NEW LIMITATION CHANGE

TO
Approved for public release, distribution unlimited

FROM
Distribution authorized to U.S. Gov't. agencies and their contractors; Critical Technology; NOV 1969. Other requests shall be referred to Naval Training Device Center, Orlando, FL.

AUTHORITY
USNTDC ltr, 9 Dec 1971
Best Available Copy
ADVANCED SUBMARINE SYSTEMS PROGRAMMING

ABSTRACT

This programing report is the result of a study leading to the determination of the optimum sets of equations of motion to be used with two general types of submarine control trainers. The starting point was the Naval Ship Research and Development Center standard equations of motion for submarine simulators.

Two complete submarine simulation programs using these equations are given; one for six-degrees-of-freedom and one for the longitudinal three-degrees-of-freedom. A number of programs are included to assist the researcher in interpreting the results of the main simulation program. A compact submarine simulation program for use with a small computer is given and a method of generating random ocean wave amplitudes is outlined along with its program.

This report describes the programs, including listing in FORTRAN, flow charts, input decks, and typical output sheets, but does not tell how they are to be used. It should be used in conjunction with Advanced Submarine Systems Equation Study, NAVTRADEVCOM 68-C-0050-1 which describes the work performed under this study.
NAVTRADEVCEN 68-C-0050-2

FOREWORD

This report presents computer programs which allow various investigations of submarine simulation. Descriptions, flow charts, and listings are presented for each program. Important uses of the programs include coefficient reduction of any class of submarine, checking accuracy of coefficients when operational data is available, and research in casualty situations.

NAVTRADEVCEN 68-C-0050-1 gives an overall description of the equations study. NAVTRADEVCEN 68-C-0050-3 presents results of the computer programs using the SS(N)594 submarine as the demonstration model.

Charles A. Rumbough
CHARLES A. RUMBROUGH
Project Engineer
TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I.</td>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>II.</td>
<td>PROGRAM DESCRIPTIONS</td>
<td>2</td>
</tr>
<tr>
<td>EB920, Submarine Simulation Program</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>ZC790, Submarine Simulation Program, Longitudinal Freedom</td>
<td>39</td>
<td></td>
</tr>
<tr>
<td>ECL70, Initial Condition Computation for Simulation</td>
<td>56</td>
<td></td>
</tr>
<tr>
<td>ECL30, Center of Gravity Computation</td>
<td>61</td>
<td></td>
</tr>
<tr>
<td>ZC300, Submarine Thrust.</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>ZC690, Error Calculator, DS and DR Control</td>
<td>68</td>
<td></td>
</tr>
<tr>
<td>ZC691, Error Calculator, DS Control</td>
<td>71</td>
<td></td>
</tr>
<tr>
<td>ECL40, Root Cracker, Longitudinal</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>EC320, Root Cracker, Lateral</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>EC150, Coefficient Estimator, Lateral</td>
<td>87</td>
<td></td>
</tr>
<tr>
<td>EC330, Coefficient Estimator, Longitudinal</td>
<td>93</td>
<td></td>
</tr>
<tr>
<td>EC310, Brown's Convergence and Comparative Plot Program</td>
<td>99</td>
<td></td>
</tr>
<tr>
<td>EC790, Calculation of Compact Coefficients</td>
<td>107</td>
<td></td>
</tr>
<tr>
<td>EC780, Compact Submarine Simulation Program</td>
<td>114</td>
<td></td>
</tr>
<tr>
<td>EC572, Wave Generator Program</td>
<td>120</td>
<td></td>
</tr>
<tr>
<td>REFERENCES</td>
<td>125</td>
<td></td>
</tr>
<tr>
<td>APPENDIX A.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Program listings and Flow Charts</td>
<td>126</td>
<td></td>
</tr>
<tr>
<td>APPENDIX B.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integration Techniques</td>
<td>362</td>
<td></td>
</tr>
<tr>
<td>GLOSSARY</td>
<td>364</td>
<td></td>
</tr>
<tr>
<td>Table</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>1</td>
<td>Rudder Position Schedule</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>Input Data Deck Program EB920</td>
<td>18</td>
</tr>
<tr>
<td>3</td>
<td>EB920, Common Layout Index</td>
<td>30</td>
</tr>
<tr>
<td>4</td>
<td>EB920, Common Layout, ALPHABETICAL</td>
<td>32</td>
</tr>
<tr>
<td>5</td>
<td>Output Variables, Program EB920</td>
<td>34</td>
</tr>
<tr>
<td>6</td>
<td>Input Data Deck, Program ZC790</td>
<td>45</td>
</tr>
<tr>
<td>7</td>
<td>Output Variables, Program ZC790</td>
<td>51</td>
</tr>
<tr>
<td>8</td>
<td>ZC790, Common Layout</td>
<td>52</td>
</tr>
<tr>
<td>9</td>
<td>Input Data Deck, Program EC470</td>
<td>57</td>
</tr>
<tr>
<td>10</td>
<td>Input Data Deck, Program EC430</td>
<td>62</td>
</tr>
<tr>
<td>11</td>
<td>Input Data Deck, Program ZC300</td>
<td>67</td>
</tr>
<tr>
<td>12</td>
<td>Input Data Deck, Program ZC690</td>
<td>68</td>
</tr>
<tr>
<td>13</td>
<td>Input Data Deck, Program ZC691</td>
<td>71</td>
</tr>
<tr>
<td>14</td>
<td>Data Submittal for ISWZ = 0, ZC691</td>
<td>72</td>
</tr>
<tr>
<td>15</td>
<td>Data Submittal for ISWZ = 1, ZC691</td>
<td>72</td>
</tr>
<tr>
<td>16</td>
<td>Input Data Deck, Program EC440</td>
<td>77</td>
</tr>
<tr>
<td>17</td>
<td>Input Data Deck, Program EC320</td>
<td>84</td>
</tr>
<tr>
<td>18</td>
<td>Input Data Deck, Program EC570</td>
<td>90</td>
</tr>
<tr>
<td>19</td>
<td>Input Data Deck, Program EC572</td>
<td>97</td>
</tr>
<tr>
<td>20</td>
<td>Input Data Deck, Program EC310</td>
<td>104</td>
</tr>
<tr>
<td>21</td>
<td>Output Deck Format, EC790</td>
<td>113</td>
</tr>
<tr>
<td>22</td>
<td>Input Data Deck, Program EC780</td>
<td>117</td>
</tr>
<tr>
<td>23</td>
<td>Input Data Deck, Program EC572</td>
<td>122</td>
</tr>
<tr>
<td>24</td>
<td>Program Listings Page Number</td>
<td>126</td>
</tr>
<tr>
<td>25</td>
<td>Popular Numerical Integration Techniques</td>
<td>363</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Figure</th>
<th>Illustration Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Research Simulation Program Block Diagram</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Subroutine CONTR, EB920, Block Diagram</td>
<td>13</td>
</tr>
<tr>
<td>3</td>
<td>Graph of Overshoot Control</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>Input Data Form, Program EB920.</td>
<td>28</td>
</tr>
<tr>
<td>5</td>
<td>Input Variable Format, EB920.</td>
<td>35</td>
</tr>
<tr>
<td>6</td>
<td>Output Variable Format, EB920.</td>
<td>36</td>
</tr>
<tr>
<td>7</td>
<td>Graphical Output, EB920</td>
<td>38</td>
</tr>
<tr>
<td>8</td>
<td>ZC790 Submarine Simulation Program Block Diagram</td>
<td>40</td>
</tr>
<tr>
<td>9</td>
<td>Subroutine CONTR, ZC790, Block Diagram</td>
<td>45</td>
</tr>
<tr>
<td>10</td>
<td>Input Variable Format, ZC790.</td>
<td>53</td>
</tr>
<tr>
<td>11</td>
<td>Output Variable Format, ZC790.</td>
<td>54</td>
</tr>
<tr>
<td>12</td>
<td>Angle of Attack</td>
<td>56</td>
</tr>
<tr>
<td>13</td>
<td>Output Variable Format, ECL470.</td>
<td>59</td>
</tr>
<tr>
<td>14</td>
<td>Output Variable Format, ECL430</td>
<td>64</td>
</tr>
<tr>
<td>15</td>
<td>Comparison Criteria for Error Calculator</td>
<td>69</td>
</tr>
<tr>
<td>16</td>
<td>Output Variable Format, ECL440</td>
<td>79</td>
</tr>
<tr>
<td>17</td>
<td>Error in Use of $F$</td>
<td>82</td>
</tr>
<tr>
<td>18</td>
<td>Output Variable Format, ECL420</td>
<td>86</td>
</tr>
<tr>
<td>19</td>
<td>Submarine Response Curve, Lateral Case</td>
<td>88</td>
</tr>
<tr>
<td>20</td>
<td>Submarine Response Curve, Longitudinal Case</td>
<td>94</td>
</tr>
<tr>
<td>21</td>
<td>Typical Converged Response Plot</td>
<td>100</td>
</tr>
<tr>
<td>22</td>
<td>Output Data Format, ECL780</td>
<td>119</td>
</tr>
<tr>
<td>23</td>
<td>Typical Wave Generator Output</td>
<td>123</td>
</tr>
</tbody>
</table>
SECTION I

INTRODUCTION

This report is the result of a study leading to the determination of the optimum sets of equations of motion, to be used with two general types of submarine control trainers.

Two complete submarine simulation programs are given; one for six-degrees-of-freedom and one for three-degrees-of-freedom. A number of programs are included to assist the researcher in interpreting the results of the main simulation program. A compact program for use with a small computer is also included.

This report describes the programs, including listing in FORTRAN, but does not tell how they are to be used. It should be used in conjunction with Advanced Submarine System Equation Study, NAVTRADEVCEN 68-C-0050-1 which describes the work performed under this study.
SECTION II
PROGRAM DESCRIPTIONS

A. PROGRAM EB920, SUBMARINE SIMULATION

1. DESCRIPTION

This program calculates the dynamic changes of a body's position and attitude as a function of time. The vehicle is a submarine in this case but any vehicle can be simulated if the coefficients of the equations of motion are known. The equations used for the mathematical model are developed in "Standard Equations of Motion for Submarine Simulation", Report 2510 by Morton Gertler and Grant R. Hagen of the Naval Ship Research and Development Center in Washington, D.C. They are designated as the NSRDC Standard Equations and the terminology in this program follows this report.

The NSRDC Standard Equations cover all phases of submarine motion simulation in six-degrees-of-freedom including emergency recoveries after casualties. NSRDC Report 2510 contains a brief history, defines the mathematical model, discusses the coefficients required, and sets a standard to be used in the simulation of submarines. Equations (1) through (6) present the equations of motion in the following order: axial force, lateral force, normal force, rolling moment, pitching moment, and yawing moment. In addition certain kinematic relations are given in equation (7).

The program is written in basic FORTRAN IV for use on any digital computer with a FORTRAN compiler. It occupied 20K words when run on an IBM 360/40 computer. Program results are printed as a function of time and individual parameters can be plotted if associated CALCOMP plotter software is available.

Figure 1 is a block diagram showing the general outline of the program and the subroutines used. A number of different options are available to the program user. They are listed below.

a. Programmed control surface and thrust values.

1. Climbing turns - fixed elevator and rudder deflection; for climb or turn (without autopilot):
   2. Meander or overshoot
   3. Modified climbing turns - surfaces deflected at controlled rates to specified values.

b. Flat turn (with autopilot)

• Superscript numbers indicate references
Figure 1. Research Simulation Program Block Diagram
AXIAL FORCE

\[ m \left[ \ddot{u} - vy - wq - x_G \left(q^2 + r^2\right) + y_G \left(pq - r\right) + z_G \left(pr + \dot{q}\right) \right] = \]

\[ + \frac{p}{2} \left[ \varepsilon \left( x_{q q} \dot{q}^2 + x_{r r} \dot{r}^2 + x_{p p} \dot{p}^2 \right) \right] \]

\[ + \frac{p}{2} \left[ \varepsilon \left( x_{u u} \ddot{u} + x_{v v} \ddot{v} + x_{w w} \ddot{w} \right) \right] \]

\[ + \frac{p}{2} \left[ \varepsilon \left( x_{t t} \dot{t} \ddot{t} + x_{s s} \dot{s} \ddot{s} + x_{b b} \dot{b} \ddot{b} \right) \right] \]

\[ + \frac{1}{3} \rho \varepsilon \left[ a_1 \dddot{u} + b_1 \dddot{u} + c_1 \dddot{u} \right] \]

\[ - (W - B) \sin \theta \]

\[ + \frac{p}{2} \left[ \varepsilon \left( x_{v v} \dot{v}^2 + x_{w w} \dot{w}^2 + x_{t t} \dot{t}^2 + x_{s s} \dot{s}^2 + x_{b b} \dot{b}^2 \right) \right] \]

\[ + \frac{1}{3} \rho \varepsilon \left[ a_1 \dddot{u} + b_1 \dddot{u} + c_1 \dddot{u} \right] \left( \eta - 1 \right) \]
LATERAL FORCE

\[
m \left[ \dot{v} - wp + ur - y_G (r^2 + p^2) + z_G (qr - \dot{p}) + x_G (qp + \dot{r}) \right] = \\
+ \frac{\rho}{2} t^4 \left[ Y_{r'} \dot{r} + Y_p' \dot{p} + Y_{p|p|} p |p| + Y_{pq} pq + Y_{qr} qr \right] \\
+ \frac{\rho}{2} t^3 \left[ Y_{v'} \dot{v} + Y_{vq} vq + Y_{wp} wp + Y_{wr} wr \right] \\
+ \frac{\rho}{2} t^3 \left[ Y_r' ur + Y_p' up + Y_{|r|\delta r'} u |r| \delta r + Y_v |r| \dot{r} + Y_{v|r|} \frac{v}{|v|} |(v^2 + w^2)^\frac{3}{2} \mid r \mid \right] \\
+ \frac{\rho}{2} t^3 \left[ Y_*' u^2 + Y_{v'} uv + Y_{v|v|} v |(v^2 + w^2)^\frac{3}{2} \mid \right] \\
+ \frac{\rho}{2} t^2 \left[ Y_{vw} vw + Y_{\delta r'} u^2 \delta r \right] \\
+ (W - B) \cos \theta \sin \phi \\
+ \frac{\rho}{2} t^3 Y_{r\eta'} ur (\eta - 1) \\
+ \frac{\rho}{2} t^2 \left[ Y_{v\eta'} uv + Y_{v|v|\eta} v |(v^2 + w^2)^\frac{3}{2} \mid + Y_{\delta r\eta} \delta r u^2 \right] (\eta - 1)
\]
NORMAL FORCE

\[ m \left[ \dot{w} - uq + v_p - z_{\alpha} (p^2 + q^2) + x_G (r_p - \dot{q}) + y_G (r_q + \dot{p}) \right] = \]

\[ + \frac{p}{2} \left[ Z_{\dot{q}} q + Z_{pp} p^2 + Z_{rr} r^2 + Z_{rp} r_p \right] \]

\[ + \frac{p}{2} \left[ Z_{\dot{w}} \dot{w} + Z_{vr} v_r + Z_{vp} v_p \right] \]

\[ + \frac{p}{2} \left[ Z_{\dot{q}} \dot{q} + Z_{\dot{q}q} u |q| \delta_s + Z_{ww} w |w| (v^2 + w^2)^{\frac{1}{2}} |q| \right] \]

\[ + \frac{p}{2} \left[ Z_{ww} w |w| + Z_{ww'} w |w (v^2 + w^2)^{\frac{1}{2}} | \right] \]

\[ + \frac{p}{2} \left[ Z_{vv} v^2 + Z_{ss} u^2 \delta_s + Z_{bb} u^2 \delta_b \right] \]

\[ + (W - B) \cos \theta \cos \phi \]

\[ + \frac{p}{2} \left[ Z_{qq} uq (\eta - 1) \right] \]

\[ + \frac{p}{2} \left[ Z_{ww} w |w| + Z_{ww} |w| \delta_s u^2 (\eta - 1) \right] \]
ROLLING MOMENT

\[ I_x \ddot{p} + (I_z - I_y) qr - (\dot{r} + pq) I_{xz} + (r^2 - q^2) I_{yz} + (pr - q) I_{xy} \]

\[ + m \left[ y_G (\dot{w} - uq + vp) - z_G (\dot{v} - wp + ur) \right] = \]

\[ + \frac{g}{2} \ell^5 \left[ K_p \dddot{p} + K_r \dddot{r} + K_{qr} qr + K_{pq} pq + K_{\phi_1} \dddot{\phi}_1 \right] \]

\[ + \frac{g}{2} \ell^4 \left[ K_p \dddot{u} + K_r \dddot{r} + K_v \dddot{v} \right] \]

\[ + \frac{\rho}{2} \ell^4 \left[ K_{vq} \dddot{v} + K_{wp} \dddot{w} + K_{wr} \dddot{w} \right] \]

\[ + \frac{\rho}{2} \ell^3 \left[ K_{uv} u^2 + K_v \dddot{v} + K_{\phi_1} \dddot{\phi}_1 \right] \]

\[ + \frac{\rho}{2} \ell^3 \left[ K_{vw} w^2 + K_r \dddot{r} \right] \]

\[ + (y_G W - y_B B) \cos \theta \cos \phi - (z_G B - z_B B) \cos \theta \sin \phi \]

\[ + \frac{\rho}{2} \ell^3 K_{\phi_1} \dddot{\phi}_1 u^2 (\eta - 1) \]
PITCHING MOMENT

\[ I_y \ddot{q} + (I_x - I_z) \dot{p} - (\dot{p} +qr) I_{xy} + (p^2 - r^2) I_{zx} + (qp - \dot{r}) I_{yz} \]

\[ + m \left( z_G (\ddot{u} - vr + wq) - x_G (\ddot{w} - uq + vp) \right) = \]

\[ + \frac{p}{2} \ell^6 \left[ M_q \dot{q} + M_{pp} p^2 + M_{rr} r^2 + M_{rp} r p + M_q |q|' q |q| \right] \]

\[ + \frac{p}{2} \ell^6 \left[ M_w \dot{w} + M_{vr} v r + M_{vp} v p \right] \]

\[ + \frac{p}{2} \ell^6 \left[ M_q u q + M_{q|q|'} u |q|' q |q| + M_{w|w|'} w |w|' (v^2 + w^2)^{\frac{1}{2}} |q| \right] \]

\[ + \frac{p}{2} \ell^6 \left[ M_w u^2 + M_w u w + M_w |w|' w |w| (v^2 + w^2)^{\frac{1}{2}} \right] \]

\[ + \frac{p}{2} \ell^6 \left[ M_{ww} u^2 + M_{ww} u w + M_{ww} |w|' w |w| (v^2 + w^2)^{\frac{1}{2}} \right] \]

\[ - (x_G W - x_B B) \cos \theta \cos \phi - (z_G W - z_B B) \sin \theta \]

\[ + \frac{p}{2} \ell^6 M_{q|q|'} u |q| (\eta - 1) \]

\[ + \frac{p}{2} \ell^6 \left[ M_{w|w|'} u w + M_{w|w|'} w |w| (v^2 + w^2)^{\frac{1}{2}} + M_{s|s|'} s u^2 \right] (\eta - 1) \]

8
YAWING MOMENT

\[ I_z \ddot{r} + (I_y - I_x) pq - (q + rp) I_{yz} + (q^2 - p^2) I_{xy} + (rq - p) I_{zx} \]

\[ + m \left[ x_G (\ddot{v} - wp + ur) - y_G (u - vr + wq) \right] = \]

\[ + \frac{\rho}{2} L^5 \left[ N_r \dot{t} + N_p \dot{p} + p q + N_q \dot{q} + N_r \dot{r} + N \right] \]

\[ + \frac{\rho}{2} L^5 \left[ N_v \dot{v} + N \dot{w} + w p + N v q \right] \]

\[ + \frac{\rho}{2} L^5 \left[ N_p \dot{u} + N_r \dot{u} + N v \dot{v} + u v \right] \]

\[ + \frac{\rho}{2} L^5 \left[ N_v w + N_r \dot{v} \right] \]

\[ + (x_G W - x_B B) \cos \theta \sin \phi + (y_G W - y_B B) \sin \theta \]

\[ + \frac{\rho}{2} L^5 N_r \eta' \dot{r} \eta (\eta - 1) \]

\[ + \frac{\rho}{2} L^5 \left[ N v \eta' u v + N v \eta' v \right] \]

\[ \times \]

\[ \times \]
KINEMATIC RELATIONS

\[ u^2 = u^2 + n^2 + \omega^2 \]

\[ \dot{\theta} = q \cos \phi - r \sin \phi \]

\[ \dot{\psi} = \frac{q \sin \phi + r \cos \phi}{\cos \theta} \]

\[ \dot{\phi} = p + \phi \sin \phi \]

\[ \dot{\rho}_o = u \cos \theta \cos \psi + n (\sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi) \]

\[ + \omega (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi) \]

\[ \dot{\psi}_o = u \cos \theta \sin \psi + n (\sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi) \]

\[ + \omega (\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi) \]

\[ \dot{\phi}_o = -u \sin \theta + n \cos \theta \sin \phi + \omega \cos \theta \cos \phi \]
6. Turning impulse (with autopilot)
7. Acceleration/deceleration (with autopilot)
8. Maximum acceleration/deceleration (with autopilot)

b. Integration Methods
1. Fourth Order Runge-Kutta
2. Second order Adams
3. Second order non-classical
4. Euler

c. Variable integration step size (H, or integration time increment). This option allows study of the allowable or optimum integration step size, H, as a function of output variable accuracy.

d. Initial Conditions

The program will accept values of initial conditions for control surface positions, speed, attitude and depth. This allows the submarine to be placed in a steady state dive, level flight, etc. at time zero or start of the computer run. These inputs are:

UC - command speed
DR - rudder position
DS - sternplane position
DB - sailplane position
W - component of velocity in the z-direction
Q - angular acceleration component about the y-axis
THETA - angle of pitch
Z - depth

All other parameters of motion, v, p, r, ψ, β, x, and y, can also be set as initial conditions. These parameters are usually zero at time zero. The values for steady-state level flight can be calculated with program EQU70.

e. Output Options

Any time interval can be set between printing out the parameters of motion. If a CALCOMP plotter is available, any parameter and the control surfaces can be plotted as a function of time in any order.

2. SUBROUTINE DESCRIPTIONS

The subroutines utilized by this program are shown in block diagram
form in figure 1. A brief description of each subroutine is included for clarification of operation of the total program.

a. INPUT - This subroutine reads a data deck specifying program options, controls and initial dynamic conditions, and the particular coefficients and constants for the submarine to be simulated. It provides for holding initial values of all variables for use on a subsequent run. It provides for reading additional cards for successive runs, and finally for exit from the program on a blank card.

b. KUTTA - This is an integration subroutine that uses the Runge-Kutta, 4th order integration method to integrate the equations of motion over the time period required in accordance with the equation:

\[ Y_{n+1}(I) = Y_n(I) + \frac{1}{6} (K_0(I) + 2K_1(I) + 2K_2(I) + K_3(I)) \]

where

\[ K_0(I) = h \cdot EVAL(Y_n(I)) \]
\[ K_1(I) = h \cdot EVAL(Y_n(I) + \frac{1}{2} K_0(I)) \]
\[ K_2(I) = h \cdot EVAL(Y_n(I) + \frac{1}{2} K_1(I)) \]
\[ K_3(I) = h \cdot EVAL(Y_n(I) + K_2(I)) \]

\( I = 1 \) to 12
\( h = \) integration time interval
\( Y_n(I) = \) motion parameter \( u, v, \) etc. at the \( n \)th cycle
\( EVAL = \) equation of motion subroutine

This subroutine takes a twelve-element matrix from subroutine EVAL corresponding to accelerations and velocities; integrates over the interval, \( h \); and returns twelve new velocities or positions. (Corresponding to \( u, v, w, p, q, r, \theta, \psi, \phi, x, y, z \). \( I \) runs from 1 to 12 to cover all the terms to be integrated. The EVAL subroutine is entered four times in order to calculate the four \( K \)'s since each one is dependent on the last one. The integration method is started by zeroing the working storage during the first pass through the subroutine.


c. CONTR - This subroutine allows various submarine control maneuvers to be selected that are used in submarine research studies. The subroutine varies the control surfaces during the run to allow meander, submerged turns, overshoot, acceleration, etc. to be made. The block diagram of figure 2 outlines the control subroutine. In all controls, the variable
SUBROUTINE CONTR

GO TO(1001, 1001, 1003, 1004, 1005, 1006, 1007), N8

**Figure 2. Subroutine CONTR, EB920, Block Diagram**
TLIM defines the total time period of the runs in seconds.

Constant controls (NS = 0) - The submarine will maintain the settings of command speed, (UC), elevator positions (DB and DS), or rudder position (DR) that were entered in the program as initial conditions. Thus, runs such as a steady turn without autopilot, climbing turn, acceleration without autopilot, etc., can be made.

Meander or Overshoot (NS = 1) - This control allows the submarine to maintain a period of level flight, followed by negative elevator movement at a specified rate to a minimum (largest negative value) elevator position as shown in figure 3. It will hold this angle until the submarine pitch angle reaches a particular value, SWMAX, at which time the elevators are reversed in position at a specified rate to a desired new position. The value of this position is determined by the type of run desired; for meanders, DELTMI = DS and for overshoot, DELTMA = DS equals DELTMI - DS in magnitude. The program allows variation of the various control parameters of this maneuver. These are:

TIME - period of initial steady state performance, or constant input terms.
R1 - negative rate of change of elevator position.
DELTME - maximum negative swing of elevator (must be more negative than DS, elevator position at zero time).
SWMAX - maximum dive angle, execute pitch angle allowed prior to turning elevators more positive. The time at which this occurs is referred to as T1.
R2 - positive rate of change of elevator position.
DELTMI - maximum position to which elevators are moved in the positive direction.

These input control parameters can be varied to achieve Meander, Overshoot, or other desired combinations of a dive to a maximum submarine pitch angle and subsequent elevator change.

Flat turn (NS = 3) - This is a turn accomplished by rudder movement in four steps to the maximum deflection, DRMAX. Each step is accomplished at a decreasing rate of change of rudder position as a function of the final maximum deflection, as shown in table 1.

<table>
<thead>
<tr>
<th>Period</th>
<th>Rudder Position Rate of Change</th>
<th>Maximum Rudder Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.08726</td>
<td>0.85</td>
</tr>
<tr>
<td>2</td>
<td>0.01336</td>
<td>0.93</td>
</tr>
<tr>
<td>3</td>
<td>0.006</td>
<td>0.97</td>
</tr>
<tr>
<td>4</td>
<td>0.0001064</td>
<td>1.0</td>
</tr>
</tbody>
</table>
The program allows selection of the final rudder position, DRMAX. During this maneuver, the autopilot is actuated to maintain nearly level flight.

The autopilot is utilized in the control subroutine for certain maneuvers, such as flat turns, acceleration, etc. This control is automatic when used through selection of the desired control routine. The bow and stern planes are moved to attempt to maintain constant depth at any speed or turn condition. The elevators are positioned in accordance with equation (8).

\[
DS = DB \times 0.006 (ZC-Z) + 3.5 (\theta) + 0.012 (U\text{Sin}e-WG050) + 2.0q \quad (8)
\]

Where:

- DB = sailplane angle, radians
- DS = sternplane angle, radians
- ZC = command depth, feet
- Z = depth, feet
- \theta = pitch angle, radians
- u = forward body velocity, ft/sec
- w = normal body velocity, ft/sec
- q = pitch rate, radians/sec
Position and rate damping in both depth and pitch are utilized. DS and DB are limited to 35 degrees by the programming.

Climbing turn (NS = 2) - This is a complex maneuver utilizing the available inputs of meander and flat turn above, with the autopilot inoperative. All the input data of each of these two other controls is required and the resultant submarine maneuver can obviously be rather unique.

DS - impulse (NS = 4) - This control allows an elevator impulse to be applied to the submarine for the first integration cycle in order to evaluate response frequency and damping factors. The initial conditions (UC, w, THETA, and z) are entered as required. These are usually for level flight at some particular speed. The impulse value, elevator position (DS), is also entered. The final value of elevator position, DSF, is specified. This value is the elevator position for level flight at the selected speed.

A useful addition of the program for elevator-impulse control is the computer output of punched cards of the submarine pitch angle versus time. These are punched in 2EL5,7 format (THETA, TIME) from T0 to TLIM. The cards are useful in other programs associated with submarine response, described elsewhere in this report. Most associated programs are based on cards punched at two-second intervals. To achieve this the input value of integration interval, H, must be 0.25 seconds. This punch card data will be received for any of the integration methods selected through the option, INTSW. It is suggested that INTSW be set to 'O' for impulse runs because the accuracy of other integration methods does not match the accuracy of the Runge Kutta method when calculating the violent initial maneuvers excited by an impulse run.

The value of DSF, after the impulse, is usually greater than zero. This is the elevator position required to maintain level flight at the steady-state speed of the submarine.

DR - impulse (NS = 5) - This control is identical to DS - impulse except that rudder position is used for this type run. In addition the final rudder position (DRF) is usually zero rather than a small finite value as for elevator impulse.

Acceleration/Deceleration Control (NS = 6) - This option takes the submarine at rest, at any depth; accelerates at command speed increments of five knots over each time increment, TIME (an input value) to 25 knots; and then decelerates in the same command speed increments. The autopilot control is activated in this run to maintain the submarine in nearly level flight.

Maximum Acceleration/Deceleration Control (NS = 7) - This control accelerates a submarine at rest, at input depth, z, by applying a command speed of 25 knots. This command speed is held for TIME seconds (input value) which may be varied to assure steady state conditions. The submarine is then slowed down by reduction of command speed (UC) to zero for
the same length of time. The autopilot control is used for level flight.

d. EVAL - This subroutine calculates the right hand side of the
equations of motion. The values are updated after each pass through the
integration routine; Kutta. It follows the mathematical model given in
Table 1.

e. PLOTROU - This subroutine transfers the run number, data names,
and calculated points for storage on magnetic tape and subsequent plot
of the variables on a California Computer plotter. This subroutine, in
addition, calls out the following subroutines that are peculiar to Cal
Comp software:

LINE PLOTS PLOT WHERE
AXIS NUMBER SYMBOL SCALE

These are all Cal Comp proprietary subroutines and thus cannot be
supplied with this contract. If this program is run on a computer with
the source decks supplied by GAC and Cal Comp plotting software is avail-
able minor program adjustments may be necessary to allow exact plotting
as at GAC.

If this software is not available, three options are open:

(1) At GAC, the 360/40 computer will operate without the subroutines
above, provided no plots are called for by leaving variable IPLIT = 0 on
the first input data card. The linkage editor map at GAC shows "Unresolved
entry message". However, the program still runs without plot. The program
may run similarly on other computers.

(2) If this does not function on the computer used, dummy subroutines
for these variables can be added to the programs. The program can then
call them as at present and return. Leave IPLIT = 0.

(3) Finally, the reference to all plotting subroutines could be
removed from the program.

f. INVER2 - This is a common library routine to invert a matrix.

g. MATMPY - This subroutine takes the inverted matrix from INVER 2
and multiplies to calculate values of \( \hat{\alpha} \), \( \hat{\beta} \), \( \hat{\delta} \), \( \hat{\phi} \), and \( \hat{\theta} \).

h. INTEG - This routine includes the three additional optional
integration methods. These include:

(1) INTSW = 1, Second order Adams

\[ Y_{n+1} = Y_n + \frac{1}{2} \left( 3 Y_n - Y_{n-1} \right) \]
(2) INTSW = 2, Second order non-classical
\[ Y_{n+1} = Y_n + \frac{h}{4} \left( 3\dot{Y}_n + \dot{Y}_{n-1} \right) \]

(3) INTSW = 3, Euler
\[ Y_{n+1} = Y_n + h \dot{Y}_n \]

where \( Y \) is the variable to be integrated

\( n \) is the number of times the variable was integrated

\( h \) is the integration time period.

1. EVAL 1 - This subroutine solves the right hand side of the equations of motion from data from each pass thru the integration routine, INTEG. It follows the mathematical model given in table 1.

3. INPUT DATA DECK

This section describes the data deck, defines the FORTRAN variables in terms of the equations of motion defined in the mathematical model, and shows how to perform the various kinds of simulated submarine operations allowed. The coefficients referred to here are those included in reference, "Standard Equations of Motion for Submarine Simulation". The coefficients are unique for each type submarine. These values are program inputs so that the program can be used for the study of different types of submarines. The input deck should follow table 2 exactly. Figure 4 shows a typical input data form for EB920.

### TABLE 2. INPUT DATA DECK, PROGRAM EB920

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control Flags</td>
<td>1</td>
<td>1-5</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>6-10</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>11-15</td>
<td>15</td>
</tr>
</tbody>
</table>

18
### TABLE 2. INPUT DATA DECK, PROGRAM EB920 (cont.)

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16-20</td>
<td>15</td>
<td>IRUN, Identification number for individual runs. If IRUN = 0 a normal exit is made</td>
</tr>
<tr>
<td>1</td>
<td>21-25</td>
<td>15</td>
<td>NPLT. Data will plot at T&lt;sub&gt;0&lt;/sub&gt; and each NPLT&lt;sup&gt;th&lt;/sup&gt; integration step.</td>
</tr>
<tr>
<td>1</td>
<td>26-30</td>
<td>15</td>
<td>IOPT. This option will allow changing any one or more of the input values for a succeeding run without putting in all the other values. IOPT = 1 to exercise; blank for no selection. Another run may still follow, for IOPT = 0, but all input data cards must be read again. More information on this variable is included at card 33.</td>
</tr>
<tr>
<td>1</td>
<td>31-35</td>
<td>15</td>
<td>ICYC. Number of integration cycles per H, time increment. ICYC = 4 for Kutta integration (INTSW = 0), = 1 for all other integration methods.</td>
</tr>
<tr>
<td>1</td>
<td>36-40</td>
<td>15</td>
<td>NS. This variable selects the type of submarine control in CONTR subroutine:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NS = 0 Fixed controls per initial conditions</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NS = 1 Overshoot, meander, etc.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NS = 2 Special climbing turn</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NS = 3 Flat turn (with autopilot)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NS = 4 Elevator impulse</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NS = 5 Rudder impulse (with autopilot)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NS = 6 Acceleration/deceleration (with autopilot)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NS = 7 Maximum acceleration/deceleration (with autopilot)</td>
</tr>
<tr>
<td>1</td>
<td>41-45</td>
<td>15</td>
<td>INTSW. This variable selects the type of integration to be used. INTSW = 0 Runge-Kutta</td>
</tr>
</tbody>
</table>
### TABLE 2. INPUT DATA DECK, PROGRAM EB920 (cont.)

<table>
<thead>
<tr>
<th>Plot Control</th>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
</table>
|              | 2    | 1-75      | 1515   | INTSW = 1  2nd order Adams  
INTSW = 2  2nd order Non-classical  
INTSW = 3  Euler  
Card must be here (clank), even if no plots are required.  
ILOC (1), I=2,16. Value of I defines which variable will be plotted against time. Sequence defines order of plot on plotting paper as:  
Order of plotting  
Plot Variable:  
ILOC (2) - U, component of velocity in the x-direction, feet/second.  
ILOC (3) - V, component of velocity in the y-direction, feet/second.  
ILOC (4) - W, component of velocity z-direction, feet/second.  
ILOC (5) - P, angular velocity component about the x-axis, radians/second.  
ILOC (6) - Q, angular velocity component about the y-axis, radians/second.  
ILOC (7) - R, angular velocity component about the z-axis, radians/second.  
ILOC (8) - THETA (θ), pitch angle, radians.  
ILOC (9) - PSI (ψ), yaw angle, radians.  
ILOC (10) - PHI (φ), roll angle, radians.  
ILOC (11) - X, coordinate |
### TABLE 2. INPUT DATA DECK, PROGRAM EB920 (cont.)

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>point of sub position, feet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LOC (12) - Y, coordinate</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>point of sub position, feet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LOC (13) - Z, coordinate</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>point of sub position, (depth), feet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LOC (11) - DR, rudder position, radians</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LOC (15) - DS, stern elevator position, radians</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LOC (16) - DB, bow elevator position, radians</td>
</tr>
<tr>
<td>Timing</td>
<td>3</td>
<td>1-10</td>
<td>F10.5</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>11-20</td>
<td>F10.5</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>21-30</td>
<td>F10.5</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>31-40</td>
<td>F10.5</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>41-50</td>
<td>F10.5</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>51-60</td>
<td>F10.5</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>61-70</td>
<td>F10.5</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>1-80</td>
<td>8F10.5</td>
</tr>
<tr>
<td>Card</td>
<td>Column(s)</td>
<td>Format</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-----------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>5</td>
<td>1-40</td>
<td>5F10.5</td>
<td>T(i), θ, X, Y, Z. Allowable deviation of these parameters.</td>
</tr>
</tbody>
</table>

**Initial Conditions Array**

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>1-10</td>
<td>F10.5</td>
<td>Y(1), U, velocity in x-direction, feet/second</td>
</tr>
<tr>
<td>6</td>
<td>11-20</td>
<td>F10.5</td>
<td>Y(2), V, velocity in y-direction, feet/second</td>
</tr>
<tr>
<td>6</td>
<td>21-30</td>
<td>F10.5</td>
<td>Y(3), W, velocity in z-direction, feet/second</td>
</tr>
<tr>
<td>6</td>
<td>31-40</td>
<td>F10.5</td>
<td>Y(4), P, velocity about x-axis, radians/second</td>
</tr>
<tr>
<td>6</td>
<td>41-50</td>
<td>F10.5</td>
<td>Y(5), Q, velocity about y-axis, radians/second</td>
</tr>
<tr>
<td>6</td>
<td>51-60</td>
<td>F10.5</td>
<td>Y(6), R, velocity about z-axis, radians/second</td>
</tr>
<tr>
<td>6</td>
<td>61-70</td>
<td>F10.5</td>
<td>Y(7), THETA, pitch angle, radians</td>
</tr>
<tr>
<td>6</td>
<td>71-80</td>
<td>F10.5</td>
<td>Y(8), PSI, yaw angle, radians</td>
</tr>
<tr>
<td>7</td>
<td>1-10</td>
<td>F10.5</td>
<td>Y(9), PHI, roll angle, radians</td>
</tr>
<tr>
<td>7</td>
<td>11-20</td>
<td>F10.5</td>
<td>Y(10), X, coordinate point of sub position, feet</td>
</tr>
<tr>
<td>7</td>
<td>21-30</td>
<td>F10.5</td>
<td>Y(11), Y, coordinate point of sub position, feet</td>
</tr>
<tr>
<td>7</td>
<td>31-40</td>
<td>F10.5</td>
<td>Y(12), Z, coordinate point of sub position (depth), feet</td>
</tr>
</tbody>
</table>

**Coefficient Cards**

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>1-80</td>
<td>8F10.5</td>
<td>XQQ, XPC, XRP, XUD, XVR, XXQ, XUU, XVV</td>
</tr>
<tr>
<td>9</td>
<td>1-80</td>
<td>8F10.5</td>
<td>XWW, XDRDR, XDSDE, XDBDB, XVE, XWE, XOMDD, XODDE</td>
</tr>
<tr>
<td>10</td>
<td>1-80</td>
<td>8F10.5</td>
<td>YRD, YPD, YPAP, YPQ, YJR, YVD, YVQ, YFW</td>
</tr>
<tr>
<td>11</td>
<td>1-80</td>
<td>8F10.5</td>
<td>YWR, YR, YP, YAR, YVAR, YVR, YIR, YVR</td>
</tr>
</tbody>
</table>
**TABLE 2. INPUT DATA DECK, PROGRAM EB920 (cont.)**

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>1-80</td>
<td>F10.5</td>
<td>YV, YVAV</td>
</tr>
<tr>
<td>13</td>
<td>1-80</td>
<td>F10.5</td>
<td>YVW, YDR, YRE, YVE, YVAVE, YDRE</td>
</tr>
<tr>
<td>14</td>
<td>1-80</td>
<td>F10.5</td>
<td>ZQ, ZPP, ZRR, ZRP, ZWD, ZVR, ZVP, ZQ</td>
</tr>
<tr>
<td>15</td>
<td>1-80</td>
<td>F10.5</td>
<td>ZADS, ZAQ, ZSTR, ZW, ZWAW, ZAW, ZW, ZVV</td>
</tr>
<tr>
<td>16</td>
<td>1-80</td>
<td>F10.5</td>
<td>ZDS, ZDB, ZQE, ZWE, ZWAWE, ZDSE</td>
</tr>
<tr>
<td>17</td>
<td>1-80</td>
<td>F10.5</td>
<td>AKPD, AKRD, AKQR, AKPQ, AKPAP, AKP, AKR, AKVD</td>
</tr>
<tr>
<td>18</td>
<td>1-80</td>
<td>F10.5</td>
<td>AKVQ, AKWP, AKWR, AKSTR, AKV, AKVAV, AKWV, AKDR</td>
</tr>
<tr>
<td>19</td>
<td>1-80</td>
<td>F10.5</td>
<td>AMQD, AMP, AMRP, AMQP, AMQAQ, AMWD, AMVR, AMVP</td>
</tr>
<tr>
<td>20</td>
<td>1-80</td>
<td>F10.5</td>
<td>AMQ, AMAQDS, AMAWQ, AMSTR, AMW, AMMAW, AMAW, AMW</td>
</tr>
<tr>
<td>21</td>
<td>1-80</td>
<td>F10.5</td>
<td>AMV, AMDS, AMDB, AMQE, AMWE, AMWAVE, AMDSE</td>
</tr>
<tr>
<td>22</td>
<td>1-80</td>
<td>F10.5</td>
<td>ANRD, ANPD, ANPQ, ANQR, ANRAR, ANVD, ANWR, ANWP</td>
</tr>
<tr>
<td>23</td>
<td>1-80</td>
<td>F10.5</td>
<td>ANVQ, ANP, ANR, ANARDR, ANAVR, ANSTR, ANV, ANVAV</td>
</tr>
<tr>
<td>24</td>
<td>1-80</td>
<td>F10.5</td>
<td>ANVW, ANDR, ANRE, ANVE, ANVAVE, ANDRE</td>
</tr>
<tr>
<td>25</td>
<td>1-10</td>
<td>F10.5</td>
<td>IX, moment of inertia about the x-axis, slug-ft²</td>
</tr>
<tr>
<td>25</td>
<td>11-20</td>
<td>F10.5</td>
<td>IY, moment about y-axis, slug-ft²</td>
</tr>
<tr>
<td>25</td>
<td>21-30</td>
<td>F10.5</td>
<td>IZ, moment about z-axis, slug-ft²</td>
</tr>
<tr>
<td>25</td>
<td>31-40</td>
<td>F10.5</td>
<td>IXY, product of inertia about xy-axis, (slug-ft²)²</td>
</tr>
</tbody>
</table>

Submarine Constants
<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>41-50</td>
<td>F10.5</td>
<td>IXZ, product of inertia about xz-axis, (slug-ft²)²</td>
</tr>
<tr>
<td>25</td>
<td>51-60</td>
<td>F10.5</td>
<td>IYZ, product of inertia about yz-axis, (slug-ft²)²</td>
</tr>
<tr>
<td>26</td>
<td>1-10</td>
<td>F10.5</td>
<td>CW, weight of submarine, including water in free-flooding space, pounds</td>
</tr>
<tr>
<td>26</td>
<td>11-20</td>
<td>F10.5</td>
<td>CB, buoyancy, pounds</td>
</tr>
<tr>
<td>26</td>
<td>21-30</td>
<td>F10.5</td>
<td>UC, initial command speed, feet/second</td>
</tr>
<tr>
<td>26</td>
<td>31-40</td>
<td>F10.5</td>
<td>XB, x-component of center of buoyancy, feet</td>
</tr>
<tr>
<td>26</td>
<td>41-50</td>
<td>F10.5</td>
<td>YB, y-component of center of buoyancy, feet</td>
</tr>
<tr>
<td>26</td>
<td>51-60</td>
<td>F10.5</td>
<td>ZB, z-component of center of buoyancy, feet</td>
</tr>
<tr>
<td>27</td>
<td>1-10</td>
<td>F10.5</td>
<td>DR, initial value of rudder position, radians</td>
</tr>
<tr>
<td>27</td>
<td>11-20</td>
<td>F10.5</td>
<td>DS, initial value of stern elevator position, radians</td>
</tr>
<tr>
<td>27</td>
<td>21-30</td>
<td>F10.5</td>
<td>DB, initial value of bow elevator position, radians</td>
</tr>
<tr>
<td>27</td>
<td>31-40</td>
<td>F10.5</td>
<td>RHO, density of sea water, slugs/ft³</td>
</tr>
<tr>
<td>27</td>
<td>41-50</td>
<td>F10.5</td>
<td>AL, submarine length, feet</td>
</tr>
<tr>
<td>27</td>
<td>51-60</td>
<td>F10.5</td>
<td>AM, submarine mass, including water in free flooding space, slugs</td>
</tr>
<tr>
<td>28</td>
<td>1-10</td>
<td>F10.5</td>
<td>DRMAX, maximum rudder position (movement), radians</td>
</tr>
<tr>
<td>28</td>
<td>11-20</td>
<td>F10.5</td>
<td>ETAHI(θ-high), upper reference value of θ/USAG, dimensionless</td>
</tr>
<tr>
<td>28</td>
<td>21-30</td>
<td>F10.5</td>
<td>ETAI C(θ-low), lower reference value of θ/USAC, dimensionless</td>
</tr>
<tr>
<td>Card</td>
<td>Column(s)</td>
<td>Format</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-----------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>28</td>
<td>31-40</td>
<td>F10.5</td>
<td>A11, value of $a_i$ for ETA ETAHI</td>
</tr>
<tr>
<td>28</td>
<td>41-50</td>
<td>F10.5</td>
<td>A12, value of $b_i$ for ETA ETAHI</td>
</tr>
<tr>
<td>28</td>
<td>51-60</td>
<td>F10.5</td>
<td>A13, value of $c_i$ for ETA ETAHI</td>
</tr>
<tr>
<td>29</td>
<td>1-10</td>
<td>F10.5</td>
<td>A21, value of $a_i$ for ETALO ETA ETAHI</td>
</tr>
<tr>
<td>29</td>
<td>11-20</td>
<td>F10.5</td>
<td>A22, value of $b_i$ for ETALO ETA ETAHI</td>
</tr>
<tr>
<td>29</td>
<td>21-30</td>
<td>F10.5</td>
<td>A23, value of $c_i$ for ETALO ETA ETAHI</td>
</tr>
<tr>
<td>29</td>
<td>31-40</td>
<td>F10.5</td>
<td>A31, value of $a_i$ for ETALO ETA ETAHI</td>
</tr>
<tr>
<td>29</td>
<td>41-50</td>
<td>F10.5</td>
<td>A32, value of $b_i$ for ETALO ETA ETAHI</td>
</tr>
<tr>
<td>29</td>
<td>51-60</td>
<td>F10.5</td>
<td>A33, value of $c_i$ for ETALO ETA ETAHI</td>
</tr>
<tr>
<td>30</td>
<td>1-10</td>
<td>F10.5</td>
<td>XG, x-component of center of gravity, feet</td>
</tr>
<tr>
<td>30</td>
<td>11-20</td>
<td>F10.5</td>
<td>YG, y-component of center of gravity, feet</td>
</tr>
<tr>
<td>30</td>
<td>21-30</td>
<td>F10.5</td>
<td>ZG, z-component of center of gravity, feet</td>
</tr>
</tbody>
</table>

**Surface Control Schedule**

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>1-10</td>
<td>F10.5</td>
<td>TIME, various periods of time depending on NS in control routine, seconds</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NS</th>
<th>TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>0,2,3, 4,5</td>
<td>Not used</td>
</tr>
<tr>
<td>1</td>
<td>Initial steady state period (See fig.4)</td>
</tr>
<tr>
<td>6</td>
<td>Period of constant application of each command speed</td>
</tr>
<tr>
<td>7</td>
<td>Period of constant application of each</td>
</tr>
</tbody>
</table>
TABLE 2. INPUT DATA DECK, PROGRAM EB920 (cont.)

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>11-20</td>
<td>F10.5</td>
<td>command speed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>R1, negative rate of change of elevator position, radians/second during meander runs (NS = 1), radians/second (See fig. 4)</td>
</tr>
<tr>
<td>31</td>
<td>21-30</td>
<td>F10.5</td>
<td>DELTMA, maximum negative swing of elevator (must be more negative than DS, initial condition) during meander runs (NS = 1), radians (See fig. 4)</td>
</tr>
<tr>
<td>31</td>
<td>31-40</td>
<td>F10.5</td>
<td>SWMAX (θ'), execute pitch angle, maximum submarine dive angle during meander-type run (NS = 1), before turning elevators to a more positive value, radians</td>
</tr>
<tr>
<td>31</td>
<td>41-50</td>
<td>F10.5</td>
<td>command speed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>R1, positive rate of change of elevator position after submarine reaches θ', radians/second (See figure 4)</td>
</tr>
<tr>
<td>31</td>
<td>51-60</td>
<td>F10.5</td>
<td>DELTM1, maximum position to which elevators are moved during 'R2' change, radians (See fig. 4)</td>
</tr>
<tr>
<td>31</td>
<td>61-70</td>
<td>F10.5</td>
<td>DSF, final value of elevator position during DS-impulse run (NS = 2), radians</td>
</tr>
<tr>
<td>31</td>
<td>71-80</td>
<td>F10.5</td>
<td>DRF, final value of rudder position during DR-impulse run (NS = 5), radians</td>
</tr>
</tbody>
</table>

Additional Run Controls

If IOPT was "O" on card 1 of the last data deck, card 32 can be either blank for a normal exit or card 1 of a new data deck for additional runs. If IOPT = F on card 1 of the last data deck, see below.

IRUN, run numbers.
IRUN = 0 (blank card) a new data deck is read, a blank card at the start of a data deck results in a normal exit so two blank cards at the end.
TABLE 2. INPUT DATA DECK, PROGRAM EB920 (cont.)

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>33</td>
<td>1-5</td>
<td>15</td>
<td>will always end the run. IRUN = Integer. Use as new run number and continue reading cards below.</td>
</tr>
<tr>
<td>33</td>
<td>11-20</td>
<td>F10.5</td>
<td>NDEX, common location for parameter to be changed. Tables 3 and 4 show the variable names versus the common locations, NDEX.</td>
</tr>
<tr>
<td>33 + n</td>
<td>1-20</td>
<td>15, 5X</td>
<td>VALUE, new value of parameter changed in columns 1-5 of this card,</td>
</tr>
<tr>
<td>33 + n + 1</td>
<td>-</td>
<td>Blank</td>
<td>Repeat card 33 as desired.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Start new run after all changes have been made.</td>
</tr>
<tr>
<td>Field</td>
<td>Value</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JOB</td>
<td>EB920</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>GENERAL PURPOSE CARD PUNCHING FORM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Punch Instructions</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**NOTES**

Data Deck Format: EB920. Right adjust all integers.

**PRINTING**

Printed in U.S.A. X20-505-01 01/05/75

**PAGE** | **OF** | 20

---

**Figure 4. Input Data Form, Program EB920**
### TABLE 3. EB920, COMMON LAYOUT INDEX

<table>
<thead>
<tr>
<th>Location</th>
<th>Term</th>
<th>Location</th>
<th>Term</th>
<th>Location</th>
<th>Term</th>
<th>Location</th>
<th>Term</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>H</td>
<td>36</td>
<td>XWQ</td>
<td>71</td>
<td>ZRR</td>
<td>106</td>
<td>AKDR</td>
</tr>
<tr>
<td>2</td>
<td>HMAX</td>
<td>37</td>
<td>XUU</td>
<td>72</td>
<td>ZRP</td>
<td>107</td>
<td>AKSTR</td>
</tr>
<tr>
<td>3</td>
<td>HMIN</td>
<td>38</td>
<td>XVW</td>
<td>73</td>
<td>ZWD</td>
<td>108</td>
<td>AMQD</td>
</tr>
<tr>
<td>4</td>
<td>DH</td>
<td>39</td>
<td>XWN</td>
<td>74</td>
<td>ZVR</td>
<td>109</td>
<td>AMPF</td>
</tr>
<tr>
<td>5</td>
<td>FCT</td>
<td>40</td>
<td>XDROR</td>
<td>75</td>
<td>ZVP</td>
<td>110</td>
<td>AMRR</td>
</tr>
<tr>
<td>6</td>
<td>TL(1)</td>
<td>41</td>
<td>XDSDS</td>
<td>76</td>
<td>ZQ</td>
<td>111</td>
<td>AMRP</td>
</tr>
<tr>
<td>7</td>
<td>TL(2)</td>
<td>42</td>
<td>XDBDB</td>
<td>77</td>
<td>ZAQDS</td>
<td>112</td>
<td>AMQAQ</td>
</tr>
<tr>
<td>8</td>
<td>TL(3)</td>
<td>43</td>
<td>XVVE</td>
<td>78</td>
<td>ZNAQ</td>
<td>113</td>
<td>AMWD</td>
</tr>
<tr>
<td>9</td>
<td>TL(4)</td>
<td>44</td>
<td>XWVE</td>
<td>79</td>
<td>ZSTR</td>
<td>114</td>
<td>AMVR</td>
</tr>
<tr>
<td>10</td>
<td>TL(5)</td>
<td>45</td>
<td>XDROR</td>
<td>80</td>
<td>ZW</td>
<td>115</td>
<td>AMVP</td>
</tr>
<tr>
<td>11</td>
<td>TL(6)</td>
<td>46</td>
<td>XDSDS</td>
<td>81</td>
<td>ZAW</td>
<td>116</td>
<td>AMQ</td>
</tr>
<tr>
<td>12</td>
<td>TL(7)</td>
<td>47</td>
<td>YRD</td>
<td>82</td>
<td>ZAW</td>
<td>117</td>
<td>AMAQDS</td>
</tr>
<tr>
<td>13</td>
<td>TL(8)</td>
<td>48</td>
<td>YPD</td>
<td>83</td>
<td>ZWW</td>
<td>118</td>
<td>AMAWQ</td>
</tr>
<tr>
<td>14</td>
<td>TL(9)</td>
<td>49</td>
<td>YPAP</td>
<td>84</td>
<td>ZV</td>
<td>119</td>
<td>AMSTR</td>
</tr>
<tr>
<td>15</td>
<td>TL(10)</td>
<td>50</td>
<td>YPD</td>
<td>85</td>
<td>ZDS</td>
<td>120</td>
<td>AM</td>
</tr>
<tr>
<td>16</td>
<td>TL(11)</td>
<td>51</td>
<td>YTR</td>
<td>86</td>
<td>ZDB</td>
<td>121</td>
<td>AMAW</td>
</tr>
<tr>
<td>17</td>
<td>TL(12)</td>
<td>52</td>
<td>YVD</td>
<td>87</td>
<td>ZQE</td>
<td>122</td>
<td>AMAW</td>
</tr>
<tr>
<td>18</td>
<td>NGS</td>
<td>53</td>
<td>YVQ</td>
<td>88</td>
<td>ZWE</td>
<td>123</td>
<td>AMW</td>
</tr>
<tr>
<td>19</td>
<td>N</td>
<td>54</td>
<td>YWP</td>
<td>89</td>
<td>ZWAWE</td>
<td>124</td>
<td>AMVV</td>
</tr>
<tr>
<td>20</td>
<td>IS1</td>
<td>55</td>
<td>YWR</td>
<td>90</td>
<td>ZDSE</td>
<td>125</td>
<td>AMDS</td>
</tr>
<tr>
<td>21</td>
<td>NPT</td>
<td>56</td>
<td>YR</td>
<td>91</td>
<td>AKPD</td>
<td>126</td>
<td>AMDB</td>
</tr>
<tr>
<td>22</td>
<td>TLM</td>
<td>57</td>
<td>YP</td>
<td>92</td>
<td>AKRD</td>
<td>127</td>
<td>AME</td>
</tr>
<tr>
<td>23</td>
<td>RHGL2</td>
<td>58</td>
<td>YARDR</td>
<td>93</td>
<td>AKQR</td>
<td>128</td>
<td>AMVE</td>
</tr>
<tr>
<td>24</td>
<td>RHGL3</td>
<td>59</td>
<td>YYAR</td>
<td>94</td>
<td>AKPF</td>
<td>129</td>
<td>AMAWE</td>
</tr>
<tr>
<td>25</td>
<td>RHGL4</td>
<td>60</td>
<td>YSTR</td>
<td>95</td>
<td>AKPAP</td>
<td>130</td>
<td>AMDS</td>
</tr>
<tr>
<td>26</td>
<td>RHGL5</td>
<td>61</td>
<td>YV</td>
<td>96</td>
<td>AKP</td>
<td>131</td>
<td>AMRD</td>
</tr>
<tr>
<td>27</td>
<td>WMB</td>
<td>62</td>
<td>YVAV</td>
<td>97</td>
<td>AKR</td>
<td>132</td>
<td>AMNP</td>
</tr>
<tr>
<td>28</td>
<td>ETA</td>
<td>63</td>
<td>YW</td>
<td>98</td>
<td>AKVP</td>
<td>133</td>
<td>AMDP</td>
</tr>
<tr>
<td>29</td>
<td>ETAR</td>
<td>64</td>
<td>YDR</td>
<td>99</td>
<td>AKVQ</td>
<td>134</td>
<td>ANQ</td>
</tr>
<tr>
<td>30</td>
<td>ISW2</td>
<td>65</td>
<td>YRE</td>
<td>100</td>
<td>AKWP</td>
<td>135</td>
<td>ANR</td>
</tr>
<tr>
<td>31</td>
<td>XQK</td>
<td>66</td>
<td>YVE</td>
<td>101</td>
<td>AKVR</td>
<td>136</td>
<td>ANV</td>
</tr>
<tr>
<td>32</td>
<td>XRR</td>
<td>67</td>
<td>YVAVE</td>
<td>102</td>
<td>AKSTR</td>
<td>137</td>
<td>ANH</td>
</tr>
<tr>
<td>33</td>
<td>XRP</td>
<td>68</td>
<td>YORE</td>
<td>103</td>
<td>AKV</td>
<td>138</td>
<td>ANWP</td>
</tr>
<tr>
<td>34</td>
<td>XUD</td>
<td>69</td>
<td>ZQD</td>
<td>104</td>
<td>AKVAV</td>
<td>139</td>
<td>ANVQ</td>
</tr>
<tr>
<td>35</td>
<td>XVR</td>
<td>70</td>
<td>ZPP</td>
<td>105</td>
<td>AKYW</td>
<td>140</td>
<td>ANP</td>
</tr>
<tr>
<td>Location</td>
<td>Term</td>
<td>Location</td>
<td>Term</td>
<td>Location</td>
<td>Term</td>
<td>Location</td>
<td>Term</td>
</tr>
<tr>
<td>----------</td>
<td>-------</td>
<td>----------</td>
<td>-------</td>
<td>----------</td>
<td>-------</td>
<td>----------</td>
<td>-------</td>
</tr>
<tr>
<td>141</td>
<td>ANR</td>
<td>176</td>
<td>A13</td>
<td>211</td>
<td>Q</td>
<td></td>
<td></td>
</tr>
<tr>
<td>142</td>
<td>ANAR</td>
<td>177</td>
<td>A21</td>
<td>212</td>
<td>R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>143</td>
<td>ANAVR</td>
<td>178</td>
<td>A22</td>
<td>213</td>
<td>G</td>
<td></td>
<td></td>
</tr>
<tr>
<td>144</td>
<td>ANSTR</td>
<td>179</td>
<td>A23</td>
<td>214</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>145</td>
<td>ANV</td>
<td>180</td>
<td>A31</td>
<td>215</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>146</td>
<td>ANVAV</td>
<td>181</td>
<td>A32</td>
<td>216</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>147</td>
<td>ANW</td>
<td>182</td>
<td>A33</td>
<td>217</td>
<td>Y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>148</td>
<td>ANDR</td>
<td>183</td>
<td>XG</td>
<td>218</td>
<td>Z</td>
<td></td>
<td></td>
</tr>
<tr>
<td>149</td>
<td>ANRE</td>
<td>184</td>
<td>YG</td>
<td>219</td>
<td>TIME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>ANVE</td>
<td>185</td>
<td>ZG</td>
<td>220</td>
<td>R1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>151</td>
<td>ANWAVE</td>
<td>186</td>
<td>ILQC(1)</td>
<td>221</td>
<td>DELTMA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>152</td>
<td>ANDRE</td>
<td>187</td>
<td>ILQC(2)</td>
<td>222</td>
<td>SWMAX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>153</td>
<td>IX</td>
<td>188</td>
<td>ILQC(3)</td>
<td>223</td>
<td>R2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>154</td>
<td>IY</td>
<td>189</td>
<td>ILQC(4)</td>
<td>224</td>
<td>DELTMI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>155</td>
<td>IZ</td>
<td>190</td>
<td>ILQC(5)</td>
<td>225</td>
<td>DSF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>156</td>
<td>IXY</td>
<td>191</td>
<td>ILQC(6)</td>
<td>226</td>
<td>DRF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>157</td>
<td>IXZ</td>
<td>192</td>
<td>ILQC(7)</td>
<td>227</td>
<td>ICIC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>158</td>
<td>IYZ</td>
<td>193</td>
<td>ILQC(8)</td>
<td>228</td>
<td>NS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>159</td>
<td>CW</td>
<td>194</td>
<td>ILQC(9)</td>
<td>229</td>
<td>INTSW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>160</td>
<td>CB</td>
<td>195</td>
<td>ILQC(10)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>161</td>
<td>UC</td>
<td>196</td>
<td>ILQC(11)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>162</td>
<td>XB</td>
<td>197</td>
<td>ILQC(12)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>163</td>
<td>YB</td>
<td>198</td>
<td>ILQC(13)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>164</td>
<td>ZB</td>
<td>199</td>
<td>ILQC(14)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>165</td>
<td>DR</td>
<td>200</td>
<td>ILQC(15)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>166</td>
<td>DS</td>
<td>201</td>
<td>ILQC(16)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>167</td>
<td>DB</td>
<td>202</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>168</td>
<td>RHQ</td>
<td>203</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>169</td>
<td>AL</td>
<td>204</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>170</td>
<td>AM</td>
<td>205</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>171</td>
<td>DRMAX</td>
<td>206</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>172</td>
<td>ETAHI</td>
<td>207</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>173</td>
<td>ETA1</td>
<td>208</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>174</td>
<td>A11</td>
<td>209</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>175</td>
<td>A12</td>
<td>210</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Location</td>
<td>Term</td>
<td>Location</td>
<td>Term</td>
<td>Location</td>
<td>Term</td>
<td>Location</td>
<td>Term</td>
</tr>
<tr>
<td>----------</td>
<td>------</td>
<td>----------</td>
<td>------</td>
<td>----------</td>
<td>------</td>
<td>----------</td>
<td>------</td>
</tr>
<tr>
<td>174</td>
<td>A11</td>
<td>195</td>
<td>ILOC(10)</td>
<td>170</td>
<td>M</td>
<td>131</td>
<td>XRD</td>
</tr>
<tr>
<td>175</td>
<td>A12</td>
<td>196</td>
<td>ILOC(11)</td>
<td>117</td>
<td>MAQDS</td>
<td>119</td>
<td>XRE</td>
</tr>
<tr>
<td>176</td>
<td>A13</td>
<td>197</td>
<td>ILOC(12)</td>
<td>122</td>
<td>MAW</td>
<td>228</td>
<td>NG</td>
</tr>
<tr>
<td>177</td>
<td>A21</td>
<td>198</td>
<td>ILOC(13)</td>
<td>118</td>
<td>MAWQ</td>
<td>114</td>
<td>NTR</td>
</tr>
<tr>
<td>178</td>
<td>A22</td>
<td>199</td>
<td>ILOC(14)</td>
<td>126</td>
<td>MDS</td>
<td>115</td>
<td>NV</td>
</tr>
<tr>
<td>179</td>
<td>A23</td>
<td>200</td>
<td>ILOC(15)</td>
<td>125</td>
<td>MDS</td>
<td>116</td>
<td>NVAN</td>
</tr>
<tr>
<td>180</td>
<td>A31</td>
<td>201</td>
<td>ILOC(16)</td>
<td>130</td>
<td>MDSE</td>
<td>151</td>
<td>NWA</td>
</tr>
<tr>
<td>181</td>
<td>A32</td>
<td>229</td>
<td>INTSW</td>
<td>109</td>
<td>MMP</td>
<td>136</td>
<td>NVD</td>
</tr>
<tr>
<td>182</td>
<td>A33</td>
<td>204</td>
<td>IOPEN</td>
<td>116</td>
<td>NQ</td>
<td>150</td>
<td>NVE</td>
</tr>
<tr>
<td>160</td>
<td>CB</td>
<td>206</td>
<td>IOPT</td>
<td>112</td>
<td>M2AQ</td>
<td>139</td>
<td>NVO</td>
</tr>
<tr>
<td>159</td>
<td>CW</td>
<td>202</td>
<td>IPOINT</td>
<td>108</td>
<td>M2QD</td>
<td>147</td>
<td>NVV</td>
</tr>
<tr>
<td>167</td>
<td>DB</td>
<td>203</td>
<td>IRUN</td>
<td>127</td>
<td>M2RE</td>
<td>138</td>
<td>NW</td>
</tr>
<tr>
<td>221</td>
<td>DELTM</td>
<td>20</td>
<td>ISL</td>
<td>111</td>
<td>M2RP</td>
<td>137</td>
<td>NXR</td>
</tr>
<tr>
<td>224</td>
<td>DELTH</td>
<td>30</td>
<td>ISW2</td>
<td>110</td>
<td>M2RR</td>
<td>210</td>
<td>P</td>
</tr>
<tr>
<td>4</td>
<td>DH</td>
<td>153</td>
<td>IX</td>
<td>119</td>
<td>MSTR</td>
<td>215</td>
<td>PHI</td>
</tr>
<tr>
<td>165</td>
<td>DR</td>
<td>156</td>
<td>IXY</td>
<td>115</td>
<td>MVP</td>
<td>214</td>
<td>PSL</td>
</tr>
<tr>
<td>226</td>
<td>DF</td>
<td>157</td>
<td>IXZ</td>
<td>114</td>
<td>MWR</td>
<td>211</td>
<td>Q</td>
</tr>
<tr>
<td>171</td>
<td>DRM</td>
<td>154</td>
<td>IY</td>
<td>121</td>
<td>M2V</td>
<td>212</td>
<td>R</td>
</tr>
<tr>
<td>166</td>
<td>DS</td>
<td>158</td>
<td>IYZ</td>
<td>120</td>
<td>NW</td>
<td>168</td>
<td>RHO</td>
</tr>
<tr>
<td>225</td>
<td>JRF</td>
<td>155</td>
<td>IZ</td>
<td>121</td>
<td>M2WA</td>
<td>23</td>
<td>RHL2</td>
</tr>
<tr>
<td>28</td>
<td>ETA</td>
<td>106</td>
<td>KDR</td>
<td>129</td>
<td>M2WE</td>
<td>21</td>
<td>RHL</td>
</tr>
<tr>
<td>172</td>
<td>ETAH</td>
<td>96</td>
<td>KP</td>
<td>113</td>
<td>M2MD</td>
<td>75</td>
<td>RHL</td>
</tr>
<tr>
<td>173</td>
<td>ETAI</td>
<td>95</td>
<td>KPAP</td>
<td>128</td>
<td>M2VE</td>
<td>26</td>
<td>RHL</td>
</tr>
<tr>
<td>29</td>
<td>ETAI</td>
<td>91</td>
<td>KPD</td>
<td>123</td>
<td>M2MV</td>
<td>220</td>
<td>R1</td>
</tr>
<tr>
<td>5</td>
<td>FCT</td>
<td>94</td>
<td>KFT</td>
<td>19</td>
<td>N</td>
<td>223</td>
<td>R2</td>
</tr>
<tr>
<td>1</td>
<td>H</td>
<td>93</td>
<td>KJR</td>
<td>113</td>
<td>M2VR</td>
<td>222</td>
<td>HRAX</td>
</tr>
<tr>
<td>2</td>
<td>HMAX</td>
<td>97</td>
<td>KR</td>
<td>112</td>
<td>M2BD</td>
<td>213</td>
<td>THETA</td>
</tr>
<tr>
<td>3</td>
<td>HMIN</td>
<td>92</td>
<td>KED</td>
<td>118</td>
<td>NDR</td>
<td>219</td>
<td>TMME</td>
</tr>
<tr>
<td>227</td>
<td>ICY</td>
<td>102</td>
<td>KSTR</td>
<td>152</td>
<td>M2RE</td>
<td>6</td>
<td>TL(1)</td>
</tr>
<tr>
<td>166</td>
<td>ILOC(1)</td>
<td>107</td>
<td>KSTRE</td>
<td>18</td>
<td>NG3</td>
<td>7</td>
<td>TL(2)</td>
</tr>
<tr>
<td>187</td>
<td>ILOC(2)</td>
<td>103</td>
<td>KY</td>
<td>110</td>
<td>NP</td>
<td>8</td>
<td>TL(3)</td>
</tr>
<tr>
<td>188</td>
<td>ILOC(3)</td>
<td>104</td>
<td>KVA</td>
<td>132</td>
<td>NPD</td>
<td>9</td>
<td>TL(4)</td>
</tr>
<tr>
<td>189</td>
<td>ILOC(4)</td>
<td>98</td>
<td>KVd</td>
<td>205</td>
<td>NPLT</td>
<td>10</td>
<td>TL(5)</td>
</tr>
<tr>
<td>190</td>
<td>ILOC(5)</td>
<td>99</td>
<td>KVT</td>
<td>21</td>
<td>NPT</td>
<td>11</td>
<td>TL(6)</td>
</tr>
<tr>
<td>191</td>
<td>ILOC(6)</td>
<td>105</td>
<td>KVH</td>
<td>133</td>
<td>NQ</td>
<td>12</td>
<td>TL(7)</td>
</tr>
<tr>
<td>192</td>
<td>ILOC(7)</td>
<td>100</td>
<td>KWP</td>
<td>134</td>
<td>NR</td>
<td>13</td>
<td>TL(8)</td>
</tr>
<tr>
<td>193</td>
<td>ILOC(8)</td>
<td>101</td>
<td>KXR</td>
<td>111</td>
<td>NR</td>
<td>14</td>
<td>TL(9)</td>
</tr>
<tr>
<td>194</td>
<td>ILOC(9)</td>
<td>169</td>
<td>L</td>
<td>135</td>
<td>NRAR</td>
<td>15</td>
<td>TL(10)</td>
</tr>
<tr>
<td>Location</td>
<td>Term</td>
<td>Location</td>
<td>Term</td>
<td>Location</td>
<td>Term</td>
<td>Location</td>
<td>Term</td>
</tr>
<tr>
<td>----------</td>
<td>------</td>
<td>----------</td>
<td>------</td>
<td>----------</td>
<td>------</td>
<td>----------</td>
<td>------</td>
</tr>
<tr>
<td>16</td>
<td>TL(11)</td>
<td>56</td>
<td>YR</td>
<td>83</td>
<td>ZWW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>TL(12)</td>
<td>47</td>
<td>YRD</td>
<td>22</td>
<td>TLIM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>207</td>
<td>U</td>
<td>60</td>
<td>YSTAR</td>
<td>161</td>
<td>UC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>208</td>
<td>V</td>
<td>59</td>
<td>YVAR</td>
<td>209</td>
<td>W</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>WMB</td>
<td>67</td>
<td>YWAVE</td>
<td>216</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>162</td>
<td>XB</td>
<td>66</td>
<td>YVE</td>
<td>42</td>
<td>XDBDB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>XDRDR</td>
<td>63</td>
<td>YW</td>
<td>45</td>
<td>XDRDRE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>XDSDS</td>
<td>55</td>
<td>YWR</td>
<td>16</td>
<td>XDSDSE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>183</td>
<td>XG</td>
<td>77</td>
<td>ZAQDS</td>
<td>31</td>
<td>XQQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>XRP</td>
<td>164</td>
<td>ZB</td>
<td>32</td>
<td>XRR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>XUD</td>
<td>85</td>
<td>ZDS</td>
<td>37</td>
<td>XUU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>XVR</td>
<td>185</td>
<td>ZG</td>
<td>38</td>
<td>XVV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>XVVE</td>
<td>76</td>
<td>ZQ</td>
<td>36</td>
<td>XWQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>XW 72</td>
<td>87</td>
<td>ZQE</td>
<td>44</td>
<td>XWWE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>217</td>
<td>Y</td>
<td>71</td>
<td>ZRR</td>
<td>58</td>
<td>YARDR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>163</td>
<td>YB</td>
<td>75</td>
<td>ZVP</td>
<td>64</td>
<td>YDR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>68</td>
<td>YDRE</td>
<td>81</td>
<td>ZVR</td>
<td>49</td>
<td>YPAP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>184</td>
<td>YG</td>
<td>80</td>
<td>ZW</td>
<td>57</td>
<td>YP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>YPD</td>
<td>89</td>
<td>ZWAWE</td>
<td>50</td>
<td>YPJ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>YQR</td>
<td>88</td>
<td>ZWE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4. OUTPUT DATA

a. Printout - This program will print out titled numbered pages with the variable as shown in table 5. All input values are printed out at the start of the run as shown in figure 5.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Format</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>E13.6</td>
<td>feet/second</td>
<td>Velocity component in the x-direction</td>
</tr>
<tr>
<td>V</td>
<td>E13.6</td>
<td>feet/second</td>
<td>Velocity component in the y-direction</td>
</tr>
<tr>
<td>W</td>
<td>E13.6</td>
<td>feet/second</td>
<td>Velocity component in the z-direction</td>
</tr>
<tr>
<td>P</td>
<td>E13.6</td>
<td>radians/second</td>
<td>Velocity about x-axis</td>
</tr>
<tr>
<td>Q</td>
<td>E13.6</td>
<td>radians/second</td>
<td>Velocity about y-axis</td>
</tr>
<tr>
<td>R</td>
<td>E13.6</td>
<td>radians/second</td>
<td>Velocity about z-axis</td>
</tr>
<tr>
<td>THETA</td>
<td>E13.6</td>
<td>radians</td>
<td>Pitch angle</td>
</tr>
<tr>
<td>PSI</td>
<td>E13.6</td>
<td>radians</td>
<td>Yaw angle</td>
</tr>
<tr>
<td>PHI</td>
<td>E13.6</td>
<td>radians</td>
<td>Yaw angle</td>
</tr>
<tr>
<td>X</td>
<td>E13.6</td>
<td>feet</td>
<td>Coordinate point of sub position</td>
</tr>
<tr>
<td>Y</td>
<td>E13.6</td>
<td>feet</td>
<td>Coordinate point of sub position</td>
</tr>
<tr>
<td>Z</td>
<td>E13.6</td>
<td>feet</td>
<td>Coordinate point of sub position</td>
</tr>
<tr>
<td>T</td>
<td>E13.6</td>
<td>seconds</td>
<td>Time during run</td>
</tr>
<tr>
<td>H</td>
<td>E13.6</td>
<td>seconds</td>
<td>Integrating time period</td>
</tr>
</tbody>
</table>

All output variables are printed as shown in sample data sheet, figure 6. This data is a function of the time of the run, T. The frequency of printout is a function of NPNT, printed each NPNTth integration cycle. If H, integration time period is 0.25 second and NPNT = 8:

a. INTSW = 0 (Runge-Kutta integration)

\[
T^* = \frac{NPNT \times H}{TCIC} = \frac{8 \times 0.25}{4} = 0.5 \text{ seconds}
\]

Printout would be each half second
<table>
<thead>
<tr>
<th>CO</th>
<th>0.1 0000E-01</th>
<th>MDC</th>
<th>0.1 0000E-01</th>
<th>ZCG</th>
<th>0.1 0000E-01</th>
<th>LGO</th>
<th>0.1 0000E-01</th>
<th>KPA</th>
<th>-0.2 0000E-04</th>
</tr>
</thead>
<tbody>
<tr>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>LBD</td>
<td>0.1 0000E-01</td>
<td>ZBP</td>
<td>0.0</td>
<td>KBD</td>
<td>0.1 0000E-01</td>
<td>KBR</td>
<td>-0.1 8000E-04</td>
</tr>
<tr>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>WBD</td>
<td>0.1 0400E-02</td>
<td>KBD</td>
<td>0.0</td>
<td>KBD</td>
<td>0.1 0000E-02</td>
<td>KBD</td>
<td>-0.1 0000E-03</td>
</tr>
<tr>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>SWD</td>
<td>0.1 0000E-02</td>
<td>KBD</td>
<td>0.1 0000E-04</td>
<td>WBD</td>
<td>0.1 0000E-02</td>
</tr>
<tr>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>SWD</td>
<td>0.1 0000E-02</td>
<td>KBD</td>
<td>0.1 0000E-04</td>
<td>WBD</td>
<td>0.1 0000E-02</td>
</tr>
<tr>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>SWD</td>
<td>0.1 0000E-02</td>
<td>KBD</td>
<td>0.1 0000E-04</td>
<td>WBD</td>
<td>0.1 0000E-02</td>
</tr>
<tr>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>SWD</td>
<td>0.1 0000E-02</td>
<td>KBD</td>
<td>0.1 0000E-04</td>
<td>WBD</td>
<td>0.1 0000E-02</td>
</tr>
<tr>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>SWD</td>
<td>0.1 0000E-02</td>
<td>KBD</td>
<td>0.1 0000E-04</td>
<td>WBD</td>
<td>0.1 0000E-02</td>
</tr>
<tr>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>SWD</td>
<td>0.1 0000E-02</td>
<td>KBD</td>
<td>0.1 0000E-04</td>
<td>WBD</td>
<td>0.1 0000E-02</td>
</tr>
<tr>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>SWD</td>
<td>0.1 0000E-02</td>
<td>KBD</td>
<td>0.1 0000E-04</td>
<td>WBD</td>
<td>0.1 0000E-02</td>
</tr>
<tr>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>SWD</td>
<td>0.1 0000E-02</td>
<td>KBD</td>
<td>0.1 0000E-04</td>
<td>WBD</td>
<td>0.1 0000E-02</td>
</tr>
<tr>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>SWD</td>
<td>0.1 0000E-02</td>
<td>KBD</td>
<td>0.1 0000E-04</td>
<td>WBD</td>
<td>0.1 0000E-02</td>
</tr>
<tr>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>SWD</td>
<td>0.1 0000E-02</td>
<td>KBD</td>
<td>0.1 0000E-04</td>
<td>WBD</td>
<td>0.1 0000E-02</td>
</tr>
<tr>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>SWD</td>
<td>0.1 0000E-02</td>
<td>KBD</td>
<td>0.1 0000E-04</td>
<td>WBD</td>
<td>0.1 0000E-02</td>
</tr>
<tr>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>SWD</td>
<td>0.1 0000E-02</td>
<td>KBD</td>
<td>0.1 0000E-04</td>
<td>WBD</td>
<td>0.1 0000E-02</td>
</tr>
<tr>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>SWD</td>
<td>0.1 0000E-02</td>
<td>KBD</td>
<td>0.1 0000E-04</td>
<td>WBD</td>
<td>0.1 0000E-02</td>
</tr>
<tr>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>SWD</td>
<td>0.1 0000E-02</td>
<td>KBD</td>
<td>0.1 0000E-04</td>
<td>WBD</td>
<td>0.1 0000E-02</td>
</tr>
<tr>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>SWD</td>
<td>0.1 0000E-02</td>
<td>KBD</td>
<td>0.1 0000E-04</td>
<td>WBD</td>
<td>0.1 0000E-02</td>
</tr>
<tr>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>SWD</td>
<td>0.1 0000E-02</td>
<td>KBD</td>
<td>0.1 0000E-04</td>
<td>WBD</td>
<td>0.1 0000E-02</td>
</tr>
<tr>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>SWD</td>
<td>0.1 0000E-02</td>
<td>KBD</td>
<td>0.1 0000E-04</td>
<td>WBD</td>
<td>0.1 0000E-02</td>
</tr>
<tr>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>SWD</td>
<td>0.1 0000E-02</td>
<td>KBD</td>
<td>0.1 0000E-04</td>
<td>WBD</td>
<td>0.1 0000E-02</td>
</tr>
<tr>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>SWD</td>
<td>0.1 0000E-02</td>
<td>KBD</td>
<td>0.1 0000E-04</td>
<td>WBD</td>
<td>0.1 0000E-02</td>
</tr>
<tr>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>SWD</td>
<td>0.1 0000E-02</td>
<td>KBD</td>
<td>0.1 0000E-04</td>
<td>WBD</td>
<td>0.1 0000E-02</td>
</tr>
<tr>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>SWD</td>
<td>0.1 0000E-02</td>
<td>KBD</td>
<td>0.1 0000E-04</td>
<td>WBD</td>
<td>0.1 0000E-02</td>
</tr>
<tr>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>SWD</td>
<td>0.1 0000E-02</td>
<td>KBD</td>
<td>0.1 0000E-04</td>
<td>WBD</td>
<td>0.1 0000E-02</td>
</tr>
<tr>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>WBD</td>
<td>0.1 0000E-01</td>
<td>SWD</td>
<td>0.1 0000E-02</td>
<td>KBD</td>
<td>0.1 0000E-04</td>
<td>WBD</td>
<td>0.1 0000E-02</td>
</tr>
</tbody>
</table>

**Figure 5. Input Variable Format, EB920**
b. INTSW = 1 (Adams integration)

\[ T' = \frac{NPNT \times H}{LCYC} = \frac{8 \times 0.25}{1} = 2.0 \text{ seconds} \]

Printout would be every two seconds.

c. Graphical - This program can optionally plot any of the printed values noted in Section 4.a on a CALCOMP plotter. In addition, the control parameters, DS, DB, and DR (elevator and rudder positions) can be plotted. These variables are each plotted (as requested) as a function of time as shown in figure 7.

5. LISTINGS AND FLOW CHART

The listings and flow chart for program EB920 are given in appendix A.
Figure 7. Graphical Output EB920
B. PROGRAM 2C790, SUBMARINE SIMULATION, LONGITUDINAL FREEDOM

1. DESCRIPTION

This program calculates the dynamic changes in submarine position, velocity, and attitude as a function of time for longitudinal freedom. It is very similar to Program EB920, except that lateral freedom (rudder change with attendant roll and turn) is not provided for. Therefore, only longitudinal runs due to thrust or elevator changes can be provided with this program.

The purpose in preparing this program, in view of the existence of EB920 (all degrees of freedom), is the shortened computer running time for the abbreviated program. This will be a real advantage where a large number of meander, overshoot, or acceleration runs are required. Figure 8 is a block diagram, showing the general layout of the program and the subroutines used. Equation (9) through (12) give the mathematical model used. It follows "Standard Equations of Motion for Submarine Limitations" except that all lateral coefficients have been removed.

The program is written in basic FORTRAN IV for use on any digital computer with a FORTRAN compiler. It occupied 6K words when run on IBM 360/40 computer. Program results are printed as a function of time and individual parameters can be plotted if associated CALCOMP plotter software is available.

The program will accept values of initial conditions for control surface positions, speed (and components) attitude and depth. This allows the submarine to be placed in a steady state dive, level flight, etc., at time zero or start of the computer run. These inputs are:

- UC - command speed
- DS - stern elevator position
- DB - bow elevator position
- W - component of velocity in the z-direction
- Q - angular acceleration component about the y-axis relative to fluid.
- THETA - angle of pitch
- Z - depth

This program will give identical results as the longitudinal channel of EB920 when the same coefficients are used but the running time is one-eighth and the space required is one-third.

2. SUBROUTINE DESCRIPTIONS

The subroutines utilized by this program are shown in block diagram form in figure 8. A brief description of each subroutine is included for clarification of operation of the total program.

a. INPUT - This subroutine reads a data deck specifying program options, control, and initial dynamic conditions and the particular
Figure 8. CO790 Submarine Simulation Program, Block Diagram

CALL INPUT
Read program controls
Read coefficients, etc.

WRITE INPUT DATA

CALL INTEG

CALL EVAL

CALL CONTR
Controls sub motion
Calc $\ddot{u}$, $\dot{w}$, $\ddot{\theta}$, $\dot{\phi}$, $\ddot{\theta}$, $\dot{\phi}$

2nd Order Adams integration

WRITE $u$, $x$, $w$, $\gamma$, $\phi$, $t$, $\theta$, $\phi$

END?

CALL PLOTR(V11st) $u$, $x$, $w$, $\gamma$, $\phi$, $t$, $\theta$, $\phi$

Figure 8. CO790 Submarine Simulation Program, Block Diagram
Equations of Motion for Longitudinal Freedom Only (ZC790 Math Model)

**AXIAL FORCE**

\[ m [ \ddot{u} + \omega_\phi^2 - X_a \dot{q}^2 + Z_a \dot{\theta}^2 ] = \]

\[ + \frac{p}{2} l^3 [ X_{bb} \dot{q}^2 ] \]

\[ + \frac{p}{2} l^3 [ X_a \ddot{u} + X_{w_\phi} \dot{w} ] \]

\[ + \frac{p}{2} l^2 [ X_{ww} u^2 + X_{ww} w^2 ] \]

\[ + \frac{p}{2} l^3 [ X_{ssss} \delta \phi^2 + X_{ssss} \delta \beta^2 ] \]

\[ + \frac{p}{2} l^2 [ a_z u^2 + b_z u \dot{u} + c_z \dot{u}^2 ] \]

\[ - (W - B) \sin \theta \]

\[ + \frac{p}{2} l^2 [ X_{ww} \dot{w}^2 + X_{ssss} \delta \theta^2 u^2 ] (\eta - 1) \]

**KINEMATIC RELATIONS**

\[ U^2 = u^2 + w^2 \]

\[ \dot{\theta} = \theta \]

\[ \dot{Z}_o = \dot{w} \cos \theta - \dot{u} \sin \theta \]

\[ \dot{X}_o = u \cos \theta + \dot{w} \sin \theta \]
PITCHING MOMENT

\[ I_2 \dot{\theta} + m [ Z_a (\dot{u} + \omega g) - X_a (\dot{\omega} - u \dot{g}) ] = \]

\[ \frac{e}{2} l^6 [ M_t \dot{\theta} + M_w \dot{\omega} g \dot{g} ] \]

\[ + \frac{e}{2} l^4 [ M_w \dot{\omega} ] \]

\[ + \frac{e}{2} l^4 [ M_s u \dot{g} + M_{ss} u \dot{g} \dot{g} + M_{sw} \dot{\omega} ] \]

\[ + \frac{e}{2} l^3 [ M_{w1} u \dot{\omega} + M_{w2} \dot{\omega} \dot{\omega} ] \]

\[ + \frac{e}{2} l^3 [ M_{s1} u^2 s + M_{s2} u^2 s ] \]

\[ - (X_c \omega - X_w B) \cos \theta - (Z_c \omega - Z_w B) \sin \theta \]

\[ + \frac{e}{2} l^4 M_{s1} u \dot{g} (\eta - 1) \]

\[ + \frac{e}{2} l^3 [ M_{w1} u \dot{\omega} + M_{w2} \dot{\omega} \dot{\omega} + M_{s1} \dot{\epsilon} s - u ] (\eta - 1) \]
NORMAL FORCE

\[ m [ \ddot{\omega} - uq_8 - Z_a q_a^2 - X_a \dot{q}_a ] = \]

\[ + \frac{c}{2} l^4 [ Z_8 \dot{q}_8 ] \]

\[ + \frac{c}{2} l^3 [ Z_8 \dot{\omega} ] \]

\[ + \frac{c}{2} l^3 [ Z_3 uq_8 + Z_{15} sss \dot{q}_8 \dot{s}_2 + Z_{w1} w_{1w} ] \]

\[ + \frac{c}{2} l^2 [ Z_{-w} u^2 + Z_{-w} w + Z_{w1w} w_{1w} ] \]

\[ + \frac{c}{2} l^2 [ Z_{w1w} w_{1w} + Z_{w1w} w_{1w} ] \]

\[ + \frac{c}{2} l^2 [ Z_{ss} u^2 \delta_s + Z_{ssb} u^2 \delta_s ] \]

\[ + (w - B) \cos \theta \]

\[ + \frac{c}{2} l^3 Z_{87} uq_8 (n - 1) \]

\[ + \frac{c}{2} l^2 [ Z_{w1} w + Z_{w1w} w_{1w} + Z_{ss} s_s u^2 ] (n - 1) \]
coefficients and constants for the submarine to be simulated. It provides for holding initial values of all variables for use on a subsequent run. It provides for reading additional cards for successive runs, and finally for exit from the program on a blank card.

b. INTEG - The only integration algorithm available is 2nd order Adams. This program is used mainly for verification runs rather than research so different integration methods are not required.

c. EVALI - This subroutine evaluates the equations of motion in accordance with the mathematical model given in section II. B. 1. It is identical to the equations used in program ZB920 except that the lateral, roll, and yaw channels have been removed and all longitudinal coefficients using \( v, p, \) and \( r \) have been set to zero.

d. CONTR - The block diagram of figure 9 outlines the CONTR subroutine. This subroutine allows selection of one of several preplanned maneuvers that are used most often in submarine research studies. The operation is identical to that of the CONTR subroutine in program ZB920 except that provisions for moving the rudder have been removed. The variable \( NS \) will set the desired bow or sternplane schedule as follows:

\[
\begin{align*}
NS &= 0 \quad \text{Constant input value} \\
NS &= 1 \quad \text{Meander or Overshoot (figure 3)} \\
NS &= 2 \quad DS - \text{Impulse} \\
NS &= 3 \quad \text{Acceleration/Deceleration} \\
NS &= 4 \quad \text{Maximum Acceleration/Deceleration}
\end{align*}
\]

A useful addition of the program for elevator-impulse control is the computer output of punched cards of the submarine pitch angle versus time. These are punched in 2E15.7 format (\( \theta, \text{TIME} \)) from \( T_0 \) to \( T_{\text{LIM}} \). The cards are useful in other programs associated with submarine response and described elsewhere in this report. Most associated programs are based on cards punched at two-second intervals. To achieve this, the input value of integration interval \( H \), must be 0.25 seconds and \( NPNT \) must be eight.

The integration in ZC 790 is performed with the second-order Adams method. The results may be slightly different from integration to that performed by the Runge-Kutta method in the EB920 program. This difference is magnified for rapidly changing conditions as prevail in impulse runs. Therefore, care should be exercised in the use of impulse runs for this program, and comparison made with the EB920 program. Greatest accuracy will accrue by the use of the Runge-Kutta integration of EB920.

The autopilot control of this program uses the identical equation (8) used by the autopilot control in program EB920.

e. PLOTROU - This is identical to PLOTROU of EB920 except that the variables \( v, p, r, \Phi, \Psi, y, \) and \( \Delta r \) are not saved or plotted.

3. INPUT DATA DECK
Figure 9. Subroutine CONTR, ZC790, Block Diagram

This section describes the input data deck, defines the FORTRAN variables in terms of the equations of motion defined in the mathematical model, and shows how to input data to perform various kinds of simulated submarine operations. The coefficients referred to here are those included in the reference "Standard Equations of Motion for Submarine Simulation". The coefficients are restricted to those in X, Z, and M, corresponding to those used in calculating axial force, normal force, and pitching moment on the submarine. The coefficients in Y, K, and N are not used because they affect only lateral forces and motion of the submarine. The input deck should follow table 6 exactly.

Table 6. Input Data Deck, Program ZC790

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control Flags</td>
<td>1</td>
<td>1-5</td>
<td>I5</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>6-10</td>
<td>If</td>
</tr>
</tbody>
</table>
### TABLE 6. INPUT DATA DECK, PROGRAM ZC790 (cont.)

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11-15</td>
<td>I5</td>
<td>IRUN. Identification number for individual runs. If IRUN = 0 a normal exit is made.</td>
</tr>
<tr>
<td>1</td>
<td>16-20</td>
<td>I5</td>
<td>NPLT. Data will plot at T0 each NPLTth integration step.</td>
</tr>
<tr>
<td>1</td>
<td>21-25</td>
<td>I5</td>
<td>IOPT. This option will allow changing any one or more of the input values for a succeeding run without putting in all the other values. IOPT = 1 to exercise; blank for no selection. Another run may still follow, for IOPT = 0, but all input data cards must be read again. More information on this variable is included at card 19.</td>
</tr>
</tbody>
</table>
| 1    | 26-30     | I5     | NS. This variable selects type of submarine control in CONTR subroutine 
NS = 0, Fixed control per initial conditions 
NS = 1, Overshoot, meander, ect. 
NS = 2, DS-impulse 
NS = 3, Acceleration/deceleration 
NS = 4, Maximum acceleration/deceleration |
| Plot card |          |        | Card must be here (blank) even if no plots are required. |
| 2    | 1-35      | 7I5    | ILOC(I) I = 2,8. Value of I defines which variable will be plotted against time. Sequence defines order of plot on plotting paper as: |
|      |           |        | Plot variable: 
ILOC(2) - U, component of velocity in the X-direction, feet/second |
## TABLE 6. INPUT DATA DECK, PROGRAM ZC790 (cont.)

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>ILOC(3) - W, component of velocity in the z-direction, feet/second</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ILOC(4) - Q, angular velocity about y-axis, radians/second</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ILOC(5) - theta (θ), pitch angle, radians</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ILOC(6) - Z, depth, feet</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ILOC(7) - DS, stern plane position, degrees</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ILOC(8) - DB, bow plane position, degrees</td>
</tr>
</tbody>
</table>

### Timing

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1-10</td>
<td>F10.5</td>
<td>T0, starting time, seconds</td>
</tr>
<tr>
<td>3</td>
<td>11-20</td>
<td>F10.5</td>
<td>HO, integration time increment, seconds</td>
</tr>
<tr>
<td>3</td>
<td>21-30</td>
<td>F10.5</td>
<td>TLM, time of run, seconds</td>
</tr>
</tbody>
</table>

### Initial conditions

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1-10</td>
<td>F10.5</td>
<td>X(1), U, velocity component in the X-direction, feet/second</td>
</tr>
<tr>
<td>4</td>
<td>11-20</td>
<td>F10.5</td>
<td>X(2), W, velocity component in z-direction, feet/second</td>
</tr>
<tr>
<td>4</td>
<td>21-30</td>
<td>F10.5</td>
<td>X(3), Q, angular velocity about y-axis, radians/second</td>
</tr>
<tr>
<td>4</td>
<td>31-40</td>
<td>F10.5</td>
<td>X(4), THETA(θ), pitch angle, radians</td>
</tr>
<tr>
<td>4</td>
<td>41-50</td>
<td>F10.5</td>
<td>X(5), Z, depth, feet</td>
</tr>
</tbody>
</table>

### Coefficient Cards

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1-80</td>
<td>8F10.5</td>
<td>XQ, XD, XU, XXU, XWW, XDSDS, XDBDB, XWWE</td>
</tr>
<tr>
<td>6</td>
<td>1-10</td>
<td>F10.5</td>
<td>XTUSXSE</td>
</tr>
<tr>
<td>7</td>
<td>1-80</td>
<td>8F10.5</td>
<td>ZQQ, ZWD, ZQ, ZAQDS, ZADQ, ZSTZ, ZZ, ZWAU</td>
</tr>
<tr>
<td>Card</td>
<td>Column(s)</td>
<td>Format</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-----------</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>8</td>
<td>1-80</td>
<td>8F10.5</td>
<td>ZAW, ZWW, ZDS, ZDB, ZQE, ZWE, ZWAWE, ZDSE</td>
</tr>
<tr>
<td>9</td>
<td>1-80</td>
<td>8F10.5</td>
<td>AMQD, AMQAQ, AMWD, AMQ, AMAQS, AMAQW, AMSTR, AMW</td>
</tr>
<tr>
<td>10</td>
<td>1-80</td>
<td>8F10.5</td>
<td>AMWAW, AMAW, AMWW, AMDS, AMDB, AMQE, AMWE, AMAWE</td>
</tr>
<tr>
<td>11</td>
<td>1-80</td>
<td>F10.5</td>
<td>AMDSE</td>
</tr>
</tbody>
</table>

**Submarine Constants**

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>1-10</td>
<td>F10.5</td>
<td>IY, moment of inertia about the y-axis, slug/ft²</td>
</tr>
<tr>
<td>13</td>
<td>1-10</td>
<td>F10.5</td>
<td>CW, weight including water in free-flooding space, pounds</td>
</tr>
<tr>
<td>13</td>
<td>11-20</td>
<td>F10.5</td>
<td>CB, buoyancy, pounds</td>
</tr>
<tr>
<td>13</td>
<td>21-30</td>
<td>F10.5</td>
<td>UC, initial command speed, feet/second</td>
</tr>
<tr>
<td>13</td>
<td>31-40</td>
<td>F10.5</td>
<td>XB, x-component of center of buoyancy, feet</td>
</tr>
<tr>
<td>13</td>
<td>41-50</td>
<td>F10.5</td>
<td>ZB, z-component of center of buoyancy, feet</td>
</tr>
<tr>
<td>14</td>
<td>1-10</td>
<td>F10.5</td>
<td>DS, initial value of stern elevator position, radians</td>
</tr>
<tr>
<td>14</td>
<td>11-20</td>
<td>F10.5</td>
<td>DB, initial value of bow elevator position, radians</td>
</tr>
<tr>
<td>14</td>
<td>21-30</td>
<td>F10.5</td>
<td>RHO, density of sea water, slugs/feet³</td>
</tr>
<tr>
<td>14</td>
<td>31-40</td>
<td>F10.5</td>
<td>AL, submarine length, feet</td>
</tr>
<tr>
<td>14</td>
<td>41-50</td>
<td>F10.5</td>
<td>AM, submarine mass, including water in free flooding space, slugs</td>
</tr>
<tr>
<td>15</td>
<td>1-10</td>
<td>F10.5</td>
<td>ETAHI(γ-high), upper reference value of UC/UMAG, dimensionless</td>
</tr>
<tr>
<td>15</td>
<td>11-20</td>
<td>F10.5</td>
<td>ETAOL(γ-low), lower reference value of UC/UMAG, dimensionless</td>
</tr>
</tbody>
</table>
**TABLE 6. INPUT DATA DECK, PROGRAM ZC790 (cont.)**

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>21-30</td>
<td>F10.5</td>
<td>A11, value of a&lt;sub&gt;1&lt;/sub&gt; for ETA ETAHI</td>
</tr>
<tr>
<td>15</td>
<td>31-40</td>
<td>F10.5</td>
<td>A12, value of b&lt;sub&gt;1&lt;/sub&gt; for ETA ETAHI</td>
</tr>
<tr>
<td>15</td>
<td>41-50</td>
<td>F10.5</td>
<td>A13, value of c&lt;sub&gt;1&lt;/sub&gt; for ETA ETAHI</td>
</tr>
<tr>
<td>16</td>
<td>1-10</td>
<td>F10.5</td>
<td>A21, value of a&lt;sub&gt;1&lt;/sub&gt; for ETALO ETA ETAHI</td>
</tr>
<tr>
<td>16</td>
<td>11-20</td>
<td>F10.5</td>
<td>A22, value of b&lt;sub&gt;1&lt;/sub&gt; for ETALO ETA ETAHI</td>
</tr>
<tr>
<td>16</td>
<td>21-30</td>
<td>F10.5</td>
<td>A23, value of c&lt;sub&gt;1&lt;/sub&gt; for ETALO ETA ETAHI</td>
</tr>
<tr>
<td>16</td>
<td>31-40</td>
<td>F10.5</td>
<td>A31, value of a&lt;sub&gt;1&lt;/sub&gt; for ETALO ETA ETAHI</td>
</tr>
<tr>
<td>16</td>
<td>41-50</td>
<td>F10.5</td>
<td>A32, value of b&lt;sub&gt;1&lt;/sub&gt; for ETALO ETA ETAHI</td>
</tr>
<tr>
<td>16</td>
<td>51-60</td>
<td>F10.5</td>
<td>A33, value of c&lt;sub&gt;1&lt;/sub&gt; for ETALO ETA ETAHI</td>
</tr>
<tr>
<td>17</td>
<td>1-10</td>
<td>F10.5</td>
<td>XG, x-component of center of gravity, feet</td>
</tr>
<tr>
<td>17</td>
<td>11-20</td>
<td>F10.5</td>
<td>ZG, z-component of center of gravity, feet</td>
</tr>
</tbody>
</table>

**Surface Control Schedule**

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>1-10</td>
<td>F10.5</td>
<td>TIME, various periods of time depending on NS in control routine, seconds</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NS</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td>Not used</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>Initial steady state period (See figure 3)</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>Not used in input</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>Period of constant application of each command speed</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>Period of constant application of each command speed</td>
</tr>
<tr>
<td>18</td>
<td>11-20</td>
<td>F10.5</td>
<td>RL, negative rate of change of elevator position, radians/second during meander runs (NS = 1), radians/second (See figure 3)</td>
</tr>
</tbody>
</table>
**TABLE 6. INPUT DATA DECK, PROGRAM ZC790 (cont.)**

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>21-30</td>
<td>F10.5</td>
<td>DELTMA, maximum negative swing of elevator (must be more negative than DS, initial condition) during meander runs (NS = 1), radians (See figure 3)</td>
</tr>
<tr>
<td>18</td>
<td>31-40</td>
<td>F10.5</td>
<td>SWMAX (θ'), execute pitch angle, maximum submarine dive angle during meander type run (NS = 1) before turning elevators to a more positive value, radians</td>
</tr>
<tr>
<td>18</td>
<td>41-50</td>
<td>F10.5</td>
<td>R2, positive rate of change of elevator position after submarine reaches θ', radians/second (See figure 3)</td>
</tr>
<tr>
<td>18</td>
<td>51-60</td>
<td>F10.5</td>
<td>DELTMI, maximum position to which elevators are moved during 'R2' change, radians (See figure 3)</td>
</tr>
<tr>
<td>18</td>
<td>61-70</td>
<td>F10.5</td>
<td>DSF, final value of elevator position during DS - impulse run (NS = 2), radians.</td>
</tr>
</tbody>
</table>

**Additional Run Controls**

If IOPT was "0" on card 1 of the last data deck, card 32 can be either blank for a normal exit or card 1 of a new data deck for additional runs. If IOPT = F on card 1 of the last data deck, see below.

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>1-5</td>
<td>15</td>
<td>IRUN, run numbers</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IRUN = 0 (blank card) a new data deck is read. A blank card at the start of a data deck results in a normal exit so two blank cards at the end will always end the run.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IRUN = integer. Use as new run number and continue reading cards below.</td>
</tr>
<tr>
<td>20</td>
<td>1-5</td>
<td>15</td>
<td>NDEX, common location for parameter to be changed. Table 8 shows the variable names versus the the common parameter names.</td>
</tr>
</tbody>
</table>
TABLE 6. INPUT DATA DECK, PROGRAM ZC790 (cont.)

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>11-20</td>
<td>F10.5</td>
<td>VALUE, new value of parameter changed in columns 1-5 of this card.</td>
</tr>
<tr>
<td>20 + n</td>
<td>1-20</td>
<td>15, 5x, 10.5</td>
<td>Repeat card 20 as desired.</td>
</tr>
<tr>
<td>20 + n - 1</td>
<td>-</td>
<td>Blank</td>
<td>Start new run after all changes have been made</td>
</tr>
</tbody>
</table>

h. OUTPUT DATA

a. Printout - This program will print out titled, numbered pages. The first page provides the input data as shown in figure 10. Subsequent pages provide the variables noted in table 7.

TABLE 7. OUTPUT VARIABLES, PROGRAM ZC790

<table>
<thead>
<tr>
<th>Variable</th>
<th>Format</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>E13.6</td>
<td>feet/second</td>
<td>Velocity component in the x-direction</td>
</tr>
<tr>
<td>W</td>
<td>E13.6</td>
<td>feet/second</td>
<td>Velocity component in the y-direction</td>
</tr>
<tr>
<td>Q</td>
<td>E13.6</td>
<td>radians/seconds</td>
<td>Angular velocity about the y-axis</td>
</tr>
<tr>
<td>THETA</td>
<td>E13.6</td>
<td>radians</td>
<td>Submarine pitch angle</td>
</tr>
<tr>
<td>Z</td>
<td>E13.6</td>
<td>feet</td>
<td>Submarine depth</td>
</tr>
<tr>
<td>T</td>
<td>E13.6</td>
<td>seconds</td>
<td>Time</td>
</tr>
</tbody>
</table>

A sample data sheet is shown in figure 11. (Coefficients used in this run are trial or synthetic coefficients). This data is a function of the time of the run, T. The frequency of printout is a function of NPNT, printed each NPNTth integration cycle. For instance:

For the integration which used ICYC = 1, if \( H \), integration time period is 0.25 second and NPNT = 8

\[
T = \frac{NPNT \times H}{ICYC} = \frac{8 \times 0.25}{1} = 2 \text{ seconds} \quad (12)
\]

This printout would be at \( t = 0 \) and each two seconds thereafter.
TABLE 8. ZC 790, COMMON LAYOUT

<table>
<thead>
<tr>
<th>Location</th>
<th>Term</th>
<th>Location</th>
<th>Term</th>
<th>Location</th>
<th>Term</th>
<th>Location</th>
<th>Term</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>H</td>
<td>37</td>
<td>ZWAWE</td>
<td>73</td>
<td>A22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>N</td>
<td>38</td>
<td>ZDSE</td>
<td>74</td>
<td>A23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>ISL</td>
<td>39</td>
<td>AMQD</td>
<td>75</td>
<td>A31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>NPNT</td>
<td>40</td>
<td>AMQAS</td>
<td>76</td>
<td>A32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>TIIM</td>
<td>41</td>
<td>AMWD</td>
<td>77</td>
<td>A33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>RHQL2</td>
<td>42</td>
<td>AMQ</td>
<td>78</td>
<td>XG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>RHQL3</td>
<td>43</td>
<td>AMAQDS</td>
<td>79</td>
<td>ZG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>RHQL4</td>
<td>44</td>
<td>AMAWQ</td>
<td>80</td>
<td>ILOC(1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>RHQL5</td>
<td>45</td>
<td>AMSTR</td>
<td>81</td>
<td>(2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>WNB</td>
<td>46</td>
<td>AMW</td>
<td>82</td>
<td>(3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>ETA</td>
<td>47</td>
<td>AMWAQ</td>
<td>83</td>
<td>(4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>ETAMQ</td>
<td>48</td>
<td>AMW</td>
<td>84</td>
<td>(5)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>ISW2</td>
<td>49</td>
<td>AMNW</td>
<td>85</td>
<td>(6)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>XQQ</td>
<td>50</td>
<td>AMDS</td>
<td>86</td>
<td>(7)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>XUD</td>
<td>51</td>
<td>AMDB</td>
<td>87</td>
<td>(8)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>XWQ</td>
<td>52</td>
<td>AMQ</td>
<td>88</td>
<td>IPlot</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>XIU</td>
<td>53</td>
<td>AMNE</td>
<td>89</td>
<td>IRUN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>XWW</td>
<td>54</td>
<td>AMAWQ</td>
<td>90</td>
<td>IOPEN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>XDSDS</td>
<td>55</td>
<td>AMDSE</td>
<td>91</td>
<td>NPLT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>XDBDB</td>
<td>56</td>
<td>IY</td>
<td>92</td>
<td>IOPT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>XWWE</td>
<td>57</td>
<td>CW</td>
<td>93</td>
<td>Y(1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>XDSDE</td>
<td>58</td>
<td>CB</td>
<td>94</td>
<td>(2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>ZQD</td>
<td>59</td>
<td>UC</td>
<td>95</td>
<td>(3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>ZWD</td>
<td>60</td>
<td>XB</td>
<td>96</td>
<td>(4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>ZQ</td>
<td>61</td>
<td>ZB</td>
<td>97</td>
<td>(5)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>ZAQDS</td>
<td>62</td>
<td>DS</td>
<td>98</td>
<td>(6)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>ZWAQ</td>
<td>63</td>
<td>DB</td>
<td>99</td>
<td>TIME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>ZSTR</td>
<td>64</td>
<td>RHO</td>
<td>100</td>
<td>R1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>TW</td>
<td>65</td>
<td>AL</td>
<td>101</td>
<td>DELTMA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>ZWA</td>
<td>66</td>
<td>AM</td>
<td>102</td>
<td>SWMAX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>ZAW</td>
<td>67</td>
<td>ETAHI</td>
<td>103</td>
<td>R2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>ZNW</td>
<td>68</td>
<td>ETAHO</td>
<td>104</td>
<td>DELTMI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>ZDS</td>
<td>69</td>
<td>ALL</td>
<td>105</td>
<td>DSF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>ZDB</td>
<td>70</td>
<td>A12</td>
<td>106</td>
<td>NS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>ZQE</td>
<td>71</td>
<td>A13</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>ZWE</td>
<td>72</td>
<td>A21</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### SIMULINK SIMULATION, LONGITUDINAL FREEDOM

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Figure 10. Input Variable Format, ZC790
b. Plots - The program can optionally plot any of the printed values. In addition, the control parameter DS and DB (plane position, radians) can also be plotted. This output is identical to that of figure 7.

5. LISTINGS AND FLOW CHARTS

Appendix A contains the listings and flow chart for this program.
C. EC 470, INITIAL CONDITION COMPUTATION FOR SIMULATION

1. DESCRIPTION

In order to evaluate longitudinal performance of the submarine, a set of initial conditions for level flight is required. These initial conditions are a set of neutral angles defined as $\Theta$, steady state pitch angle; $DS$, sternplane angle; and $DB$, bow (sail) plane angle at a particular speed. This requires that the angle of attack equal the pitch angle and that the accelerations $\ddot{u}$, $\dot{w}$, and $\dot{\phi}$ equal zero. Reference is made to angle of attack in many texts, including the NSRDC reports but the parameter does not appear in the equations of motion. This is because a body axis system of coordinates is used instead of a flight path set. The angle of attack is the angle between the flight path and the body when roll angle is zero.

![Figure 12. Angle of Attack.](image)

The angle of attack (and in the case of steady level flight, the pitch angle) can be calculated from the relationship

$$
\sin \Theta = \sin \alpha = \frac{\omega_z}{U} - \frac{\omega}{\sqrt{\omega_x^2 + \omega_y^2}}
$$

as seen in figure 12. This removes the parameter $\Theta$ by replacing it with a function of $\omega$. For any speed, $U$, the equations of motion can be solved for level flight, in terms of $U$ and $\delta_z$ or $\delta_y$, by setting $\dot{w}$, $q$, $\dot{\phi}$, and all lateral terms to zero. Equation (3) is determined from Normal Equation (3) of the NSRDC Standard Equations and

$$
Z_{\omega_x} \omega_x^2 + Z_{\omega_y} \omega_y^2 + 2 \omega_y u \omega_y + Z_{\omega_x} \omega_x \omega - \frac{\omega}{\sqrt{\omega_x^2 + \omega_y^2}} = 0
$$

equation from the pitching moment equation (5).
The program utilizes all coefficients of interest so that it can be used with any set of coefficients applicable to the NSRDC equations. The pitch equation is solved for \( \delta \) with a trial value of \( \omega \), and this value is used in the normal equation via Newton's method. Theta is then calculated by equation (17).

\[ \theta = \tan^{-1} \frac{\omega}{u} \text{ radians} \quad (17) \]

This program serves another purpose in addition to determining the initial conditions for longitudinal runs. The values for \( \delta \) are determined for the operational speed range of the submarine and at some point the values become very large. This speed is known as the critical speed because at this speed the controls are ineffective in controlling the pitch attitude of the submarine. The critical speed points determined by this program can be checked against the data sources to insure effective training through the simulation.

2. Input Data Deck

The inputs to the program include the appropriate submarine coefficients, including choices of ZDS or ZDB and MDS or MDB for stern or sail-planes respectively. The other planes is considered to be set at zero and has no effect on the submarine trim. The submarine physical constants, and the speed of operation are needed also. These values are identified in detail with their locations on punch cards in table 9.

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1-50</td>
<td>SF10.5</td>
<td>ZWAW, ZWW, ZW, ZAW, ZSTR</td>
</tr>
<tr>
<td>1</td>
<td>51-60</td>
<td>F10.5</td>
<td>ZDEL, use either ZDS or ZDB as desired.</td>
</tr>
<tr>
<td>2</td>
<td>1-50</td>
<td>SF10.5</td>
<td>MWAW, MW, MW, MAW, MSTR</td>
</tr>
<tr>
<td>2</td>
<td>51-60</td>
<td>F10.5</td>
<td>MDEL, use either MDS or</td>
</tr>
</tbody>
</table>
TABLE 9. INPUT DATA DECK, PROGRAM ECL70 (cont.)

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1-10</td>
<td>F10.5</td>
<td>MDB in accord with selection of ZDEL above.</td>
</tr>
<tr>
<td>3</td>
<td>11-20</td>
<td>F10.5</td>
<td>B, buoyancy, pounds</td>
</tr>
<tr>
<td>3</td>
<td>21-30</td>
<td>F10.5</td>
<td>AB, z-component of center of buoyancy location, feet</td>
</tr>
<tr>
<td>3</td>
<td>31-40</td>
<td>F10.5</td>
<td>RHO, density of sea water, slugs/ft$^3$</td>
</tr>
<tr>
<td>4</td>
<td>1-10</td>
<td>F10.5</td>
<td>AL, submarine length, feet</td>
</tr>
<tr>
<td>4</td>
<td>11-20</td>
<td>F10.5</td>
<td>U, submarine forward velocity, feet/second. This should be highest speed desired.</td>
</tr>
<tr>
<td>4</td>
<td>21-30</td>
<td>F10.5</td>
<td>WZERO, trial value of as required in solution of problem. This initial guess should be close; 1st approximation is .002 U radians (U in feet/second).</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>-</td>
<td>ULIM, submarine forward velocity, feet/second. This is the slowest speed desired. The program will calculate initial conditions from U thru ULIM in one (1) knot (1.689 feet/second) intervals.</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>-</td>
<td>Blank card for normal end of job</td>
</tr>
</tbody>
</table>

3. OUTPUT DATA

Figure 13 is a sample output data sheet using trial or synthetic coefficients. The output is presented in rows of data at one knot intervals in speed from the highest requested speed, U to the lowest, ULIM. The following parameters are printed:

a. Input data - All coefficients and submarine physical constants as provided in the first three input data cards. Units are same as those for the input data.

b. Output data

U - forward speed, feet/second
U - forward speed, knots
W - Normal speed, feet/second
| TIME | $V_{10}$ | $V_{11}$ | $V_{12}$ | $V_{13}$ | $V_{14}$ | $V_{15}$ | $V_{16}$ | $V_{17}$ | $V_{18}$ | $V_{19}$ | $V_{20}$ | $V_{21}$ | $V_{22}$ | $V_{23}$ | $V_{24}$ | $V_{25}$ | $V_{26}$ | $V_{27}$ |
|------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| 0.1  | 0.537740 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 |
| 0.2  | 0.415740 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 |
| 0.3  | 0.298200 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 |
| 0.4  | 0.261300 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 |
| 0.5  | 0.224200 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 |
| 0.6  | 0.186700 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 |
| 0.7  | 0.148700 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 |
| 0.8  | 0.110700 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 |
| 0.9  | 0.072200 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 |
| 1.0  | 0.033200 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 | 0.216040 |

Figure 13. Output Variable Format, ECH70
DEL - DS or DB, depending on the selection of the coefficients, ADS and MDS or ZDB and MDB respectively, radians.

THETA = θ, submarine pitch angle, radians
DEL = DS or DB, degrees
THETA = θ, degrees

4. LISTINGS AND FLOW CHART

The listings and flow chart for program EC470 are included in appendix A.
D. **EQUIP30, CENTER OF GRAVITY COMPUTATION**

1. **DESCRIPTION**

This program calculates the new location of the center of gravity, in three planes, due to the flooding of any combination of tanks on a submarine.

The program solves equations stated below, and points out the new total weight, three components of center of gravity position, and a record of the tanks filled.

\[
X_G = \frac{W_G + \sum W_i x_i}{W_G + \sum W_i}
\]

\[
Y_G = \frac{W_G + \sum W_i y_i}{W_G + \sum W_i}
\]

\[
Z_G = \frac{W_G + \sum W_i z_i}{W_G + \sum W_i}
\]

where

- \( X_G \) = x - component of center of gravity location, feet
- \( Y_G \) = y - component of center of gravity location, feet
- \( Z_G \) = z - component of center of gravity location, feet
- \( W_i \) = weight of submarine, pounds
- \( X_1 \) = x - component of center of gravity basic submarine, feet
- \( Y_1 \) = y - component of center of gravity basic submarine, feet
- \( Z_1 \) = z - component of center of gravity basic submarine, feet
- \( W_i \) = weight of water in \( i \)th tank, pounds
- \( X_i \) = x - component of center of gravity of \( i \)th tank, feet
- \( Y_i \) = y - component of center of gravity of \( i \)th tank, feet
- \( Z_i \) = z - component of center of gravity of \( i \)th tank, feet
- \( W = W_0 + \sum W_i \), pounds
- \( i \) = integer from 2 to 50, depending on number of tanks considered

2. **INPUT DATA DECK**

The input variables to the program include the numbers of particular
NAVTRADEVCEN 68-C-0050-2

tanks, weight of water when full, three components of center of gravity of each tank when full, and particular control variables. These inputs are described in detail with their locations on punch cards in Table 10.

**Table 10. Input Data Deck, Program ECL30**

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1-5</td>
<td>I5</td>
<td>N, number of tanks (including one (1) for submarine, considered as a tank)</td>
</tr>
<tr>
<td>1</td>
<td>6-10</td>
<td>I5</td>
<td>IPNT controls printout data IPNT = 1 - Input data printed IPNT = 0 - Input data not printed</td>
</tr>
<tr>
<td>2</td>
<td>1-10</td>
<td>F10.5</td>
<td>WI, W1, weight of water in tanks, pounds. First weight, W1, is weight of submarine</td>
</tr>
<tr>
<td>2</td>
<td>11-20</td>
<td>F10.5</td>
<td>XI, X1, x-component of center of gravity, pounds. First value, X1, is for submarine (usually = 0).</td>
</tr>
<tr>
<td>2</td>
<td>21-30</td>
<td>F10.5</td>
<td>YI, Y1, y-component of center of gravity, pounds. First value, Y1, is for submarine (usually = 0).</td>
</tr>
<tr>
<td>2</td>
<td>31-40</td>
<td>F10.5</td>
<td>ZI, Z1, z-component of center of gravity, pounds. First value, Z1, is for submarine (usually = 0).</td>
</tr>
<tr>
<td>2</td>
<td>41-50</td>
<td>F10.5</td>
<td>WI, W2, weight of water in #2 tank.</td>
</tr>
<tr>
<td>2</td>
<td>51-60</td>
<td>F10.5</td>
<td>XI, X2 = X(2), x-component of C.O. for #2 tank</td>
</tr>
<tr>
<td>2</td>
<td>61-70</td>
<td>F10.5</td>
<td>YI, Y2 = Y(2), y-component of C.O. for #2 tank</td>
</tr>
<tr>
<td>2</td>
<td>71-80</td>
<td>F10.5</td>
<td>ZI, Z2 = Z(2), z-component of C.O. for #2 tank</td>
</tr>
<tr>
<td>3-n</td>
<td>1-80</td>
<td>8F10.5</td>
<td>Repeat card number two (2) until N weights and C.O. components have been entered, one set for each tank. Maximum number of tanks (sets) is 62</td>
</tr>
</tbody>
</table>
TABLE 10. INPUT DATA DECK, PROGRAM EC430 (cont.)

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>n+1</td>
<td>1-n</td>
<td>nIl</td>
<td>fifty (50), ICTL(I) = 1, N. This single array represents all tanks to be considered, #1 (the submarine itself) thru N (the largest number of tanks up thru 50), ICTL(I) = 0, ith tank is empty, ICTL(I) = 1, ith tank is full and the W, X, Y, and Z values of this tank will be included in the calculations.</td>
</tr>
<tr>
<td>M+2-m</td>
<td></td>
<td></td>
<td>Stack as many of these cards as desired, one card for each run</td>
</tr>
<tr>
<td>m+1</td>
<td>1</td>
<td>d</td>
<td>9, this is normal end of job card.</td>
</tr>
</tbody>
</table>

3. OUTPUT DATA

A sample of the output data sheets is shown in figure 11. This sample is in two pages, numbered sequentially and each identified by the program number and title.

Page one shows the input data, N, W, X, Y, and Z, number of the tank, weight of water of the tank and the x, y, and z coordinates of the center of gravity of each full tank.

Page two contains two sets of data. The first includes a listing of all tanks considered from 1 thru N. Below each tank number is a second digit which is "1" if the tank is filled and the weights and moment arms for that tank are considered in the calculations. This number is "0" if the tank is empty.

4. LISTINGS AND FLOW CHART

The listings and flow chart for program EC430 are included in appendix A.
<table>
<thead>
<tr>
<th>i</th>
<th>x</th>
<th>y</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.239702</td>
<td>0.000000</td>
<td>0.2</td>
</tr>
<tr>
<td>2</td>
<td>0.790007</td>
<td>0.000000</td>
<td>0.2</td>
</tr>
<tr>
<td>3</td>
<td>0.290000</td>
<td>0.000000</td>
<td>0.2</td>
</tr>
<tr>
<td>4</td>
<td>0.790000</td>
<td>0.000000</td>
<td>0.2</td>
</tr>
<tr>
<td>5</td>
<td>0.290000</td>
<td>0.000000</td>
<td>0.2</td>
</tr>
<tr>
<td>6</td>
<td>0.390000</td>
<td>0.000000</td>
<td>0.2</td>
</tr>
<tr>
<td>7</td>
<td>0.490000</td>
<td>0.000000</td>
<td>0.2</td>
</tr>
<tr>
<td>8</td>
<td>0.590000</td>
<td>0.000000</td>
<td>0.2</td>
</tr>
<tr>
<td>9</td>
<td>0.690000</td>
<td>0.000000</td>
<td>0.2</td>
</tr>
<tr>
<td>10</td>
<td>0.790000</td>
<td>0.000000</td>
<td>0.2</td>
</tr>
</tbody>
</table>

Figure 1h. Output Variable Format, ECL30
<table>
<thead>
<tr>
<th>KG</th>
<th>YG</th>
<th>ZG</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.000000E 07</td>
</tr>
<tr>
<td>1 2 3 4 5 6 7 8 9 10 11 12 13 14 15</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>KG</th>
<th>YG</th>
<th>ZG</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.176684E 01</td>
<td>0.0</td>
<td>0.0101490F 00</td>
<td>0.106020E 08</td>
</tr>
<tr>
<td>1 2 3 4 5 6 7 8 9 10 11 12 13 14 15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 1 1 1 0 0 1 1 0 1 0 0 1 1 1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>KG</th>
<th>YG</th>
<th>ZG</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.196244F 01</td>
<td>0.0</td>
<td>-0.795070E-01</td>
<td>0.105160E 08</td>
</tr>
<tr>
<td>1 2 3 4 5 6 7 8 9 10 11 12 13 14 15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 1 1 1 0 0 1 1 0 1 1 0 1 1 1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>KG</th>
<th>YG</th>
<th>ZG</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.269755F 01</td>
<td>0.0</td>
<td>-0.728146E-01</td>
<td>0.106160E 08</td>
</tr>
<tr>
<td>1 2 3 4 5 6 7 8 9 10 11 12 13 14 15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 1 1 1 0 0 1 1 0 1 0 0 1 1 1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>KG</th>
<th>YG</th>
<th>ZG</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.119837F 01</td>
<td>0.0</td>
<td>-0.728146E-01</td>
<td>0.104160E 08</td>
</tr>
<tr>
<td>1 2 3 4 5 6 7 8 9 10 11 12 13 14 15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 0 1 1 0 0 1 1 0 1 0 1 1 1 1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>KG</th>
<th>YG</th>
<th>ZG</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.102222F 01</td>
<td>0.0</td>
<td>-0.398103E-01</td>
<td>0.104160E 08</td>
</tr>
<tr>
<td>1 2 3 4 5 6 7 8 9 10 11 12 13 14 15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 0 0 1 0 0 1 1 0 1 1 1 1 1 1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>KG</th>
<th>YG</th>
<th>ZG</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.173953F 01</td>
<td>0.0</td>
<td>0.426318E-01</td>
<td>0.100160E 08</td>
</tr>
<tr>
<td>1 2 3 4 5 6 7 8 9 10 11 12 13 14 15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 0 0 0 0 0 1 1 0 1 1 0 1 1 1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>KG</th>
<th>YG</th>
<th>ZG</th>
<th>W</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.114180F 01</td>
<td>0.0</td>
<td>0.196114F 00</td>
<td>0.931100E 07</td>
</tr>
</tbody>
</table>

Figure 14. Output Variable Format, ECH30 (cont.)
E. ZC 300, SUBMARINE THRUST

1. DESCRIPTION

This program solves equation 18 for axial thrust or force, \( m \ddot{u} \), and axial acceleration \( \ddot{u} \) :

\[
m \ddot{u} = \frac{\rho}{2} l^2 [a \cdot u^2 + b \cdot u u_c + c \cdot u_c^2]
\]

- \( \ddot{u} \) = axial acceleration, (ft/sec²)
- \( m \ddot{u} \) = force, (lb)
- \( m \) = mass, (slug 16-sec²) 
  \[ \frac{32.2}{\text{ft}} \]
- \( \rho \) = density, (1.995 slug/ft³)
- \( l \) = length, (ft)
- \( a_i, b_i, c_i \) = dimensionless constants depending on value of \( \gamma \), \( (U_c / U) \)
- \( U \) = submarine axial velocity, (ft/sec)
- \( U_c \) = command speed, (ft/sec)

Three sets of constants are used for \( a_i, b_i, \) and \( c_i \) depending on the value of \( \gamma \) as compared to \( \gamma \) high and \( \gamma \) low in equation 19.

\[
a_1 \ b_1 \ c_1 \quad \gamma > \gamma_{\text{high}} \\
\quad a_2 \ b_2 \ c_2 \quad \gamma_{\text{low}} \leq \gamma \leq \gamma_{\text{high}} \\
\quad a_3 \ b_3 \ c_3 \quad \gamma < \gamma_{\text{low}}
\]

At \( u = 0 \) equation 20 is used.

\[
\begin{align*}
\text{if } U_c > 0 & \quad \gamma = 1 \\
\text{if } U_c < 0 & \quad \gamma = -1
\end{align*}
\]

2. INPUT DATA DECK

The input variables are submarine length, mass, \( \gamma \) high, \( \gamma \) low, and the three values of \( a_i, b_i, \) and \( c_i \). They are punched on cards according to table 11.
### TABLE 11. INPUT DATA DECK, PROGRAM ZC300

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1-10</td>
<td>F10.3</td>
<td>AL, submarine length</td>
</tr>
<tr>
<td></td>
<td>11-20</td>
<td>F10.3</td>
<td>AM, submarine mass</td>
</tr>
<tr>
<td></td>
<td>21-30</td>
<td>F10.3</td>
<td>ETAHI, upper value of for coefficient change</td>
</tr>
<tr>
<td></td>
<td>31-40</td>
<td>F10.3</td>
<td>ETALO, lower value of for coefficient change</td>
</tr>
<tr>
<td>2</td>
<td>1-10</td>
<td>F10.6</td>
<td>A1</td>
</tr>
<tr>
<td></td>
<td>11-20</td>
<td>F10.6</td>
<td>A2</td>
</tr>
<tr>
<td></td>
<td>21-30</td>
<td>F10.6</td>
<td>A3</td>
</tr>
<tr>
<td>3</td>
<td>1-10</td>
<td>F10.6</td>
<td>B1</td>
</tr>
<tr>
<td></td>
<td>11-20</td>
<td>F10.6</td>
<td>B2</td>
</tr>
<tr>
<td></td>
<td>21-30</td>
<td>F10.5</td>
<td>B3</td>
</tr>
<tr>
<td>4</td>
<td>1-10</td>
<td>F10.6</td>
<td>C1</td>
</tr>
<tr>
<td></td>
<td>11-20</td>
<td>F10.6</td>
<td>C2</td>
</tr>
<tr>
<td></td>
<td>21-30</td>
<td>F10.6</td>
<td>C3</td>
</tr>
</tbody>
</table>

3. OUTPUT DATA

The output includes

a. All input variables

b. Table of force (thrust) as a function of command speed from -15 to +30 knots (\(\Delta\) 5 knots) and for values of \(U\) from 0 to 30 knots (\(\Delta\) 2.5 knots)

c. Table of acceleration as function of same velocity variables as b. above.

4. LISTINGS AND FLOW CHARTS

The listings and flow chart for program ZC300 are included in appendix I.
F. ZC690, ERROR CALCULATOR, DS * DR CONTROL

1. DESCRIPTION

This program calculates the percent error of change in a variable. Inputs are from EB920, Submarine Simulation Program - U at 60 seconds, theta high, phi mini, phi (peak near T = max), and psi at 60 seconds. The program calculates the change of each variable from value at t = 0 for the original program coefficients. Then a coefficient is set to zero and the changes recalculated and compared to the reference run as

\[
\% \text{ change } (U) = 100 \left( \frac{\Delta U_{ref} - \Delta U}{\Delta U_{ref}} \right)
\]

These calculations are printed out with all input data for 5, 15, and 25 knots. Figure 15 shows the criteria used for comparison for each of the variables.

2. INPUT DATA DECK

Table 12 gives the format for the input data deck for this program.

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1-4</td>
<td>I4</td>
<td>NO01, 1st run at 15 knots</td>
</tr>
<tr>
<td></td>
<td>11-14</td>
<td>I4</td>
<td>NO25, 1st run at 25 knots</td>
</tr>
<tr>
<td>2</td>
<td>1-4</td>
<td>I4</td>
<td>Run number, NO</td>
</tr>
<tr>
<td></td>
<td>11-18</td>
<td>2A4</td>
<td>Coefficient set to zero, COEF</td>
</tr>
<tr>
<td></td>
<td>21-30</td>
<td>F10.5</td>
<td>Value of u at 60 sec, U60</td>
</tr>
<tr>
<td></td>
<td>31-40</td>
<td>F10.6</td>
<td>Peak value of theta, THETHI</td>
</tr>
<tr>
<td></td>
<td>41-50</td>
<td>F10.8</td>
<td>Minimum value of phi, PHIMIN</td>
</tr>
<tr>
<td></td>
<td>51-60</td>
<td>F10.8</td>
<td>Peak value of last oscillation of phi, PHIUP</td>
</tr>
<tr>
<td></td>
<td>61-70</td>
<td>F10.6</td>
<td>Value of psi at 60 sec., PSI60.</td>
</tr>
<tr>
<td>2-N</td>
<td></td>
<td></td>
<td>Repeat above card as desired</td>
</tr>
<tr>
<td>N</td>
<td>Blank card for end of data</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Criteria for Comparison

\[
\% \text{ change} = 100 \cdot \frac{\Delta u(t) - \Delta u}{\Delta u(t)}
\]

\[
\% \text{ change} = 100 \cdot \frac{\Delta \theta_{\text{ref}} - \Delta \theta}{\Delta \theta_{\text{ref}}}
\]

\[
\% \text{ change} = 100 \cdot \frac{\Delta \phi_{\text{ref}} - \Delta \phi}{\Delta \phi_{\text{ref}}}
\]

\[
\% \text{ change} = 100 \cdot \frac{\Delta \psi_{\text{ref}} - \Delta \psi}{\Delta \psi_{\text{ref}}}
\]

Figure 15. Comparison Criteria for Error Calculator
3. OUTPUT DATA

The output includes:

a. Percent change in (see figure 15):
   1. $\Delta u(60)$
   2. $\Delta \theta(\text{peak})$
   3. $\Delta \phi(\text{min})$
   4. $\Delta \phi(\text{lobe})$
   5. $\Delta \psi(60)$

b. All input variables

4. LISTINGS AND FLOW CHART

The listings and flow chart for program Z0690 are given in appendix A.
NAVTRADEVCEN 68-G-0050-2

G. 40691, ERROR CALCULATOR, DS CONTROL

1. DESCRIPTION

This program calculates the percent error or change in a variable compared to change of a reference. Inputs are from EB920, submarine simulation program. Reference data is data for \( u, \theta, \) and \( z \) from EB920 for original coefficients at 5, 15, and/or 25 knots. Comparable data, (see figure 15) is taken from the output of EB920 for setting a coefficient (COEF) to zero, or varying values of XG and ZG for center of gravity studies. The program calculates values such as:

\[
\% \text{ change (u)} = 100 \left( \frac{\Delta u_{\text{ref}} - \Delta u}{\Delta u_{\text{ref}}} \right)
\]

This program differs from 40690 in that a new reference run can be used at each comparison rather than only one reference at the start. The reference run can be compared to either a single run (variable ISW2 = 1 in the program) or to a group of runs (ISW2 = 0).

2. INPUT DATA DECK

Table 13 gives the format of the input data deck for this program.

<table>
<thead>
<tr>
<th>Table 13. INPUT DATA DECK, PROGRAM 40691</th>
</tr>
</thead>
<tbody>
<tr>
<td>Card</td>
</tr>
<tr>
<td>------</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
### TABLE 13. INPUT DATA DECK, PROGRAM ZC691 (cont.)

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>51-60</td>
<td>F10</td>
<td>Xg (if used), XG</td>
</tr>
<tr>
<td></td>
<td>61-70</td>
<td>F10</td>
<td>Zg (if used), ZG</td>
</tr>
<tr>
<td>3-n</td>
<td></td>
<td></td>
<td>Repeat card No. 2 as desired for each comparison run</td>
</tr>
<tr>
<td>n+1</td>
<td></td>
<td></td>
<td>New reference card for new (15 knot or 25 knot) speed range</td>
</tr>
<tr>
<td>n-m</td>
<td></td>
<td></td>
<td>Repeat card No. 2 as desired for each comparison run</td>
</tr>
<tr>
<td>m-k</td>
<td></td>
<td></td>
<td>Repeat cards n+1 and n-M for 25 knot speed range</td>
</tr>
<tr>
<td>k+1</td>
<td></td>
<td></td>
<td>Blank card</td>
</tr>
</tbody>
</table>

Table 14 and 15 give examples of the two types of input decks that can be used.

### TABLE 14. DATA SUBMITTAL FOR ISW2 = 0, ZC691

<table>
<thead>
<tr>
<th>Card</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Speed change run NO's. Comparison switch (ISW2 = 0)</td>
</tr>
<tr>
<td>2</td>
<td>Reference run data</td>
</tr>
<tr>
<td>3</td>
<td>Comparison run data</td>
</tr>
<tr>
<td>4-n</td>
<td>Repeat card 3 as required</td>
</tr>
<tr>
<td>n+1</td>
<td>New ref run data at next speed range</td>
</tr>
<tr>
<td>n+2</td>
<td>Comparison run data</td>
</tr>
<tr>
<td>n-m</td>
<td>Repeat card n+2 as required, etc.</td>
</tr>
</tbody>
</table>
TABLE 15. DATA SUBMITTAL FOR ISW2 = 1, EC961

<table>
<thead>
<tr>
<th>Card</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Speed change run numbers. Comparison switch (ISW2 = 1)</td>
</tr>
<tr>
<td>2</td>
<td>Reference run data</td>
</tr>
<tr>
<td>3</td>
<td>Comparison run data</td>
</tr>
<tr>
<td>4</td>
<td>Reference run data</td>
</tr>
<tr>
<td>5</td>
<td>Comparison run data</td>
</tr>
<tr>
<td>6-n</td>
<td>Repeat card 2 as required right through speed changes</td>
</tr>
<tr>
<td>7-n+1</td>
<td>Repeat card 3 as required right through speed changes</td>
</tr>
</tbody>
</table>

3. OUTPUT DATA
The output includes
a. Percent change in
   1. $\Delta u$
   2. $\Delta \theta$
   3. $\Delta z$

b. All input quantities and initial conditions

4. LISTINGS AND FLOW CHART
The listings and flow chart for program EC961 are given in Appendix A.
H. ECU40, ROOT CRACKER PROGRAM, LONGITUDINAL

1. DESCRIPTION

This program is used to help analyze submarine elevator impulse response for natural frequencies and damping factors. The program solves the roots of the characteristic equation which in turn was solved from a matrix of the longitudinal force terms of the equations of motion.

These roots are equivalent to the values $\alpha$, $\beta$, and $\gamma$ of equation (21).

$$y = a_1 e^{-\alpha t} + a_2 e^{-\beta t} \cos(\beta t + \gamma)$$

which is the longitudinal time output response to an impulse input. The variables $a_1$, $a_2$, and $\gamma$ (as well as an additional estimate of $\alpha$, $\beta$, and $\gamma$) can be estimated from program EC330, Time Response Coefficient Estimator, Longitudinal. Then the frequency response to a particular impulse as described by equation (21) can be calculated and plotted in program EC310, Brown's Convergence and Comparative Plot Program, against an actual impulse plot made from an EB920, Submarine Simulation Program run.

Equation (22) shows the matrix formed from the longitudinal terms of the equations of motion. The assumptions and modifications that were used to simplify the matrix are noted. The values of $\bar{w}$ and $\bar{q}$ are average values of these terms as taken from a number of actual full set runs of a submarine at steady state conditions on program EB920, Submarine Simulation Program. These values are used as a compensation for the non-linear terms in the longitudinal loop by modifying the basic coefficients. For example,

$$Z_{\omega|q|} \text{ replace } Z_{\omega|q|} \rightarrow Z_{\omega|q|} \text{ with } \bar{q}$$

where $\bar{q}$ is an average value of $|q|$ used as a constant.

Several of the coefficients shown in the matrix of equation (22) are combined inside this program to provide more simple matrix elements. These are used in the actual matrix solved and are printed out for convenience. These new values of combined coefficients will be shown as primes here (are printed without primes) such as

$$M_{\omega}' = M_{\omega} + \frac{M_{\omega|q|} \bar{w}}{\bar{u}}$$
### Matrix in \( S \), Longitudinal Case, (22)

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( P )</td>
<td>( w )</td>
<td>( u )</td>
<td>( \Delta )</td>
</tr>
<tr>
<td>( \Phi )</td>
<td>( \theta )</td>
<td>( \phi )</td>
<td>( \phi )</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\text{From "Standard Equations of Motion for Submarine Simulation"} \\
X_{uu} & = 0 \\
X_{ww} & = 0 \\
X_{ww} & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
W & = 0 \\
The matrix element, \((1, 2)\), equation (22), is then carried to the matrix as

\[
S^2 \left( \frac{1}{2} l^3 M_{\omega} \right) + S \left( \frac{\rho}{\omega} l^3 M_{\omega}' \right)
\]

This same simplification is carried out for the other modified coefficients

\[
M_{\tilde{b}} = M_{\tilde{b}} + \frac{M_{\omega} l^3 \bar{\omega}}{u}
\]

and

\[
Z_{\omega} = Z_{\omega} + \frac{l^2 M_{\omega} l^3 \bar{\omega}}{u} + \frac{Z_{\omega} l^4 \bar{\omega}}{u}
\]

The characteristic equation resulting from the expansion of equation (22) is a fourth-order equation which has four roots, \(\lambda \pm j\beta\), \(\sigma \pm j\theta\). The characteristic equation results in the following roots:

\[
\lambda + j\beta, \quad \lambda - j\beta, \quad \sigma + j\theta, \quad \sigma + j\theta
\]

\(\lambda\) and \(\beta\) are the real and imaginary components of the complex roots. \(\sigma\) is the larger of the two real roots, \(\theta\) is the smaller. Only three roots are used in the analysis of longitudinal natural frequencies and damping factor. The fourth root (smallest real magnitude) has a very small coefficient in the time domain and represents a long term effect due to changes in \(U\) (forward speed). This term can thus be ignored when considering pitch motions, but the other roots are more accurate if this term is included in the matrix.

Certain measurement parameters are calculated from the roots by this program. They are the natural frequency

\[
\omega_n = \sqrt{\lambda^2 + \beta^2}
\]

the time to damp to one-half

\[
T_{1/2} = \frac{\ln 2}{\lambda}
\]

the period

\[
P = \frac{2\pi}{\beta}
\]
and the damping ratio

$$\zeta = \frac{\alpha}{\omega_n} = \frac{\alpha}{\sqrt{\alpha^2 + \beta^2}}$$

2. SUBROUTINE DESCRIPTIONS

a. PLACE - This subroutine places each set of elements of the matrix into the matrix so that the characteristic equation can be expanded.

b. CHREQN - This subroutine expands the matrix formed by subroutine PLACE into the characteristic equation by taking the determinant of the matrix.

c. MULLER - This subroutine, by the method of Muller, calculates the roots of the characteristic equation.

3. INPUT DATA DECK

The input variables to the program include the coefficients and submarine constants noted in the matrix of equation (22). They are identified in detail with their locations in table 16.

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1-80</td>
<td>8F10.5</td>
<td>AWD, ZW, ZQD, ZQ, MWD, MW, MQD, MQ</td>
</tr>
<tr>
<td>2</td>
<td>1-10</td>
<td>F10.5</td>
<td>AW, \bar{w}, average value of normal component of submarine velocity based on a number of full set runs, feet/second</td>
</tr>
<tr>
<td>2</td>
<td>11-20</td>
<td>F10.5</td>
<td>AQ, \bar{q}, average pitch rate, radians/second</td>
</tr>
<tr>
<td>2</td>
<td>21-60</td>
<td>4F10.5</td>
<td>AQ, \bar{q}, average pitch rate, radians/second</td>
</tr>
<tr>
<td>3</td>
<td>1-10</td>
<td>F10.5</td>
<td>XUD</td>
</tr>
<tr>
<td>3</td>
<td>11-20</td>
<td>F10.5</td>
<td>All, a_1, thrust coefficient</td>
</tr>
<tr>
<td>3</td>
<td>21-30</td>
<td>F10.5</td>
<td>A12, b_1, thrust coefficient</td>
</tr>
<tr>
<td>3</td>
<td>31-70</td>
<td>4F10.5</td>
<td>ZSTR, MSTR, XQ, XWQ</td>
</tr>
<tr>
<td>4</td>
<td>1-10</td>
<td>F10.5</td>
<td>L, submarine length, feet</td>
</tr>
</tbody>
</table>
TABLE 16. INPUT DATA DECK, PROGRAM ECI40 (cont.)

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>11-20</td>
<td>F10.5</td>
<td>M, submarine mass, slugs</td>
</tr>
<tr>
<td>4</td>
<td>21-30</td>
<td>F10.5</td>
<td>II, moment of inertia about the y-axis, slug-ft²</td>
</tr>
<tr>
<td>4</td>
<td>31-40</td>
<td>F10.5</td>
<td>B, submarine buoyancy, pounds</td>
</tr>
<tr>
<td>4</td>
<td>41-50</td>
<td>F10.5</td>
<td>ZB, z component of center of buoyancy, feet</td>
</tr>
<tr>
<td>5</td>
<td>51-60</td>
<td>F10.5</td>
<td>RHO, density of seawater, slugs/ft³</td>
</tr>
<tr>
<td></td>
<td>1-10</td>
<td>F10.5</td>
<td>U, u, forward velocity, knots</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Stack as many U cards as desired. Place a blank card after the last U card for that set. Repeat all above cards as desired for a new case with different coefficients.</td>
</tr>
<tr>
<td>last</td>
<td>1-4</td>
<td>F4.3</td>
<td>999. The program takes normal end of job with this card.</td>
</tr>
</tbody>
</table>

4. OUTPUT DATA

Figure 16 shows a typical output sheet from this program.

a. Input data - all input data is printed out in the units described above.

b. The factors A1, A2, etc., of the characteristic equation are printed out. For this 4th order equation

\[ A_1 s^4 + A_2 s^3 + A_3 s^2 + A_4 s + A_5 = 0 \]

c. Roots - The four roots of the characteristic equation are printed out.

d. Measurement Parameters - The natural frequency, time to damp to one-half, period and the damping ratio are printed out.

5. LISTINGS AND FLOW CHART

The listings and flow chart for program ECI40 are given in Appendix A.
I. EC320 ROOT CRACKER PROGRAM, LATERAL

1. DESCRIPTION

This program is used to help analyze submarine rudder impulse response for natural frequencies and damping factors. The program solves for the roots of the characteristics equation which in turn was solved from a matrix of the lateral force terms of the equations of motion.

These roots are equivalent to the values $\zeta$, $\beta$, $\sigma$, and $\delta$ of the equation below:

$$\ddot{y} = a_1 e^{-\sigma t} + a_2 e^{-\beta t} \cos(\sigma t + \delta) + a_3 e^{-\zeta t}$$

which is the lateral time output response to an impulse input. The variables $a_1$, $a_2$, $a_3$, and $\delta$ can be estimated from program EC150, Time Response Coefficient Estimator, Lateral. Then the frequency response to a particular impulse, can be calculated and plotted in program EC310, Browns Convergency and Comparative Plot Program, against an actual impulse plot made from an EB920, Submarine Simulation Program run.

Equation (23) shows the matrix formed from the lateral terms of the equations of motion. The assumptions and modifications that were used to simplify the matrix are noted. The values of $p$, $P$, and $\Psi$ are average values of these terms as taken from a number of actual full set runs of the submarine at steady state conditions on program EB920, Submarine Simulation Program. Figure 17 qualitatively shows the errors inherent in the substitution of one of the values, say $P$ for $|P|$. These values are used as a compensation for the non-linear terms in the lateral loop by modifying the basic coefficients. For example,

$$Y_{\omega r \omega l} \rightarrow \frac{Y_{\omega r \omega l}}{|\omega r|} \frac{1}{|P|}$$

where $\bar{\omega}$ is an average value of $|\omega r|$ used as a constant.

$Y_{\omega r}$ and $Y_{\omega r \omega l}$ are combined into a single term in the matrix. A similar process is used for $|P|$ and $|\omega l|$, set to $P$ and $P$ respectively.

Several of the coefficients shown in the matrix are combined within this program to provide more simple matrix elements. These are used in the actual matrix solved and are printed out for convenience. These new values of combined coefficients will be shown as primes here (are printed without primes) such as:

$$Y'_{\omega r} = Y_{\omega r} + \frac{Y_{\omega r \omega l} \bar{\omega} + l Y_{\omega r |r|} P}{u}$$

80
<table>
<thead>
<tr>
<th>Matrix in $S$, Lateral Case (23)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s(\xi^2 Y_H - m) + \xi^2 Y_{\text{min}}$</td>
</tr>
<tr>
<td>$+ \xi^2 Y_H u_0 + \xi^2 Y_{\text{min}}$</td>
</tr>
<tr>
<td>$s^2(\xi^2 K_H) +$</td>
</tr>
<tr>
<td>+ $\frac{1}{2} \xi^2 (K_H u_0 + K_{\text{min}} u_0)$</td>
</tr>
<tr>
<td>$s(\xi^2 N_H) + \xi^2 N_{\text{min}} u_0$</td>
</tr>
<tr>
<td>$+ \xi^2 N_{\text{min}} u_0$</td>
</tr>
</tbody>
</table>

From "Standard Equations of Motion for Submarine Simulation"

$x_0 = x_0 = y_n = z_0 = y_0 = 0$
$\dot{x} = \dot{y} = 0$
$w = 0$
$\delta_r = \delta_0 = \delta_0 = 0$

All $\eta$-terms = 0

$Y_{\text{min}} = y_n = k_y = N_{\text{min}} = 0$
$u_r = u_{0r}$
$u_p = u_{0p}$
$\psi = \gamma_{0r}$
$u_{\delta} = u_{0\delta}$
$u_{\theta} = u_{0\theta}$
$\gamma_{0r}$
Figure 17. Error in Use of $\tilde{F}$
The matrix element, \((1,1)\) equation (1), is then carried to the matrix as
\[
\frac{\partial}{\partial x} \left( \frac{\partial}{\partial x} \left( \frac{\partial}{\partial x} \right) \right) - m = \frac{\partial}{\partial x} \left( \frac{\partial}{\partial x} \right) \frac{\partial}{\partial x} \frac{\partial}{\partial x} \frac{\partial}{\partial x}
\]

This same simplification is carried out for the other modified coefficients:
\[
Y_p' = Y_p + \frac{\partial Y_p}{\partial x} \frac{\partial}{\partial x} \\
K_n' = K_n + \frac{\partial K_n}{\partial x} \frac{\partial}{\partial x} \\
K_p' = K_p + \frac{\partial K_p}{\partial x} \frac{\partial}{\partial x} \\
N_n' = N_n + \frac{\partial N_n}{\partial x} \frac{\partial}{\partial x}
\]

The characteristic equation resulting from the expansion of equation (23) is a fourth-order equation which has four roots:
\[
\lambda + j\beta, \lambda - j\beta, \sigma + j\omega, \sigma - j\omega
\]

which can be used in impulse response equation.

2. SUBROUTINE DESCRIPTIONS

a. PLACE - This routine places each element of the matrix similar to table 1 in its location in the matrix ready to expand into the characteristic equation.

b. CHREQN - This subroutine expands the matrix formed by subroutine PLACE into the characteristic equation by taking the determinant of the matrix.

c. MULLER - This subroutine, by the method of Muller, calculates the roots of the characteristic equation.

3. INPUT DATA DECK

The input variables to the program include the coefficient and submarine constants noted in the matrix of equation (23). They are identified in detail with their locations in table 17.
**TABLE 17. INPUT DATA DECK, PROGRAM EC320**

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1-60</td>
<td>6F10.5</td>
<td>YVD, YV, YPD, YP, YRD, YR</td>
</tr>
<tr>
<td>2</td>
<td>1-60</td>
<td>6F10.5</td>
<td>AKVD, AKV, AKPD, AKP, AKRD, AKR</td>
</tr>
<tr>
<td>3</td>
<td>1-60</td>
<td>6F10.5</td>
<td>ANVD, ANY, ANPD, ANP, ANRD, ANR</td>
</tr>
<tr>
<td>4</td>
<td>1-10</td>
<td>F10.5</td>
<td>AL, submarine length, feet</td>
</tr>
<tr>
<td>4</td>
<td>11-20</td>
<td>F10.5</td>
<td>AM, submarine mass, slugs</td>
</tr>
<tr>
<td>4</td>
<td>21-30</td>
<td>F10.5</td>
<td>IX, movement of inertia about the X-axis, slug-ft²</td>
</tr>
<tr>
<td>4</td>
<td>31-40</td>
<td>F10.5</td>
<td>IZ, movement of inertia about the axis slug-ft²</td>
</tr>
<tr>
<td>4</td>
<td>41-50</td>
<td>F10.5</td>
<td>B, buoyancy, pounds</td>
</tr>
<tr>
<td>4</td>
<td>51-60</td>
<td>F10.5</td>
<td>ZB, z-component of center of buoyancy - feet</td>
</tr>
<tr>
<td>4</td>
<td>61-70</td>
<td>F10.5</td>
<td>RHO, density of sea water - slugs/ft³</td>
</tr>
<tr>
<td>5</td>
<td>1-10</td>
<td>8F10.5</td>
<td>YVAV, YVAR, YPAP, AKVAV, AKPAP, ANVAV, ANRAR, ANAVR</td>
</tr>
<tr>
<td>6</td>
<td>1-10</td>
<td>F10.5</td>
<td>RBAR, F, average yaw rate, based on a number of full set runs, radians/second</td>
</tr>
<tr>
<td>6</td>
<td>11-20</td>
<td>F10.5</td>
<td>VBAR, V, average normal velocity, feet/second</td>
</tr>
<tr>
<td>6</td>
<td>21-30</td>
<td>F10.5</td>
<td>PBAR, P, average roll rate, radians/second</td>
</tr>
<tr>
<td>7</td>
<td>1-5</td>
<td>I5</td>
<td>IDIV, this option allows the program user to divide YVAV<em>VBAR, YVAR</em>RBAR, YPAP<em>PBAR, AKVAV</em>VBAR, AKPAP<em>PBAR, ANVAV</em>VBAR, NAVY<em>VBAR and NAV</em>VBAR by velocity, before punching cards or letting the computer do the division.</td>
</tr>
</tbody>
</table>
TABLE 17. INPUT DATA DECK, PROGRAM EC320 (cont.)

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>1-10</td>
<td>F10.5</td>
<td>0 User to divide</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1 Computer divides by</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>forward velocity, knots.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Stack as many per set of coefficients as desired.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Place a blank card after last card of each set.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>This causes next case to be read.</td>
</tr>
<tr>
<td>last</td>
<td>1-5</td>
<td>F5.4</td>
<td>9999. The program takes normal end of job with this card.</td>
</tr>
</tbody>
</table>

4. OUTPUT DATA

Figure 18 of program EC140 shows a typical output from this program except that the measurement parameters are not calculated.

a. Input data - all input data (except IDIV) is printed out in the units detailed above.

b. The factors A1, A2, etc., of the characteristics equation are printed out. For this 4th-order equation

\[ A_1 \cdot c^4 + A_2 \cdot c^3 + A_3 \cdot c^2 + A_4 \cdot c + A_5 = 0 \]

c. Roots - the four roots of the characteristic equation are printed out.

5. LISTINGS AND FLOW CHART

The listings and flow chart for the main program of EC320 are given in appendix A. The subroutine listing and flow charts are identical to those of program EC140.

85
| VV      | VV      | VV      | VV      | VV      | VV      | VV      | VV      | VV      | VV      | VV      | VV      | VV      | VV      | VV      | VV      | VV      | VV      | VV      | VV      | VV      | VV      | VV      | VV      |
|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| -0.150 | -0.050  | -0.450  | -0.750  | -0.250  | -0.350  | -0.050  | -0.650  | -0.250  | -0.550  | -0.550  | -0.050  | -0.450  | -0.750  | -0.250  | -0.350  | -0.050  | -0.650  | -0.250  | -0.550  | -0.550  | -0.050  | -0.450  | -0.750  | -0.250  | -0.350  |
| 0.460  | 0.540  | 0.540  | 0.540  | 0.540  | 0.540  | 0.540  | 0.540  | 0.540  | 0.540  | 0.540  | 0.540  | 0.540  | 0.540  | 0.540  | 0.540  | 0.540  | 0.540  | 0.540  | 0.540  | 0.540  | 0.540  | 0.540  | 0.540  | 0.540  | 0.540  |

Figure 18. Output Variable Format, ECJ20
J. EC150, COEFFICIENT ESTIMATOR, LATERAL

1. DESCRIPTION

The response to a rudder impulse (lateral case) of a submarine can be approximated by the equation (24)

\[ y(t) = C_1 e^{-\xi t} + C_2 e^{-\zeta t} \cos(\beta t + \psi) + C_3 e^{-\delta t} \]  

(24)

where:
- \( y \) = roll angle of the submarine, radians
- \( t \) = time, seconds
- \( \xi, \zeta, \delta \) = roots of the characteristic equation or damping factor of the individual components
- \( \beta \) = frequency of oscillating term, radians/second
- \( \psi \) = phase delay, radians
- \( C_1, C_2, C_3 \) = coefficients of magnitude of each term

It is often quite useful to determine the magnitude of all the terms on the right side of equation (24) for a particular response to a lateral control impulse to a submarine in motion.

Values for \( y(t) \) can be calculated by solving the detailed "Standard Equations of Motion for Submarine Simulation". GAC program EB920 will perform these calculations over the time period desired. This program will also punch output cards with time and \( \phi \). This data deck will be used later. Program EB920 further provides a computer plot of \( \phi \) versus time as shown in figure 19 when CALCOMP plotter subroutines are available.

Program EC320, Root Cracker Program (Lateral Case), solves for the roots of the characteristic equation using the matrix Laplace form taken from the equations of motion. These roots are used for initial values \( \xi, \beta, \zeta, \delta \) in equation (14) above. They are not exact because of the program linearization. These values are inputs to program EC150, time response coefficient estimator, lateral, which will estimate the values \( C_1, C_2, C_3 \), and \( \psi \). The punched data deck from EB920 is the only other input data required.

The program calculates the following:

a. To find \( \psi \):

At \( t_1 \) (figure 19), the actual detailed calculated curve has decayed sufficiently that, for this calculation, the \( a_1 \) and \( a_3 \) terms may be considered zero. For all response curves having the first lobe after \( t = 0 \), the \( a_2 \) term is negative. Therefore, the positive peaks of figure 19 are equivalent to peak negative (-) values of the cosine term

\[ \cos(\beta t_1 + \psi) = -1 \]  

\[ \beta t_1 + \psi = \pi \]  

(25)
Figure 19. Submarine Response Curve, Lateral Case
The last term of equation (26) merely subtracts multiples of 360° from this calculated value of $\Psi$ to keep it an equivalent smallest number. This program automatically uses the value of the peak of the last lobe (searched out from the data deck) for the magnitude and time of $t_y$.

b. Time, $T_0$, at which the $a_3$ term of equation (24) is negligible:

$$T_0 = \frac{\Psi_{61}}{5}$$

The printout data will provide $T_0$ and also the values of $t_y$ and $t_5$. Value of $T_0$ should be checked to verify that it is less than $t_y$. Otherwise, ES920 may have to be run for a longer time to provide good input data for this program.

c. To find $a_2$:

Using values of $t_y$ and $t_5$ (selected automatically from the data deck input by this program) and assuming that the $a_3$ term is zero:

$$\gamma_1 = a_1e^{-\alpha t_5} - a_2e^{-\alpha t_y}$$

$$\gamma_5 = a_1e^{-\alpha t_5} - a_2e^{-\alpha t_5}$$

$$a_2 = \frac{-t_4 - t_5 \epsilon}{\epsilon(t_c - t_y)}$$

$$a_2 = -\frac{\epsilon(t_c - t_y) - \alpha t_5}{\epsilon + \epsilon x_2}$$

(d. To find $a_1$):

Substitute the value of $a_2$ just found into equation (27)

$$a_1 = \frac{\gamma_4 + a_2 \epsilon^{-\alpha t_y}}{e^{-\alpha t_y}}$$

(e. To find $a_3$):

At $t = 0$, equation (24) can be arranged as:

$$a_3 = -(a_1 + a_2 \cos t_y)$$

These values $\Psi$, $a_1$, $a_2$, and $a_3$ are outputs of the program E9150.
They may then be used, with the values of $\alpha$, $\beta$, $\gamma$, and $\delta$ from the Root Cracker program in EC310, Curve Fitting Program. This program will take these eight terms, calculate a set of values versus time, and plot against the exact calculated response values of EB920 for comparison as shown in figure 19.

Program EC310 has an option for convergence of six of the terms of equation (24) for optimum values by use of Brown's routine, $\delta$ and $a_3$ are usually held constant. This option can be exercised separately and plots run. The function will sometimes converge more closely than figure 19, and a very satisfactory set of values for the terms of equation (24) will be available. However, due to the approximations involved in EC320 and EC150, the function (which is converged through Brown's routine and needs very close initial values of all terms) may not converge. In this event, additional calculations or intuitive estimates for closer values of some or all terms, $\alpha$, $\beta$, $\gamma$, $\delta$, $a_1$, and $a_3$ may need to be made and the last one or two steps above repeated.

Even though final convergence may not be achieved with terms calculated through EC320 and EC150, the first plot of estimated values (of the nature of figure 19) is a very useful starting place for final convergence attempts.

2. INPUT DATA DECK

The input data consists principally of:

a. Punched cards from impulse run of EB920 Submarine Simulation Program, punched at two second intervals.

b. Values for $\alpha$, $\beta$, $\gamma$, and $\delta$ from EC320, Root Cracker, 3D, Lateral program.

The data deck format is given in table 18.

**TABLE 18. INPUT DATA DECK, PROGRAM EC150**

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1-5</td>
<td>15</td>
<td>N, number of data points from EB920 run (one per punched card)</td>
</tr>
<tr>
<td>2</td>
<td>1-10</td>
<td>F10.5</td>
<td>U, submarine speed, knots</td>
</tr>
<tr>
<td>2</td>
<td>11-18</td>
<td>2A4</td>
<td>NAME, use the word, PHI. For use as y-axis label on tabulated printout data sheets</td>
</tr>
<tr>
<td>3-n</td>
<td>1-10</td>
<td>E15.7</td>
<td>PHI(I) values on punched cards from EB920, radians</td>
</tr>
</tbody>
</table>
TABLE 18. INPUT DATA DECK, PROGRAM EC150 (cont.)

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-n</td>
<td>11-20</td>
<td>E15.7</td>
<td>T values on punched cards from EB920, seconds</td>
</tr>
<tr>
<td>n+1</td>
<td>1-10</td>
<td>F10.5</td>
<td>A, ( \alpha ), value of root calculated in EC320. Input value must be a positive number</td>
</tr>
<tr>
<td>n+1</td>
<td>11-20</td>
<td>F10.5</td>
<td>B, ( \varphi ), value of root from EC320, input as a positive number, radians/second</td>
</tr>
<tr>
<td>n+1</td>
<td>21-30</td>
<td>F10.5</td>
<td>C, ( \xi ), value of root from EC320, input as a positive number</td>
</tr>
<tr>
<td>n+1</td>
<td>31-40</td>
<td>F10.5</td>
<td>D, ( \xi ), value of root from EC320, input as a positive number</td>
</tr>
<tr>
<td>n+2</td>
<td></td>
<td></td>
<td>Blank card for normal end of job</td>
</tr>
</tbody>
</table>

4. OUTPUT DATA

The output includes sequentially numbered papers each identified with EC150. The following terms are printed:

- **U** - submarine speed, knots, \( \Phi(I) \) - all the values of the impulse run of EB920 as read by the card inputs to this program
- **YMIN** - the value of \( \Phi \) at \( \tau \xi \) (see figure 19) as selected from the input cards from EB920, radians
- **YMAX** - the value of \( \Phi \) at \( \tau \xi \), radians
- **T0** - \( \tau 0 \), time at which the a3 term of equation (24) is negligible, seconds
- **T1** - \( \tau 1 \), last peak value of oscillating function as selected from the input cards from EB920, seconds
- **TS** - \( \tau 0 \), last minimum value of oscillating function, seconds
- **A** - \( \alpha \), value from EC320
- **B** - \( \beta \), value from EC320
- **G** - \( \gamma \), value from EC320
- **P** - \( \psi \) or PSI, calculated phase delay, radians
A1 - $a_1$, calculated constant for equation (24)
A2 - $a_2$, calculated constant for equation (24)
D - $\delta$, value from EC320
A3 - $a_3$, calculated constant for equation (24)

5. LISTINGS AND FLOW CHART

The listings and flow chart for program EC150 are given in appendix A.
K. EC330, COEFFICIENT ESTIMATOR, LONGITUDINAL

1. DESCRIPTION

The response to an elevator impulse (longitudinal case) of a submarine can be approximated by equation (33)

\[ y(t) = a_1 e^{-\sigma t} + a_2 e^{-\alpha t} \cos(\beta t + \psi) \]  

(33)

where

- \( y \) = pitch angle (theta) of the submarine, radians
- \( t \) = time, seconds
- \( \alpha, \sigma \) = roots of the characteristic equation or damping factor of the individual components
- \( \beta \) = frequency of oscillating term, radians/second
- \( \psi \) = phase delay, radians
- \( a_1, a_2 \) = dimensionless coefficients of magnitude for each term

It is sometimes necessary to determine the magnitude of all the terms on the right side of equation (33) for a particular response to a longitudinal control impulse to a submarine in motion. This program provides a method for securing first, reasonable approximations of the terms:

- \( \alpha_1, \beta_1, \gamma, \psi, a_1, \text{ and } a_2 \)

The exact response to a specific impulse can be calculated by solving the equations of motion. GAC program EB920 will perform these calculations over the time period desired. This program (EB920), in addition, will punch cards of the output value, \( Y \) or theta, versus time. (Time intervals of punch data should be two seconds for use in EC330). This data deck will be used as an input to the subject program, EC330. Program EB920 further can provide a computer plot of \( Y \) (THETA) versus time as shown in figure 20.

The computer makes the following sets of calculations:

\( \beta \) - The first approximation for (See figure 20) is taken as a half cycle for the period \( t_1 \) through \( t_2 \). The computer searches the input data for these two points and takes the differences

\[ P = t_2 - t_1 \quad \beta = \frac{P}{F} \]
Figure 20. Submarine Response Curve, Longitudinal Case
\[ y = \text{at } t_2, \text{ assuming } a_1 \text{ term is zero:} \]
\[ \cos(\beta t_2 - \psi) = 0 \]
\[ \psi = \beta t_2 - \frac{\pi}{2} \]

\[ \alpha = \text{assuming } a_1 \text{ term is zero. Assume} \]
\[ t_3 = t_1 - \frac{\rho}{2} ; \quad t_4 = t_1 + \frac{\rho}{2} \]

The computer will interpolate from the input data to find the ordinate values of points \( t_3 \) and \( t_4 \). Then
\[ \frac{y_2}{y_4} = -\frac{a_2 e^{-\alpha t_3}}{a_2 e^{-\alpha t_4}} \]
\[ e^{\alpha(t_4 - t_3)} = -\frac{y_2}{y_4} \]
\[ \alpha = \frac{\ln\left(-\frac{y_2}{y_4}\right)}{t_4 - t_3} \]

\[ a_2 = \text{assume } a_1 \text{ term is zero} \]
\[ a_2 = y_3 / e^{-\alpha t_3} \]

\[ a_1 \]
\[ a_1 = -a_2 \cos(\psi) \]

\[ \gamma = \text{from equation (33)} \]
\[ y(t) = a_1 e^{-\gamma t} + a_2 e^{-\alpha t} \cos(\beta t - \psi) \]

\[ \gamma = -\frac{1}{t} \ln\left(\frac{y(t) - a_2}{a_2}\right) \]
These calculated values are printed out as the initial estimates. Since the first estimate assumed the term $a_1$ was zero and we now have a value for this term, new estimates can be made.

\[
AFLN = \frac{y_3 - a_1 e^{-y_t} t_3}{y_3 + a_1 e^{-y_t}}
\]

\[
\alpha_{n=0} = \ln(\text{AFLN}) / P
\]

\[
a_{1,n=0} = \frac{y_3 - a_1 e^{-y_t} t_3}{e^{-y_t} t_3}
\]

\[
a_{1,n=0} = -a_2 \cos \psi
\]

\[
\gamma_{n=0} = -\frac{1}{t(6)} \ln\left(\frac{y(6) - A2}{a_{1,n=0}}\right)
\]

These values are printed out in the second (improved) estimates along with the estimates of $\beta$ and $\Psi$ made above.

The second estimates may not be adequate due to unsatisfactory mathematical assumptions. The values of $a_1$ and $a_2$ can be estimated a third way and used if they subsequently prove more accurate than 2nd Estimate values.

\[
a_7 = \frac{y_3}{-\cos \Psi \ e^{-y_t} t_3 + e^{-y_t} \ cos(\beta t_3 - \Psi)}
\]

\[
a_1 = -a_2 \cos \Psi
\]

These new, alternate, values of $a_1$ and $a_2$ are printed out as the third estimate along with the other printed values of the second estimate.

The values computed above (first choice should be the second estimate
output values) may be used in program EC310, Curve Fitting Program along with the data deck cards (T and THETA) from the impulse run on EB920 and originally used as inputs to the subject program, EC330. This program, EC310, will calculate and plot the values of y versus t from equation (33) for the estimated coefficients. It will also plot the original impulse data from EB920 as recorded on the punch card data. The first graphical recording of these two curves will show whether a satisfactory agreement of the equation (33) values (with newly estimated variables) and the equations of motion data are adequate. Figure 20 shows excellent convergence of these values.

If these curves do not overlay satisfactorily program EC310 may be rerun with the convergence option. The variables of equation (33) are optimized to convergence by Brown's routine. The function will sometimes converge and a new, satisfactory set of values (\(a, b, c, \text{ etc.}\)) will be calculated. If convergence is not achieved, additional calculations or intuitive estimates of closer values can be used. These can be resubmitted to EC310 for proof till convergence is reached.

2. INPUT DATA DECK

The input data consists principally of

a. Punched cards, cards of time and Theta from impulse run of EB920, Submarine Simulation Program, punched at two-second intervals.

b. Additional control data

The data deck format is given in table 19.

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1-5</td>
<td>15</td>
<td>N, number of data points, Y(1) from EB920 run (one per punched card)</td>
</tr>
<tr>
<td>2</td>
<td>1-10</td>
<td>F10.5</td>
<td>U, submarine speed, knots</td>
</tr>
<tr>
<td>2</td>
<td>11-18</td>
<td>2A4</td>
<td>NAME, the name of the dependent variable, THETA</td>
</tr>
<tr>
<td>3-n</td>
<td>1-10</td>
<td>E15.7</td>
<td>THETA(I) values on punched cards from EB920, radians</td>
</tr>
<tr>
<td>3-n</td>
<td>11-20</td>
<td>E15.7</td>
<td>T, time values on punched cards from EB920, seconds</td>
</tr>
<tr>
<td>n+1</td>
<td></td>
<td></td>
<td>Blank card for normal end of job</td>
</tr>
</tbody>
</table>
3. OUTPUT DATA

The output data includes sequentially-numbered pages, each identified as EC330. The following terms are printed:

U - submarine speed, knots

T(I) and THETA(I) - all the values of the impulse run of EB920 as read by the card inputs to the subject program

A, B, G, P, A1, A2 - First estimates as described in equations 2 through 7 above.

A, B, G, P, A1, A2 - Improved estimates as noted in equations 8 through 11 above.

A, B, G, P, A1, A2 - Alternate estimates for a1 and a2 as described in equations 12 and 13 above.

4. LISTINGS AND FLOW CHART

The listings and flow chart for program EC330 are given in appendix A.
1. DESCRIPTION

This program has the option of calculating and plotting the curve of impulse response in accordance with equation (24) or (33). It can, in addition, calculate the difference at each time, $t_i$, between a tabulated EB920 run and the calculated values of an EC310 program run; set up a series of non-linear equations that are the summations of partial derivatives of either equation (24) and (33) times the difference between the two ordinate values at $t_i$; and finally solve these non-linear equations by Brown's routine. The values of $\alpha, \beta, \gamma, \delta, \varphi, \phi, \psi, \varphi, \psi, \alpha, Q_2 \cdots Q_6$ are optimized or converged by repeating the above solutions to limits within desired difference values (usually .0001) of successive estimates within the program. These converged values are then printed out along with graphs of the final solution of equation (24) or equation (33) and the original EB920 plot as in figure 21.

This program may have difficulties in convergence if eight variables are used. In this case the values of $\delta$ and $\alpha_3$ are held fixed at initial input values.

The equations utilized in Brown's routine are the summations of the partial derivatives times the difference in $Y$ (ordinate) values between the calculated and tabulated conditions. These are included below for clarity.

Let $Y_{ri}$ be the value of the tabulated points (EB920 values on the punched data cards).

Assume the following function is to be fitted to the data points:

$$y_i = a_0 e^{-\delta t_i} + a_2 e^{-\psi t_i} + ... + a_3 e^{-\phi t_i}$$

where $\delta, \psi, \phi, \varphi, \psi, \alpha, \psi, \alpha_3$ are to be determined by the method of least squares. The mean square error becomes

$$E = \frac{1}{N} \sum_{i=1}^{M} (y_i - Y_{ri})^2$$

Taking partial derivatives with respect to each parameter and equating to zero gives the following eight equations to be solved by Brown's routine. The routine will recalculate this till the $n^{th}$ and $(n-1)^{th}$ term agree to a preset number of significant digits (called NUMSIG in this program).
Figure 21. Typical Converged Response Plot
The following two values can be added to those for requested optimizing, but usually will cause divergence rather than convergence:

\[
\begin{align*}
\alpha & = \sum (y_i - y_{1i}) (-\tau_i \alpha_3 e^{-\alpha \tau_i}) \\
\beta & = \sum (y_i - y_{1i}) (-\tau_i \alpha_2 e^{-\beta \tau_i} \sin(\beta \tau_i + \psi)) \\
\gamma & = \sum (y_i - y_{1i}) (-\tau_i \alpha_1 e^{-\gamma \tau_i}) \\
\psi & = \sum (y_i - y_{1i}) (a_2 e^{-\alpha \tau_i} \sin(\beta \tau_i + \psi)) \\
\alpha_1 & = \sum (y_i - y_{1i}) (e^{-\alpha \tau_i}) \\
\alpha_2 & = \sum (y_i - y_{1i}) (e^{-\alpha \tau_i} \cos(\beta \tau_i + \psi)) \\
\end{align*}
\]

Brown's routine then solves the above system of simultaneous, non-linear equations. The algorithm used is quadratically convergent and requires only \( (n^2 + n)/2 \) function evaluations per iterative step as compared with \( (n^2 + n) \) evaluations for Newton's Method. This results in a savings of computational effort for sufficiently complicated functions. A detailed description of the general method and proof of convergence are included in reference 2. Basically the technique consists in expanding the first equation in a Taylor series about the starting guess, retaining only linear terms, equating to zero and solving for one variable, say \( x_k \), as a linear combination of the remaining \( n-1 \) variables. In the second equation, \( x_k \) is eliminated by replacing it with its linear representation found above, and again the process of expanding through linear terms, equating to zero and solving for one variable in terms of the now remaining \( n-2 \) variables is performed. One continues in this fashion, eliminating one variable per equation, until for the \( n \)th equation, we are left with one equation in one unknown. A single Newton step is now performed, followed by back-substitution in the triangularized linear system generated for the \( x_i \)'s. A pivoting effect is achieved by choosing for elimination at any step that variable having a partial derivative of largest absolute value. The pivoting is done without physical interchange of rows or columns.
The vector of initial guesses, \( X \), the number of significant digits desired, the maximum number of iterations to be used, and the number of equations to be solved are input data. After execution of the procedure, the vector \( x \) is the solution of the system (or best approximation thereto). A printout that a Jacobian-related matrix was singular is indicative of the process "blowing-up". If this occurs, try another initial estimate of values:

\[
\alpha, \beta, \delta, \psi, a_1, a_2, s \text{ and/or } a_3
\]

This program can be used with either a rudder or elevator impulse. To find frequencies of oscillation in response to a rudder (lateral) impulse:

a. Run EB920 for the DR(lateral or rudder) impulse of desired magnitude, producing punched cards of output \( \phi(\theta) \) and time \( t \) as well as printout data of dynamic response.

b. Exercise EC320, Root Cracker Program, 3D, Lateral to determine the roots of Equation (35), namely:

\[
\alpha, \beta, \delta \text{ and } s
\]

c. Using the cards from EB920 and \( \alpha, \beta, \delta \), and \( s \) from EC320 as inputs, exercise program EC150, Time Response Coefficient Estimator, Lateral. This program provides estimated values of several terms of the lateral impulse equation:

\[
y(t) = a_1 e^{-\delta t} + a_2 e^{-\beta t} \cos(\beta t + \psi) + a_3 e^{-\delta t}
\]

where \( y \) represents bank angle, \( \phi(\theta) \)

The values solved are:

\[
a_1, a_2, a_3, \text{ and } \psi
\]

d. Using the coefficients and roots of b. and c. above, and the punch cards of EB920 of a. above, exercise program EC310. This program plots the tabulated data of the punch cards (dotted curve similar to figure 21) and the calculated response curve per equation 35 (solid curve of figure 21). A printout of the calculated and tabulated data is provided.

To find frequencies of oscillation in response to an elevator (longitudinal) impulse:

a. Run EB920, Submarine Simulation Program for the DS (longitudinal
or elevator) impulse of desired magnitude. The program output will
supply a set of punched cards each carrying a value of theta (θ) and
time (t) as well as the printout data of these same values. This data
is the true response of the submarine to an elevator impulse.

b. Using these cards as input data, exercise program EC330, Time
Response Coefficient Estimator, Longitudinal. This program provides
estimated values for the constants of the longitudinal impulse equation:

\[ y = a_1 e^{-σ t} + a_2 e^{-τ t} + \cos(β t + φ) \]

where y represents pitch angle, theta (θ)

The values solved are:

\[ α, β, σ, φ, a_1, a_2 \]

You may use the values \( α \), \( β \), \( σ \), \( φ \), \( a_1 \), and \( a_2 \) provided through program
EC140, Root Cracker Program, 3D, Longitudinal.

c. Using these estimated roots and coefficients along with the
punch card data from EB920 noted in a. above as inputs, exercise program
EC310. This program provides a computer plot of the actual response
curve provided through the punch cards of EB920 (dotted curve of figure
21), the calculated response curve per equation (33)(solid curve of
figure 21). A printout of the calculated and tabulated data is also
provided.

2. SUBROUTINE DESCRIPTIONS

PLOTS - plotting routine purchased from California Computer
for use with their plotter. They must be secured
from this company.

PLOT - plotting routine from Cal Comp, to start pen

FCNPLT - plot routine which labels axes, draws lines and symbols
and calls for other subroutines

SCALE - plot routine from Cal Comp for automatic scaling of
axes.

AXIS - plot routine from Cal Comp to produce axis

SYMBOL - plot routine from Cal Comp to produce letters and
numbers on graph

NUMBER - plot routine from Cal Comp to write a floating point
number from the program

103
LINE - plot routine from Cal Comp that connects a series of points

AUXFCN - this routine calculates the function of equations 1 or 7 and prepares the matrix of equations (Eq's 5 through 12) for Brown's routine (also known as subroutine SYSTEM in this program). This subroutine also calculates the magnitude of each of the equations 5 through 12.

SYSTEM - Brown's routine which solves six or eight (usually restrict the value to six in order to secure convergence) simultaneous, non-linear equations.

3. INPUT DATA DECK

The input data to this program include the punch card data from the particular impulse run on program EB920, control data for Brown's routine, plot control, and initial guesses or starting values of

\[ \omega, p, \sigma, \gamma, a, q, \alpha \]  - longitudinal

\[ \omega, p, \gamma, a, q, \alpha, \sigma \]  - lateral

These are identified in detail with their locations in table 20.

### TABLE 20. INPUT DATA DECK, PROGRAM EC310

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1-5</td>
<td>I5</td>
<td>N, number of equations to be used (N = 6 for longitudinal, N = 6 or 8 for lateral (8 will seldom converge, so 6 is recommended))</td>
</tr>
<tr>
<td>1</td>
<td>6-10</td>
<td>I5</td>
<td>NUMSIG, number of significant digits of agreement between successive iterates which will cause convergence of the program. This value has been set at 4 for work on this program.</td>
</tr>
<tr>
<td>1</td>
<td>11-15</td>
<td>I5</td>
<td>MAXIT, maximum number of iterations to be performed by Brown's routine.</td>
</tr>
<tr>
<td>1</td>
<td>16-20</td>
<td>I5</td>
<td>IPRINT, will print out all iterations of values, up to MAXIT above if set to 1. IPRINT = 0 will print only first and</td>
</tr>
</tbody>
</table>
### TABLE 20. INPUT DATA DECK, PROGRAM EC310 (cont.)

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>21-25</td>
<td>I5</td>
<td>last values of values calculated (&lt;, β, γ, γ₁, α₁, φ, γ₂)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NPTS, number of data prints. This is the number of data cards from EB920.</td>
</tr>
<tr>
<td>1</td>
<td>26-30</td>
<td>I5</td>
<td>ISWI = 1 - will calculate and print the equations (1 or 2) at the initial guess.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ISWI = 0 - will attempt to converge for better values of γ₁, β, γ, γ₁, γ₂, φ with Brown's routine and calculate equations 1 or 2 for the converged values.</td>
</tr>
<tr>
<td>1</td>
<td>31-35</td>
<td>I5</td>
<td>IPLOT = 1 - will plot the tabulated and calculated data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IPLOT = 0 - no plot</td>
</tr>
<tr>
<td>2</td>
<td>1-10</td>
<td>F10.5</td>
<td>U, forward speed, knots</td>
</tr>
<tr>
<td>2</td>
<td>11-18</td>
<td>2A4</td>
<td>IX, word &quot;PHI&quot; for lateral case word &quot;THETA&quot; for longitudinal case</td>
</tr>
<tr>
<td>3-n</td>
<td>1-15</td>
<td>E15.7</td>
<td>Y(I), values of dependent variable (phi or theta) from cards of EB920 run, I = NPTS</td>
</tr>
<tr>
<td>3-n</td>
<td>16-30</td>
<td>E15.7</td>
<td>T(I), values of independent variable (t) from cards of EB920</td>
</tr>
<tr>
<td>u+1</td>
<td>1-80</td>
<td>8F10.5</td>
<td>A, B, C, P, A1, A2, D, A3 (&lt;, f, γ, γ₁, α₁, φ, γ₂), initial estimates of unknowns of equations 1 or 2.</td>
</tr>
<tr>
<td>m+2</td>
<td>-</td>
<td>-</td>
<td>Blank card for normal end of job.</td>
</tr>
</tbody>
</table>

### 4. OUTPUT DATA

Printout Data:

a. Output values - The printout across the page will include columns of:
Nth time through Brown's routine - 0 = initial estimate

\[ x, f, r, \psi, a_1, a_2, \ldots \]

b. Value of partial derivatives of equation 1 at convergence in a row of the following order:

\[ \frac{\partial y}{\partial x}, \frac{\partial y}{\partial f}, \frac{\partial y}{\partial r}, \frac{\partial y}{\partial \psi}, \frac{\partial y}{\partial a_1}, \frac{\partial y}{\partial a_2}, \ldots \]

c. Converged, or final values of calculated terms in a row in following order:

\[ a, f, r, \psi, a_1, a_2, \ldots \]

d. Columns of time (T), and the calculated and tabulated values of Y(I) in the following order:

\[ T, YE, YT \]

Graphical Data:

If the CAL COMP subroutines are available a plot as shown in figure 21 can be produced.

5. LISTINGS AND FLOW CHART

The listings and flow chart for program EC310 are given in appendix A.
M. EC790, CALCULATION OF COMPACT COEFFICIENTS

1. DESCRIPTION

Program EB920, Submarine Simulation Program, was written to provide a flexible, analytical tool for examining submarine performance with a number of optional devices of submarine control, program control and graphical as well as printed output. To provide the computer control necessary for a submarine training device, the many optional controls and outputs of EB920 are not required; a number of the coefficients and constants can be combined to require fewer multiplications and other operations; and a much smaller computer can be used. This program uses the normal input coefficients, submarine physical constants, and control values of program EB920 and the exact input data deck from an EB920 run may be used. The program then combines a number of coefficients, multiplies and divides them by the appropriate constants. This program thus, takes data identical to that for a scientific run with program EB920; combines coefficients and multiplies by constants, prints the original and combined coefficients separately; and finally punches data cards for input to program EC780 compact submarine simulation program when it has been compiled on a small computer. Use of this smaller computer demonstrates that a much smaller machine (8K core) can provide adequate storage for real time control of an actual training device.

Equation group number (36) shows the mathematical operations performed on the original coefficients and constants used for the inputs to program EB920. The primed values on the right side of the equation are identical to those used in EB920 and are defined in the glossary. The unprimed values on the left side of the equation are repeated as FORTRAN variables used by program EC790 and EC780.

2. INPUT DATA DECK

The program requires data input exactly as required by program EB920. All units coefficients, and constants, etc., as used in an EB920 run are identical. This input data deck assembly and use has been described in that program.

Since EC780, which will use the output deck from this program, cannot perform many of the operation of EB920, some of the data normally required in EB920 runs can be left blank in the proper punch locations on the data deck if desired. However, all cards required by EB920, even if completely blank, must be present at input to this program.

3. OUTPUT DATA

Two data sheets are printed by this program. The first one is identical to figure of program EB920 and shows the original coefficients present on the input data deck. The second one is printed out in the same format but the values of those coefficients to be used by program EC780 are printed instead. The other coefficients that were not modified in this program are printed on this same page, but are not used further.
Coefficient Description (EC780)

\[\begin{align*}
X_{ORDR}' &= X_{5_{or}5_{or}} = \frac{\ell^2}{2} X_{5_{or}5_{or}} \\
X_{DSDS}' &= X_{5_{s5}5_{s5}} = \frac{\ell^2}{2} X_{5_{s5}5_{s5}} \\
X_{DBDB}' &= X_{5_{b5}5_{b5}} = \frac{\ell^2}{2} X_{5_{b5}5_{b5}} \\
A_{11}' &= a_{i_1} = \frac{\ell^2}{2} a_{i_1} \\
A_{12}' &= b_{i_2} = \frac{\ell^2}{2} b_{i_2} \\
A_{13}' &= c_{i_3} = \frac{\ell^2}{2} c_{i_3} \\
A_{21}' &= a_{i_2} = \frac{\ell^2}{2} a_{i_2} \\
A_{22}' &= b_{i_3} = \frac{\ell^2}{2} b_{i_3} \\
A_{23}' &= c_{i_3} = \frac{\ell^2}{2} c_{i_3}
\end{align*}\]
\[ A 31' = a_{z_3} = \frac{\mu}{2} l^3 a_{z_3} \]
\[ A 32' = b_{z_3} = \frac{\mu}{2} l^3 b_{z_3} \]
\[ A 33' = c_{z_3} = \frac{\mu}{2} l^3 c_{z_3} \]
\[ X UD' = X \omega' = \frac{\mu}{2} l^3 X \omega \]
\[ Y R' = Y_\rho' = \left( \frac{\mu}{2} l^3 Y_\rho - m \right) \]
\[ Y RD' = Y_\nu' = \frac{\mu}{2} l^3 Y_\nu \]
\[ Y PD' = Y_\phi' = \frac{\mu}{2} l^3 Y_\phi \]
\[ Y P' = Y_\nu' = \frac{\mu}{2} l^3 Y_\nu \]
\[ Y V' = Y_\eta' = \frac{\mu}{2} l^3 Y_\eta \]
\[ Y VAV' = Y_\eta_{1\lambda} = \frac{\mu}{2} l^2 Y_{\eta_{1\lambda}} \]
\[ Y DR' = Y_{\nu_{b_1}} = \frac{\mu}{2} l^2 Y_{\nu_{b_1}} \]
\[ Y VD = Y_\nu = \frac{\mu}{2} l^2 Y_\nu \]
\[ Z Q' = Z_i = \frac{\mu}{2} l^3 Z_i \]
\[ Z QD' = Z_i = \frac{\mu}{2} l^4 Z_i \]
\[ Z R' = Z_{rr} = \frac{\mu}{2} l^4 Z_{rr} \]
$$ZV_R' = Z_{V_R}' = \frac{\mu}{2} l^3 Z_{V_R}$$
$$ZSR' = Z_{SR}' = \frac{\mu}{2} l^2 Z_{SR}$$
$$ZW' = Z_{W}' = \frac{\mu}{2} l^2 Z_{W}$$
$$ZWAW' = Z_{W_{AWW}}' = \frac{\mu}{2} l^2 Z_{W_{AWW}}$$
$$ZV' = Z_{V_R}' = \frac{\mu}{2} l^2 Z_{V_R}$$
$$ZDS' = Z_{DS}' = \frac{\mu}{2} l^2 Z_{DS}$$
$$ZDB' = Z_{DB}' = \frac{\mu}{2} l^2 Z_{DB}$$
$$ZWD' = Z_{W_D}' = \frac{\mu}{2} l^3 Z_{W_D}$$

$$T_A = \frac{1}{(I_A - \frac{\mu}{2} l^5 K_{K_A})}$$
$$AKRO' = K_{RO}' = \frac{\mu}{2} l^5 K_{RO} \cdot T_A$$
$$AKP' = K_{P}' = \frac{\mu}{2} l^4 K_{P} \cdot T_A$$
$$AKVD' = K_{VD}' = \frac{\mu}{2} l^4 K_{VD} \cdot T_A$$
$$AKV' = K_{V}' = \frac{\mu}{2} l^3 K_{V} \cdot T_A$$
$$AKVA' = K_{VA}' = \frac{\mu}{2} l^3 K_{VA} \cdot T_A$$
$$AKPD' = K_{PD}' = T_A$$
\[ T_s = \frac{1}{(I_T - \frac{3}{2} l^3 M_b)} \]

\[ \text{AMRP}' = M_{r_p}' = (I_e - I_x + \frac{3}{2} l^3 M_{r_p}) \cdot T_s \]

\[ \text{AMRR}' = M_{r_r}' = \frac{3}{2} l^3 M_{r_r} \cdot T_s \]

\[ \text{AMWD}' = M_{w}' = \frac{3}{2} l^3 M_{w} \cdot T_s \]

\[ \text{AMVR}' = M_{v_r}' = \frac{3}{2} l^3 M_{v_r} \cdot T_s \]

\[ \text{AMQ}' = M_{q}' = \frac{3}{2} l^3 M_{q} \cdot T_s \]

\[ \text{AMAWQ}' = M_{awq}' = \frac{3}{2} l^3 M_{awq} \cdot T_s \]

\[ \text{AMSTR}' = M_{str}' = \frac{3}{2} l^3 M_{str} \cdot T_s \]

\[ \text{AMW}' = M_{w}' = \frac{3}{2} l^3 M_{w} \cdot T_s \]

\[ \text{AMWAW}' = M_{awaw}' = \frac{3}{2} l^3 M_{awaw} \cdot T_s \]

\[ \text{AMVV}' = M_{vv}' = \frac{3}{2} l^3 M_{vv} \cdot T_s \]

\[ \text{AMOS}' = M_{os}' = \frac{3}{2} l^3 M_{os} \cdot T_s \]

\[ \text{AMDB}' = M_{db}' = \frac{3}{2} l^3 M_{db} \cdot T_s \]

\[ \text{AMQO}' = M_{qo}' = T_s \]
\begin{align*}
T_c & = \frac{1}{(I_x - \frac{\rho}{2} l^6 N_P)} \\
ANPQ' & = N_{P_b}' = \left( I_x - I_y + \frac{\rho}{2} l^6 N_{P_b} \right) \cdot T_c \\
ANPD' & = N_P' = \frac{\rho}{2} l^6 N_{P_b} \cdot T_c \\
ANVD' & = N_{\omega}' = \frac{\rho}{2} l^4 N_{\omega} \cdot T_c \\
ANP' & = N_P' = \frac{\rho}{2} l^4 N_{P_b} \cdot T_c \\
ANR' & = N_P' = \frac{\rho}{2} l^4 N_P \cdot T_c \\
ANV' & = N_{\omega}' = \frac{\rho}{2} l^3 N_{\omega} \cdot T_c \\
ANVAV' & = N_{\omega_{avr}}' = \frac{\rho}{2} l^3 N_{\omega_{avr}} \cdot T_c \\
ANDR' & = N_{\omega_{avr}}' = \frac{\rho}{2} l^3 N_{\omega_{avr}} \cdot T_c \\
ANRD' & = N_{\omega_{avr}}' = T_c \\
EB' & = g_0 = CB \cdot g_0
\end{align*}
This program also punches the newly calculated, modified coefficients on data cards for direct use in EC780, Compact Submarine Simulation Program. This data is punched in 6E13.6 format on the data cards. The terms are identified in table 21.

<table>
<thead>
<tr>
<th>Card</th>
<th>Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>XDRDR, XDSDS, XDBDB, A11</td>
</tr>
<tr>
<td>2</td>
<td>A12, A13, A21, A22, A23, A31</td>
</tr>
<tr>
<td>3</td>
<td>A32, A33, XUD, YR</td>
</tr>
<tr>
<td>4</td>
<td>YRD, YPD, YP, YV, YVAB, YDR</td>
</tr>
<tr>
<td>5</td>
<td>YPD, ZQ, ZRD, ZRR</td>
</tr>
<tr>
<td>6</td>
<td>ZVR, ZSTR, ZM, ZWAV, ZV, ZDS</td>
</tr>
<tr>
<td>7</td>
<td>ZDB, ZWD, AKRD, AKP, AKV</td>
</tr>
<tr>
<td>8</td>
<td>AKV, AKVAV, AKPD, AMRP, AMRR</td>
</tr>
<tr>
<td>9</td>
<td>AMWD, AMVR, AMQ, AMAWQ, AMSTR, AMW</td>
</tr>
<tr>
<td>10</td>
<td>AMWAV, AMVV, AMDS, AMDB, AMQD</td>
</tr>
<tr>
<td>11</td>
<td>AMQ, ANPD, ANVD, ANP, ANR, ANV</td>
</tr>
<tr>
<td>12</td>
<td>ANVAV, ANDR, ANRD, DRMAX, ETAHI</td>
</tr>
<tr>
<td>13</td>
<td>ETALO, CW, CR, XG, ZG, AL</td>
</tr>
<tr>
<td>14</td>
<td>AM, DR, DS, DB, ZB, UC</td>
</tr>
<tr>
<td>15</td>
<td>TIME, RL, DELTMA, SWMAX, R2, DELTMI</td>
</tr>
<tr>
<td>16</td>
<td>DSF, DRF, Y(1), Y(2), Y(3), Y(4), Y(5), Y(6), Y(7), Y(8), Y(9), Y(10), Y(11), Y(12)</td>
</tr>
</tbody>
</table>

4. LISTINGS AND FLOW CHART

The listings and flow chart for program EC790 are given in appendix A.
EC780, COMPACT SUBMARINE SIMULATION PROGRAM

1. DESCRIPTION

This program is a limited, modified version of EB920, Submarine Simulation Program. It will provide all output data in accordance with the equations of motion. A number of types of preplanned submarine maneuvers have been included for testing the program. This program has been modified from EB920 by utilizing a number of "synthetic coefficients" which were calculated in program EC790, Calculation of Compact Coefficients. These synthetic coefficients are the normal coefficients from the equations of motion (as used directly in program EB920) that have been combined and multiplied by appropriate constants in this separate program rather than the simulation program. These new coefficients, along with the removal of a number of options from the scientific research program, EB920, allow this program, EC780, to solve the equations of motion with fewer operations and much less core.

This program operates with the modified coefficients from program EC790. The specific mathematical model is therefore somewhat different from that described in Reference 1 and used in program EB920, and are given in equations (37) through (42).

Coefficient terms used above are defined in the associated input data program, EC790.

2. SUBROUTINE DESCRIPTIONS

CONTR - This subroutine allows control of submarine motion by programed movement of elevators and/or rudder. This allows selection of the various maneuvers:

a. Steady dive, turn, or combination
b. Meander or overshoot
c. Flat turn with autopilot
d. Climbing turn, combination of a programed turn and meander or overshoot
e. JS or elevator impulse (no punched cards are prepared by the computer as in this maneuver with program EB920) with autopilot
f. DR or rudder impulse (no punched cards)
g. Acceleration/deceleration
h. Maximum acceleration/deceleration

These controls function and are input exactly as in program EB920, so reference is made to this program for detailed information.

UPDATE - This subroutine sets the propeller thrust constants, ai, bi, ci in accordance with the current value of \( \frac{\mathbf{U}}{\mathbf{U}} \); (Ua / U). It solves the equations of motion at each updated time increment, H. These values, \( \dot{u}, \dot{\psi}, \ldots \)
Compact Mathematical Model
Submarine Equations of Motion

Axial

\[ \ddot{u} = \left[ m ( \omega r - \omega q ) \\
+ X_a ( q^2 + r^2 ) - Z_a ( pr + \dot{q} ) \\
+ u^2 ( X_{6v5r} \delta_r + X_{6s6s} \delta_s^2 + X_{6w6w} \delta_w^2 + a_1 ) \\
+ u_c ( b_c u + c_c u_c ) \\
- (W - B) \sin \theta \right] \] \[ \left[ m - X_u \right] \]

Lateral

\[ \ddot{\nu} = \left[ u ( Y_u r + Y_p p + Y_{\omega} \nu ) \\
- m ( Z_a ( q r - \dot{p} ) + X_a ( q p + \dot{r} ) - \omega p + \omega r ) \\
+ Y_c \dot{r} + Y_p \dot{p} + Y_{\omega m} \nu ( \nu^2 + \omega^2 ) t \\
+ Y_{6s} u^2 \delta_r + (W - B) \cos \theta \sin \phi \right] \left[ m - Y_u \right] \]

115
Normal

\[ \dot{\omega} = \left[ Z_{\omega \rho} \nu \rho + u (Z_\theta \theta + Z_\omega \omega) - m (Z_\alpha (\dot{\rho}^2 + \dot{\theta}^2) - X_\alpha (r \rho - \dot{\theta}) + g u - \nu \rho) + Z_\theta \dot{\theta} + Z_{\rho \rho} r^2 + Z_{\rho \rho \rho} \nu \rho + Z_{\rho \rho \rho} \nu^2 + Z_{w w w} w (\nu^2 + \omega^2)^{1/2} + u^2 (Z_{\delta \delta} \delta_\delta + Z_{\delta \delta} \delta_\omega + Z_{\delta \omega}) + (W - B) \cos \theta \cos \phi \right] / [m - Z_\rho] \]

Roll

\[ \dot{\phi} = m (Z_\alpha (\dot{\rho} - \nu \rho + u r) K_p + K_\phi \dot{\rho} + K_{\phi \phi} \dot{\phi} + K_{\phi \phi} \dot{\phi}) + u (K_p \rho + K_{\phi \phi} \nu \rho) + K_{\phi \phi} (\nu^2 + \omega^2)^{1/2} - K_p (Z_\alpha W - Z_\alpha B) \cos \theta \cos \phi \]

Pitch

\[ \dot{\theta} = m (Z_\alpha (\dot{\rho}^2 + \nu \rho + \theta) - X_\alpha (\dot{\rho} - \nu \rho + \nu \rho) \theta) \theta + M_{\rho \rho} r^2 + M_{\rho \rho \rho} \nu \rho + M_{\rho \rho \rho} \nu^2 + M_{\omega \omega} g (\nu^2 + \omega^2)^{1/2} + M_{\omega \omega} w (\nu^2 + \omega^2)^{1/2} + u^2 (M_{\delta \delta} + M_{\delta \delta} \delta_\delta + M_{\delta \delta} \delta_\omega) - M_\delta (Z_\alpha W \cos \theta \cos \phi + (Z_\alpha W - Z_\alpha B) \sin \theta) \]

Yaw

\[ \dot{\rho} = m (X_\alpha (\dot{\rho} - \nu \rho + u r) N_\rho + N_\rho \dot{\rho} + N_{\rho \rho} \nu \rho + u^2 N_{\delta \delta} \delta_\delta + u (N_p \rho + N_r r + N_{\rho \rho} \nu) + N_\rho (Z_\alpha W) \cos \theta \sin \phi \]
\[ \dot{\dot{\psi}}, \dot{\dot{\phi}}, \dot{\dot{\psi}}, \dot{\dot{\phi}}, \text{with kinematics, } \dot{\psi}, \dot{\phi}, \phi, \text{ and } \dot{x}, \dot{y}, \text{ and } z \text{ are then integrated over the time interval, } H. \text{ These new values returned to the main program for possible printout, and the old derivative values replaced by the new values each pass through the subroutine.} \]

3. INPUT DATA DECK

The input data consists of two special control cards plus the punch card output data deck from EC790, Calculation of Compact Coefficients. The inputs to that program are exactly as for inputs to EB920 with the exception that values for terms not used may be left blank. The subject program is set up so that an input data deck that had been used for a particular run on EB920 can be used for input to EC790 and the output cards of EC790 used with the two special control cards as the input data deck for the subject program, EC780. The data deck format for this program is given in table 22.

**TABLE 22. INPUT DATA DECK, PROGRAM EC780**

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1-5</td>
<td>IS</td>
<td>IRUN, run number</td>
</tr>
<tr>
<td>1</td>
<td>6-10</td>
<td>IS</td>
<td>NPNT, calculated values will printout each NPNT integration cycle (If ( H = 0.25 ) seconds, and ( NPNT = 8 ), printout interval is ( NPNT \times H = 2 ) seconds)</td>
</tr>
<tr>
<td>1</td>
<td>11-15</td>
<td>IS</td>
<td>NS. This variable selects the type of submarine control in CONTR subroutine: ( NS = 0 ), Fixed controls per initial conditions ( NS = 1 ), Overshoot, meander, etc. ( NS = 2 ), Special climbing term ( NS = 3 ), Flat turn (with autopilot) ( NS = 4 ), Elevator impulse ( NS = 5 ), Rudder impulse (with autopilot) ( NS = 6 ), Acceleration/deceleration (with autopilot) ( NS = 7 ), Maximum acceleration/deceleration (with autopilot) (Details of these controls are in EB920 write up)</td>
</tr>
</tbody>
</table>
| 2    | 11-20     | F10.5  | H, integration time increment, seconds. This step size is used throughout the entire run.
TABLE 22. INPUT DATA DECK, PROGRAM EC780 (cont.)

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>21-30</td>
<td>F10.5</td>
<td>TLIM, time at the end of the run, seconds.</td>
</tr>
<tr>
<td>3-18</td>
<td>1-78</td>
<td>6E13.6</td>
<td>Punch card data from EC790, calculation of compact coefficients. Table defines the variables punched on these cards.</td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
<td>Blank card for normal end of job.</td>
</tr>
</tbody>
</table>

4. OUTPUT DATA

A sample output data sheet from this program (using trial or synthetic input coefficients from EC790) is shown in figure 23. The run number and control routine value NS are provided in the first line of data. Then, at the desired intervals of time, the calculated values (and time) from the solutions to the equations of motion are printed (u, v, w, p, q, r, θ, ψ, φ, x, y, z, and t).

5. COMPUTER RUN TIME

For the SDS Sigma 2 computer, cycle and add time is 2.25 microseconds. The subject programs required 0.2 seconds for complete integration period with no I/O.

6. LISTINGS AND FLOW CHART

The program is written in basic FORTRAN, primarily for use on an SDS, Sigma 2 or other small computers. It is a streamlined version of program EB920, and has operated on the IBM 360/40 computer. It is not intended for normal use on this larger computer as the input has been tailored for card-reader input and eighty-column typewriter output available on the Sigma 2. The listings and flow chart are given in appendix A.
**Figure 22. Output Data Format, EC780**

<table>
<thead>
<tr>
<th>( \theta )</th>
<th>( \phi )</th>
<th>( \phi_{\phi} )</th>
<th>( \phi_{\theta} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.344500E+01</td>
<td>0.0</td>
<td>0.435340E-01</td>
<td>0.0</td>
</tr>
<tr>
<td>0.0</td>
<td>0.515625E-02</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0.0</td>
<td>0.607000E-03</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0.749529E+01</td>
<td>0.0</td>
<td>0.479612E-01</td>
<td>0.0</td>
</tr>
<tr>
<td>0.0</td>
<td>0.501091E-02</td>
<td>0.0</td>
<td>0.806341E-02</td>
</tr>
<tr>
<td>0.0</td>
<td>0.803222E-03</td>
<td>0.109700E-02</td>
<td></td>
</tr>
<tr>
<td>0.463147E+01</td>
<td>0.0</td>
<td>0.412510E-01</td>
<td>0.0</td>
</tr>
<tr>
<td>0.0</td>
<td>0.456127E-02</td>
<td>0.0</td>
<td>0.151214E-03</td>
</tr>
<tr>
<td>0.0</td>
<td>0.809101E-03</td>
<td>0.200700E-02</td>
<td></td>
</tr>
<tr>
<td>0.539151E+02</td>
<td>0.0</td>
<td>0.396526E-01</td>
<td>0.0</td>
</tr>
<tr>
<td>0.0</td>
<td>0.374146E-02</td>
<td>0.0</td>
<td>0.213483E-03</td>
</tr>
<tr>
<td>0.0</td>
<td>0.803222E-03</td>
<td>0.300900E-02</td>
<td></td>
</tr>
<tr>
<td>0.608852E+03</td>
<td>0.0</td>
<td>0.356788E-01</td>
<td>0.0</td>
</tr>
<tr>
<td>0.0</td>
<td>0.261402E-02</td>
<td>0.0</td>
<td>0.269074E-03</td>
</tr>
<tr>
<td>0.0</td>
<td>0.809101E-03</td>
<td>0.400000E-02</td>
<td></td>
</tr>
</tbody>
</table>
1. DESCRIPTION

Program EC572, WAVE GENERATOR Program, was written to provide a means of generating a random ocean wave surface at one point in time. It can be used with the Submarine Simulation Program to affect submarine motion close to the surface; but at present the outputs are only a choice of printed points, a graph plotted against time, or points and time punched on cards for PSD analysis. The program uses a mathematical model developed in "Mathematical Generation of a Realistic Sea", Hydromatnics, Inc. Technical Report 001-13 (DDC # AD 609906) prepared for the Bureau of Ships in October 1963. However, the wave spectra used is not the standard Neumann used in the report, but the more up-to-date Pierson-Moskowitz spectra. This spectra has the equations

\[ A^2(\omega) = \frac{g}{10^3} \frac{3^2}{\omega^6} e^{-\frac{\gamma}{U}} \left( \frac{g}{U^2} \right)^{\gamma} \]

- \(A^2(\omega) = \) wave spectra
- \(\omega = \) frequency
- \(g = \) gravity
- \(U = \) wind speed

The program first determines the frequency points at which the area under \(A^2(\omega)\) can be divided into equal sections. This is done in closed form by integrating \(A^2(\omega)\) and finding the area between two limits.

\[ A_{\omega} = \int_{\omega_1}^{\omega_2} A^2(\omega) d\omega = \frac{81 \times 10^{-3} U^3}{g^2} e^{-\frac{\gamma}{U}} \left( \frac{g}{U^2} \right)^{\gamma} \]

The total area is divided into the number of specified bands and the program solves for the upper limit of the expression to give this value. This value is then used as a lower limit for the next value until \(R\) is reached. The series of frequencies are stored, the program then calls a random number generator in the system library to generate enough random numbers \(U_j\) between zero and \(2\pi\) for use in the expression below.

The surface amplitude is then calculated by means of the expression...
\[ \bar{y}(t) = B \sum_{j=1}^{S} \sum_{i=1}^{n} (\omega_j)^{-\alpha} e^{-\gamma_j (\omega_j)^{\alpha}} \cos(\omega_j t + \epsilon_{ij}) (\pi, \omega_j)^{\gamma} \]

when

\[ C_1 = C_5 = 0.9058 \]
\[ C_2 = C_7 = 0.43305 \]
\[ C_3 = 0.53254 \]

\[ B = \frac{2}{\pi} \times 8.1 \times 10^{-3} g^2 \]

\( g \) is the speed in knots

Development of this equation is covered in the reference 3. The rest of the program consists of control for inputs, outputs, error messages, and times the program is to be run.

2. SUBROUTINE DESCRIPTIONS

The CAL COM subroutines must be supplied by the operating system if plots are desired. They are

<table>
<thead>
<tr>
<th>PLOTS</th>
<th>LINE</th>
<th>SCALE</th>
<th>INEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>INIT</td>
<td>PLOT</td>
<td>AXIS</td>
<td></td>
</tr>
</tbody>
</table>

Their function is described in program EC310. Other subroutines needed are

| RANDOM | Random number generator. Returns a floating point fraction between 0 and 1. |
| EXP    | ALOG SQRT COS EXIT |
|        | Standard FORTRAN calls |

3. INPUT DATA DECK

The input data deck format is given in table 23.
### TABLE 23. INPUT DATA DECK, PROGRAM EC572

<table>
<thead>
<tr>
<th>Card</th>
<th>Column(s)</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1-5</td>
<td>I5</td>
<td>NBAND - Number of energy bands in spectrum</td>
</tr>
<tr>
<td></td>
<td>6-10</td>
<td>I5</td>
<td>NT - Number of times points are to be calculated (time)</td>
</tr>
<tr>
<td></td>
<td>11-15</td>
<td>I5</td>
<td>NFLT - 0 = No Plot, 1 = Plot</td>
</tr>
<tr>
<td></td>
<td>16-20</td>
<td>I5</td>
<td>NPCH - 0 = No Card Output, 1 = Card Output</td>
</tr>
<tr>
<td></td>
<td>21-30</td>
<td>F10.5</td>
<td>A - Lower frequency limit (radians)</td>
</tr>
<tr>
<td></td>
<td>31-40</td>
<td>F10.5</td>
<td>B - Upper frequency limit (radians)</td>
</tr>
<tr>
<td></td>
<td>41-50</td>
<td>F10.5</td>
<td>DT - Seconds between each time of calculation</td>
</tr>
<tr>
<td>2</td>
<td>1-80</td>
<td>8F10.8</td>
<td>VI - The speed (U) at which the program is to be run. Any number of speeds up to eight can be used. Program will calculate only the number on this card.</td>
</tr>
</tbody>
</table>

### OUTPUT DATA

a. Printed

- NBAND: number of energy bands in spectrum
- NT: number of points
- NFLT: plot control
- NPCH: punch control
- A: lower frequency limit
- B: upper frequency limit
- DT: time interval
- V: wind velocity, knots
- U: wind velocity, ft/sec.
- GOU: \( g/U \)
- FACT1: \( 8.1 \times 10^{-3} g/U \)
- FACT2: \( 2.96g^2 \)
- YO: area below lower limit
- AY: area below upper limit
- AREA: area between limits
- AR: area in energy band
- X1: \( g/U (0.74/\ln(YO))^{1/2} \)
- Y1: frequency limit of each band (X1*FACT1)

1??
If CALCOMP plotter software is not available, the plotting calls must either be removed from the source deck or a dummy subroutine deck must be used.
c. Cards. If NPCH is set equal to 1, a card deck is punched with identifying data on the first card followed by cards with the card number and value of $T(t)$ in a $15, 5x, F10.4$ format. This output is for use in a PSD program.

d. A number of error messages and plot instructions are printed out on the computer operator's console to assist in running the program. If some other device number is used at another installation, the correct one will have to be used.

<table>
<thead>
<tr>
<th>Device Number</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Card Reader</td>
</tr>
<tr>
<td>3</td>
<td>Line Printer</td>
</tr>
<tr>
<td>2</td>
<td>Card Punch</td>
</tr>
<tr>
<td>15</td>
<td>Operator's Typewriter</td>
</tr>
</tbody>
</table>

5. LISTINGS AND FLOW CHART

The listings and flow chart for program E0572 are given in appendix A.
REFERENCES


APPENDIX A
PROGRAMMING LISTINGS AND FLOW CHARTS

This appendix contains the listings and flow charts for each of the computer programs described in this report. The page number for the start of each program is given in table 24.

<table>
<thead>
<tr>
<th>Program</th>
<th>Page</th>
<th>Program</th>
<th>Page</th>
<th>Program</th>
<th>Page</th>
<th>Program</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>EB920</td>
<td>127</td>
<td>ZC300</td>
<td>235</td>
<td>EC320</td>
<td>277</td>
<td>EC790</td>
<td>315</td>
</tr>
<tr>
<td>ZC790</td>
<td>193</td>
<td>ZC690</td>
<td>240</td>
<td>EC150</td>
<td>283</td>
<td>EC780</td>
<td>328</td>
</tr>
<tr>
<td>EC470</td>
<td>226</td>
<td>ZC691</td>
<td>244</td>
<td>EC330</td>
<td>287</td>
<td>EC572</td>
<td>353</td>
</tr>
<tr>
<td>ECL430</td>
<td>231</td>
<td>ECL410</td>
<td>248</td>
<td>EC310</td>
<td>294</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The programs are written in FORTRAN IV and should run on any computer equipped with this compiler. The higher level features of this language are not used and its programs are written to be as machine independent as possible. Each subroutine starts on a new page and the total program listing is followed by flow charts for that program.

The flow charts were generated from the FORTRAN listings. The program number, subroutine name, and page number of the flow chart are listed at the top of each page. The page number is used to connect the various charts together. Each input and output is labeled with a decimal number. The number to the left of the decimal point is the page number (upper right-hand corner) of the connection and the right-hand part gives the box number on that page. Subroutine calls are given in the subroutine box in the same manner. All flow chart symbols are conventional.
JOB ER960
EXEC FFORTRAN

DIMENSION Y(131), TII(12)
DIMENSION SAVE(300,16), ILGC(16), BUFF(3000)
REAL IX, IY, IZ, IXY, IYZ

COMMON H, HMAX, HMIN, DH, FCT, TL, NGS, N, IS1, NPNT

COMMON TLIM, RHOL2, RHOL3, RHOL4, RHOL5, WMB, ETA, ETAM1, ISW2

COMMON XQQ, XRR, XRQ, XUD, XVR, XWQ, XIU, XVV,
XWW, XORDR, XSOS, XDBDA, XVE, XWFE, XORDRE, XSDOS

COMMON YRD, YPD, YPAD, YPR, YQR, YVD, YVQ, YWP,
YWR, YR, YP, YARDR, YYAR, YSTR, YY, YVAV,
YVV, YDR, YRF, YVF, YVAVE, YDRE

COMMON ZQD, ZPP, 7RR, ZRP, ZWD, ZVR, ZVP, ZQ,
ZADS, ZWAQ, ZSTR, ZW, ZWAN, ZAW, ZWW, ZVV,
ZDS, ZDB, ZQE, ZWE, ZWAFE, ZDSE

COMMON AKPD, AKRD, AKQR, AKPQ, AKPAP, AKP, AKR, AKVD,
AKVQ, AKWP, AKWR, AKSTR, AKV, AKVAV, AKVW, AKDR,
AKSTRF

COMMON AMQO, AMPP, AMRP, AMQ, AMPAQ, AMWH, AMMR, AMVP,
AMQ, AMQQOS, AMARQ, AMSTR, AMW, AMWAQ, AMAW, AMWV,
AMV, AMDS, AMDR, AMQE, AMWF, AMWAWE, AMDSE

COMMON ANRD, ANPO, ANPQ, ANQR, ANRAR, ANVR, ANWD, ANWR, ANWP,
ANVQ, ANP, ANR, ANARD, ANAVR, ANSTR, ANV, ANVAV,
ANVW, ANDR, ANRE, ANVE, ANVAVF, ANDRE

COMMON IX, IY, IZ, IXY, IXY, IYZ

COMMON CW, CR, UC, XR, YB, ZB

COMMON DR, DS, DR, RHO, AL, AM

COMMON DMAX, ETAHI, FFTAL1, A11, A12, A13

COMMON A21, A22, A23, A31, A32, A33

COMMON XG, YG, ZG

COMMON ILGC, IPILOT, IRUIN, IOPFN, NPLT, IOPT

COMMON Y

COMMON TIME, RI, DFLTMA, SWMAX, R2, DELTMI, DSF, DRF, ICYCY, NS,
INTSW

PI = 3.141593
IOPT = 0

127
IOPEN = 0
N = 17
46 CONTINUE
CALL INPUT
IF(IPLNT)44, 50, 48
48 IF(IOPEN) 50, 49, 50
49 CALL PLOTS(NUFF, 12000, 7)
IOPEN = 1
50 CONTINUE
NLOC=16
K = 0
IOUT = 3
LNSPP=50
LINS=99
IPAGE=1
WRITE(IOUT, 241) IPAGE
ICNT = NNPNT
ICNT? = NPLT
C
C COMPUTE RHO * L CONSTANTS
C
RHOH = RHO * .5
RHOH2 = RHOH * AL * AL
RHOH3 = RHOH2 * AL
RHOH4 = RHOH3 * AL
C
WRITE OUT HYDRODYNAMIC COEFFICIENTS
C
WRITE(IOUT, 11) XQP, YRP, ZQP, AKP, AMQP, ANRP, XPR, YPR, ZPR
ACRP, AMRP, ANRP, XVP, YVR, ZVR
C
WRITE(IOUT, 11) XQP, YRP, ZQP, AKP, AMQP, ANRP, XPR, YPR, ZPR
C
WRITE(IOUT, 11) XQP, YRP, ZQP, AKP, AMQP, ANRP, XPR, YPR, ZPR
C
WRITE(IOUT, 11) XQP, YRP, ZQP, AKP, AMQP, ANRP, XPR, YPR, ZPR
C
WRITE(IOUT, 11) XQP, YRP, ZQP, AKP, AMQP, ANRP, XPR, YPR, ZPR
C
WRITE(IOUT, 11) XQP, YRP, ZQP, AKP, AMQP, ANRP, XPR, YPR, ZPR
C
WRITE(IOUT, 11) XQP, YRP, ZQP, AKP, AMQP, ANRP, XPR, YPR, ZPR
C
WRITE(IOUT, 11) XQP, YRP, ZQP, AKP, AMQP, ANRP, XPR, YPR, ZPR
C
WRITE(IOUT, 11) XQP, YRP, ZQP, AKP, AMQP, ANRP, XPR, YPR, ZPR
C
WRITE(IOUT, 11) XQP, YRP, ZQP, AKP, AMQP, ANRP, XPR, YPR, ZPR
C
WRITE(IOUT, 11) XQP, YRP, ZQP, AKP, AMQP, ANRP, XPR, YPR, ZPR
C
WRITE(IOUT, 11) XQP, YRP, ZQP, AKP, AMQP, ANRP, XPR, YPR, ZPR
C
WRITE(IOUT, 11) XQP, YRP, ZQP, AKP, AMQP, ANRP, XPR, YPR, ZPR
C
WRITE(IOUT, 11) XQP, YRP, ZQP, AKP, AMQP, ANRP, XPR, YPR, ZPR
C
WRITE(1OUT,5) IRUN
      5 FORMAT(1H,'RUN NC ',15/)
C
C COMPUTE W-B
C
WMB=CW+CR
IS1 = 0
ISW2 = 0
IF(INTSW) 60, 67, 62
60 INTSW = 1
62 IF(INTSW-3)=66, 66, 64
64 INTSW = 1
66 INTGSW=1
GO TO 9
67 INTGSW=0
9 CONTINUE
IF(INTGSW) 69, 68, 69
69 CALL KUTTA(Y)
GO TO 75
68 CALL INTEGRAL(Y,INTSW)
75 CONTINUE
C
C SAVE VALUES FOR PLOTS IF IPLIM = 1
C
IF(IPLIM) 9, 12, 9
9 CONTINUE
C
IF ARRAY FULL DENT OVERUN
C
IF(K=300) 63, 12, 12
63 CONTINUE
IF(NPLT-ICNT2) 52, 52, 51
52 ICNT2 = 0
K = K + 1
C
SAVE TIME
SAVE(K,1) = Y(1)
C
SAVE U
SAVE(K,2) = Y(1)
C
SAVE V
SAVE(K,3) = Y(2)
C
SAVE W
SAVE(K,4) = Y(3)
C
SAVE P
SAVE(K,5) = Y(4)
C
SAVE Q
SAVE(K,6) = Y(5)
C
SAVE R
SAVE(K,7) = Y(6)
C
SAVE THETA
SAVE(K,9) = Y(7)
C
SAVE PSI
SAVE(K,9) = Y(8)
C
SAVE PHI
SAVE(K,10) = Y(9)
C  SAVE X
   SAVE(K,11) = Y(10)
C  SAVE Y
   SAVE(K,12) = Y(11)
C  SAVE Z
   SAVE(K,13) = Y(12)
C  SAVE DR
   SAVE(K,14) = DR
C  SAVE DS
   SAVE(K,15) = DS
C  SAVE DR
   SAVE(K,16) = DR
51  ICNT2 = ICNT2 + 1
12  CONTINUE
   IF(INPNT-ICNT) 16,16,17
16  ICNT = 0
   IF(LINSPP-LINS)20,20,30
20  LINS=0
   IPAGE=IPAGE+1
   WRITE(IOUT,24)IPAGE
24  FORMAT(1X,'ER920',30X,'SURMARINE SIMULATION',45X,'PAGE',18/)
   WRITE(IOUT,25)
25  FORMAT(1X,'G0',1X,'U',16X,'V',16X,'W',16X,'P',16X,'Q',16X,'R',
   1 16X,'THETA'/9X,'PSI',14X,'PHI',15X,'X',16X,'Y',15X,'Z',16X,'T',
   2 16X,'H')
30  CONTINUE
C  IF VARIABLE STEP SIZE NOT BEING USED DONT PRINT H
C  IF(CH170,72,70
70  WRITE(IOUT,10) Y(I),i=1,13)
   GO TO 74
74  WRITE(IOUT,10) Y(I),i=1,13
10  FORMAT(140,7(2X,F13.6,2X),12X,7(2X,F13.6,2X))
74  CONTINUE
   LINS=LINS+1
17  ICNT = ICNT + 1
   IF(ABS(Y(13)-TL14)-H135,a,a
35  CONTINUE
   IF(IPLOT) 4C,46,40
40  CALL PLTRM(SAVE,K,1LOC,NEOF,ITRM)
   GO TO 45
END
SUBROUTINE INPUT

DIMENSION Y(13), TL(12), ILOC(16), YHOLD(13), COM(219)
EQUIVALENCE (COM(1),H)
REAL IX, IY, IZ, IXY, IX7, IY2

COMMON H, HMAX, HMIN, DH, FCT, TL, NGS, N, IS1, NPNW

COMMON TLIM, RHOL2, RHOL3, RHOL4, RHOL5, WMB, FTA, ETA1, ISW2

COMMON XQC, XRR, XQP, XUD, XVP, XWD, XUH, XV, XW, XDPKR, XDSQ, XDBDR, XVFF, XWFF, XDFDRE, XDSDF

COMMON YRD, YPD, YPAQ, YPD, YQR, YVO, YVQ, YWP,
YVW, YDR, YRE, YVE, YVAF, YDRF

COMMON ZOR, ZPP, ZPP, ZWD, ZVP, ZVP, ZV,
ZADS, ZAQQ, ZSRT, ZH, ZHAW, ZAW, ZAW, ZVV,
ZDS, ZDB, ZOE, ZWF, ZWAF, ZDSE

COMMON AKPD, AKRD, AKOR, AKPP, AKPAQ, AKP, AKR, AKVP
AKKQ, AKWP, AKK, AKK, AKV, AKK, AKVW, AKK, AKR
AKKRF

COMMON AMON, AMPB, AMRR, AMRP, AMQAQ, AMWD, AMVP, AMVP,
AMQ, AMAQDS, AAMAWQ, AMSTR, AMW, AAMAWQ, AMAW, AMW,
AMW, AMW, AMWQ, AMOF, AMW, AMWAF, AMWQ

COMMON ANRD, ANPD, ANPQ, ANQR, ANAR, ANVD, ANWR, ANWP,
ANVQ, ANP, ANP, ANAPRD, ANAVR, ANSTR, ANV, ANRV
ANVW, ANDR, ANPE, ANRE, ANVE, ANVAF, ANRE

COMMON IX, IY, IZ, IXY, IX7, IY7

COMMON CW, CR, UC, X8, Y8, 78

COMMON OR, OS, OR, ORH, ORL, ORM

COMMON ORMAX, ETAHI, ETAIN, A11, A12, A13

COMMON A27, A27, A27, A27, A27

COMMON XG, YG, 7G

COMMON ILOC, IPLST, IRUN, IOPEN, NPLT, NPT

COMMON Y

COMMON TIME, RI, DELTMA, SWMAX, R2, DELTMI, OSF, OPE, IVC, NS,
1 INTSW

IN = 1
IF (INTPT) 150, 150
5 CONTINUE
READ(IN,50) NGS, NPNT, IPRINT, NPLT, IOPT, ICYC, NS, NTSW
IF(IRUN)70,60,70
60 IF(IOPT) 62,64,62
62 CALL PLOT(5.0,0.0,999)
64 CONTINUE
CALL EXIT
70 CONTINUE
READ(IN,50) (ILCC(I), I = 1, 16)
50 FORMAT(16I5)
READ(IN,100) TO, HO, HMAX, HMINT, FCT, TLM
H = HO
100 FORMAT(9F10.5)
READ(IN,100) (TL(I), I=1,12)
READ(IN,100) (Y(I), I=1,12)
Y(13) = TO
C READ(IN,100) XQQ, XRR, XRP, XUD, XVR, XWQ, XUU, XVV,
1XWW, XORDR, XDSDS, XDBDR, XVVE, XWVE, XORDRE, XDSDSE
C READ(IN,100) YRD, YPD, YPA, YRQ, YOR, YVD, YVQ, YWP,
1YW, YV, YRE, YVF, YYAV, YYAV
2 YYV, YDR, YRE, YVF, YYAVE, YDFR
C READ(IN,100) ZOD, ZPP, ZRR, ZRP, ZWD, ZVR, ZVP, 70,
1ZQDS, ZWAQ, ZSTR, ZW, ZWAW, ZAW, ZWW, ZVV,
2 ZDS, ZDR, ZOF, ZWE, ZWAWE, ZDSF
C READ(IN,100) AKPD, AKRD, AKQR, AKPQ, AKPAP, AKP, AKR, AKVD,
1AKVQ, AKWP, AKWR, AKSTR, AKV, AKVAV, AKVW, AKDR,
2 AKSTRE
C READ(IN,100) AMON, AMPP, AMRP, AMRP, AMQAC, AMWD, AMVR, AMVP,
1AMQ, AMAODS, AMAWQ, AMSTR, AMW, AMWAW, AMAW, AMWW,
2 AMVV, AMDS, AMDR, AMQE, AMWE, AMWAWF, AMDSF
C READ(IN,100) ANRO, ANPO, ANPQ, ANQR, ANRDR, ANVR, ANWP,
1ANVQ, ANP, ANR, ANARDP, ANAVR, ANSTR, ANV, ANVAV,
2 ANVW, ANDR, ANRE, ANVE, ANVAVE, ANDRF
C READ(IN,100) IX, IY, IZ, IXY, IZI, IYZ
C READ(IN,100) CW, CR, UC, XB, YB, 79
C READ(IN,100) DP, DS, DB, RHO, AL, AM
C READ(IN,100) DMAX, ETAH, FTALN, A11, A12, A13
C READ(IN,100) A21, A22, A23, A31, A32, A33
C READ(IN,100) XG, YG, ZG
C READ(IN,100) TIME, PI, DELTMA, SWMAX, R2, DELTMI, DSE, DRE
C SAVE INITIAL CONDITIONS FOR POSSIBLE RESTORE
C
DO 110 I=1, 13
   YHOLD(I) = Y(I)
110 CONTINUE

SAVE DR, DS, DB FOR POSSIBLE RESTORE

DRHOLD = DR
DSHOLD = DS
DBHOLD = DB
RETURN

150 CONTINUE

RESTORE INITIAL VALUES

DO 152 I=1, 13
   Y(I) = YHOLD(I)
152 CONTINUE

RESTORE INITIAL DR, DS, DB, H

DP = DRHOLD
DS = DSHOLD
DB = DBHOLD
H = H0
READ(IN, 165) IRUN
IF(IRUN) 155, 05, 155
155 CONTINUE

160 READ(IN, 165) NDEX, VALUE
165 FORMAT(5, 5X, F10.5)
IF(NDEX) 140, 170, 140
170 RETURN

140 COM(NDEX) = VALUE
   GO TO 160
END
SUBROUTINE KUTTA(Y)
DIMENSION Y(I), F(12), Y2(13), Q(12), TL(12)
COMMON H, HMAX, HMIN, DH, FCT, TL, NGS, N, IS
IF(IS1=1,1,2)
  1 N = N + 1
  K = 0
  IS1 = 1
  DH4 = DH * DH * DH * DH
  CALL EVAL(Y,F)
  RETURN
  2 DO 3 I = 1,N
  3 Q(I) = F(I)
        Y2(M) = Y(M) * .5 * H
        CALL EVAL(Y2,F)
        DO 4 I = 1,N
        Y2(I) = Y(I) * .5 * H*F(I)
      4 CALL EVAL(Y2,F)
        DO 5 I = 1,N
        Y2(I) = Y(I) * .16666667 * H * (Q(I) + F(I))
      5 Q(I) = F(I)
        CALL EVAL(Y2,F)
        IF(DH)17,13,7
      6 IF(DH)17,13,7
        Q(I) = H * ABS(F(I) - Q(I))
        IF(Q(I) - TL(I))9,9,15
      8 CONTINUE
        DO 9 I = 1,N
        Q(I) = FCT * DH4 * Q(I)
        IF(Q(I) - TL(I))9,9,12
      9 CONTINUE
        K = K + 1
        IF(K - NGS)13,10,10
      10 IF(H - HMAX)11,12,12
        H = H * DH
      12 K = 0
      13 DO 14 I = 1,N
      14 Y(I) = Y2(I)
                Y(M) = Y2(M)
                PRETURN
      15 IF(H-HMIN)12,12,16
      16 H = H/DH
        CALL EVAL(Y,F)
        GC TO 2
        END
SUBROUTINE EVAL(YI,F)
DIMENSION YI(1),F(1),TL(12),A(6,6),R(6)
REAL IX,IY,IZ,IXY,IYZ

COMMON H, HMAX, HMIN, DH, FCT, TL, NGS, N, IS1, NPNT
COMMON TLIM,200

COMMON COMMON
XQQ, XRR, XRP, XUP, XVR, XUR, XVV,
1 XWW, XORDR, XSDS5, XORDA, XVVE, XWWE, XORDSE

COMMON YRD, YPD, YPAR, YPR, YOR, YVD, YVQ, YWP,
YWR, YR, YP, YARDP, YVAR, YSTR, YV, YVAV,
2 YVV, YDR, YRF, YVF, YVAVE, YDFR

COMMON ZQD, ZPP, ZRR, ZRP, ZWD, ZVR, ZVP, ZO,
ZAOQS, ZWAO, ZSTP, ZW, ZWAY, ZAW, ZWW, ZVV,
2 ZWS, ZOA, ZQF, ZWE, ZWAYF, ZDFS

COMMON AKPD, AKRD, AKQR, AKPO, AKPAP, AKP, AKR, AKVD,
AKVQ, AKWP, AKWR, AKSTR, AKV, AKVAV, AKVH, AKOR,
2 AKSTRE

COMMON AMOD, AMP, AMPP, AMPR, AMQ, AMQO, AMWD, AMVP, AMV,
1 AMQ, AMQDS, AMAQO, AMSTR, AMW, AMWAW, AMWH, AMWH,
2 AMVV, AMDS, AMDR, AMOE, AMWE, AMAWF, AMDFS

COMMON ANRD, ANPD, ANPO, ANQR, ANRQ, ANMR, ANDR, ANV, ANWR, ANWP,
ANVO, ANP, ANP, ANMDR, ANAVR, AMANS, AMN, ANVAV,
2 ANVW, ANDR, ANRE, ANVF, ANVAVE, ANDFR

COMMON IX, IY, IZ, IXY, IXZ, IYZ

COMMON CW, CR, UC, XR, YR, 7R

COMMON DR, DS, DR, RHO, AL, AM

COMMON DMAX, ETA, ETAI, ETAI, AI1, AI2, AI3

COMMON A11, A22, A33, A11, A22, A33

COMMON XG; YG, ZG

EQUVALENCES(R(1),FA),(R(2),FL),(R(3),FN),(R(4),FO),(R(5),FW), (R(6),FY).
1 (R(6),FY)

PULL PRESENT VALUES OF VARIABLES C:IT OF ARRAY YI

U = YI(1)
V = YI(2)
W = YI(3)
P = YI(4)
Q = YI(5)
R = YI(6)
THETA = YI(7)
PSI = YI(8)
PHI = YI(9)
X = YI(10)
Y = YI(11)
Z = YI(12)
CALL CONTRITHETA)

C COMPUTE QUANTITIES TO BE USED MORE THAN ONCE
C
VR = V*R
Q2 = Q*Q
P2 = P*P
R2 = R*R
U2 = U*U
V2 = V*V
W2 = W*W
DR2 = DR*DR
DR2U2 = DR2 * U2
DS2 = DS*DS
DS2U2 = DS2 * U2
WP = W*P
UR = U*R
PABSP = P*ABS(P)
PQ = P*Q
QR = Q*R
VQ = V*Q
WQ = W*Q
WR = W*R
UP = U*P
RONTW = SQRT(V2+W2)
VRTW = V*ROOTW
ABSR = ABS(R)
UARDR=U*ABSP*DP
UV=U*V
VW=V*W
PRU2=DR*U2
UQ=U*Q
VP=V*P
P2=P*P
ABSC=ABS(Q)
UAQDS=U*ABSQ*DS
WRTW=WP*ROTW
UW=U*W
AASW=ABS(W)
UASW=U*AASW
SU2=DS*U2
DSU2=DS2*U2
ABSV = ABS(V)
UMAG = SQRT(U2+V2+W2)
IF (UMAG>24,24,24)
24 FTA = 20.
GO TO 29
29 CONTINUE
ETA = UE/UMAG
28 ETAM1 = ETA-1.
   IF(ETA-ETAM1)32,30,30
30 A1=A11
A2=A12
A3=A13
   GO TO 39
32 IF(ETA-ETAM1)35,35,37
35 A1=A31
A2=A32
A3=A33
   GO TO 38
37 A1=A21
A2=A22
A3=A23
38 CONTINUE
   IF(V)7,1,2
1 RATVAV = 0.
   GO TO 2
2 RATVAV = V/ABSV
3 IF(W)5,4,5
4 RATWAV = 0.
   GO TO 6
5 RATWAV = W/ABSW
6 CONTINUE
C
C COMPUTE TRIG FUNCTIONS
C
SPHI = SIN(PHI)
CPHI = COS(PHI)
STTA = SIN(THETA)
CTTA = COS(THETA)
SPSI = SIN(PSI)
CPSI = COS(PSI)
TRIG = CTTA*SPHI
TRIG2 = CTTA*CPHI
TRIG3 = SPHI*STTA
TRIG4 = CPHI*STTA
TRIG5 = H*CTTA
37 ISW=1
C
C SET COEFFICIENTS OF UD,VD,W,D,PD,CD,PD, IN MATRIX FOR INVERTING
C
A[1,1] = AM-RHOU*(EXUM)
A[1,2] = 0.
A[1,3] = 0.
A[1,4] = 0.
A[1,5] = AM*7G
A[1,6] = -AM*YG
A[2,1] = 0.
A[2,2] = AM-RHUP*UVU
A[2,3] = 0.
A[2,4] = -RHUP*UVU-AM*7G
A[2,5] = 0.
A[2,6] = 0.
A(2,5) = 0.
A(2,6) = -RHOL4*YRN+AM*XG
A(3,1) = 0.
A(3,2) = 0.
A(3,3) = AM-RHOL3*ZWD
A(3,4) = AM*YG
A(3,5) = -RHOL4*ZQD-AM*XG
A(3,6) = 0.
A(4,1) = 0.
A(4,2) = -RHOL4*AKVD-AM*ZG
A(4,3) = AM*YG
A(4,4) = IX-RHOL5*AKPD
A(4,5) = -IXY
A(4,6) = -IXZ-RHOL5*AKPD
A(5,1) = AM*ZG
A(5,2) = 0.
A(5,3) = -RHOL4*AMWD-AM*XG
A(5,4) = -IXY
A(5,5) = IY-RHOL5*AMQD
A(5,6) = -IXZ
A(6,1) = -AM*YG
A(6,2) = -RHOL4*ANVD+AM*XG
A(6,3) = 0.
A(6,4) = -IXZ-RHOL5*ANPD
A(6,5) = -IXZ
A(6,6) = IZ-RHOL5*ANRD

C INVERT A MATRIX

FPS = .000001
CALL INVER2(A,6,4,6,IER,EPS)
IF(IER) 14,16,14
14 FORMAT(1H1, 'SINGULAR MATRIX')
CALL EXITD
16 CONTINUE

C COMPUTE RIGHT SIDE OF AXIAL FORCE FON

C 40 CONTINUE
FA = (AM*(VR-WQ) + RHOL4*(XQQ*Q2+XRR*R7+XRP*RP) +
1RHO LCDi (XVR*VF+XWO*WO)*RHOL2*(XUJ+U2+YV*YQ+XWW*W7) +
2RHOL2*(XWO*WO)+XDS+S52+XRD3+XRD4) +
3RHOL2*(XW*W7+XPP*PP+XCS+S52+XDP6+XPD7)*STTA +
4XQ*Q2*Q2+XDS+S52+XDP6+XPD7)*FTAM1) +
5*(AM*(VG*(Q2+Q2)-VG*PO-ZG*RP))

C COMPUTE RIGHT SIDE OF LATERAL FORCE FON

C 50 CONTINUE
FL = AM*(WP-UR)+RHOL4*(YPAP+PARSP+YQQ+PO+YQQ+OR) +
1RHO L3*(YQQ*QQ*WP=W+YWR=WR+YW+IP+Y=IP+YAR+D=R+D*D+DP


NAVTRADEVSCN 68-C-LG50-2

?YVAR*YAVTAV*ROOTVW*ABSR) + RHOL2*(YSTR*U)+YV*U+YVAV*VRTVW + YVAV*VR + YRDRVU) + WMB*TRIG1 + RHOL3*YRE*UP*FTIM1 + RHOL2*(YVE*UV+YVAVE*VRTVW+YDFR*DRU) + FTIM1

? + AM*(YR*(2P2)-ZG*Q+XG*PQ)

C COMPUTE RIGHT SIDE OF NORMAL FORCE EON

60 CONTINUE

FN = AM*(UQ-VP) + RHOL4 *(ZPP*P+7RR*V+ZRP*P) + RHOL3 *(7VR*VR+ 1ZVP*VP + ZQ*UQ + 7AODS* UQDS + ZWAO* RATWAW+RONTVW+AMSO) + ZHOL2*(ZSTR*U2 + ZH*UW+ZWAV + WRTTW + ZAW*AW + ZWAV*AW + WROONTVW

C+ ZV*V+ZDS*DSU + ZDR*PQ(J) + WMB + TRIG2 + RHOL3 * ZOF*UQ* FTIM1

4+ RHOL2*(ZVF*UW+ZWAVF*WRTTW+ZDS*USU) + FTIM1

4 + AM*(VQ*(D+Q)+XG-RP-YG*QR)

C COMPUTE RIGHT SIDE OF ROLLING MOMENT EON

70 CONTINUE

RM = (lY-1Z)*QR+IY7*PO+1YZ*(02-Q2)-IYY*RP+RHOL5*(AKQR+OR + 1AKPQ*PO+AKPAP*APSP)+RHOL4*(AKP*UP + AKUR+AKP*VQ + AKWP+WP + AKWR*W) + RHOL3*(AKST+R2*Q2 + AKVR+AKF*VTP+WAV+AKDR+DRU) 3+(YG*W+Y*CR)*TRIG2-(ZG*CW-ZB*C3)*TRIG1 + RHOL3*AKSTRF*12*FTIM1

4 + AM*(VQ*(D+Q)+XG-RP-YG*QR)

C COMPUTE RIGHT SIDE OF PITCHING MOMENT EON

90 CONTINUE

PM = (1Z-1X)*RP+IY7*QR+(P2-Q2)*IY7-1YZ*PO + RHOL5*(APMD*P2+APMD* 1R2 + ANRP*RP + ANAOQ*Q*ARQS) + RHOL4*(AMVR*VR + AMVP+VP + AMQ*UQ + AMQ*Q + 2AUQDS + AMAWQ + Q*RONTVW) + RHOL3*(AMSTR+U2+AMWE+UW+AMWE*WRTTW + 3AMAW*U+AMAW + AMWE+UW+AMWE+WRTTW + AM117+IN12+ANDR)+DRU) 4-(XG*CW-XSP*CR)*TRIG2-(YQ*CW-ZP*C3)*STTA + RHOL4*AQ*UQ*FTIM1

5+RHOL3*AMWE*UW+AMWE*WRTTW + AMODE*NSU 1 + FTIM1

6 + AM*(VQ*(D-P+Q)+XG*(UP-JQ))

C COMPUTE RIGHT SIDE OF YAWING MOMENT EON

90 CONTINUE

YM = (1X-1Y)*PC+1Y7*RP +(P2-Q2)*IY7-QP + RHOL5*(ANPQ= 0P + ANDP*QR + ANAR*APR*APSR) + RHOL4*(ANMR*WR+AMDP*WP + ANV*U7) + 2ANUP*UP + ANVR + ANARP + UAR + ANVR + RRONTTW) + RHOL3*(ANSTR+U2+ANV*UV+ANVAF+VTP+WAV+ANWE*UQ + AMODURU) + 3+(XG*CW-X*CR)*TRIG2-(YQ*CW-Y*C3)*STTA + RHOL4*ANF*UQ*FTIM1 + RHOL3*(ANF*IV+ANVAF*VTP+WAV + ANSF*USU) + FTIM1

6 + AM*(VQ*(D-P+Q)+XG*(UP-JQ))

C MULTIPLY TO GET UD, UD, WD, WD, WD, WD, WD

CALL VATMIXP(A,AR,F,A,F,1,A,F,1)

C COMPUTE KINETICS - DTETA DOT, PSI DOT, PHI DOT

F(7) = COSPHI-RESUM
F(8) = COSPHI+RESUM/CTTA
F(9) = P + F(8) * STTA

C COMPUTE X NOT, Y NOT, Z NOT

F(10) = TRIG5 * CPSI + V * (TRIG3 * CPSI - CPSH * SPSI) +
       1. W*(TRIG4 * CPSI + CPSH * SPSI)
F(11) = TRIG5 * SPSI + V * (TRIG3 * SPSI + CPSH * CPSI) +
       1. W*(TRIG4 * SPSI - CPSH * CPSI)
F(12) = -U * STTA + V * TRIG1 + W * TRIG2
RETURN
END
SUBROUTINE INVER2(A,M,N,L,IC,FPS)
C A THE INPUT MATRIX CONTAINS THE INVERTED MATRIX A-1 UPON EXIT
C THE SOLUTION IS STORED IN A (I,J) FOR EACH EQUATION
C M IS THE NUMBER OF ROWS STORED IN MATRIX A.
C N IS THE NUMBER OF COLUMNS STORED IN MATRIX A.
C L IS THE MAX NUMBER OF ROWS ALLOCATED IN A.
C IC IS VALUE XX IN DIMENSION A(XX,YY)
C IC IS 0 IF A IS INVERTED SUCCESSFULLY, IF EACH DIAGONAL ELEMENT
C IS GREATER IN ARS THAN FPS.
C IC IS 1 IF A IS NOT INVERTED SUCCESSFULLY.
C FPS IS A VALUE SAY .00001 THAT IS USED FOR SINGULARITY CHECKING.
C FOR A DOUBLE PRECISION VERSION CHANGE ARS TO DABS IN S, AND THE
C LITERAL 1. IN STATEMENT 10 TO 1.00
D IMENSION A(1)
C DOUBLE PRECISION A
IC = 0
DO 80 I=1,M
LI = L*I-L
L1 = L1+1
5 IF(ABS(A(L1))<FPS)90,90,10
10 A(L1) = 1./A(L1)
DO 50 K = 1,N
IF(K-I)20,50,20
20 LK = L*K-L
IK = LK+1
A(IK) = A(IK)*A(L1)
DO 40 J=1,M
IF(J-I)3040,30
30 J1 = L1+J
JK = LK+J
A(JK) = A(JK) - A(J1)*A(IK)
40 CONTINUE
50 CONTINUE
DO 70 J=1,M
IF(J-I)60,70,60
60 J1 = L1+J
A(J1) = - A(J1)*A(L1)
70 CONTINUE
80 CONTINUE
RETURN
IC = 1
RETURN
END
SUBROUTINE MATMPY (A, B, C, M, N, L, MA, MB, MC, IOPT)
DIMENSION A(M), B(N), C(L)

DOUBLE PRECISION A, B, C

KK = -MB
II = -MC
LLL = 0
30 LLL=LLL+1
II=II+MC
I = II
KK = KK+MB
III=0
JJ = -MA+1
40 K=KK
J=JJ
I=I+1
KKK=0
III=III+1
C(I)=0.
50 K=K+1
J=MA+J
KKK=KKK+1
C(I)=C(I)+A(J)*B(K)
60 JJ=JJ+1
IF(III-M)40, 70, 70
70 IF(LLL-L)30, 80, 80
90 RETURN
END
SUBROUTINE INTFG(Y,INTSW)
DIMENSION Y(1), F(12), TL(12), F1(12)
COMMON H, HMAX, HMIN, DH, FCT, TL, NGS, N, IS1
IF(IS1)30, 10, 30
10 M= N + 1
DO 20 I = 1,N
F(I) = 0.
20 F1(I) = 0.
IS1 = 1
RETURN
30 CALL FVAL1(Y,F)
DO 40 I = 1,N
GO TO (42,44,4A),INTSW
42 CONTINUE
C         Y(I) = Y(I) + .5*H*(3.*F(I)-F1(I))
C         GO TO 48
44 CONTINUE
C         Y(I) = Y(I) + .25*H*(3.*F(I)+F1(I))
C         GO TO 48
46 CONTINUE
C         Y(I) = Y(I) + H * F(I)
C         49 CONTINUE
F1(I) = F(I)
40 CONTINUE
Y(M) = Y(M)+H
RETURN
END
SUBROUTINE EVALI(YI,F)
DIMENSION YI(11),F(11),TL(12)
REAL IX,IX,Y,IY,IXZ,IVZ
COMMON H, HMAX, HMIN, DH, FCT, TL, NGS, N, IS1, NPNT
COMMON TLIM, RHOL2, RHOL3, RHOL4, RHOL5, WMB, FTA, FTAM1, ISW2
COMMON XQC, XRR, XRP, XUD, XVR, XWO, XU, XVV, XWW, XORDR, XSDSD, XDBDR, XVVF, XWWE, XRDR, XSDSE
COMMON YRD, YPD, YPAP, YPQ, YQR, YVD, YVQ, YWP, YVR, YR, YP, YARDR, YVAR, YSTR, YYV, YYVAV, YWW, YDR, YRF, YVE, YYAVE, YDRF
COMMON ZQG, ZPP, ZRR, ZRP, ZWD, ZVP, ZVP, ZQ, ZAQDS, ZWAQ, ZSTR, ZW, ZMAW, ZAW, ZWW, ZUV, ZDS, ZDR, ZQE, ZWE, ZWAWE, ZDSE
COMMON AKPD, AKRD, AKQR, AKPQ, AKPAP, AKP, AKR, AKVD, AKVQ, AKWP, AKWR, AKSTR, AKV, AKVAV, AKVW, AKDR, AKSTRF
COMMON AMQD, AMPP, AMRR, AMPP, AMQAO, AMWD, AMVP, AMVP, AMQ, AMAQDS, AMAWQ, AMSTR, AMW, AMWAW, AMAW, AMWW, AMVV, AMDS, AMDR, AMQE, AMWF, AMWAWF, AMDF
COMMON ANRD, ANPD, ANPQ, ANQR, ANRQR, ANVR, ANWR, ANWP, ANVQ, ANP, ANR, ANARDR, ANAVR, ANSTR, ANV, ANVAV, ANVW, ANDR, ANRF, ANVE, ANVAVE, ANDRE
COMMON IX, IX, IZ, IXY, IXZ, IY
COMMON CW, CR, UA, XB, YB, ZB
COMMON DR, DS, IH, RH, AL, AM
COMMON CRMAX, ETAHI, ETALO, A11, A12, A13
COMMON A21, A22, A23, A31, A32, A33
COMMON XG, YG, ZG

PULL PRESENT VALUES OF VARIABLES OUT OF ARRAY YI
U = YI(1)
V = YI(2)
W = YI(3)
P = YI(4)
Q = YI(5)
R = YI(6)
THETA = YI(7)
PSI = YI(8)
\[ \phi_i = y_i(q) \]
\[ x = y_i(10) \]
\[ y = y_i(11) \]
\[ z = y_i(12) \]
\[ \text{CALL CONTR(\theta)} \]

C

\text{COMPUTE QUANTITIES TO BE USED MORE THAN ONCE}

C

\[ v_r = v \cdot r \]
\[ q_2 = q \cdot q \]
\[ r_2 = r \cdot r \]
\[ r_p = r \cdot p \]
\[ u_2 = u \cdot u \]
\[ v_2 = v \cdot v \]
\[ w_2 = w \cdot w \]
\[ d_r^2 = d_r \cdot d_r \]
\[ d_r 2 u_2 = d_r^2 \cdot u_2 \]
\[ d_s^2 = d_s \cdot d_s \]
\[ d_s 2 u_2 = d_s^2 \cdot u_2 \]
\[ w_p = w \cdot p \]
\[ u_r = u \cdot r \]
\[ p_a b_s^p = p \cdot a b s(p) \]
\[ p_q = p \cdot q \]
\[ c_r = q \cdot r \]
\[ v_q = v \cdot q \]
\[ w_q = w \cdot q \]
\[ w_r = w \cdot r \]
\[ u_p = u \cdot p \]

\[ \text{ROOTVW} = \text{SORT(V2+W2)} \]
\[ \text{VRTVW} = v \cdot \text{ROOTVW} \]
\[ a b s_r = a b s(p) \]
\[ u a d r = u \cdot a b s_r \cdot d_r \]
\[ u v = u \cdot v \]
\[ v w = v \cdot w \]
\[ d_r u_2 = d_r \cdot u_2 \]
\[ u c = u \cdot q \]
\[ v p = v \cdot p \]
\[ p_2 = p \cdot p \]
\[ a b s_q = a b s(q) \]
\[ u a d s = u \cdot a b s_q \cdot d_s \]
\[ w r t v w = w \cdot \text{ROOTVW} \]
\[ u w = u \cdot w \]
\[ a b s_w = a b s(w) \]
\[ u a b s w = u \cdot a b s_w \]
\[ c b u_2 = c b \cdot u_2 \]
\[ n s u_2 = n s \cdot u_2 \]
\[ a b s_v = a b s(v) \]
\[ u m a g = \text{SORT(U2+V2+W2)} \]
\[ i f(u m a g > 24, 76) \]

24 FTA = 79.
26 CONTINUE
28 FTA = UC/UMAG.
29 FTAM1 = FTA-1.
ETAMIL = ETA-1.

IF(ETA-ETAHI)32, 30, 30
30 A1 = A11
  A2 = A12
  A3 = A13
  GO TO 38
32 IF(ETA-ETALO)35, 35, 37
35 A1 = A31
  A2 = A32
  A3 = A33
  GO TO 38
37 A1 = A21
  A2 = A22
  A3 = A23
38 CONTINUE:
  IF(V) ?, ?, 2
  1 RATEAV = 0.
     GO TO 3
  2 RATEAV = V/ABSV
  3 IF(W) 5, 4, 5
  4 RATEAW = 0.
     GO TO 6
  5 RATEAW = W/ABSW
  6 CONTINUE

C COMPUTE TRIG FUNCTIONS

  SPHI = SIN(PHI)
  CPHI = COS(PHI)
  STTA = SIN(THETA)
  CTTA = COS(THETA)
  SPSI = SIN(PSI)
  CPSI = COS(PSI)
  TRIG1 = CTTA*SPHI
  TRIG2 = CTTA*CPHI
  TRIG3 = SPHI*STTA
  TRIG4 = CPHI*STTA
  TRIG5 = U*CTTA
  IF(ISW2 = 10)
10 ISW2 = 1

C SET COEFFICIENTS OF UD, VN, WD, PD, QD, RD

  FAU = AM-RHOL*XM
  FAQ = AM*ZG
  FAR = -AM*YG
  FLV = AM-RHOL*YV
  FLP = -RHOL4*YPD-AM*ZG
  FLR = -RHOL4*YPD-AM*YG
  FNW = AM-RHOL*W
  FNP = AM*YG
  FNQ = -RHOL4*QD-AM*YG
  RMV = -RHOL4*AKV-AM*ZG
  RMW = AM*YG

147
RMP = IX-RHOL5*AKPD
RMQ = IXY
RMR = IXZ-RHOL5*AKPD
PMU = AM*ZG
PMW = -RHOL4*AMWD-AM*ZG
PMP = IXY
PMQ = IY-RHOL5*AMQD
PMR = -IYZ
YMU = -AM*YG
YMW = -RHOL4*AHP*AM*ZG
YMP = -IXZ-RHOL5*ANPD
YMQ = -IY
YMR = -IZ-RHOL5*ANPD

20 CONTINUE

C COMPUTE UD FROM AXIAL FORCE EQUATION
C
40 CONTINUE
F(1) = (AM*(VP-WQ) + RHOL4*(XQQ*Q2+XRR*R2+XRP*RP) +
RHL3*(XVR*VP+XQO*Q) + RHL2*(XUU*U2+XVAV*V2+XWW*W2) +
RHOL2*(XDRDP*DR2+XDQDS*DS2+XDRDR*DR2) +
3*RHL2*(A1*U2+A2*UC+3*UC-UW)+XWR*STTA + RHL2*(XVVF*V2+XWFE*W2+
4*XRDR*DR2)+XODOE*O2(U2)*FTAM1)
5 + AM*(XG*(Q2+R2)-YQ*PQ-ZG*RP)
6 - FLP* F(5) - FAP* F(6)/FA11

C COMPUTE UD FROM LATERAL FORCE EQUATION
C
50 CONTINUE
F(2) = (AM*(VP-UR) + RHOL4*(YPP*A2SP+YQP*QP+YP*OR) +
RHL3*(YQO*QO+YWP*WP+XQ*QR)+RHL2*(YSTR*U2+YQAV*V2+YVAV*VRTVW +
YV*YD*YDR)+WAP*TRIG+RHL3*YQ*UR*FTAM1 +
4*RHL2*(YQ*QO+YVAV*VRTVW+YD*DR2)*FTAM1
5 + AM*(YR*Q2+P2)-YQ*PQ-ZG*RP
6 - FLP* F(4)-FLR*F(4)/FLV

C COMPUTE UD FROM NORMAL FORCE EQUATION
C
40 CONTINUE
F(3) = (AM*(UQ-VP) + RHOL4*(ZPP*P2+ZRR*R2+ZRP*RP) + RHL3*(7UR*VR+
ZV*VP+7Q*QO+ZAO*S2+7A*W)+RATWAV*ROTVW*ARSO1) +
2*RHL2*(ZSTP*U2+ZUW*ZAW+WRTV+ZAW*JAP+ZWW*APSRO+ROTVW +
ZV*VP+ZDS*SU2+ZDR*DR2) + WMR*TRIG + RHL3*ZQU*FTAM1
4*RHL2*(ZWE*UW+ZAWF*WRTV+ZDS*SU2)*FTAM1
5 + AM*(ZG*(P2+Q2)-XG*RP-YG*QR)
6 - FAP* F(4)-FNP*F(5)/FNP

C COMPUTE UD FROM ROLLING MOMENT EQUATION
C
70 CONTINUE
F(4) = (11Y-12)*(Q*IXZ*QO+YZ*O2-R2-IXY*PP+RHL5*AKQP*OR +
1AKQP*Q2+AK*AP+P*AP)+RHL6*(AKP+UP+AKR*UP+AKV*VQ+AKW*WP)
2 + AKWR * WR + RHOL3 * (AKSTR * U2 + AKV * UV + AKVAV * VRT VW + AKVW * VW + AKDR * DRU2) + RHOL3 * (AKSTR * U2 + ETAM1
4 + AM * (Y G * UO * VP) + ZG * (UR * WP))
5) - R MV * F (2) - RMW * F (3) - RMQ * F (5) - RMN * F (6) / RMP

C COMPUTE QN FROM PITCHING MOMENT EQN

C 90 CONTINUE
F (5) = ( (1 Z - IX) * RP + IXY * QR + (R2 - P2) * IXZ - IYZ * P Q + RHOL5 * (AMPP * P2 + AMRR * P 2 + AMRP * RP + AMQAQ * Q * ARS0) + RHOL4 * (AMVR * VR + ANVP * VP + AMQP * Q + AMAODS * 2UAAQS + AMANW * Q * RODTVW) + RHOL3 * (AMSTR + U2 + AWMU * UW + AMWAV * WRT VW + 3AM AW * WABSW + A M W W * A BS W * R OD T V W + AMVW * V2 + AMDS * U3U2 + AMDR * DBU2)
4 - (XG * CW - XB * CB) * TRIG2 - (ZG * CW - ZB * CB) * STTA + RHOL4 * AMQF * UQ * TAM1
5 + RHOL3 * (AMWE * UW + AMWAWE * WRT VW + AMDS * DSU2) * FTAM1
6 + AM * (7G * (VP - WQ) + XG * (VP - UQ))
7 - PMU * F (1) - PMW * F (3) - PMP * F (4) - PMN * F (6) / PMQ

C COMPUTE RD FROM YAWING MOMENT EQN

C 90 CONTINUE
F (6) = ( (1X - IY) * PQ + IYZ * RP + (P2 - Q2) * IXY - IXZ * QR + RHOL5 * (ANPQ * 1P Q + ANQRP * ANQRP + ANDR * ADRP + ANAVP * VP + ANVQ * Q + 2ANP * UP + ANUR * UR + ANADR * ADRP + ANAVP + R * RODTVW) + 3RHOL3 * (ANSTR * U2 + ANUV * UV + ANVAV * VRT VW + ANVW * VW + ANDR * DRU2)
4 + (XG * CW - XB * CB) * TRIG1 + (YG * CW - YB * CB) * STTA + RHOL4 * ANRE * UR * ETAM1 + 5RHOL3 * (ANWE * UW + ANWAWE * WRT VW + ANDRE * DBU2) * FTAM1
6 + AM * (7G * (VP - WQ) + XG * (VP - UQ))
7 - VMU * F (1) - VMV * F (2) - VMP * F (4) - VMN * F (6) / VMQ

C COMPUTE KINEMATICS - THETA DOT, PSI DOT, PHI DOT

C F(7) = Q * CPHI - R * SPHI
F(8) = (Q * SPHI + R * CPHI) / CTTA
F(9) = P * F(R) / STTA

C COMPUTE X DOT, Y DOT, Z DOT

C F(10) = TRIG5 * CPS1 * V * (TRIG3 * CPS1 - CPSI * SPQ1)
1 W * (TRIG4 * CPS1 + SPHI * SPQ1)
F(11) = TRIG5 * CPSI * V * (TRIG3 * CPSI + CPHI * CPS1)
1 W * (TRIG4 * CPSI - SPHI * CPS1)
F(12) = -U * STTA + V * TRIG1 + W * TRIG2
RETURN
END
SUBROUTINE PLTRNU(SAVE,K,ILOC,NLOC,IRUN)
DIMENSION SAVE(300,1),ILOC(1), IY(16), IR(2)
IR(1)=IHEX(13, 9, 14, 4, 13, 5, 4, 0)
IR(2)=IHEX(13, 5, 13, 6, 4, 11, 4, 0)
C T
IY(1) = IHEX(14, 3, 4, 0, 4, 0, 4, 0)
C U
IY(2)=IHEX(14, 4, 4, 0, 4, 0, 4, 0)
C V
IY(3)=IHEX(14, 5, 4, 0, 4, 0, 4, 0)
C W
IY(4)=IHEX(14, 6, 4, 0, 4, 0, 4, 0)
C P
IY(5)=IHEX(13, 7, 4, 0, 4, 0, 4, 0)
C Q
IY(6)=IHEX(13, 8, 4, 0, 4, 0, 4, 0)
C R
IY(7)=IHEX(13, 9, 4, 0, 4, 0, 4, 0)
C THETA
IY(8)=IHEX(14, 3, 12, 8, 14, 3, 12, 1)
C PSI
IY(9)=IHEX(13, 7, 14, 2, 12, 9, 4, 0)
C PHI
IY(10)=IHEX(13, 7, 12, 8, 12, 9, 4, 0)
C X
IY(11)=IHEX(14, 7, 4, 0, 4, 0, 4, 0)
C Y
IY(12)=IHEX(14, 8, 4, 0, 4, 0, 4, 0)
C Z
IY(13)=IHEX(14, 9, 4, 0, 4, 0, 4, 0)
C NR
IY(14)=IHEX(12, 4, 13, 9, 4, 0, 4, 0)
C NS
IY(15)=IHEX(12, 4, 14, 7, 4, 0, 4, 0)
C DB
IY(16)=IHEX(12, 4, 12, 2, 4, 0, 4, 0)
DIV = 70,
CALL SCALE(SAVF(1,1),6.0,K,1,DIV,1)
ICTL=0
CALL PLNT(0.0,75,73)
IF (K<>LT(I1),NLOC,J=ILOC(I1))
IF(J) 30,90,30
30 CONTINUE
CALL SCALE(SAVF(1,J),4.0,K,1,DIV,2)
CALL AXIS(0.0,0.0,IY(1),-4.6,0.0,0.0,DIV,1)
CALL AXIS(0.0,0.0,IY(J),14.4,0.090.0,DIV,7)
CALL SYMBOL(4.0,1.5,0.14,R,0.0,8)
AIRUN = IRUN
CALL NUMFR(-0.0,-0.0,-0.0,AIRUN,0.0,-1)
CALL LINE(SAVF(1,1),SAVF(1,J),K,1,0,0)
IF (ICTL(IT)) 50,60,40
50 CALL PLNT(0.0,4.50,-73)
ICTL=1
GO TO 90
40 CALL PLOT(8.5,-4.5,0,-73)
ICTL=0
90 CONTINUE
90 IF(ICTL)110,100,110
100 CALL PLOT(0.0,-.75,23)
RETURN
110 CALL PLOT(8.5,-5.25,23)
RETURN
END
SUBROUTINE CONTR(THFTA)
C
C TO CONTROL DS AND DR FOR DYNAMIC CONDITIONS
C
DIMENSION TL(121), ILOC(16), Y(13)
REAL IX, IY, IZ, IXY, IXZ, IYZ
REAL K
C
COMMON H, HMAX, HMIN, DH, FCT, TL, NGS, N, IS1, NPNT
C
COMMON TLIM, RHOL2, RHOL3, RHOL4, RHOL5, WMH, FTA, ETA01, ISW2
C
COMMON XQQ, XRR, XRP, XUP, XVR, XWQ, XJJU, XVV,
1 XWW, XR0R, XDSR, XR0DR, XVVE, XWWE, XDP0RF, XDS0SF
C
COMMON YRN, YRP, YPAP, YPQ, YQR, YVO, YVQ, YWP,
1 YWR, YR, YP, YARDP, YVAR, YVTR, YV, YVAV,
2 YVW, YDR, YRF, YVE, YVAVE, YDFR
C
COMMON Z0D, ZPP, ZRR, ZRP, ZWD, ZVP, ZVP, ZO,
1 ZAOQ, ZWAQ, ZSTR, ZW, ZWAE, ZAW, ZW, ZVV,
2 ZDS, ZDR, ZQE, ZWF, ZWAE, ZDSF
C
COMMON AKPD, AKRD, AKQR, AKPQ, AKRAP, AKP, AKR, AKVD,
1 AKVO, AKWP, AKSTR, AKV, AKVAV, AKVW, AKOR,
2 AKSTRE
C
COMMON AMOD, AMP, AMRP, AMQP, AQQAO, AMMD, AMVR, AMVP,
1 AMQ, AMQDS, AMWQC, AMSTR, AMW, AMWAW, AMAM, AMWW,
2 AMVW, AMDS, AMDR, AMQE, AMWF, AMWAE, AMDSF
C
COMMON ANRO, ANPD, ANPQ, ANQR, ANAR, ANVD, ANWP, ANWP,
1 ANVQ, ANP, ANR, ANADR, ANAVR, ANSTR, ANV, ANVAV,
2 ANVW, ANDR, ANRE, ANVE, ANVAVF, ANDRF
C
COMMON IX, IY, IZ, IXY, IXZ, IYZ
C
COMMON CW, CR, UC, XR, YP, ZR
C
COMMON DR, DS, DR, RHO, AL, AM
C
COMMON DPMAX, FTAHI, FTAJ0, A11, A12, A13
C
COMMON A21, A22, A23, A31, A32, A33
C
COMMON XG, YG, ZG
C
COMMON ILOC, IPLICIT, IRUN, TOPEN, NPLT, IOPT
C
COMMON Y, TIME, R1, DELTMA, SWMAX, R2, DELTMI
C
COMMON DSF, DSR, ICYC, NC
IF(NS)=15, 15, 16
15 RETURN

152
CONTINUE
GO TO (1001,1001,1003,1004,1005,1006,1007),NS
C
C CONTROL DS
C
1001 IF(ISW?)21,20,21
20 N1 = ?
 1 NN2 = 1
  NC2 = ((TIME*ICYC)/H) + .5
  NC3 = (ABS(DS -DELTMA))ICYC/ABS(R1*H) + .5
  NC5 = (ABS(DELTMI-DELTMA))ICYC/ABS(R2*H) + .5
  GO TO 11
21 GO TO (1,2,3,4,5,11),NI
C
C CYCLES TO START
C
2 NN2 = NN2 + 1
  IF (NN2 - NC2) 11,11,7
C
C DS DOWN
C
 7 N1 = 3
  NN3 = 0
  3 NN3 = NN3 + 1
  DS = DS + H*R1/ICYC
  IF (NN3 - NC3) 11,10,8
C
C DS LEVEL
C
 9 N1 = 4
  GO TO 11
  4 IF (ABS(THETA) - SWMAX) 11,9,9
C
C DS UP
C
 9 N1 = 5
  NN5 = 0
  5 NN5 = NN5 + 1
  DS = DS + H*R2/ICYC
  IF (NN5 - NC5) 11,10,10
C
C DS LEVEL
C
10 N1 = 6
C
11 IF (NS - 2) 13,1003,1003
13 CONTINUE
  GO TO 2000
C
C CONTROL DR + AUTOPILOT
C
1003 IF (ISW?) 301,3CO,301
300 ZC = Y(12)
500 DOT1 = 0.
DOE = ABS(DRMAX)
NC10 = ((TIME * ICYC) / H) + .5
NC6 = .85 * DOE * ICYC / (.08726 * H) + .5
NC7 = .08 * DOE * ICYC / (.1336 * H) + .5
NC8 = .04 * DOE * ICYC / (.006 * H) + .5
NC9 = .03 * DOE * ICYC / (.01064 * H) + .5
NN10 = 1
N2 = 2
IF(DRMAX)313, 314, 314
313 R6 = -.08726
R7 = -.01336
R8 = -.006
R9 = -.001064
GO TO 350
314 R6 = .08726
R7 = .01336
R8 = .006
R9 = .001064
GO TO 350
301 GO TO 302, 303, 304, 305, 306, 350, N2
302 NN10 = NN10 + 1
IF (NN10-NC10) 350, 350, 309
C
C FROM 0 TO .85 OF DRMAX
C
309 N2 = 3
NN6 = 0
303 NN6 = NN6 + 1
DR = DR + H*6/ICYC
IF (NN6-NC6) 350, 309, 309
C
C FROM .85 TO .93 OF DRMAX
C
309 N2 = 4
NN7 = 0
GO TO 350
304 NN7 = NN7 + 1
DR = DR + H*7/ICYC
IF (NN7-NC7) 350, 310, 310
C
C FROM .93 TO .97 OF DRMAX
C
310 N2 = 5
NN8 = 1
GO TO 350
315 NN8 = NN8 + 1
DR = DR + H*8/ICYC
IF (NN8-NC8) 350, 311, 311
C
C FROM .97 TO 1.0 OF DRMAX
C
311 N9 = 6
NN9 = 2
GO TO 350
306 NN9 = NN9 + 1
   DR = DR + H*NR/ICYC
   IF (NN9 - NC9) 350, 312, 317

C  LEVFL, DRM0X

C  312 N2 = 7
   350 IF(NS-2)2000,2000,352

C  C AUTOPILOT

C  352 DSC =.008*(ZC-Y(12)) + 3.5*Y(7) +.012*(Y(1)*SIN(Y(7))-Y(3)*COS(Y(7)))
      1+2.*Y(9)
   103 IF (DSC) 110,107,107
   107 IF (DSC -.436) 101,108,108
   110 IF (DSC +.436) 109,101,101
   108 DSC = .436
      GO TO 101
   109 DSC = -.436
   101 SDOT = 3 * (DSC -DS)
      DS = DS + .5 * H/ICYC * (3. * SDOT - SDOT1)
      SDOT1 = SDOT
      DB = -DS
   351 CONTINUE
      GO TO 2000

C  CONTROL DS (IMPULSE), LONGITUDINAL

C  1004 IF (ISW2)401,400,401
   400 IF (ICYC-1)411,411,412
   411 N4=0
      NTST=1
      NMOD=R
      GO TO 401
   412 N4=-1
      NTST=3
      NMOD=32
   401 IF(N4-NTST)403,402,403
   402 DS = DSF
   403 IF(MOD(N4,NMOD))410,406,410

C  PUNCH THETA AND TIME FOR FREQUENCY STUDY (LONGITUDINAL)

C  404 WRITE(?,408)Y(7),Y(13)
   408 FORMAT(2E15.7)
   410 N4 = N4 + 1
      GO TO 7000

C  CONTROL DR (IMPULSE), LATERAL

C  1005 IF (ISW2)501,500,501
   500 IF (ICYC-1)511,511,512
   511 N5 = 0
      NTST =1
NMOC=A
GO TO 501
512 NS=-1
NTST=3
NMOD=32
501 IF(N5-NTST)503,502,503
502 DR = DRF
503 IF(NMOD(N5,NMOD))510,506,510
C
C PUNCH PHI AND TIME FOR FREQUENCY STUDY(LATERAL)
C
506 WRITE(?,408) Y(9), Y(13)
510 NS = NS + 1
GO TO 350
C
C CONTROL ACCEL/DECEL + AUTOPILOT
C
1006 IF(ISW=1)601,60,601
600 NS=1
ISW6=0
NN11=1
TLIFM= 10.*TIME*60.
VC = Y(12)
SOFT1 = 0.
NC11=60*(1CYC/H)
NC12=TIME*1CYC/H
UC=0.
GO TO 352
601 GO TO(602,603,604,605,606,607,608),NS
C
602 NN11=NN11+1
IF(NN11-NC11)352,352,609
C
608 NS=2
UC=.445
NN12=0
603 NN12=NN12+1
IF(NN12-NC12)352,352,609
C
609 IF(ISW6=1)617,61A,617
617 NS=7
UC=0.
GO TO 352
C
614 NS=2
UC=16.,99
NN12=0
634 NN12=NN12+1
IF(NN12-NC12)352,352,617
C
610 IF(ISW6=1)61A,61S,61A
618 GO TO 408
C
615 N6=4
    UC=25.335
    NN12=0
605 NN12=NN12+1
    IF(NN12-NC12)352,352,611

C
611 IF(ISW6)619,614,619
619 GO TO 616
C
614 N6=5
    UC=33.78
    NN12=0
606 NN12=NN12+1
    IF(NN12-NC12)352,352,612

C
612 IF(ISW6)620,621,620
620 GO TO 615
C
621 N6=6
    UC=42.225
    NN12=0
607 NN12=NN12+1
    IF(NN12-NC12)352,352,613

C
613 ISW6 = 1
    GO TO 614
C
    CONTROL MAXIMUM ACCEL/DECCEL + AUTOPilot
C
1007 IF(ISW7)701,700,701
700 N7=1
    NN13=1
    TLIM=60.+2.0TIME
    NC13=60*ICYC/H
    NC14=TIME*ICYC/H
    SDOT1=0.
    ZC=Y(1?)
    UC=0.
    GO TO 352
701 GO TO(702,703,352),N7
C
C
702 NN13=NN13+1
    IF(NN13-NC13)352,352,705
C
705 N7=2
    UC=42.225
    NN14=0
703 NN14=NN14+1
    IF(NN14-NC14)352,352,706
C
706 N7=3
    UC=0.
    GO TO 352

157
2000 RETURN
END
/*
*/
**CHART TITLE - SUBROUTINE INVER(A,M,N,L,E,N,F,S)**

**NOTE 01**
- If E = 0, then go to STATEMENT 10.
- If E > 0, go to STATEMENT 7.
- If E < 0, go to STATEMENT 22.

**NOTE 02**
- If E = 0, then go toInSection 10.
- If E > 0, go toInSection 7.
- If E < 0, go toInSection 22.

- If E = 0, then go to SECTION 10.
- If E > 0, go to SECTION 7.
- If E < 0, go to SECTION 22.

- If E = 0, then go to SECTION 10.
- If E > 0, go to SECTION 7.
- If E < 0, go to SECTION 22.

**NOTE 03**
- If E = 0, then go to SECTION 10.
- If E > 0, go to SECTION 7.
- If E < 0, go to SECTION 22.

**NOTE 04**
- If E = 0, then go to SECTION 10.
- If E > 0, go to SECTION 7.
- If E < 0, go to SECTION 22.

**NOTE 05**
- If E = 0, then go to SECTION 10.
- If E > 0, go to SECTION 7.
- If E < 0, go to SECTION 22.

**NOTE 06**
- If E = 0, then go to SECTION 10.
- If E > 0, go to SECTION 7.
- If E < 0, go to SECTION 22.

**NOTE 07**
- If E = 0, then go to SECTION 10.
- If E > 0, go to SECTION 7.
- If E < 0, go to SECTION 22.

**NOTE 08**
- If E = 0, then go to SECTION 10.
- If E > 0, go to SECTION 7.
- If E < 0, go to SECTION 22.

**NOTE 09**
- If E = 0, then go to SECTION 10.
- If E > 0, go to SECTION 7.
- If E < 0, go to SECTION 22.

**NOTE 10**
- If E = 0, then go to SECTION 10.
- If E > 0, go to SECTION 7.
- If E < 0, go to SECTION 22.

**NOTE 11**
- If E = 0, then go to SECTION 10.
- If E > 0, go to SECTION 7.
- If E < 0, go to SECTION 22.

**NOTE 12**
- If E = 0, then go to SECTION 10.
- If E > 0, go to SECTION 7.
- If E < 0, go to SECTION 22.

**NOTE 13**
- If E = 0, then go to SECTION 10.
- If E > 0, go to SECTION 7.
- If E < 0, go to SECTION 22.

**NOTE 14**
- If E = 0, then go to SECTION 10.
- If E > 0, go to SECTION 7.
- If E < 0, go to SECTION 22.

**NOTE 15**
- If E = 0, then go to SECTION 10.
- If E > 0, go to SECTION 7.
- If E < 0, go to SECTION 22.

**NOTE 16**
- If E = 0, then go to SECTION 10.
- If E > 0, go to SECTION 7.
- If E < 0, go to SECTION 22.

**NOTE 17**
- If E = 0, then go to SECTION 10.
- If E > 0, go to SECTION 7.
- If E < 0, go to SECTION 22.

**NOTE 18**
- If E = 0, then go to SECTION 10.
- If E > 0, go to SECTION 7.
- If E < 0, go to SECTION 22.

**NOTE 19**
- If E = 0, then go to SECTION 10.
- If E > 0, go to SECTION 7.
- If E < 0, go to SECTION 22.

**NOTE 20**
- If E = 0, then go to SECTION 10.
- If E > 0, go to SECTION 7.
- If E < 0, go to SECTION 22.

**NOTE 21**
- If E = 0, then go to SECTION 10.
- If E > 0, go to SECTION 7.
- If E < 0, go to SECTION 22.

**NOTE 22**
- If E = 0, then go to SECTION 10.
- If E > 0, go to SECTION 7.
- If E < 0, go to SECTION 22.
/ END /

26.09-39

FULL PRESENT VALUE
OF VARIABLES OUT OF
ARRAY Y1

01 U = Y1(1)
  V = Y1(2)
  W = Y1(3)
  P = Y1(4)
  Q = Y1(5)
  R = Y1(6)

02 T = Y1(7)
  PHI = Y1(8)
  PHI = Y1(9)
  Z = Y1(10)
  Y = Y1(11)
  I = Y1(12)

03 I = 1
  N = 1
  M = 1
  I = (TETSA)
  IO = 0
  II = 0

COMPUTE QUANTITIES TO
BE USED FOR THE

04 V = Y1
  Q = QI
  S = S
  S = S
  U1 = U
  U = N

05 g = W
  DO2 = DO2
  DDO2 = M2
  D2 = D2
  D3P = D3P
  D3P = D3P

/ END /

176
<table>
<thead>
<tr>
<th>No.</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>CONTINUE</td>
</tr>
<tr>
<td>02</td>
<td>CONTINUE</td>
</tr>
<tr>
<td>03</td>
<td>CONTINUE</td>
</tr>
<tr>
<td>04</td>
<td>CONTINUE</td>
</tr>
<tr>
<td>05</td>
<td>CONTINUE</td>
</tr>
<tr>
<td>06</td>
<td>CONTINUE</td>
</tr>
<tr>
<td>07</td>
<td>CONTINUE</td>
</tr>
<tr>
<td>08</td>
<td>CONTINUE</td>
</tr>
<tr>
<td>09</td>
<td>CONTINUE</td>
</tr>
<tr>
<td>10</td>
<td>CONTINUE</td>
</tr>
</tbody>
</table>

**Chart Title:** Subroutine FVLI1111

**Flowchart:**
- Compute V0 from axial force
- Compute V0 from bending moment
- Compute V0 from lateral force
- Compute V0 from axial force

**Notes:**
- Note 01
- Note 02
- Note 03
- Note 04
- Note 05
- Note 06
- Note 07
- Note 08
- Note 09
- Note 10

**Diagram:**
- Flowchart indicating the sequence of operations and data flow.
J0R  ZC790
EXEC F80PT8RN
DIMENSION Y(6)
DIMENSION SAVE(300, 8), ILOC( 8), BUFF(3000)
REAL IY
COMMON H, N, I51, NPNT
COMMON TL14, RHOL2, RHOL3, RHOL4, RHOL5, W4R, ETA, FTAN1, ISW2
COMMON XQQ, XUQ, XWQ, XUU, XWW, XDSDS, XNBAR, XWWE,
   1
   XDSDF
COMMON ZWD, ZWD, ZQ, ZAQDS, ZWAQ, ZSTR, ZW, ZWAW,
   1
   ZAV, ZW, ZDS, ZDR, ZOF, ZWF, ZWAWF, ZDSF
COMMON AMQO, AMQAO, AMWD, AMQ, AMAOQDS, AMAWQ, AMSTR, AMW,
   1
   AMWAW, AMAW, AMWAW, AMDS, AMDR, AMOF, AMWF, AMWAWE,
   2
   AMDSF
COMMON IY
COMMON CW, CB, UC, XR, 7B
COMMON DS, DB, RHO, AL, AM
COMMON FTAHI, FTALO, A11, A12, A13
COMMON A21, A22, A23, A31, A32, A33
COMMON XG, 7G
COMMON ILOC, IPLIT, IPUN, IOPEN, NPLT, I0PT
COMMON Y
COMMON TIME, R1, DELTMA, SWMAX, R2, DELTMI, DSF, ICYC, NS
ICYC = 1
PI = 3.141593
ICPT = 0
ICPFN = 0
N = 5
CONT INUE
CALL INPUT
46 CONT INUE
48 IF(IOPEN) 50, 69
49 CALL PLOTS(BUFF, 12000, 7)
44 CONT INUE
NLOC = 14
40 CONT INUE
K = 0
IOUT = 3
LINSPD = 90
102
COMPUTE RHO * L CONSTANTS

RHOH = RHO * .5
RHOH2 = RHOH * AL * AL
RHOH3 = RHOH2 * AL
RHOH4 = RHOH3 * AL
RHOH5 = RHOH4 * AL

WRITE OUT HYDRODYNAMIC COEFFICIENTS

WRITE (IOUT, 1) XQC, ZOD, AMOD, XUC, ZWD, AMQAQ, XWQ, ZO, AMWN, XJJ, ZAONS, AMQ,
1XWW, ZWQAQ, AMQAQDS, XDSDS, ZSTR, AMWAQ, XXDDBB, ZW, AMSTR, XWBF, ZWAW, AMW,
2XDSDSE, ZAW, AMWAN, ZWN, AMWN, ZDS, AMWN, ZBO, AMDS
1 FORMAT (1H , 'XQC', T9, E12.5, T44, 'ZOD', T51, E12.5, T86, 'MOD', T93, E12.5,
1T1H , 'XUC', T9, E12.5, T44, 'WQ', T51, E12.5, T86, 'WAQ', T93, E12.5/,
1T1H , 'XDDBB', T9, E12.5, T44, 'BB', T51, E12.5, T86, 'MSTR', T93, E12.5/,
1T1H , 'XWBF', T9, E12.5, T44, 'WBF', T51, E12.5, T86, 'MAW', T93, E12.5/,
1T1H , 'XWAN', T9, E12.5, T44, 'WAN', T51, E12.5, T86, 'MAWF', T93, E12.5/,
1T1H , 'XWAN', T9, E12.5, T44, 'WAN', T51, E12.5, T86, 'MAWF', T93, E12.5/,
1T1H , 'XWAN', T9, E12.5, T44, 'WAN', T51, E12.5, T86, 'MAWF', T93, E12.5/,
1T1H , 'XWAN', T9, E12.5, T44, 'WAN', T51, E12.5, T86, 'MAWF', T93, E12.5/,
1T1H , 'XWAN', T9, E12.5, T44, 'WAN', T51, E12.5, T86, 'MAWF', T93, E12.5/,
1T1H , 'XWAN', T9, E12.5, T44, 'WAN', T51, E12.5, T86, 'MAWF', T93, E12.5/,
1T1H , 'XWAN', T9, E12.5, T44, 'WAN', T51, E12.5, T86, 'MAWF', T93, E12.5/,
2T65, 'DA', T73, F12.5, T30, F12.5, T44, 'UC', T51, F12.5,
2T65, 'RA', T73, F12.5, T30, F12.5, T44, 'DA', T51, F12.5,
2T65, 'RA', T73, F12.5, T30, F12.5, T44, 'DA', T51, F12.5,
2T65, 'RA', T73, F12.5, T30, F12.5, T44, 'DA', T51, F12.5,
2T65, 'RA', T73, F12.5, T30, F12.5, T44, 'DA', T51, F12.5,
2T65, 'RA', T73, F12.5, T30, F12.5, T44, 'DA', T51, F12.5,
2T65, 'RA', T73, F12.5, T30, F12.5, T44, 'DA', T51, F12.5,
2T65, 'RA', T73, F12.5, T30, F12.5, T44, 'DA', T51, F12.5,
2T65, 'RA', T73, F12.5, T30, F12.5, T44, 'DA', T51, F12.5,
2T65, 'RA', T73, F12.5, T30, F12.5, T44, 'DA', T51, F12.5,
2T65, 'RA', T73, F12.5, T30, F12.5, T44, 'DA', T51, F12.5,
2T65, 'RA', T73, F12.5, T30, F12.5, T44, 'DA', T51, F12.5,
```fortran
2.5/1H, 'NSF', T9, F12.5, T44, 'ICYC', T51, 12, T65, 'NS', T72, 12/
WRITE(OUT, 5) IRUN
5 FORMAT(1H, 'RUN NO ', 15/)
C
C COMPUTE W-B
C
WMB = CW - CB
IS1 = 0
ISW2 = 0
CALL INTFG(Y)
C
C SAVE VALUES FOR PLOTS IF I PLOT = 1
C
IF (I PLOT) 19, 12, 9
9 CONTINUE
C
C IF ARRAY FULL DCNT OVERUN
C
IF (K - 300) 63, 12, 12
63 CONTINUE
IF (NPLT - ICNT?) 52, 52, 51
52 ICNT? = 0
K = K + 1
C
SAVE TIME
SAVE(K, 1) = Y(4)
C
SAVE U
SAVE(K, 2) = Y(1)
C
SAVE W
SAVE(K, 3) = Y(2)
C
SAVE Q
SAVE(K, 4) = Y(3)
C
SAVE THETA
SAVE(K, 5) = Y(4)
C
SAVE Z
SAVE(K, 6) = Y(5)
C
SAVE DS
SAVE(K, 7) = DS
C
SAVE DR
SAVE(K, 8) = DR
51 ICNT? = ICNT? + 1
12 CONTINUE
IF (INPNT - ICNT?) 16, 14, 17
16 ICNT? = 0
IF (LINSPP = INS) 20, 20, 30
30 LINS = 0
IPAGE = IPAGE + 1
WRITE(OUT, 24) IPAGE
24 FORMAT(1H, 7X, 'SHARKINE SIMULATION, LONGITUDINAL PPLE
DOM', 45X, 'PAGE', 18/)
WRITE(OUT, 25)
9 CONTINUE
72 WRITE(OUT, 10) (Y(I), I = 1, 6)
10 FORMAT(1HO, 6(2X, F13.6, 2X))
```
74 CONTINUE
   LINS = LINS + 2
17 ICNT = ICNT + 1
   IF(ABS(Y(A) - TLIM) - H) 35, 69, 69
35 CONTINUE
   IF(IPMT) 40, 46, 40
40 CALL PLTROU(SAVE, K, ILOC, NLOC, IRUN)
   GO TO 46
END
SUBROUTINE INPUT
DIMENSION Y(6), ILOC(8), YHOLD(6), COM(219)
EQUIVALENC (COM(1),H)
REAL IV
C
COMMON H, N, IS1, NPNT
C
COMMON TLIM, RHOL2, RHOL3, RHOL4, RHOL5, WMR, ETA, ETA4L, IS2
C
COMMON XQQ, XUD, XWO, XVU, XWW, XDSO5, XDROR, XWWE, 1
XDSDF
C
COMMON ZOD, ZWD, Z0, ZAQDS, ZWAQ, ZSTR, ZW, ZWAW, 1
ZAW, ZWW, ZDS, ZOK, ZOE, ZWF, ZWAWF, ZDSF
C
COMMON AMOD, AMODAQ, AMWD, AMQ, AMAQDS, AMAWD, AMSTR, AMW, 1
AMWAN, AMAW, AMWW, AMDS, AMDR, AMOE, AMWF, AMWAWF, AMDSF
COMMON IV
C
COMMON CW, CA, UC, XR, ZP
C
COMMON DS, DA, RHQ, AL, AM
C
COMMON ETAHI, ETALSO, A11, A12, A13
C
COMMON A21, A22, A23, A31, A32, A33
C
COMMON XG, ZG
C
COMMON ILOC, IPLOT, IRUN, IOPT, NPLT, I OPT
C
COMMON Y
C
COMMON TIME, R1, RELTMA, SWMAX, R2, DELTIM, OSE, ICYC, NS

IN = 1
IF(I OPT)50,5,150
5 CONTINUE
READ(IN,50) NPNT, IPLOT, IRUN, NPLT, IOPT, NS
IF(IRUN)70,60,70
60 IF(IOPEN)62,64,62
62 CALL PLOT(5.0,0.0,1)
64 CONTINUE
70 CONTINUE
READ(IN,50) (ILOC(I), I = 1,8)
50 FORMAT(16F5)
READ(IY,100) HO, TLIM
H = HO
100 FORMAT(9F12.5)
READ(IY,100) (Y(I), I=1,5)
Y(6) = TO
SAVE INITIAL CONDITIONS FOR POSSIBLE RESTORE

DO 110 I=1,6
  YHOLD(I) = Y(I)
110 CONTINUE

SAVE DS, DA FOR POSSIBLE RESTORE

DSHOLD = DS
DASHOLD = DA
RETURN

RESTORE INITIAL VALUES

DO 152 I=1,6
  Y(I) = YHOLD(I)
152 CONTINUE

RESTORE INITIAL DS, DA, H

DS = DSHOLD
DA = DASHOLD
H = HO
READ(IN,145) IRUN
IF(IRUN) 155,05,155
155 CONTINUE
160 READ(IN,145) NDIX, VALUE
165 FORMAT(15,5X,9F5.5)
IF (INDEX) 190, 170, 180
170 RETURN
190 COM (INDEX) = VALUE
GO TO 160
END
SUBROUTINE INTFG(Y)
DIMENSION Y(1), F(5), F1(5)
COMMON H, N, ISI
IF(ISI)30,10,30
10 M = N + 1
   DO 20 I = 1,N
   F(I) = 0.
20 F1(I) = 0.
   ISI = 1
   RETURN
30 CALL EVAL1(Y,F)
   DO 40 I = 1,N
C       Y(I) = Y(I) + .5*H*(3.*F(I)-F1(I))
C       F1(I) = F(I)
40 CONTINUE
   Y(M) = Y(M) + H
   RETURN
END
SUBROUTINE EVAL1(YI,F)
DIMENSION YI(1), F(1)
REAL Y
C
COMMON H, N, IS1, NPNT
C
COMMON TLIM, RHOL?, RHOL?, RHOL4, RHOL5, WMR, FTA, FTAM1, ISW2
C
COMMON XQQ, XUD, XWQ, XUU, XWW, XDSNS, XDSDR, XWFF, 1
C
COMMON ZQQ, ZWD, ZB, ZAQS, ZWAQ, ZSTR, ZW, ZWAW, 1
C
COMMON AMCO, AMOAQ, AMWQ, AMQS, AMAOS, AMAW, AMST, AMW, 1
C
COMMON AMWAW, AMAH, AMWW, AMDS, AMDR, AMQF, AMWE, AMWAE, 1
C
COMMON IY
C
COMMON CW, CB, UC, X8, ZP
C
COMMON DS, DB, RHQ, AB, AM
C
COMMON ETAH1, ETAH0, A11, A12, A13
C
COMMON A21, A22, A23, A31, A32, A33
C
COMMON XG, ZG
C
CALL CONTR(THETA)
C
COMPUTE QUANTITIES TO BE USED MORE THAN ONCE
C
Q2 = Q*Q
U2 = U*U
W2 = W*W
DS2 = DS*DS
DS2UP = DS2*U2
WQ = W*Q
ROOTW=SORT(W2)
UQ=U*Q
ARSO=ARSI(Q)
UAQOS=U*ARSO*DS
WRTW=ROOTWTW
UW=U*W
ABS\textsc{w}=\textsc{ars}(\textsc{w})
UABS\textsc{w}=U*\textsc{abs}\textsc{w}
DBUz=DR*U2
OSU2=DS*U2
UMAG = \text{SQRT}(U2+W2)
\text{IF}(\text{UMAG})26,24,26

24 \text{ FTA} = 20.
\text{GO TO} 28

26 \text{ CONTINUE}

\text{ETA} = UC/\text{UMAG}

28 \text{ ETAH}1 = \text{ ETA} - 1.
\text{IF}(\text{ETA} - \text{ ETAH}1)32,30,30

30 A1=A1
A2=A12
A3=A13
\text{GO TO} 38

32 \text{IF}(\text{ ETA} - \text{ FTAL})35,35,37

35 A1=A31
A2=A32
A3=A33
\text{GO TO} 38

37 A1=A21
A2=A22
A3=A23

38 \text{ CONTINUE}
1 \text{ RATVAV} = 0.
3 \text{IF}(W)5,4,5
4 \text{ RATWAW} = 0.
\text{GO TO} 6
5 \text{ RATWAW} = W/\text{ ARSW}
6 \text{ CONTINUE}

C
C \text{COMPUTE TRIG FUNCTIONS}
C

SPHI=0.
C PHI=1.
STTA = \text{SIN}(\text{ THETA})
C TTA = \text{COS}(\text{ THETA})
SPSI=0.
CPSI=1.
TRIG1=0.
TRIG2=TTA
TRIG3=3.
TRIG4=STTA
TRIG5 = TTA
\text{IF}(\text{ISW}2)20,10,20

10 ISW2 = 1

C
C \text{SET COEFFICIENTS OF UD, WD, ON}
C
FAU = \text{AM} - \text{PHI} * \text{Y} * \text{ID}
FAQ = \text{AM} * \text{PI}
FWN = \text{AM} - \text{PHI} * \text{I} * \text{WD}
FNQ = -\text{PHI} * \text{I} * \text{CN} - \text{AM} * \text{X};
PMU = AM*7G
PMW = -RHOLO4*AMWD-AM*XG
PMQ = IY-RHOL5*AMQD

20 CONTINUE

COMPUTE UD FROM AXIAL FORCE EQN

40 CONTINUE

F(1) = ((AM*(-WQ) + RHOLO4*(XQQ*Q2) +
1PHOL4*(XQO*Q0) + RHOLO4*(XUO*U2 + XWW*W*W)) +
2RHOL2*U2*(1+XDSDS*DS?+XBDBR*DB*DB1) +
3RHOL2*(A1*U2+A2*UC+A3*UC+UC)-WMR*STTA+ RHOLO2*(XWWE*W2+ XWDSSE*DSSU2)*FTAM1) + AM*XG*Q2
5-FAQ* F(1) / FAU

COMPUTE WD FROM NORMAL FORCE EQN

60 CONTINUE

F(2) = (AM*UQ + RHOLO4((
1ZQ*UQ + ZAQS* UAQS + /WAI, - RATWAW*ROOTV*ABSQ) +
2RHOL2 * ZSTR*U2 + ZE*UW+ZWW * WRTVW + ZAW+UARSW+ZWW*ARSW*ROOTVW +
3ZCS*DSU2+ZDS*DBU2) + WMR = TRIG2 + RHOLO3* ZQF*UQ* FTA M1
4*RHOL2*(ZWF*UW+ZAFW*WRTVW+ZDSE*DSU2)*FTAM1
5 +AM*ZQ*Q2-FNQ*F(1)/FNW

COMPUTE QD FROM PITCHING MOMENT EQN

90 CONTINUE

F(3) = (RHOLO5*(AMCAQ*Q*ABSQ) + RHOLO4*(AMQ*U2) +AMAADS* UAOOS +AMAW*Q+ROOTVW)+ RHOLO4*(AMSTR*U2+AMW*UW+AMWW*WRTVW +
2AMAW*UARSW +AMWW*ARSW*ROOTVW+AMDS*DSU2+AMDR*DBU2)
3-(XG*CW-XB*CR)*TRIG2-(ZG*CW-ZB*CR)*STTA+RHOLO4*AMQE*UO*FTAM1
4*RHOL2*(AMWF*UW+AMWW*WRTVW+AMDS*DSU2)*FTAM1
5 +AM*(ZG*(-WC)+XG*(-UQ))
6 +PMI*F(1)-PMW*F(2)/PMQ

COMPUTE KINEMATICS - THETA DOT

F(4) = 0

COMPUTE Y DOT

F(5) = -U*STTA+W*TRIG2
RETURN
END
SUBROUTINE PLTRU(SAVE,K,ILOC,NLOC,IRUN)

DIMENSION SAVE(300,1), ILOC(1), IY(16), IR(2)

IR(1)=IHEX(13,9,14,4,13,5,4,0)
IR(2)=IHEX(13,5,13,6,4,11,4,0)

C T
IY(1) = IHEX(14,3,4,0,4,0,4,0)
C U
IY(2) = IHEX(14,4,4,0,4,0,4,0)
C W
IY(3) = IHEX(14,6,4,0,4,0,4,0)
C Q
IY(4) = IHEX(13,8,4,0,4,0,4,0)
C THTA
IY(5) = IHEX(14,3,12,8,14,3,12,1)
C Z
IY(6) = IHEX(14,4,4,0,4,0,4,0)
C DS
IY(7) = IHEX(17,4,14,2,4,0,4,0)
C DR
IY(8) = IHEX(12,4,12,2,4,0,4,0)
DIV = 20.
CALL SCALE(SAVE(1,1),-.0,K,1,DIV,1)
ICTL=0
CALL PLCT(0.0,-75,73)
DN 80 I=1,NLOC
J=ILOC(I)
IF(J) 30,90,30
30 CONTINUE
CALL SCALE(SAVE(1,J),-.0,K,1,DIV,2)
CALL AXIS(0.0,0.0,IY(1),-4.6,0.0,0.0,DIV,1)
CALL AXIS(0.0,0.0,IY(J),4.4,0.0,90.0,0.0,DIV,2)
CALL SYMROL(4.0,3.5,0.14,IR,0.0,9)
AIRUN = IRUN
CALL NUMBER(-0.0,-0.0,-0.0,AIRUN,0.0,-1)
CALL LINE(SAVE(1,1),SAVE(1,J),K,1,0,1)
IF(ICTL)60,50,60
50 CALL PLCT(0.0,4.50,73)
ICTL=1
GO TO 90
60 CALL PLCT(4.5,,-4.3,,-23)
ICTL=0
90 CONTINUE
100 CALL PLCT(0.0,-.75,73)
RETURN
110 CALL PLCT(4.5,,-5.25,23)
RETURN
END
SUBROUTINE CONTR(THETA)
C TO CONTROL DS FOR DYNAMIC CONDITIONS
C
DIMENSION ILCC(8), Y(6)
REAL IY
REAL K
C
COMMON H, N, IS1, NPNT
C
COMMON TLIM, RHOL2, RHOL3, RHOL4, RHOL5, WMB, ETA, FTAMI, ISW2
C
COMMON XQC, XUD, XWQ, XUU, XWW, XDSNS, XDBDB, XWWE, 1
XDSDFS
C
COMMON ZQD, ZWD, ZJ, ZAPPX, ZWAQ, ZSTR, ZW, ZWAW, 1
ZAK, ZWW, ZOS, ZDR, ZOE, ZWF, ZMAWF, ZOSF
C
COMMON AMQ, AMQAO, AMWP, AMQ, AMAOQ, AMWAQ, AMSTR, AMW, 1
AMWAW, AMMAW, AMWNN, AMDD, AMOE, AMWF, AMWAWE, AMQF
C
COMMON IY
C
COMMON CW, CR, UC, X8, ZP
C
COMMON DS, DB, RH, AL, AM
C
COMMON ETAHI, FTAI1, A11, A12, A13
C
COMMON A21, A22, A23, A31, A32, A33
C
COMMON XG, ZG
C
COMMON ILOC, IPILOT, IIPUN, IOPEN, NPLT, INPT
C
COMMON Y, TiWF, Pi, DELTMA, SWMAX, P2, DELTMI
C
COMMON DSE, ICYC, NS
IF(NS)15,15,16
15 RETURN
1A CONTINUE
GO TO(1001,1002,1003,1004),NS
C
CONTROL DS
C
1001 IF(ISW2=1)21,20,21
20 N1 = 2
1 NN2 = 1
NC2 = ((TWF*ICYC)/H) + .5
NC3 = (ABS(PS -DELTMA)*ICYC/ABS(R1*H) + .5
NC5 = (ABS(DELTM1-DELTMA)*ICYC/ABS(R2*H) + .5
GO TO 11
21 GO TO (1,2,3,4,5,11),N1
C
C CYCLES TO START
C
7 NN2 = NN2 + 1
   IF (NN2 - NC2) 11,11,7
C
C DS DOWN
C
7 N1 = 7
   NN3 = 0
3 NN3 = NN3 + 1
   DS = DS + H*R1/ICYC
   IF(NN3 - NC3) 11,0A,8
C
C DS LEVEL
C
8 N1 = 4
   GO TO 11
4 IF (ABS(THETA) - SWMAX) 11,9,9
C
C DS UP
C
9 N1 = 5
   NN5 = C
5 NN5 = NN5 + 1
   DS = DS + H*R2/ICYC
   IF(NN5 - NC5) 11,10,10
C
C DS LEVEL
C
10 N1 = 6
11 CONTINUE
   GO TO 2000
C
C AUTCPILCT
C
352 DSC =.009* (7C-V(5)) +3.54*Y(4)+.012*Y(11)*SIN(Y(4)) -Y(2)*COS(Y(4))
   1+2*Y(3)
103 IF (DSC) 110,107,107
107 IF (DSC - .436) 101,104,108
110 IF (DSC + .436) 109,101,101
104 DSC = -.436
    GO TO 101
109 DSC = -.436
101 Soot = 3 * (DSC -NS)
    DS = DS +.5 * H*ICYC * (1. * Soot - Soot)
    Soot = Soot
    RB = -DS
351 CONTINUE
   GO TO 2000
C
C CONTROL DS (IMPULSF), LONGITUDINAL
C
1002 IF (ISW2)401,400,401
400 N4 = 0
401 IF (N4-1)403,402,403
402 DS = NSE
403 IF (MOD(N4, 8))410,406,410
C
C PUNCH THETA AND TIME FOR FREQUENCY STUDY (LONGITUDINAL)
C
406 WRITE(2,408)Y(4),Y(6)
408 FORMAT(2E15.7)
410 N4 = N4 + 1
GO TO 7000
C
C CONTROL ACCEL/DECEL + AUTOPILOT
C
1003 IF (ISW2)601,600,601
600 N6=1
ISW6=0
NN11=1
TLIM= 10.*TIME+60.
ZC = Y(5)
SD01 = 0.
NC11=60*(ICYC/H)
NC12=TIME*ICYC/H
UC=0.
GO TO 252
601 GO TO(402,603,604,605,606,607,352?),N4
C
UC=0.
602 NN11=NN11+1
IF (NN11-NC11)352,352,608
C
608 N6=2
UC=8.445
NN12=0
603 NN12=NN12+1
IF (NN12-NC12)352,352,600
C
609 IF (ISWA)617,616,617
617 N6=7
UC=0.
GO TO 352
C
616 N6=3
UC=16.99
NN12=0
604 NN12=NN12+1
IF (NN12-NC12)352,352,617
C
610 IF (ISWA)618,615,618
614 GO TO 408
C
615 N6=4
UC=25.338
NAVTRADEVCEN 68-C-0050-2

NN12=0
605 NN12=NN12+1
   IF(NN12-NC12)352,352,611
C
611 IF(ISW6)619,614,610
619 GO TO 616
C
614 N6=5
   UC=33.78
   NN12=0
606 NN12=NN12+1
   IF(NN12-NC12)352,352,612
C
612 IF(ISW6)620,621,620
620 GO TO 615
C
621 N6=6
   UC=42.225
   NN12=0
607 NN12=NN12+1
   IF(NN12-NC12)352,352,613
C
613 ISW6 = 1
   GO TO 614
C
   CONTROL MAXIMUM ACCEL/DECEL + AUTOPILOT
C
1004 IF(ISW2)701,700,701
700 N7=1
   NN13=1
   TLIM=60.+2.*TIME
   NC13=40.*ICYC/H
   NC14=TIME*ICYC/H
   S001=0,
   ZC=Y(5)
   UC=0.
   GO TO 352
701 GO TO(702,703,352,N7
C
C
702 NN13=NN13+1
   IF(NN13-NC13)352,352,705
C
705 N7=2
   UC=42.225
   NN14=0
703 NN14=NN14+1
   IF(NN14-NC14)352,352,706
C
706 N7=3
   UC=0.
   GO TO 352
7000 RT1UPN
FND
// JOB EC470
// EXEC FFORTRAN
C IN THE MANNER OF NEWTON
C
CONV = 57.29578
IN=1
IOUT=3
LNSPP=54
3 CONTINUE
READ(IN,100) ZWAW, ZWW, ZW, ZAW, ZSTR, ZDEL
IF(ZWAW=999.15,4,4
4 CALL EXIT
5 CONTINUE
READ(IN,100) AMWAW, AMWW, AMW, AMAW, AMSTR, AMDEL
READ(IN,100) B, ZR, RHO, AL
2 CONTINUE
READ(IN,100) U, WZERO, UMAX
IF(WZERO)9,3,9
9 CONTINUE
WRITE(IOUT,60)
WRITE(IOUT,252) ZWAW, ZWW, ZW, ZAW, ZSTR, ZDEL
WRITE(IOUT,61)
WRITE(IOUT,252) AMWAW, AMWW, AMW, AMAW, AMSTR, AMDEL
WRITE(IOUT,62)
WRITE(IOUT,252) B, ZR, RHO, AL
60 FORMAT(IH,50X,'EC470 - INITIAL CONDITIONS'/
I3H,3X,'ZWAW',14X,'ZWW',11X,'ZW',14X,'ZAW',12X,'ZSTR',13X,
2'ZDEL')
61 FORMAT(IH,3X,'AMWAW',14X,'AMWW',11X,'AMW',14X,'AMAW',12X,'AMSTR',13X,
?AMDEL')
62 FORMAT(IH,3X,'B',17X,'ZR',12X,'RHO',13X,'L')
WRITE(IOUT,215)
LINS = 9
100 FORMAT(9F10.5)
8 CONTINUE
ICNT = 0
WWK=WZERO
U2=U=0
R = ZDEL/AMDEL
C1= ZWAW-R*AMWAW
C2= ZWW-R*AMWW
C3= (ZW -R*AMW)*U
C4= (ZAW-R*AMAW)*U
C5= (ZSTR-R*AMSTR)*U
COEF=2.*B*ZR/(RHO*AL*AL*AL)
C6 = -R*COEF
10 CONTINUE
ICNT = ICNT + 1
IF(ICNT < 200) 15,15,12
12 WRITE(IOUT,13)
13 FORMAT(IH, 'ITERATIONS EXCEEDED 200')
GO TO 70
15 CONTINUE
ABS = ABS(WWK)
W2 = WWK * WWK
WABS = WWK * ABS
ABS WU = ABS * U
WU = WWK * U
T1 = W2 + U?
ROOT = SQRT(T1)
T2 = COEF * WWK / ROOT
F = C1 * WABS + C2 * W2 + C3 * WWK + C4 * ABS + C5 + C6 * WWK / ROOT
IF (WWK) = 20, 18, 30
19 WRITE (OUT, 19)
19 FORMAT (1H, 'W IS ZERO'
20 S = 1.
GO TO 40
30 S = 1. 
40 CONTINUE
FP = 2.5 * C1 * WWK + 2 * C2 * WWK + C3 + S * C4 + C6 * U2 / (ROOT * T1)
WS = WWK - F / FP
IF (ABS(WWK - WS) - .0001 * ABS(WWK)) 200, 200, 150
150 WWK = WS
GO TO 10
200 CONTINUE
DS = (AMW + WARB + AMX + W2 + AMW + WU + AMW + ABSWU +
1 AMSTR + U2 + T2) / (AMDEL * U2)
IF (LINSP - LINS) 210, 210, 250
210 WRITE (OUT, 211)
211 FORMAT (1H1)
WRITE (OUT, 215)
215 FORMAT (1H1, 'X', 'U/SEC', 'X', 'U(KTS)', 'X', 'W/SEC', 'X', 'THETA(RAD)', 'X', 'DEL(SEC)', 'X', 'THETA(SEC)'
LINS = 0
250 LINS = LINS + 1
UKTS = U / 1.689
THTA = ATAN(WS / U)
DS = DS * CONV
THTAD = THTA * CONV
WRITE (OUT, 252) U, UKTS, WS, DS, THTA, DS1, THTAD
252 FORMAT (1H1, F13.6, 5F16.4)
WZFO = WS
U = U - 1.689
IF (U - U1) = 2, 9, 6
760 GO TO 2
END
/*
*/
11H = 4E16.61
LINS = LINS + 6
GO TO 105
END

/*
*/

/*
C SUBMARINE THRUST
DIMENSION THRUST(10)
READ(10,30)AL,AM,ETAH,FTALO
READ(10,40)A1,A2,A3
READ(10,40)A1,B2,R3
READ(10,40)C1,C2,C3
30 FORMAT(8F10.3)
40 FORMAT(8F10.6)
WRITE(3,36)
36 FORMAT(1H1,5X,23H/C300/, SUBMARINE THRUST/T6, ' ', T16, ' ', M ', T24, ' ', ETAH
   21', T34, ' ', FTALO')
WRITE(3,41)AL,AM,ETAH,FTALO
41 FORMAT(9F9.2, F11,2,2F10.3)
WRITE(3,42)
42 FORMAT(1H0, T10, ' ', A1', T20, ' ', A2', T30, ' ', A3')
WRITE(3,43)A1,A2,A3
43 FORMAT(5X,3F10.6)
WRITE(3,44)
44 FORMAT(1H0, T10, ' ', B1', T20, ' ', B2', T30, ' ', B3')
WRITE(3,45)
45 FORMAT(1H0, T10, ' ', C1', T20, ' ', C2', T30, ' ', C3')
WRITE(3,46)C1,C2,C3
J=1
DO 25 K=1,7
WRITE(3,76)
26 FORMAT(1H1)
IF(J32,T33,33)
32 WRITE(3,34)
33 WRITE(3,31)
34 FORMAT(1H0, X, 40HACCELERATION - (FEET PER SECOND-SQUARED) /
25 GO TO 35
35 WRITE(3,31)
36 FORMAT(50X,16HFORCE - (POUNDS) /
37 UC1=-.0
38 DO 8 N=1,10
39 UC1=UC1+.5
40 THRUST(N)=UC1
41 WRITE(3,12)THRUST
42 FORMAT(5X,2HUC,5X,10F11.7)
43 WRITE(3,11)
44 FORMAT(1H0, 6X, 1HUC)
45 U1=-.6
46 DO 7 N=1,13
47 UC1=UC1+.5
48 WRITE(3,13)
49 FORMAT(5X,2HUC,5X,10F11.7)
50 UC=UC1*1.689
51 IF(U151,52,51
52 UC=UC1+5
53 UC=UC1+1.689
54 IF(U151,52,51
55 FTA=UC/UC
56 GO TO 55
57 UC=UC1+5
52 IF(UC)<54,57,53
53 ETA=1.
   GO TO 55
54 ETA=-1.
55 RHOL=.9975*AL*AL
   IF(ETA-ETAH1)2,3,3
2  IF(ETA-ETALO)4,5,5
3  TX=RHOL*(A1*U+U*U*B1*U+U+C1*UC*UC)
   GO TO 6
4  TX=RHOL*(A3*U*U+B3*U+U+C3*UC*UC)
   GO TO 6
5  TX=RHOL*(A2*U*U*B2*U+U+C2*UC*UC)
6  IF(J)23,23,24
23 TX=TX/AM
24 THRUST(N)=TX
   IF(J)39,39,37
39 WRITE(3,10)U1,THRUST
10 FORMAT(1HO,F11.2,10F11.4)
   GO TO 7
37 WRITE(7,38)U1,THRUST
38 FORMAT(1HO,F11.2)
7 CONTINUE
25 J=-1
26 END
/*
/
JOB ZC690
EXEC FFORTRAN
C
ZC690 ERROR CALCULATOR, DS + DP CONTROL
C
DIMENSION COEF(2)
WRITE(3,14)
14 FORMAT(1H1,T5,'ZC690, ERROR CALCULATIONS, DS + DR CONTROL')
  NS = 1
  ISW=1
  WRITE(3,11)
11 FORMAT(1HO,T3,'NO.',T7,'SPFED',T14,'COEF',T29,'PERCENT CHANGE OF V
  ARIABLE',T93,'INPUT DATA',/T7,'(KTS)',T23,'DU',T30,'DTHETA',T38,
  '2*PHI1',T46,'DPHI2',T54,'DPSI1',T73,'U60',T83,'THETHI',T95,'PHIMIN'
  3,T107,'PHIU',T119,'PS160'//)
  READ(1,15)NO15,NO15
15 FORMAT(I4,6X,I14)
  1 READ(1,10)NO,COEF,U60,THETHI,PHIMIN,PHIU,PS160
10 FORMAT(I4,6X,2A4,2X,F10.5,F10.6,2F10.8,F10.6)
  IF(NO-1)GO TO (2,3,4),NS
  2 IF(NO-NO15)21,22,22
  21 IF(ISW)4,4,23
23 ISPEED=5
  PHI0=0.
  PSI0=0.
  U0=8.445
  THETO=.005155
  DUR=U0-U60
  DTHET=THETO-THETHI
  DPHI1=PHIO-PHIMIN
  DPHI2=PHIO-PHIUP
  DPSI1=PS10-PS160
  ISW=0
  WRITE(3,12)NO,ISPEED,COEF,U60,THETHI,PHIMIN,PHIU,PS160
12 FORMAT(1H10,T2,14,T9,12,T14,2A4,T70,F9.5,T82,F10.6,2F10.8,F10.6)
  1F10.8,T117,F9.6)
  GO TO 1
  4 DU=U0-U60
  DTHET=THETO-THETHI
  DPHI1=PHIO-PHIMIN
  DPHI2=PHIO-PHIUP
  DPSI1=PS10-PS160
  DELU=100.*(DUR-DU)/DUR
  DELTH=100.*(DTHET-DTHET)/DTHET
  DELPHI1=100.*(DPHI1R-DPHI1)/DPHI1R
  DELPHI2=100.*(DPHI2R-DPHI2)/DPHI2R
  DELPSI1=100.*(DPSI1R-DPSI1)/DPSI1R
  WRITE(3,13)NO,ISPEED,COEF,DELU,DELTH,DELPHI,DELPHI,DELPHI,DELPHI,DELPHI,DELPHI,DELPHI,DELPHI,DELPHI
  1THETHI,PHIMIN,PHIU,PS160
13 FORMAT(I7,14,T9,12,T14,2A4,T70,F9.5,T82,F10.6,2F10.8,F10.6)
  1F10.8,T117,F9.6)
  GO TO 1
  22 ISW=1
240
NS=2
3 IF(40-1025)24,25,25
24 IF(ISW14,4,26
26 ISPEED=15
U0=25.335
THETO=.001895
GO TO 7
25 ISW=1
NS=3
IF(ISW14,4,27
27 ISPEED=25
U0=42.725
THETO=.001809
GO TO 7
100 CALL EXIT
FND
/*
*/
241
C JOB ZC691
C EXEC FFORTRAN
C
C ZC691, ERROR CALCULATOR, DS CONTROL
C
DIMENSION COEF(2)
WRITE(3,14)
14 FORMAT(1H1,T5,'ZC691, ERROR CALCULATIONS, DS CONTROL')
   NS=1
   ISW=1
   WRITE(3,11)
11 FORMAT(1H4,'ERROR CALCULATOR, OS CONTROL')
   WRITE(1,15)
   WRITE(3,12)
   WRITE(3,13)
15 FORMAT(14,6X,14,6X,11)
   READ(1,10)NO,COEF,U1,THET1,Z1,XG,ZG
10 FORMAT(14,6X,2A4,2X,5F10.5)
   IF(NO-1100,6,6
6 GO TO (2,3,20),NS
2 IF(NO-NO15)21,22,22
21 IF(ISW)4,4,23
23 Z0=800.
   ISPRED=5
   UO=8.445
   THETO=.005155
7 DUR=U0-U1
   DTHET=THETO-THET1
   NZR=Z0-Z1
   ISW=0
   WRITE(3,12)NO,ISPRED,COEF,U1,THET1,Z1,XG,ZG
12 FORMAT(14,6X,14,6X,11)
   GO TO 1
4 DU=U0-U1
   DTHET=THETO-THET1
   NZ=Z0-Z1
   DELU=100.*(DUR-DU)/DUR
   DELTH=100.*((DTHETP-DTHET)/DTHET)
   DELZ=100.*((CZR-DZ)/DZ)
   WRITE(3,13)NO,ISPRED,COEF,DELU,DELTH,DELZ,U1,THET1,Z1,XG,ZG
13 FORMAT(14,6X,14,6X,11)
   GO TO 1
74 ISW=1
29 GO TO 1
22 ISW=1
NS=2
3 IF(NO-NO25)24,25,25
74 IF(ISW)4,4,26
24 ISPRED=15
   UO=25.335
   THETO=.001895
GO TO 7
25 ISW=1
26 NS=3
30 IF(ISW)4,4,27
27 ISPEED=25
28 U0=42.75
29 THETO=.001903
30 GO TO 7
100 CALL FXIT
END
//
// EXEC FORTRAN
//
COMMON A(55,25), AA(25), NP11
DOUBLE PRECISION A, AA, C(3), A1, A2, A3, ROOTR(10), ROOTI(10)
DIMENSION ISW(10)
EQUIVALENCE (C(1), A1), (C(2), A2), (C(3), A3)
IN=1
IOUT=3
TWOPI=6.28318
LINSPP=54
1 CONTINUE
READ(IN,100) ZWD, ZW, ZQD, ZQ, AMWD, AMW, AMQD, AMQ
IF(ZWD-999) 110,90,90
10 CONTINUE
READ(IN,100) AW, AQ, AMAWQ, AMWAW, ZWAQ, ZWAW
READ(IN,100) XUD, A11, A12, ZSTR, AMSTR, XQQ, XWO
READ(IN,100) AT, AM, AIYY, B, ZB, RHO
IPAGF=1
WRITE(IOUT, 2) IPAGF
2 FORMAT(1H1,55X,'FC140',50X,'PAGE',I6/I)
WRITE(IOUT, 4) ZWD, ZW, ZQD, ZQ, AMWD, AMW, AMQD, AMQ,
1 AM, AQ, AMAWQ, AMWAW, ZWAQ, ZWAW,
AXUD, A11, A12, ZSTR, AMSTR, XQQ, XWO,
7 AL, AM, AIYY, B, ZB, RHO
4 FORMAT(IH1,13X,'ZWD',14X,'ZW',13X,'ZQD',14X,'ZQ',13X,'AMWD',14X,
1'MW',13X,'AMW',14X,'AMQ')
ZIH ,RF16.6/I
A1H ,8X,'AW',15X,'AQ',13X,'AMAWQ',13X,'AMWAW',11X,'ZWAQ',13X,'ZWAW'/
R1H ,6F16.6/I
C1H ,8X,'XUD',14X,'A11',12X,'A12',14X,'ZSTR',11X,'AMSTR',13X,'XQQ',
C12X,'XWO'/I
31H ,10X,'1Y',15X,'1Y',15X,'1Y',14X,'1R',12X,'1R'/
41H ,6E16.6/I
ZWH = 7W
AMWH = AMH
AMQH = AMQ
WRITE(IOUT, 6)
6 FORMAT(IH1,9X,'MW',14X,'MQ',14X,'7W')
1 1H ,9X,'AI',14X,'A2',14X,'A3',14X,'4A',14X,'4A',15X,'1!'/
7IH ,10X,'1R',15X,'1R',15X,'1R',15X,'1R',15X,'1R',15X,
A1R ,15X,'1R'/
31H ,9X,'XWO',12X,'T1/21',13X,'P',15X,'D'/
1NS=16
90 CONTINUE
READ(IN,100) U
:00 FORMAT(1F10.5)
IF(U) 40,1,40
40 CONTINUE
U = U + 1.64Q
WBAR = AW/UI
QRAP = AQ+AL/UI
AMW = AMWH + AMWAW+WRAP
AMQ = AMQH + AMQAW+WRAP
ZW = ZWH + ZWAQ+QRAP+ZWAW+WRAP
N = 2
NCOL = 3
M = N*NCOL+1
DO 62 I = 1, NCOL
DO 62 J = 1, NCOL
DO 62 K = 1, M
62 A(I, J, K) = 0.
AL7=AL*AL*RHO/2.
AL3=AL2*AL
AL4=AL3*AL
AL5=AL4*AL
T1=AL3*ZWD-AM
T2=AL4*AMWD
T3=B*ZA
T4=U*AM+AL3*U*ZQ
T5=AL?*U*ZW
T6=AL4*U*AMO
T7=AL5*AMQD-ATYY
T8=AL3*U*AMW
T9=AL4*ZQD
A1 = T7
A2 = T6
A3 = T5
CALL PLACE(1, 1, C)
A1 = T2
A2 = T8
A3 = 0.
CALL PLACE(1, 2, C)
A1 = 0.
A2 = T9
A3 = T4
CALL PLACE(2, 1, C)
A2 = T1
A3 = T5
CALL PLACE(2, 2, C)
A2 = 0.
A3 = AL3*U*(XQO*QRAP+XWO*WRAP)-AM*WRAP*U
CALL PLACE(3, 1, C)
A3 = 0.
CALL PLACE(3, 2, C)
A2 = AL3*XUD-AM
A3 = AL2*U*(A11+A11+A12)
CALL PLACE(3, 3, C)
A2 = AL3*AMSTR*2.0U
A3 = 0.
CALL PLACE(1, 3, C)
A2 = 0.
A3 = AL3*AMSTR*2.0U
CALL PLACE(2, 3, C)
CALL CHRFQV
NP1=NP1+1
CALL MILLER(AA, NP11, ROOT, ROOT1, ISW, IERR)
DC 74 I = 1, NP11
IF(DARS(ROOT1(I))=1.0-5)74, 74, 74
74 CONTINUE
    WRITE(IOUT,75)
75 FORMAT(1H,'NO ROOT WITH IMAGINARY PART FOUND')
    GO TO 50
76 TR = DABS(RONTR(I))
    TI = DABS(ROOTI(I))
    WN = SQRT(TR*TR+TI*TI)
    THALF = .693/TR
    P = TWOPI/TI
    D = TR/WN
    IF(LINS-LINSPP)80,79,79
79 IPAGE=IPAGE+1
    WRITE(IOUT,2) IPAGE
    WRITE(IOUT,6)
    LINS=6
80 CONTINUE
    WRITE(IOUT,82) AMW,AMQ,ZW
    WRITE(IOUT,82)(AA(I),I=1,NP1),U
    WRITE(IOUT,82)(RONTR(I),ROOTI(I),I=1,NP1)
    WRITE(IOUT,87) WN,THALF,P,D
    WRITE(IOUT,93)
82 FORMAT(1H,/)  LINS=LINS+5
83 FORMAT(1H/)
    GO TO 90
90 CALL EXIT
END
SUBROUTINE PLACE(I,J,X)
DOUBLE PRECISION A(5,5,25), X(1)
COMMON A
KK = 3
DO 400 K=1,3
   A(I,J,KK) = X(K)
400 KK=KK-1
RETURN
END
SUBROUTINE CHREON
DOUBLE PRECISION CNF,C1,C2,AA,A,SB,SA
DIMENSION A(5,5,25),CNF(25),MAT(8,8),C1(25),C2(25),AA(25)
COMMON A, AA, NP11
N = 2
NCOL = 3
1 M=N*NCOL+1
1501 N1=N+1
NCOL2= NCOL*NCOL
L1 = N1 + 1
DO 303 I=1,N
C1(I)=0.
AA(I)=0.
303 C2(I) = 0.
NP1=N+1
C FIND DEGREE OF EACH MATRIX ELEMENT
DO 2 I=1,NCOL
DO 2 J=1,NCOL
MAT(I,J)=0
DO 2 K=1,N
IF(A(I,J,K)) 600,?,600
600 MAT(I,J)=K
2 CONTINUE
C TRIANGULARIZE THE MATRIX
J3=0
J9=0
DO 3 I=J1,NCOL
IF(MAT(I,J3)) 100,301, 601
601 IF(MAT(I,J3)) 602,3,602
602 J9=J9+1
J3=J3+1
3 CONTINUE
C J1 = COLUMN NUMBER
C J9 = NUMBER OF NON-ZERO ELEMENTS IN THIS COLUMN
C J3 = LAST NON-ZERO ELEMENT IN THIS COLUMN
10 IF(J9-J3) 100,603,12
603 IF(J3-J1) 100,112,204
204 DO 4 J=J1,NCOL
J2= MAX0(MAT(J3,J),MAT(J1,J))
J4=MAT(J3,J)
MAT(J3,J)=MAT(J1,J)
MAT(J1,J)= J4
DO 4 K=1,J2
SA=A(J3,J,K)
A(J3,J,K)=A(J1,J,K)
A(J1,J,K)=SA
4 CONTINUE
GO TO 112
12 J3=J1+1
DO 111 I=J3,NCOL
11 IF(MAT(I,J3)) 100,111,205
205 IF(MAT(J3,J1)) 100,14,206
206 IF(MAT(I,J3)-MAT(J1,J3)) 14,15,15
252
C INTERCHANGE ROW I WITH J1
14 DO 6 J= J1, NCOL
J2= MAXO(MAT(J1, J), MAT(I, J))
J4= MAT(J1, J)
MAT(J1, J)=MAT(I, J)
MAT(I, J)=J4
6 CONTINUE
GO TO 13

C
15 J7=MAT(I, J1)
J5=MAT(J1, J1)
J6=J7-J5
SB=A(I, J1, J7)/A(I, J1, J5)
IF(ABS(SB)-4.116,207,207)
207 IF(J6)100,14,16
16 DO 19 J= J1, NCOL.
J5=MAT(J1, J1)
DO 19 K=1, J5
J7= K+J6
IF(J7-M)17,17,110
17 IF(ABS(A(I, J7)-SB*A(I, J,K))-2,F-15)18,18,208
208 A(I, J7)= A(I, J7)-SB*A(I, J,K)
GO TO 19
19 A(I, J7):=0.
19 CONTINUE
110 DO 7 J=J1, NCOL
J7= MAXO(MAT(I, J), MAT(J1, J)+J6)
MAT(I, J)=0
7 CONTINUE
111 CONTINUE
GO TO 10
117 J1=J1+1
IF(J1-NCOL)10,210,210
C GET PRODUCT OF DIAGONAL ELEMENTS
210 DO 115 J=J1, NCOL
J2=MAT(I, J)
9 C1(K)=A(I, J, K)
113 IF(J-I)100,14,211
211 DO 9 K=1, NP1
9 C2(K)=COF(K)
116 GO TO 116
116 COE(K)=0.
IF(J2)100,115,212
212 DO 117 K=1, J2
117 J10=1, NP1
J11= K+J10-1

253
COE(J11) = COE(J11) + C1(K) * C2(J10)
117 CONTINUE
NP1 = J11
GO TO 115
114 DO 118 K = 1, J2
119 COE(K) = C1(K)
NP1 = J2
115 CONTINUE
100 I = NP1
DO 401 I = 1, NP1
AA(I) = COE(I)
401 I = I - 1
NP1 = NP1 - 1
RETURN
END
SUBROUTINE MULLER(COE, N1, ROOTR, ROOTI, ISW, IERR)
MULLER ROUTINE FOR ZEROS OF POLYNOMIALS WITH REAL COEFFICIENTS

COE is the array of polynomial coefficients ordered from highest
to lowest power of x
N1 is the degree of the polynomial
ROOTR is the array of real components of the roots
ROOTI is the array of imaginary components of the roots
ISW is an array defining the validity of the roots
ISW(N) = 0 THE NTH ROOT HAS BEEN STORED IN ROOTR(N) AND
ROOTI(N)
ISW(N) = 1 THE NTH ROOT HAS BEEN STORED IN ROOTR(N) AND
ROOTI(N), BUT IT MAY NOT BE VALID
IERR is an error code which has the following significance.
IERR = 0 ALL ROOTS FOUND CORRECTLY
IERR = 1 ONE OR MORE ROOTS MAY BE INVALID. TEST THE
ISW ARRAY.
IERR = 2 POLYNOMIAL DEGREE IS LESS THAN 1
IERR = 3 POLYNOMIAL DEGREE IS LESS THAN N1

For a polynomial of degree N1 the COE array should be dimensioned
N1+1 in the user program. The other arrays should be dimensioned
N1 in the user program.
The polynomial is scaled to avoid arithmetic overflow. All
scaling uses factors of 16. To change to factors of x, set BASE
= X and CONST = LN(X) in subroutine.
This subroutine uses double precision arithmetic. Single precision
is not recommended.

DIMENSION COE(1), ROOTR(1), ROOTI(1), ISW(1)
DOUBLE PRECISION COE, TF7, TF3, NIV, TEE7, DE15, TF13, HELL, TF42, ALP1R,
1 ALP2R, ALP3R, TEST1, TAU2, AXB, TF5, TF4, UPP, TEMR, DE16, TF14, HELL, ROOTR,
2 ALP1I, ALP2I, ALP3I, TEST2, AXI, TF4, TF7, TET, TF51, TELI, TF15, TAU2, ROOTI,
3 BET1P, BET2R, BET3R, ALP4R, TEL1, TEM, TEQ, TAU, TF10, TF12, TF1A, TEM1, Z1,
4 BET1I, BET2I, BET3I, ALP4I, Z2, N1, 0, FACTOR

BASE = 16.
CONST = 2.77259

CALL MASK(0)
IF(N1 - 1) 27, 7A, 7A
27 IERR = 2
GO TO 193
78 IERR = 0
FACTOR = 0.
N2=N1+1
N4=0
I=N1+1
10 IF(COE(1)) 9, 7, 9
7 N4=N4+1
ROOTR(N4)=0.
ROOTI(N4)=0.
I=I+1
19 IF(N4-N1) 19, 37, 19

255
9 CONTINUE
IF(COE(I)) = 190, 192, 190
190 TEMP = DABS(COE(I)/COF(I))
TEMP = ALOG(TEMP)/CONST/(I-1)
K2 = TEMP + SIGN(.5, TEMP)
TEMP = DABS(COE(I))
TEMP = ALOG(TEMP)/CONST
K1 = TEMP + SIGN(.5, TEMP)
DO 191 I = 1, N2
191 COE(I) = COE(I)/BASE**(K1 + K2*(I-1))
FACTOR = BASE**K2
GO TO 10
192 IERR = 3
193 DO 194 I = 1, N1
ROOTR(I) = 0.
ROOTI(I) = 0.
194 ISWU(I) = 1
RETURN
10 AXR=0.8
AXI=0.
L=1
N3=1
ALP1R=AXR
ALP1I=AXI
M=1
GO TO 99
11 BET1R=TFMR
BET1I=TEMI
AXR=0.85
ALP2R=AXR
ALP2I=AXI
M=2
GO TO 99
12 BET2R=TFMR
BET2I=TEMI
AXR=0.9
ALP3R=AXR
ALP3I=AXI
M=3
GO TO 99
13 BET3R=TFMR
BET3I=TEMI
14 TE1=ALP1R-ALP3R
TE2=ALP1I-ALP3I
TE5=ALP3R-ALP2R
TE6=ALP3I-ALP2I
TEM=TE5*TE5*TE6*TE6
TE3=(TF1*TE5+TF2*TF6)/TF4
TE4=(TF2*TF5-TE1*TE6)/TF4
TE7=TF3+1.
TE9=TF3*TE3-TF4*TF4
TE10=2.*TE3*TF4
TE15=TF7*TF3-TF4*TF31
TE16=TF7*TF31+TF4*TF3P
TE11=TE3*GET2R-TE4*GET2I+GET1R-DE15
TE12=TE3*GET2I+TE4*GET2R+3ET1I-DE16
TE7=TE9-1.5
TE1=TE9*GET2R-TE10*GET2I
TE2=TE9*GET2I+TE10*GET2R
TE13=TE1-GET1R-TE7*GET3P+TE10*GET3I
TE14=TE2*GET11-TE7*GET1-TE10*GET3R
TE15=DE15*TF3-DE16*TF4
TE16=DE15*TF4+DE16*TF3
TE1=TE13*TF13-TE14*TF14-4.*(TF11*TE15-TE12*TF16)
TE2=2.*TF13*TF14-4.*(TF17*TE15+TF11*TF16)
TEST1=0ABS(TF1)
TEST2=0ABS(TF2)
IF(TEST1-TEST2) 300,301,301
300 DIV=TEST2
UPP=TFST1
GO TO 303
301 DIV=TFST1
UPP=TEST2
IF(DIV-1.0-70) 999,303,303
999 DIV= 1.0-70
303 TEM=DIV*DOSRT(1.0*(UPP/DIV)*(UPP/DIV))
IF(TE1113,113,112
113 TE4=DOSRT(1.0*(TE111)
TE3=5*TF2/TF4
GO TO 111
112 TF3=DOSRT(1.5*(TFM+TE1))
IF(TE2110,200,200
110 TE3=-TF3
200 TF4=5*TF2/TF3
111 TE7=TE13+TF3
TFR=TE14+TF4
TE9=TE13-TE3
TE10=TF14-TE4
TE1=2.*TE15
TE2=2.*TF4
IF(TEF7+TFR+TE9-TE9-TE10*TF10)*TF0,204,205
204 TE7=TF0
TF8=TF10
205 TEM=TE7*TF7+TFR+TFR
IF(TEM-1.0-70) 999,997,997
999 TEM= 1.0-70
997 TE3=(TE1*TE7+TF7+TFR)/TFM
TE4=(TF5*TF7-TF1+TFR)/TFM
AXR=ALP3R+TE3*TE5-TE4*TF5
AX1=ALP3I+TE3*TE6+TF4*TF5
ALP4K=AXR
ALP4I=AX1
N=4
GO TO 99
15 N=1
38 N1=DARS(HELI)+DARS(RELI)
TE7=DARS(ALP3R-AXR)+DARS(ALP3I-AX1)
TFE7=DARS(AXR)+DABS(AX1)

257
C IS THE FUNCTION VALUE NEAR ZERO?

\[
\text{IF(01} -1.0-20 \quad 161,161, \quad 16
\]

C IS THE ROOT SMALL?

\[
161 \quad \text{IF(TEF7} -1.00-031162,16,16
\]

C IS THE CURRENT ESTIMATE FOR THE ROOT ESSENTIALLY

\[
C \quad \text{THE SAME AS THE PREVIOUS ESTIMATE?}
\]

\[
162 \quad \text{IF(TE7} -1.00-12118,17,17
\]

C ARE THE CURRENT AND PREVIOUS ESTIMATES OF THE ROOT ESSENTIALLY

\[
C \quad \text{THE SAME WHEN COMPARED TO THE MAGNITUDE OF THE ROOT?}
\]

\[
16 \quad 02=\text{TE7/ TEF7}
\]

\[
\text{IF(02} -1.0-7)18,18,17
\]

\[
17 \quad N3=N3+1
\]

\[
\text{ALP1R=ALP}2R
\]

\[
\text{ALP1I=ALP}2I
\]

\[
\text{ALP2R=ALP}3R
\]

\[
\text{ALP2I=ALP}3I
\]

\[
\text{ALP3R=ALP}4R
\]

\[
\text{ALP3I=ALP}4I
\]

\[
\text{RET1R=RET}2R
\]

\[
\text{RET1I=RET}2I
\]

\[
\text{RET2R=RET}3R
\]

\[
\text{RET2I=RET}3I
\]

\[
\text{RET3R=TEMR}
\]

\[
\text{RET3I=TEMI}
\]

\[
\text{IF(N3-200)14,25,25}
\]

\[
25 \quad ISWT = 1
\]

\[
26 \quad \text{ISWT} = 0
\]

\[
24 \quad N4 = N4 + 1
\]

\[
\text{ISW(N4) = ISWT}
\]

\[
\text{ROOTR(N4)=ALP}4R
\]

\[
\text{ROOTI(N4)=ALP}4I
\]

\[
N3=0
\]

\[
41 \quad \text{IF(N4-N1)}30,37,37
\]

\[
37 \quad \text{CONTINUE}
\]

\[
\text{IF(FACTOR)140,140,138}
\]

\[
138 \quad \text{DO 139 I} = 1,N1
\]

\[
\text{ROTR(I) = ROOTR(I)*FACTOR}
\]

\[
\text{ROOTI(I) = ROOTI(I)*FACTOR}
\]

\[
139 \quad \text{COE(I)} = \text{COF(I)}*\text{BASE}**(K1*K2*(I-I))
\]

\[
\text{COE(N2)} = \text{COF(N2)}*\text{BASE}**(K1*K2*N1)
\]

\[
140 \quad \text{DO 141 I} = 1,N1
\]

\[
\text{IF(ISWT11)}141,141,142
\]

\[
141 \quad \text{CONTINUE}
\]

\[
\text{IERR = 0}
\]

\[
\text{GO TO 3001}
\]

\[
142 \quad \text{IERR = 1}
\]

\[
3001 \quad \text{RETURN}
\]

\[
30 \quad \text{IF(DARS(ROOTI(N4)/ROOTR(N4))} -1.0-5)10,10,131
\]

\[
131 \quad \text{IF(ISWT} 31,137,31
\]

\[
132 \quad \text{GO TO (133,134), L}
\]

\[
133 \quad N4 = N4 + 1
\]

\[
N4 = N4
\]

\[
\text{GO TO 135}
\]

258
134 \( M_4 = N_4 - 1 \)
135 \( \text{ROOTR}(M_4) = AXR \)
\( \text{ROOTI}(M_4) = -AXI \)
\( \text{ISW}(M_4) = 0 \)
\( \text{IF}(M_4 - N_1) \) 10, 37, 37
31 GO TO(32, 10), L
32 AXR = ALP1R
\( AXI = -ALP1I \)
\( ALP2I = -ALP1I \)
\( M = 5 \)
GO TO 99
33 BET1R = TFM1R
\( \text{RET1I} = \text{TEM1} \)
\( AXR = ALP2R \)
\( AXI = -ALP2I \)
\( ALP2I = -ALP2I \)
\( M = 6 \)
GO TO 99
34 BET2R = TEMR
\( \text{RET2I} = \text{TEM1} \)
\( AXR = ALP3R \)
\( AXI = -ALP3I \)
\( ALP3I = -ALP3I \)
\( L = 2 \)
\( M = 3 \)
99 TEMR = COE(1)
\( \text{TEM} = 0.0 \)
DO 100 I = 1, N1
\( \text{TEM1} = \text{TFMR} \cdot AXR - \text{TFMI} \cdot AXI \)
\( \text{TEM1} = \text{COE} \cdot A X R + \text{TFMR} \cdot AXI \)
100 TEMR = \( \text{TE1} + \text{COE}(1+1) \)
\( \text{HELI} = \text{TFMR} \cdot A X R \)
\( \text{BEL} = \text{TFMI} \cdot A X R \)
42 IF(N4) 102, 103, 10?
102 DO 101 I = 1, N4
\( \text{TEM1} = \text{AXR} - \text{ROOTR}(I) \)
\( \text{TEM2} = \text{AXI} - \text{ROOTI}(I) \)
\( \text{TE1} = \text{TEM1} \cdot \text{TEM1} + \text{TEM2} \cdot \text{TEM2} \)
\( \text{TE2} = (\text{TFMR} \cdot \text{TFM1} + \text{TFMI} \cdot \text{TEM2}) / \text{TE1} \)
\( \text{TEMI} = (\text{TFMI} \cdot \text{TFM1} - \text{TEMR} \cdot \text{TEM2}) / \text{TE1} \)
101 TEMR = TF2
103 GO TO(11, 12, 13, 15, 33, 34), M
END
/*
*/
259
ARE THE CURRENT AND
PREVIOUS ESTIMATES OF
THE ROOT ESSENTIALLY
THE SAME WHEN
COMARED TO THE
MAGNITUDE OF THE ROOT

17.034-9
16 - 1
17

17.045-2
27

\[ c_{ij} = a_{ii} \]
\[ c_{ij} = a_{ii} \]
\[ c_{ij} = a_{ii} \]
\[ c_{ij} = a_{ii} \]
JOB EC320
// EXEC FORTRAN
COMMON A(5,5,25), AA(25), NP1
DOUBLE PRECISION A, AA, C(3), A1,A2,A3, ROOTR(10), ROOTI(10)
DIMENSION ISW(10)
EQUIVALENCE (C(1),A1),(C(2),A2), (C(3),A3)
IN = 1
IOUT = 3
LINSPP=56
4 CONTINUE
READ (IN,10)YVD, YV, YPD, YP, YRD, YR
IF(YVD-999. )R,6,6
6 CALL EXIT
8 CONTINUE
READ (IN,10)AKVD, AKV, AKPD, AKP, AKRD, AKR
READ (IN,10)ANVD, ANV, ANPD, ANP, ANRD, ANR
READ (IN,10)AL, AM, AIX, AIZ, R, ZB, RHO
READ(IN,10) YVAV, YVAR, YPAP, AKVAV, AKPAP, ANVAV, ANRAR, ANAVR
READ(IN,11) RRAR, VBAR, PBRAR
READ(IN,11) IDIV
10 FORMAT(8F10.5)
11 FORMAT(16I5)
YVH= YV
YPH= YP
AKVH=AKV
AKPH= AKP
ANVH= ANV
ANRH= ANR
IPAGE = 1
WRITE(IOUT,12) IPAGE
12 FORMAT(1H1,50X,'EC320','0X,'PAGE',110/)
WRITE(IOUT,42)
WRITE(IOUT,44)
WRITE(IOUT,6)
WRITE(IOUT,46)
WRITE(IOUT,48)
WRITE(IOUT,49)
WRITE(IOUT,50)
WRITE(IOUT,50) RRAR, VBAR, PBRAR
40 FORMAT(1H ,PF16.6)
42 FORMAT(1H ,8X,'YVD',13X,'YPD',13X,'YP',13X,'YRD',13X,'YR',113X,'YR')
44 FORMAT(1H ,8X,'KVD',13X,'KV',14X,'KPD',14X,'KP',14X,'KR',113X,'KR')
46 FORMAT(1H ,8X,'NVD',13X,'NV',14X,'NPD',14X,'NP',14X,'NR',113X,'NR')
48 FORMAT(1H ,8X,'L',15X,'M',15X,'IX',15X,'IX',15X,'R',15X,'ZR',114X,'ZR')
49 FORMAT(1H ,8X,'YVAV',12X,'YVAR',12X,'YPAP',12X,'AKVAV',12X,'AKPAP',12X,'ANVAV',12X,'ANRAR',12X,'ANAVR')
50 FORMAT(IH,8X,'RBAR',12X,'VRAR',12X,'PBAR')
WRITE(INOUT,52)
52 FORMAT(1H/)
WRITE(INOUT,54)
54 FORMAT(1H,8X,'YV',14X,'YP',14X,'KV',14X,'KP',14X,'NV',14X,'NP'/
        1    1H,8X,'A1',14X,'A2',14X,'A3',14X,'A4',14X,'A5',14X,'U'/
LINS = 10
56 CONTINUE
READ(IN,10) U
IF(U)58,4,58
58 CONTINUE
U = U* 1.689
IF(DIV)62,64,67
62 CONTINUE
DIV = U
GO TO 65
64 DIV = 1.
65 CONTINUE
YV = YV + (YVAV*VRAR+AL*YVAR*PBAR)/DIV
YP = YPH + AL*YPAP*PBAR/DIV
AKV = AKVH+ AKVAV*VBAR/DIV
AKP = AKPH + AL*AKPAP*PBAR/DIV
ANV = ANVH + ANVAV*VAR/DDV
ANR = ANRH + (AL*ANRAR*VAR+ANAVR*VAR)/DIV
N = 2
NCOL = 3
M = N*NCOL+1
DO 60 I=1,NCOL
DO 60 J=1,NCOL
DO 60 K=1,M
60 AI,J,K)=0.
RL = RHO*AL/Z
RL2 = RL* AL
RL3 = RL2* AL
RL4 = RL3* AL
RL5 = RL4* AL
RL4U = RL4* U
RL3U = RL3* U
A1 = 0,
A2 = RL3*YVD-AM
A3 = RL7*U*YV
CALL PLACE(1,1,C)
A2 = RL4*YPD
A3 = RL7*U*YP
CALL PLACE(1,2,C)
A2 = RL4*YPD
A3 = RL7*U*ANV
CALL PLACE(1,3,C)
A2 = RL4*ANVD
A3 = RL7*U*ANV
CALL PLACE(1,4,C)
A2 = RL5*ANVD
A3 = RL7*U*ANV
CALL PLACE(1,5,C)
A2 = RL5*ANVD
A3 = RL7*U*ANV
A3=RL4U*ANP
CALL PLACE(3,2,C)
A2=RL5*ANRD-AIZ
A3=RL4U*ANR
CALL PLACE(3,3,C)
A1=RL4*AKVD
A2=RL3U*AKV
A3=0.
CALL PLACE(2,1,C)
A1=RL5*AKPD-AIX
A2=RL4U*AKP
A3=R*7B
CALL PLACE(2,2,C)
A1=RL5*AKRD
A2=RL4U*AKR
A3=0.
CALL PLACE(2,3,C)
CALL CHREQN
NP1=NP1+1
IF(LINS-LINSPP)=0,70,70
70 IPAGE=IPAGE+1
WRITE(OUT,12) IPAGF
WRITE(OUT,54)
LINS=5
80 WRITE(OUT,40) YV, YP, AKV, AKP, ANV, ANP
WRITE(OUT,40) (AA(I),I=1,NP1), U
CALL MULLER(AA, NP1, ROOT, ROOTI, ISW, IFRR)
WRITE(OUT,40) (ROOTR(I),ROOTT(I),I=1,NP11)
WRITE(OUT,52)
LINS = LINS + 4
GC TO 56
END
// JOB EC150
// EXEC FORTRAN
DIMENSION Y(500), T(500), NAME(2)
IN=1
IOUT=3
PI = 3.14159
TWOPI = 6.28318
10 READ(IN,20) N
20 FORMAT(16I5)
IF(IN>23,22,23
22 CALL EXIT
23 CONTINUE
24 FORMAT(F10.5,?A4)
READ(IN,24) U, NAME
25 FORMAT(2E15.7)
READ(IN,25) ARG, T(1)
Y(1) = 0.
DO 30 I=2,N
READ(IN,25) Y(I), T(I)
Y(I) = Y(I)-ARG
30 CONTINUE
READ(IN,35) A,R,G,D
35 FORMAT(8F10.5)
IPAGE = 1
LINS = 6
LINSPP = 52
WRITE(IOUT,36) IPAGE
IPAGE = IPAGE+1
U = U+1.689
WRITE(IOUT,38) U
WRITE(IOUT,40) NAME
36 FORMAT((H150,'EC150', 40X,'PAGE',17/)
38 FORMAT((H150,'U=', 6F13.6/)
40 FORMAT((H150,'Y=', 14X,4A4/)
41 FORMAT((H150,'REE16.6)
DO 43 I=1,N
WRITE(IOUT,41) T(I), Y(I)
LINS = LINS + 1
IF(LINS-LINSPP)43,47,47
42 WRITE(IOUT,36) IPAGF
IPAGE = IPAGF + 1
WRITE(IOUT,40) NAME
LINS = 4
43 CONTINUE
C FIND LAST MIN
DO 100 J=1,N
IF( Y(J) - Y(J-1));100,100,50
50 IF(Y(J-1) - Y(J-2));110,100,100
100 CONTINUE
110 YMIN = Y(J-1)
TS = T(J-1)
K = J-1
C FIND PRECEDING MAX
283
DO 150 I = 1, K
J = K-I+1
IF(Y(J)-Y(J-1))125,150,150
125 IF(Y(J-1)-Y(J-2))150,150,160
150 CONTINUE
160 YMAX = Y(J-1)
T4 = T(J-1)
C COMPUTE TO
TZ = 4.61/D
PEST = AMOD(R*T4, TWOPI)
PEST = PEST - D
TEMP1 = EXP(G*(T5-T4))
TEMP2 = EXP(-A*T4)
A2EST = -(YMAX-YMIN*TEMP1)/(TEMP1*EXP(-A*T5)+TEMP2)
A1EST = (YMAX+A2EST*TEMP2)/EXP(-G*T4)
A3EST = -(A1EST+A2EST*COS(PEST))
WRITE(16,36) IPAGE
IPAGE = IPAGE + 1
WRITE(16,200) YMIN, YMAX, TZ, T4, T5
WRITE(16,204) A, B, G, PEST, A1EST, A2EST, D, A3EST
          1H, 5F16.6/)
          13X, 'D', 14X, 'A3'/1H, 8F16.6/)
GO TO 10
/
/
// JOB EC330
// EXEC FORTRAN
DIMENSION THTA(500), T(500), NAME(2)
IN= 1
IOUT = 3
PI= 3.141593
2 READ(IN,10) N
IF(N)4,3,4
3 CALL EXIT
4 CONTINUE
READ(IN,8) U, NAME
8 FORMAT(F10.5,2A4)
10 FORMAT(16I5)
READ(IN,20) YZ, T(1)
20 FORMAT(2E15.7)
DO 30 I=2,N
READ(IN,20) THTA(I), T(T)
THTA(I)= THTA(I)- YZ
30 CONTINUE
IPAGE=1
LINS=6
LINSPP=52
WRITE(IOUT,31)IPAGE
31 FORMAT(1H1,50X,'EC330',40X,'PAGE',I7f)
IPAGE = IPAGE + 1
U = U*1.689
WRITE(IOUT,32) U
32 FORMAT(1H1,'U=',F13.6/)WRITE(IOUT,33) NAME
33 FORMAT(1H1,9X,'T',14X,2A4/)DO 36 I=1,N
WRITE(IOUT,34)T(I), THTA(I)
34 FORMAT(1H1,PE16.6)
LINS = LINS + 1
IF(LINS=LINSPP)36,35,35
35 WRITE(IOUT,31)IPAGE
IPAGE = IPAGE + 1
WRITE(IOUT,33)NAME
LINS = 4
36 CONTINUE
WRITE(IOUT,31) IPAGE
C FIND TIME OF FIRST CROSSING
DO 40 I=6, N
IF (THTA(I)) 50, 40, 40
40 CONTINUE
50 Y1 = THTA(I-1)
Y2 = THTA(I)
X1 = T(I-1)
X2 = T(I)
Y1 = X1* Y1* (X2-X1)/(Y1-Y2)
C FIND TIME OF SECOND CROSSING
DO 60 K= 1, N
IF (THTA(K)) 60, 60, 70
60 CONTINUE
70 Y1 = THTA(K-1)
    Y2 = THTA(K)
    X1 = T(K-1)
    X2 = T(K)
    T2 = X1* Y1*(X2-X1)/(Y1-Y2)
C COMPUTE PERIOD
    P = T2-T1
C COMPUTE BFTA
    BETA = PI/P
C COMPUTE PSI
    PSI = BFTA*T1- PI/2
C COMPUTE T3 AND T4
    TEMP = P/2
    T3 = T1-TEMP
    T4 = T1+TEMP
C FIND X3
DO 80 I = 1, N
    IF(T(I)-T3)80,80,90
80 CONTINUE
90 Y1 = THTA(I-1)
    Y2 = THTA(I)
    X1 = T(I-1)
    X2 = T(I)
    Y3 = Y1+(T3-X1)*(Y2-Y1)/(X2-X1)
C FIND X4
DO 100 K = 1, A
    IF(T(K)-T4)100,100,110
100 CONTINUE
110 Y1 = THTA(K-1)
    Y2 = THTA(K)
    X1 = T(K-1)
    X2 = T(K)
    Y4 = Y1+(T4-X1)*(Y2-Y1)/(X2-X1)
C COMPUTE ALPHA
    ALPHA = ALOG(-Y3/Y4)/P
C COMPUTE A2
    A2 = Y3/FXP(-ALPHA*T3)
C COMPUTE A1
    A1 = -A2*COS(PSI)
C COMPUTE GAMMA
    TEMP = A2*EXP(-ALPHA*6+1)*COS(BETA*6.-PSI)
    ARG = THTA(41.-T4)/A1
    IF(ARG)112,112,114
112 GAMAN = 10.* ALPHA
    ALPN = ALPHA
    WRITE(TOUT,170) ALPHA,BETA,GAMAN,PSI,A1,A2
    GO TO 126
114 CONTINUE
    GAMAN = -ALOG(ARG)*A.
    WRITE(TOUT,170) ALPHA,BETA,GAMAN,PSI,A1,A2
    TEMP1 = A1*EXP(-GAMA*T3)
    TEMP2 = Y3-TEMP1
    ALPN = TEMP2/(-Y4+A1*EXP(-GAMA*T4))
    ALPN = ALOG(ALPHA)/P
A2N  =  TEMP?/FXP(-ALPN*TA)
A1N  =  -COS(PSI)*A2N
TEMP  =  A2N*EXP(-ALPN*A,T)*COS(BETA*A,-PSI)
ARG  =  (THTA(4)-TEMP)/A1N
IF(ARG)  121, 121, 122
121  GAMAN  =  10. * ALPN
      GO TO  125
122  GAMAN  =  -ALCG(ARG) /6.
125  CONTINUE
      WRITE(INUT,170)ALPN,BETA,GAMAN,PSI,A1N,A2N
126  CONTINUE
C  FIND  MAX  ORD
   DO  130  J=  1,N
      IF(THTA(J)-THTA(J+1) 130,130,140
130      CONTINUE
140  T3P  =  T(J)
      Y3P  =  THTA(J)
C  FIND  MIN  ORD
   DD  150  K=  1,N
      IF(THTA(K) 145,150,150
145      IF(THTA(K)-THTA(K+1) 160,150,150
150      CONTINUE
160  T4P  =  T(K)
      Y4P  =  THTA(K)
      A2  =Y3P/(-COS(PSI)*EXP(-GAMAN*T3P)+EXP(-ALPN*T3P)*
      COS(BETA*T3P-PSI))
      A1  =-A2*COS(PSI)
      WRITE(INOT,170)ALPN,BETA,GAMAN,PSI,A1,A2
170  FORMAT(/1H ,9X,'A',15X,'R',15X,'G',15X,'P',15X,'A',14X,'A2'/
      11X,6F15.6//)
      GO TO  2
END
/
*/
// JOB EC310
// EXEC FORTRAN
DIMENSION T(500), Y(500), X(10)
DIMENSION RUFF(3000), FCN1(500), IY(2)
COMMON T,Y,NPTS,FCN1,ISWI
IN = 1
ICUT = 3
IOPEN = 0
3 CONTINUE
LINS = 99
LINSPP = 57
READ(IN,5) N, NUMSIG, MAXIT, IPRINT, NPTS, ISWI, IPlot
5 FORMAT(16I5)
IF(IPlot) 6,9,6
6 IF(IOPEN) 9,7,R
7 CALL PLOTS(RUFF, 12000, 7)
IOPEN = 1
8 CONTINUE
IF(N) 20,10,20
10 CONTINUE
IF(IOPEN)12,15,12
12 CONTINUE
CALL PLOT(6,0,0,0,999)
15 CALL EXIT
20 CONTINUE
READ(IN,22) U, IY
22 FORMAT(F10.5,?A4)
U = U*1.689
READ(IN,25) ARG, T(I)
Y (I) = 0.
DO 30 I = 1,NPTS
READ(IN,25) Y(I),T(I)
Y(I) = Y(I) - ARG
30 CONTINUE
C READ INITIAL A, R, G, P, A1, A2, D, A3
READ(IN,35) (X(I), I = 1,P)
35 FORMAT(10F10.5)
IF(ISWI) 40,37,40
37 CONTINUE
CALL SYSTEMIN, NUMSIG, MAXIT, IPRINT, X
ISWI = 1
40 CONTINUE
CALL AUXFCN(X,DUM,INUM)
DO DO I = 1, NPTS
IF(LINSPP - LINS)4?,42,44
42 LINS = 0
WRITE(IN, 44) (X(I),K=1,P)
44 FORMAT(1H1,9X,'A',15X,'P',15X,'G',15X,'D',15X,'A3/1H,9E16.6//1H,14X,'T',19X,'VF',19X,'YT/1)
47 CONTINUE
WRITE(IN, 44) T(I), FCN1(I), Y(I)
LINS = LINS + 1
48 FORMAT(1H, 3E20.6)  
90 CONTINUE  
   IF( IPILOT ) 50, 3, 50  
50 CONTINUE  
   CALL FCNPLT(T, F1, NI, Y, NPTS, X, U, R, IY)  
   WRITE( IOUT, 90)  
90 FORMAT(1H1)  
   GO TO 2  
FND
SUBROUTINE FCNPLT(X, YF, YT, N, VAR, U, NVAR, IV)
DIMENSION VAR(1)
DIMENSION X(1), YF(1), YT(1), NUMX(2), NUMY(2)
DIMENSION ITAB(3), IFIT(2), IWK(10), IV(1)
IU =IHEX(14,4,0,7,14,4,0)
IWK(1)=IHEX(12,1,4,0,7,14,4,0)
IWK(2)=IHEX(12,2,4,0,7,14,4,0)
IWK(3)=IHEX(12,7,4,0,7,14,4,0)
IWK(4)=IHEX(13,7,4,0,7,14,4,0)
IWK(5)=IHEX(12,1,15,1,7,14,4,0)
IWK(6)=IHEX(12,1,15,2,7,14,4,0)
IWK(7)=IHEX(12,4,4,0,7,14,4,0)
IWK(8)=IHEX(12,1,15,3,7,14,4,0)
IDSH=IHEX(6,0,6,0,6,0,4,0)
ITAB(1)=IHEX(14,3,12,1,12,2,14,4)
ITAB(2)=IHEX(13,3,12,1,14,3,12,5)
ITAB(3)=IHEX(12,4,4,0,4,0,4,0)
IFIT(1)=IHEX(12,9,12,9,14,3,14,3)
IFIT(2)=IHEX(12,5,12,4,14,4,0)
IX=IHEX(14,3,4,0,4,0,4,0)
DIV=20.
NUMX(1)=X(1),
NUMX(2) = X(N)
YMAX=999.
YMIN=999.
DO 100 I=1,N
IF(YF(I)-YMAX)40,40,30
30 YMAX=YF(I)
40 IF(YF(I)-YMIN)50,50,60
50 YMIN=YF(I)
60 IF(YT(I)-YMAX)80,80,70
70 YMAX=YT(I)
80 IF(YT(I)-YMIN)90,90,100
90 YMIN=YT(I)
100 CONTINUE
DUMY(1)=YMIN
DUMY(2)=YMAX
CALL PLOT(0.0,1.0,23)
CALL SCALE(DUMX,9.5,2,1,DIV,1)
CALL SCALE(DUMY,6.0,2,1,DIV,1)
CALL AXIS(4.0,0.0,XH,4.0,90.0,DIV,1)
CALL AXIS(4.0,0.0,YH,4.0,90.0,DIV,1)
HIGH=.134
STR=.5
XH=.5
CALL SYMROI(STR,XH,HIGH,NVAR,0,0,5)
CALL VINSER(-.5,.0,.0,0,0,0,0,0,0,3)
YH=STR+.25
DO 110 I=1,NVAR
110 IF(VAR(I))105,111,105
105 CONTINUE
CALL SYMROI(YH,XX,HIGH,IV(1),90.0,4)
CALL VINSER(-.0,-.0,-.0,0,VAR(1),90.0,4)
YH=YH+.25
110 CONTINUE
111 CONTINUE
   CALL SYMBOL(YH, XH, HIGH, IDSH, 90.0, 4)
   CALL SYMBOL(-0.0, -0.0, -0.0, ITAB, 90.0, 12)
   YH = YH + .125
   CALL PLOT(YH, XH, 3)
   CALL PLOT(YH, XH+.3125,?)
   YH = YH + .125
   CALL SYMBOL(YH, XH+.5, HIGH, IFIT, 90.0, P)
   DO 120 I = 1, N
     YT(I) = -YT(I)
     YF(I) = -YF(I)
120 CONTINUE
   CALL LINE(RMIN, DL, 0, -1, 1, 0)
   RMIN = -(RMIN - DL*6.0)
   CALL LINE(RMIN, DL, 0, 1, 1, 0)
   CALL LINE(YT, X, N, 1, 0, 0, 0, 0, 0)
   CALL LINE(YF, X, N, 1, 0, 0, 0, 0, 0)
   CALL PLOT(9.5, -1.0, -23)
   RETURN
END
SUBROUTINE AUXFCN (X, F, K)
DIMENSION T(500), Y(500), X(1)
COMMON: T, Y, NPTS, FCN1(500), ISW1
A = X(1)
R = X(2)
G = X(3)
P = X(4)
A1 = X(5)
A2 = X(6)
D = X(7)
A3 = X(8)
F = 0.
DO 80 I = 1, NPTS
AT = -A * T(I)
BT = -B * T(I)
GT = -G * T(I)
DT = -D * T(I)
FAT = EXP(AT)
EGT = EXP(GT)
EDT = EXP(DT)
T1 = BT - P
CSN = COS(T1)
SSN = SIN(T1)
T2 = EAT * CSN
T3 = A2 * T2
T4 = A1 * FAT
T5 = A2 * FAT * SSN
T6 = A3 * EDT
FCN = T4 + T3 + T6 - Y(I)
IF(ISW1)2, 4, 2
2 FCN1(I) = T4 + T3 + T6
GO TO 90
4 CONTINUE
GO TO (10, 20, 30, 40, 50, 60, 62, 64), K
10 CONTINUE
PAR = -T(I) + T3
GO TO 70
20 CONTINUE
PAR = -T(I) + T5
GO TO 70
30 CONTINUE
PAR = -T(I) + T4
GO TO 70
40 CONTINUE
PAR = T5
GO TO 70
50 CONTINUE
PAR = FAT
GO TO 70
60 CONTINUE
PAR = T2
GO TO 70
62 CONTINUE
PAR = -T(I) * T5
GO TO 70
161 CALL AUXFCN(X,FPLUS,K)
PART(ITEMP)=(FPLUS-F)/H
ITEMP=HOLD
IF(ABS(PART(ITEMP)))* 305, 310, 305
305 IF(ABS(F/PART(ITEMP))-1.0E+20) 200, 200, 310
310 ITALLY=ITALYK+1
200 CONTINUE
IF(ITALY-N+K)202, 202, 311
311 CONTINUE
FACTOR=FACTOR*10.0
IF(FACCTOR-.15) 203, 312, 312
202 IF(K-N) 203, 312, 312
312 CONTINUE
IF(ABS(PART(ITEMP)))313, 775, 313
313 CONTINUE
COE(K,N+1)=0.0
KMAX=ITEMP
GO TO 500
C
C FIND PARTIAL DERIVATIVE OF LARGEST ABSOLUTE VALUE.
C
203 KMAX=LOOKUP(K,K)
DERMAX=ABS(PART(KMAX))
KPLUS=K+1
DO 210 I=KPLUS,N
JSUB=LOOKUP(K,I)
TEST=ABS(PART(JSUB))
IF(TEST-DERMAX) 209, 314, 314
314 CONTINUE
DERMAX=TEST
LOOKUP(KPLUS,I)=KMAX
KMAX=JSUB
GO TO 210
209 LOOKUP(KPLUS,I)=JSUB
210 CONTINUE
IF(ABS(PART(KMAX)))315, 775, 315
315 CONTINUE
C
C SET UP COEFFICIENTS FOR KTH ROW OF TRIANGULAR LINEAR SYSTEM USED
C TO BACK-SOLVE FOR THE FIRST K X(I) VALUES.
C
1SUb(K)=KMAX
COE(K,N+1)=0.
DO 220 J=KPLUS,N
JSUB=LOOKUP(KPLUS,J)
COE(K,JSUB)=-PART(JSUB)/PART(KMAX)
COE(K,N+1)=COE(K,N+1)+PART(JSUB)*X(JSUB)
220 CONTINUE
500 COE(K,N+1)=(COE(K,N+1)-F)/PART(KMAX)+X(KMAX)
C
C BACK SUBSTITUTE TO OBTAIN NEXT APPROXIMATION TO X.
C
X(KMAX)=COE(N,N+1)
IF(N-1)316, 610, 316
316 CONTINUE
   CALL RACK(N-1,N,X,ISUB,CONF,LOOKUP)
610 IF(M-1)650,650,675

C TEST FOR CONVERGENCE.
C
625 DO 630 I=1,N
   IF(ABS(TEMP(I)-X(I))/X(I)-AFCON) 630,630,649
630 CONTINUE
   JTEST=JTEST+1
   IF(JTEST-3) 650,725,725
649 JTEST=1
650 DO 660 I=1,N
660 TEMP(I)=X(I)
700 CONTINUE
725 IF(IPRINT-1)800,3,800
317 CONTINUE
   DO 750 K=1,N
   CALL AUXFCN(X,PART(K),K)
750 CONTINUE
   WRITE(KOUT,751) (PART(K),K=1,N)
751 FORMAT(/, 'FUNCTION VALUES EVALUATED AT FINAL APPROXIMATION FOLLOW
   ', 120X,' in Table ',6E20.9)
GO TO 990
775 WRITE(KOUT,776)
776 FORMAT(/20X, 'MODIFIED JACOBIAN IS SINGULAR. TRY A DIFFERENT INITIAL APPROXIMATION. '
   )
900 RETURN
END
SUBROUTINE BACK (KMIN, N, X, ISUB, COE, LOOKUP)

C THIS SUBROUTINE BACK-SOLVES THE FIRST KMIN ROWS OF A TRIANGULARIZE
C LINEAR SYSTEM FOR IMPROVED X VALUES IN TERMS PREVIOUS ONES.

DIMENSION X(30), ISUB(30), COE(30, 31), LOOKUP(30, 30)

DO 200 KK = 1, KMIN
   KM = KMIN - KK + 2
   KMAX = ISUB(KM - 1)
   X(KMAX) = 0.0
   DO 100 J = KM, N
      JSUB = LOOKUP(KM, J)
      X(KMAX) = X(KMAX) + COE(KM - 1, JSUB) * X(JSUB)
   100 CONTINUE
   X(KMAX) = X(KMAX) + COE(KM - 1, N + 1)

200 CONTINUE

RETURN
END

/*
*/
/}
THIS SUBROUTINE
SOLVES THE FIRST
EQUATION OF A
TRIANGULAR
SYSTEM OF
EQUATIONS.

** Note 01
• • • • • • • • •
• OPEN ON LOOP
• J = 1, J = N
• • • • • • • • •

** **
• • • • • • • • •
• CH = KEEN + B
• 2
• MN = SUBRN - T
• TERM1 = 0.0

** Note 02
• • • • • • • • •
• OPEN ON LOOP
• 100 J = MN, H
• • • • • • • • •

** **
• • • • • • • • •
• JUMP (F specification)
• TERM1 = TERM1 - 
• IF (TERM1 < 0.0)
• GOTO 1

** **
• • • • • • • • •
• END ON LOOP
• H = MN
• • • • • • • • •

** **
• • • • • • • • •
• END ON LOOP
• H = MN
• • • • • • • • •

** **
• • • • • • • • •
• END ON LOOP
• H = MN
• • • • • • • • •

** **
• • • • • • • • •
• END ON LOOP
• H = MN
• • • • • • • • •
JDA EC790
EXEC FORTRAN
DIMENSION Y(113), TL(12)
DIMENSION SAVE(300,16), ILOC(16), BUFF(3000)
REAL IX, IY, I7, IXY, Ixz, IY7

COMMON H, HMAX, HMIN, DH, FCT, TL, NGS, N, IS1, NPNT
COMMON TLM, RHOL?, RHOL?, RHOL4, RHOL5, WM, FTA, FTAMl, ISW2
COMMON XQQ, XR, XR?, XR?, XUN, XV, XWO, XVU, XVV,
XWW, XDONR, XDONS, XDOSR, XV, XWRF, XDORRF, XDORSF
COMMON YPD, YPD, YPAP, YPO, YQR, YVO, YVP,
1 YWR, YP, YP, YARP, YVAR, YSTR, YV, YVAV,
2 YVW, YDR, YOF, YVF, YVAVF, YDRF
COMMON ZON, ZPP, ZRR, ZRP, ZWN, ZVP, ZVP, 70,
1 ZAONS, ZWAQ, ZSTR, 7W, ZWAW, 7AW, ZWW, 7VV,
2 ZNS, ZDR, ZOF, ZW, ZWAWE, ZDSE
COMMON AKPO, AKRO, AKQ, AKPO, AKPO, AKP, AKR, AKVD,
1 AKVO, AKWP, AKWP, AKST?, AKV, AKVAV, AKVW, AKOR,
2 AKSTRE
COMMON AMPO, AMPO, AMPO, AMPO, AMPO, AMPO, AMPO,
1 AMQ, AMQNS, AMAOW, AMST, AMO, AMAOW, AMAW, AMWW,
2 AMV, AMO, AMO, AMO, AMW, AMW, AMW, AMW
COMMON ANPO, ANPO, ANPO, ANPO, ANPO, ANPO, ANPO,
1 ANPO, ANP, ANP, ANP, ANP, ANP, ANP, ANP, ANP,
2 ANVQ, ANW, ANW, ANW, ANW, ANW, ANW, ANW
COMMON IX, IY, I7, IXY, IXZ, IY7
COMMON CW, CR, UC, XA, YR, ZA
COMMON DR, DS, DR, DR, AL, AN
COMMON DFW, DFW, DFW, DFW, DFW, DFW, DFW
COMMON AP1, A27, A27, A31, A32, A32
COMMON XR, YG, ZG
COMMON ILQ, IPlot, IRUN, IOPEN, NPLT, INPT
COMMON Y
COMMON TIME, R1, DELTM, SWMAX, RP, DELTM1, NS, RS, NSF, NSF, ICY, NS,
1 INTSW
CALL INPUT
**COMPUTE RH0 * L CONSTANTS**

RH0H = RH0 * .5
RH0L1 = RH0H * A1 * A1
RH0L2 = RH0L1 * A1
RH0L3 = RH0L2 * A1
RH0L4 = RH0L3 * A1
RH0L5 = RH0L4 * A1

**WRITE OUT HYDRODYNAMIC COEFFICIENTS**

CALL WRITE
T = 1
XDRNA = RH0L2 * XDRNA*T
XDSNS = RH0L2 * XDSNS*T
XDBDA = RH0L2 * XDBDA*T
A11 = RH0L2 * A11 * T
A12 = RH0L2 * A12 * T
A13 = RH0L2 * A13 * T
A21 = RH0L2 * A21 * T
A22 = RH0L2 * A22 * T
A23 = RH0L2 * A23 * T
A31 = RH0L2 * A31 * T
A32 = RH0L2 * A32 * T
A33 = RH0L2 * A33 * T
XUN = RH0L2 * XUN
YR = RH0L2 * YR
YPD = RH0L2 * YPD * T
YP0 = RH0L2 * YPO * T
VY = RH0L2 * VY * T
VVAV = RH0L2 * VVAV * T
VDR = RH0L2 * VDR * T
YDV = RH0L2 * YDV * T
ZQ = RH0L2 * ZQ
ZDN = RH0L2 * ZDN * T
ZRR = RH0L2 * ZRR * T
ZVR = RH0L2 * ZVR * T
ZSTQ = RH0L2 * ZSTQ * T
ZV = RH0L2 * ZV * T
ZVAV = RH0L2 * ZVAV * T
ZVY = RH0L2 * ZVY * T
ZV0 = RH0L2 * ZV0 * T
ZV0 = RH0L2 * ZV0 * T
IT = 1
AKMD = RH0L2 * AKMD * T
AKP = RH0L2 * AKP * T
AKV = RH0L2 * AKV * T
AKV = RH0L2 * AKV * T
AKVAV = RH0L2 * AKVAV * T
AKP0 = T
T = 1
NAVTRADEVCO 68-C-0050-2

AMRP = (I2 - IX + RHO45 * AMRP) * T
AMRR = RHO45 * AMRR + T
AMWD = RHO45 * AMWD + T
AMVR = RHO45 * AMVR + T
AMQ = RHO45 * AMQ + T
AMAWQ = RHO45 * AMAWQ + T
AMSTR = RHO45 * AMSTR + T
AMW = RHO45 * AMW + T
AMWAW = RHO45 * AMWAW + T
AMVV = RHO45 * AMVV + T
AMDS = RHO45 * AMDS + T
AMOR = RHO45 * AMOR + T
AMQP = T
T = I7-RHO45*ANPD
T = 1/T
ANPO = (I5-IY+RHO45*ANPO)*T
ANPD = RHO45 * ANPD + T
ANVD = RHO45 * ANVD + T
ANP = RHO45 * ANP + T
ANR = RHO45 * ANR + T
ANV = RHO45 * ANV + T
ANVAV = RHO45 * ANVAV + T
ANOR = RHO45 * ANOR + T
ANRD = T
ZB = CT * ZP
CALL WRITE
IPCH = ?
WRITE(IPCH,RO) XORDP, YORDP, XORDR, A11, A12, A13
WRITE(IPCH,RO) A21, A22, A23, A31, A32, A33
WRITE(IPCH,RO) XUN, YR, YDN, YPN, YP, YV
WRITE(IPCH,RO) YVAV, YDH, YDV, ZO, ZOD, ZOO
WRITE(IPCH,RO) ZOR, ZOR, AKPD, AKPD, AKV, AKV
WRITE(IPCH,PO) AKV, AKPD, AMPD, AMR, AMWN, AMVD
WRITE(IPCH,PO) AMO, AMAO, AMSTR, AMW, AMWAW, AMVV
WRITE(IPCH,PO) AMS, AMOR, AMRD, AMDP, AMPO, AMDP, AMVD
WRITE(IPCH,PO) AMNP, AMNP, AMNV, AMVAV, AMOR, AMRD
WRITE(IPCH,RO) ORMAY, FTAHI, FTAM, CW, CR, YG
WRITE(IPCH,RO) YG, AL, AM, OR, DS, OR
WRITE(IPCH,RO) PA, UC, TIMM, PT, DELTMA, SWAY
WRITE(IPCH,RO) P2, DELTMI, NSF, NSF, Y(I1), I=1,12
90 FORMAT(14F13.5)
GO TO 45
END

317
SUBROUTINE WRITE
DIMENSION Y(13), TL(12)
DIMENSION SAVE(300,16), TL0C(14), RUFF(3000)
REAL IX, IY, IT, IXY, IXY, IY
COMMON H, HMAX, HMIN, DH, FCT, TL, NGS, N, IS1, NOST
COMMON TLIM, RHOL2, RHOL3, RHOL4, RHOL5, WM, ETA, ETA1, ETA2, ETA3, ETA4,
COMMON XQQ, XPR, XPP, XQP, XVR, XH0, XVU, XVV,
NUMBER XWW, XXDP, XOSDS, XODAD, XVVF, XVFE, XDFRE, XOCNSF
COMMON YPP, YDP, YQAP, YPQ, YOR, YVD, YVO, YWP,
NUMBER YVP, YP, YAP, YAPP, YVAR, YSTR, YV, YVAV,
COMMON YVW, YOR, YRE, YVF, YVAVE, YQF
COMMON ZOF, ZPP, ZPR, ZRP, ZWD, ZPR, ZP, ZO,
NUMBER ZAOQ, ZSTR, ZW, ZVAM, ZAW, ZWH, ZWH, ZV,
COMMON ZOM, ZO, ZOMF, ZOF, ZWF, ZAWAF, ZOSF
COMMON AKRQ, AKRD, AKQR, AKPQ, AKPQ, AKP, AKP, AKVP,
NUMBER AKQ, AKVP, AKWP, AKSTR, AKV, AKVAV, AKVW, AKQ
COMMON AMOQ, AMPP, AMPR, AMQRF, AMQAOQ, AMQ, AMQ0, AMQ0, AMQ0, AMQ0,
NUMBER AMQ0, AMQ0, AMQ0, AMQ0, AMQ0, AMQ0, AMQ0, AMQ0, AMQ0
COMMON ANPO, ANPD, ANPR, ANPR, ANRAZ, ANVR, ANVR, ANVR, ANVR,
NUMBER ANVR, ANVR, ANVR, ANVR, ANVR, ANVR, ANVR, ANVR, ANVR
COMMON IX, IY, IT, IXY, IXY, IY
COMMON GH, CP, HC, VR, VR, 71
COMMON HD, DC, DR, RHQ, AL, AM
COMMON ORMAY, ETAH, ETAH, AL, AL, AL
COMMON AQ2, AQ2, AQ2, AQ2, AQ2, AQ2
COMMON XY, YC, YC
COMMON HP, IDP, TPW, TOPN, NPLT, NORT
COMMON V
COMMON TINE, QT, DELMA, SWMAX, QQ, DELTIME, OSE, OSE, OSE, OSE
COMMON INT

INIT = 1
\* FORMATTING, E0X, DATA PREPARATION*)
NAVTRADEVGEN 68-G-0050-2

WRITE(T(IOUT,24)
WRITE(T(IOUT,11)
WRITE(T(IOUT,11)
WRITE(T(IOUT,11)
WRITE(T(IOUT,11)
10X, YVE, 4x, F12.5, 10X, ?WF, 4X, F12.5, 21X, WOF, 4X, F12.5,
7V NVE, 4X, F12.5/1H, 10X, YVAVE, 7X, F12.5, 7WAVE, 7X, F12.5,
W 21X, WF, 4X, F12.5, YVAVE, 7X, F12.5/1H, 19X, YVEF, 3X,
9F12.5, 7DSF, 3X, F12.5, 2X, YVAVE, 2X, F12.5, YVEF, 3X,
A E12.5/1H, A5X, MDSF, 3X, F12.5/1H
WRITE(OUT, A) IX, IY, IZ, IXY, IY7, CX, CR, IC, IX, YA,
1 ZR, DR, NS, DA, LH, AO, AL, AU
4 FORMAT(1H, 'IX', '5X', 'F12.5', 'IY', '5X', 'F12.5', 'I7', '5X', 'F12.5',
1 IXY, '4X', 'F12.5', 'IX7', '4X', 'F12.5', 'IX7', '4X', 'F12.5/1H',
2IV, '6X', 'F12.5', 'IX', '6X', 'F12.5', 'I7', '6X', 'F12.5/1H',
3Y, '6X', 'F12.5', 'IX', '6X', 'F12.5',
4 OR, '5X', 'F12.5', 'IX', '5X', 'F12.5',
5 Y, '6X', 'F12.5',
WRITE(OUT, A) A11, A21, A31, NMAX, ETAHI, ETALN, A12, A22, A32,
1 XG, YG, ZG, A13, A23, A33
7, H, INTSW, TIME, PI, DELTMA, SUMAX, N2, DELTMI, NSE, NRE, ICYC, NS
4 FORMAT(1H, 'A11', '4X', 'F12.5', 'A21', '4X', 'F12.5', 'A31', '4X', 'F12.5',
1 'A41', '4X', 'F12.5', 'A51', '4X', 'F12.5', 'A61', '4X', 'F12.5/1H',
2H, 'A2', '4X', 'F12.5', 'A3', '4X', 'F12.5', 'A4', '4X', 'F12.5',
3XG, '5X', 'F12.5', 'YG', '5X', 'F12.5', 'ZG', '5X', 'F12.5/1H',
4 A11', '4X', 'F12.5', 'A21', '4X', 'F12.5', 'A31', '4X', 'F12.5',
5 H, '6X', 'F12.5',
WRITE(OUT, A) INTSW, 'X', '17/
41H, '1WF', '3X', 'F12.5', 'R1', '5X', 'F12.5', 'DELTMA', '1X', 'F12.5', SUMAX',
72X, 'F12.5', 'NS', '5X', 'F12.5', 'DELTMI', '1X', 'F12.5',
91H, 'DSF', '4X', 'F12.5', 'NRF', '4X', 'F12.5', ICYC, '3X', '12, 10X,
1 NS, 'X', '17/
WRITE(OUT, 5) RUN
5 FORMAT(1H, 'RUN NO ', '15/)
RETURN
END
SUBROUTINE INPUT
DIMENSION Y(131), TL(12), ILNC(16), YHOLD(13), CMN(19)
EQUIVALENCE (CMN(1), Y)
REAL IX, IY, IT, IXV, IXV, TVT

COMMON H, HMAX, HMIN, DO, FCT, TL, NGS, N, ISI, NPNT
COMMON TLIM, RHOL3, RHOL4, RHOL5, WMR, ETA, ETAW1, I5W2

COMMON XQO, XPR, XDP, XUN, XVR, XWO, XIJL, XVV,
  XWW, XDPR, XNSDS, XRDB, XVVF, XWWW, XRDRF, XNSDF

COMMON YPR, YPD, YPA, YPR, YQ, YSO, YVQ, YWD,
  YVR, YP, YAR, YVAR, YSTR, YV, YVAV,
2 YVW, YOR, YPE, YVF, YVAE, YDR

COMMON Z21, Z22, Z23, Z24, Z25, Z26, Z27, Z28, Z29, Z30,
1 ZAOQ, ZAW, ZSTR, ZW, ZWAN, ZAW, ZWW, ZVV,
2 ZDS, ZDR, ZDE, ZWF, ZWAF, ZDSF

COMMON AKPQ, AKPO, AKPQ, AKPQ, AKKAP, AKP, AKP, AKVP,
  AKQ, AKWP, AKW, AKSTR, AKV, AKVAV, AKVW, AKVP,
2 AKSTRF

COMMON AMON, AMPP, AMRF, AMRP, AMPO, AMW, AMVQ, AMVP,
  AMQ, AMQDS, AMAOD, AMSTR, AMW, AMWAW, AMAW, AMWW,
2 AMV, AMOP, AMDR, AMQF, AMWF, AMWAF, AMDSF

COMMON ANPH, ANPD, ANPQ, ANQR, ANAR, ANVQ, ANVP, ANWP,
  ANVQ, ANP, ANP, ANAR, ANVQ, ANSTR, ANV, ANVAV,
2 ANWM, ANQP, ANPF, ANVF, ANVAVF, ANDR

COMMON IX, IY, IT, IXV, IXV, TVT

COMMON CW, CP, UC, XB, YR, ZR

COMMON DP, DS, NO, DPM, AL, AN

COMMON ORWAX, ETA1, ETA2, A11, A12, A13

COMMON A21, A22, A23, A21, A32, A23

COMMON XG, YG, ZG

COMMON ILNC, IPLOT, IPUN, IOPEN, NPLT, IORT

COMMON Y

COMMON TIME, R1, DELTWA, SWAY, OP, DELTM1, OSE, OSE, ICYC, NS,
1 NTSM

IN = 1
READ(3, 51) NIS, NPNT, IPLOT, IPUN, NPLT, IORT, ICYC, VS, ITRN
IF(IRUN < 20, 30, 70
321
CALL EXIT
CONTINUE
READ(IN,50) (I,OC(I), I = 1, 16)
FORMAT(1615)
READ(IN,100) TN,HO,HN,HMAX,MIN,ECT,TLIH
H = HO
FORMAT(9F10.5)
READ(IN,100)(TI(I), I = 1, 17)
READ(IN,100) Y(I), I = 1, 17
Y(13) = TO
READ(IN,100) X00, Y00, X0Q, Y0Q, XY0, Y0, XUV, XUV,
X0W, X0W, X0S, X0S, X0W, X0W, X0Y, X0Y
READ(IN,100) Y0O, Y0O, Y0D, Y0Q, Y0Q, Y0, Y0, Y0,
Y0, Y0, Y0, Y0, Y0, Y0, Y0, Y0, Y0
READ(IN,100) Y00, Y00, Y0D, Y0D, Y0, Y0, Y0, Y0,
Y0, Y0, Y0, Y0, Y0, Y0, Y0, Y0, Y0
READ(IN,100) AKP0, AKP0, AKP0, AKP0, AKP0, AKP0, AKP0,
AKP0, AKP0, AKW, AKW, AKP0, AKP0,
READ(IN,100) AKP0, AKP0, AKP0, AKP0, AKP0, AKP0, AKP0,
AKP0, AKP0, AKP0, AKP0, AKP0, AKP0,
READ(IN,100) AKP0, AKP0, AKP0, AKP0, AKP0, AKP0, AKP0,
AKP0, AKP0, AKP0, AKP0, AKP0
READ(IN,100) IY, IV, IT, IVY, I0Z, IY7
READ(IN,100) CW, CR, CU, CR, VR, ZH
READ(IN,100) OP, OR, OR, OR, OR, OR, VR
READ(IN,100) P, P, P, P, P, P, P
READ(IN,100) X0, Y0, Z0
READ(IN,100) XUV, UV1, UV2, UV3, UV4, UV5, UV6
RETURN
REAL ICYC
DIMENSION Y(13), F(13), R(13), X(13)
COMMON XNDR, XNDRS, XMM, A12, A13, A22, A23, A31, A32, A33, X1P
COMMON YR, YRD, YPP, YP, YV, YVAV, YND, YUN
COMMON ZQ, ZQD, ZPP, ZPD, ZRP, ZW, ZWAW, ZVV, ZOS, ZPG,
COMMON AMDR, AKP, AKPD, AKV, AKVY, AKVZ, AKVZ1
COMMON AHRP, AMRP, AMDR, AMRP, AMQ, AMQV, AMW, AMWAW,
COMMON AMRP, AMRP, AMRP, AMRP, AMRP, AMRP, AMRPA, AMRPA,
COMMON AMQV, ETAQ, ETAQ, CW, CA, XG, AI, AM, NP, OS, OR,
COMMON ICYC, ICYC, ICYC, ICYC, ICYC, ICYC, ICYC, ICYC
COMMON Y, F, F1, ICYC, NS, H, HH, 1SW,
EQUIVALENCE (X(13), YRPP)
IN = 1
ICYC = 1.
20 READ(IUN,50) IRUN, NDNT, NS
50 FORMAT(1615)
30 CALL EXIT
40 CONTINUE
READ(IUN,60) TO, H, TL:
60 FORMAT(8E10.5)
70 FORMAT(9E13.6)
90 FORMAT(9E15.6)
50 FORMAT(1615)
100 CALL UNDRT
110 IF(IN - 1) THEN
120 WRITE(IOUT,77)
130 WRITE(IOUT,78) IRUN, NS
140 WRITE(IOUT,79)
77 FORMAT(1615,25X, "REAL VARIABLES: SIMULATION: ")
78 FORMAT(1615,4X, ANDOG, NO, TD, EX, H, IN, 14/)
79 FORMAT(1615, 4X, H, IN, TD, EX, H, IN, 14/)
100 FORMAT(1615, 4X, H, IN, TD, EX, H, IN, 14/)
110 FORMAT(1615, 4X, H, IN, TD, EX, H, IN, 14/)
120 WRITE(IOUT,20) (Y(I), I=1, 13)
95 FORMAT(1615)
96 FORMAT(1615)
97 FORMAT(1615)
98 FORMAT(1615)
99 FORMAT(1615)
100 FORMAT(1615)
101 CALL UNDRT
102 IF(IN - 1) THEN
103 WRITE(IOUT,60) (A(I), I=1, 13)
104 WRITE(IOUT,85)
32C
110 ICNT = ICNT + 1
IF (Y(1M-Y(13)) < 20, 20, 90
END
SUBROUTINE UPDATE
REAL ICY,
DIMENSION Y(12), F(12), E(12)
COMMON X00,JX, X01X, X01A, A1, A2, A3, A4, A5, A6, A7, A8, A9,
1 A10, A11, X0
COMMON YR, YP, YR, YP, VV, VV, YR, YP
COMMON ZQ, ZQ, ZQ, ZQ, ZQ, ZQ, ZQ, ZQ, ZQ,
1 ZQ
COMMON AKR, AKR, AKR, AKR, AKR, AKR
COMMON AMP, AMP, AMP, AMP, AMP, AMP, AMP, AMP, AMP, AMP, AMP
COMMON AMP, AMP, AMP, AMP, AMP, AMP, AMP, AMP, AMP, AMP, AMP
COMMON AMQ, AMQ, AMQ, AMQ, AMQ, AMQ, AMQ, AMQ, AMQ, AMQ, AMQ
COMMON AMR, AMR, AMR, AMR, AMR, AMR, AMR, AMR, AMR, AMR, AMR
COMMON AMQ, AMQ, AMQ, AMQ, AMQ, AMQ, AMQ, AMQ, AMQ, AMQ, AMQ
COMMON AMR, AMR, AMR, AMR, AMR, AMR, AMR, AMR, AMR, AMR, AMR
COMMON Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, Y
EQUIVALENCE (Y(1),Y(1),Y(1),Y(1),Y(1),Y(1),Y(1),Y(1),Y(1),Y(1),Y(1))

CALL TO CONTROL
CALL CONTRITHETA

COMPUTE QUANTITIES TO BE USED MORE THAN ONCE

IIl = Y * Y
V2 = V * V
PR = R * R
POTTVW = SQRT(V2 + W2)
VRTVW = V * POTTVW
WRTVW = W * POTTVW
VR = V * R
WP = W * P
UP = R * P
CWR = 2*GSW - 79
WWR = G - CR

SET PROPELLER THROTTLE CONSTANTS

1 IF(I) 10,20,70
2) ETA = I * I
3 IF(I,4) = (0,0,0)
7 A1 = 111
A2 = 111
A3 = 111
GO TO 40
2) IF(I,4) = 70,40,60
4) A1 = 111
A2 = 111
A3 = 111
GO TO 40
5) A1 = 111
A2 = 111
A3 = 111
6) CONTINUE
C COMPUTE TRIG FUNCTIONS

SPHI = SIN(PHI)
CPHI = COS(PHI)
STTA = SIN(THETA)
CTTA = COS(THETA)
SPSI = SIN(PSI)
CPSI = COS(PSI)
TRIG1 = CTTA*SPHI
TRIG2 = CTTA*CPSI
TRIG3 = SPHI*STTA
TRIG4 = CPHI*STTA
TRIG5 = U*CTTA

C COMPUTE VN FROM AXIAL FORCE FON

F(1) = (AM*(VW-WN)+XG*(Q+P)+ZG*(RP+F(51)))*11*(XDRNR+DR*NR
1 +XOSNS*SNS*XNRDR+NR*NR+AA)*UC*(A2U+A3UIC)-WMR*STTA)/(AM-XUD)

C COMPUTE VN FROM LATERAL FORCE FON

F(2) = (U*{(V0-A*+)*R+YD*+VV*V})-AM*{(ZG*(Q-P-F(41)*XG*7P+F(6))}
1 +YDN*F(4)+YDN*F(4)+YVAV*VRTVW+YDR*11*DR+WMR*TRIG1+AM*WP
2 /(AM-VYN)

C COMPUTE VN FROM NORMAL FORCE FON

F(3) = (U*{(7Q-A*)+7W*W}+AM*{(7G*(P*+Q)+XG*7R-F(51))}+1
1 +7Q*F(51)+7PR*PR+7VR*VR+7STR*STR+7WAW*WRTVW+7VV*V7)*17*7NS*7S+
2 7NR*NR+WMR*TRIG2)/(AM-VWN)

C COMPUTE DP FROM ROLLING MOMENT FON

F(4) = (AM*7G*(F(7)-WP+1*8)-GWR*TRIG1)*AKP+AKP*F(4)+1*(AKP
1 +AKV*V)+AKVAV*VRTVW+AKVID*F(2)

C COMPUTE DP FROM PITCHING MOMENT FON

F(5) = AMR*PR*AP-AMQ0*AP*(ZG*(F(1)*-VR+WQ*-XG*(F(3)-11*VP)))+ANRP*UC
1 +AMW*F(3)+AMV*V+U*(AMQ0*AMWQ)+17*(AMSTR+ANDR+ANDP+ANDS+NS)
2 7AMW*Q*ROOTVW+AMW*A+AMVAV*VRTVW+AMVAV*AMVAV+?7AMQ0*{(XG*G*TRIG2)+GWR*STTA}

C COMPUTE DP FROM YAWING MOMENT FON

F(6) = ANDP*Q+ANDAP=AMQ0*(XG*(F(4)-WP+10*F)))+ANPD*F(4)+ANV*F(4)+
1 11*(ANDP+ANDP+ANDP+ANDP)+ANP*AMWAV+VTRTW+ANDR*ANDP+ANDR*XG*TRIG1

C COMPUTE KINEMATICS - THETA DOT, PSI DOT, PHI DOT

F(7) = 2*(CPHI*R*SPHI
F(8) = (SPHI*R+CPHI)/CTTA
F(9) = *F(9)*CTTA

331
C COMPUTE X DOT, Y DOT, Z DOT
C
F(10)=TRIG5*PSI+V*(TRIG3*CPSI-CPHI*SPSI)+
1*W*(TRIG4*CPSI+SPHI*SPSI)
F(11)=TRIG5*PSI+V*(TRIG3*SPSI+CPHI*CPSI)+
1*W*(TRIG4*SPSI-SPHI*CPSI)
F(12)=-U*STTA+V*TRIG1+W*TRIG2
C
C INTEGRATE AND REPLACE OLD DERIVATIVES WITH NEW
C
DO 80 I = 1,12
Y(I) = Y(I) + H*F(I)*F(I-1)
F(I) = F(I-1)
80 CONTINUE
C UPDATE TIME
Y(13) = Y(13) + H
RETURN
END
SUBROUTINE CONTR(THETA)
REAL ICYC
DIMENSION Y(113), F(112), F1(112)
COMMON XHIN, XHOS, XHOP, A11, A12, A13, A21, A22, A27, A77, A71, 
1 A72, A23, XIN
COMMON YR, YPD, YPD, YV, YVAV, YDR, YVW
COMMON ZO, ZOQ, ZR8, ZVR, ZSTR, ZW, ZHAAW, ZVW, ZDS, ZDW,
1 ZWO
COMMON AKRD, AKP, AKV, AKV, AKV, AKPO
COMMON AMDR, AMDS, AMDQ, ANM, AMNO, ANMO, ANMO, ANOP, ANOP, ANOP
COMMON DRYAV, ETAH1, ETAH2, CW, CR, XS, ZS, AL, NV, OR, NS, OA,
1 ZA, ZS, TIFE, P1, DELTA, CWMAX, R2, DELTA, DSP, DSP
COMMON Y, F, F1, ICYC, NS, H, HH, ISW
IF(NS).EQ.15,16,"RETURN"
GO TO 1101,1001,1002,1004,1005,1006,1007,NS
C CONTROL DS
C 1001 IF(IS/2).EQ.21,20,21
20 N1 = 2
30 NN2 = 1
40 MC2 = ((TIME*ICYC)/H) + .5
50 NS2 = (ARS*ICYC)/(ICYC - DELTMA) + ICYC/ARS*ICYC - H + .5
60 NSC = (ARS - ICYC)/(ICYC - DELTMA) + ICYC/ARS - H + .5
GO TO 11
70 GO TO 11,12,13,14,15,16,17
C CYCLES TO START
80 NSR = ICYC
t H(NN) = 2 " H(NN) = 2 " H(NN) = 11,11,7
C DS HORN
C 90 N1 = 1
100 N2 = 0
110 NN1 = NN0 + 1
120 NS = NS + H/ICYC
130 IF(NS- NC+1) = 11,00,0
C DS LEVEL
C 140 N1 = 4
GO TO 11
150 IF(ARS(THETA) - CWMAX) = 11,0,0
C DS UP
C 160 N1 = 5
NSC = 1
333
5 NN5 = NN5 + 1
DS = DS + H*27/ICYC
IF (NN5 = VC5) 11, 10, 10
C
10 N1 = 6
C
11 IF (NS = 213, 1003, 1003)
13 CONTINUE
GO TO 2000
C
CONTROL OR AUTOPILOT
C
1003 IF (ISW2) 301, 300, 301
300 ZC = Y(12)
SDOT1 = 0.
DDR = ABS(NDMAX)
NC10 = (TIME * ICYC / H) + .5
NC6 = .95 * DDR * ICYC / 1.0072641 + .5
NC7 = .08 * DDR * ICYC / 1.0032641 + .5
NC8 = .04 * DDR * ICYC / 1.0006841 + .5
NC9 = .03 * DDR * ICYC / 1.0010241 + .5
NN10 = 1
N2 = 2
IF(NDMAX) 313, 314, 314
313 R6 = -.97726
R7 = -.11336
R8 = -.006
R9 = -.001044
GO TO 350
314 P6 = -.97726
P7 = -.11336
P8 = -.006
P9 = -.001044
GO TO 350
301 GO TO 1300, 102, 292, 304, 305, 300, 1501, NN
300 NN10 = NN10 + 1
IF (NN10 = VC10) 1250, 160, 319
C
FP3M 0 IF (.95 OF DMAX
C
309 N2 = 0
NN4 = 0
313 NN6 = NN4 + 1
NR = .95 + H*27/ICYC
IF (NN4 = NN6) 250, 299, 312
C
FP3M .95 TO .32 OF DMAX
C
303 N2 = 6
NN7 = 0
GO TO 250
274 NN7 = NN7 + 1
DR = DR + H*RR/ICYC  
IF (NN7 - NC7) 350, 310, 310

C FROM .93 TO .97 OF NMAX
C
310 N2 = 5  
NN8 = 0  
GO TO 350
305 NN8 = NN8 + 1  
DR = DR + H*RR/ICYC  
IF (NN8 - NS8) 350, 311, 311
C FROM .97 TO 1. OF NMAX
C
311 N2 = 6  
NN9 = 7  
GO TO 350
304 NN9 = NN9 + 1  
DR = DR + H*RR/ICYC  
IF (NN9 - NC9) 350, 312, 312
C
LEVEL, NMAX
C
350 IF((NS-7)2000,2000,352
C
AUTOPILOT
C
352 DSC=.729*(7C-Y(12))+.55Y(7)+.012*(Y(11)*SIN(Y(7))-Y(13)*COS(Y(7)))  
1+7.9Y(9)
193 IF (DSC) 110, 107, 107
107 IF (DSC -.436) 101, 109, 103
110 IF (DSC + .436) 109, 101, 101
109 DSC = .436  
GO TO 101
109 DSC = -.436
121 SNDT = 3 * (DSC - NS)  
NS = NS + .5 * H/ICYC * (1. * SNDT - SNDT1)  
SNDT1 = SNDT  
DR = -NS  
351 CONTINUE  
GO TO 3000
C
CONTROL NS (IMPHASE), LONGITUDINAL
C
1074 IF (115421) 1401, 400, 471
400 IF (ICYC-1) 1411, 411, 412
411 N4=0
NTST=1
NMND=0  
GO TO 401
412 N4=-1
NTST=2
NMND=33

335
401 IF (N4-NTST) = 0, 402, 403
402 DS = NSF
403 IF (MOD(N4,NMOD)) = 410, 406, 410

C PUNCH THETA AND TIME FOR FREQUENCY STUDY (LONGITUDINAL)

406 WRITE (?,409) Y(7), Y(13)
409 FORMAT (?F15.7)
410 N4 = N4 + 1
GO TO 7000

C CONTROL DR (IMPULSE), Lateral

1005 IF (ISW2) = 0, 1006, 500
500 IF (ICYC = 1) = 511, 512
511 N5 = 0
NTST = 1
NMOD = 0
GO TO 501
512 N5 = 1
NTST = 1
NMOD = 32
501 IF (N5-NTST) = 0, 502, 503
502 DR = ORF
503 IF (MOD(N5,NMOD)) = 10, 504, 510

C PUNCH PHI AND TIME FOR FREQUENCY STUDY (LATERAL)

504 WRITE (?,409) Y(9), Y(12)
510 N5 = N5 + 1
GO TO 750

C CONTROL ACCEL/DECEL + AUTOPILOT

600 IF (ISW2) = 0, 601, 600, 601
601 N6 = 1
TSW6 = 0
NN11 = 1
TYIM = 10.8*TIME+60.
TC = Y(12)
S001 = 0.
NC11 = 69*(ICYC/H)
NC12 = TIME*ICYC/H
UC = 0.
GO TO 652
651 GO TO(417,401,404,605,606,607,352), 419
652 UC = 0.
602 NN11 = NN11 + 1
IF (NN11-NC11) = 352, 367, 408

609 NN12 = 1
UC = 9, 455
NN12 = 0

336
603 NN12=NN12+1
IF(NN12-NC12) 352, 352, 600

609 IF(ISW6) 617, 616, 617
617 N6=7
UC=0.
GO TO 352

616 N6=3
UC=16,99
NN12=0
604 NN12=NN12+1
IF(NN12-NC12) 352, 352, 610

610 IF(ISW6) 619, 615, 619
619 GO TO 409

615 N6=4
UC=25,275
NN12=0
605 NN12=NN12+1
IF(NN12-NC12) 352, 352, 611

611 IF(ISW6) 619, 614, 619
619 GO TO 614

614 N6=5
UC=32,78
NN12=0
604 NN12=NN12+1
IF(NN12-NC12) 352, 352, 612

612 IF(ISW6) 620, 621, 623
623 GO TO 415

621 N6=6
UC=47,295
NN12=0
607 NN12=NN12+1
IF(NN12-NC12) 352, 352, 613

613 I$WA=1
GO TO 414

CONTROL MAXIMUM ACCEL/DECEL + AUTOPILOT

1007 IF(ISW2) 1701, 700, 701
710 N7=1
NN12=1
TL= $W+2.0$ TM
NC1=670 CYC/H
NC14=TIME*1 CYC/H
SPO3=1
ZC=V117

337
// JOB FC572
// EXEC FFNPTRAN
DIMENSION VI(8)
DIMENSION CB(5), Al(5), WJ1(10), DWJ(10)
DIMENSION XP(512), YP(512)
DIMENSION DATA(2000)
DIMENSION FRPR(50)
1 FORMAT(8F10.5)
2 FORMAT(1X, 8F15.5)
3 FORMAT(1X, 6E70.7)
4 FORMAT(415, 4F10.5)
5 FORMAT(70A4)
8 FORMAT(/5X, 29HTON MANY POINTS FOR DIMENSION/)
11 FORMAT(1H1)
12 FORMAT(15, 5X, F10.4)
13 FORMAT(1X, 27HPLNT THE TAPE ON UNIT '1R1')
14 FORMAT(1X, 3THNO PLOT CREATED, SORRY 'ROUT THAT')
15 FORMAT(/4X, 1HN, 3X, RHCONTROLS/)
16 FORMAT(16I5)
18 FORMAT(/1)
25 FORMAT(17HPSD OF FC572 DATA, 10X, THV =, F10.2, 2X, 3HKS )
1 /5F10.4/9X, 1H1/9X, 1H1/)
26 FORMAT(13HEND, 7X, 5H1024.,)
CALL PLOTS(DATA, RO00.5)
IN = 1
IOUT = 1
IPCH = 2
ICNSL = 15
NP = 512
IUSED = 0
ZPRC = 0.
XPNO = 9.
YPNO = 9.
IHX = 1HEX(14, 3, 4, 0, 4, 0, 4, 0)
IHY = 1HFX(14, 4, 4, 0, 4, 0, 4, 0)
PI = 3.141592654
CALL INITT(-1)
TWOPI = 6.283185308
SORT3D = .8660254
LM = 5
CR(1) = .5
CB(2) = SORT3D
CR(3) = 1.
CR(4) = SORT3D
CB(5) = .5
AI(1) = .09094
AI(2) = .47526
AI(3) = .51254
AI(4) = .47305
AI(5) = .09094
G = 32.2
CONG = 4.10F-3 * GEC
BC = (CONG+CONG)/PI
READ(IIN,4) NBAND,NT,NPLT,NPC,H,A,B,DT
WRITE(IOUT,15)
NBAND = 10
IF(NPLT)115,115,112
112 IF( NT - NPLT*NDP )115,115,113
113 CONTINUE
NPLT = (NT-10)/NDP + 1
115 CONTINUE
WRITE(IOUT,4) NRAND,NT,NPLT,NPC,H,A,B,DT
NPLTS = NPLT
RAND = NRAND
100 CONTINUE
READ(IIN,1) VI
C TEST FOR END
IF(VI(1))1999,999,10E
105 CONTINUE
DO 600 IV=1,8
V = VI(IV)
IF(V)605,605,205
205 CONTINUE
WRITE(IOUT,11)
U = V*1.688944
GOU = G/U
WRITE(IOUT,2) V,U,GOU
FACT1 = (R(IF-3*U*U))(/(2.94*GOU*GOU))
FACT2 = -.74*GOU*GOU*GOU
YO = EXP(FACT2/(A**A*A**A))
AY = EXP(FACT2/(R*R*A*AR)) - YO
ARFA = FACT1*AY
AR = ARFA/RAND
WRITE(IOUT,3) FACT1,FACT2,YO,AY,AREA,AR
AY = AY/BAND
WJ = A
IJ = 0
DO 200 J=1,NBAND
YO = YO + AY
AT = -.74/ALOG(YO)
X1 = GOU*SORT(SCRT(AT))
Y1 = YO*FACT1
WRITE(IOUT,3) X1,Y1,YO
WJ1 = WJ
WJ = X1
WJI(J) = .5*(WJ+WJ1)
NWJ(J) = WJ - WJ1
DO 200 L=1,LM
IJ = IJ + 1
CALL RAND*(RN)
FPRI(IJ) = RN*TWPRI
200 CONTINUE
C
JJ = 0
DO 500 IT=1,NT
TSUM = 0.
500 CONTINUE
TT = FLOAT(IT-1)*DT
IJ = 0
DO 400 J=1,NBRAND
WJ = WJI(J)
DW = DWJ(J)
WJ2 = WJ*WJ
WJ4 = WJ2*WJ2
WJ5 = WJ4*WJ
AR = BC*EXP(FACT2/WJ4)/WJ5
AT = WJ2/GOU
SUM = 0.
DO 390 L=1,LM
IJ = IJ + 1
EIJ = FPR(IJ)
GIJ = (AT*CR(L) - WJ)*TT + FIJ
SUM = SUM + COS(GIJ)*SORT(A(I))*DW
390 CONTINUE
TSUM = TSUM + AR*SUM
400 CONTINUE
C WRITE(IOUT,3)TT,TSUM
C IF(NPLT)425,425,405
405 IF(II(T/NPLT)*NPLT-IT)425,410,425
410 JJ = JJ+1
IF(JJ-NDP)420,420,415
415 WRITE(IOUT,1) 
IF(NPCH)525,525,418
418 NPLT = 0
GO TO 425
420 CONTINUE
XP(JJ) = TT
YP(JJ) = TSUM
425 CONTINUE
C IF(NPCH)465,465,450
450 CONTINUE
IF(IT-1)452,452,457
457 WRITE(IOUT,25) V,TURN,VT,XP,YP,TURN
453 CONTINUE
WRITE(IOUT,12) IT,TSUM
IF(IT-NT)466,466,465
464 WRITE(IOUT,7A)
465 CONTINUE
C 490 CONTINUE
C PLOT
475 CONTINUE
NPLT = NPLTc
IF(NPLT)575,575,550
550 CONTINUE
J = MIND(JJ,NDP)
XL = XJ.
YL = YJ.
DIV = T.

355
HT = .125
CALL SCALE(XP, XL, J, 1, DIV, 1)
CALL SCALE(YP, YL, J, 1, DIV, 1)
CALL AXIS(ZERO, ZERO, IHX, -4, XL, 0, 0, DIV, 1)
CALL AXIS(ZERO, ZERO, IHY, 4, YL, 90, 0, DIV, 2)
CALL LINE(XP, YP, J, 1, 0, 0)
IUSED = IUSED + 1
CALL PLOT(XL, 0, 0, -3)
575 CONTINUE

C
600 CONTINUE
605 CONTINUE
GO TO 100
999 CONTINUE
IF (IUSED) 998, 997, 999
998 CONTINUE
CALL PLOT(8., 0., 999)
WRITE(ICASL, 13)
GO TO 996
997 CONTINUE
WRITE(ICNSL, 14)
996 CONTINUE
CALL EXIT
END
The subroutine INTEG used with program ES920 Submarine Simulation is programmed to use three different integration techniques; Euler, 2nd Order Adams, and a 2nd non-classical method (012). Table 24 contains the coefficients for these and 19 other methods that can be programmed into this subroutine by means of the equation:

\[ Y_n = \sum_{i=1}^{3} c_i Y_{n-i} + h \sum_{i=0}^{2} b_i \frac{d^2}{dt^2} Y_{n-i}. \]
### TABLE 25: POPULAR NUMERICAL INTEGRATION TECHNIQUES

\[ y_n = \sum_{i=1}^{3} a_i y_{n-i} + h \sum_{i=0}^{3} b_i y_{n-i} \]

<table>
<thead>
<tr>
<th>Method</th>
<th>Type</th>
<th>( a_1 )</th>
<th>( a_2 )</th>
<th>( a_3 )</th>
<th>( b_0 )</th>
<th>( b_1 )</th>
<th>( b_2 )</th>
<th>( b_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Euler</td>
<td>O₁₁</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Backward Rectangular</td>
<td>C₁₁</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2nd Order Adams</td>
<td>O₁₂</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>3/2</td>
<td>-1/2</td>
<td></td>
</tr>
<tr>
<td>Trapezoidal</td>
<td>C₁₂</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>1/2</td>
<td>1/2</td>
<td></td>
</tr>
<tr>
<td>Ösg Mod Gurk</td>
<td>O₃₁</td>
<td>1.1462</td>
<td>-0.2011</td>
<td>0.0549</td>
<td></td>
<td>1.6416</td>
<td>-1.0080</td>
<td>0.2751</td>
</tr>
<tr>
<td>Classic Ö₂₃</td>
<td>O₃₃</td>
<td>-18</td>
<td>9</td>
<td>10</td>
<td></td>
<td>9</td>
<td>18</td>
<td>3</td>
</tr>
<tr>
<td>Simpson</td>
<td>C₁₃</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>1/3</td>
<td>4/3</td>
<td>1/3</td>
</tr>
<tr>
<td>Ösg C₃₁ Mod Gurk</td>
<td>O₃₀</td>
<td>1.807</td>
<td>-1.109</td>
<td>0.303</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Classic Ö₃₀</td>
<td>C₃₀</td>
<td>1.146</td>
<td>-0.201</td>
<td>0.055</td>
<td></td>
<td>0.909</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Classic C₃₁</td>
<td>C₃₁</td>
<td>18/11</td>
<td>-9/11</td>
<td>2/11</td>
<td></td>
<td>6/11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adams - Bashforth</td>
<td>C₁₄</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>9/24</td>
<td>19/24</td>
<td>-5/24</td>
</tr>
<tr>
<td>Best O₁₂ Method Based† on Stability Alone</td>
<td>O₁₂</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>3/4</td>
<td>1/4</td>
<td></td>
</tr>
<tr>
<td>1/3 Rule</td>
<td>C₂₄</td>
<td>1/2</td>
<td>1/2</td>
<td></td>
<td>17/48</td>
<td>51/48</td>
<td>3/48</td>
<td>1/48</td>
</tr>
<tr>
<td>Parabolic</td>
<td>O₁₃</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>23/12</td>
<td>-4/3</td>
<td>5/12</td>
</tr>
<tr>
<td>Classic</td>
<td>O₁₁</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Classic</td>
<td>O₂₂</td>
<td>-4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Classic</td>
<td>C₂₂</td>
<td>8/10</td>
<td>2/10</td>
<td></td>
<td>4/10</td>
<td>8/10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Classic</td>
<td>C₁₃</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td>5/12</td>
<td>2/3</td>
<td>-1/12</td>
</tr>
<tr>
<td>Classic</td>
<td>C₂₂</td>
<td>9/17</td>
<td>9/17</td>
<td>-1/17</td>
<td>6/17</td>
<td>18/17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1/3 Rule</td>
<td>C₂₄</td>
<td>1/3</td>
<td>1/3</td>
<td>1/3</td>
<td>12/36</td>
<td>39/36</td>
<td>15/36</td>
<td>5/36</td>
</tr>
<tr>
<td>2/3 Rule</td>
<td>C₂₄</td>
<td>2/3</td>
<td>1/3</td>
<td></td>
<td>25/72</td>
<td>91/72</td>
<td>43/72</td>
<td>9/72</td>
</tr>
</tbody>
</table>

† Denotes a non-classic method
## GLOSSARY

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Dimensionless Form</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>( B' = \frac{B}{\frac{1}{2} \rho l^2 U^2} )</td>
<td>Buoyancy force, positive upward</td>
</tr>
<tr>
<td>CB</td>
<td></td>
<td>Center of buoyancy of submarine</td>
</tr>
<tr>
<td>CG</td>
<td></td>
<td>Center of mass of submarine</td>
</tr>
<tr>
<td>( I_x )</td>
<td>( I'_x = \frac{1}{\frac{1}{2} \rho l^2} )</td>
<td>Moment of inertia of submarine about x axis</td>
</tr>
<tr>
<td>( I_y )</td>
<td>( I'_y = \frac{1}{\frac{1}{2} \rho l^2} )</td>
<td>Moment of inertia of submarine about y axis</td>
</tr>
<tr>
<td>( I_z )</td>
<td>( I'_z = \frac{1}{\frac{1}{2} \rho l^2} )</td>
<td>Moment of inertia of submarine about z axis</td>
</tr>
<tr>
<td>( I_{xy} )</td>
<td>( I'_{xy} = \frac{1}{\frac{1}{2} \rho l^2} )</td>
<td>Product of inertia about xy axis</td>
</tr>
<tr>
<td>( I_{yz} )</td>
<td>( I'_{yz} = \frac{1}{\frac{1}{2} \rho l^2} )</td>
<td>Product of inertia about yz axes</td>
</tr>
<tr>
<td>( I_{zx} )</td>
<td>( I'_{zx} = \frac{1}{\frac{1}{2} \rho l^2} )</td>
<td>Product of inertia about zx axes</td>
</tr>
<tr>
<td>K</td>
<td>( K' = \frac{K}{\frac{1}{2} \rho l^2 U^2} )</td>
<td>Hydrodynamic moment component about x axis (rolling moment)</td>
</tr>
<tr>
<td>( K_e )</td>
<td>( K'_e = \frac{K_e}{\frac{1}{2} \rho l^2 U^2} )</td>
<td>Rolling moment when body angle (( \alpha, \beta )) and control surface angles are zero</td>
</tr>
<tr>
<td>( K_{e\eta} )</td>
<td>( K'<em>{e\eta} = \frac{K</em>{e\eta}}{\frac{1}{2} \rho l^2 U^2} )</td>
<td>Coefficient used in representing ( K_e ) as a function of ( \eta - 1 )</td>
</tr>
<tr>
<td>( K_p )</td>
<td>( K'_p = \frac{K_p}{\frac{1}{2} \rho l^2 U} )</td>
<td>First order coefficient used in representing ( K ) as a function of ( p )</td>
</tr>
<tr>
<td>( K_{p</td>
<td>p</td>
<td>} )</td>
</tr>
<tr>
<td>( K_{pq} )</td>
<td>( K'<em>{pq} = \frac{K</em>{pq}}{\frac{1}{2} \rho l^2} )</td>
<td>Coefficient used in representing ( K ) as a function of the product ( pq )</td>
</tr>
<tr>
<td>Symbol</td>
<td>Expression</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>$K_{qr}$</td>
<td>$K_{qr}' = \frac{K_{qr}}{\delta \rho \lambda^2}$</td>
<td>Coefficient used in representing $K$ as a function of the product $qr$</td>
</tr>
<tr>
<td>$K_r$</td>
<td>$K_r' = \frac{K_r}{\delta \rho \lambda^2}$</td>
<td>First order coefficient used in representing $K$ as a function of $r$</td>
</tr>
<tr>
<td>$K_\dot{r}$</td>
<td>$K_\dot{r}' = \frac{K_\dot{r}}{\delta \rho \lambda^2}$</td>
<td>Coefficient used in representing $K$ as a function of $\dot{r}$</td>
</tr>
<tr>
<td>$K_v$</td>
<td>$K_v' = \frac{K_v}{\delta \rho \lambda^2}$</td>
<td>First order coefficient used in representing $K$ as a function of $v$</td>
</tr>
<tr>
<td>$K_\dot{v}$</td>
<td>$K_\dot{v}' = \frac{K_\dot{v}}{\delta \rho \lambda^2}$</td>
<td>Coefficient used in representing $K$ as a function of $\dot{v}$</td>
</tr>
<tr>
<td>$K_{v</td>
<td>v</td>
<td>}$</td>
</tr>
<tr>
<td>$K_{vq}$</td>
<td>$K_{vq}' = \frac{K_{vq}}{\delta \rho \lambda^2}$</td>
<td>Coefficient used in representing $K$ as a function of the product $vq$</td>
</tr>
<tr>
<td>$K_{ww}$</td>
<td>$K_{ww}' = \frac{K_{ww}}{\delta \rho \lambda^2}$</td>
<td>Coefficient used in representing $K$ as a function of the product $ww$</td>
</tr>
<tr>
<td>$K_{wp}$</td>
<td>$K_{wp}' = \frac{K_{wp}}{\delta \rho \lambda^2}$</td>
<td>Coefficient used in representing $K$ as a function of the product $wp$</td>
</tr>
<tr>
<td>$K_{wr}$</td>
<td>$K_{wr}' = \frac{K_{wr}}{\delta \rho \lambda^2}$</td>
<td>Coefficient used in representing $K$ as a function of the product $wr$</td>
</tr>
<tr>
<td>$K_{\delta r}$</td>
<td>$K_{\delta r}' = \frac{K_{\delta r}}{\delta \rho \lambda^2}$</td>
<td>First order coefficient used in representing $K$ as a function of $\delta_r$</td>
</tr>
<tr>
<td>$l$</td>
<td>$l' = l$</td>
<td>Overall length of submarine</td>
</tr>
<tr>
<td>$m$</td>
<td>$m' = \frac{m}{\delta \rho \lambda^2}$</td>
<td>Mass of submarine, including water in free-flooding spaces</td>
</tr>
<tr>
<td>$M$</td>
<td>$M' = \frac{M}{\delta \rho \lambda^2}$</td>
<td>Hydrodynamic moment component about $y$ axis (pitching moment)</td>
</tr>
<tr>
<td>$M_\phi$</td>
<td>$M_\phi' = \frac{M_\phi}{\delta \rho \lambda^2}$</td>
<td>Pitching moment when body angles ($\phi$, $\theta$) and control surface angles are zero</td>
</tr>
<tr>
<td>$M_{pp}$</td>
<td>$M_{pp}' = \frac{M_{pp}}{\delta \rho \lambda^2}$</td>
<td>Second order coefficient used in representing $M$ as a function of $p$. First order coefficient is zero.</td>
</tr>
<tr>
<td>$M_q$</td>
<td>$M_q' = \frac{M_q}{\delta \rho \lambda^2}$</td>
<td>First order coefficient used in representing $M$ as a function of $q$</td>
</tr>
<tr>
<td>$M_{\eta\eta}$</td>
<td>$M_{\eta\eta}' = \frac{M_{\eta\eta}}{\delta \rho \lambda^2}$</td>
<td>First order coefficient used in representing $M$ as a function of $(\eta-1)$</td>
</tr>
<tr>
<td>$M_q$</td>
<td>$M_q' = \frac{M_q}{\delta \rho \lambda^2}$</td>
<td>Coefficient used in representing $M$ as a function of $\dot{q}$</td>
</tr>
<tr>
<td>Term</td>
<td>Expression</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>------------</td>
<td></td>
</tr>
<tr>
<td>$M_q</td>
<td>q$</td>
<td>$M_q</td>
</tr>
<tr>
<td>$M</td>
<td>q</td>
<td>\delta s$</td>
</tr>
<tr>
<td>$M_{rp}$</td>
<td>$M_{rp}</td>
<td>= \frac{M_{rp}}{\frac{\partial}{\partial r}}$</td>
</tr>
<tr>
<td>$M_{rr}$</td>
<td>$M_{rr}</td>
<td>= \frac{M_{rr}}{\frac{\partial}{\partial r}}$</td>
</tr>
<tr>
<td>$M_{vp}$</td>
<td>$M_{vp}</td>
<td>= \frac{M_{vp}}{\frac{\partial}{\partial v}}$</td>
</tr>
<tr>
<td>$M_{vr}$</td>
<td>$M_{vr}</td>
<td>= \frac{M_{vr}}{\frac{\partial}{\partial v}}$</td>
</tr>
<tr>
<td>$M_{vv}$</td>
<td>$M_{vv}</td>
<td>= \frac{M_{vv}}{\frac{\partial}{\partial v}}$</td>
</tr>
<tr>
<td>$M_w$</td>
<td>$M_w</td>
<td>= \frac{M_w}{\frac{\partial}{\partial w}}$</td>
</tr>
<tr>
<td>$M_w</td>
<td>\eta$</td>
<td>$M_w</td>
</tr>
<tr>
<td>$M_w</td>
<td>w$</td>
<td>$M_w</td>
</tr>
<tr>
<td>$M_w</td>
<td>w</td>
<td>\eta$</td>
</tr>
<tr>
<td>$M_{wb}$</td>
<td>$M_{wb}</td>
<td>= \frac{M_{wb}}{\frac{\partial}{\partial w}}$</td>
</tr>
<tr>
<td>$M_{\delta s}$</td>
<td>$M_{\delta s}</td>
<td>= \frac{M_{\delta s}}{\frac{\partial}{\partial \delta s}}$</td>
</tr>
<tr>
<td>$M_{\delta s</td>
<td>\eta}$</td>
<td>$M_{\delta s</td>
</tr>
</tbody>
</table>

Second order coefficient used in representing $M$ as a function of $q$.

Coefficient used in representing $M_{\delta s}$ as a function of $q$.

Coefficient used in representing $M$ as a function of the product $r$. First order coefficient is zero.

Coefficient used in representing $M$ as a function of the product $v$.

Coefficient used in representing $M$ as a function of the product $v$.

Second order coefficient used in representing $M$ as a function of $v$.

First order coefficient used in representing $M$ as a function of $w$.

First order coefficient used in representing $M_w$ as a function of $(\eta - 1)$.

Coefficient used in representing $M$ as a function of $\delta s$.

First order coefficient used in representing $M_{\delta s}$ as a function of $\delta s$.

First order coefficient used in representing $M_{\delta s | \eta}$ as a function of $\eta - 1$.
Hydrodynamic moment component about \( z \) axis (yawing moment):

\[ N = N' = \frac{N}{\frac{d}{\rho c^2} U^2} \]

Yawing moment when body angles \((\alpha, \beta)\) and control surface angles are zero:

\[ N_a = N_a' = \frac{N_a}{\frac{d}{\rho c^2} U} \]

First order coefficient used in representing \( N \) as a function of \( p \):

\[ N_p = N_p' = \frac{N_p}{\frac{d}{\rho c^2} U} \]

Coefficient used in representing \( N \) as a function of \( \dot{p} \):

\[ N_{\dot{p}} = N_{\dot{p}}' = \frac{N_{\dot{p}}}{\frac{d}{\rho c^2}} \]

Coefficient used in representing \( N \) as a function of the product \( pq \):

\[ N_{pq} = N_{pq}' = \frac{N_{pq}}{\frac{d}{\rho c^2}} \]

Coefficient used in representing \( N \) as a function of the product \( qr \):

\[ N_{qr} = N_{qr}' = \frac{N_{qr}}{\frac{d}{\rho c^2}} \]

First order coefficient used in representing \( N \) as a function of \( r \):

\[ N_r = N_r' = \frac{N_r}{\frac{d}{\rho c^2} U} \]

First order coefficient used in representing \( N_r \) as a function of \((\eta-1)\):

\[ N_{r\eta} = N_{r\eta}' = \frac{N_{r\eta}}{\frac{d}{\rho c^2} U} \]

Coefficient used in representing \( N \) as a function of \( \dot{r} \):

\[ N_{\dot{r}} = N_{\dot{r}}' = \frac{N_{\dot{r}}}{\frac{d}{\rho c^2}} \]

Second order coefficient used in representing \( N \) as a function of \( r \):

\[ N_{r|r} = N_{r|r}' = \frac{N_{r|r}}{\frac{d}{\rho c^2}} \]

Coefficient used in representing \( N_{\delta r} \) as a function of \( r \):

\[ N_{r|\delta r} = N_{r|\delta r}' = \frac{N_{r|\delta r}}{\frac{d}{\rho c^2} U} \]

First order coefficient used in representing \( N \) as a function of \( v \):

\[ N_v = N_v' = \frac{N_v}{\frac{d}{\rho c^2} U} \]

First order coefficient used in representing \( N_v \) as a function of \((\eta-1)\):

\[ N_{v\eta} = N_{v\eta}' = \frac{N_{v\eta}}{\frac{d}{\rho c^2} U} \]

Coefficient used in representing \( N \) as a function of \( \dot{v} \):

\[ N_{\dot{v}} = N_{\dot{v}}' = \frac{N_{\dot{v}}}{\frac{d}{\rho c^2}} \]

Coefficient used in representing \( N \) as a function of the product \( vq \):

\[ N_{vq} = N_{vq}' = \frac{N_{vq}}{\frac{d}{\rho c^2} U} \]

Coefficient used in representing \( N \) as a function of \( v \):

\[ N_{|v|r} = N_{|v|r}' = \frac{N_{|v|r}}{\frac{d}{\rho c^2}} \]

Coefficient used in representing \( N_r \) as a function of \( v \):

\[ N_{v|v} = N_{v|v}' = \frac{N_{v|v}}{\frac{d}{\rho c^2} U} \]

Second order coefficient used in representing \( N \) as a function of \( v \):

\[ N_{v|v|v} = N_{v|v|v}' = \frac{N_{v|v|v}}{\frac{d}{\rho c^2}} \]

Coefficient used in representing \( N \) as a function of \((\eta-1)\):

\[ N_{v|v|\eta} = N_{v|v|\eta}' = \frac{N_{v|v|\eta}}{\frac{d}{\rho c^2} U} \]
<table>
<thead>
<tr>
<th>Variable</th>
<th>Equation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N_{vw}$</td>
<td>$N_{vw}' = \frac{N_{vw}}{\theta_\theta t^2}$</td>
<td>Coefficient used in representing $N$ as a function of the product $vw$</td>
</tr>
<tr>
<td>$N_{wp}$</td>
<td>$N_{wp}' = \frac{N_{wp}}{\theta_\theta t^2}$</td>
<td>Coefficient used in representing $N$ as a function of the product $wp$</td>
</tr>
<tr>
<td>$N_{wr}$</td>
<td>$N_{wr}' = \frac{N_{wr}}{\theta_\theta t^2}$</td>
<td>Coefficient used in representing $N$ as a function of the product $wr$</td>
</tr>
<tr>
<td>$N_{\delta r}$</td>
<td>$N_{\delta r}' = \frac{N_{\delta r}}{\theta_\theta t^2 U^2}$</td>
<td>First order coefficient used in representing $N$ as a function of $\delta r$</td>
</tr>
<tr>
<td>$N_{\delta_\eta}$</td>
<td>$N_{\delta_\eta}' = \frac{N_{\delta_\eta}}{\theta_\theta t^2 U^2}$</td>
<td>First order coefficient used in representing $N_{\delta r}$ as a function of $(\eta - 1)$</td>
</tr>
<tr>
<td>$p$</td>
<td>$p' = -\frac{\theta_\theta t}{U}$</td>
<td>Angular velocity component about $y$ axis relative to fluid (roll)</td>
</tr>
<tr>
<td>$\dot{p}$</td>
<td>$\dot{p}' = \frac{\theta_\theta t^2}{U^2}$</td>
<td>Angular acceleration component about $x$ axis relative to fluid</td>
</tr>
<tr>
<td>$q$</td>
<td>$q' = -\frac{\theta_\theta t}{U}$</td>
<td>Angular velocity component about $y$ axis relative to fluid (pitch)</td>
</tr>
<tr>
<td>$\dot{q}$</td>
<td>$\dot{q}' = \frac{\theta_\theta t^2}{U^2}$</td>
<td>Angular acceleration component about $y$ axis relative to fluid</td>
</tr>
<tr>
<td>$r$</td>
<td>$r' = \frac{\theta_\theta t}{U}$</td>
<td>Angular velocity component about $z$ axis relative to fluid (yaw)</td>
</tr>
<tr>
<td>$\dot{r}$</td>
<td>$\dot{r}' = \frac{\theta_\theta t^2}{U^2}$</td>
<td>Angular acceleration component about $z$ axis relative to fluid</td>
</tr>
<tr>
<td>$U$</td>
<td>$U' = \frac{U}{U}$</td>
<td>Linear velocity of origin of body axes relative to fluid</td>
</tr>
<tr>
<td>$u$</td>
<td>$u' = \frac{u}{U}$</td>
<td>Component of $U$ in direction of the $x$ axis</td>
</tr>
<tr>
<td>$\dot{u}$</td>
<td>$\dot{u}' = \frac{\theta_\theta t}{U^2}$</td>
<td>Time rate of change of $u$ in direction of the $x$ axis</td>
</tr>
<tr>
<td>$u_c$</td>
<td>$u_c' = \frac{u_c}{U}$</td>
<td>Command speed: steady value of ahead speed component $u$ for a given propeller rpm when body angles $(\alpha, \beta)$ and control surface angles are zero. Sign changes with propeller reversal</td>
</tr>
<tr>
<td>$v$</td>
<td>$v' = \frac{v}{U}$</td>
<td>Component of $U$ in direction of the $y$ axis</td>
</tr>
<tr>
<td>$\dot{v}$</td>
<td>$\dot{v}' = \frac{\theta_\theta t}{U^2}$</td>
<td>Time rate of change of $v$ in direction of the $y$ axis</td>
</tr>
</tbody>
</table>
Component of U in direction of the \( \nu \) axis

Time rate of change of \( w \) in direction of the \( z \) axis

Weight, including water in free flooding spaces

Longitudinal body axis; also the coordinate of a point relative to the origin of body axes

The \( x \) coordinate of \( CB \)

The \( x \) coordinate of \( CG \)

A coordinate of the displacement of \( CG \) relative to the origin of a set of fixed axes

Hydrodynamic force component along \( x \) axis (longitudinal, or axial, force)

Second order coefficient used in representing \( X \) as a function of \( q \). First order coefficient is zero

Coefficient used in representing \( X \) as a function of the product \( rp \)

Second order coefficient used in representing \( X \) as a function of \( r \). First order coefficient is zero

Coefficient used in representing \( X \) as a function of \( \dot{u} \)

Second order coefficient used in representing \( X \) as a function of \( u \) in the non-propelled case. First order coefficient is zero

Coefficient used in representing \( X \) as a function of the product \( \dot{v}r \)

Second order coefficient used in representing \( X \) as a function of \( v \). First order coefficient is zero

First order coefficient used in representing \( X_{vv} \) as a function of \( (\eta - 1) \)

Coefficient used in representing \( X \) as a function of the product \( wq \)
Second order coefficient used in representing $X$ as a function of $w$. First order coefficient is zero

First order coefficient used in representing $X_{ww}$ as a function of $(\eta-1)$

Second order coefficient used in representing $X$ as a function of $\delta_b$. First order coefficient is zero

Second order coefficient used in representing $X$ as a function of $\delta_r$. First order coefficient is zero

First order coefficient used in representing $X_{br}$ as a function of $(\eta-1)$

Second order coefficient used in representing $X$ as a function of $\delta_s$. First order coefficient is zero

First order coefficient used in representing $X_{s\delta}$ as a function of $(\eta-1)$

Lateral body axis; also the coordinate of a point relative to the origin of body axes

The $y$ coordinate of CB

The $y$ coordinate of CG

A coordinate of the displacement of CG relative to the origin of a set of fixed axes

Hydrodynamic force component along $y$ axis (lateral force)

Lateral force when body angles $(\alpha, \beta)$ and control surface angles are zero

First order coefficient used in representing $Y$ as a function of $p$

Coefficient used in representing $Y$ as a function of $\dot{p}$

Second order coefficient used in representing $Y$ as a function of $p$
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Equation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y_{pq}$</td>
<td>$Y'<em>{pq} = \frac{Y</em>{pq}}{\delta pq}$</td>
<td>Coefficient used in representing $Y$ as a function of the product $pq$</td>
</tr>
<tr>
<td>$Y_{qr}$</td>
<td>$Y'<em>{qr} = \frac{Y</em>{qr}}{\delta qr}$</td>
<td>Coefficient used in representing $Y$ as a function of the product $qr$</td>
</tr>
<tr>
<td>$Y_r$</td>
<td>$Y'<em>{r} = \frac{Y</em>{r}}{\delta r}$</td>
<td>First order coefficient used in representing $Y$ as a function of $r$</td>
</tr>
<tr>
<td>$Y_{\eta \eta}$</td>
<td>$Y'<em>{\eta \eta} = \frac{Y</em>{\eta \eta}}{\delta \eta \eta}$</td>
<td>First order coefficient used in representing $Y_\eta$ as a function of $(\eta-1)$</td>
</tr>
<tr>
<td>$Y_\tau$</td>
<td>$Y'<em>{\tau} = \frac{Y</em>{\tau}}{\delta \tau}$</td>
<td>Coefficient used in representing $Y$ as a function of $\tau$</td>
</tr>
<tr>
<td>$Y</td>
<td>_{</td>
<td>r</td>
</tr>
<tr>
<td>$Y_v$</td>
<td>$Y'<em>{v} = \frac{Y</em>{v}}{\delta v}$</td>
<td>First order coefficient used in representing $Y$ as a function of $v$</td>
</tr>
<tr>
<td>$Y_{v \eta}$</td>
<td>$Y'<em>{v \eta} = \frac{Y</em>{v \eta}}{\delta v \eta}$</td>
<td>First order coefficient used in representing $Y_v$ as a function of $(\eta-1)$</td>
</tr>
<tr>
<td>$Y_\psi$</td>
<td>$Y'<em>{\psi} = \frac{Y</em>{\psi}}{\delta \psi}$</td>
<td>Coefficient used in representing $Y$ as a function of $\psi$</td>
</tr>
<tr>
<td>$Y_{v q}$</td>
<td>$Y'<em>{v q} = \frac{Y</em>{v q}}{\delta v q}$</td>
<td>Coefficient used in representing $Y$ as a function of the product $v q$</td>
</tr>
<tr>
<td>$Y_{v</td>
<td>r</td>
<td>}$</td>
</tr>
<tr>
<td>$Y_{v v}$</td>
<td>$Y'<em>{v v} = \frac{Y</em>{v v}}{\delta v v}$</td>
<td>Second order coefficient used in representing $Y$ as a function of $v$</td>
</tr>
<tr>
<td>$Y_{v v v}$</td>
<td>$Y'<em>{v v v} = \frac{Y</em>{v v v}}{\delta v v v}$</td>
<td>First order coefficient used in representing $Y_{v v}$ as a function of $(\eta-1)$</td>
</tr>
<tr>
<td>$Y_{v w}$</td>
<td>$Y'<em>{v w} = \frac{Y</em>{v w}}{\delta v w}$</td>
<td>Coefficient used in representing $Y$ as a function of the product $v w$</td>
</tr>
<tr>
<td>$Y_{w p}$</td>
<td>$Y'<em>{w p} = \frac{Y</em>{w p}}{\delta w p}$</td>
<td>Coefficient used in representing $Y$ as a function of the product $w p$</td>
</tr>
<tr>
<td>$Y_{w r}$</td>
<td>$Y'<em>{w r} = \frac{Y</em>{w r}}{\delta w r}$</td>
<td>Coefficient used in representing $Y$ as a function of the product $w r$</td>
</tr>
<tr>
<td>$Y_{8 r}$</td>
<td>$Y'<em>{8 r} = \frac{Y</em>{8 r}}{\delta 8 r}$</td>
<td>First order coefficient used in representing $Y$ as a function of $8 r$</td>
</tr>
<tr>
<td>$Y_{8 r \eta}$</td>
<td>$Y'<em>{8 r \eta} = \frac{Y</em>{8 r \eta}}{\delta 8 r \eta}$</td>
<td>First order coefficient used in representing $Y_{8 r}$ as a function of $(\eta-1)$</td>
</tr>
</tbody>
</table>
Normal body axis; also the coordinate of a
point relative to the origin of body axes

The z coordinate of CB

The z coordinate of CG

A coordinate of the displacement of CG
relative to the origin of a set of fixed axes

Hydrodynamic force component along z
axis (normal force)

Normal force when body angles (α, β) and
control surface angles are zero

Second order coefficient used in representing
Z as a function of p. First order coefficient
is zero

First order coefficient used in representing
Z as a function of q

First order coefficient used in representing
Zq as a function of (η-1)

Coefficient used in representing Z as a
function of q

Coefficient used in representing Z6s as a
function of q

Coefficient used in representing Z as a
function of the product rp

Coefficient used in representing Z as a
function of r. First order coefficient
is zero

First order coefficient used in representing
Z as a function of w

First order coefficient used in representing
Zw as a function of (η-1)

Coefficient used in representing Z as a
function of \( \dot{w} \)

First order coefficient used in representing
Z as a function of \( \dot{w} \); equal to zero for sym-
matical function

Coefficient used in representing Zw as a
function of q
Second order coefficient used in representing $Z$ as a function of $w$

First order coefficient used in representing $Z_{w|w|}$ as a function of $(\eta-1)$

Second order coefficient used in representing $Z$ as a function of $w$; equal to zero for symmetrical function

First order coefficient used in representing $Z$ as a function of $\delta_b$

First order coefficient used in representing $Z$ as a function of $\delta_s$

First order coefficient used in representing $Z_{\delta_s\eta}$ as a function of $(\eta-1)$

$\alpha$
Angle of attack

$\beta$
Angle of drift

$\delta_b$
Deflection of bowplane or sailplane

$\delta_r$
Deflection of rudder

$\delta_s$
Deflection of sternplane

$\eta$
The ratio $\frac{u_c}{U}$

$\theta$
Angle of pitch

$\psi$
Angle of yaw

$\phi$
Angle of roll

$a_i$, $b_i$, $c_i$
Sets of constants used in the representation of propeller thrust in the axial equation
This report covers the FORTRAN programming for two submarine simulations using the Naval Ship Research and Development Center standard equations of motion. Programs are given to extend the main simulation programs for research. A small computer simulation and a method of generating random ocean waves is included. The report covers program descriptions, FORTRAN listings, flow charts, input decks and output sheets.
ADVANCED SUBMARINE SYSTEMS PROGRAMING

**Abstract**

This programming report is the result of a study leading to the determination of the optimum sets of equations of motion to be used with two general types of submarine control trainers. The starting point was the Naval Ship Research and Development Center standard equations of motion for submarine simulators.

Two complete submarine simulation programs using these equations are given; one for six-degrees-of-freedom and one for the longitudinal three-degrees-of-freedom. A number of programs are included to assist the researcher in interpreting the results of the main simulation program. A compact submarine simulation program for use with a small computer is given and a method of generating random ocean wave amplitudes is outlined along with its program.

This report describes the programs, including listing in FORTRAN, flow charts, input decks, and typical output sheets, but does not tell how they are to be used. It should be used in conjunction with Advanced Submarine Systems Equation Study, NAVTRADEVCEN 68-C-0050-1 which describes the work performed under this study.
<table>
<thead>
<tr>
<th>KEY WORDS</th>
<th>LINK A</th>
<th></th>
<th>LINK B</th>
<th></th>
<th>LINK C</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ROLE</td>
<td>WT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Simulators</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Training Devices</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Digital Simulation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Submarine Simulators</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Equations of Motion</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mathematical Models</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Numerical Integration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ocean Waves</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>