 |
|           | 1.000 |
|           | .260 |

| ACINCRAFTR| 1     |
| AFT       | 1     |
| CASE1     | 1     |
|           | 0.000 |
|           | 4.863 |
|           | 1.000 |
|           | .260 |

| BINCRAFTR | 1     |
| AFT       | 1     |
| CASE1     | 1     |
|           | 0.000 |
|           | 4.863 |
|           | 1.000 |
|           | .260 |

| BINCRAFTR | 1     |
| AFT       | 1     |
| CASE1     | 1     |
|           | 0.000 |
|           | 4.863 |
|           | 1.000 |
|           | .260 |

| \_INCRAFTR| 1     |
| AFT       | 1     |
| CASE1     | 1     |
|           | 0.000 |
|           | 4.863 |
|           | 1.000 |
|           | .260 |

| \_INCRAFTR| 1     |
| AFT       | 1     |
| CASE1     | 1     |
|           | 0.000 |
|           | 4.863 |
|           | 1.000 |
|           | .260 |

| ACINCRAFTR| 1     |
| AFT       | 1     |
| CASE1     | 1     |
|           | 0.000 |
|           | 4.863 |
|           | 1.000 |
|           | .260 |

| ACINCRAFTR| 1     |
| AFT       | 1     |
| CASE1     | 1     |
|           | 0.000 |
|           | 4.863 |
|           | 1.000 |
|           | .260 |
```
### APPENDIX F - Continued

<table>
<thead>
<tr>
<th>LONG AIRCRAFT B</th>
<th>FLT 1 CASE 11</th>
<th>0.000</th>
<th>16.199</th>
<th>1.000</th>
<th>.260</th>
</tr>
</thead>
<tbody>
<tr>
<td>A 3 4 1 6</td>
<td>-.059539</td>
<td>0.000000</td>
<td>0.000000</td>
<td>-.012968</td>
<td></td>
</tr>
<tr>
<td>B 3 6</td>
<td>-.013692</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td></td>
</tr>
<tr>
<td>AC</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td></td>
</tr>
<tr>
<td>BC</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LONG AIRCRAFT B</th>
<th>FLT 1 CASE 12</th>
<th>0.000</th>
<th>17.993</th>
<th>1.600</th>
<th>.260</th>
</tr>
</thead>
<tbody>
<tr>
<td>A 3 4</td>
<td>-.064619</td>
<td>0.000000</td>
<td>0.000000</td>
<td>-.015460</td>
<td></td>
</tr>
<tr>
<td>B 3 6</td>
<td>-.013692</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td></td>
</tr>
<tr>
<td>AC</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td></td>
</tr>
<tr>
<td>BC</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LONG AIRCRAFT B</th>
<th>FLT 1 CASE 15</th>
<th>0.000</th>
<th>20.088</th>
<th>1.000</th>
<th>.260</th>
</tr>
</thead>
<tbody>
<tr>
<td>A 3 4</td>
<td>-.016936</td>
<td>0.000000</td>
<td>0.000000</td>
<td>-.003470</td>
<td></td>
</tr>
<tr>
<td>B 3 6</td>
<td>-.036625</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td></td>
</tr>
<tr>
<td>AC</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td></td>
</tr>
<tr>
<td>BC</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LONG AIRCRAFT B</th>
<th>FLT 1 CASE 17</th>
<th>0.000</th>
<th>21.376</th>
<th>1.600</th>
<th>.260</th>
</tr>
</thead>
<tbody>
<tr>
<td>A 3 4</td>
<td>-.055533</td>
<td>0.000000</td>
<td>0.000000</td>
<td>-.028290</td>
<td></td>
</tr>
<tr>
<td>B 3 6</td>
<td>-.011031</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td></td>
</tr>
<tr>
<td>AC</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td></td>
</tr>
<tr>
<td>BC</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
<td></td>
</tr>
</tbody>
</table>

|                |               |       |        |       |      |
### APPENDIX F — Continued

<table>
<thead>
<tr>
<th>LONG</th>
<th>AIRCRAFT M</th>
<th>FLT</th>
<th>CASE 19</th>
<th>.000</th>
<th>22.641</th>
<th>1.000</th>
<th>.260</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.057218</td>
<td>.000000</td>
<td>.000000</td>
<td>.000000</td>
<td>-0.005720</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.011844</td>
<td>.160000</td>
<td>.000000</td>
<td>.000000</td>
<td>.000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.006115</td>
<td>.000000</td>
<td>.000000</td>
<td>.000000</td>
<td>.000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.012764</td>
<td>.000000</td>
<td>.000000</td>
<td>.000000</td>
<td>.000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.007367</td>
<td>.000000</td>
<td>.000000</td>
<td>.000000</td>
<td>.000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.001190</td>
<td>.000000</td>
<td>.000000</td>
<td>.000000</td>
<td>.000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AC</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.001052</td>
<td>.000000</td>
<td>.000000</td>
<td>.000000</td>
<td>.000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.000000</td>
<td>.161192</td>
<td>.000000</td>
<td>.000000</td>
<td>.000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.000000</td>
<td>.362000</td>
<td>.000000</td>
<td>.000000</td>
<td>.000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BC</td>
<td>3</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.000772</td>
<td>.000000</td>
<td>.000000</td>
<td>.000000</td>
<td>.000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.000118</td>
<td>.000000</td>
<td>.000000</td>
<td>.000000</td>
<td>.000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.000000</td>
<td>.500000</td>
<td>.000000</td>
<td>.000000</td>
<td>.000000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PLOT
LATR
CMB
LONG
CN
CNA
CMDE
CNDE
CMQ
DE
CMDC
## APPENDIX F – Continued

### OUTPUT LISTING

**MMLE SUMMARY PLOTTING PROGRAM**  
**1 JULY 1974**  
**VERSION 2**

**SAMPLE CASE FOR SUMMARY**

Confidence levels will be plotted multiplied by 10.0.

<table>
<thead>
<tr>
<th>Aircraft Type</th>
<th>Flight</th>
<th>Case</th>
<th>Confidence Level</th>
<th>Coefficients</th>
<th>CMDC Coefficients</th>
<th>Tolerance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Long Aircraft</td>
<td>FLT 1</td>
<td>CASE 1</td>
<td>0.0000</td>
<td>1.6000</td>
<td>1.0000</td>
<td>1.2610</td>
</tr>
<tr>
<td>Long Aircraft</td>
<td>FLT 1</td>
<td>CASE 4</td>
<td>0.0000</td>
<td>10.0310</td>
<td>1.0000</td>
<td>1.2610</td>
</tr>
<tr>
<td>Long Aircraft</td>
<td>FLT 1</td>
<td>CASE 8</td>
<td>0.0000</td>
<td>18.6710</td>
<td>1.0000</td>
<td>1.2610</td>
</tr>
<tr>
<td>Long Aircraft</td>
<td>FLT 1</td>
<td>CASE 11</td>
<td>0.0000</td>
<td>16.3950</td>
<td>1.0000</td>
<td>1.2610</td>
</tr>
<tr>
<td>Long Aircraft</td>
<td>FLT 1</td>
<td>CASE 12</td>
<td>0.0000</td>
<td>17.9930</td>
<td>1.0000</td>
<td>1.2610</td>
</tr>
<tr>
<td>Long Aircraft</td>
<td>FLT 1</td>
<td>CASE 15</td>
<td>0.0000</td>
<td>20.0880</td>
<td>1.0000</td>
<td>1.2610</td>
</tr>
<tr>
<td>Long Aircraft</td>
<td>FLT 1</td>
<td>CASE 17</td>
<td>0.0000</td>
<td>21.1760</td>
<td>1.0000</td>
<td>1.2610</td>
</tr>
<tr>
<td>Long Aircraft</td>
<td>FLT 1</td>
<td>CASE 19</td>
<td>0.0000</td>
<td>22.6410</td>
<td>1.0000</td>
<td>1.2610</td>
</tr>
</tbody>
</table>

**LAST PLOTS**

**PARAM = 0.0000**  
**TOLERANCE = -0.0000**

No wind tunnel data available for CNB.

No flight data available for CMOC.

Coefficients to be plotted.
A sample plot from the SUMARY program is shown. The plot is presented in four parts to avoid loss of detail from a large reduction. The plot as produced by the automatic plotter is shown within the heavy lines. Explanatory material is included to aid the user in implementing the program. Solid lines denote predicted derivatives. Vertical bars (\(\text{I}\)) indicate confidence levels.
SAMPLE CASE FOR SUMMARY

$C_{m_{\delta e}}$

$C_{N_{\delta e}}$

$\alpha$

SYMB MACH
$\bigcirc$ 0.50
APPENDIX F - Continued

SAMPLE CASE FOR SUMMARY

$C_m\alpha$

$C_m\eta$

$\alpha$
$\delta_{e_{trim}}$ vs $\alpha$

Sample case for summary
REFERENCES


