A NONLINEAR MATHEMATICAL MODEL OF MOTIONS OF A PLANING BOAT IN REGULAR WAVES

by

Ernest E. Zarnick

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

SHIP PERFORMANCE DEPARTMENT
RESEARCH AND DEVELOPMENT REPORT
A nonlinear mathematical model has been formulated of a craft having a constant deadrise angle, planing in regular waves, using a modified low-aspect-ratio or strip theory. It was assumed that the wavelengths would be large in comparison to the craft length and that the wave slopes would be small. The coefficients in the equations of motion were determined by a combination of theoretical and empirical relationships. A simplified version for the case of a craft or model being towed at constant speed was programmed for computations on a digital computer, and the results were compared with existing experimental data.

(Continued on reverse side)
Comparison of computed pitch and heave motions and phase angles with corresponding experimental data was remarkably good. Comparison of bow and center of gravity vertical accelerations was fair to good.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>1</td>
</tr>
<tr>
<td>ADMINISTRATIVE INFORMATION</td>
<td>1</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>MATHEMATICAL FORMULATION</td>
<td>2</td>
</tr>
<tr>
<td>GENERAL</td>
<td>2</td>
</tr>
<tr>
<td>TWO-DIMENSIONAL HYDRODYNAMIC FORCE</td>
<td>3</td>
</tr>
<tr>
<td>TOTAL HYDRODYNAMIC FORCE AND MOMENT</td>
<td>6</td>
</tr>
<tr>
<td>EQUATIONS OF MOTION, GENERAL</td>
<td>8</td>
</tr>
<tr>
<td>EQUATIONS OF MOTION, SIMPLIFIED FOR CONSTANT SPEED</td>
<td>9</td>
</tr>
<tr>
<td>COMPARISON OF COMPUTED RESULTS WITH EXPERIMENTS</td>
<td>10</td>
</tr>
<tr>
<td>CONCLUSIONS AND RECOMMENDATIONS</td>
<td>13</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>13</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>33</td>
</tr>
<tr>
<td>APPENDIX A – EVALUATION OF HYDRODYNAMIC FORCE AND MOMENT INTEGRALS</td>
<td>35</td>
</tr>
<tr>
<td>APPENDIX B – COMPUTER PROGRAM DESCRIPTIONS</td>
<td>39</td>
</tr>
</tbody>
</table>

**LIST OF FIGURES**

1 – Coordinate System ................................................................. 14
2 – Types of Two-Dimensional Flow ................................................ 14
3 – Lines of Prismatic Models ...................................................... 15
4 – Sample Time Histories of Computed Pitch and Heave Motions .......... 16
5 – Sample Time Histories of Computed Accelerations of Bow and Center of Gravity ................................................................. 17
6 – Variation of Pitch and Heave with Wave Height .......................... 18
7 – Variation of Acceleration of Bow and Center of Gravity with Wave Height .................................................................................. 19
8 – Trajectory of Computer Model Relative to Wave

9 – Heave Response for 10-Degree Deadrise Model at $V/\sqrt{L} = 6.0$

10 – Pitch Response for 10-Degree Deadrise Model at $V/\sqrt{L} = 6.0$

11 – Heave Response for 20-Degree Deadrise Model at $V/\sqrt{L} = 6.0$

12 – Pitch Response for 20-Degree Deadrise Model at $V/\sqrt{L} = 6.0$

13 – Heave Response for 30-Degree Deadrise Model at $V/\sqrt{L} = 6.0$

14 – Pitch Response for 30-Degree Deadrise Model at $V/\sqrt{L} = 6.0$

15 – Heave Response for 20-Degree Deadrise Model at $V/\sqrt{L} = 4.0$

16 – Pitch Response for 20-Degree Deadrise Model at $V/\sqrt{L} = 4.0$

17 – Bow Acceleration for 10-Degree Deadrise Model at $V/\sqrt{L} = 6.0$

18 – Center of Gravity Acceleration for 10-Degree Deadrise Model at $V/\sqrt{L} = 6.0$

19 – Bow and Center of Gravity Accelerations for 20-Degree Deadrise Model at $V/\sqrt{L} = 4.0$ and $V/\sqrt{L} = 6.0$

20 – Bow and Center of Gravity Accelerations for 30-Degree Deadrise Model at $V/\sqrt{L} = 6.0$

Table 1 – Model Characteristics and Wave Conditions for Computations

Page
20
21
22
23
24
25
26
27
28
29
30
31
32
<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Mass matrix</td>
</tr>
<tr>
<td>A&lt;sub&gt;R&lt;/sub&gt;</td>
<td>Section area</td>
</tr>
<tr>
<td>a</td>
<td>Correction factor for buoyancy force</td>
</tr>
<tr>
<td>b</td>
<td>Half-beam of craft</td>
</tr>
<tr>
<td>C&lt;sub&gt;D,c&lt;/sub&gt;</td>
<td>Crossflow drag coefficient</td>
</tr>
<tr>
<td>C&lt;sub&gt;Δ&lt;/sub&gt;</td>
<td>Load coefficient Δ/ρg(2b)&lt;sup&gt;3&lt;/sup&gt;</td>
</tr>
<tr>
<td>C&lt;sub&gt;λ&lt;/sub&gt;</td>
<td>Wavelength coefficient L/λ[C&lt;sub&gt;Δ&lt;/sub&gt;/(L/2b)&lt;sup&gt;2&lt;/sup&gt;]&lt;sup&gt;1/3&lt;/sup&gt;</td>
</tr>
<tr>
<td>D</td>
<td>Friction drag force</td>
</tr>
<tr>
<td>F&lt;sub&gt;x&lt;/sub&gt;</td>
<td>Total hydrodynamic force in x direction</td>
</tr>
<tr>
<td>F&lt;sub&gt;z&lt;/sub&gt;</td>
<td>Total hydrodynamic force in z direction</td>
</tr>
<tr>
<td>F&lt;sub&gt;θ&lt;/sub&gt;</td>
<td>Total hydrodynamic moment about pitch axis</td>
</tr>
<tr>
<td>f</td>
<td>Two-dimensional hydrodynamic force</td>
</tr>
<tr>
<td>g</td>
<td>Acceleration of gravity</td>
</tr>
<tr>
<td>H</td>
<td>Wave height, crest to trough</td>
</tr>
<tr>
<td>h</td>
<td>Vertical submergence of point below free surface</td>
</tr>
<tr>
<td>h&lt;sub&gt;z&lt;/sub&gt;</td>
<td>Double amplitude of heave</td>
</tr>
<tr>
<td>I</td>
<td>Pitch moment of inertia</td>
</tr>
<tr>
<td>I&lt;sub&gt;a&lt;/sub&gt;</td>
<td>Added pitch, moment of inertia</td>
</tr>
<tr>
<td>k</td>
<td>Wave number</td>
</tr>
<tr>
<td>k&lt;sub&gt;a&lt;/sub&gt;</td>
<td>Two-dimensional added-mass coefficient</td>
</tr>
<tr>
<td>L</td>
<td>Hull length</td>
</tr>
<tr>
<td>LCG</td>
<td>Longitudinal center of gravity, percent of L</td>
</tr>
<tr>
<td>M</td>
<td>Mass of craft</td>
</tr>
<tr>
<td>M&lt;sub&gt;a&lt;/sub&gt;</td>
<td>Added mass of craft</td>
</tr>
</tbody>
</table>
\( m_a \)  
Sectional (two-dimensional) added mass

\( N \)  
Hydrodynamic force normal to baseline

\( r \)  
Wave elevation \( r = r_0 \cos(\omega t) \)

\( r_0 \)  
Wave amplitude

\( U \)  
Relative fluid velocity parallel to baseline

\( V \)  
Relative fluid velocity normal to baseline

\( V/\sqrt{L} \)  
Speed-to-length ratio in knots/ft\(^{1/2}\)

\( W \)  
Weight of craft

\( w_z \)  
Vertical component of wave orbital velocity

\( \dot{w}_z \)  
Vertical component of wave orbital acceleration

\( x \)  
Fixed horizontal coordinate

\( \bar{x} \)  
Vector of state variables

\( \dot{x}_{CG} \)  
Surge velocity

\( \ddot{x}_{CG} \)  
Surge acceleration

\( x_{CG} \)  
Surge displacement

\( z \)  
Fixed vertical coordinate

\( \dot{z}_{CG} \)  
Heave velocity

\( \ddot{z}_{CG} \)  
Heave acceleration

\( z_{CG} \)  
Heave displacement

\( \beta \)  
Deadrise angle

\( \Delta \)  
Hull displacement \( W \)

\( \xi \)  
Body coordinate normal to baseline

\( \lambda \)  
Wavelength

\( \theta \)  
Pitch angle

\( \dot{\theta} \)  
Pitch angular velocity
\( \ddot{\theta} \quad \text{Pitch angular acceleration} \\
\theta_p \quad \text{Double amplitude of pitch} \\
\xi \quad \text{Body coordinate parallel to baseline} \\
\rho \quad \text{Density of water} \\
\omega \quad \text{Wave frequency} \\
\ell \quad \text{Wetted length} \)
ABSTRACT

A nonlinear mathematical model has been formulated of a craft having a constant deadrise angle, planing in regular waves, using a modified low-aspect-ratio or strip theory. It was assumed that the wavelengths would be large in comparison to the craft length and that the wave slopes would be small. The coefficients in the equations of motion were determined by a combination of theoretical and empirical relationships. A simplified version for the case of a craft or model being towed at constant speed was programed for computations on a digital computer, and the results were compared with existing experimental data. Comparison of computed pitch and heave motions and phase angles with corresponding experimental data was remarkably good. Comparison of bow and center of gravity vertical accelerations was fair to good.

ADMINISTRATIVE INFORMATION

This investigation was authorized by the Naval Sea Systems Command with initial funding under Task Area SR-023-0101 and completion under Task Area ZF-43-421001.

INTRODUCTION

Computer programs for estimating the motions of displacement ships in waves for all headings and speeds have been in existence for some time. Comparable computational schemes for planing craft do not exist except in limited and restricted cases. A program for planing craft would be quite useful to the small craft designer, providing a means for systematically exploring the effects of numerous design variations on performance of the craft in waves. With minor modification, the program could also be used to examine the merits of a hybrid craft design, e.g., a combination of planing craft and hydrofoil.

Predicting the motions of a planing craft in waves is by no means a simple problem. The analytical description of a high-speed craft, planing in waves, involves several different types of flow phenomena, including planing; hydrodynamic impact, and, to a lesser extent, surface wave generation and hydrostatics. Also, the mathematics tend to become nonlinear rapidly as the motion increases or, like the real craft, can in some instances exhibit large instabilities such as porpoising.

Development of a computer program that would take into account all of the previously described factors and would be applicable for a wide range of speed and wave conditions requires a careful and systematic study in several stages with appropriate verification at each stage. To lay the foundation for such a general program, a simpler problem has been
formulated in this report with potential for expansion and generalization to the more complicated case. The simpler problem is that of a V-shaped prismatic body with hard chines and constant deadrise planing at high speed in regular head waves.

The mathematical formulation is analogous to low-aspect-ratio wing theory with provisions for including hydrodynamic impact loads, essentially a strip theory. Surface wave generation and forces associated with unsteady circulatory flow are neglected, and the flow is treated as quasi-steady. The mathematical formulation is an empirical synthesis of several theoretically derived flows describing the overall craft hydrodynamics. Wave input is restricted to monochromatic linear deepwater waves with moderate wavelengths and low wave slopes.

**MATHEMATICAL FORMULATION**

**GENERAL**

Consider a fixed coordinate system \((x, z)\) (Figure 1) with \(x\) axis in the undisturbed free surface, pointing in the direction of craft travel, and the \(z\) axis, pointing downward. If the motions of the craft are restricted to pitch \(\theta\), heave \(z_{CG}\), and surge \(x_{CG}\), the equation of motions can be written as

\[
\begin{align*}
M\ddot{x}_{CG} &= T_x - N\sin \theta - D\cos \theta \\
M\ddot{z}_{CG} &= T_z - N\cos \theta + D\sin \theta + W \\
I\ddot{\theta} &= N x_c - D x_d + T x_p
\end{align*}
\]

(1)

where \(M\) is mass of craft

\(I\) is pitch moment of inertia of craft

\(N\) is hydrodynamic normal force

\(D\) is friction drag

\(W\) is weight of craft

\(T_x\) is thrust component in \(x\) direction

\(T_z\) is thrust component in \(z\) direction

\(x_c\) is distance from center of gravity (CG) to center of pressure for normal force

\(x_d\) is distance from CG to center of action for friction drag force

\(x_p\) is moment arm of thrust about CG.

Equation (1) is exact; however, defining the hydrodynamic forces and moments in waves can be extremely difficult.
A high-speed craft moving in waves may transit through several regimes that have different hydrodynamic flow characteristics. For example, as the craft moves away from the crest of wave, the flow may be characterized by unsteady-state planing until the craft collides with the oncoming wave crest and enters another regime in which impact forces are important. After the impact, the craft may enter still another regime in which it is planing but in which buoyancy forces are rather significant.

The most promising approach to a method that would incorporate all three types of flow conditions into a general formulation would seem to be a modified strip theory. The mathematical justification for this approach is not rigorous; however, there is sufficient precedent to expect promising results. For example, impact loads on landing seaplanes can be estimated reasonably well using a strip theory incorporating the Wagner two-dimensional (2-D), expanding-wedge theory,¹ and Chuang² has provided a strip method for determining loads on an impacting prismatic form that agrees extremely well with experimental results.

More recently, Martin³ has developed a linear strip theory for estimating motions of a planing craft at high speed, which shows good agreement with experimental results. A nonlinear model of the equations of motion would be expected to provide, in addition to the motions, reasonable estimates of the vertical accelerations which are an important consideration in designing a planing craft.

TWO-DIMENSIONAL HYDRODYNAMIC FORCE

Implicit with any strip method is the need to define the 2-D hydrodynamic force acting on an arbitrary cross section of the body. The 2-D flow problem is not simple; however, it lends itself to an empirical approach, using a combination of techniques used in hydrodynamic impact and low-aspect-ratio theories.

The typical cross section of a hard-chine, V-shaped prismatic body such as that being considered here is shown in Figure 2. Figure 2 actually illustrates two different idealized-flow conditions, assumed to represent the crossflow during unsteady planing, depending upon whether the flow separates from the chine (Figure 2a) or not (Figure 2b). Nonwetted-chine flow conditions are typical of the sections near the leading edge of the wetted length of the craft. Wetted-chine flow conditions are more typical of sections near the stern, except possibly in the most extreme motion and wave conditions. Some sections between leading edge and stern may alternate between flow conditions as the wetted length changes with the motions.

*A complete listing of references is given on page 33.
The normal hydrodynamic force per unit length \( f \), acting at a section, is treated as quasi-steady and is assumed to contain components proportional to the rate of change of momentum and the velocity squared (drag term), i.e.

\[
f = - \left\{ \frac{D}{Dt} (m_a V) + C_{D,c} \rho b V^2 \right\}
\]

where \( V \) is the velocity in plane of the cross section normal to the baseline

- \( m_a \) is the added mass associated with the section form
- \( C_{D,c} \) is the crossflow drag coefficient
- \( \rho \) is the density of the fluid
- \( b \) is the half beam.

For sections near the leading edge of the wetted length with nonwetted chine, the added mass is assumed to be defined in the same manner as during an impact which for a \( V \)-shaped wedge is given by

\[
m_a = k_a \pi/2 \rho b^2
\]

where \( k_a \) is an added-mass coefficient that may also include a correction for water pileup—\( k_a \) is assumed to be 1.0 without pileup correction.

The rate of change of momentum of the fluid at a section is given by

\[
\frac{D}{Dt} (m_a V) = m_a \dot{V} + V \dot{m}_a - \frac{\partial}{\partial \xi} (m_a V) \frac{d \xi}{dt}
\]

where \( \xi \) is the body coordinate parallel to the baseline; see Figure 1. The last term on the right-hand side of Equation (4) takes into account the variation of the section added mass along the hull. This contribution can be visualized by considering the 2-D flow plane as a substantive surface moving past the body with velocity \( U = -d \xi/dt \) tangent to the baseline. As the surface moves past the body, the section geometry in the moving surface may change with a resultant change in added mass. This term exists even in steady-state conditions and is the lift-producing factor in low-aspect-ratio theory.

The added mass of a section with fully wetted chines has not been developed to the same extent as the \( V \) wedge. In steady-state planing problems such as those of Shuford,\(^4\).
the crossflow is treated as a Helmholtz-type flow in which the Bobyleff results are used for estimating drag coefficients. Helmholtz flows are applicable only to steady-state conditions; so, it is assumed that the added mass for the fully wetted chine flow can be determined from Equation (3) using the value of the half-beam at the chine. In using the Shuford approach, it is assumed that the crossflow drag coefficient for a V-section is equal to the drag of a flat plate \( C_{D,c} = 1.0 \) corrected by the Bobyleff flow coefficient approximated by \( \cos \beta \), i.e.

\[
C_{D,c} = 1.0 \cos \beta
\]  

The Bobyleff flow coefficient is the theoretical ratio of the pressure on a V-section to that experienced by a flat plate for a Helmholtz-type flow.

The same approximation is used for estimating the drag coefficient for nonwetted chine sections, using the instantaneous value of the half-beam at the free surface.

An additional force acting on the body is the buoyancy force \( f_B \). This force is assumed herein to act in the vertical direction and to be equal to the equivalent static buoyancy force multiplied by a correction factor, i.e.

\[
f_B = -a \rho g(A)
\]  

where \( A \) is the cross-sectional area of the section, and \( a \) is a correction factor. The full amount of the static buoyancy is not realized because at planing speeds the water separates from the transom and chines, reducing the pressure at these locations to atmospheric or less than the equivalent hydrostatic pressure. A greater reduction is realized in the buoyancy moment because of the corresponding shift in the center of pressure. Shuford\(^4\) in his work on steady-state planing recommended a factor of one-half to obtain the correct buoyancy force. In the following computations, the buoyancy force was corrected by a factor of one-half, i.e., \( a = 1/2 \). The buoyancy moment, computed as the static buoyancy force multiplied by its corresponding moment arm, was corrected by an additional factor of one-half to obtain the proper mean-trim angles.

Equation (2) is a synthesis of several idealized flow conditions combined in an empirical manner. In all of these flows, it is assumed that the net relative movement of the fluid past the body is in an upward direction. This condition may not always be met in the case of unsteady planing in waves. Closer scrutiny will be required to determine what limitations will be imposed upon the problem as formulated and/or what modifications will be required to improve the formulation.
TOTAL HYDRODYNAMIC FORCE AND MOMENT

The total normal hydrodynamic force acting on the body is obtained by integrating the stripwise, 2-D, hydrodynamic force given by Equations (2) and (6) over the wetted length \( \xi \) of the body. A body coordinate system \((\xi,\zeta)\) with its origin at \( CG \) and the \( \xi \) axis pointing forward parallel to the baseline of the body is defined in Figure 1 to facilitate this integration. The hydrodynamic force acting in the vertical or \( z \) direction of the fixed integral coordinate system is given by

\[
-N \cos \theta = F_z(t) = \int_\delta^\ell f \cos \theta \, d\xi + \int_\delta^\ell g \, d\xi
\]

\[
= -\int_\delta^\ell \left\{ m_a(\xi,t) \dot{V}(\xi,t) + \dot{m}_a(\xi,t)V(\xi,t) \right\} \sin \theta \, d\xi
\]

where the integration is taken over the instantaneous wetted length. Similarly the force \( F_x \) acting in the horizontal or \( x \) direction is given by

\[
F_x = \int_\delta^\ell f \sin \theta \, d\xi
\]

\[
= -\int_\delta^\ell \left\{ m_a(\xi,t) \dot{V}(\xi,t) + \dot{m}_a(\xi,t)V(\xi,t) \right\} \cos \theta \, d\xi
\]

\[
+ u \rho g A d\xi
\]

(7)

Wave forces are obtained by neglecting diffraction and assuming that the wave excitation is caused both by the geometrical properties of the wave, altering the wetted length and draft of the craft, and by the vertical component of the wave orbital velocity at the surface \( w_z \), altering the normal velocity \( V \). The horizontal component of orbital velocity is neglected.
since it is assumed small in comparison with the forward speed $\dot{x}_{CG}$. The velocities $U$ and $V$ may then be written as

$$U = \dot{x}_{CG} \cos \theta - (\dot{x}_{CG} - w_z) \sin \theta$$

$$V = \dot{x}_{CG} \sin \theta - \dot{\theta} \dot{\xi} + (\dot{x}_{CG} - w_z) \cos \theta$$

(9)

The depth of submergence $h$ of the body at any point $P(\xi, \zeta)$ may be determined by

$$h = z_{CG} - \xi \sin \theta + \zeta \cos \theta - r$$

(10)

where $r$ is the instantaneous value of the wave elevation directly above the point.

For regular head waves the wave elevation for a linear deepwater wave is

$$r = r_o \cos k(x + ct)$$

(11)

where $r_o$ is the wave amplitude.

$k$ is the wave number.

$c$ is the wave celerity.

At point $P(\xi, \zeta)$

$$x = x_{CG} + \xi \cos \theta + \zeta \sin \theta$$

(12)

where $x_{CG} = \int_0^t \dot{x}_{CG} \, dt$

The hydrodynamic moment $F_\theta$ about $CG$ is obtained in a similar manner by integrating over the wetted length the product of the normal force per unit length and the corresponding moment arm.
\[
F_\theta = - \int_\xi f(\xi, t) d\xi - \int_\xi f_b \cos \theta d\xi
\]

\[
= \int_\xi \left\{ \dot{m}_a(\xi, t) \dot{V}(\xi, t) + \ddot{m}_a(\xi, t) V(\xi, t) \\
- U(\xi, t) \frac{\partial}{\partial \xi} (m_a(\xi, t) V(\xi, t)) + C_{D, c}(\xi, t) \rho_b(\xi, t) V^2(\xi, t) \\
+ a \rho g A \cos \theta \right\} \xi d\xi
\]

(13)

EQUATIONS OF MOTION, GENERAL

Integrating the first term in Equations (7), (8), and (13) provides hydrodynamic forces and moments proportional to acceleration of the motion. These can be combined with the inertial terms of the rigid body to give the following equation of motion

\[
(M + M_a \sin^2 \theta) \ddot{x}_{CG} + (M_a \sin \theta \cos \theta) \ddot{y}_{CG} - (Q_a \sin \theta) \ddot{\theta}
= T_x + F'_x - D \cos \theta
\]

(14)

\[
(M_a \sin \theta \cos \theta) \ddot{x}_{CG} + (M + M_a \cos^2 \theta) \ddot{y}_{CG} - (Q_a \cos \theta) \ddot{\theta}
= T_z + F'_z + D \sin \theta + W
\]

\[-(Q_a \sin \theta) \ddot{x}_{CG} - (Q_a \cos \theta) \ddot{y}_{CG} + (I + I_a) \ddot{\theta}
= F'_\theta - D x_d + T x_p
\]

where

\[
M_a(t) = \int_\xi m_a(\xi, t) d\xi
\]

\[
Q_a(t) = \int_\xi m_a(\xi, t) \xi d\xi
\]

\[
I_a(t) = \int_\xi m_a(\xi, t) \xi^2 d\xi
\]

\[
F'_x = F_x - \left\{ -(M_a \sin^2 \theta) \ddot{x}_{CG} - (M_a \sin \theta \cos \theta) \ddot{y}_{CG} + (Q_a \sin \theta) \ddot{\theta} \right\}
\]

\[
F'_z = F_z - \{ \text{appropriate acceleration terms} \}
\]

\[
F'_\theta = F_\theta - \{ \text{appropriate acceleration terms} \}
\]

A detailed evaluation of the integral expressions for the hydrodynamic forces and moments is provided in Appendix A.
The solution to Equation (14) is cumbersome; however, it can be accomplished using standard numerical techniques. Introducing the state vector \( \{x_1, x_2, x_3, x_4, x_5, x_6\} \) where
\[
\begin{align*}
  x_1 &= \dot{y}_{CG} \\
  x_2 &= \dot{z}_{CG} \\
  x_3 &= \dot{\theta} \\
  x_4 &= x_{CG} \\
  x_5 &= z_{CG} \\
  x_6 &= \theta
\end{align*}
\]
Equation (14) can be rewritten, using matrix algebra, as
\[
A \ddot{x} = \ddot{g} \tag{15}
\]
so that
\[
\ddot{x} = A^{-1} \ddot{g} \tag{16}
\]
where \( A^{-1} \) is inverse of the inertial matrix \( A \). Equation (16) is now in a form that lends itself to integration by using a numerical method such as the Runge-Kutta-Merson integration routine.

EQUATIONS OF MOTION, SIMPLIFIED FOR CONSTANT SPEED

Assuming that the perturbation velocities in the forward direction are small in comparison to the speed of the craft, the equations of motion may be further simplified by neglecting the perturbations and setting the forward velocity equal to a constant, i.e.
\[
\dot{x}_{CG} = \text{CONSTANT}
\]
If it is also assumed that the thrust and drag forces are small in comparison to the hydrodynamic forces and that they are acting through the center of gravity, the equations of motion may be written as
\[ x_{CG} = 0 \]
\[ (M + M_a \cos^2 \theta) \ddot{x}_{CG} - (Q_a \cos \theta) \dot{u} = F'_z + W \]
\[ -(Q_a \cos \theta) \ddot{z}_{CG} + (1 + I_a) \dot{\theta} = F'_\theta \]

These equations also represent the case of the craft (model) being towed through CG at CONSTANT speed. Based upon the previously described equations of motion, a computer program has been written in FORTRAN language to compute the motions of a prismatic body, planing in regular head waves at high speed. A listing of the program along with the appropriate flow chart is presented in Appendix B. The listing contains reference to thrust and drag terms; however, they have no significance, except to provide a starting point for possible updating of the program to include these terms in the future.

**COMPARISON OF COMPUTED RESULTS WITH EXPERIMENTS**

Computations of pitch and heave motions and heave and bow accelerations were made, using the computer program for comparison with the experimental results of Fridsma. Fridsma tested a series of constant-deadrise models of various lengths in regular waves to define the effects of deadrise, trim, loading, speed, length-to-beam ratio and wave proportions on the added resistance, heave and pitch motions, and impact accelerations at the bow and center of gravity. Figure 3 shows the lines of the prismatic models. The models were towed at CG with a system that permitted freedom in surge. The computer program simulates the model being towed at constant speed with CG at the baseline.

Table 1 presents some characteristics of the model and experimental conditions for which comparisons were made. Most of the comparisons have been made at a speed-to-length ratio \( V/\sqrt{L} \) of 6.0 where the mathematical model is expected to be most representative. A limited comparison has also been made at \( V/\sqrt{L} = 4.0 \); however, no comparison has been made at \( V/\sqrt{L} = 2.0 \). At this speed, the model (or craft) operates in the displacement mode for which the mathematical formulation is not valid.

The average computer run corresponded to 10-second, real-time, model scale; however, only the last 2 seconds were considered free of transient effects. An example of the computer time histories of pitch and heave motions is shown in Figure 4. Although the motions are periodic, they are not perfectly sinusoidal; consequently, in determining phase relationship, the peak, positive-pitch value (bow up) and the peak, negative-heave value (maximum upward position of CG) were used as reference points. There was a difference when the opposite peaks were used.
TABLE 1 – MODEL CHARACTERISTICS AND WAVE CONDITIONS FOR COMPUTATIONS

(Model Length = 114.3 cm (3.75 ft); L/b = 5; CΔ = 0.608)

<table>
<thead>
<tr>
<th>CONFIGURATIONS</th>
<th>SYMBOL</th>
<th>β deg</th>
<th>LCG percent L</th>
<th>Radius of Gyration percent L</th>
<th>V/√L</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>20</td>
<td>59.0</td>
<td>25.1</td>
<td></td>
<td>4.0</td>
</tr>
<tr>
<td>B</td>
<td>20</td>
<td>62.0</td>
<td>25.5</td>
<td></td>
<td>6.0</td>
</tr>
<tr>
<td>J</td>
<td>10</td>
<td>68.0</td>
<td>26.2</td>
<td></td>
<td>6.0</td>
</tr>
<tr>
<td>M</td>
<td>30</td>
<td>60.5</td>
<td>24.8</td>
<td></td>
<td>6.0</td>
</tr>
</tbody>
</table>

WAVE CONDITIONS FOR CONFIGURATION

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th></th>
<th>B</th>
<th></th>
<th>J</th>
<th></th>
<th>M</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>H/b</td>
<td>λ/L</td>
<td>H/b</td>
<td>λ/L</td>
<td>H/b</td>
<td>λ/L</td>
<td>H/b</td>
<td>λ/L</td>
</tr>
<tr>
<td>0.111</td>
<td>1.0</td>
<td>0.111</td>
<td>1.0</td>
<td>0.111</td>
<td>1.0</td>
<td>0.111</td>
<td>1.0</td>
<td>0.111</td>
</tr>
<tr>
<td>0.111</td>
<td>1.5</td>
<td>0.111</td>
<td>1.5</td>
<td>0.111</td>
<td>1.5</td>
<td>0.111</td>
<td>1.5</td>
<td>0.111</td>
</tr>
<tr>
<td>0.111</td>
<td>2.0</td>
<td>0.111</td>
<td>2.0</td>
<td>0.111</td>
<td>2.0</td>
<td>0.111</td>
<td>2.0</td>
<td>0.111</td>
</tr>
<tr>
<td>0.111</td>
<td>3.0</td>
<td>0.111</td>
<td>3.0</td>
<td>0.111</td>
<td>3.0</td>
<td>0.111</td>
<td>3.0</td>
<td>0.111</td>
</tr>
<tr>
<td>0.111</td>
<td>4.0</td>
<td>0.111</td>
<td>4.0</td>
<td>0.111</td>
<td>4.0</td>
<td>0.111</td>
<td>4.0</td>
<td>0.111</td>
</tr>
<tr>
<td>0.111</td>
<td>6.0</td>
<td>0.222</td>
<td>6.0</td>
<td>0.111</td>
<td>6.0</td>
<td>0.111</td>
<td>6.0</td>
<td>0.111</td>
</tr>
<tr>
<td></td>
<td>0.334</td>
<td>4.0</td>
<td>0.111</td>
<td>6.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Corresponding time histories of bow and CG accelerations are shown in Figure 5. The bow acceleration was computed at Station 0. As can be seen in these plots, the impact accelerations ranged in magnitude from cycle to cycle. The maximum impact (or negative value) acceleration computed during the final 2 seconds of run was used in the comparisons with experimental values. In some instances, particularly near resonance, the maximum impact acceleration was more than twice the average impact value.

Figure 6 shows a comparison of variation of computed and experimental pitch and heave motion with wave height for the 20-degree deadrise model in a 15-foot wavelength and for a speed-to-length ratio of 6.0. Figure 7 shows the corresponding impact acceleration at the bow and CG. The computed results closely follow the experimental data, except for CG acceleration at the extreme wave height condition, where the computed value is apparently much lower. Experimental data show that the model was leaving the water at this wave-height condition. The computer model did not leave the water but came very close.
see Figure 8. Figure 8 is a trajectory of the computer model relative to the wave for a selected cycle of motion. The computer model behaves very much as expected. On the left-hand side of the figure, the craft is planing down the crest of the wave and, as it approaches the wave trough, comes very close to leaving the water before slamming and submerging itself deeply into the front of the oncoming wave crest.

Figures 9 through 14 show comparisons of the computed and experimental pitch and heave motions at $V/\sqrt{\lambda} = 6.0$ through a range of wavelengths and at a constant wave height of 2.54 centimeters (1 inch) for deadrise models with 10, 20, and 30 degrees. The data have been plotted with respect to the coefficient $C_\lambda$, defined by Frisenda as $L/\lambda [C_\Delta/(L/2b)^2]^{1/3}$. Note that in our notation, $b$ is the half-beam.

Comparisons of heave and pitch for the 10-degree deadrise model shown in Figures 9 and 10, respectively, show excellent results. The computer model accurately predicts the secondary peaks in the pitch and heave responses at $C_\lambda = 0.19$. At this condition, the physical experimental model rebounds so as to fly over alternate waves. The computer model oscillates at half the wave-encounter frequency and comes close to leaving the water at alternate encounters with the wave. It does not quite leave the water to fly over alternate wave crests; nonetheless, it is a good representation of the actual motion.

The heave and pitch comparison for the 20-degree deadrise model at $V/\sqrt{\lambda} = 6.0$ is also excellent as can be seen in Figures 11 and 12, respectively. No experimental phase data for the condition were reported for $C_\lambda$ greater than 0.072; however, extrapolated results (not shown) are in line with the computed results. The pitch and heave results shown in Figures 13 and 14 for the 30-degree deadrise model are good; however, responses at $C_\lambda = 0.048$ and $C_\lambda = 0.072$ are higher than the experimental results.

For practical considerations a computational scheme for planing boat motions should be valid for a range from approximately $V/\sqrt{\lambda} = 4.0$ to $V/\sqrt{\lambda} = 6.0$. Computations of the motions were made for $V/\sqrt{\lambda} = 4.0$ for the 20-degree deadrise model; see Figures 15 and 16. Again the comparison of the computed heave and pitch response with experimental results is excellent.

Comparisons of the computed and experimental impact accelerations (or largest negative values) are presented in Figures 17 through 20. Figures 17 and 18 show bow and CG accelerations for the 10-degree deadrise model; Figure 19 shows similar results for the 20-degree deadrise model. Figure 20 shows the results for the 30-degree deadrise model. In all cases, the comparison appears to be fair to good. In the shorter wavelengths, $\lambda/L = 1.0$ and $\lambda/L = 1.5$, the computed accelerations are higher than the corresponding experimental values. This is most pronounced for the 10-degree deadrise angle model.
CONCLUSIONS AND RECOMMENDATIONS

A mathematical model of a craft having a constant deadrise angle, planing in regular waves, has been formulated using a modified low-aspect-ratio or strip theory. It was assumed that the wavelengths were long in comparison to the craft length and that the wave slopes were small. The coefficients in the equations of motion were determined by a combination of theoretical and empirical relationships.

A simplified version for the case of a craft or model being towed at constant speed was programmed for computations on a digital computer, and the results were compared with existing experimental data.

The comparison of the computed pitch and heave motions and phase angles with the corresponding experimental data gave remarkably satisfying results. Comparison of the bow and CG accelerations was fair to good.

In summary, the previously described mathematical model appears to be a valid representation of a planing craft in waves for the specific craft geometry and wave conditions considered.

To make the computer program more valuable to the designer the following additional work is recommended:

1. Improve estimates of hydrodynamic coefficients to obtain better acceleration data and to include more complicated ship geometry.
2. Determine added resistance in waves.
3. Include freedom to surge and to add components of propulsion.
4. Extend to the case of irregular waves.

ACKNOWLEDGMENTS

Acknowledgment is given to Dr. Joseph Whalen and Ms. Sue Fowler of Operations Research, Inc., who translated the equations of motion into an operational computer program.
Figure 1 – Coordinate System

Figure 2a – Flow Separation from Chine

Figure 2b – Nonwetted Chine

Figure 2 – Types of Two-Dimensional Flow
Figure 3 - Lines of Prismatic Models
(From Reference 5)
Figure 4 - Sample Time Histories of Computed Pitch and Heave Motions

\[ \beta = 20^\circ \]
\[ V/\sqrt{\ell} = 6.0 \]
\[ \lambda/L = 4.0 \]
\[ r_o = 5.08 \text{ cm (2 in.)} \]
Figure 5 - Sample Time Histories of Computed Accelerations of Bow and Center of Gravity

\[ \beta = 20^\circ \]
\[ V/\sqrt{C} = 6.0 \]
\[ \lambda/L = 4.0 \]
\[ H = 5.08 \text{ cm (2 in.)} \]
Figure 6 - Variation of Pitch and Heave with Wave Height

LEAVING WATER

\( V/\sqrt{T} = 6 \)
\( \lambda/L = 4 \)
\( \beta = 20^\circ \)

\( \triangle \) EXPERIMENTAL (REFERENCE 5)
\( \bigcirc \) COMPUTED

DUAL AMPLITUDE OF HEAVE (m)

DUAL AMPLITUDE OF PITCH MOTION (deg)

WAVE HEIGHT/BEAM
Figure 7 — Variation of Acceleration of Bow and Center of Gravity with Wave Height.
Figure 8 – Trajectory of Computer Model Relative to Wave
Figure 9 - Heave Response for 10-Degree Deadrise Model at $V/\sqrt{L} = 0.0$
Figure 10 – Pitch Response for 10-Degree Deadrise Model at $V/\sqrt{T} = 6.0$
Figure 11 – Heave Response for 20-Degree Deadrise Model at $V\sqrt{L} = 6.0$
Figure 12 - Pitch Response for 20-Degree Deadrise Model at \( V/\sqrt{T} = 6.0 \)
Figure 13 – Heave Response for 30-Degree Deadrise Model at $V/\sqrt{L} = 6.0$
Figure 14 – Pitch Response for 30-Degree Deadrise Model at $V/\sqrt{L} = 6.0$
Figure 15 - Heave Response for 20-Degree Deadrise Model at $V/\sqrt{L} = 4.0$
Figure 16 - Pitch Response for 20-Degree Deadrise Model at $V/\sqrt{L} = 4.0$
Figure 17 - Bow Acceleration for 10-Degree Deadrise Model at $V/\sqrt{L} = 6.0$
Figure 18 – Center of Gravity Acceleration for 10-Degree Deadrise Model at $V/\sqrt{L} = 6.0$
Figure 19 - Bow and Center of Gravity Accelerations for 20-Degree Dendricle Model at $V/\sqrt{\ell} = 4.0$ and $V/\sqrt{\ell} = 6.0$
Figure 20 - Bow and Center of Gravity Accelerations for 30-Degree Deadrise Model at $\sqrt{\frac{V}{T}} = 6.0$
REFERENCES


EVALUATION OF HYDRODYNAMIC FORCE AND MOMENT INTEGRALS

The hydrodynamic force the craft experiences in the vertical direction as derived in the text is:

\[ F_z = -\int Q \left\{ m_a \ddot{V} - U \frac{\partial m_a}{\partial \xi} + \dot{m}_a V + C_D \rho b V^2 \right\} \cos \theta \, d\xi + \int Q a \rho g A \, d\xi \]

where \( U = \dot{x}_{CG} \cos \theta - (\ddot{x} - w_z) \sin \theta \)

and

\( V = \dot{x}_{CG} \sin \theta + (\ddot{x} - w_z) \cos \theta - \dot{\theta} \)

Another force acting in the vertical direction is the weight of the craft.

The first two terms of the integral are evaluated by making the substitutions

\[ \dot{V} = \dot{x}_{CG} \sin \theta - \dot{\theta} \ddot{\xi} + \dot{x}_{CG} \cos \theta - \dot{w}_z \cos \theta \]

\[ + \dot{\theta} (\ddot{x}_{CG} \cos \theta - \ddot{x}_{CG} \sin \theta) + w_z \dot{\theta} \sin \theta \]

\[ \frac{\partial V}{\partial \xi} = -\dot{\theta} - \frac{\partial w_z}{\partial \xi} \cos \theta \]

\[ \frac{\partial U}{\partial \xi} = \frac{\partial w_z}{\partial \xi} \sin \theta \]

\[ \frac{\partial w_z}{\partial t} = \dot{w}_z - U \frac{\partial w_z}{\partial \xi} \]

and noting that

\[ \int Q U V \frac{\partial m_a}{\partial \xi} \, d\xi = -U V m_a \bigg|_{\text{stern}} - \int Q m_a \frac{\partial U V}{\partial \xi} \, d\xi \]

Using the previously described substitutions, the force becomes
\[ F_z = \left\{ -M_a \cos \theta \dot{x}_{CG} - M_a \sin \theta \ddot{x}_{CG} + Q_a \dot{\theta} + M_a \dot{\theta} \left( \dot{x}_{CG} \sin \theta - \ddot{x}_{CG} \cos \theta \right) \right. \]
\[ + \int m_a \frac{d}{dt} \cos \theta \xi d\xi - \int m_a w_2 \sin \theta \xi d\xi \]
\[ - \int m_a V \frac{\partial w_2}{\partial \xi} \sin \theta \xi d\xi + \int m_a U \frac{\partial w_2}{\partial \xi} \cos \theta \xi d\xi \]
\[ - UV m_a \left|_{\text{stem}} \right. - \int V \dot{m}_a \xi d\xi - \rho \int C_{D,c} b V^2 \xi d\xi \right\} \cos \theta \]
\[ + \int a \rho g A d\xi \]

where \( M_a = \int m_a d\xi \)

and

\[ Q_a = \int m_a \xi d\xi \]

This is essentially the form in which the integrals have been computed in the program.

The rate of change of the sectional added mass in the third term of the integral expression is derived by relating it to the rate of change of depth of fluid penetration of the section. The added mass of a section is assumed to be equal to

\[ m_a = k_a \frac{\pi}{2} \rho b^2 \]

for which the time derivative is

\[ \dot{m}_a = k_a \rho \dot{b} b \]

where \( b \) is the instantaneous half-beam of the section, and \( k_a \) is an added-mass coefficient, assumed to be constant. A value of \( k_a = 1.0 \) was used in the computations contained in this report. For sections with constant deadrise, which is an imposed limitation of this work, the half-beam is related to the depth of penetration by

\[ b = d \cot \beta \]
where \( \text{d} \) is depth of penetration, and \( \beta \) is deadrise angle.

Taking into account the effect of water pileup, the effective depth of penetration \( d_e \) is, according to Wagner

\[
d_e = \frac{\pi}{2} \text{d}
\]

and

\[
b = d_e \cot \beta = \frac{\pi}{2} \text{d} \cot \beta
\]

where \( \frac{\pi}{2} \) is the factor by which the wedge immersion is increased by the pileup. Using this expression for the half-beam, the rate of change of sectional added mass becomes

\[
\dot{m}_a = k_a \pi \rho \phi (\frac{\pi}{2} \cot \beta) \dot{d}
\]

This expression is valid for penetration of the section up to the chine. When the immersion exceeds the chine, the sectional added mass is assumed to be constant, i.e.,

\[
m_a = k \frac{\pi}{2} \rho \dot{b}^2_{\text{max}}
\]

\[
\dot{m}_a = 0
\]

where \( b_{\text{max}} \) is the half-beam at chine.

The submergence of a section in terms of the motions is given by

\[
h = z - r
\]

where \( z = z_{\text{CG}} - \xi \sin \theta + \xi \cos \theta \)

\[
r = r_{\text{CG}} \cos \left( k (z_{\text{CG}} + \xi \cos \theta + \xi \sin \theta) + \omega t \right)
\]

For wavelengths which are long in comparison to the draft and for small wave slopes, the immersion of a section measured perpendicular to the baseline is approximately

\[
d \approx \frac{z - r}{\cos \theta - \nu \sin \theta}
\]

where \( \nu \) = wave slope

37
The rate change of submergence $d$ is given by

$$
\dot{d} = \frac{\dot{z} - \dot{\eta}}{\cos \theta - u \sin \theta} + \frac{(z - r)}{(\cos \theta - u \sin \theta)^2} \cdot \frac{\partial (\cos \theta - u \sin \theta)}{\partial t}
$$

Since immersion $(z - r)$ is always small in the valid range of the previously described expression, the relationship can be further simplified to

$$
\dot{d} \approx \frac{\dot{z} - \dot{\eta}}{\cos \theta - u \sin \theta}
$$

and

$$
\dot{m}_a \approx k_a \pi \rho b (\pi/2 \cot \beta) \frac{(\dot{z} - \dot{\eta})}{\cos \theta - u \sin \theta}
$$

The expansion of the integral expression for the hydrodynamic moment in pitch follows the procedure used for the vertical force. The results are summarized as follows

$$
F_\theta = -I_\theta \ddot{\theta} + Q_\theta \cos \theta \ddot{z}_{CG} - Q_\theta \dot{\theta} (\dot{z}_{CG} \sin \theta - \dot{x}_{CG} \cos \theta)
$$

$$
- \int Q \cos \theta \frac{\partial w_x}{\partial \xi} \xi \, d\xi + \int Q \dot{\theta} \sin \theta \, w_z \xi \, d\xi
$$

$$
+ \int Q \nu \dot{m}_a \xi \, d\xi + \int Q \rho C_D b V^2 \xi \, d\xi
$$

$$
+ m_a U V \xi \bigg|_{stern} + \int Q m_a V Ud\xi
$$

$$
+ \int Q m_a V \frac{\partial w_z}{\partial \xi} \sin \theta \xi \, d\xi
$$

$$
- \int Q m_a U \frac{\partial w_z}{\partial \xi} \cos \theta \xi \, d\xi
$$

$$
+ \int a \rho g A \cos \theta \xi \, d\xi
$$

The only additional moments are the buoyancy moments. All other moments are considered to be zero for the specific problem considered in this report.
APPENDIX B
COMPUTER PROGRAM DESCRIPTIONS

OVERVIEW

The equations of motions developed in the previous sections of this report have been solved by means of digital computer programs. Two major programs have been developed: the first (MAIN) solves the equations of motion using the Runge-Kutta-Merson integration algorithm and generates time histories that are stored on the system disk. The second (PLTHSP) generates California Computer Products Company (CALCOMP) pen plots from the disk files. All programs were designed to operate on the Control Data Corporation computer system, located at the David W. Taylor Naval Ship Research and Development Center in Carderock, Md.

Descriptions of input data required to execute the programs, job control cards, and programs follow. Sufficient detail is presented for this appendix to serve as a manual for use and maintenance.

JOB CONTROL CARDS FOR PROGRAM MAIN

Job control cards for program MAIN which computes time histories of the motion variables, are described as follows. If CALCOMP plots are not desired, TAPES need not be cataloged.

Job Control Language Card:

<table>
<thead>
<tr>
<th>Job Card</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Charge Card</td>
<td>Standard facility card</td>
</tr>
<tr>
<td>REQUEST,TAPE9,*PF.</td>
<td>Standard facility card</td>
</tr>
<tr>
<td>REQUEST,TAPE2,*PF.</td>
<td>Reserves space for CALCOMP plot data</td>
</tr>
<tr>
<td>REQUEST,TAPE4,*PF.</td>
<td>Print output file 1 request</td>
</tr>
<tr>
<td>ATTACH,BINAR,SEFZARNICKNEWB, ID=XXXX.</td>
<td>Print output file 2 request</td>
</tr>
<tr>
<td>ATTACH,NSRDC.</td>
<td>Attaches binary run file</td>
</tr>
<tr>
<td>LDSET(LIB=NSRDC).</td>
<td>Attaches library routines</td>
</tr>
<tr>
<td>BINAR.</td>
<td>Loads library routines</td>
</tr>
<tr>
<td>REWIND,TAPE2.</td>
<td>Loads and executes run file</td>
</tr>
<tr>
<td>REWIND,TAPE4.</td>
<td>Rewinds time-history files for printing</td>
</tr>
<tr>
<td>COPY(TAPE2,OUTPUT)</td>
<td>Prints time-history file</td>
</tr>
<tr>
<td>COPY(TAPE4,OUTPUT)</td>
<td>Prints time-history file</td>
</tr>
</tbody>
</table>
Job Control Language Card:

<table>
<thead>
<tr>
<th>Comment</th>
<th>Catalogues file for plot.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATALOG, TAPE9, SEFZARNICKDATA...</td>
<td>(SEFZARNICKDATA CAN BE ANY NAME)</td>
</tr>
<tr>
<td>ID=XXXX.</td>
<td></td>
</tr>
<tr>
<td>7/8/9 END OF RECORD</td>
<td></td>
</tr>
<tr>
<td>DATA CARDS (1-5)</td>
<td></td>
</tr>
<tr>
<td>6/7/8/9 END OF FILE</td>
<td></td>
</tr>
</tbody>
</table>

**INPUT DATA CARDS FOR PROGRAM MAIN**

Input data used by program MAIN are read from data cards in NAMELIST and in standard format. A description of the FORTRAN symbols appearing in NAMELIST follows. For simplicity in the text that follows, it is assumed that NAMELIST input occupies only one card. More cards can be used if necessary.

<table>
<thead>
<tr>
<th>Card 1 (NAMELIST FORMAT, / / )</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A</strong></td>
</tr>
<tr>
<td>The absolute error for KUTMER (six values)</td>
</tr>
<tr>
<td><strong>NPRINT</strong></td>
</tr>
<tr>
<td>If = 1, print normal output</td>
</tr>
<tr>
<td>If = 2, matrix, inverse matrix, F-column matrix, and KUTMER results</td>
</tr>
<tr>
<td>If = 3, integral results</td>
</tr>
<tr>
<td>If = 4, calculated values constant for given input values</td>
</tr>
<tr>
<td><strong>NLOT</strong></td>
</tr>
<tr>
<td>If = 0, no plot</td>
</tr>
<tr>
<td>If = 1, printer plot of results</td>
</tr>
<tr>
<td><strong>END</strong></td>
</tr>
<tr>
<td>Number of runs to be made</td>
</tr>
<tr>
<td><strong>W</strong></td>
</tr>
<tr>
<td>Weight of craft in pounds</td>
</tr>
<tr>
<td><strong>BL</strong></td>
</tr>
<tr>
<td>Boat length in feet</td>
</tr>
<tr>
<td><strong>TZ</strong></td>
</tr>
<tr>
<td>Thrust component in z direction</td>
</tr>
<tr>
<td><strong>TX</strong></td>
</tr>
<tr>
<td>Thrust component in x direction</td>
</tr>
<tr>
<td><strong>XECG</strong></td>
</tr>
<tr>
<td>Distance from center of gravity to center of pressure for drag force in feet</td>
</tr>
<tr>
<td><strong>XP</strong></td>
</tr>
<tr>
<td>Moment arm of propeller thrust</td>
</tr>
<tr>
<td><strong>XD</strong></td>
</tr>
<tr>
<td>Distance from center of gravity to center</td>
</tr>
<tr>
<td><strong>DRAG</strong></td>
</tr>
<tr>
<td>Friction for drag force</td>
</tr>
<tr>
<td><strong>RO</strong></td>
</tr>
<tr>
<td>Wave height</td>
</tr>
<tr>
<td><strong>LAMBDA</strong></td>
</tr>
<tr>
<td>Wavelength</td>
</tr>
<tr>
<td><strong>RG</strong></td>
</tr>
<tr>
<td>Radius of gyration in feet</td>
</tr>
<tr>
<td><strong>T</strong></td>
</tr>
<tr>
<td>Propeller thrust in pounds</td>
</tr>
<tr>
<td><strong>GAMMA</strong></td>
</tr>
<tr>
<td>Propeller thrust angle in degrees</td>
</tr>
</tbody>
</table>
Card 1 (continued)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECG</td>
<td>Longitudinal center of gravity</td>
</tr>
<tr>
<td>NCG</td>
<td>Vertical center of gravity, nondimensionalized by ship length</td>
</tr>
<tr>
<td>KAR</td>
<td>Added-mass coefficient</td>
</tr>
<tr>
<td>BETA(I)</td>
<td>Dead-rise angle in degrees</td>
</tr>
<tr>
<td>EST(I)</td>
<td>Station position in feet</td>
</tr>
<tr>
<td>NUM</td>
<td>Number of stations</td>
</tr>
<tr>
<td>XA</td>
<td>Initial time</td>
</tr>
<tr>
<td>XE</td>
<td>Stop time</td>
</tr>
<tr>
<td>HMIN</td>
<td>Minimum step size</td>
</tr>
<tr>
<td>HMAX</td>
<td>Maximum step size</td>
</tr>
<tr>
<td>EPS</td>
<td>Error criterion</td>
</tr>
</tbody>
</table>

Card 2 (Format 8F10.0)

\[(X(I), I=1,6)\] Initial conditions

\[X(1)\] Velocity
\[X(2)\] Z
\[X(3)\] 0
\[X(4)\] X
\[X(5)\] Z
\[X(6)\] 0 degrees

Card 3 (8F10.0)

START     Time to turn on (RMP) function (see page 48)
RISE      Duration of RMP

Card 4 (8F10.0)

TME        Time at which integration interval is to be changed*
HMX        New maximum interval size after TME
HMN        New minimum interval size for KUTMER to subdivide

*If this option is not used set TME to stop time on run.
Card 5 (8F10.0)
PERCNT     Percentage of boat length subtracted from longitudinal center of gravity to obtain X - point where acceleration computations are made

JOB CONTROL CARDS FOR PROGRAM PLTHSP

Job control cards for program PLTHSP which generates CALCOMP plots of time histories computed by program MAIN are described in this section.

Job Control Language Card:

<table>
<thead>
<tr>
<th>Job Card</th>
<th>Standard facility card</th>
</tr>
</thead>
<tbody>
<tr>
<td>REQUEST.TAPE7,H1.</td>
<td>Tape for CALCOMP plot data</td>
</tr>
<tr>
<td>VSN(TAPE7=CK0323).</td>
<td>Volume serial number of tape for CALCOMP plot</td>
</tr>
<tr>
<td>ATTACH,CALC936.</td>
<td>Attaches CALCOMP library routine</td>
</tr>
<tr>
<td>ATTACH,BINAR,SEFZARNICKPLOTB, ID=XXXX.</td>
<td>Attaches plot program run file</td>
</tr>
<tr>
<td>LDSET(LIB=CALC936)</td>
<td>Loads CALCOMP library routines</td>
</tr>
<tr>
<td>BINAR.</td>
<td>Runs plot program</td>
</tr>
</tbody>
</table>

7/8/9 END OF RECORD
DATA CARDS
6/7/8/9 END OF FILE

INPUT DATA CARDS FOR PROGRAM PLTHSP

Two or three data cards are made ready by PLTHSP, depending on the options selected. Standard input format is employed. A description of the necessary data cards follows.

Card 1 (8F10.0 Format)

<table>
<thead>
<tr>
<th>Card 1 (8F10.0 Format)</th>
</tr>
</thead>
<tbody>
<tr>
<td>XAXIS</td>
</tr>
<tr>
<td>YAXISP</td>
</tr>
<tr>
<td>YAXISH</td>
</tr>
<tr>
<td>HT</td>
</tr>
</tbody>
</table>

Card 2 (I10 Format)

<table>
<thead>
<tr>
<th>Card 2 (I10 Format)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IA</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

42
Card 3 (8F10.0 Format) - Only Necessary If IA = 1.

YAXISB    Height of bow acceleration axis in inches
YAXISC    Height of CG acceleration axis in inches

PROGRAM MAIN

Program MAIN reads all necessary input data from cards, sets up initial values, computes constants, calls KUTMER to determine the state variables at TIME for the period from XA to XE in increments of HMAX. A table state variables is created for every PTIME-th value. The values for λ/H and θp/2πH/λ are calculated and printed. If the plot option is on, a printer plot will be produced.

Subroutine COMPUT(X)

This routine computes pitch moment NL and lift force FL, excluding added mass terms, using values of integrals computed in subroutine FUNCT. The argument X contains the state vector.

Subroutine DAUX

This subroutine is called from KUTMER or EULER. It determines the values of m_a, b, and b^1, based on the following equations

\[ h_w(l) = z_{CG} - \xi(l) \sin \theta + \xi(l) \cos \theta - r(l) \]

where \( r(l) = r_0 \cos k \left[ x_{CG} + \xi(l) \cos \theta + \xi(l) \sin \theta + \omega t \right] \)

Then for

\[ h_w(l) > 0, \]

\[ d(l) = \frac{h_w(l)}{\cos \theta - (l) \sin \theta} \]

where \( V(l) = -r_0 k \sin \theta \left[ x_{CG} + \xi(l) \cos \theta + (l) \sin \theta + \omega t \right] \)

If

\[ d(l) \geq h_m(l) \tan (\beta(l) 2/\pi) \]

43
set

\[ m_a(l) = m_{a\text{max}}(l) \]
\[ b(l) = b_m(l) \]
\[ b_1(l) = 0 \]
\[ m_{a\text{max}}(l) = k(l)(\rho/2)\pi b_m^2(l) \]

If

\[ d(l) < b_m(l) \tan(\beta(l))(2/\pi) \]

set

\[ b(l) = d(l) \cot(\beta(l))(\pi/2) \]
\[ b_1(l) = b(l) \]
\[ m_a(l) = k_a(l)(\rho/2)\pi b^2(l) \]

for

\[ h_w(l) < 0; \]
\[ m_a(l) = 0, \quad b(l) = 0, \quad b_1(l) = 0 \]

This subroutine then calls FUNCT which in turn calls COMPUT to determine the values of \( N_L \) and \( F_L \), the lift force and moment. The values of \( N_L \) and \( F_L \) are used to compute the following:

\[ F_1 = T_x + F_L \sin \theta - D \cos \theta \]
\[ F_2 = T_z + F_L \cos \theta + D \sin \theta + W \]
\[ F_3 = N_L - D_{x_d} + T_{x_p} \]

---

*The b1 array is set up for integrations for portion of hull for which chine is not immersed.*
The mass inertia matrix is

\[
\begin{align*}
A_{11} & = M + M_\theta \sin^2 \theta \\
A_{12} & = M_\theta \sin \theta \cos \theta \\
A_{13} & = -Q_\theta \sin \theta \\
A_{21} & = A_{12} \\
A_{22} & = M + M_\theta \cos^2 \theta \\
A_{23} & = -Q_\theta \cos \theta \\
A_{31} & = A_{13} \\
A_{32} & = A_{23} \\
A_{33} & = I + I_\theta
\end{align*}
\]

The matrix is inverted by the system routine MATINS. The inverted matrix is then used to solve the following equations which determine the state vectors.

\[
\begin{align*}
\ddot{x}_{CG} & = A_{11}^{-1} F_1 + A_{12}^{-1} F_2 + A_{13}^{-1} F_3 \\
\ddot{z}_{CG} & = A_{21}^{-1} F_1 + A_{22}^{-1} F_2 + A_{23}^{-1} F_3 \\
\dot{\theta} & = A_{31}^{-1} F_1 + A_{32}^{-1} F_2 + A_{33}^{-1} F_3
\end{align*}
\]

Subroutine FUNCT (X)

This routine evaluates various integrals appearing in the force and moment mathematical models. The integrals are evaluated, using a trapezoidal integration algorithm. The argument X contains the state vector. A list of integrals that are evaluated is presented.
\[ \int q m_a \xi d\xi \quad \int q m_a \xi^2 d\xi \quad \int q m_a \xi^3 d\xi \quad \int q m_a U V d\xi \]

\[ \int q m_a w_2 d\xi \quad \int q m_a w_x d\xi \quad \int q m_a \frac{dw_x}{dt} d\xi \quad \int q m_a \frac{dw_x}{dt} \xi d\xi \]

\[ \int q m_a V \frac{\partial w_x}{\partial \xi} d\xi \quad \int q m_a V \frac{\partial w_x}{\partial \xi} \xi d\xi \quad \int q m_a U \frac{\partial w_x}{\partial \xi} d\xi \quad \int q m_a U \frac{\partial w_x}{\partial \xi} \xi d\xi \]

\[ \int q m_a V d\xi \quad \int q m_a V \xi d\xi \quad \int b V^2 d\xi \quad \int b V^2 \xi d\xi \]

\[ \int b \left( h - \frac{b}{2} \tan \beta \right) d\xi \quad \int b \left( h - \frac{b}{2} \tan \beta \right) \xi d\xi \]

**Subroutine INPUT**

This routine reads in NAMELIST/HSP/ which contains the initial data concerning the craft and sea conditions pertinent to all the runs to be made. It is set up so that most of the data are given default values by means of data statements in subroutine INPUT. These data statements can be overridden during execution by reading values in on cards. For further explanation of the specific variables see section on the input data cards.

This routine also "initializes" constant such as \( \pi, \rho, \) and \( g. \) It uses the input values to calculate the keel profile and planform arrays, NO and BM, wave constants, system mass and inertia, and maximum mass and depth of chine at each station.

**Subroutine KUTMER (NEQS, TIME, HMAX, X, EPSE, A, HMIN, FIRST)**

This is a Runge-Kutta-Merson integration routine that is capable of changing the size of the interval over which it integrates to meet specified error criteria. It is therefore an
accurate method for a system that may oscillate more rapidly than the initial integration interval. A minimum step size prevents the routine from subdividing the interval indefinitely.

The input arguments are:

- **NEQS**: Number of dependent variables in the x array
- **TIME**: Actual time (independent variable)
- **HMAX**: Increment for which the solution is to be returned
- **X**: Vector of dependent variables
- **EP3E**: Relative error criteria specified for each component of x and used for the components of x less than the absolute value of A
- **A**: Absolute error criteria
- **HMIN**: Minimum step size allowed
- **FIRST**: Set to zero on first call; a value of 1 is assigned by KUTMER on subsequent calls for which the error criteria are satisfied, otherwise a value of 2 is assigned

**Subroutine PLOT2 (F, FMIN, FMAX, NVAR, NFUN, N1, N, XO, DELX)**

Data stored in the two-dimensional array F are plotted, using the printer by subroutine PLOT2. As many as 26 different functions, having evenly spaced abscissa values, can be plotted. The output is written on Unit 6. A description of variables follows.

- **F**: Array containing data to be plotted; the Jth point of the Ith function is stored in F(I,J)
- **FMIN**: An array of minimum functional values; the minimum of the Ith function is stored in FMIN(I)
- **FMAX**: Same as FMIN only for maximum values
- **NVAR**: An array of titles for each function to be plotted
- **NFUN**: Number of functions to be plotted
- **N1**: First dimension of array F
- **N**: Number of points to be plotted
- **XO**: First abscissa value
- **DELX**: Abscissa increment

**Subroutine PLOTER (FX, XA, HMAX, LAMBDA, IB, NAVE)**

The routine initializes various values required to generate printer plots and computes pitch-and-heave ratios. The printer plots that are generated consist of pitch-and-heave time histories. A description of input variables follows.
FX  A two-dimensional array, containing time histories to be plotted  
XA  Initial time  
HMAX Time-interval increment; time interval between values in FX is given by 
       HMAX*PTIME  
LAMBDA Wavelength  
IB  Number of values to be plotted  
N WAVE Position in FX at which wave is completely turned on  

Function RMP (T, START, RISE)  
The RMP is a function that calculates a value between 0 and 1 corresponding to time T, 

   [91x544]based on a straight line from time START with a value of 0 to time START plus RISE with 
   [91x544]a value of 1. It is used to lower the initial wave amplitude to avoid large transients at start 
   [91x544]of the computations. 

   The arguments are:  
   T  Actual time  
   START Time at which to begin the ramp from 0 to 1  
   RISE Duration of rise from 0 to 1  

The function reaches the value 1 at time START plus RISE, if the rise is 0.0, RMP will 
return a value of 0.5.  

Subroutine TRAP (F, DX, NPTS, ANS)  

This routine performs the evaluation of an integral using a trapezoidal approximation. 

The argument variables are defined as follows:  
F  Array of integrand values  
DX  Increments at which F is evaluated  
NPTS Number of values in F  
ANS Result, which is equal to  

\[
\text{DX} \left\{ \sum_{i=1}^{\text{NPTS}} F(i) - 0.5 \{ F(1) + F(\text{NPTS}) \} \right\}
\]

PROGRAM PLTHSP  

This program uses a data file created by program MAIN to create CALCOMP plots. The 
data are read from logical Unit 9 and are rewritten on Unit 7 for CALCOMP input. Program 
PLTHSP sets the tape output unit equal to 7 and calls SUBROUTINE CALPHI to execute the 
plot procedures.
Subroutine CALPLT

This subroutine manages all the I/O operations and performs the necessary calculations required to generate the plots. After reading the card data (two or three cards) subroutine READT is called to read the data file (Tape 9) created by program MAIN. The CALCOMP initializing routines are called next, after which a call to subroutine ESCALE calculates the necessary scaling factors. Subroutine EXAXIS is called next to determine the placement of the plot tick marks and identifying digits. The CALCOMP plot-generation subroutines are now called and, depending on the option defined by the IA parameter on card 2, plots of pitch and heave at the bow and CG location are generated as functions of time if IA = 1.

Subroutine EAXIS

The subroutine is analogous to the CALCOMP AXIS routine. The only exception is that the tick marks are not necessarily inch, and the height of the characters is defined by the input parameter HT. Function NDIGIT is called to determine the number of digits necessary to print an even increment of the plots functions on the axis.

Subroutine ESCALE, ADJUST, and FUNCTION UNIT

These subroutines find the scale to be used on the plot axis. Function UNIT is called to determine the axis increment size after which subroutine ADJUST is called to extend the minimum (AMIN) and maximum (AMAX) values so that they are even multiples of the axis increments.

FUNCTION NDIGIT

This function finds the number of digits necessary to print even increments of the function on the axis. Both the number of places in the entire number (NDIGIT) and the number of decimal places (ND) are determined, after which the value of each increment on the axis (ANUM) is calculated.

Subroutine READT

This subroutine reads the data file created by program MAIN. Data file records are read until the message end of file is encountered. Each record is read in the same format as it was written in MAIN. The information is printed to allow the user to inspect the created file.
LISTING OF COMPUTER PROGRAM FOR MOTION COMPUTATIONS

PROGRAM MAIN(INPUT, OUTPUT, TAPE5=INPUT, TAPE6=OUTPUT, TAPE3=S12, TAPE2=S12, TAPE4=S12, TAPE9)

C REAL I, K, LAMBDA, M, MA, MMAX, N, NCG, NU, MASS, NL, IA, KAR
C INTEGER END
C DIMENSION X(6), F(2, 400)
C COMMON /CONST/ NCG, ECG, PI, OPR, RPD, GRAVITY, RHO, K, NUM, MA(120), CO, TA,
C B(120), BETA, MM(120), T2, URAG, XX, TAP, M, IT,
C DELTAS, TX, EST(120), CR, RO, KAR, MMAX(10), TEST(120),
C N(120), PHALF
C COMMON /SHIP/ MASS, CINT, GA, CE, CE2, CE3, DMU, E2DMU, E3DMU, BF, BHM

C CALL INPUT

C COMPUTE INTEGRATION INTERVAL INFORMATION

NLESS = NUM - 1
I = 1
II = 1
DIFFER = EST(I+1) - EST(I)
KTT(I) = I
DIFF(I) = DIFFER
GO 25 IF = NLESS
DIFFER = EST(I+1) - EST(I)
KTT(I) = KTT(I) + 1
IF (DIFFER < NLESS) GO TO 24
GO TO 25
24 II = II + 1
KTT(I) = KTT(I) + 1
DIFF(I) = DIFFER
CONTINUE

C * * * CHECK IF NUMBER OF INTERVALS EXCEEDS DIMENSION
IF (II - 0T 10) WRITE(6, 28) (KTT(I) + DIFF(I), I = 1, II)
IF (II0T 10) STOP 4
CONTINUE

TIME = xx
COUNT = 1
END = END - 1
WRITE(6, 39)

C * * * POINT AT WHICH MULTIPLE RUNS START

39 FORMAT(111)
C * * * READ IN INITIAL CONDITIONS
X(1) = VELOCITY, A(2) = 0 DOT, X(3) = THETA DOT
X(4) = X, X(5) = 2, X(6) = THETA
THETA IS READ IN DEGREES THEN CONVERTED TO RADIANS IN PROGRAM
READ(S, 10) (X(I), I = 1, 6)

C
DATA, USED IN RAMP FUNCTION, TO TURN ON WAVE
READ (5, 10) START, RISE

10 FORMAT (8F10.4)
C * * * * * * WRITE OUT THE INPUT VALUES
WRITE (6, 10) START, RISE, KAR
19 FORMAT (* START = ',F10.4,' RISE = ',F10.4,' KAR = ',F10.4)

THE IS THE TIME AT WHICH THE INTEGRATION INTERVAL IS TO BE CHANGED
MAX IS THE NEW MAXIMUM INTERVAL SIZE AFTER TIME THE MIN IS THE NEW MINIMUM INTERVAL SIZE FOR RADIUS TO SUB-DIVIDE THE MAXIMUM INTERVAL UP TO IF THIS OPTION IS NOT USED SET THE TO THE STOP TIME OF THE RUN

READ (5, 10) TIME, MM, MMN
WRITE (6, 11) TIME, MM, MMN, MMN

ADJUST THE TIME FOR CHANGE OF INTEGRATION INTERVAL FOR CHECK AGAINST TIME IN THE INTEGRATION LOOP
TH = TIME - (MMAX / 2.)

SET SWITCH FOR CALCULATION OF PITCH AND HEAVE RATIOES ON NEXT CALL TO PLOTTER
IPT = 0
IF TIME .LE. ЕХ ЕХ ЕХ ЕХ = 1 ИТЕ

READ (5, 10) PERCENT
KACCL = ЕСН - PERCENT * 0.86666666
WRITE (6, 12) PERCENT, XACCL
12 FORMAT (* THE X VALUE USED FOR THE BOW AND CG ACCELERATION COMPUTATIONS * IS EQUAL TO ЕСН = ,F10.4, TH*0.8666 OR =F10.4)

WRITE (6, 23)
WRITE (6, 47)
23 FORMAT (* STATION NO. *, 3X, "DEAUX RISE", 8X, "EST", 8X, "NO.", * 10X, "HEAM")
WRITE (6, 55) (1), BETAXEST (I), NUM(I), BM(I), I = 1, NUM
55 FORMAT (6X, 12, 5X, F10.4, 4X, F10.4, 4X, F10.4, 4X, F10.4, 3X, F10.4)
WRITE (6, 23)
WRITE (6, 56) (1), I = 1, 6
56 FORMAT (* X VALUES", 4X, 6(F10.4, 2X))

CHANGE INPUT FROM DEGREES TO RADIANS
X(3) = X(3) * RPD
X(6) = X(6) * RPD

WAVE = START - RISE
WAVE = 0

WRITE OUT COMPUTED ARRAYS
WRITE (6, 57) N, I, K, C, PHASE, P, 1, GRAVITY
IF (NPRINT .LT. 4) GO TO 62
WRITE (6, 58) (1), I = 1, NUM
WRITE (6, 59) (K), I = 1, NUM
WRITE (6, 64) (MMAX(I), I = 1, NUM)
WRITE (6, 64) (DIST(I), I = 1, NUM)
62 CONTINUE
WRITE(6,28) (KTT(I), DIFF(I), I=1,11)
28 FORMAT(*
KTT,DIFF,110,2X,F10.4)
57 FORMAT(4H MA =, F10.4, 4H IN =, F10.4, 4H KB =, F10.4, 10H GRAVITY=, F10.4)
58 FORMAT(16H F(I) =, 10F10.4)
59 FORMAT(16H N(I) =, 10F10.4)
64 FORMAT(16H MMAX(I) =, 10F10.4)
68 FORMAT(16H TEST(I) =, 10F10.4)

IB = 1
IPRINT = NPRINT
WRITE(4,91)
91 FORMAT(16H TIME","9X","DOT",9X","2DOT",9X","THETA DOT",6X,
6H MX,9X,1H2,9X,5HMTETA,9X,2HNL,9X,9FL
6H "X",9X,1H2,9X,5HACCL,6X,"HCO ACCL"/)
WRITE(4,92) TIME,(X(I),I=1,6),NL,FL,BWACL,COACL
WRITE(9) TIME,(X(I),I=1,6),BWACL,COACL
KOUNT = KOUNT+1
FX(1) = X(5)
FX(2) = X(6)
IKUTME = XE = XA/MMAX + 0.5
IKUTM = (TIME - XA)/MMAX + (XE - TIME)/MMAX + 0.5
FIRST = 0.0
NEG = 6
IKUTS = 0

C C START OF INTEGRATION LOOP
C
851 CONTINUE
NPRINT = IPRINT
C ** CHECK PITCH, GT. 3236 RADIANS
IF(X(6) GT 3236) GO TO 853
C ** PERFORM INTEGRATIONS
IF(TIME.LT.TM OR THE.EQ.XE) GO TO 98
IF(IPT.EQ.1) GO TO 98
HMIN = HMIN
MMAX = MMAX
FIRST = 0.0
98 CONTINUE
CALL KUTME(NEG,TIME,MMAX,X,EPSA,A,HMIN,FIRST)
IKUTS = IKUTS+1
IF(FIRST.EQ.2) GO TO 861
IF(KOUNT.NF.1 AND KOUNT.NE.41) GO TO 99
WRITE(4,91)
KOUNT=1

C ** WRITE OUT TIME INTERVAL RESULTS
99 WRITE(4,92) TIME,(X(I),I=1,6),NL,FL,BWACL,COACL
WRITE(6,93) T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12
WRITE(9) TIME,(X(I),I=1,6),BWACL,COACL
IF(TIME.LT.TM OR THE.EQ.XE) GO TO 200
IF(IPT.EQ.1) GO TO 200
CALL PLUTE((FX,XA,MMAX,LAMBDA,IB,NWAVE,IPT)
IPT = 1
I7 = 0
XA = TIME
FIRST = 0.0
HMIN = HMIN
MMAX = MMAX
93 CONTINUE
   IF(NWAVE,GT,2100 TO 21
   IF(TIME,GT,WAVE)NWAVE=KOUNT

21 CONTINUE
   IF(TIME.LE.XE.AND.IKUTS.LT.IKUTM)GU TO 851
   WRITE(2,852)
854 CONTINUE
852 FORMAT( " END OF KUTMER"
853 CONTINUE
   CALL PLUTEI(F~,FMAMNVARNFUNNMODgLX)
   C ** Call Plutei(F~,FMAMNVARNFUNNMODgLX)
   C PLOT FIRST N POINTS OF UP TO 26 FUNCTIONS F(X)
   C F(I,J) CONTAINS THE VALUE FOR THE JTH POINT OF THE ITH FUNCTION
   C FMIN(I) AND FMAX(I) CONTAIN THE MIN AND MAX ORDINATE VALUES FOR
   C THE ITH FUNCTION.
   C NVAR(I) AN ARRAY OF TITLES FOR THE VARIOUS FUNCTIONS
   C TO BE PLOTTED AGAINST THE ABSCISSA
   C NFUN NUMBER OF FUNCTIONS TO BE PLOTTED - DIMENSION OF
   C NVAR, FMIN, FMAX
   C N1 USED ONLY IN (N1,1) AS PASSED DIMENSION
   C N NUMBER OF POINTS IN A SINGLE PLOT FRAME
   C XO FIRST ABSCISSA VALUE
   C DELX ABSCISSA INCREMENT
   C DIMENSION 2*STEP(26),F(N1,1),FMIN(NFUN),FMAX(NFUN),VLAST(26),
   C VF=STEP(26),MEAD(6),STEP(26)
   C INTEGER CH(NFUN),NVAR(NFUN),DOT,ASTER,PLUS,BLANK
   C INTEGER C
   C INTEGER A(101)
   C DATA BLANK,DOT,ASTER,PLUS/IM,*IM,*,IM,*/
   DATA CH(1),CH(2),CH(3),CH(4),CH(5),CH(6),CH(7),CH(8),CH(9),CH(10)
   DATA CH(11),CH(12),CH(13),CH(14),CH(15),CH(16),CH(17),CH(18)
   2 / IM, IM, IM, IM, IM, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1.00/
   DATA CH(11),CH(12),CH(13),CH(14),CH(15),CH(16),CH(17),CH(18)
   2 / IM, IM, IM, IM, IM, IM, IM, IM/
   DATA CH(19),CH(20),CH(21),CH(22),CH(23),CH(24),CH(25),CH(26)
C IF (NFUN.LE.0.0.0.N.0.0) RETURN

C PRINT HEADINGS.
WRITE(*,*)
46 FORMAT (//)
GO TO 40 IF (NFUN)
30 TERM=AUX(FMAX(I)-FMIN(I))
EXP=1.
IF (TERM.EQ.0.0) GO TO 2
C BRING TERM TO A VALUE BETWEEN 1 AND 10
IF (TERM.LT.1.0) GO TO 1
3 IF (TERM.LT.10.0) GO TO 2
EXP=EXP*1.0
TERM=TERM/10.0
GO TO 3
1 EXP=EXP*10.0
IF (TERM.GT.10.0) GO TO 2
GO TO 1
C SET UP VALUE BETWEEN GRID LINES, RSTEP.
2 RSTEP=5.
IF (TERM.GE.5.0) RSTEP=10.
IF (TERM.LT.5.0) RSTEP=2.
5 RSTEP(I)=STEP*EXP(I)
C COMPUTE VALUE OF STARTING LINE, VFIRST.
FIRST=FMIN(I)/RSTEP(I)
IF (FMIN(I).LT.0.0) FIRST=FIRST-1.
FIRST=INT(FIRST)
VFIRST(I)=FIRST*RSTEP(I)
C CHECK END LINE VALUE, VLAST.
VLAST(I)=VFIRST(I)*10.0*RSTEP(I)
IF (VLAST(I).LT.FMAX(I)) GO TO 4
C IF GRAPH IS TOO SMALL TAKE NEXT LARGER STEP.
AA=STEP(I)
IF (AA.LT.5.0) AA=AA/5.
IF (AA.GE.5.0) AA=AA/10.
IF (AA.LT.10.0) AA=AA/10.
PSTEP(I)=AA
EXP=AA
GO TO 5
C COMPUTE VALUE BETWEEN POINTS, STEP
4 STEP(I)=RSTEP(I)*.1
RK=0.
GO 6 RK=1.0
HEAT(KK)=FIRST(I)*2.0*RK*RSTEP(I)
6 WRITE(*,*) CM(I)+NVAR(I), (HEAT(KK),KK=1,6)
45 FORMAT (1X,A1,1H = ,A10,5X,1PE12,4,5(8X,1PE12,4))
GO 50 J=1,101
AJ=BLANK
IF (MOD(J,11)).EQ.1) A(J)=OUT
50 CONTINUE
WRITE(6,55) A,A
55 FORMAT (25X,10I1/15X,4MTIME,6X,10I1)
C PLOT EACH POINT
DO 100 J=1.N
B=EXP(FLAT(J-I)*DELX)
100 Y(J,K)=B
C PLOT2 29
C PLOT2 30
C PLOT2 31
C PLOT2 32
C PLOT2 33
C PLOT2 34
C PLOT2 35
C PLOT2 36
C PLOT2 37
C PLOT2 38
C PLOT2 39
C PLOT2 40
C PLOT2 41
C PLOT2 42
C PLOT2 43
C PLOT2 44
C PLOT2 45
C PLOT2 46
C PLOT2 47
C PLOT2 48
C PLOT2 49
C PLOT2 50
C PLOT2 51
C PLOT2 52
C PLOT2 53
C PLOT2 54
C PLOT2 55
C PLOT2 56
C PLOT2 57
C PLOT2 58
C PLOT2 59
C PLOT2 60
C PLOT2 61
C PLOT2 62
C PLOT2 63
C PLOT2 64
C PLOT2 65
C PLOT2 66
C PLOT2 67
C PLOT2 68
C PLOT2 69
C PLOT2 70
C PLOT2 71
C PLOT2 72
C PLOT2 73
C PLOT2 74
C PLOT2 75
C PLOT2 76
C PLOT2 77
C PLOT2 78
C PLOT2 79
C PLOT2 80
C PLOT2 81
C PLOT2 82
C PLOT2 83
C PLOT2 84
C PLOT2 85
C PLOT2 86
C PLOT2 87
A(K)=BLANK
IF(MOD(K,10),EQ.1) A(K)=OUT
IF(MOD(J,5),EQ.1) A(K)=OUT
70 CONTINUE
DO 80 I=1,NFUN
LOC=IF(I.J)-VFIRST(I)/STEP(I)+1.5
C=ALOC
A(LOC)=CH(I)
IF(C.NE.BLANK..AND.C.NE.DOT) A(LOC)=ASTER
80 CONTINUE
IF(MOD(J,10),EQ.1) GO TO 95
WRITE(6,85) P
85 FORMAT(25X,101A),
GO TO 100
95 WRITE(6,19) B,A
15 FORMAT(12X,1PE12.4,-1X,101A1)
100 CONTINUE
RETURN
END
SUBROUTINE KUTNER(NOTMY0,EPSE,HCX,FIRST)
DIMENSION K0S(1),T6,98.1(h18,PEb9E)
COMMON /UUT /NPRINT,NPLOTENF
COMMON /ACCEL,IXACCEL,ACCL,ACL,ML
DATA NAMI,?IAMI,IEHY1,2MYS I/UNE
C NO *NUMBER
C H E M O . E Q UATIONS, NO. OF COMPONENTS OF YQ
C INCREMENT FOR WHICH SOLUTION IS TO RETURNED OR -
C Y0 = THE VECTOR OF DEPENDENT VARIABLES, ENTER WITH INITIAL
C VALUES AT T AND RETURN WITH VALUES AT T+H
C EPSE = RELATIVE ERROR CRITERION FOR COMPONENTS OF YQ ,OT ABS(A)
C NOTE -- EPSE AND A MUST BE SPECIFIED FOR EACH COMPONENT OF THE SYSTEM
C MCX = THE SMALLEST STEP SIZE USED IN THE INTEGRATION
C FIRST SHOULD BE 0 WHEN KUTNER IS ENTERED FOR THE FIRST TIME
C AFTER THAT FIRST IS 1 IF KUTNER IS ENTERED WITH THE SAME M OR
C IF IT IS ENTERED WITH A CHANGED M
C IF FIRST IS 2 THE ERROR CRITERIA CANNOT BE MEET AND THE STEP SIZE
C REDUCED TO M/128.
C IF (FIRST.) 20,10,20
C - - - - - - - - FIRST ENTRY
10 HC=M
IPLUC=1
FIRST=1.
C - - - - - - OTHER ENTRY
20 LOC=0
MCX=HC
IF (MCX,NE.0.) GO TO 30
WRITE(6,800)
800 FORMAT(5X,4SHKUTNER ENTERED WITH NULL INTEGRATION INTERVAL )
FIRST=2.
RETURN
C - - - - - - - 5 CALLS TO DAUX
30 CALL DAUX(T,Y,V0,F0)
IF(NPRINT.EQ.5) WRITE(6,1000) Y0,T,F0
1000 FORMAT(6(2X,F10.4),4HTIME),T,F0)
408 FORMAT(6(2X,F10.4))
IF(NPRINT.EQ.5) WRITE(6,800) HC
39 DO 40 1=1,ND
40 CONTINUE
55
40 Y1(I) = Y0(I)*(HC/3.0)*F0(I)
   IF(NPRINT,EQ.5)WRITE(6,400)Y1(I)
   CALL DAUX(T+MC/3.0,Y1,F1)
   IF(NPRINT,EQ.5)WRITE(6,400)F1(I)
   DO 50 I=1,ND
50 Y1(I) = Y0(I)*(HC/6.0)*F0(I)+(MC/6.0)*F1(I)
   IF(NPRINT,EQ.5)WRITE(6,400)Y1(I)
   CALL DAUX(T+MC/2.0,Y1,F2)
   IF(NPRINT,EQ.5)WRITE(6,400)F2(I)
   DO 60 I=1,ND
60 Y2(I) = Y0(I)+(MC/2.0)*F0(I)+(MC/6.0)*F1(I)
   IF(NPRINT,EQ.5)WRITE(6,400)Y2(I)
   INC = 0
C = - - - - - - - CHECK ERROR CRITERIA
   DO 110 I=1,ND
110 ZZZ = ABS(Y1(I))-A(I)
   IF(ZZZ) 85,07,07
C = - - - - - - - ABSOLUTE ERROR
   85 ERROR = ABS(.2*(Y1(I)-Y2(I)))
   IF(ERROR-A(I)) 100,100,90
C = - - - - - - - RELATIVE ERROR
   87 ERROR = ABS(.2*(Y2(I)-Y1(I)))/Y1(I)
   IF(ERROR=EPS(I)) 100,100,90
C = - - - - - - - IF YES THEN HALVE INTERVAL, OTHERWISE STOP.
   90 X = 128.*A5(MC)-A8(M)
   IF(X) 91,05,95
C = - - - - - - - ERROR TOO LARGE
   91 WRITE(6,92)T,ERROR,MC
92 FORMAT(10H FOR EQUATION NO. 12,27H, THE RELATIVE ERROR AT T = ,
#15.8, 4H IS E15.8,13H STEP SIZE = E15.8)
   FIRST = 2.
   RETURN
C = - - - - - - - HALVE INTERVAL
   95 MC = MC/2.
   IPLOC = 2*IPLOC
   LOC = 2*LOC
   HCX = HC
   WRITE(2,71)T,1,ERROR,MC
71 FORMAT(3H TIME = ,F10.3,5X,HALVE INTERVAL, EQUATION +13,
#13H HAS ERROR = E16.8,6X,17H STEP SIZE NOW = E15.8)
   WRITE(2,72)NAM2,Y(I,J),J=1,ND
72 FORMAT(2X,A2,7X,NAM1,Y(I,J),J=1,ND)
C --- --- --- --- --- TEST IF INTERVAL LENGTH CAN BE DOUBLED
100 IF (ERROR*44.+EPSL(E1)) 110,110,111
110 INC = 1
111 CONTINUE
C --- --- --- --- --- UPDATE T AND SOLUTION
112 T = T+HC
114 Y0(I) = YE(I)
C --- --- --- --- --- GET SOLUTION IN NEXT INTERVAL
LOC = LUC
120 IF (ILOC)=100.100 210 219
130 IF (LUC-(LOC/2)=210,149,214
140 IF (ILOC-1)=210,210,200
C --- --- --- --- --- DOUBLE INTERVAL LENGTH
C200 HC = 2*HC
ILOC = ILOC/2
210 IF (ILOC-LOC)=30,329,30
329 GwACL = F0(2)-KACCL*FO(3)
C0ACL = FO(2)
RETURN
END
END
SUBROUTINE DAUX(TIME,X,RHS)
C TIME AT WHICH SYSTEM IS TO BE EVALUATED
C X STATE VECTOR
C RHS THE RIGHT HAND SIDE OF THE EQUATION S = F A
C REAL IAT,MK,MA,MMP,NCNXL,NL,NMAX
INTEGER ENBI,TIME
DIMENSION X(6),RHS(6),F(3,1),A(3,3),INDEX(3,3),
* R(120),V(120),D(120)
C COMMON /SHIP/MASS,CE1,CE2,CE3,DMU,EDMU,E2DMU,E3UMU,ES,MMX,
* NL,FL,IA1,L(120)
COMMON /CUTST/NCG,ECC,P1,DRP,RPD,GRAVITY,NUM,K,NAM,L(120),G,TA,
* B(120),PRE,TH,MT,TH,XO,T,XP,H,IT,
* DLTAS,I,KST(120),C,RK,MMAX(1,0),TEST(120),
* N(120),PHALF
COMMON /IN/ RM(120),RL(120),V,LIN
COMMON /OUTPRNT/NPLOT,END
COMMON /SEAC/START,RISE,RAMP
COMMON /WAVE/ PI,PT(120),ZMA,ZMA,E2MA,E3MA,E2MA,
* ZWONT(120)
C RAMP = RMP(TIME,START,RISE)
P1M = PI/2
CT = C*TIME
C6 = C6S(X(6))
S6 = SIN(X(6))
COM---SET VALUES OF MA AND B
DO 75 I=1,NUM
PT(I) = (X(4)+S(I))*C6S(N(I))*S6+CT)*K
R(I) = HUG*CO5(PT(I)*RAMP
COM*** COMPUTE HW SUBSEQUENCE OF A POINT AND R THE WAVE
C HW(I) IS IN THE FIXED COORDINATE SYSTEM

57
MW(I) = X(I)*X(I)*C6+N(I)*C6-R(I)

IF(MW(I) < 0.0) GO TO 65

C
MA(I) = 0.
B1(I) = 0.
W(I) = 0.
GO TO 75

65 V(I) = -R1*K*SIN(P1(T(I))*RAMP
D(I) = MW(I)/C6-V(I)*S6

D(I) IS IN THE BODY AXIS SYSTEM AND IS THE SUBMERGENCE

C
IF(D(I) < 0.0) TEST(I) GO TO 70

C
MA(I) = CRAFT IS NOT SUBMERGED

B1(I) = D(I)*R1/T1*PIM
MA(I) = K1*HALF*R1(I)*B(I)
GO TO 75

C
CHINE IS IMMERSSED

81 ARRAY IS USED FOR THE INTEGRALS OVER THE PORTION OF THE HULL FOR WHICH THE CHINE IS NOT IMMERSED

C
MA(I) = MAX(I)
W(I) = WM(I)
W1(I) = 0.

75 CONTINUE
IF(NPRINT < LT,4) GO TO 85

WRITE(6,74) TIME
WRITE(6,76) X(I), I = 1, 6
WRITE(6,77) R(I), I = 1, NUM
WRITE(6,78) V(I), I = 1, NUM
WRITE(6,79) (O(I), I = 1, NUM)
WRITE(6,80) (O(I), I = 1, NUM)
WRITE(6,81) (O(I), I = 1, NUM)
WRITE(6,82) (O(I), I = 1, NUM)

76 FORMAT(' ', R(I), 10F10.4)
77 FORMAT(' ', X(I), 10F10.4)
78 FORMAT(' ', V(I), 10F10.4)
79 FORMAT(' ', W(I), 10F10.4)
80 FORMAT(' ', W1(I), 10F10.4)
81 FORMAT(' ', MA(I), 10F10.4)
82 FORMAT(' ', W1(I), 10F10.4)
83 CONTINUE

C
COMPUTES NL AND FL AND THE ASSOCIATED INTEGRALS
CALL FUNCT(X)

C
IF(NPRINT < LT,4) GO TO 17
WRITE(6,15) TX, FL, DRAG, T2, W, NL, XD, TX, XP

15 FORMAT(' ', 10E12.6)
17 CONTINUE

C
COMPUTE THE F VECTOR
F(1,1) = TX*FL*DSX*DRAG*CX6
F(1,1) = 0.0
F(2,1) = T2*FL*DSX*DRAG*DSX+W
F(3,1) = NL*DRAG*DSX*TX
IF(NPRINT < LT,4) GO TO 18
WRITE(6,16) (F(I,1), I = 1, 3)

18 CONTINUE

C
COMPUTE THE A MATRIX
A(1,1) = M*MASS*DSX*DSX

58
A(1,2) = MASS*AX6*CX6
A(1,3) = -MA*AX6
A(1,2) = 0.
A(1,3) = 0.
A(2,1) = A(1,2)
A(2,2) = M*MASS*AX6*CX6
A(2,3) = -MA*CX6
A(3,1) = A(1,3)
A(3,2) = A(2,3)
A(3,3) = IT+IA

IF(NPRINT,LT,3) GO TO 25
WRITE(6,12) (A(i,j),I=1,3)
WRITE(6,13) (A(i,j),I=1,3)
WRITE(6,14) (A(i,j),I=1,3)

C     INVERT THE MATRIX
25 CALL MATINV(A(3,3)+1,DETERMINU+INDEX)
IF(ID.EQ.2) WRITE(6,26)
26 FORMAT("MATRIX IS SINGULAR")

C     ON RETURN WILL CONTAIN THE INVERSE MATRIX
C
C     ID=2 MATRIX IS SINGULAR
C
C
C     INVERSE WAS FOUND
C
C
C     COMPUTE THE RIGHT HAND SIDE
RHS(1) = F(1,1)
RHS(2) = F(2,1)
RHS(3) = F(3,1)
RHS(1) = 0.
RHS(4) = X(1)
RHS(5) = X(2)
RHS(6) = X(3)

10 FORMAT("F(1,1) ",3(2X,E12.4))
12 FORMAT("A(1,1) ",3(2X,E12.4))
13 FORMAT("A(1,2) ",3(2X,E12.4))
14 FORMAT("A(1,3) ",3(2X,E12.4))
39 IF(NPRINT,LT,2) GO TO 40
WRITE(6,12) (A(i,j),I=1,3)
WRITE(6,13) (A(i,j),I=1,3)
WRITE(6,14) (A(i,j),I=1,3)
WRITE(6,35) (RHS(i),I=1,6)
35 FORMAT("RHS(I) ",6(2X,E12.6))
49 CONTINUE
RETURN
END
SUBROUTINE FUNCT(X)
REAL K, IA, IPART, KPI, MA, MASS, NL, NCG, IT, M, MM, N
INTEGER END
DIMENSION IPART(120), C1(120), C2(120), C3(120),
U(120), U2(120), U3(120), U4(120), U5(120), U6(120),
V(120), Z1(120), Z2(120), Z3(120), Z4(120), Z5(120),
Z6(120), Z7(120),
X(6), YMAX(120)

C
COMMON /SHIP/ MASS, CINT, OA, CE, CEZ, CE3, DHMU, EDMU, E2DHMU, E3DHMU, AMUF, BMUF
COMMON /NINF/ NL, FL, IA, E(120)
COMMON /CUVST/ NCG, ECG, PI, DPAR, XD, DHAVTY, RHO, K, NUMA(120), CO, TA,
B(120), BETAM(120), T2, UMAU, U20, XD, XP, M, IT,
DELTA1, T1, EST(120), C, RU, KAR, MMAX(10), TEST(120),
N(120), PHALF
COMMON /IN/ BM(120),BL(120),VELIN
COMMON/OUT/NPRINT,NPLOT,END
COMMON /WAVE/ R(120),PT(120),ZMA,ZWMA,EMAS,ZZMA,ZWMA,FAEMAZ

* 2WDOT(120)
COMMON /INTER/ II,KTT(10),DIF(10)
COMMON /SEWAVE/ START,RISE,RAMP
COMMON /TEST/ VMA

C * * * * * * * * INITIALIZE INTEGRAL SUMS
MASS = 0.0
QD = 0.0
IA = 0.0
CE = 0.0
CCE = 0.0
EMU = 0.0
EDMU = 0.0
EDMU = 0.0
MF = 0.0
BMW = 0.0
ZMA = 0.0
ZWMA = 0.0
EMAS = 0.0
ZMA = 0.0
ZWMA = 0.0
EEMAS = 0.0

VPART = X(1)*SIN(X(6))*X(2)*COS(X(6))
SX6 = SIN(X(6))
CX6 = COS(X(6))

C * * * * * * * * SET UP THE FUNCTIONS FOR THE INTEGRALS
DO 90 I=1,NUM
IPART(I)*E(I)*E(I)*MA(I)
QPART(I)*E(I)*MA(I)
2WDOT(I) = -RUNIT*SIN(P1(I))*RAMP
U = X(1)*CXA*X(2)*SIN*2WOUT(I)*SX
VEL = VPART*X(1)*E(I)*2WOUT(I)*CXA
Z1(I) = MA(I)*2WDOT(I)
Z2(I) = -MA(I)*COS(P1(I))*RAMP
Z3(I) = E(I)*Z2(I)
Z4(I) = E(I)*Z1(I)
Z5(I) = 1/Z2(I)
Z6(I) = E(I)*Z4(I)
Z7(I) = MA(I)*VEL*U
IF (VEL*LE,0.), GO TO 60
IF (RUNIT*,LE,0.) GO TO 50

DO 90 D1 = 0.
D1 = E(I)*D1(I)
C1 = VEL*VEL*B(I)
C2 = E(I)*C1(I)
GO TO 61

GO TO 51

GO TO 52

GO TO 53

GO TO 54

GO TO 55

GO TO 56

GO TO 57

GO TO 58

GO TO 59

GO TO 60

GO TO 61

GO TO 62

GO TO 63

GO TO 64

GO TO 65

GO TO 66

GO TO 67

GO TO 68

GO TO 69

GO TO 70

GO TO 71

GO TO 72

GO TO 73

GO TO 74

GO TO 75

GO TO 76

60
61 CONTINUE

D3(I) = Z2(I)*VEL
D4(I) = E(I)*D3(I)

PIM = PI/2.

.05(I) = B(I)*(M(I)-B(I)*TA/2.)
66 06(I) = D5(I)*E(I)*S
90 CONTINUE

RHOG=F/H0*GAVITY

C * * * * * SET UP THE FUNCTIONS FOR THE INTEGRALS (PAGE 5 OF NOTES)

PIM = PI/2.

KPI = KAR*PI

C EVALUATE INTEGRALS USING TRAP METHOD

INDEX = 1

91 CALL TRAP(MA(INDEX),DIFF(I),KTT(I),TMASS)
91 CALL TRAP(MPART(INDEX),DIFF(I),KTT(I),GA1)
91 CALL TRAP(C1(INDEX),DIFF(I),KTT(I),CEA)
91 CALL TRAP(C2(INDEX),DIFF(I),KTT(I),CE2A)
91 CALL TRAP(IPART(INDEX),DIFF(I),KTT(I),IAA)
91 CALL TRAP(D1(INDEX),DIFF(I),KTT(I),UWMA)
91 CALL TRAP(N2(INDEX),DIFF(I),KTT(I),EDMU)
91 CALL TRAP(N3(INDEX),DIFF(I),KTT(I),E2DMU)
91 CALL TRAP(N4(INDEX),DIFF(I),KTT(I),E3DMU)
91 CALL TRAP(N5(INDEX),DIFF(I),KTT(I),BFA)
91 CALL TRAP(N6(INDEX),DIFF(I),KTT(I),UMMA)
91 CALL TRAP(Z1(INDEX),DIFF(I),KTT(I),ZMAA)
91 CALL TRAP(Z2(INDEX),DIFF(I),KTT(I),ZMAA)
91 CALL TRAP(Z3(INDEX),DIFF(I),KTT(I),EMAS)
91 CALL TRAP(Z4(INDEX),DIFF(I),KTT(I),ZEMAS)
91 CALL TRAP(Z5(INDEX),DIFF(I),KTT(I),ZEMAS)
91 CALL TRAP(Z6(INDEX),DIFF(I),KTT(I),ZEMAS)
91 CALL TRAP(Z7(INDEX),DIFF(I),KTT(I),EZMA)

C 93 CONTINUE

MASS = MASS + TMASS

QA = QA + GA1

IA = IA + IAA

CE = CE + CEA

CE2 = CE2 + CE2A

DMU = DMU + DUMA

EDMU = EDMU + EDMU

E2DMU = E2DMU + E2DMU

E3DMU = E3DMU + E3DMU

BF = BF + RHOG*BA

BMM = BMM + RHOG*BMMA

ZMA = ZMA + ZMAA

ZEMAS = ZEMAS + ZEMAS

ZEMAS = ZEMAS + ZEMAS

ZEMAS = ZEMAS + ZEMAS

E2MAZ = E2MAZ + E2MAZ

94 CONTINUE

IF ( I = 0 ) GO TO 92

INDEX = INDEX + KTT(I)-1

92 CONTINUE
CALL COMPUT TO FIND THE VALUE OF VL AND FL USING

CALL COMPUT(X)

IF (NPRINT.LT.3) GO TO 111
IF (NPRINT.EQ.3) GO TO 108
IF (NPRINT.EQ.4) GO TO 108
WRITE(6,97) (IPART(I),I=1,NUM)
WRITE(6,98) (QPART(I),I=1,NUM)
WRITE(6,99) (C1(I),I=1,NUM)
WRITE(6,100) (C2(I),I=1,NUM)
WRITE(6,101) (C3(I),I=1,NUM)
WRITE(6,102) (D1(I),I=1,NUM)
WRITE(6,103) (D2(I),I=1,NUM)
WRITE(6,104) (D3(I),I=1,NUM)
WRITE(6,105) (D4(I),I=1,NUM)
WRITE(6,106) (D5(I),I=1,NUM)
WRITE(6,112) (D6(I),I=1,NUM)
WRITE(6,113) (Z1(I),I=1,NUM)
WRITE(6,114) (Z2(I),I=1,NUM)
WRITE(6,115) (Z3(I),I=1,NUM)
WRITE(6,116) (Z4(I),I=1,NUM)
WRITE(6,117) (Z5(I),I=1,NUM)
WRITE(6,118) (Z6(I),I=1,NUM)
WRITE(6,120) (Z7(I),I=1,NUM)
WRITE(6,107) KPI, RHO, PHM
WRITE(6,121) 1A
WRITE(6,110) DMU, EDMU, E3MU, E3MU, BW, BMH
WRITE(6,117) ZMA, ZWMA, EMAS, ZZMA, ZWMA, ZZMA, E2MAZ
WRITE(6,112) 1A
WRITE(6,111) D4M, EDMU, E3MU, E3MU, BMH
WRITE(6,117) ZMA, ZWMA, EMAS, ZZMA, ZWMA, ZZMA, E2MAZ
WRITE(6,121) 1A
WRITE(6,111) D4M, EDMU, E3MU, E3MU, BMH
WRITE(6,117) ZMA, ZWMA, EMAS, ZZMA, ZWMA, ZZMA, E2MAZ

100 FORMAT(* IA =E10.4)
WRITE(6,111) DMU, EDMU, E3MU, E3MU, BW, BMH
WRITE(6,117) ZMA, ZWMA, EMAS, ZZMA, ZWMA, ZZMA, E2MAZ
WRITE(6,112) 1A
WRITE(6,111) D4M, EDMU, E3MU, E3MU, BMH
WRITE(6,117) ZMA, ZWMA, EMAS, ZZMA, ZWMA, ZZMA, E2MAZ

C FORMAT(* CPART(I) =10(2X,E10.4))
96 FORMAT(" CPART(I)" =10(2X,E10.4))
97 FORMAT(" IPART(I)" =10(2X,E10.4))
98 FORMAT(" QPART(I)" =10(2X,E10.4))
99 FORMAT(" C1" =10(2X,E10.4))
100 FORMAT(" C2" =10(2X,E10.4))
101 FORMAT(" C3" =10(2X,E10.4))
102 FORMAT(" D1" =10(2X,E10.4))
103 FORMAT(" D2" =10(2X,E10.4))
104 FORMAT(" D3" =10(2X,E10.4))
105 FORMAT(" D4" =10(2X,E10.4))
106 FORMAT(" D5" =10(2X,E10.4))
107 FORMAT(" D6" =10(2X,E10.4))
108 FORMAT(" KPHI" =E10.4, "RHO" =E10.4, "PHM" =E10.4)
109 FORMAT(" MASS" =E10.4, "CINT" =E10.4, "QA" =E10.4, "CE" =E10.4)
110 FORMAT(" ZM" =E10.4, "CE" =E10.4)
111 FORMAT(" DMU" =E10.4, "EDMU" =E10.4, "E3MU" =E10.4, "E3MU" =E10.4)
112 FORMAT(" BF" =E10.4, "BMH" =E10.4)
113 FORMAT(* Z1 =10(2X,E10.4))
114 FORMAT(* Z2 =10(2X,E10.4))
115 FORMAT(* Z3 =10(2X,E10.4))
116 FORMAT(* Z4 =10(2X,E10.4))
117 FORMAT(* Z5 =10(2X,E10.4))
118 FORMAT(* Z6 =10(2X,E10.4))
119 FORMAT(* Z7 =10(2X,E10.4))
120 FORMAT(* Z8 =10(2X,E10.4))
117 FORMAT(* ZMA =E10.4, 6HM ZMA =E10.4, 6HM EMAS =E10.4, 6HM ZWMA =E10.4, 7HM ZZMA =E10.4, 7HM ZZMA =E10.4, 7HM E2MAZ =E10.4)
111 CONTINUE
RETURN
END

SUBROUTINE COMPUT(x)
DIMENSION x(6)
REAL KAR,KPI
REAL M,MASS,NGM,IT,IA,K,MA,NMAX,N
INTEGER END

C

COMMON /SHIP/ MASS,CINT,GA,CE,CE2,CE3,DMU,EDMU,EZUMU,EJUMU,BF,BMM

REAL NL,MASS,NGM,IT,IA,K,MA,NMAX,N

INTEGER END

C

COMMON /CONST/ NGM,ECG,P1,DP1,RP1,GRV,RYH,K,NUM,MA(120),CD,TA,

B(120),BETA,NUM(120),TW,DRAG,WD,TD,XP,MIT,

DELTA,TF,EST(120),C,KAR,MAX(1,0),TEST(120),

N(120),PHALF

COMMON/OUT/NPRINT,NPLOT,END

COMMON /TERMS/ T1,T2,T3,T4,T5,T6,T7,T8

COMMON /SAVE/ P(120),PT(120),ZMA,ZWMA,EMAS,ZZWMA,ZWMA,ZWMA,

E2MAZ,ZWDOT(120)

COMMON /TEST/ VMA

C

CX6 = COS(x(6))
SX6 = SIN(x(6))
NO = K+C
P1M = P1/2.0
KPI = KAR+P1
CONS1 = RO*W0*W0*C6X
CONS2 = (KPI*RMU+P1M/TA)/CX6
CONS3 = RO*W0*K*CX6*SX6
CONS4 = RO*W0*K*CX6*CX6

TERM1 = X(1)*CX6
TERM2 = X(1)*SX6
UVNUM = (X(1)*CX6-X(2)-ZWDOT(NUM))*(SA)*

(X(1)*SX6-X(3)-ZWDOT(NUM))*CX6)

ZMA = ZMA*X(3)*SX6
ZWMA = ZWMA*X(3)*SX6
ZWMA = ZWMA+CUNS1
EMAS = EMAS+CUNS1
DMU = DMU+CUNS2
EDMU = EMDU+CUNS2
CE = CE+CD+RMU
CE2 = CE2+CD+RM0
E2MU = E2MU+CUNS3
E3MU = E3MU+CUNS3
ZWMA = ZWMA+CUNS4
ZWMA = ZWMA+CUNS4

20 T1 = QA*X(1)*(TERM1-TERM2)
T1 = T1+ZWMA-EMAS
T2 = EMDU
T3 = CE2
T4 = MA(NUM)*E(NUM)*UVNUM + E2MAZ + E3MU - ZWMA + BM
NL = T1 + T2 + T3 + T4 + BM
T5 = MASS*X(3)*TERM2-TERM1)
T5 = T5+ZWMA-ZMA
T6 = -DMU
T7 = -CE
SUBROUTINE INPUT

C * * * * * * DEFINITION OF INPUT VARIABLES
C
C XA = INITIAL TIME
C XE = FINAL TIME
C MMIN = MINIMUM STEP SIZE
C MMAX = MAXIMUM STEP SIZE
C EPS = RELATIVE ERROR CRITERION USED FOR VALUES OF Y GT A
C EPS = ERROR CRITERION IN KUTHER
C A = ABSOLUTE ERROR CRITERIA USED IN KUTHER
C NPRINT = 1 FINAL PRINTOUT
C = 2 MATRIX INVERSE MATRIX, F COLUMN MATRIX, AND KUTHER
C = 3 INTEGRAL VALUES
C = 4 CALCULATED VALUES-CONSTANT FOR GIVEN INPUT VALUES
C NPRINT = 0 NO PRINTOUT
C = 1 PRINTER PLOT
C END = NUMBER OF RUNS
C M = MASS OF CRAFT
C W = WEIGHT OF CRAFT
C TZ = THRUST COMPONENT IN Z DIRECTION
C TX = THRUST COMPONENT IN X DIRECTION
C XECO = DISTANCE FROM CO TO CENTER OF PRESSURE FOR NORMAL FORCE
C XP = MOMENT ARM OF PROPELLEN THRUST
C XD = DISTANCE FROM CC TO CENTER OF PRESSURE FOR DRAG FORCE
C KAI(I) = ADDED MASS COEFFICIENT
C N(II) = BEAM AT FREE SURFACE OR AT CHINE
C DRAG = FRICTION DRAG
C K = WAVE NUMBER
C RO = WAVE HEIGHT
C NU = WAVE SLOPE
C NUM = NUMBER OF STATIONS
C BL = BOAT LENGTH
C LAMBO = WAVE LENGTH
C RO = RADIUS OF GENERATION IN FEET
C T = PROPELLEN THRUST IN LBS
C GAMMA = PROPELLEN THRUST ANGLE IN DEGREES
C DELTAG = STATION SPACING IN FEET
C EGO = LONGITUDINAL CENTER OF GRAVITY
C NCG = VERTICAL CG
C BETA(I) = DEAD RISE
C NO(I) = HEIGHT OF MEAN BUTTOCK
C RHO = DENSITY OF WATER
C GRAVITY = GRAVITY FT/SEC=2
C DPR = DEGREES PER RADIUS
C RAD = RADIANS PER DEGREE
C PI = 3.14159 . . . . . .
EST(1) = STATION POSITION
START = START TIME OF THE RAMP FUNCTION FOR SEA WAVE
RISE = DURATION OF THE RISE FROM ZERO TO ONE OF THE RAMP

**IC OPTIONS**

IC(1) = 1 USE WAVE 2 DISTANCE IN COMPUTING LIFT COMPONENT
OF NL AND FL

REAL IT*,LAMDA,MA,MAX,N,NGC,NOM,MASS,NL,IA,KAR
INTEGER EN*

COMMON /CONST/N,ECG,PI,DPR,K,GRAVITY,RHO,K,NUM,MA(120),CD,TA,
* R(120),BETA,HI(120),TZ,DRAG,RO,T,XP,M,IT,
* DELTA,T,EST(120),C,RO,KAR,M,MA(1-0),TEST(120),*
* H(120),PA,M

COMMON /SHIP/MAX,CIN,TA,C,CE1,CE2,CE3,DMU,EMU,E3DMU,EDMU,B,MM,
* NL,FL,IA,E(120)

COMMON /IN/ BM(120)+B1(120)+VELIN
COMMON /INP/ NO(120),XAE,HI(120),HMIN,A(6),EPS(6)+LAMDA
COMMON/UUT/PRINT,INPLOT,END
COMMON /ACCEL/*ACCL,8WAC+COACL,UL

NAMELIST/HARIO/ANPRINT,END,HI,ML,TZ,XE,ECO,XP,AD,
* DRAG,RO,T,GA,ECU,NGC,KAR,RO,ALAMBA,NUM+BETA,EST
*
*
,XX,VE,MIN,HI,MAX,EPS,VELIN

DATA A/ 0.001,0.001,0.001,0.001/ DATA NPRINT,INPLOT,END/1,1,1,1/
DATA XL,TL,TX,XT,XX,ECO,XP,AD,
* DRAG,RO,T,GA,ECU,NGC,KAR,RO,ALAMBA,NUM+BETA,EST
*
*
,XX,VE,MIN,HI,MAX,EPS,VELIN

READ IN AND WRITE OUT KUTMER PARAMETERS AND PROGRAM

LETTER OPTIONS
READ(5,*,*P)
WRITE(6,*)
DO 101 W=1,4
10 EPSE(1) = FPS

END
IF (EST (NUM) .LT .375) STOP 3
C
COMPUTE NO AND RM ARRAYS
C
DO 32 I = 1, NUM
IF (EST (I) .GE .75) GO TO 30
NO (I) = 0.06675 * (1.0 - SQRT (EST (I) / 0.75 - (EST (I) / 0.75) ** 2.0))
BM (I) = 0.375 * SQRT (1.0 - (EST (I) / 0.75 - 1.0) ** 2.0)
GO TO 32
30 NO (I) = 0.0
BM (I) = 0.175
32 CONTINUE
C
COMPUTE CONSTANTS AND INITIALIZE ARRAYS
M = W / GRAVITY
RHO = 1.99
IT = 10*RO
K = 2.0*1/lambda
C = SQRT (GRAVITY / K)
NUM = RHO * K
PMHALF = (PI / 2.) * RHO
C
BETA = BETA*RPD
CD = COS (BETA)
TA = TAN (BETA)
DO 60 I = 1, NUM
C (I) = ECO - EST (I)
N (I) = NCQ + NO (I)
MMAK (I) = KAR * PMHALF * BM (I) * BM (I)
TEST (I) = (2. * BM (I) * TA) / PI
60 CONTINUE
C
END = END + 1
RETURN
END
C
SUBROUTINE PLUTER (FX, XA, HMAX, LAMBDA, IB, NWAVE, IPT)
C
INPUT:
FX
A TWO DIMENSIONAL ARRAY CONTAINING PITCH AND
HEAVE VALUES AT EACH TIME STEP
XA
INITIAL TIME
HMAX
TIME INTERVAL, PTIME*HMAX = INTERVAL BETWEEN
FX VALUES
LAMBDA
WAVELENGTH USED IN CALCULATING PITCH AND
HEAVE RATIOS
IB
NUMBER OF FX VALUES
NWAVE
START OF VALUES AFTER WAVE IS COMPLETELY ON
C
REAL IT, K, LAMBDA, MA, MMAK, N, NCQ
INTEGER ENQ
C
DIMENSION FX (2, 400), FMIN (2), FMAX (2), NVAR (2)
C
COMMON /CONST/ NCQ, ECO, PI, DPR, RPD, GRAVITY, RHO, K, NUM, MA (120), CD, TA,
* B (120), BETA, MA (120), T, DRAG, N, XO, T, XP, M, IT,
* DELTAS, TX, EST (120), C, RO, KA, MMAK (120), TEST (120),
* N (120), PHALF
COMMON /OUT/ NPRINT, NPLOT, ENQ
C
••••••••••• SET UP VALUES FOR PLOT AND CREATE PLOT
C * * * * * * SET UP MIN AND MAX LIMITS FOR PLOT
   FMIN(1)=FX(1,1)
   FMIN(2)=FX(2,1)
   FMAX(1)=FX(1,1)
   FMAX(2)=FX(2,1)
C * * * * * * SET UP MIN AND MAX LIMITS FOR PITCH AND HEAVE RATIO
   FMNP=FX(2,1)WAVE)
   FMNHF=FX(1,1)WAVE)
   FMNHF=FX(1,1)WAVE)
C
   DO 200 I=1,1M
   IF(FX(1,1),LT,FMIN(1))FMIN(1)=FX(1,1)
   IF(FX(1,1),LT,FMAX(1))FMAX(1)=FX(1,1)
   IF(FX(2,1),LT,FMIN(2))FMIN(2)=FX(2,1)
   IF(FX(2,1),LT,FMAX(2))FMAX(2)=FX(2,1)
   IF(I,LE,NWAVE)GO TO 200
   IF(FX(1,1),LT,FMIN(FMNHF))FMNH=FX(1,1)
   IF(FX(1,1),LT,FMAX(FMNHF))FMNH=FX(1,1)
   IF(FX(2,1),LT,FMIN(FMNHP))FMNH=FX(2,1)
   IF(FX(2,1),LT,FMAX(FMNHP))FMNH=FX(2,1)
200 CONTINUE
C * * * * * * COMPUTE RATIOS
   COL3=(FMNH-FMNHF)/(2.*RO)
   COL4=(FMNH-FMNHP)/(2.*R1/RO/LAMDA)
   WRITE(*,700) COL3,COL4
700 FORMAT(1H1," MEAVE AMPLITUDE/WAVEHEIGHT = "2E12.6/2X,
        " PITCH AMPLITUDE/WAVEHEIGHT/LAMDA = "2E12.6)
C
C899 CONTINUE
   NVAR(1)=10* HEAVE
   NVAR(2)=10* PITCH
   N1=2
   X0=XA
   DELA=MAMX
   IF(NPLOT,E,1,1)CALL PLOTZ(FX,FMIN,FMAX,NVAR,NFUN,N1,IB,X0,DELX)
   RETURN
END
SUBROUTINE TRAP(F,DX,NPTS,ANS)
C
C INPUT:       ARRAY OF FUNCTIONAL VALUES OF THE INTEGRAND
   F       DX       THE X INTERVAL BETWEEN VALUES
   NPTS    THE NUMBER OF VALUES GIVEN
C OUTPUT:     THE VALUE OF THE INTEGRAL
C
DIMENSION F(NPTS)
   ANS=0.0
   IF(NPTS.LT.2)GO TO 999
   DO 1 I=1,NPTS
   1 ANS=ANS+F(I)
   ANS=DX*(ANS+0.5*(F(1)+F(NPTS)))
999 CONTINUE
RETURN
END
FUNCTION RMP(T,START,RISE)
C
C
67
C * * * * * * * THIS FUNCTION IS USED TO GRADUALLY IMPLEMENT THE WAVE
C
C  T    CURRENT TIME
C  START  TIME TO START RAMP FROM 0.0 TO 1.0
C  RISE  THE LENGTH OF THE RISE FROM 0.0 TO 1.0
C
M=0.0
IF(T.LT.START) GO TO 99
IF(RISE.EQ.0.0) GO TO 80
TOP=T-START
M=1.0
IF(TOP.LT.RISE) M=TOP/RISE
GO TO 99
80 M=1.0
IF(T.EQ.START) M=0.5
90 RMP=M
RETURN
END
LISTING OF COMPUTER PROGRAM FOR CALCOMP PLOTS

PROGRAM PLTNSP (INPUT,OUTPUT,TAPES=INPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE7,TAPE9) MAIN 2
ITAPE = 7
CALL CALPLT (ITAPE)
STOP
END

SUBROUTINE CALPLT (ITAPE)
DIMENSION TIME (4003), PITCH (4003), MEAVE (4003)
CALL BUF (1000), BWACL (4003), CGACL (4003)
LOGICAL ACCCL
CALCUMP PLOT OF PITCH AND MEAVE VERSUS TIME

READ (10) XAXIS, YAXISP, YAXISM, MT
READ (10) XAXIS, YAXISP, YAXISM, MT

IF (IA, EQ, .0) ACCCL = .TRUE.
IF (ACCCL) READ (10) YAXISM, YAXISC
CALL REAUT (TIME, MEAVE, PITCH, BWACL, CGACL, NPTS)
CALL PLUT (100, 1000, 7)
CALL PLUT (0.5, 1.5, -3)
CALL ESCALE (TIME, XAXIS, NPTS, 1)
CALL ESCALE (NMEAVE, YAXISM, NPTS + 1)
CALL ESCALE (PITCH, YAXISP, NPTS + 1)

CALL ESCLFX (TIME, MEAVE, NPTS, 1)
CALL ESCLFX (TIME, PITCH, NPTS, 1)

IF (ACCCL) CALL ESCALE (BWACL, YAXISH, NPTS, 1)
IF (ACCCL) CALL ESCALE (CGACL, YAXISC, NPTS, 1)

N1 = NPTS + 1
N2 = NPTS + 2
N3 = NPTS + 3

CALL EAXIS (0,0,0,0,15, TIME IN SECONDS, 15, XAXIS, 0,0,0,)
TIME (N1), TIME (N2), TIME (N3), MT)
CALL EAXIS (0,0,0,13, TIME IN FEET, 13, YAXISH, 0,0,0,)
MEAVE (N2), MEAVE (N2), MEAVE (N3), MT)

TEMP = TIME (N2)
TIME (N2) = TIME (N2)/TIME (N3)
MEAVE (N2) = MEAVE (N2)/MEAVE (N3)
CALL LINE (TIME, MEAVE, NPTS + 1, 0, 0)
TIME (N2) = TEMP
KNEW = XAXIS + 3,
TH = .5

CALL PLUT (KNEW, 0, 0, -3)
CALL EAXIS (0,0,0,0,15, TIME IN SECONDS, 15, XAXIS, 0,0,0,)
TIME (N1), TIME (N2), TIME (N3), MT)
CALL EAXIS (0,0,0,13, TIME IN FEET, 13, YAXISP, 0,0,0,)
PITCH (N2), PITCH (N2), PITCH (N3), MT)
TIME (N2) = TIME (N2)/TIME (N3)
PITCH (N2) = PITCH (N2)/PITCH (N3)
CALL LINE (TIME, PITCH, NPTS + 1, 0, 0)
IF (NUT, ACCCL) GO TO 30
TIME (N2) = TEMP

CALL PLUT (KNEW, 0, 0, -3)
CALL EAXIS (0,0,0,0,15, TIME IN SECONDS, 15, XAXIS, 0,0,0,)
TIME (N1), TIME (N2), TIME (N3), MT)
CALL EAXIS (0,0,0,16, TIME IN SECONDS, 15, XAXIS, 0,0,0,)
BWACL (N2), BWACL (N3), MT)
TIME (N2) = TIME (N2)/TIME (N3)
BWACL (N2) = BWACL (N2)/BWACL (N3)

STOP
END MAIN
CALL LINE (TIME, NWACL, NPTS, 1, 0, 0)

C

TIME(N2) = TEMP
CALL PLUT(XNEW, 0, 0, -3)
CALL EAXIS(0, 0, 0, 0, 15, TIME(N1),
TIME(N2), TIME(N3), HT)
CALL EAXIS(0, 0, 0, 0, 15, ACCELERATION, 15, YAXIS, 90, 0, CGACL(N1),
GO TO 9, CGACL(N2), CGACL(N3),
TIME(N2) = TIME(N2)/TIME(N3))
CGACL(N2) = CGACL(N2)/CGACL(N3)
CALL LINE (TIME, CGACL, NPTS, 1, 0, 0)

30 CONTINUE
CALL PLUT(10, 9, 0, 0, 999)
RETURN
END

SUBROUTINE RCAOTITIMEHEAVEPITCM, NWACL, CGACL, NPTS)
READ 2
DIMENSION X(6), MEAVE(I), PITCH(I),
READ 3
TIME(I), NWACL(I), CGACL(I)
READ 4
4 CONTINUE
READ 5
READ 6
SUBROUTINE READ(TIME, MEAVE, PITCH, NWACL, CGACL, NPTS)
READ 7
TIME(I) = TIME(I) + (X(I), Y = PITCH(I),
READ 8
TIME(I), NWACL(I), CGACL(I)
READ 9
READ 10
SUBROUTINE EAXIS(AXPAGE, YPAGE, XHC0NCHAR, AXLEN, ANGLE, FIRSTY,
READ 11
DELTA Y, DELTA T, HT)
DIMENSION IBCD(I)

C

THIS ROUTINE WORKS LIKE THE CALCUMP AXIS WITH THE
C
EXCEPTION THAT THE TICK MARKS ARE NOT Necessarily
C
EVERY INCH AND THE HEIGHT OF THE CHARACTERS IS INPUTTED
C

CALL PLUT(AXPAGE, YPAGE, 3)
ISN = ISIGN(1, NCHAR)
ISGN = SIGN(1, DELTA Y)
AMIN = FIRSTY
X = AXPAGE
Y = YPAGE
XNUM = FIRSTY - DELTA Y
N = AXLEN/DELTAY
IF N < DELTA Y OR LT, AXLEN, N = N
AMAX = AMIN + (N * DELTA Y)
NDIO = NDIGIT(AMIN, AMAX, DELTA Y, ND)
10 CONTINUE
TEST = (NDIO = HT) + HT
IF (TEST < 0), DELTA Y > HT/2,
IF (TEST < 0), DELTA Y) GO TO 10
AYN = (1.5 * HT)
BYN = ((NDIO - 2) * HT) / 2.0, 5 * HT)
N = N + 1
TANG = (90.*ANGLE)/57.2958
ANG = ANGLE/57.2958
ST = SIN(TANG)
CT = COS(TANG)
S = SIN(ANG)
C = COS(ANG)
DO 30 I = 1,N
   IF(I.EQ.1) GO TO 20
   X = X + DELTAUC
   Y = Y + DELTAUS
   CALL PLUT(X,Y,2)
   IF(I.EQ.N) GO TO 20
   XT = X + (1.*CT)*ISN
   YT = Y + (1.*ST)*ISN
   CALL PLUT(XT,YT,2)
20  AN = X + Y*NCT*ISN + Y*NTN*C
   XNUM = XNUM + DELTAV
   CALL NUMBER(XN,YN,HT,XNUM,ANGLE,NU)
   CALL PLUT(X,Y,3)
30  CONTINUE
   XSP = ((AXLEN/HT)/2.5 - (IABS((NCHAH)/2.5)))*HT
   YSP = 3.5*HT
   XT = XHAGE + XSP*C + ISN*YSP*CT
   YT = YHAGE + YSP*S + ISN*YSP*ST
   CALL SYMBOL(XT,YT,HT,IRCD,ANGLE,IADS(NCHAR))
   RETURN
END

FUNCTION NDIGIT(AMIN,AMAX,ANUM,NU)
    FINOS THE NUMBER OF Digits NECESSARY TO PRINT
    EVEN INCREMENT OF THE FUNCTION ON THE AXIX
    NDIGIT  THE NUMBER OF PLACES IN THE ENTIRE NUMBER
    NO  THE NUMBER OF DECIMAL PLACES
    ANUM  THE VALUE GIVEN TO EACH INCREMENT ON THE AXIS

    IF(ABS(AMIN).LT.ABS(AMAX)) GO TO 20
    IF(AMAX.EQ.ABS(AMAX).AND.AMAX.NE.0) GO TO 20
    IF(ABS(AMIN).GT.ABS(AMAX)) GO TO 10
    AMAX = 1.
    AMIN = -1.
    GO TO 20
10   AMAX = ABS(AMIN)
20   IF(AMAX.LE.1.) GO TO 50
    NDIV = 10
    I = 1
30   IF(AMAX/NDIV.LT.1.) GO TO 40
    I = I + 1
    NDIV = NDIV*10
    GO TO 70
40   NDIGIT = I + 3
    ND = 2
    GO TO 60
50   NDIV = 10
    I = 1
60   IF(AMAX/NDIV.GT.1.) GO TO 70
    I = I + 1
70   RETURN
END

71
SUBROUTINE SCAL(ARRAY, AXLEN, PNTS, INC)

    DIMENSION ARRAY(1)
    AMIN = ARRAY(1)
    AMAX = ARRAY(1)
    ISGN = ISIGN(1+INC)
    INC = NABS(INC)
    DO 10 I = 1+NPTS, INC
        IF (ARRAY(I).LT. AMIN) AMIN = ARRAY(I)
        IF (ARRAY(I).GT. AMAX) AMAX = ARRAY(I)
    10 CONTINUE

    CALL AJUST(AMIN, AMAX, AUNIT, AXLEN, N, ANUM)

    ARRAY(NPTS+I) = AMIN
    ARRAY(NPTS+2) = AMAX
    IF (ISIGN(I).EQ.-I) ARRAY(NPTS+1) = AMAX
    ARRAY(NPTS+3) = AUNIT
    IF (ABS(ANUM).EQ.AUNIT) ARRAY(NPTS+2) = 1.*ISGN
    IF (ABS(ANUM).EQ.AUNIT) ARRAY(NPTS+3) = 1.
    RETURN
END

SUBROUTINE AJUST(AMIN, AMAX, AUNIT, AXLEN, N, ANUM)

    GIVEN AMIN AND AMAX WHICH ARE DISTINCT VALUES, ADJUST
    THEM SO THAT THEY ARE EVEN MULTIPLES OF AUNIT

    K = 1
    MIN = AMIN/ANUM
    IF (MIN.LT. MIN*ANUM) MIN = MIN+I
    AMIN = MIN*ANUM
    MAX = AMAX/ANUM
    IF (MAX.LT. MAX*ANUM) MAX = MAX+I
    AMAX = MAX*ANUM

    IF (TERM.LT. AMAX) GO TO 20

    CONTINUE
    RETURN
END
FUNCTION UNIT(AMIN, AMAX, AXLEN, N, NUM)

FINDS THE INCREMENT BETWEEN VALUES TO BE USED ON THE
AXIS AS FAR AS LABELING THE TICK MARKS

FINDS THE NUMBER OF DIVISIONS TO BE MADE ON THE AXIS

FINDS THE SIZE IN INCHES OF THESE DIVISIONS

IF(AMIN > AMAX) GOTO 10

AMIN = AMIN - 1
AMAX = AMAX + 1

10 IF(AMAX < 1. AND. AMIN > -1) G0 TO 110

90 IF(MIN < LT, 1 AND. MAX > GT, 0) G0 TO 100

MIN = MIN
MAX = MAX

IF(MAX < GT, MAX) MAX = MAX + 1
IF(MIN < LT, MIN) MIN = MIN - 1
IF(MIN < LT, 0) NWID = MAX + IABS(MIN)
IF(MIN < GE, 0) NWID = MAX - MIN

NUM = 10

40 IF(NWID < LT, NUM) G0 TO 60

NUM = NUM + 10

GO TO 40

50 N = NWID / (NUM / 10)

IF(N < (NUM / 10) < LT, IABS(MIN)) N = NN + 1

UNIT = AXLEN / N

GO TO 160

70 NN = IABS(MIN) / (NUM / 10)

IF(N < (NUM / 10) < LT, IABS(MIN)) N = NN + 1

N = MAX / (NUM / 10)

IF(N < (NUM / 10) < LT, MAX) N = N + 1

N = N + N

ANUM = NUM / 10.

UNIT = AXLEN / N

GO TO 160

110 NUM = 10

120 IF(AMAX = NUM < GT, 1) G0 TO 130

NUM = NUM + 10

GO TO 120

130 UNIT = 1. / NUM

140 N1 = AMIN / NUM

N2 = AMAX / NUM

IF(AMIN < NUM < LT, N1) N1 = N1 - 1

IF(AMAX < NUM < GT, N2) N2 = N2 + 1

IF(N1 > N2) G0 TO 130

AMIN = AMIN - UNIT

AMAX = AMAX - UNIT

GO TO 140

150 N = N2 - N1

ANUM = UNIT

IF(AMIN < LT, 0 AND AMAX < LT, 0) N = N1 - N2

IF(AMIN < LT, 0 AND AMAX > GE, 0) N = N2 - N1

UNIT = AXLEN / N
160 IF(N.GT.5) GO TO 170
   N = N*2
   ANUM = ANUM/2.
   AUNIT = AUNIT/2.
   GO TO 160
170 UNIT = AUNIT
   RETURN
END
<table>
<thead>
<tr>
<th>Copies</th>
<th>WES</th>
<th>Copies</th>
<th>NAVSHIPYD MARE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CHONR/438 Cooper</td>
<td>2</td>
<td>Library</td>
</tr>
<tr>
<td>1</td>
<td>Code 2027</td>
<td>1</td>
<td>Code 250</td>
</tr>
<tr>
<td>2</td>
<td>NRL</td>
<td>1</td>
<td>NAVSHIPYD BREM/Lib</td>
</tr>
<tr>
<td>1</td>
<td>Code 2829</td>
<td>1</td>
<td>NAVSHIPYD PEARL/Code 202.32</td>
</tr>
<tr>
<td>1</td>
<td>ONR/Boston</td>
<td>8</td>
<td>NAVSEC</td>
</tr>
<tr>
<td>1</td>
<td>ONR/Chicago</td>
<td>1</td>
<td>SEC 6034B</td>
</tr>
<tr>
<td>1</td>
<td>ONR/Pasadena</td>
<td>1</td>
<td>SEC 6110</td>
</tr>
<tr>
<td>1</td>
<td>NORDA</td>
<td>1</td>
<td>SEC 6114H</td>
</tr>
<tr>
<td>4</td>
<td>USNA</td>
<td>1</td>
<td>SEC 6120</td>
</tr>
<tr>
<td>1</td>
<td>Tech Lib</td>
<td>1</td>
<td>SEC 6138</td>
</tr>
<tr>
<td>1</td>
<td>Nav Sys Eng Dept</td>
<td>1</td>
<td>SEC 6140B</td>
</tr>
<tr>
<td>1</td>
<td>B. Johnson</td>
<td>1</td>
<td>SEC 6144G</td>
</tr>
<tr>
<td>1</td>
<td>Bhattacharyya</td>
<td>1</td>
<td>SEC 6148</td>
</tr>
<tr>
<td>3</td>
<td>NAVPGSCOL</td>
<td>12</td>
<td>NAVSEC, NORVA/6680.03 Blount</td>
</tr>
<tr>
<td>1</td>
<td>Library</td>
<td>1</td>
<td>DDC</td>
</tr>
<tr>
<td>1</td>
<td>T. Sarkeya</td>
<td>1</td>
<td>AFOSR/NAM</td>
</tr>
<tr>
<td>1</td>
<td>J. Miller</td>
<td>1</td>
<td>AFFOL/FYS, J. Olsen</td>
</tr>
<tr>
<td>1</td>
<td>NADC</td>
<td>1</td>
<td>NSF/Eng Lib</td>
</tr>
<tr>
<td>3</td>
<td>NOSC</td>
<td>1</td>
<td>LC/Sci and Tech</td>
</tr>
<tr>
<td>1</td>
<td>Library</td>
<td>1</td>
<td>LOT/Lib TAD-491.1</td>
</tr>
<tr>
<td>1</td>
<td>Fabric</td>
<td>1</td>
<td>MMA, Library</td>
</tr>
<tr>
<td>1</td>
<td>Hoyt</td>
<td>1</td>
<td>U. of BRIDGEPORT/E. Uram</td>
</tr>
<tr>
<td>1</td>
<td>NCSL/712 D. Humphreys</td>
<td>4</td>
<td>U. of CAL/Dept Naval Arch, Berkeley</td>
</tr>
<tr>
<td>1</td>
<td>NCEL/Code L31</td>
<td>1</td>
<td>Library</td>
</tr>
<tr>
<td>1</td>
<td>NSWC, Deihgren</td>
<td>1</td>
<td>Webster</td>
</tr>
<tr>
<td>1</td>
<td>NUSC/Lib</td>
<td>1</td>
<td>Pauling</td>
</tr>
<tr>
<td>1</td>
<td>NUSC/Lib</td>
<td>1</td>
<td>Wehausen</td>
</tr>
<tr>
<td>7</td>
<td>NAVSEA</td>
<td>2</td>
<td>U. of CAL, San Diego</td>
</tr>
<tr>
<td>1</td>
<td>SEA 0322</td>
<td>1</td>
<td>A.T. Ellis</td>
</tr>
<tr>
<td>1</td>
<td>SEA 033</td>
<td>1</td>
<td>Scripps Inst Lib</td>
</tr>
<tr>
<td>1</td>
<td>SEA 03512/Pierce</td>
<td>3</td>
<td>CIT</td>
</tr>
<tr>
<td>1</td>
<td>SEA 037</td>
<td>1</td>
<td>Aero Lib</td>
</tr>
<tr>
<td>1</td>
<td>3 SEA 06G32</td>
<td>1</td>
<td>T.Y. Wu</td>
</tr>
<tr>
<td>1</td>
<td>NAVFAC/Code 032C</td>
<td>1</td>
<td>A. Acosta</td>
</tr>
<tr>
<td>1</td>
<td>NAVSHIPYD PTSMH/Lib</td>
<td>1</td>
<td>CATHOLIC U. of AMER/CIVIL</td>
</tr>
<tr>
<td>1</td>
<td>NAVSHIPYD PHILA/Lib</td>
<td>1</td>
<td>MECH ENG</td>
</tr>
<tr>
<td>1</td>
<td>NAVSHIPYD NORVA/Lib</td>
<td>1</td>
<td>COLORADO STATE U./ENG RES CEN</td>
</tr>
<tr>
<td>1</td>
<td>NAVSHIPYD CHASN/Lib</td>
<td>1</td>
<td>U. of CONNECTICUT/Scottron</td>
</tr>
<tr>
<td>1</td>
<td>NAVSHIPYD LBEACH/Lib</td>
<td>1</td>
<td>CORNELL U./Sears</td>
</tr>
</tbody>
</table>

75
### CENTER DISTRIBUTION

<table>
<thead>
<tr>
<th>Copies</th>
<th>Code</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1500</td>
<td>W.E. Cummins</td>
</tr>
<tr>
<td>1</td>
<td>1504</td>
<td>V.J. Monocella</td>
</tr>
<tr>
<td>1</td>
<td>1508</td>
<td>M.K. Ochi</td>
</tr>
<tr>
<td>1</td>
<td>1507</td>
<td>D. Cieslowski</td>
</tr>
<tr>
<td>1</td>
<td>1512</td>
<td>J.B. Hadler</td>
</tr>
<tr>
<td>1</td>
<td>1520</td>
<td>R. Wermter</td>
</tr>
<tr>
<td>1</td>
<td>1521</td>
<td>P. Pian</td>
</tr>
<tr>
<td>1</td>
<td>1524</td>
<td>Y.T. Shen</td>
</tr>
<tr>
<td>1</td>
<td>1524</td>
<td>W.C. Lin</td>
</tr>
<tr>
<td>1</td>
<td>1532</td>
<td>G. Dobay</td>
</tr>
<tr>
<td>1</td>
<td>1532</td>
<td>R. Roddy</td>
</tr>
<tr>
<td>1</td>
<td>1540</td>
<td>W.B. Morgan</td>
</tr>
<tr>
<td>1</td>
<td>1552</td>
<td>J. McCarthy</td>
</tr>
<tr>
<td>1</td>
<td>1552</td>
<td>N. Salvesen</td>
</tr>
<tr>
<td>1</td>
<td>1550</td>
<td>G. Hagen</td>
</tr>
<tr>
<td>1</td>
<td>1550</td>
<td>N. Hubble</td>
</tr>
<tr>
<td>10</td>
<td>1582</td>
<td>M. Martin</td>
</tr>
<tr>
<td>1</td>
<td>1584</td>
<td>J. Feldman</td>
</tr>
<tr>
<td>1</td>
<td>1588</td>
<td>G. Cox</td>
</tr>
<tr>
<td>1</td>
<td>1572</td>
<td>M.D. Ochi</td>
</tr>
<tr>
<td>1</td>
<td>1572</td>
<td>C.M. Lee</td>
</tr>
<tr>
<td>1</td>
<td>1576</td>
<td>W.E. Smith</td>
</tr>
<tr>
<td>10</td>
<td>5214.1</td>
<td>Reports Distribution</td>
</tr>
<tr>
<td>1</td>
<td>522.1</td>
<td>Unclassified Library (C)</td>
</tr>
<tr>
<td>1</td>
<td>522.2</td>
<td>Unclassified Library (A)</td>
</tr>
</tbody>
</table>
DTNSRDC Issues Three Types of Reports

1. DTNSRDC reports, a formal series, contain information of permanent technical value. They carry a consecutive numerical identification regardless of their classification or the originating department.

2. Departmental reports, a semiformal series, contain information of a preliminary, temporary, or proprietary nature or of limited interest or significance. They carry a departmental alphanumerical identification.

3. Technical memoranda, an informal series, contain technical documentation of limited use and interest. They are primarily working papers intended for internal use. They carry an identifying number which indicates their type and the numerical code of the originating department. Any distribution outside DTNSRDC must be approved by the head of the originating department on a case-by-case basis.