DTNSRDC - 78/032



Bethesda, Md. 20084

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

March 1978

DTNSRDC-78/032

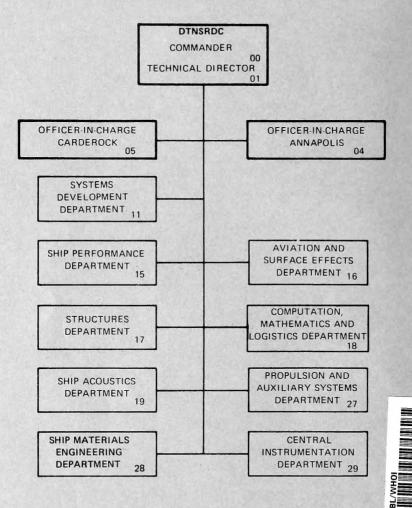
MATHEMATICAL MODEL OF MOTIONS OF A PLANING BOAT IN REGULAR WAVES

GC.

D3 no. A/032

the second second

**DTNSRDC-78/032** 



гu

m

Ln;

1032

1020 0

## MAJOR DTNSRDC ORGANIZATIONAL COMPONENTS

UNCLASSIFIED	(ered)			
REPORT DOCUMENTATION P	READ INSTRUCTIONS BEFORE COMPLETING FORM			
1. REPORT NUMBER 2 DTNSRDC-78/032	GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER		
A NONLINEAR MATHEMATICAL MODEL OF MOTIONS OF A PLANING BOAT		5. TYPE OF REPORT & PERIOD COVERED		
IN REGULAR WAVES	6. PERFORMING ORG. REPORT NUMBER			
7. AUTHOR(*)	8. CONTRACT OR GRANT NUMBER(*)			
Ernest E. Zarnick				
9. PERFORMING ORGANIZATION NAME AND ADDRESS David W. Taylor Naval Ship Research		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS		
and Development Center	ZF 43 421001			
Bethesda, Maryland 20084	Work Unit 1-1500-100			
1. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE March 1978		
Naval Sea Systems Command (SEA 035) Washington, D.C. 20362		13. NUMBER OF PAGES		
4. MONITORING AGENCY NAME & ADDRESS(if different f	rom Controlling Office)	86 15. SECURITY CLASS. (of this report)		
		UNCLASSIFIED		
		15. DECLASSIFICATION/DOWNGRADING SCHEDULE		
7. DISTRIBUTION STATEMENT (of the ebetrect entered in	Block 20, 11 dillerent froz	n Report)		
8. SUPPLEMENTARY NOTES				
9. KEY WORDS (Continue on reverse side if necessary and i	dentify by block number)			
Planing Boat Motions Hydrodynamic Impact Small Boat Worthiness Nonlinear Ship Motions in Waves				
ABSTRACT (Continue on reverse side if necessary and ic A nonlinear mathematical model has been if in regular waves, using a modified low-aspect-ratio would be large in comparison to the craft length in the equations of motion were determined by a A simplified version for the case of a craft or motion computations on a digital computer, and the result.	formulated of a craft o or strip theory. It and that the wave slo a combination of theo del being towed at co	was assumed that the wavelengths ppes would be small. The coefficients retical and empirical relationships. onstant speed was programed for		
		(Continued on reverse side)		
D 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLET	E	UNCLASSIFIED		

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

#### SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

#### (Block 20 continued)

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.

#### UNCLASSIFIED SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

# TABLE OF CONTENTS

# Page

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

# 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

# Page

8 - Trajectory of Computer Model Relative to Wave	20
9 – Heave Response for 10-Degree Deadrise Model at $V/\sqrt{L} = 6.0$	21
10 – Pitch Response for 10-Degree Deadrise Model at $V/\sqrt{L} = 6.0$	22
11 – Heave Response for 20-Degree Deadrise Model at $V/\sqrt{L} = 6.0$	23
12 – Pitch Response for 20-Degree Deadrise Model at $V/\sqrt{L} = 6.0$	24
13 – Heave Response for 30-Degree Deadrise Model at $V/\sqrt{L} = 6.0$	25
14 – Pitch Response for 30-Degree Deadrise Model at $V/\sqrt{L} = 6.0$	26
15 – Heave Response for 20-Degree Deadrise Model at $V/\sqrt{L} = 4.0$	27
16 – Pitch Response for 20-Degree Deadrise Model at $V/\sqrt{L}$ = 4.0	28
17 – Bow Acceleration for 10-Degree Deadrise Model at $V/\sqrt{L} = 6.0$	29
18 - Center of Gravity Acceleration for 10-Degree Deadrise Model at $V/\sqrt{L} = 6.0$	30
19 - Bow and Center of Gravity Accelerations for 20-Degree Deadrise Model at $V/\sqrt{L}$ = 4.0 and $V/\sqrt{L}$ = 6.0	31
20 – Bow and Center of Gravity Accelerations for 30-Degree Deadrise Model at $V/\sqrt{L} = 6.0$	32

Table 1 - Model Characteristics and Wave Conditions for Computations ..... 11

A	Mass matrix
A <sub>R</sub>	Section area
а	Correction factor for buoyancy force
b	Half-beam of craft
C <sub>D,c</sub>	Crossflow drag coefficient
$C_{\Delta}$	Load coefficient $\Delta/pg(2b)^3$
$c_{\lambda}$	Wavelength coefficient $L/\lambda \left[C_{\Delta}/(L/2b)^2\right]^{1/3}$
D	Friction drag force
F <sub>x</sub>	Total hydrodynamic force in x direction
Fz	Total hydrodynamic force in z direction
$F_{ heta}$	Total hydrodynamic moment about pitch axis
f	Two-dimensional hydrodynamic force
g	Acceleration of gravity
Н	Wave height, crest to trough
h	Vertical submergence of point below free surface
hz	Double amplitude of heave
I	Pitch moment of inertia
I <sub>a</sub>	Added pitch, moment of inertia
k	Wave number
k <sub>a</sub>	Two-dimensional added-mass coefficient
L	Hull length
LCG	Longitudinal center of gravity, percent of L
М	Mass of craft
M <sub>a</sub>	Added mass of craft

v

m	Sectional (two-dimensional) added mass
m <sub>a</sub>	
N	Hydrodynamic force normal to baseline
r	Wave elevation $r = r_0 \cos(kx + \omega t)$
r <sub>o</sub>	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/ft1/2
W	Weight of craft
wz	Vertical component of wave orbital velocity
w <sub>z</sub>	Vertical component of wave orbital acceleration
х	Fixed horizontal coordinate
x	Vector of state variables
х <sub>сс</sub>	Surge velocity
<sup>x</sup> <sub>CG</sub>	Surge acceleration
X <sub>C</sub> G	Surge displacement
Z	Fixed vertical coordinate
ż <sub>CG</sub>	Heave velocity
Ζ <sub>CG</sub>	Heave acceleration
<sup>Z</sup> CG	Heave displacement
β	Deadrise angle
Δ	Hull displacement W
ζ	Body coordinate normal to baseline
λ	Wavelength
θ	Pitch angle
$\dot{ heta}$	Pitch angular velocity

vi

### 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 wave's 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

1

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

$$\begin{split} M\ddot{x}_{CG} &= T_{x} - N \sin \theta - D \cos \theta \\ M\ddot{z}_{CG} &= T_{z} - N \cos \theta + D \sin \theta + W \\ I\ddot{\theta} &= Nx_{c} - Dx_{d} + Tx_{p} \end{split} \tag{1}$$

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

 $\boldsymbol{x}_d$  is distance from CG to center of action for friction drag force

x<sub>n</sub> 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, <sup>1</sup> and Chuang<sup>2</sup> has provided a strip method for determining loads on an impacting prismatic form that agrees extremely well with experimental results.

More recently, Martin<sup>3</sup> 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.

<sup>\*</sup>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.

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

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<sub>D,c</sub> 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

$$m_{a} = k_{a} \pi/2 \rho b^{2}$$
(3)

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

$$\frac{D}{Dt} (m_a V) = m_a \dot{V} + V \dot{m}_a - \frac{\partial}{\partial \xi} (m_a V) \frac{d\xi}{dt}$$
(4)

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,<sup>4</sup>

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.

$$C_{D,c} = 1.0 \cos\beta \tag{5}$$

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.

$$f_{\rm B} = -a\rho g(A) \tag{6}$$

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<sup>4</sup> 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  $\ell$  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

$$-N\cos\theta = F_{z}(t) = \int_{Q} f\cos\theta \,d\xi + \int_{Q} f_{B} \,d\xi$$
$$= -\left[\int_{Q} \left\{ m_{a}(\xi, t) \dot{V}(\xi, t) + \dot{m}_{a}(\xi, t) V(\xi, t) - U(\xi, t) \frac{\partial}{\partial \xi} \left[ m_{a}(\xi, t) V(\xi, t) \right] + C_{D,c}(\xi, t) \rho \,b(\xi, t) V^{2}(\xi, t) \right\} \cos\theta \,d\xi$$
$$+ a\rho g A d\xi \right]$$
(7)

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

$$\begin{split} F_{\mathbf{x}} &= \int_{\varrho} f \sin \theta \, d\xi \\ &= -\int \left\{ m_{\mathbf{a}}(\xi, t) \, \dot{\mathbf{V}}(\xi, t) + \dot{m}_{\mathbf{a}}(\xi, t) \, \mathbf{V}(\xi, t) \right. \\ &\left. - \, \mathbf{U}(\xi, t) \, \frac{\partial}{\partial \xi} \left[ m_{\mathbf{a}}(\xi, t) \, \mathbf{V}(\xi, t) \right] \right. \\ &\left. + \, \mathbf{C}_{\mathrm{D}, \, c}(\xi, t) \, \rho \, b(\xi, t) \, \mathbf{V}^{2}(\xi, t) \right\} \sin \theta \, d\xi \end{split}$$

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

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

$$\mathbf{r} = \mathbf{r}_{o} \cos \mathbf{k} (\mathbf{x} + c\mathbf{t}) \tag{11}$$

where  $r_0$  is the wave amplitude

k is the wave number

c is the wave celerity.

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

$$x = x_{CG} + \xi \cos \theta + \zeta \sin \theta \tag{12}$$

where  $x_{CG} = \int_{Q} \dot{x}_{CG} dt$ 

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.

$$\begin{aligned} F_{\theta} &= -\int_{\varrho} f(\xi,t)\xi d\xi - \int_{\varrho} f_{b} \cos\theta \xi d\xi \\ &= \int_{\varrho} \left\{ m_{a}(\xi,t)\dot{V}(\xi,t) + \dot{m}_{a}(\xi,t)V(\xi,t) \right. \\ &- U(\xi,t)\frac{\partial}{\partial\xi} \left( m_{a}(\xi,t)V(\xi,t) \right) + C_{D,c}(\xi,t)\rho b(\xi,t)V^{2}(\xi,t) \\ &+ a\rho gA \cos\theta \right\} \xi d\xi \end{aligned} \tag{13}$$

# EQUATIONS OF MOTION, GENERAL

 $F'_x$  $F'_{z}$  $F'_{\theta}$ 

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

$$(M + M_{a} \sin^{2} \theta) \ddot{x}_{CG} + (M_{a} \sin \theta \cos \theta) \ddot{z}_{CG} - (Q_{a} \sin \theta) \ddot{\theta}$$

$$= T_{x} + F'_{x} - D \cos \theta$$

$$(14)$$

$$(M_{a} \sin \theta \cos \theta) \ddot{x}_{CG} + (M + M_{a} \cos^{2} \theta) \ddot{z}_{CG} - (Q_{a} \cos \theta) \ddot{\theta}$$

$$= T_{z} + F'_{z} + D \sin \theta + W$$

$$- (Q_{a} \sin \theta) \ddot{x}_{CG} - (Q_{a} \cos \theta) \ddot{z}_{CG} + (1 + I_{a}) \ddot{\theta}$$

$$= F'_{\theta} - Dx_{d} + Tx_{p}$$

$$where M_{a}(t) = \int_{Q} m_{a}(\xi, t) \xi d\xi$$

$$I_{a}(t) = \int_{Q} m_{a}(\xi, t) \xi^{2} d\xi$$

$$F'_{x} = F_{x} - \left\{ -(M_{a} \sin^{2} \theta) \ddot{x}_{CG} - (M_{a} \sin \theta \cos \theta) \ddot{z}_{CG} + (Q_{a} \sin \theta) \ddot{\theta} \right\}$$

$$F'_{z} = F_{z} - \left\{ appropriate acceleration terms \right\}$$

$$F'_{\theta} = F_{\theta} - \left\{ appropriate acceleration terms \right\}.$$

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  $x_1 = \dot{y}_{CG}$ 

$$x_{2} = \dot{z}_{CG}$$
$$x_{3} = \dot{\theta}$$
$$x_{4} = x_{CG}$$
$$x_{5} = z_{CG}$$
$$x_{6} = \theta$$

Equation (14) can be rewritten, using matrix algebra, as

$$A\vec{\dot{x}} = \vec{g}$$
(15)

so that

$$\vec{x} = A^{-1}\vec{g}$$
(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.

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

$$\ddot{\mathbf{x}}_{CG} = 0$$

$$(\mathbf{M} + \mathbf{M}_{\mathbf{a}} \cos^2 \theta) \ddot{\mathbf{z}}_{CG} - (\mathbf{Q}_{\mathbf{a}} \cos \theta) \ddot{\theta} = \mathbf{F}'_{\mathbf{z}} + \mathbf{W}$$

$$- (\mathbf{Q}_{\mathbf{a}} \cos \theta) \ddot{\mathbf{z}}_{CG} + (\mathbf{I} + \mathbf{I}_{\mathbf{a}}) \ddot{\theta} = \mathbf{F}'_{\theta}$$

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.<sup>5</sup> 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

	CONFIGURATIONS							
SYMBO	SYMBOL β deg		LC perce		Radius o Gyration percent I	f V	$V/\sqrt{L}$	
A		20		0.0	25.1		4.0	
В	20		62	2.0	25.5		6.0	
J		10	10 68.0 26		26.2	6.0		
м		30	60	0.5	24.8		6.0	
WAVE CONDITIONS FOR CONFIGURATION								
1	A B			J	]	М		
<u>H/b</u>	$\lambda/L$	<u>H/b</u>	$\lambda/L$	<u>H/b</u>	$\lambda/L$	<u>H/b</u>	$\lambda/L$	
0.111	1.0	0.111	1.0	0.111	1.0	0.111	1.0	
0.111	1.5	0.111	1.5	0.111	1.5	0.111	1.5	
0.111	2.0	0.111	2.0	0.111	2.0	0.111	2.0	
0.111	3.0	0.111	3.0	0.111	3.0	0.111	3.0	
0.111	4.0	0.111	4.0	0.111	4.0	0.111	4.0	
0.111	6.0	0.222	6.0	0.111	6.0	0.111	6.0	
		0.334	4.0					
		0.111	6.0					

(Model Length = 114.3 cm (3.75 ft); L/b = 5;  $C_{\Lambda} = 0.608$ )

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 waveheight 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  $V/\sqrt{L} = 6.0$  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 Fridsma 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  $V/\sqrt{L} = 6.0$  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  $V/\sqrt{L} = 4.0$  to  $V/\sqrt{L} = 6.0$ . Computations of the motions were made for  $V/\sqrt{L} = 4.0$  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 accelerations and the 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 programed 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.

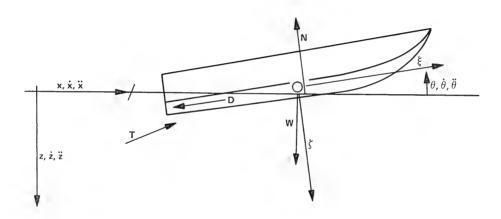


Figure 1 - Coordinate System

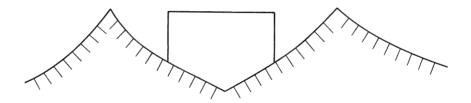


Figure 2a - Flow Separation from Chine

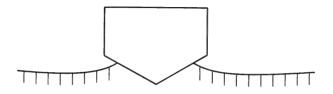
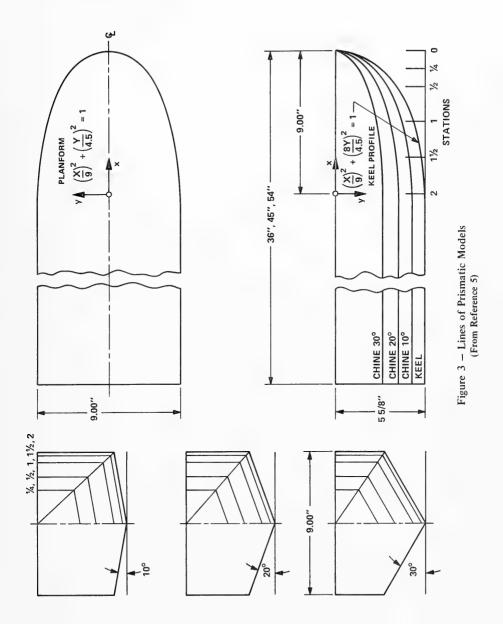


Figure 2b - Nonwetted Chine

Figure 2 - Types of Two-Dimensional Flow



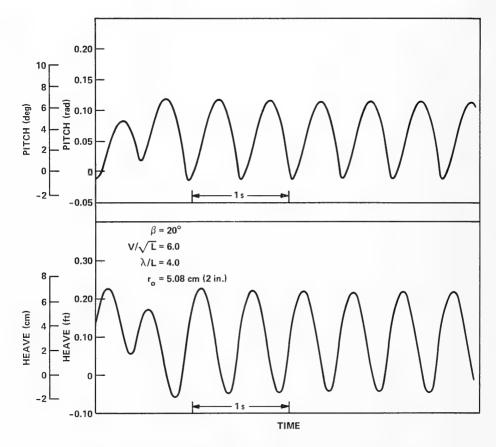


Figure 4 - Sample Time Histories of Computed Pitch and Heave Motions

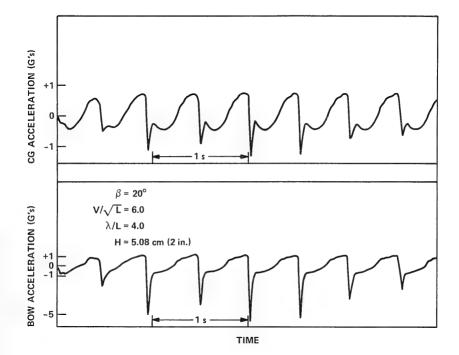


Figure 5 – Sample Time Histories of Computed Accelerations of Bow and Center of Gravity

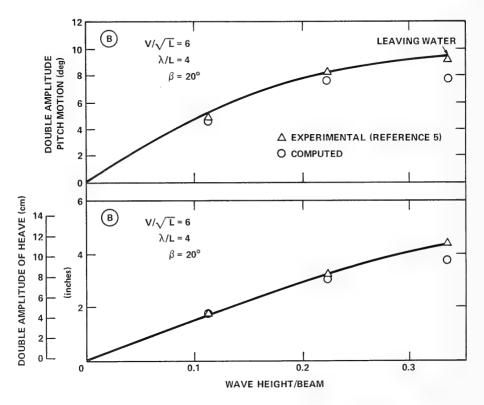


Figure 6 - Variation of Pitch and Heave with Wave Height

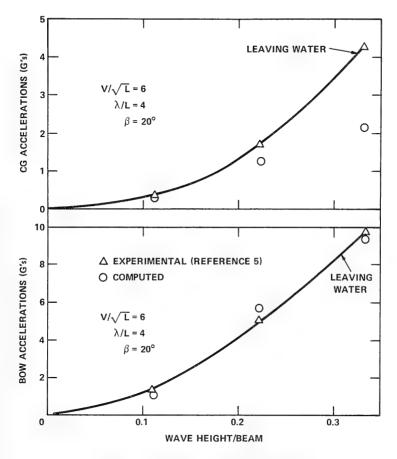


Figure 7 – Variation of Acceleration of Bow and Center of Gravity with Wave Height

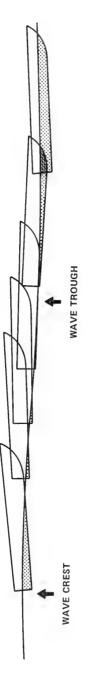


Figure 8 - Trajectory of Computer Model Relative to Wave

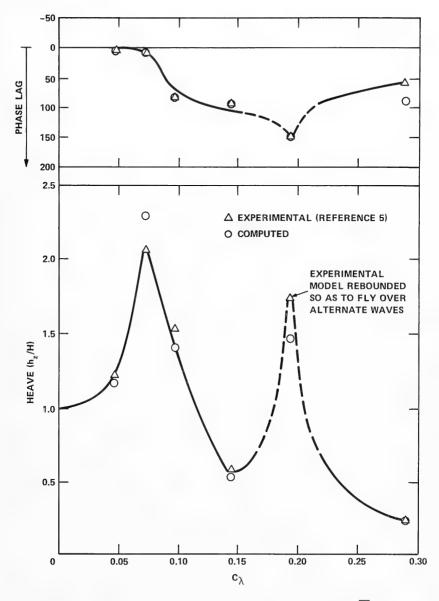


Figure 9 – Heave Response for 10-Degree Deadrise Model at  $V/\sqrt{L} = 6.0$ 

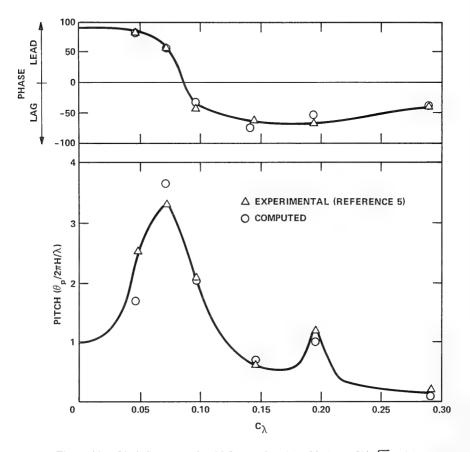


Figure 10 – Pitch Response for 10-Degree Deadrise Model at  $V/\sqrt{L} = 6.0$ 

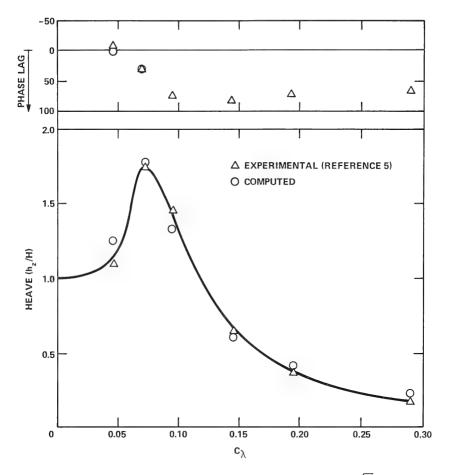
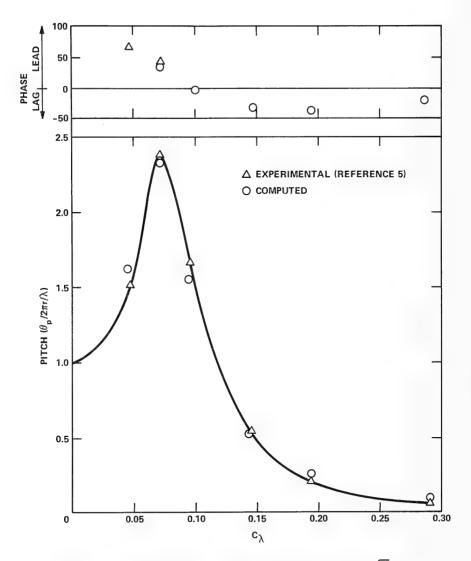
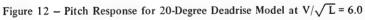


Figure 11 – Heave Response for 20-Degree Deadrise Model at  $V/\!\sqrt{L}\!=\!6.0$ 





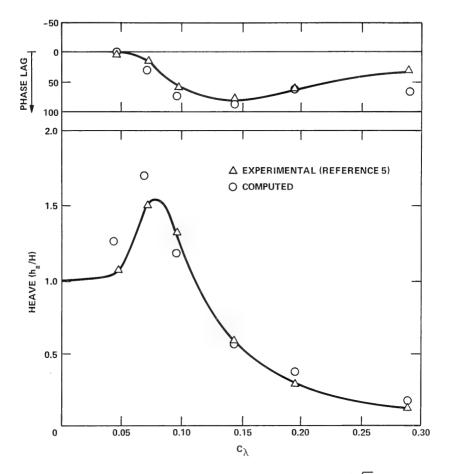
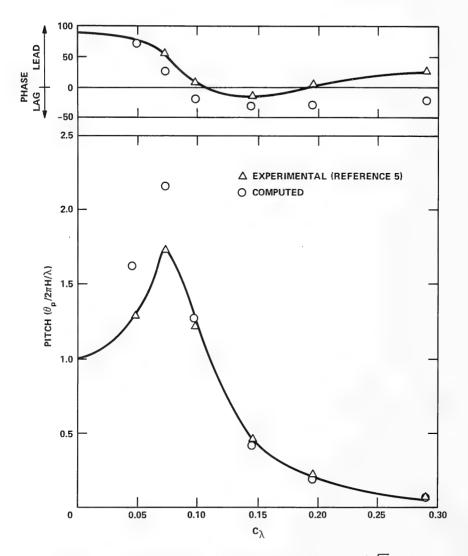


Figure 13 – Heave Response for 30-Degree Deadrise Model at  $V/\sqrt{L}$  = 6.0





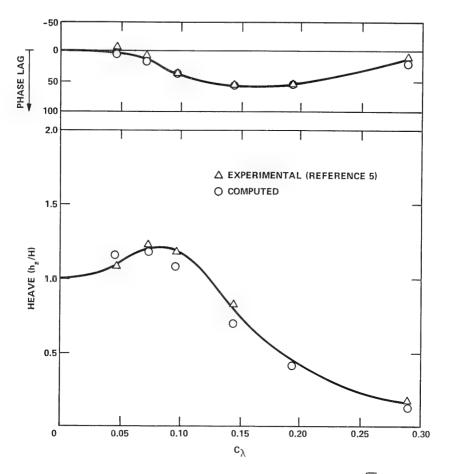


Figure 15 – Heave Response for 20-Degree Deadrise Model at  $V/\sqrt{L}$  = 4.0  $\,$ 

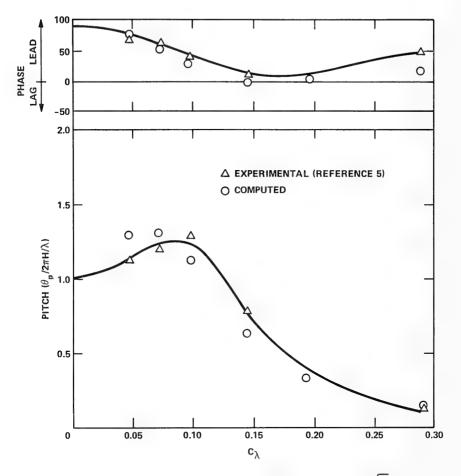


Figure 16 – Pitch Response for 20-Degree Deadrise Model at  $V/\sqrt{L} = 4.0$ 

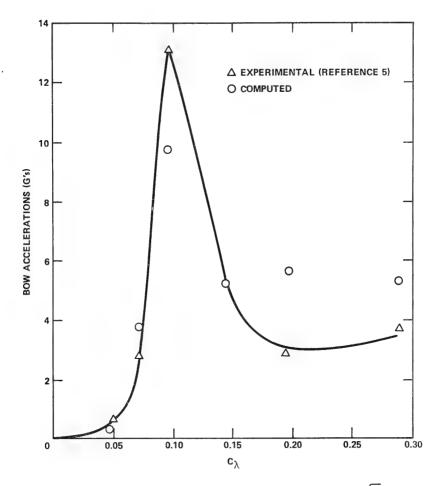


Figure 17 – Bow Acceleration for 10-Degree Deadrise Model at  $V/\sqrt{L} = 6.0$ 

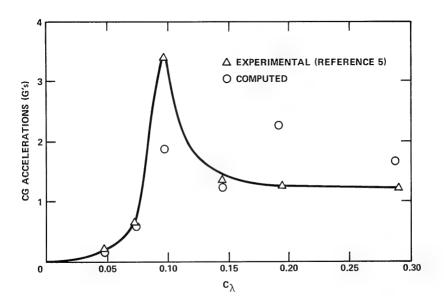


Figure 18 – Center of Gravity Acceleration for 10-Degree Deadrise Model at  $V/\sqrt{L} = 6.0$ 

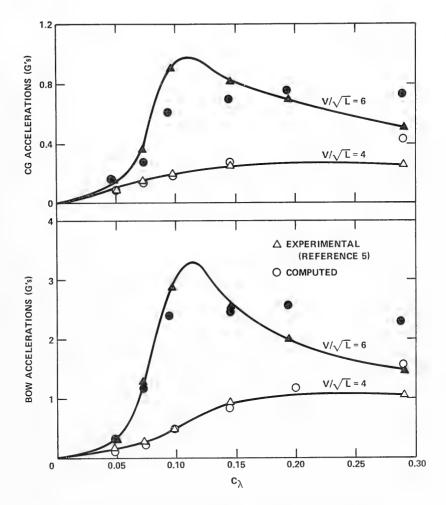
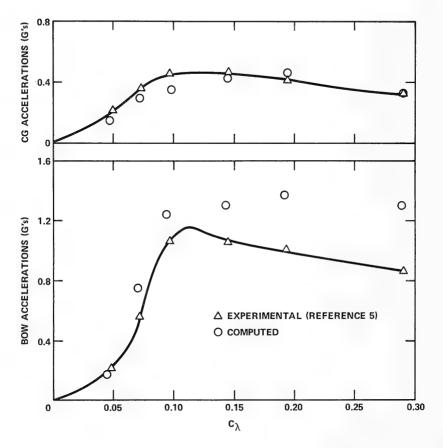
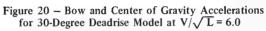


Figure 19 – Bow and Center of Gravity Accelerations for 20-Degree Deadrise Model at  $V/\sqrt{L}$  = 4.0 and  $V/\sqrt{L}$  = 6.0





### REFERENCES

1. Wagner, H., "Landing of Seaplanes," leitschrift fur Flegtechnik und Motorluflsshifffahrt, (14 Jan 1931); National Advisory Committee for Aeronautics TM 672 (May 1931).

2. Chuang, S.L., "Slamming Tests of Three-Dimensional Models in Calm Water and Waves," NSRDC Report 4095 (Sep 1973).

3. Martin, M., "Theoretical Predictions of Motions of High-Speed Planing Boats in Waves," DTNSRDC Report 76-0069 (Apr 1976).

4. Shuford, S.L., Jr., "A Theoretical and Experimental Study of Planing Surfaces Including Effects of Cross Section and Plan Form," National Advisory Committee for Aeronautics Report 1355 (1957).

5. Fridsma, G., "A Systematic Study of the Rough-Water Performance of Planing Boats," Davidson Laboratory, Stevens Institute of Technology Report R1275 (Nov 1969).

#### APPENDIX A EVALUATION OF HYDRODYNAMIC FORCE AND MOMENT INTEGRALS

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

$$F_{z} = -\int_{\varrho} \left\{ m_{a} \dot{V} - U \frac{\partial m_{a} V}{\partial \xi} + \dot{m}_{a} V + C_{D} \rho b V^{2} \right\} \cos \theta \, d\xi + \int_{\varrho} a \rho g A d\xi$$

where  $\mathbf{U} = \dot{\mathbf{x}}_{CG} \cos \theta - (\dot{\mathbf{z}} - \mathbf{w}_z) \sin \theta$ 

and

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

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

$$\dot{\mathbf{V}} = \ddot{\mathbf{x}}_{CG} \sin \theta - \ddot{\theta} \xi + \ddot{\mathbf{z}}_{CG} \cos \theta - \dot{\mathbf{w}}_{z} \cos \theta + \dot{\theta} (\dot{\mathbf{x}}_{CG} \cos \theta - \dot{\mathbf{z}}_{CG} \sin \theta) + \mathbf{w}_{z} \dot{\theta} \sin \theta \frac{\partial \mathbf{V}}{\partial \xi} = -\dot{\theta} - \frac{\partial \mathbf{w}_{z}}{\partial \xi} \cos \theta \frac{\partial \mathbf{U}}{\partial \xi} = \frac{\partial \mathbf{w}_{z}}{\partial \xi} \sin \theta \frac{d \mathbf{w}_{z}}{dt} = \dot{\mathbf{w}}_{z} - \mathbf{U} \frac{\partial \mathbf{w}_{z}}{\partial \xi}$$

and noting that

$$\int_{Q} UV \frac{\partial m_{a}}{\partial \xi} d\xi = -UV m_{a} \Big|_{\text{stern}} - \int_{Q} m_{a} \frac{\partial UV}{\partial \xi} d\xi$$

Using the previously described substitutions, the force becomes

$$\begin{split} F_{z} &= \left\{ -M_{a} \cos \theta \ \ddot{z}_{CG} - M_{a} \sin \theta \ \ddot{x}_{CG} + Q_{a} \ddot{\theta} + M_{a} \dot{\theta} (\dot{z}_{CG} \sin \theta - \dot{x}_{CG} \cos \theta) \right. \\ &+ \int_{\varrho} m_{a} \ \frac{d w_{z}}{dt} \cos \theta \ d\xi - \int_{\varrho} m_{a} w_{z} \dot{\theta} \sin \theta \ d\xi \\ &- \int_{\varrho} m_{a} V \ \frac{\partial w_{z}}{\partial \xi} \sin \theta \ d\xi + \int_{\varrho} m_{a} U \ \frac{\partial w_{z}}{\partial \xi} \cos \theta \ d\xi \\ &- UV m_{a} \Big|_{stern} - \int_{\varrho} V \dot{m}_{a} d\xi - \rho \int_{\varrho} C_{D,c} b V^{2} d\xi \Big\} \cos \theta \\ &+ \int_{\varrho} a \rho g A d\xi \end{split}$$

where  $M_a = \int_{Q} m_a d\xi$ 

and

$$Q_a = \int_{\mathcal{Q}} m_a \xi d\xi$$

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

$$m_a = k_a \pi / 2 \rho b^2$$

for which the time derivative is

$$\dot{m}_a = k_a \pi \rho b \dot{b}$$

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 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

$$d_{e} = \pi/2 \, d$$

and

$$b = d_{\rho} \cot \beta = \pi/2 \ d \cot \beta$$

where  $\pi/2$  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

$$\dot{m}_{a} = ka\pi\rho b(\pi/2 \cot\beta)\dot{d}$$

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.,

$$m_a = k \pi/2 \rho b_{max}^2$$
$$\dot{m}_a = 0$$

where b<sub>max</sub> is the half-beam at chine.

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

$$h = z - r$$

where  $z = z_{CG} - \xi \sin \theta + \zeta \cos \theta$ 

$$\mathbf{r} = \mathbf{r}_0 \cos \left\{ k(\mathbf{x}_{CC} + \xi \cos \theta + \zeta \sin \theta) + \omega t \right\}$$

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

$$d \approx \frac{z - r}{\cos \theta - v \sin \theta}$$

where v = wave slope

The rate change of submergence d is given by

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

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

$$\dot{d} \approx \frac{\dot{z} - \dot{r}}{\cos \theta - v \sin \theta}$$

and

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

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

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

$$-\int_{\varrho} m_{a}\cos\theta \, \frac{dw_{z}}{dt} \, \xi \, d\xi + \int_{\varrho} m_{a} \, \dot{\theta} \, \sin\theta \, w_{z} \, \xi \, d\xi$$

$$+ \int_{\varrho} V \dot{m}_{a} \, \xi \, d\xi + \int_{\varrho} \rho \, C_{D} \, b \, V^{2} \, \xi \, d\xi$$

$$+ m_{a} \, UV \, \xi \Big|_{stern} + \int_{\varrho} m_{a} \, V \, U \, d\xi$$

$$+ \int_{\varrho} m_{a} \, V \, \frac{\partial w_{z}}{\partial \xi} \, \sin\theta \, \xi \, d\xi$$

$$- \int_{\varrho} m_{a} \, U \, \frac{\partial w_{z}}{\partial \xi} \, \cos\theta \, \xi \, d\xi$$

$$+ \int_{\varrho} a \rho g A \, \cos\theta \, \xi \, d\xi$$

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**

T

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.

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

Job Control Language Card: CATALOG,TAPE9, SEFZARNICKDATA.., ID=XXXX. 7/8/9 END OF RECORD DATA CARDS (1-5) 6/7/8/9 END OF FILE

## Comment

Catalogues file for plot. (SEFZARNICKDATA CAN BE ANY NAME)

## 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.

## Card 1(NAMELIST FORMAT, / /)

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

# Card 1 (continued)

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

# Card 2 (Format 8F10.0)

(X(I),I=1,6)	Initial conditions
X(1)	Velocity
X(2)	Z
X(3)	θ
X(4)	Х
X(5)	Z
X(6)	$\theta$ 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

<sup>\*</sup>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:	Comment
Job Card	Standard facility card
Charge Card	Standard facility card
REQUEST, TAPE7, HI.	Tape for CALCOMP plot data
VSN(TAPE7=CK0323).	Volume serial number of tape for CALCOMP plot
ATTACH,CALC936.	Attaches CALCOMP library routine
ATTACH,BINAR,SEFZARNICKPLOTB, ID=XXXX.	Attaches plot program run file
LDSET(LIB=CALC936)	Loads CALCOMP library routines
BINAR.	Runs plot program
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)

XAXIS	Length of x axis in inches
YAXISP	Height of pitch component axis in inches
YAXISH	Height of heave component axis in inches
HT	Height of lettering in inches

### Card 2 (I10 Format)

IA If =0, no plots for bow acceleration and center of gravity acceleration If =1, plots previously mentioned information

#### 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  $\lambda/H$  and  $\theta_p/2\pi H/\lambda$  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 b1\*, based on the following equations

$$h_{w}(I) = z_{CG} - \xi(I) \sin \theta + \zeta(I) \cos \theta - r(I)$$

where  $r(I) = r_0 \cos k [x_{CG} + \xi(I) \cos \theta + \zeta(I) \sin \theta + ct]$ 

Then for

$$h_{w}(I) > 0,$$

$$d(I) = \frac{h_{w}(I)}{1 + 1}$$

$$\frac{1}{\cos\theta} = \frac{1}{\cos\theta} = (1)\sin\theta$$

where  $V(I) = -r_0 k \sin \theta [x_{CG} + \xi(I) \cos \theta + (I) \sin \theta + ct]$ 

If

$$d(I) \ge b_m(I) \tan (\beta(I) 2/\pi)$$

set

$$m_{a}(I) = m_{amax}(I)$$

$$b(I) = b_{m}(I)$$

$$b1(I) = 0$$

$$m_{amax}(I) = k(I) (\rho/2) \pi b_{m}^{2}(I)$$
If
$$d(I) < b_{m}(I) \tan (\beta(I)) (2/\pi)$$
set
$$b(I) = d(I) \cot (\beta(I)) (\pi/2)$$

$$b1(I) = b(I)$$

for

$$h_w(I) \le 0;$$
  
 $m_a(I) = 0, \quad b(I) = 0, \quad bl(I) = 0$ 

 $m_a(I) = k_a(I) (\rho/2) \pi b^2(I)$ 

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

$$F_{1} = T_{x} + F_{L} \sin \theta - D \cos \theta$$

$$F_{2} = T_{z} + F_{L} \cos \theta + D \sin \theta + W$$

$$F_{3} = N_{L} - D_{xd} + T_{xp}$$

<sup>\*</sup>b1 array is set up for integrations for portion of hull for which chine is not immersed.

The mass inertia matrix is

 $A_{11} = M + M_a \sin^2 \theta$   $A_{12} = M_a \sin \theta \cos \theta$   $A_{13} = -Q_a \sin \theta$   $A_{21} = A_{12}$   $A_{22} = M + M_a \cos^2 \theta$   $A_{23} = -Q_a \cos \theta$   $A_{31} = A_{13}$   $A_{32} = A_{23}$   $A_{33} = I + I_a$ 

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.

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

#### 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.

$$\begin{split} & \int_{\mathbb{Q}} m_{a} d\xi & \int_{\mathbb{Q}} m_{a} \xi d\xi \\ & \int_{\mathbb{Q}} m_{a} \xi^{2} d\xi & \int_{\mathbb{Q}} m_{a} UV d\xi \\ & \int_{\mathbb{Q}} m_{a} w_{z} d\xi & \int_{\mathbb{Q}} m_{a} w_{z} \xi d\xi \\ & \int_{\mathbb{Q}} m_{a} \frac{dw_{z}}{dt} d\xi & \int_{\mathbb{Q}} m_{a} \frac{dw_{z}}{dt} \xi d\xi \\ & \int_{\mathbb{Q}} m_{a} V \frac{\partial w_{z}}{\partial \xi} d\xi & \int_{\mathbb{Q}} m_{a} V \frac{\partial w_{z}}{\partial \xi} \xi d\xi \\ & \int_{\mathbb{Q}} m_{a} U \frac{\partial w_{z}}{\partial \xi} d\xi & \int_{\mathbb{Q}} m_{a} U \frac{\partial w_{z}}{\partial \xi} \xi d\xi \\ & \int_{\mathbb{Q}} m_{a} V d\xi & \int_{\mathbb{Q}} m_{a} V \xi d\xi \\ & \int_{\mathbb{Q}} b V^{2} d\xi & \int_{\mathbb{Q}} b V^{2} \xi d\xi \\ & \int_{\mathbb{Q}} b \left(h - \frac{b}{2} \tan \beta\right) d\xi & \int_{\mathbb{Q}} b \left(h - \frac{b}{2} \tan \beta\right) \xi d\xi \end{split}$$

### 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
Х	Vector of dependent variables
EPSE	Relative error criteria specified for each component of x and used for the components of x less than the absolute value of A
Α	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, NWAVE)

The routine initializes various values required to generate printer plots and computes pitch-and-heave ratios. The printer plots that are generated consists 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
NWAVE	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, based on a straight line from time START with a value of 0 to time START plus RISE with a value of 1. It is used to lower the initial wave amplitude to avoid large transients at start of the computations.

The arguments are:

Т	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

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

### **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=512.
                                                                               MATN
                                                                                       2
         TAPF2=512.TAPF4=512.TAPF9)
                                                                               ΜΔΤΝ
                                                                                       3
С
                                                                               MATN
                                                                                       4
      REAL IT .K .LAMBDA .M.MA.MMAX .N.NCG .NU.MASS .NL . IA .KAR
                                                                               MATN
                                                                                       5
      INTEGER END
                                                                               MATN
                                                                                       6
с
                                                                               MATN
                                                                                       7
      DIMENSION X(6) .FX(2.400)
                                                                               MATN
                                                                                       Â
                                                                                       ā
с
                                                                               MATN
      COMMON /CONST/ NCG. ECG. PI, DPR. RPD. GRAVTY. RHO.K. NUM. MA(120). CD. TA. MAIN
                                                                                      10
                B(120) .BETA, HW(120) .TZ, URAG. W. XD. T. XP. M. IT.
                                                                               MATN
                                                                                      11
     .
                DELTAS.TX.EST(120).C.RO.KAR.MMAX(1.0).TEST(120).
                                                                               MAIN
     20
                                                                                      12
                     N(120) PHALF
                                                                               MAIN
                                                                                      13
      COMMON /SHIP/ MASS.CINT.QA.CE.CE2.CE3.DMU.EDMU.E2DMU.E3DMU.BF.BMM.MAIN
                                                                                      14
                    NL .FL . IA .E (120)
                                                                               MATN
                                                                                      15
      COMMON /IN/ BM(120) .B1(120) .VELIN
                                                                               MATN
                                                                                      16
                                                                               MATN
      COMMON/ULT/NPRINT .NPLOT . END
                                                                                      17
      COMMON/TERMS/T1+T2+T3+T4+T5+T6+T7+T8
                                                                               MATN
                                                                                      18
      COMMON /SEAWAVE/ START RISE RAMP
                                                                               MATN
                                                                                      19
      COMMON /INTER/ II .KTT(10) .DIFF(10)
                                                                               MATN
                                                                                      20
      COMMON /IN2/ NO(120) . XA . XE . HMAX . HMIN . A (6) . EPSE (6) . LAMBDA
                                                                               MATN
                                                                                      21
      COMMON /ACCEL / XACCL .BWACL .CGACL .BL
                                                                               MATN
                                                                                      22
с
                                                                               MATN
                                                                                      23
      CALL INPUT
                                                                               MATN
                                                                                      24
с
                                                                               MATN
                                                                                      25
С
         COMPUTE INTEGRATION INTERVAL INFORMATION
                                                                               MATN
                                                                                      26
ċ
                                                                               MATN
                                                                                      27
      NLESS = NUM-1
                                                                               MATN
                                                                                      28
      I = 1
                                                                               MAIN
                                                                                      29
      II = 1
                                                                               MATN
                                                                                      30
      DIFFFR = FST(I+1) - FST(I)
                                                                               MATN
                                                                                      31
      KTT(11) = 1
                                                                               MATN
                                                                                      32
      DIFF(II) = DIFFFR
                                                                               MATN
                                                                                      33
                                                                               MAIN
      DO 25 I=2.NLESS
                                                                                      34
                                                                               MAIN
                                                                                      35
      DIFFER= EST(I+1)-EST(I)
                                                                               MAIN
      KTT(II) = KTT(II)+1
                                                                                      36
      IF (DIFFER.NE. UIFF (II)) GO TO 24
                                                                               MAIN
                                                                                      37
      GO TO 25
                                                                               MATN
                                                                                      38
   24 II = II+1
                                                                               MAIN
                                                                                      39
                                                                               MAIN
      KTT(II) = 1
                                                                                      40
      DIFF(II) = DIFFER
                                                                               MATN
                                                                                      41
   25 CONTINUE
                                                                               MAIN
                                                                                      42
      KTT(II) = KTT(II)+1
                                                                               MATN
                                                                                      43
   * * * * CHECK IF NUMBER OF INTERVALS EXCEEDS DIMENSION
С
 MATN
                                                                                      44
      IF (II.GT.10) WRITE(6,28) (KTT(I), DIFF(I), I=1, II)
                                                                               MATN
                                                                                      45
      IF(II.GT.10) STOP 4
                                                                               MATN
                                                                                      46
C
 * * * * * POINT AT WHICH MULTIPLE RUNS START
                                                                               MATN
                                                                                      47
    8 CONTINUE
                                                                               MAIN
                                                                                      48
      TIME=XA
                                                                               MAIN
                                                                                      49
                                                                                      50
      KOUNT=1
                                                                               MATN
      END=END-1
                                                                               MAIN
                                                                                      51
      WRITE(6,39)
                                                                               MATN
                                                                                      52
                                                                               MAIN
                                                                                      53
   39 FORMAT(1H1)
 * * * * * * * * READ IN INITIAL CONDITIONS
                                                                               MAIN
                                                                                      54
С
     X(1) = VELOCITY, X(2) = Z DOT, X(3) = THETA DOT
                                                                               MAIN
                                                                                      55
С
С
      X(4) = X_{9}
                         X(5) = Z_{2}
                                         X(6) = THETA
                                                                               MATN
                                                                                      56
Ċ
       THETA IS PEAD IN DEGREES THEN CONVERTED TO RADIANS IN PROGRAM
                                                                               MAIN
                                                                                      57
                                                                               MAIN
                                                                                      58
С
                                                                                      59
      READ (5.10) (X(I).I=1.6)
                                                                               MAIN
                                                                               MAIN
                                                                                      60
с
```

C         DATA , USED IN RAMP FUNCTION, TO TURN ON WAVE         MAIN 62           READ(5,10)START,RISE         MAIN 63           10 FORMAT(8F10,4)         MAIN 64           C         MAIN 65           WRITE (6.19) START,RISE,KAR         MAIN 66           10 FORMAT("         START,RISE,KAR           MAIN 64         MAIN 66           C         THE IS THE TIME AT WHICH THE INTEGRATION INTERVAL IS           MAIN 70         MAIN 70           THE IS THE TIME AT WHICH THE INTEGRATION INTERVAL IS         MAIN 70           C         THE IS THE TIME AT WHICH THE INTEGRATION INTERVAL SUB-DIVIDE MAIN 72           C         HMN IS THE NEW MAINUM INTERVAL SIZE FOR KUTHER TO SUB-DIVIDE MAIN 72           C         HMN IS THE NEW MAINUM INTERVAL SIZE FOR INTEGRATIAN 74           C         IF F THIS OPTION IS NOT USED SET TME TO THE STOP TIME OF THE RUM HAIN 72           C         THE MAINUM INTERVAL SIZE FOR INTEGRATIAN 74           MIL BE CHANGED FOM *FID4.47         MAIN 74           MUIL BE CHANGED FOM *FID4.47         MAIN 74           C         O' *FID6.41           C         ADJUST THE TIME FOR CHANGE OF INTEGRATION			
C       MAIN 63         10 FORMAT (0F10.4)       MAIN 64         C ****** WRITE OUT THE INPUT VALUES       MAIN 66         19 FORMAT ("START, RISE, KAR       MAIN 66         10 FORMAT ("START, RISE, KAR       MAIN 66         C       TME IS THE TIME AT WHICH THE INTEGRATION INTERVAL IS       MAIN 70         C       TME MAXIMUM INTERVAL SIZE AFTER TIME TME       MAIN 70         C       HNN IS THE NEW MAXIMUM INTERVAL SIZE FOR KUTMER TO SUB-DIVIDE MAIN 73       MAIN 76         C HIF THIS OPTION IS NOT USED SET TME TO THE STOP TIME OF THE RUN MAIN 75       MAIN 76         READ(5.10) TME, MAX, HMX, MMIN HANN       MAIN 76       MAIN 76         WRITE (6.11) TME, HAX, HMX, MMIN HANN       MAIN 77       MAIN 77         WRITE (6.11) TME, HAX, HMX, MMIN HANN       MAIN 77       MAIN 76         WRITE (6.11) TME, HAX, HMX, MMIN HANN       MAIN 77       MAIN 76         WRITE (6.11) TME, HAX, HMX, MMIN HANN       MAIN 76       MAIN 77         WRITE (6.11) TME, HAX, HMX, MMIN HANN       MAIN 77       MAIN 77         WRITE (6.11) TME, HAX, HMX, MMIN HANN       MAIN 76       MAIN 77         WRITE (6.11) THE		MAIN	61
10         FORMAT (0F10.4)         MAIN         64           C         ****         ***         WRITE (6.19)         START RISE,KAR         MAIN         65           0         FORMAT ("         START RISE,KAR         RISE = ",F10.4,/,"         KAR = ",F10.4,/,"         KAR = ",F10.4,/"           ****         MAIN         65           ****         THE IS THE TIME AT WHICH THE INTEGRATION INTERVAL IS         MAIN         70           C         THE IS THE TIME AT WHICH THE INTEGRATION INTERVAL IS         MAIN         70           C         THE IS THE TIME AT WHICH THE INTEGRATION INTERVAL IS         MAIN         70           C         THE IS THE TIME AT WHICH THE INTEGRATION INTERVAL IS         MAIN         70           C         THE NATHUM INTERVAL UP TO         SUBD-DIVIDE         MAIN         71           C         THE MAXIHUM INTERVAL UP TO         SUBD-DIVIDE         MAIN         76           READ(5.10)         THE, FNN, HNN         MAIN         77         71	READ (5,10) START, RISE		
C ****** WRITE OUT THE INDUT VALUES MAIN 65 WRITE(619) START, RISE, KAR HAIN 66 19 FORMAT(" START, RISE, KAR HAIN 67 ***0 THE IS THE TIME AT WHICH THE INTEGRATION INTERVAL IS MAIN 69 C THE IS THE TIME AT WHICH THE INTEGRATION INTERVAL IS MAIN 70 TO HE CHANGED HMX IS THE NEW MAXIMUM INTERVAL SIZE AFTER TIME TME MAIN 70 THE MAXIMUM INTERVAL UP 70 THE MAXIMUM INTERVAL UP 70 TIF THIS OFTION IS NOT USED SET TME TO THE STOP TIME OF THE RN MAIN 76 READ(5,10) TME, HMX, HMN MIN, HMN MAIN 76 READ(5,10) TME, HMX, HMX, HMIN, HMN MIN, HMN MAIN 76 ***0 ***0 MILL BE CHANGED FROM **F10.**,* TO **F10.**,* *** AND THE MINIMUM SIZE FOR HALVING CHANGES FROM **F10.**,* *** AND THE MINIMUM SIZE FOR HALVING CHANGES FROM **F10.**,* *** TO **F10.** *** TO **F10.** *** TO **F10.** *** TO **F10.** C SET SWITCH FOR CALCULATION UF PITCH AND HEAVE RATIOES MAIN 80 *** TO **F10.** *** TO **F10.** C SET SWITCH FOR CALCULATION UF PITCH AND HEAVE RATIOES MAIN 80 IF THE _CALCULATION UF PITCH AND HEAVE RATIOES MAIN 80 FOR CHECK AGAINST TIME IN THE INTEGRATION LOOP MAIN 83 IF THE_CALL TO PLOTER MAIN 84 IF THE_CALL TO PLOTER MAIN 88 IF (THE_EG, 2E) FPT = 1 MAIN 88 IF (THE_EG, 2E) FPT = 1 MAIN 88 IF (THE_EG, 2E) FPT = 1 MAIN 89 ** SET SWITCH FOR CALCULATION UF PITCH AND HEAVE RATIOES MAIN 80 ** SET SWITCH FOR CALCULATION UF PITCH AND HEAVE RATIOES MAIN 89 ** IS EQUAL TO ECC=**F10.**, T**BU MAIN 90 ** IS EQUAL TO			
WRITE (6,19)         START, PRISE, KAR         MAIN         MAIN         66           19         FORMAT("         START = ",FI0.44,/,"         RISE = ",FI0.44,/,"         KAH = ",FI0AIN         67           *.4)         MAIN         68         MAIN         66           C         THE IS THE TIME AT WHICH THE INTEGRATION INTERVAL IS         MAIN         70           C         THE CHANGED         MAIN         77           C         HHX IS THE NEW MAINUM INTERVAL SIZE AFER TIME THE         MAIN         72           C         THE MAXIMUM INTERVAL UP TO         SUBDIVIDE         MAIN         74           C         THE MAXIMUM INTERVAL UP TO         MAIN         76         MAIN         76           C         THE MAXIMUM INTERVAL UP TO         MAIN         76         MAIN         76           READ(5.10) THE, HMX, HMX, HMIN         MAIN         76         MAIN         76           PON WILL BE CHANGED FROM * AFID.4,* TO * FID.4,* TO * FID.4,* TO * MAIN         MAIN         76           ***         TO *FIG.4'         MAIN         77         MAIN         76           ***         TO *FIG.4'         MAIN         77         MAIN         76           ***         TO *FIG.4'         MAIN			
19       FORMAT("       START = ",FI0.4,*,*"       RISE = ",FI0.4,*,*"       KAH = ",FI0MAIN 67         *.*0       MAIN 68         C       THE IS THE TIME AT WHICH THE INTEGRATION INTERVAL IS       MAIN 70         C       THE IS THE TIME AT WHICH THE INTEGRATION INTERVAL IS       MAIN 70         C       THE IS THE NEW MAIHUM INTERVAL SIZE AFTER TIME THE       MAIN 71         C       HHN IS THE NEW MAIHUM INTERVAL UP TO       MAIN 76         C       IF THIS OPTION IS NOT USED SET TME TO THE STOP TIME OF THE RUN MAIN 76       MAIN 76         READ (5.10) THE, HMX, HMN       MAIN 76       MAIN 76         READ (5.10) THE, HMX, HMN, HMN, HMN       MAIN 77       MAIN 76         WRITE (6.11) THE, HMX, HMN, HMN, HMN       MAIN 77       MAIN 77         * FO *, FI0.4, '       MAIN 72.2,* THE MAXIMUM INTERVAL SIZE FOR INTEGRATIAN 79       *00         * ON WILL BE CHANGED FROM *, FI0.4,*,* MAIN, HMN       MAIN 76       MAIN 77         * ON VILL BE CHANGE OF CONT *, FI0.4,*,*       MAIN 80       *10,*,* THE *, FI0.4,*,*         * ON VILL BE CHANGE OF ROM *, FI0.4,*,*       MAIN 80       MAIN 80         * ON VILL BE CHANGE OF ON THE INTERVAL MAIN INTERVAL       MAIN 80         C       TH = TME-(MAX/2,*)       MAIN 80         C       ON NEXT CALL TO PLOTEN       MAIN 80			
***)			
C THE IS THE TIME AT WHICH THE INTEGRATION INTERVAL IS MAIN 70 TO HE CHANGED MAINUM INTERVAL SIZE AFTER TIME THE MAIN 71 HMX IS THE NEW MAXINUM INTERVAL SIZE FOR KUITHER TO SUB-DIVIDE MAIN 72 HMX IS THE NEW MINIMUM INTERVAL SIZE FOR KUITHER TO SUB-DIVIDE MAIN 73 THE MAXIMUM INTERVAL UP 70 IF THIS OFILON IS NOT USED SET THE TO THE STOP TIME OF THE RUN MAIN 75 READ (5,10) THE, HMX, HMN MINITHERVAL SIZE FOR KUITHER TO SUB-DIVIDE MAIN 77 WRITE (6,11) THE, HMX, HMN MINITHERVAL SIZE FOR INTEGRATIMAIN 79 *00N WILL BE CHANGED FROM *, FI0.4,* TO *, FI0.4,*, MAIN 81 * * TO *, FI0.4) C ADJUST THE TIME FOR CHANGE OF INTEGRATION INTERVAL MAIN 83 FOR CHECK AGAINST TIME IN THE INTEGRATION INTERVAL MAIN 84 FOR CHECK AGAINST TIME IN THE INTEGRATION INTERVAL MAIN 85 C ADJUST THE THE FOR CHANGE OF INTEGRATION INTERVAL MAIN 86 C ADJUST THE THE FOR CHANGE OF INTEGRATION INTERVAL MAIN 85 C SET SWITCH FOR CALCULATION UF PITCH AND HEAVE RATIOES MAIN 86 C ON NEXT CALL TO PLOTER MAIN 87 IF THE 0 HATCH THE INTEGRATION INTERVAL MAIN 88 IF (THE, E0, KE) IPT = 1 MAIN 87 C C MAIN 87 C MAIN 89 WRITE (6.12) PERCNT MAIN 80 C MAIN 90 READ (5,10) PERCNT MAIN 90 READ (5,10) PERCNT MAIN 90 WRITE (6.23) MAIN 90 WRITE (6.47) C WRITE (6.47) C WRITE (6.47) MAIN 90 WRITE (6.47) C WRITE (6.47) C WRITE (6.47) MAIN 90 WRITE (6.47) C WRITE (6.47) C * * * * * CHANGE INPUT FROM DEGRES TO RADIANS MAIN 94 * IS EQUAL TO ECG-*,FI0.4,7THEBL OR ,FI0.4) MAIN 99 47 FORMAT(* TYALEST(I),NU(I),BM(I)),I=1,NUM) MAIN 103 WRITE (6.455) (1(1),I=1,6) 55 FORMAT(* XUALUES;*,4X,6(F10,4+2X)) C * * * * * * CHANGE INPUT FROM DEGRES TO RADIANS MAIN 103 WRITE (6.55) (X(I),I=1,6) 56 FORMAT(* XUALUES;*,4X,6(F10,4+2X)) C * * * * * * CHANGE INPUT FROM DEGRES TO RADIANS MAIN 106 X(3) = X(3) *RPD MAIN FOR MAIN 107 WAIN 105 FIGHAT(* XUALUES;*,4X,6(F10,4+2X)) C * * * * * * * WRITE OUT COMPUTED ARAYS MAIN 108 X(6) = X(6) *RPD MAIN 100 WAIN 104 WRITE (6,55) (X(I),I=1,NUM) MAIN 116 WAIN 105 C * * * * * * * WRITE OUT COMPUTED ARAYS MAIN 116			
C       TME 15 THE TIRE AT WHICH THE INTEGRATION INTERVAL IS       MAIN 70         C       TO GE CHANGED       MAIN 71         HMX 15 THE NEW MAXINUM INTERVAL SIZE AFTER TIME THE       MAIN 72         THE MAXIMUM INTERVAL UP TO       NAIN 73         THE MAXIMUM INTERVAL UP TO       MAIN 76         G       IF THIS OPTION IS NOT USED SET TME TO THE STOP TIME OF THE RUN MAIN 76         READ(5.10) TME.HMX.HMN       MAIN 77         WRITE(6.11) TME.HMX.HMN       MAIN 77         TO TO HE HMX.HMN       MAIN 76         #AND 71       MAIN 76         #AND 71       MAIN 76         #AND 71       MAIN 76         #AND 71       MAIN 76         #AND 710       MAIN 76         #AND 710       MAIN 76         #AND 710       MAIN 76         #AND 710       MAIN 77         WRITE(6.11) TME.HMX.HMN       MAIN 76         #AND 711       #AND 76         #AND 712       #AND 71         #AND 714       #AND 71         #AND 715       #AND 71         #AND 71       #AND 71			
C         TO HE CHANGED         MAIN 72           C         HHX IS THE NEW MAXIMUM INTERVAL SIZE AFTER TIME TME         MAIN 72           C         HHX IS THE NEW MAXIMUM INTERVAL SIZE FOR KUTMER TO SUB-DIVIDE         MAIN 72           C         THE MAXIMUM INTERVAL UP TO         MAIN 72           C         IF THIS OPTION IS NOT USED SET TME TO THE STOP TIME OF THE NUM MAIN 75         MAIN 76           C         READ(5,10) TME,HMX,HMN, MIN,HNN         MAIN 77           WRITE(6,11) TME,HMX,HMN,HMIN,HNN         MAIN 77           WRITE(6,11) TME,HMX,HMN,HMIN,HNN         MAIN 77           WRITE(6,11) TME,HMX,HMN,HMIN,HNN         MAIN 77           MAN 77         MAIN 77           WRITE(6,11) TME,HMX,HMN,HMIN,HNN         MAIN 77           WAILL BE CHANGED FROM +FI0,4,* TO *F10,4,*         MAIN 77           MAIN 72         MAIN 72           * AND THE MINHUM SIZE FOR HALVING CHANGES FROM *F10,4,*         MAIN 80           C         ADJUST THE THE THE TOR CHANGE OF INTEGRATION INTERVAL         MAIN 83           C         SET SWITCH FOR CALCULATION UF PITCH AND HEAVE RATIOES         MAIN 83           C         SET SWITCH FOR CALCULATION UF PITCH AND HEAVE RATIOES         MAIN 83           IF (TME,E0,2KE) IPT = 1         MAIN 83         MAIN 83           C         SET SWITCH FOR CALCU			
C HMX IS THE NEW MAXIMUM INTERVAL SIZE AFTER TIME TME MAIN 72 C HMX IS THE NEW MINIMUM INTERVAL SIZE FOR KUTHER TO SUB-DIVIDE MAIN 73 THE MAXIMUM INTERVAL UP TO IF THIS OPTION IS NOT USED SET TME TO THE STOP TIME OF THE RUN MAIN 75 READ(5,10) TME,HMX,HMN MAIN NTERVAL SIZE FOR INTEGRATIMAIN 79 "ON WILL BE CHANGED FROM *,FI0.4,* TO *,FI0.4,*,* MAIN 81 * AND THE MINIMUM SIZE FUR HALVING CHANGES FROM *,FI0.4,* MAIN 80 * AND THE MINIMUM SIZE FUR HALVING CHANGES FROM *,FI0.4,* MAIN 80 * AND THE MINIMUM SIZE FUR HALVING CHANGES FROM *,FI0.4,* MAIN 80 C ADJUST THE TIME FOR CHANGE OF INTEGRATION INTERVAL MAIN 82 FOR CHECK AGAINST IME IN THE INTEGATION LOOP MAIN 82 C SET SWITCH FOR CALCULATION UF PITCH AND HEAVE RATIOES MAIN 83 IPT = 0 MAIN 84 IPT = 0 MAIN 84 IPT = 0 MAIN 87 C READ(5,10) PERCNT XACCL MAIN 95 IZ FORMAT(* THE X USED FOR THE 80W AND CG ACCELERATION COMPUTATIONS MAIN 93 IZ FORMAT(* THE X USED FOR THE 80W AND CG ACCELERATION COMPUTATIONS MAIN 95 WRITE(6,47) MAIN 93 IZ FORMAT(* THE X USED FOR THE 80W AND CG ACCELERATION COMPUTATIONS MAIN 95 MAIN 95 C ****** CHANGE FOI.(1),+7)***********************************			
C HHN IS THE NEW MINIMUM INTERVAL SIZE FOR KUTHER TO SUB-DIVIDE MAIN 73 THE MAXIMUM INTERVAL UP TO MAIN 76 IF THIS OPTION IS NOT USED SET THE TO THE STOP TIME OF THE RUN MAIN 75 READ(5,10) THE,HMX,HMN MAIN,HMIN,HMN MAIN 77 WRITE(6,11) THE,HMX,HMX,HMX,HMIN,HMN MAIN 77 WRITE(6,11) THE,HMX,HMX,HMX,HMIN,HMN MAIN 77 ** AND THE HINIMUM SIZE FUR HAXIMUM INTERVAL SIZE FOR INTEGRATIMAIN 79 ** AND THE HINIMUM SIZE FUR HALVING CHANGES FROM *,F10,4, ** AND THE HINIMUM SIZE FUR HALVING CHANGES FROM *,F10,4, ** AND THE HINIMUM SIZE FUR HALVING CHANGES FROM *,F10,4, ** TO *,F10,4) ** TO *,F10,4) C ADJUST THE TIME FOR CHANGE OF INTEGRATION INTERVAL MAIN 83 FOR CHECK AGAINST TIME IN THE INTEGRATION LOOP MAIN 86 TH = THE-(HMAX/2,) C SET SWITCH FOR CALCULATION UF PITCH AND HEAVE RATIOES MAIN 86 C ON NEXT CALL TO PLOTER MAIN 88 IF (THE,E0,XE) IPT = 1 C MAIN 88 IF (THE,E0,XE) IPT = 1 C MAIN 89 VALCE ECG-PERCNT*8L WAIN 92 VALCE (= ECG-PERCNT*8L WAIN 92 VALCE (= CG-PERCNT*8L WAIN 92 VALCE (= CG-PERCNT*8L WAIN 92 VALTE(6,47) VRITE(6,47) VRITE(6,47) VRITE(6,47) C WITE(6,52) ((1,BETA,EST(I),NU(I),BM(I)),I=1,NUM) MAIN 93 23 FORMAT(4 THE X USED FOR THE BOW AND CG ACCELERATION COMPUTATIONS MAIN 94 * 10X,*BEAM") MAIN 95 C WRITE(6,52) ((1,BETA,EST(I),NU(I),BM(I)),I=1,NUM) MAIN 102 * 10X,*BEAM") MAIN 95 C WRITE(6,55) ((1,1,1=1,6) C * * * * * * CHANGE INPUT FROM DEGREES TO RADIANS MAIN 106 C * C * * * * * CHANGE INPUT FROM DEGREES TO RADIANS MAIN 107 X(3) = X(3)*PPD X(3) = X(3)*PPD MAIN 106 C * * * * * * WRITE OUT COMPUTED ARRAYS MAIN 113 WRITE (6,57) (M,II,1=1,NUM) MAIN 105 C * * * * * * WRITE OUT COMPUTED ARRAYS MAIN 116 WAIN 126 C * * * * * * WRITE OUT COMPUTED ARRAYS MAIN 116 WAIN 126 WAIN 126 W	C ID HE CHANGED		
C       THE MAXIAUM INTERVAL UP TO       MAIN 75         G       IF THIS OPTION IS NOT USED SET THE TO THE STOP TIME OF THE RUM MAIN 75         READ(5,10) THE, HMAX, HMA, HMIN, HMN       MAIN 77         11 FORMAT(* AT TIME *, FT, 2,* THE MAXIMUM INTERVAL SIZE FOR INTEGRATIMAIN 78         11 FORMAT(* AT TIME *, FT, 2,* THE MAXIMUM INTERVAL SIZE FOR INTEGRATIMAIN 78         ** AND THE MINIHUM SIZE FUR HALVING CHANGES FROM *, F10,4,*       MAIN 80         ** AND THE MINIHUM SIZE FUR HALVING CHANGES FROM *, F10,4,*       MAIN 82         C       ADJUST THE TIME FOR CHANGE OF INTEGRATION INTERVAL       MAIN 82         C       ADJUST THE TIME FOR CHANGE OF INTEGRATION LOOP       MAIN 84         TH = TME-(HMAX/2,)       MAIN 86         C       ON NEXT CALL TO PLOTER       MAIN 87         IPT = 0       MAIN 86         ON NEXT CALL TO PLOTER       MAIN 87         IPT = 0       MAIN 87         READ (5,10) PERCNT       MAIN 87         XACCL = ECG-PERCNT*8L       MAIN 93         12 FORMAT(* THE X USED FOR THE BOW AND CG ACCELERATION COMPUTATIONS MAIN 94         WRITE (6,23)       MAIN 95         WRITE (6,47)       MAIN 95         VRITE (6,52) ((1), FTA+551 (I), NU (I) + BM (I)) + I=1 + NUM)       MAIN 102         * 0 * * * * * CHANGE LIPUT FROM DEGREES TO RADIANS       MAIN 105 </td <td>C INTA IS THE NEW MAXIMUM INTERVAL SIZE AFTER THE THE</td> <td></td> <td></td>	C INTA IS THE NEW MAXIMUM INTERVAL SIZE AFTER THE THE		
C IF THIS OPTION IS NOT USED SET TME TO THE STOP TIME OF THE RUN MAIN 75 READ(5,10) TME,HMX,HMN, MAIN,HMN,HMIN,HMN MAIN 77 WRITE(6(511) TME,HMX,HMX,HMX,HMIN,HMN,HMN MAIN 77 WRITE(6(511) TME,HMX,HMX,HMX,HMIN,HMN MIN,TREVAL SIZE FOR INTEGRATIMAIN 79 *ON WILL BE CHANGED FROM *;F10.4,* TO *;F10.4,*,* MAIN 80 ** AND THE HINIMUM SIZE FUR HALVING CHANGES FROM *;F10.4, MAIN 80 ** TO *;F10.4) MAIN 62 ADJUST THE TIME FOR CHANGE OF INTEGRATION INTERVAL ADJUST THE TIME FOR CHANGE OF INTEGRATION LOOP MAIN 82 FOR CHECK AGAINST TIME IN THE INTEGARTION LOOP MAIN 83 C ADJUST THE TIME FOR CLAULATION UF PITCH AND HEAVE RATIOES MAIN 86 IF (TME,ECO,XE) IPT = 1 MAIN 87 PT = 0 READ(5,10) PERCNT XACCL = ECG-PERCNT*8L WRITE(6,12) PERCNT*ALCL WRITE(6,47) WRITE(6,47) WRITE(6,47) MAIN 90 47 FORMAT(" STATION NO.",3X,"DEAU RISE",8X,"EST",8X,"NO", MAIN 90 47 FORMAT(", STATION NO.",3X,"DEAU RISE",8X,"EST",8X,"NO", MAIN 90 WRITE(6,55) ((1)6ETA+EST(1),NU(1)+BM(1)),I=1+NUM) MAIN 92 WRITE(6,55) ((1)6ETA+EST(1),NU(1)+BM(1)),I=1+NUM) MAIN 103 WRITE(6,55) ((1)6ETA+EST(1),NU(1)+BM(1)),I=1+NUM) MAIN 103 WRITE(6,55) ((1),1=1,5) MAIN 92 C * * * * * * CHANGE INPUT FROM DEGREES TO RADIANS MAIN 96 C * * * * * * CHANGE INPUT FROM DEGREES TO RADIANS MAIN 103 WRITE(6,55) ((1),1=1,5) MAIN 104 WRITE(6,55) (1),1=1,5) MAIN 105 C * * * * * * CHANGE INPUT FROM DEGREES TO RADIANS MAIN 105 C * * * * * * CHANGE INPUT FROM DEGREES TO RADIANS MAIN 107 X(3) = X(3) *RD MAIN 105 FORMAT(CX,12,5X,F10,4,4X,F10,4,4X,F10,4,3X,F10,4) MAIN 105 C * * * * * * CHANGE INPUT FROM DEGREES TO RADIANS MAIN 104 WRITE (6,55) (K(1),1=1,5) MAIN 105 MAIN 105 MAIN 105 MAIN 106 MAIN 107 MAIN 106 MAIN 107 MAIN 106 MAIN 107 MAIN 107 MAIN 106 MAIN 107 MAIN 106 MAIN 106 MAIN 106 MAIN 106 MAIN 107 MAIN 107 MAIN 106 MAIN 106 MAIN 106 MAIN 106 MAIN 107 MAIN 106 MAIN 107 MAIN 106 MAIN 106 MAIN 106 MAIN 106 MAIN 106 MAIN 106 MAIN 107 MAIN 106 MAIN 106 MAIN 106 MAIN 106 MAIN 106 MAIN 106 MAIN 106 MAIN 106 MAIN 106 MAIN 106 MAI	C THE MAYIMUM INTERVAL SIZE FOR NUMBER TO SUB-DIVIDE		
C READ(5,10) TME,HMX,HMN WRITE(6,11) TME,HMX,HMN,HMIN,HMN WRITE(6,11) TME,HMAX,HMX,HMIN,HMN 11 FORMAT(* AT TIME *,F7,2,* THE MAXIMUM INTERVAL SIZE FOR INTEGRATIMAIN 70 ** AND THE MINIMUM SIZE FOR HALVING CHANGES FROM *,F10,4, ** AND THE MINIMUM SIZE FOR HALVING CHANGES FROM *,F10,4, MAIN 82 ** AND THE MINIMUM SIZE FOR HALVING CHANGES FROM *,F10,4, MAIN 82 C ADJUST THE TIME FOR CHANGE OF INTEGRATION INTERVAL ADJUST THE TIME FOR CHANGE OF INTEGRATION LOOP MAIN 83 C FOR CHECK AGAINST TIME IN THE INTEGARTION LOOP MAIN 86 ON NEXT CALL TO PLOTER MAIN 86 ON NEXT CALL TO PLOTER MAIN 87 IPT = 0 MAIN 87 C READ(5,10) PERCNT XACCL = ECG-PERCNT/86L WRITE(6,12) PERCNT,XACCL WRITE(6,12) PERCNT,XACCL WRITE(6,12) PERCNT,XACCL WRITE(6,12) PERCNT,XACCL MAIN 95 C WRITE(6,12) PERCNT,XACCL WRITE(6,12) MAIN 99 47 FORMAT(* TATION NO,*,3X,*DEAU RISE*,8X,*EST*,8X,*NO*, * MAIN 98 WRITE(6,55) ((1,9ETA,EST(1),NU(1),8M(1)),1=1,NUM) WRITE(6,55) ((1,1,1=1,50) C ** ** ** * CHANGE INPUT FROM DEGREES TO RADIANS MAIN 102 WRITE(6,55) ((1,1,1=1,50) C ** ** ** * WRITE OUT COMPUTED ARRAYS MAIN 103 WRITE (6,55) ((1,1,1=1,NUM) WAIN 110 WAIN 126 C ** ** ** WRITE OUT COMPUTED ARRAYS MAIN 111 WAVE = 0 C ** ** ** WRITE OUT COMPUTED ARRAYS MAIN 116 WRITE (6,56) ((1,1,1=1,NUM) WAIN 117 WAIN 116 WRITE (6,56) ((1,1,1=1,NUM) WAIN 116 WRITE (6,56) ((1,1	C IF THIS OPTION IS NOT USED SET THE TO THE STOP TIME OF THE DUN		
READ(5,10) TME,HMX,HMX       MAIN       77         WRITE(6,11) TME,HMX,HMX,HMIN,HMN       MAIN       78         11 FORMAT(* AT TIME *,F7,2,* THE MAXIMUM INTERVAL SIZE FOR INTEGRATIMAN       79         *00N WILL BE CHANGED FROM *,F10,4,* TO *,F10,4,*,       MAIN       80         ** AND THE HINIMUM SIZE FOR HALVING CHANGES FROM *,F10,4,       MAIN       81         ** TO *,F10.4)       MAIN       82         C       ADJUST THE TIME FOR CHANGE OF INTEGRATION INTERVAL       MAIN         TM = TME-(HMAX/2,)       MAIN       83         C       SET SWITCH FOR CALCULATION UF PITCH AND HEAVE RATIOES       MAIN         TPT = 0       MAIN       86         IPT = 0       MAIN       87         IPT = 0       MAIN       88         IF (TME,EQ.XE) IPT = 1       MAIN       88         C       READ(5,10) PERCNT       MAIN       90         XACCL = ECG-PERCNT*BL       MAIN       91         XACCL = ECG-PERCNT*BL       MAIN       93         12 FORMAT(* THE X USED FOR THE BOW AND CG ACCELERATION COMPUTATIONS MAIN       94         *15 EQUAL TO ECG-*F.F10.4.*71*BL OR *F10.4.)       MAIN       95         C       WRITE(6,6.23)       MAIN       97         WRITE(6.6.23)       MAIN			
WRITE (6.11) TME,HMAX,HMIX,HMIN,HMN       MAIN 78         11 FORMAT(** AT TIME +F7.2.* THE MAXIMUM INTERVAL SIZE FOR INTEGRATIMAIN 79         *ON WILL BE CHANGED FROM **F10.4* TO **F10.4*,*       MAIN 80         ** AND THE MINIHUM SIZE FOR HALVING CHANGES FROM **F10.4*,       MAIN 81         ** AND THE MINIHUM SIZE FOR HALVING CHANGES FROM **F10.4*,       MAIN 82         C       ADJUST THE TIME FOR CHANGE OF INTEGRATION INTERVAL       MAIN 82         C       ADJUST THE TIME FOR CHANGE OF INTEGRATION INTERVAL       MAIN 82         C       ADJUST THE TIME FOR CHANGE OF INTEGRATION LOOP       MAIN 85         C       SET SWITCH FOR CALCULATION UF PITCH AND HEAVE RATIOES       MAIN 85         C       SET SWITCH FOR CALCULATION UF PITCH AND HEAVE RATIOES       MAIN 85         ON NEXT CALL TO PLOTER       MAIN 86       MAIN 86         IF (THME.EO.*E) IPT = 1       MAIN 87       MAIN 89         C       READ(5,10) PERCNT       MAIN 92       MAIN 92         WRITE(6.12) PERCNT,XACCL       MAIN 93       MAIN 93         VRITE(6.423)       WAIN 100 FFIO.4*,7H*8L OR *FIO.4*)       MAIN 93         VRITE(6.421)       MAIN 93       MAIN 93         VRITE(6.520)       MAIN 93       MAIN 93         VRITE(6.421)       MAIN 93       MAIN 93         VRITE(6.550)			
11 FORMAT(* AT TIME *,F7,2,* THE MAXIMUM INTERVAL SIZE FOR INTEGRATIMAIN 79 *ON WILL BE CHANGED FROM *,F10.40,* TO *,F10.40,* MAIN 80 ** AND THE MINIMUM SIZE FOR HALVING CHANGES FROM *,F10.40, MAIN 81 ** TO *,F10.40, MAIN 83 C ADJUST THE TIME FOR CHANGE OF INTEGRATION INTERVAL MAIN 83 FOR CHECK AGAINST TIME IN THE INTEGATION LOOP MAIN 86 C SET SWITCH FOR CALCULATION UF PITCH AND HEAVE RATIOES MAIN 86 C ON NEXT CALL TO PLOTER MAIN 180 FOR CHECK AGAINST TIME IN THE INTEGATION LOOP MAIN 86 C ON NEXT CALL TO PLOTER MAIN 86 C ON NEXT CALL TO PLOTER MAIN 86 IF (THE,E0,XE) IPT = 1 MAIN 86 C READ (5,10) PERCNT MAIN 86 C READ (5,10) PERCNT MAIN 80 IF (THE,E0,XE) IPT = 1 MAIN 89 WRITE (6,12) PERCNT,XACCL MAIN 99 WRITE (6,23) MAIN 99 WRITE (6,71) PERCNT,XACCL MAIN 92 WRITE (6,71) PERCNT,XACCL MAIN 92 WRITE (6,71) PERCNT,XACCL MAIN 92 WRITE (6,71) MAIN 93 C MAIN 92 C MAIN 92 C MAIN 92 C MRITE (6,623) MAIN 92 WRITE (6,67) MAIN 92 C MRITE (6,57) (I.BETA,EST (I),NO (I),BM (I)),I=1,NUM) MAIN 92 C MRITE (6,55) ((I.BETA,EST (I),NO (I),BM (I)),I=1,NUM) MAIN 92 C MAIN			
<pre>*0N WILL BE CHANGED FROM *,Fl0,4,* TO *,Fl0,4,*, MAIN 80 ** AND THE MINIMUM SIZE FOR HALVING CHANGES FROM *,Fl0,4, MAIN 81 * TO *,Fl0,4, MAIN 82 C ADJUST THE TIME FOR CHANGE OF INTEGRATION INTERVAL MAIN 82 C ADJUST THE TIME FOR CHANGE OF INTEGARTION INTERVAL MAIN 83 C FOR CHECK AGAINST TIME IN THE INTEGARTION LOOP MAIN 84 TM = TME-(HMAX/2,) MAIN 85 C SET SWITCH FOR CALCULATION UF PITCH AND HEAVE RATIOES MAIN 85 C ON NEXT CALL TO PLOTER MAIN 87 IPT = 0 MAIN 87 READ(5,10) PERCNT MAIN 89 READ(5,10) PERCNT MAIN 89 VRITE(6,12) PERCNT MAIN 99 VRITE(6,12) PERCNT MAIN 99 VRITE(6,12) PERCNT *,XACCL MAIN 99 VRITE(6,23) MAIN 99 23 FORMAT(" STATION NO.",3X,"DEAU RISE",8X,"EST",8X,"NO", MAIN 99 23 FORMAT(" STATION NO.",3X,"DEAU RISE",8X,"EST",8X,"NO", MAIN 99 C MAIN 65 S FORMAT(" STATION NO.",3X,"DEAU RISE",8X,"EST",8X,"NO", MAIN 100 * 10X,"BEAMMI WRITE(6,55) ((I) HETA,EST(I),NO(I) +BM(I)),I=1,NUM) MAIN 102 S FORMAT(" X VALUES",4X,6(FI0,4+2X,FI0,4) MAIN 103 WRITE(6,55) ((XI),1=1,6) MAIN 105 C * * * * * WRITE OUT COMPUTED ARRAYS MAIN 107 C WAVE = STAUTARISE NUME FOR DEGREES TO RADIANS MAIN 108 C * * * * * WRITE OUT COMPUTED ARRAYS MAIN 107 C * * * * * WRITE OUT COMPUTED ARRAYS MAIN 107 MAIN 102 C * * * * * WRITE OUT COMPUTED ARRAYS MAIN 107 MAIN 102 C * * * * * WRITE OUT COMPUTED ARRAYS MAIN 107 MAIN 104 MAIN 105 MAIN 105 C * * * * * WRITE OUT COMPUTED ARRAYS MAIN 107 MAIN 104 MAIN 105 MAIN 10</pre>			
<pre>** AND THE MINIMUM SIZE FOR HALVING CHANGES FROM *,F10,4, MAIN 81 * * TO *,F10,4) MAIN 82 C #DJUST THE TIME FOR CHANGE OF INTEGRATION INTERVAL MAIN 83 C #DOUST THE TIME FOR CHANGE OF INTEGRATION LOOP MAIN 86 T # = THE-(HMAX/2,) MAIN 85 C SET SWITCH FOR CALCULATION UF PITCH AND HEAVE RATIOES MAIN 86 C ON NEXT CALL TO PLOTER MAIN 87 IPT = 0 MAIN 88 IF(TME,EQ,XE) IPT = 1 MAIN 88 C #AIN 99 C #AIN 99 C #AIN 90 C #AIN 92 WRITE(6,21) #ERCNT, #ACCL #AIN 92 WRITE(6,55) ((I,BETA,EST(I),NU(I),BM(I)),I=1,NUM) #AIN 90 VRITE(6,55) ((I,BETA,EST(I),AU(I),BM(I)),I=1,NUM) #AIN 101 #AIN 102 WRITE(6,55) ((I,BETA,EST(I),AU(I),BM(I)),I=1,NUM) #AIN 103 WRITE(6,55) ((I,BETA,EST(I),AU(I),BM(I)),I=1,NUM) #AIN 104 KITE(6,55) ((I,BETA,EST(I),AU(I),BM(I)),I=1,NUM) #AIN 105 C * * * * * &amp; CHANGE INPUT FROM DEGREES TO RADIANS #AIN 105 C * * * * * &amp; WRITE OUT COMPUTED ARRAYS #AIN 105 AIN 90 C #AIN 105 C * * * * * WRITE OUT COMPUTED ARRAYS #AIN 105 #</pre>			
<pre>MAIN 82 C ADJUST THE TIME FOR CHANGE OF INTEGRATION INTERVAL ADJUST THE TIME FOR CHANGE OF INTEGRATION INTERVAL ADJUST THE TIME FOR CHANGE OF INTEGRATION LOOP MAIN 85 FOR CHECK AGAINST TIME IN THE INTEGARTION LOOP MAIN 86 TM = TME-(HMAX/2.) MAIN 86 C SET SWITCH FOR CALCULATION UF PITCH AND HEAVE RATIOES MAIN 86 C ON NEXT CALL TO PLOTER MAIN 86 IPT = 0 MAIN 86 C ON NEXT CALL TO PLOTER MAIN 87 IPT = 0 MAIN 88 C READ(5,10) PERCNT MAIN 89 C READ(5,10) PERCNT MAIN 92 WRITE(6,12) PERCNT,XACCL MAIN 92 WRITE(6,12) PERCNT,XACCL MAIN 92 WRITE(6,12) PERCNT,XACCL MAIN 92 WRITE(6,23) MAIN 95 C WRITE(6,23) MAIN 96 47 FORMAT(" STATION NO.",3X,"DEAU RISE",8X,"EST",8X,"NO", MAIN 96 47 FORMAT(" STATION NO.",3X,"DEAU RISE",8X,"EST",8X,"NO", MAIN 100 * 10X,"BEAM" MAIN 102 S5 FORMAT(6X,12:5X;F10.4;4X;F10.4;4X;F10.4;3X;F10.4) MAIN 105 S5 FORMAT(1', X VALUES",4X,46(F10.4;4X;F10.4;3X;F10.4) MAIN 105 C C WAITE (6,55) ((1),I=1,6) MAIN 106 X(6) = X(6) #RPD ADD ADD ADD ADD ADD ADD ADD ADD ADD A</pre>	** AND THE MINIMUM SIZE FUR HALVING CHANGES FROM *.F10.4.		
C       ADJUST THE TIME FOR CHANGE OF INTEGRATION INTERVAL       MAIN 05         C       FOR CHECK AGAINST TIME IN THE INTEGRATION LOOP       MAIN 06         TM = TME-(HMAX/2,)       MAIN 05         C       SET SWITCH FOR CALCULATION UF PITCH AND HEAVE RATIOES       MAIN 06         ON NEXT CALL TO PLOTER       MAIN 08         IPT = 0       MAIN 08         IF (TME,EQ.XE) IPT = 1       MAIN 08         C       MAIN 08         C       READ(5,10) PERCNT       MAIN 91         XACCL = ECG-PERCNT#BL       MAIN 92         wRITE(6,12) PERCNT,XACCL       MAIN 95         *IS EQUAL TO ECG-*,FI0.4+,7H*BL OR ,FI0.4)       MAIN 95         C       WRITE(6,47)       MAIN 97         WRITE(6,47)       MAIN 98         23 FORMAT(1H ,//)       MAIN 98         47 FORMAT(* STATION NO.*,3X,*"DEAU RISE",0X,*"EST",8X,*"NO",       MAIN 100         * 10X,*"BEAM"       MAIN 101         WRITE(6,55) ((I,BETA,EST(I),NU(I).BM(I)).JI=1,NUM)       MAIN 102         S5 FORMAT(* VALUES",4X,4K;FI0.4+4X,FI0.4+3X,FI0	* * TO *,F10.4)		
TM = TME-(HMAX/2.)       MAIN 85         C       SET SWITCH FOR CALCULATION UF PITCH AND HEAVE RATIOES       MAIN 86         IPT = 0       MAIN 87         IPT = 0       MAIN 88         IF(TME,E0.XE) IPT = 1       MAIN 89         C       MAIN 90         READ(5.10) PERCNT       MAIN 90         XACCL = ECG-PERCNT*8L       MAIN 92         wRITE(6.12) PERCNT,XACCL       MAIN 92         wRITE(6.12) PERCNT,XACCL       MAIN 92         wRITE(6.47)       MAIN 96         •IS EQUAL TO ECG-**,FI0.4.*,7M*8L OR *,FI0.4)       MAIN 96         wRITE(6.47)       MAIN 97         wRITE(6.47)       MAIN 97         wRITE(6.55) ((I:BETA*EST(I)*NO.", 3X,"DEAU RISE", 8X,"EST", 8X,"NO",       MAIN 98         23 FORMAT(" STATION NO.", 3X,"DEAU RISE", 8X,"EST", 8X,"NO",       MAIN 100         * 10X,"BEAM")       MAIN 102       MAIN 102         S5 FORMAT(" X VALUES", 4X,6(F10.4.*,4X,F10.4.*,3X,F10.4.*)       MAIN 103         wRITE(6.55) (X(I),I=I=1,6)       MAIN 106         C * * * * * * CHANGE INPUT FROM DEGREES TO RADIANS       MAIN 106         C * * * * * * * CHANGE INPUT FROM DEGREES TO RADIANS       MAIN 106         C * * * * * * * WRITE OUT COMPUTED ARRAYS       MAIN 108         A(6) = X(6) * PPD       MAIN 112	C ADJUST THE TIME FOR CHANGE OF INTEGRATION INTERVAL		
C       SET SWITCH FOR CALCULATION UF PITCH AND HEAVE RATIOES       MAIN 86         C       ON NEXT CALL TO PLOTER       MAIN 87         IPT = 0       MAIN 88         IF(THE,EQ.XE) IPT = 1       MAIN 89         C       READ(5,10) PERCNT       MAIN 90         XACCL = ECG-PERCNT+8L       MAIN 92         wRITE(6,12) PERCNT,XACCL       MAIN 92         wRITE(6,12) PERCNT,XACCL       MAIN 92         *IS EQUAL TO ECG-*,FI0.44,7H*BL OR ,FI0.4)       MAIN 95         C       MAIN 95         VRITE(6,47)       MAIN 95         VRITE(6,47)       MAIN 96         VRITE(6,47)       MAIN 97         VRITE(6,47)       MAIN 98         23 FORMAT(" STATION NO.", 3X, "DEAU RISE", 8X, "EST", 8X, "NO",       MAIN 98         vRITE(6,55) ((I,BETA,EST(I),NU(I),BM(I)),I=1+NUM)       MAIN 100         vRITE(6,55) ((I,BETA,EST(I),NU(I),BM(I)),I=1+NUM)       MAIN 102         vRITE(6,55) (X(I),I=1,6)       MAIN 103         vRITE(6,55) (X(I),I=1,6)       MAIN 103         vRITE(6,55) (X(I),I=1,6)       MAIN 107         VRITE (6,56) (X(I),I=1,NUM)       MAIN 107         vRITE(6,57)M,II,T,K,C,PHALF,PI,GRAVTY       MAIN 110         VRITE (6,57)M,II,T,K,C,PHALF,PI,GRAVTY       MAIN 114         VRITE (6,58	C FOR CHECK AGAINST TIME IN THE INTÉGARTION LOOP	MAIN	84
C       ON NEXT CALL TO PLOTER       MAIN 87         IPT = 0       MAIN 88         IFT = 0       MAIN 88         C       MAIN 90         READ(5,10) PERCNT       MAIN 90         XACCL = ECG-PERCNT*BL       MAIN 92         WRITE(6,12) PERCNT,XACCL       MAIN 92         WRITE(6,12) PERCNT,XACCL       MAIN 92         ************************************		MAIN	85
IPT = 0       MAIN 88         IF(THE,EQ.XE) IPT = 1       MAIN 89         C       MAIN 90         READ(5,10) PERCNT       MAIN 91         XACCL = ECG-PERCNT*BL       MAIN 91         wRITE(6,12) PERCNT,XACCL       MAIN 93         12 FORMAT(* THE X USED FOR THE BOW AND CG ACCELERATION COMPUTATIONS MAIN 94         •IS EQUAL TO ECG-**F10.4.7M*BL OR *F10.4)       MAIN 95         C       MAIN 96         VRITE(6.23)       MAIN 97         WRITE(6.47)       MAIN 97         VRITE(6.47)       MAIN 97         VRITE(6.55)       MAIN 97         WRITE(6.55)       MAIN 97         WRITE(6.47)       MAIN 97         VRITE(6.55)       MAIN 97         WRITE(6.55)       MAIN 97         WRITE(6.55)       MAIN 97         WRITE(6.55)       MAIN 97         WRITE(6.55)       MAIN 97         WAIN 97       MAIN 97         WRITE(6.55)       MAIN 97         WAIN 97       MAIN 97         WRITE(6.55)       MAIN 98         (1.9ETA*EST(1)*NU(1)*BURISE"*8X,"EST"*8X,"NO"*       MAIN 100         WRITE(6.55)       MAIN 100       MAIN 102         WRITE(6.55)       MAIN 100       MAIN 102 <tr< td=""><td></td><td>MAIN</td><td>86</td></tr<>		MAIN	86
IF(TME,EQ,XE) IPT = 1       MAIN       90         C       MAIN       91         XACCL = ECG-PERCNT*BL       MAIN       92         wRITE(6,12) PERCNT,XACCL       MAIN       93         12 FORMAT(* THE X USED FOR THE BOW AND CG ACCELERATION COMPUTATIONS       MAIN       94         *IS EQUAL TO ECG-**FI0.4+7M*BL OR *F10.4)       MAIN       95         C       WRITE(6+23)       MAIN       98         23 FORMAT(1H, //)       MAIN       97       MAIN       98         47 FORMAT(" STATION NO.", 3X, "DEAU RISE", 8X, "EST", 8X, "NO",       MAIN       98         * 10X,**HEBAM**       MAIN       99         * TORMAT(6+71)       MAIN       98         * SFORMAT(1H, //)       MAIN       98         * 10X,**HEBAM**       MAIN       98         wRITE(6+23)       MAIN       98         wRITE(6+55) ((I+BETA+EST(I),+NU(I)+BM(I))+I=1+NUM)       MAIN       102         S5 FORMAT(6X,12,25X,*F10,4+4X,*F10,4+4X,*F10,4+4X,*F10,4)       MAIN       103         wRITE(6+56) (X(I),I=1+6)       MAIN       103       WAIN       104         wRITE(6+56) (X(I),FET, 4X,*G(F10,4+4X,*F10,4+4X,*F10,4)       MAIN       105       56 FORMAT(" X VALUES",4X,*G(F10,4+4X,*F10,4+4X,*F10,4)       MAIN       105		MAIN	87
C MAIN 90 READ(5,10) PERCNT MAIN 91 XACCL = ECG-PERCNT*BL MAIN 92 WRITE(6,12) PERCNT,XACCL MAIN 93 12 FORMAT(* THE X USED FOR THE BOW AND CG ACCELERATION COMPUTATIONS MAIN 94 •IS EQUAL TO ECG-*,FI0.44,7H*BL OR ,FI0.4) MAIN 95 C MRITE(6.23) MAIN 96 23 FORMAT(1H ,//) MAIN 97 47 FORMAT(" STATION NO.",3X,"DEAU RISE",8X,"EST",8X,"NO", MAIN 99 47 FORMAT(" STATION NO.",3X,"DEAU RISE",8X,"EST",8X,"NO", MAIN 100 * 10X,"BEAM") MAIN 97 47 FORMAT(CX,12.5X,FI0.44,4X,FI0.44,4X,FI0.44,3X,FI0.44) MAIN 102 55 FORMAT(6X,12.5X,FI0.44,4X,FI0.44,4X,FI0.44,3X,FI0.44) MAIN 103 WRITE(6.56) (X(I),I=1,6) MAIN 105 56 FORMAT(" X VALUES",4X,6(FI0.4,2X)) MAIN 106 C * * * * * * CHANGE INPUT FROM DEGREES TO RADIANS MAIN 107 X(3) = X(3)*RPD MAIN 107 C MAIN 108 X(6) = X(6)*RPD MAIN 107 C * * * * * * * WRITE OUT COMPUTED ARRAYS MAIN 112 C * * * * * * * WRITE OUT COMPUTED ARRAYS MAIN 115 MAIN 114 NWAVE = 0 MAIN 104 C * * * * * * * WRITE OUT COMPUTED ARRAYS MAIN 116 MAIN 116			
READ (5,10) PERCNT     MAIN 91       XACCL = ECG-PERCNT*BL     MAIN 92       wRITE(6,12) PERCNT,XACCL     MAIN 93       12 FORMAT(* THE X USED FOR THE BOW AND CG ACCELERATION COMPUTATIONS MAIN 94       *IS EQUAL TO ECG-**F10.4,7M*BL OR *F10.4)     MAIN 95       C     MAIN 97       WRITE(6,23)     MAIN 97       WRITE(6,47)     MAIN 97       23 FORMAT(H ,//)     MAIN 97       47 FORMAT(" STATION NO.",3X,"DEAU RISE",8X,"EST",8X,"NO",     MAIN 100       * 10X,"BEAM")     MAIN 100       * 10X,"BEAM")     MAIN 102       S5 FORMAT(6X,12,5X,FI0.4+4X,FI0.4+4X,FI0.4+3X,FI0.4)     MAIN 103       WRITE(6,55)     (I),BETA,EST(I),NU(I)+BM(I)),I=1,NUM)     MAIN 103       WRITE(6,55)     (I),BETA,EST(I),AU(I)+BM(I)),I=1,NUM)     MAIN 103       WRITE(6,55)     (I),BETA,EST(I),AU(I)+BM(I)),I=1,NUM)     MAIN 102       55 FORMAT(6X,12,5X,FI0.4+4X,FI0.4+4X,FI0.4+3X,FI0.4)     MAIN 103       WRITE(6,56)     (X I),I=1,6)     MAIN 103       WRITE(6,56)     (X I),I=1,6)     MAIN 105       56 FORMAT(" X VALUES",4X,6(F10.4+2X))     MAIN 106       C * * * * * CHANGE INPUT FROM DEGREES TO RADIANS     MAIN 107       X(3) = X(3)*RPD     MAIN 108       X(4) = Z     MAIN 110       C * * * * * WRITE OUT COMPUTED ARRAYS     MAIN 110       C * * * * * WRITE OUT COM			
XACCL = ECG-PERCNT*BL     MAIN     92       WRITE(6,12)     PERCNT,XACCL     MAIN     93       12     FORMAT(*     THE X USED FOR THE BOW AND CG ACCELERATION COMPUTATIONS     MAIN     94       •IS     EQUAL TO ECG-*,FI0.4,7H*BL OR ,F10.4)     MAIN     95       C     MAIN     95       WRITE(6,23)     MAIN     96       23     FORMAT(1+,//)     MAIN     97       * 10X:*BEAM*N)     MAIN     90       * 10X:*BEAM*N)     MAIN     90       WRITE(6,55)     ((I,BETA,EST(I),NU(I),BM(I)),I=1,NUM)     MAIN     102       S5     FORMAT(6×,12,5X:F10.4;4X:F10.4;4X:F10.4;4X:F10.4)     MAIN     103       WRITE(6,56)     (X(I),I=1,6)     MAIN     105       S6     FORMAT(" X VALUES";4X:6(F10.4;2X))     MAIN     105       C * * * * * * * CHANGE INPUT     FROM DEGREES TO RADIANS     MAIN     106       X(6) = X(6)*RPD     MAIN     107     X(3) = X(3)*RPD     MAIN     108       X(6) = X(6)*RPD     MAIN     MAIN     110     MAIN     111       NAVE = 0     C     MAIN     112     MAIN     113       C * * * * * * WRITE OUT COMPUTED ARRAYS     MAIN     MAIN     112       WRITE (6,57)M,IT:*K,C,PMALF,PI;GRAVTY     MAIN     MA			
wRITE(6,12) PERCNT,XACCL       MAIN 93         12 FORMAT(* THE X USED FOR THE BOW AND CG ACCELERATION COMPUTATIONS MAIN 94       *IS EQUAL TO ECG-*,FI0.44,7H*BL OR ;FI0.4)       MAIN 95         c       MAIN 96       MAIN 96         wRITE(6,73)       MAIN 97         v#RITE(6,47)       MAIN 97         c       MAIN 97         v#RITE(6,47)       MAIN 97         c       MAIN 97         v#RITE(6,47)       MAIN 97         c       MAIN 97         v#ITE(6,47)       MAIN 97         c       MAIN 97         v#ITE(6,47)       MAIN 97         v#ITE(6,57)       MAIN 100         v#ITE(6,55)       (I)*BETA*,EST(I)*,9X,"EST",8X,"NO",         wAIN 100       * 10X,"BEAM")       MAIN 101         wRITE(6,55)       (I)*BETA*,EST(I)*,NU(I)*BM(I)*,I=1*,NUM)       MAIN 102         wRITE(6,55)       (XI)*I=1,6)       MAIN 103         wRITE(6,55)       (XI)*I=1,6)       MAIN 106         swRITE(6,56)       (XI)*I=1,6)       MAIN 106         c * * * * * CHANGE INPUT FROM DEGREES TO RADIANS       MAIN 107         x(3) = X(3)*RPD       MAIN 108         X(6) = X(6)*RPD       MAIN 110         wAITE (6,571,M+IT*,K*C*,PMALF*,PI*,GRAVTY       MAIN 111			
12 FORMAT(* THE X USED FOR THE BOW AND CG ACCELERATION COMPUTATIONS MAIN 94 *IS EQUAL TO ECG-**F10.4,7M*BL OR *F10.4) MAIN 95 C MAIN 97 WRITE(6.23) MAIN 97 WRITE(6.47) MAIN 97 23 FORMAT(" STATION NO.",3X,"DEAU RISE",8X,"EST",8X,"NO", MAIN 99 47 FORMAT(" STATION NO.",3X,"DEAU RISE",8X,"EST",8X,"NO", MAIN 100 * 10X,"BEAM") MAIN 100 * 10X,"BEAM") MAIN 101 WRITE(6.55) ((I;9ETA*EST(I)*NU(I)*BM(I))*I=1*NUM) MAIN 102 55 FORMAT(6X,12*5X*F10.4+4X*F10.4+4X*F10.4+3X*F10.4) MAIN 103 WRITE(6.56) (X(I)*I=1*6) MAIN 105 56 FORMAT(" X VALUES",4X*6(F10.4+2X)) MAIN 106 C * * * * * * CHANGE INPUT FROM DEGREES TO RADIANS MAIN 107 X(3) = X(3)*RPD MAIN 107 C MAVE = STATT*RISE MAIN 107 C * * * * * * WRITE OUT COMPUTED ARRAYS MAIN 112 C * * * * * * WRITE OUT COMPUTED ARRAYS MAIN 113 WRITE(6.57), WIT*,C,PHALF,PI,GRAVYT MAIN 114 WRITE(6.591 (E(1)*I=1*NUM) MAIN 118 WRITE(6.591 (E(1)*I=1*NUM) MAIN 118 WRITE(6.591 (E(1)*I=1*NUM) MAIN 118			
<ul> <li>*IS EQUAL TO ECG-*,F10.4,7H*BL OR ,F10.4)</li> <li>*IS EQUAL TO ECG-*,F10.4,7H*BL OR ,F10.4)</li> <li>*MAIN 95</li> <li>*MAIN 95</li> <li>*WRITE(6,23)</li> <li>*MAIN 98</li> <li>*AIN 99</li> <li>*AIN 99</li> <li>*AIN 99</li> <li>*AIN 98</li> <li>*AIN 98</li> <li>*AIN 99</li> <li>*AIN 100</li> <li>*AIN 101</li> <li>*AIN 102</li> <li>*AIN 104</li> <li>*AIN 105</li> <li>*AIN 104</li> <li>*AIN 105</li> <li>*AIN 105</li> <li>*AIN 106</li> <li>*AIN 106</li> <li>*AIN 106</li> <li>*AIN 107</li> <li>*AIN 107</li> <li>*AIN 108</li> <li>*AIN 107</li> <li>*AIN 108</li> <li>*AIN 114</li> <li>*AIN 115</li> <li>*AIN 115</li> <li>*AIN 116</li> <li>*AIN 116</li> <li>*AIN 116</li> <li>*AIN 117</li> <li>*AIN 118</li> </ul>			
C MAIN 96 WRITE (6+23) WRITE (6+7) 23 FORMAT (1H ,//) 47 FORMAT (" STATION NO.", 3X, "DEAU RISE", 8X, "EST", 8X, "NO", * 10X, "BEAM") WRITE (6+55) ((1+BETA+EST(1), NU(1)+BM(1)), I=1+NUM) WRITE (6+55) ((1+BETA+EST(1), NU(1)+BM(1)), I=1+NUM) MAIN 102 55 FORMAT (6X, 12+5X, F10.4+4X, F10.4+4X, F10.4+3X, F10.4) WRITE (6+55) (X(1), I=1,6) 56 FORMAT (" X VALUES", 4X, 6(F10,4+2X)) C * * * * * * * CHANGE INPUT FROM DEGREES TO RADIANS X(6) = X(6) * RPD X(3) = X(3) * RPD MAIN 109 C * * * * * * * WRITE OUT COMPUTED ARRAYS MAIN 101 C * * * * * * * WRITE OUT COMPUTED ARRAYS MAIN 112 C * * * * * * * WRITE OUT COMPUTED ARRAYS WRITE (6+57), WIT, *, C, PHALF, PI, GRAVTY WRITE (6+57) ((1), I=1, NUM) WRITE (6+59) ((1), I=1, NUM) WRITE (6+59) ((1), I=1, NUM) WRITE (6+59) ((1), I=1, NUM) WRITE (6+64) (MMAX(1), I=1, NUM) MAIN 118			
WRITE (6+23)       MAIN 97         WRITE (6+23)       MAIN 98         WRITE (6+47)       MAIN 99         23 FORMAT (" STATION NO.", 3X, "DEAU RISE", 8X, "EST", 8X, "NO",       MAIN 99         47 FORMAT (" STATION NO.", 3X, "DEAU RISE", 8X, "EST", 8X, "NO",       MAIN 100         * 10x, "BEAM")       MAIN 100         wRITE (6555)       (I)BETA, EST (I), NU(I)+BM(I)), I=1, NUM)       MAIN 100         WRITE (6555)       (I)BETA, EST (I), NU(I)+BM(I)), I=1, NUM)       MAIN 102         S5 FORMAT (6x, 12, 5x, F10.4+4x, F10.4+4x, F10.4+3x, F10.4)       MAIN 103         WRITE (6, 56)       (X (I), I=1, 6)       MAIN 103         WRITE (6, 56)       (X (I), I=1, 6)       MAIN 105         56 FORMAT (" X VALUES", 4x, 6 (F10.4, 2X))       MAIN 105         C * * * * * CHANGE INPUT FROM DEGREES TO RADIANS       MAIN 106         X(3) = X (3) * RPD       MAIN 107         X(3) = X (3) * RPD       MAIN 107         X(4) = X (3) * RPD       MAIN 109         C * * * * * WRITE OUT COMPUTED ARRAYS       MAIN 112         NWAVE = 0       MAIN 112         C * * * * * WRITE OUT COMPUTED ARRAYS       MAIN 113         WRITE (6, 57), M, IT, K, C, PHALF, PT, GRAVTY       MAIN 113         WRITE (6, 591) (KII), I=1, NUM)       MAIN 116         WRITE (6, 659)			
WRITE (6,47)       MAIN 98         23 FORMAT(1H, //)       MAIN 99         47 FORMAT(" STATION NO.", 3X, "DEAU RISE", 8X, "EST", 8X, "NO",       MAIN 99         47 FORMAT(" STATION NO.", 3X, "DEAU RISE", 8X, "EST", 8X, "NO",       MAIN 100         * 10X, "BEAM")       MAIN 101         WRITE (6,55) ((I, 8ETA, EST(I), NU(I), 8M(I)), I=1, NUM)       MAIN 102         55 FORMAT(6X, 12, 5X, FI0.4+4X, FI0.4+4X, FI0.4+3X, FI0.4)       MAIN 103         WRITE (6,56) (X(I), I=1,6)       MAIN 105         56 FORMAT(" X VALUES", 4X, 6(F10.4+2X))       MAIN 105         C * * * * * CHANGE INPUT FROM DEGREES TO RADIANS       MAIN 106         X(3) = X(3)*RPD       MAIN 107         X(3) = X(6)*RPD       MAIN 108         X(6) = X (6)*RPD       MAIN 110         WAVE = STAT+RISE       MAIN 111         NWAVE = 0       MAIN 112         C * * * * * * WRITE OUT COMPUTED ARRAYS       MAIN 112         WRITE (6,57) M, IT, *K, C, PMALF, PI, GRAVTY       MAIN 113         WRITE (6,59) (E(I), I=1, NUM)       MAIN 116         WRITE (6,56) (E(I), I=1, NUM)       MAIN 118			
23 FORMAT(1H ,//) 47 FORMAT(" STATION NO.", 3X, "DEAU RISE", 8X, "EST", 8X, "NO", * 10X, "BEAM"] WRITE(6,55) ((I, BETA, EST(I), NU(I), BM(I)), I=1, NUM) MAIN 101 WRITE(6,55) ((I, BETA, EST(I), NU(I), BM(I)), I=1, NUM) MAIN 102 55 FORMAT(6X, 12, 5X, F10, 4, 4X, F10, 4, 4X, F10, 4, 3X, F10, 4) WRITE(6,23) WRITE(6,56) (X(I), I=1,6) 56 FORMAT(" X VALUES", 4X, 6(F10, 4, 2X)) C * * * * * * CHANGE INPUT FROM DEGREES TO RADIANS X(6) = X(3) * RPD X(3) = X(3) * RPD MAIN 108 X(6) = X(6) * RPD C MAVE = STAT+RISE MAIN 101 WAVE = 0 C * * * * * * * WRITE OUT COMPUTED ARRAYS MAIN 112 C * * * * * * * WRITE OUT COMPUTED ARRAYS MAIN 113 WRITE(6,57) M, IT, *K, C, PHALF, PI, GRAVTY IF (NPRINT, LT, 4) GO TO 62 WAIN 115 WRITE (6,59) (E(I), I=1, NUM) WRITE (6,59) (MI, I, I=1, NUM) WRITE (6,59) (MAX(I), I=1, NUM) WAIN 118			
47 FORMAT(" STATION NO.", 3X, "DEAU RISE", 8X, "EST", 8X, "NO", * 10X, "BEAM") WRITE(6,55) ((I;BETA;EST(I), NU(I), BM(I)), I=1, NUM) MAIN 102 55 FORMAT(6X, 12, 5X, F10.4, 4X, F10.4, 4X, F10.4, 3X, F10.4) WRITE(6,52) ((I;BETA;EST(I), 4, 4X, F10.4, 3X, F10.4) WRITE(6,52) (X(I), I=1,6) 56 FORMAT(" X VALUES", 4X, 6(F10.4, 2X)) C * * * * * * CHANGE INPUT FROM DEGREES TO RADIANS X(3) = X(3) * RPD X(3) = X(3) * RPD MAIN 100 C MAVE = STA∵T+RISE MAIN 101 WAVE = 0 C * * * * * * WRITE OUT COMPUTED ARRAYS MAIN 112 C * * * * * * WRITE OUT COMPUTED ARRAYS MAIN 113 WRITE(6,57) M, IT, *, C, PHALF, PI, GRAVTY MAIN 114 WRITE(6,591 (E(I), I=1, NUM) WRITE (6,591 (MI1, I=1, NUM) WRITE (6,591 (MAX(I), I=1, NUM) WAIN 118 WRITE (6,564) (MMAX(I), I=1, NUM)			
* 10x>"BEAM") WRITE(6*55) ((I*BETA*EST(I)*NU(I)*BM(I))*I=1*NUM) MAIN 101 WRITE(6*55) ((I*BETA*EST(I)*NU(I)*BM(I))*I=1*NUM) MAIN 102 S5 FORMAT(6x;12*5X*F10*4*X*F10*4*AX*F10*4) MAIN 103 WRITE(6*23) WAIN 104 WRITE(6*55) (X(I)*I=1*6) S6 FORMAT("X VALUES"*AX*6(F10*4*2X)) C * * * * * * * CHANGE INPUT FROM DEGREES TO RADIANS X(3) = X(3)*RPD X(3) = X(3)*RPD MAIN 106 X(6) = X(6)*RPD C WAVE = STATT+RISE MAIN 101 WAVE = 0 C * * * * * * * MITE OUT COMPUTED ARRAYS WRITE(6*57)M*IT*K*C*PHALF*FI*GRAVTY MAIN 112 WRITE(6*57)(I)*IT*K*C*PHALF*FI*GRAVTY MAIN 115 WRITE(6*59)(I)*I=1*NUM) WRITE(6*59)(I)*I=1*NUM) WAIN 118 WAIN 118	47 FORMAT(" STATION NO.".3X."DEAU RISE".8X."EST".8X."NO".		
WRITE (6,55) ((I,BETA,EST (I),NU (I),BM (I)),I=1,NUM)       MAIN 102         55 FORMAT(6X,I2,5X,FI0,4,4X,FI0,4+,4X,FI0,4+,3X,FI0,4)       MAIN 103         WRITE (6,56) (X (I),I=1,6)       MAIN 106         WRITE (6,56) (X (I),I=1,6)       MAIN 105         56 FORMAT(" X VALUES",4X,6(FI0,4,2X))       MAIN 106         C * * * * * * CHANGE INPUT FROM DEGREES TO RADIANS       MAIN 106         X(3) = X(3)*RPD       MAIN 108         X(6) = X(6)*RPD       MAIN 108         C * * * * * * WRITE OUT COMPUTED ARRAYS       MAIN 111         NHAVE = 0       MAIN 112         C * * * * * * WRITE OUT COMPUTED ARRAYS       MAIN 112         WRITE (6,57) M,IT,*K,C,PMALF,PI,GRAVTY       MAIN 115         WRITE (6,57) M,IT,*K,C,PMALF,PI,GRAVTY       MAIN 115         WRITE (6,59) (E(I),I=1,NUM)       MAIN 116         WRITE (6,59) (MI,I),I=1,NUM)       MAIN 118	* 10X, "BEAM")		
WRITE (6+23)       MAIN 104         WRITE (6+23)       MAIN 105         S6 FORMAT(" X VALUES"+4X+6(F10_4+2X))       MAIN 105         C * * * * * * CHANGE INPUT FROM DEGREES TO RADIANS       MAIN 106         X(3) = X(3)*RPD       MAIN 107         X(3) = X(3)*RPD       MAIN 108         X(6) = X(6)*RPD       MAIN 109         C       MAIN 100         WAVE = STAPT+RISE       MAIN 110         NWAVE = 0       MAIN 112         C * * * * * * * NITE OUT COMPUTED ARRAYS       MAIN 112         WRITE (6+57)M, IT+K*,C*PHALF+PI, GRAVTY       MAIN 113         WRITE (6+58) (E(I)+I=1,NUM)       MAIN 115         WRITE (6+59) (MI)+I=1+NUM)       MAIN 116	WRITE(6,55) ((I,BETA,EST(I),NU(I),BM(I)),I=1,NUM)		
wRITE (6,56) (X(I),I=1,6)       MAIN 105         56 FORMAT(" X VALUES",4X;6(F10,4,2X))       MAIN 106         C * * * * * * CHANGE INPUT FROM DEGREES TO RADIANS       MAIN 106         X(3) = X(3)*RPD       MAIN 107         X(3) = X(6)*RPD       MAIN 108         X(6) = X(6)*RPD       MAIN 109         C       WAVE = STANT+RISE         NWAVE = 0       MAIN 110         C * * * * * * WRITE OUT COMPUTED ARRAYS       MAIN 112         WRITE (6,57) M,IT,*K,*C,*PHALF,*PI,*GRAVTY       MAIN 113         WRITE (6,57) (E(I),II=1,NUM)       MAIN 116         WRITE (6,59) (E(I),II=1,NUM)       MAIN 117         WRITE (6,64) (MMAX(I),I=1,NUM)       MAIN 118	55 FORMAT(6X,12,5X,F10,4,4X,F10,4,4X,F10,4,3X,F10,4)	MAIN	103
56       FORMAT(" X VALUES", 4X,6(F10,4,2X))       MAIN 106         C * * * * * * CHANGE INPUT FROM DEGREES TO RADIANS       MAIN 107         X(3) = X(3)*RPD       MAIN 108         X(6) = X(6)*RPD       MAIN 109         C       WAVE = STAPT+RISE         WAVE = 0       MAIN 111         NWAVE = 0       MAIN 112         C * * * * * * WRITE OUT COMPUTED ARRAYS       MAIN 112         C * * * * * * WRITE (0,57) M, IT, *K, *C, PMALF, PI, GRAVTY       MAIN 113         WRITE (6,57) M, IT, *K, *C, PMALF, PI, GRAVTY       MAIN 115         WRITE (6,57) M, IT, *K, *C, PMALF, PI, GRAVTY       MAIN 115         WRITE (6,57) M, IT, *K, *C, PMALF, PI, MAIN       MAIN 116         WRITE (6,57) M, IT, *K, *C, PMALF, PI, MAIN       MAIN 116         WRITE (6,57) M, IT, *K, *C, PMALF, *PI, MAIN       MAIN 116         WRITE (6,57) M, IT, *K, *C, PMALF, *PI, MAIN       MAIN 116         WRITE (6,59) (MII, I=1, NUM)       MAIN 116         WRITE (6,59) (M, II, I=1, NUM)       MAIN 117         WRITE (6,64) (MMAX(I), I=1, NUM)       MAIN 118		MAIN	104
C * * * * * * * CHANGE INPUT FROM DEGREES TO RADIANS X(3) = X(3)*RPD X(6) = X(6)*RPD MAIN 109 C WAVE = STAUT+RISE WAIN 110 WAVE = 0 C * * * * * * WRITE OUT COMPUTED ARRAYS WRITE (6,57) M, IT, K, C, PHALF, PI, GRAVTY MAIN 112 WRITE (6,57) (E(I), I=1, NUM) WRITE (6,59) (I), I=1, NUM) WRITE (6,664) (MMAX(I), I=1, NUM) WAIN 118 MAIN 118		MAIN	105
X(3) = X(3)*RPD X(6) = X(6)*RPD C WAVE = START+RISE WAIN 100 WAVE = 0 C ************************************			
X(6) = X(6)*RPD MAIN 109 C WAVE = STAPT+RISE NWAVE = 0 C * * * * * * WRITE OUT COMPUTED ARRAYS MAIN 112 C * * * * * * * WRITE OUT COMPUTED ARRAYS MAIN 112 C * * * * * * * WRITE (0.57) WIT+K*C.PMALF,PI,GRAVTY IF (NPRINT,LT*C,PMALF,PI,GRAVTY MAIN 115 WRITE (6.57) (E(1)*I=1,NUM) WRITE (6.59) (N(1)*I=1,NUM) WRITE (6.664) (MMAX(1)*I=1*NUM) MAIN 118			
C MAIN 110 WAVE = STAPT+RISE MAIN 111 NWAVE = 0 MAIN 111 C * * * * * * WRITE OUT COMPUTED ARRAYS MAIN 113 WRITE (6,57)M,IT,K,C,PHALF,PI,GRAVTY MAIN 113 WRITE (6,57)M,IT,K,C,PHALF,PI,GRAVTY MAIN 114 IF (NPRINT_LT_4) GO TO 62 MAIN 115 WRITE (6,57) (E(I),I=1,NUM) MAIN 116 WRITE (6,659) (N(I),I=1,NUM) MAIN 117 WRITE (6,64) (MMAX(I),I=1+NUM) MAIN 118			
wave = STAPT+RISE       MAIN 111         NWAVE = 0       MAIN 112         C * * * * * * wRITE OUT COMPUTED ARRAYS       MAIN 112         wRITE (6,57) M, IT, K, C, PHALF, PI, GRAVTY       MAIN 113         wRITE (6,57) M, IT, K, C, PHALF, PI, GRAVTY       MAIN 114         IF (NPRINT, LT, 4)       GO TO 62       MAIN 115         wRITE (6,58) (E(I), I=1, NUM)       MAIN 116         wRITE (6,59) (N(I), I=1, NUM)       MAIN 117         wRITE (6,64) (MMAX(I), I=1, NUM)       MAIN 118			
NWAVE = 0       MAIN 112         C * * * * * * wRITE OUT COMPUTED ARRAYS       MAIN 113         WRITE (6.57) M.IT.*K.C.PMALF.PI,GRAVTY       MAIN 114         IF (NPRINT.LT.*)       GO TO 62         WRITE (6.559) (E(1).1=1.NUM)       MAIN 116         WRITE (6.564) (MMAX(1).1=1.NUM)       MAIN 118			
C * * * * * * * WRITE OUT COMPUTED ARRAYS WRITE (6,57)M,IT,K,C,PMALF,PI,GRAVTY IF (NPRINT_LT_4) GO TO 62 WRITE (6,58) (E(I),I=1,NUM) WRITE (6,59) (N(I),I=1,NUM) WRITE (6,64) (MMAX(I),I=1,NUM) MAIN 118			
WRITE (6,57)M,IT,K,C,PHALF,PI,GRAVTY       MAIN 114         IF (NPRINT_LT_4)       GO TO 62       MAIN 115         WRITE (6,58) (E(I),I=1,NUM)       MAIN 116         WRITE (6,59) (N(I),I=1,NUM)       MAIN 117         WRITE (6,64) (MMAX(I),I=1,NUM)       MAIN 118			
IF(NPRINT_LT_64)       G0 TO 62       MAIN 115         WRITE (6,558)       (E(I),I=1,NUM)       MAIN 116         WRITE (6,59)       (N(I),I=1,NUM)       MAIN 117         WRITE (6,64)       (MMAX(I),I=1,NUM)       MAIN 118			
WRITE (6,58) (E(I),I=1,NUM)       MAIN 116         WRITE (6,59) (N(I),I=1,NUM)       MAIN 117         WRITE (6,64) (MMAX(I),I=1,NUM)       MAIN 118			
WRITE (6,59) (N(I),I=1,NUM) MAIN 117 WRITE (6,64) (MMAX(I),I=1,NUM) MAIN 118			
WRITE (6+64) (MMAX(I)+I=1+NUM) MAIN 118	WRITE $(6,59)$ (N(I),I=1,NUM)		
WRITE (6,66) (TEST(I),I=1,NUM) MAIN 119	WRITE (6+64) (MMAX(I)+I=1+NUM)		
	WRITE (6,66) (TEST(I),I=1,NUM)	MAIN	119

```
62 CONTINUE
                                                                             MATH 120
      WRITE (6.28) (KTT(I) .DIFE(I) .I=1.II)
                                                                             MATN 121
   28 FORMAT (* KTT+DIFF *+110+2X-F10.4)
                                                                             MATH 122
   57 FORMAT(4H M= +F10,4,4H I= +F10,4,4H K= +F10,4,4H C= +F10,4,11H PI*MAIN 123
     #RH0/2= ,F10_4,5H PI= ,F10_4,10H GRAVITY= ,F10_4)
                                                                             MAIN 124
   58 FORMAT (" F(I)"+10F10+4)
                                                                             MAIN 125
   59 FORMAT (" N(I)"+10F10-4)
                                                                             MATN 126
   64 FORMAT (" MMAX(I)"+10F10-4)
                                                                             MAIN 127
   66 FORMAT (" TEST(1)", 10F10,4)
                                                                             MAIN 128
                                                                             MAIN 129
      IR = 1
      IPRINT = NPRINT
                                                                             MATN 130
      WPITE(4.91)
                                                                             MAIN 131
C * * * * * * * * WRITE HEADINGS AND CONDITIONS AT TIME = 0.
                                                                             MAIN 132
   91 FORMAT(1H1+2X+"TIMF"+9X+"XDOT"+9X+"ZDOT"+9X+"THFTA DUT"+6X+
                                                                             MAIN 133
     4 1HX+9X+1HZ+9X+5HTHETA+9X+2HNL+9X+2HFL+
                                                                             MATN 134
       4X.8HBUW ACCL.4X.7HCG ACCL.//)
                                                                             MATN 135
      WRITE (4:92) TIME . (X(I) .I=1.6) .NL .FL .BWACL .CGACL
                                                                             MATN 136
      WRITE(9) TIME . (X(I) . I=4.6) . BWACL . CGACL
                                                                             MAIN 137
      KOUNT = KOUNT+1
                                                                             MATN 138
      FX(1 \circ IB) = X(5)
                                                                             MAIN 139
      FX(2 \cdot IB) = X(6)
                                                                             MAIN 140
      IKUTH=(XE-XA)/HMAX+.05
                                                                             MATN 141
      IKUTM = (TME-XA)/HMAX + (XE-TME)/HMX + .05
                                                                             MAIN 142
      FIRST=0.0
                                                                             MAIN 143
      NEOS=6
                                                                             MAIN 144
      IKUTS=0
                                                                             MATN 145
                                                                             MAIN 146
С
Ċ
        START OF INTEGRATION LOUP
                                                                             MAIN 147
č
                                                                             MAIN 148
  A51 CONTINUE
                                                                             MAIN 149
      NPRINT = IPRINT
                                                                             MAIN 150
    * * * * * * * CHECK PITCH .GT. .5236 RADIANS
                                                                             MAIN 151
С
      IF(X(6).GT.,5236)G0 TO 853
                                                                             MATN 152
    * * * * * * * PERFORM INTEGRATIONS
                                                                             MAIN 153
      IF (TIME.LT. TH. OR. TME.EQ. XE) GU TO 98
                                                                             MAIN 154
        IF (IPT.F0.1) GO TO 98
                                                                             MAIN 155
          HMIN = HMN
                                                                             MAIN 156
          HMAX = HMX
                                                                             MAIN 157
                                                                             MAIN 158
          FIRST = 0.0
   98 CONTINUE
                                                                             MAIN 159
                                                                             MAIN 160
      CALL KUTMED (NEQS.TIME.HMAX.X.EPSE.A.HMIN.FIRST)
                                                                             MAIN 161
      IKUTS=IKUTS+1
                                                                             MAIN 162
      IF(FIRST_E0_2)G0 T0 861
                                                                             MAIN 163
      IF (KOUNT.NE.1.AND.KOUNT.NE.41) GO TO 99
                                                                             MATN 164
      WRITE(4,91)
                                                                             MAIN 165
      KOUNT=1
C * * * * * * * * * WRITE OUT TIME INTERVAL RESULTS
                                                                             MAIN 166
   99 WRITE (4,92) TIME, (X(I), I=1,6), NL, FL, BWACL, CGACL
                                                                             MAIN 167
      WRITE (6,93) T1. T2. T3. T4. T5. T6. T7. T8. 8MM. 8F
                                                                             MAIN 168
                                                                             MAIN 169
      WRITE(9) TIME, (X(I), I=4,6), BWACL, CGACL
      IF (TIME+LT. TM+OR. TME.EQ. XE) GU TO 200
                                                                             MAIN 170
      IF(IPT.E0.1) GO TO 200
                                                                             MAIN 171
                                                                             MAIN 172
      CALL PLUTED (FX.XA.HMAX.LAMBDA.IB.NWAVE.IPT)
      IPT = 1
                                                                             MAIN 173
      IB = 0
                                                                             MAIN 174
                                                                             MAIN 175
      XA = TIME
                                                                             MAIN 176
      FIRST = 0.0
                                                                             MAIN 177
      HMIN = HMN
                                                                             MAIN 178
      HMAX = HMX
```

	200	CONTINUE	MAIN 179
		I8=I8+1	MAIN 180
		$F_X(1, IB) = X(5)$	MAIN 181
		FX(2,IB)=X(6)	MAIN 182
	93	FORMAT(" ",10E10.4)	MAIN 183
	92	FORMAT(1X+11(F10+4+2X))	MAIN 184
	190	CONTINUE	MAIN 185
		KOUNT=KOUNT+1	MAIN 186
		IF (NWAVE.GT.0)G0 TO 21	MAIN 187
		IF (TIME.GT. WAVE) NWAVE=KOUNT	MAIN 188
	21	CONTINUÉ	MAIN 189
		IF (TIME.LE.XE.AND.IKUTS.LT.IKUTM) GU TO 851	MAIN 190
		WRITE (2,852)	<b>MAIN 191</b>
	854	CONTINUE	MAIN 192
		FORMAT( " END OF KUTMER")	MAIN 193
		CONTINUE	MAIN 194
		CALL PLUTER (FX, XA, HMAX, LAMBDA, IB, NWAVE, IPT)	MAIN 195
С			MAIN 196
č		NEW DATA FOR NEXT RUN	MAIN 197
-		IF (END_NE_1) GU TO 8	MAIN 198
		GO TO 999	MAIN 199
С			MAIN 200
		WRITE (6,86?)	MAIN 201
		FORMAT (" ERROR CRITERION IN KUTMER CAN NOT BE MET")	MAIN 202
		WRITE (6,56) (X(I),I=1,6)	MAIN 203
		WRITE (6,86) TIME	MAIN 204
	86	FORMAT (" TIME =" +F10.4)	MAIN 205
		IF (END_NE_1) GO TO B	MAIN 206
		GO TO 853	MAIN 207
	999	CONTINUE	MAIN 208
		END FILÉ 9	MAIN 209
		END	MAIN 210
		SUBROUTINE PLOT2(F,FMIN,FMAX,NVAR,NFUN,NI,N,X0,DELX)	PLOT2 2
С			PLOT2 3
С	PL0.	FIRST N POINTS OF UP TO 26 FUNCTIONS F(X)	PLOT2 4
С	– F ()	I,J) CONTAINS THE VALUE FOR THE JTH POINT OF THE ITH FUNCTION	PLOT2 5
С	EM)	IN(I) AND FMAX(I) CONTAIN THE MIN AND MAX ORDINATE VALUES FOR	PLOT2 6
С		THE ITH FUNCTION.	PLOT2 7
С		NVAR(I) AN ARRAY OF TITLES FOR THE VARIOUS FUNCTIONS	PLOT2 8
С		TO BE PLOTTED AGAINST THE ABSCISSA	PLOT2 9
С		NFUN NUMBER OF FUNCTIONS TO BE PLOTTED - DIMENSION OF	PLOT2 10
С		NVAR, FMIN, FMAX	PLOT2 11
С		NI USED ONLY IN F (N1,1) AS PASSED DIMENSION	PLOT2 12
С		N NUMBER OF POINTS IN A SINGLE PLOT FRAME	PLOT2 13
С		X0 FIRST ABSCISSA VALUE	PLOT2 14
С		DELX ABSCISSA INCREMENT	PLOT2 15
С			PLOT2 16
		DIMENSION STEP (26), F(N1, N), FMIN (NFUN), FMAX (NFUN), VLAST (26),	PLOT2 17
		VFIPST(26), HEAD(6), STEP(26)	PLOT2 18
		INTEGER CH (26), NVAR ( NFUN), DOT, ASTER, PLUS, BLANK	PLOT2 19
		INTEGER C	PLOT2 20
~		INTEGER A(101)	PL0T2 21 PL0T2 22
С		DATA DI ANY DOT ACTED DI UCALLE AND AND AND AND	
		DATA BLANK, DOT, ASTER, PLUS/IH , 1H., 1H*, 1H*, 1H+/	PLOT2 23
	,	DATA CH(1), CH(2), CH(3), CH(4), CH(5), CH(6), CH(7), CH(8), CH(9), CH(10)	PL012 24 PL012 25
		2 / 1HA , 1HB , 1HC , 1HD , 1HE , 1HF , 1HG , 1HH , 1HJ /	PLOT2 25
		DATA CH(11), CH(12), CH(13), CH(14), CH(15), CH(16), CH(17), CH(18)	
	"	2. / 1HK , 1HL , 1HM , 1HN , 1HO , 1HP , 1HQ , 1HR/	PLOT2 27
		DATA CH(19),CH(20),CH(21),CH(22),CH(23),CH(24),CH(25),CH(26)	PLOT2 28

2 / 1HS , 1HT , 1HU , 1HV , 1HW , 1HX , 1HY , 1HZ /	PLOT2 29
C	PLOT2 30
IF (NFUN.LE.0.0R.N.LE.0) RETURN	PLOT2 31
C PRINT HEADINGS	PLOT2 32
WRITE (6,46)	PLOT2 33
46 FORMAT (///)	PLOT2 34
DO 40 I=1.5VEUN	PLOT2 35
30 TENM=AUS(FMAX(I)-FMIN(I))	
EXP=1.	PLOT2 36
	PLOT2 37
IF (TENM.EQ.0.) GO TO 2	PLOT2 38
C BRING TENM TO A VALUE BETWEEN 1 AND 10	PLOT2 39
IF (TENM.LT.1.) GO TO 1	PLOT2 40
3 IF(TENM.LT.10.) GO TO 2	PLOT2 41
EXP=EXP+10.	PLOT2 42
TENM=TENM+.1	PLOT2 43
GO TO 3	PLOT2 44
1 EXP=EXP+.1	PLOT2 45
TENM=TÊNM+10.	PLOT2 46
IF (TENM.GT.1.) GO TO 2	PLOT2 47
GO TO 1	PLOT2 48
C SET UP VALUE RETWEEN GRID LINES, RSTEP.	PLOTZ 49
2 PSTEP=5.	PLOT2 50
IF (TENM.GE.5.)PSTEP=10.	PLOT2 51
IF (TENM.LT.2.) PSTEP=2.	PLOT2 52
5 RSTEP(I)=PSTEP*EXP*.1	
C CUMPUTE VALUE OF STARTING LINE, VFIRST.	PLOT2 53
	PLOT2 54
FIRST=FMIN(I)/RSTEP(I)	PLOT2 55
IF (FMIN(I),LT,0,)FIRST=FIRST-1,	PLOT2 56
FIRST=AINT (FIRST)	PLOT2 57
VFIRST(I)=FIRST*RSTEP(I)	PLOT2 58
C CHECK END LINE VALUE, VLAST.	PLOT2 59
VLAST(I)=VFIRST(I)+10.*RSTEP(I)	PLOT2 60
IF(VLAST(I).GT.FMAX(I))G0 TO 4	PLOT2 61
C IF GRAPH IS TOO SMALL TAKE NEXT LARGER STEP.	PLOT2 62
AA=PSTÉP	PLOT2 63
IF (AA_LT_5,)PSTEP=5.	PLOT2 64
IF (AA.EQ.5.)PSTEP=10.	PLOT2 65
IF (AA.LT.10.) GO TO 5	PLOT2 66
PSTEP=2.	PLOT2 67
EXP=10.*EXP	PLOT2 68
GO TO 5	PLOT2 69
C CUMPUTE VALUE BETWEEN POINTS,STEP.	PLOT2 70
4 STEP (I) = RSTEP (I) * 1	PLOT2 71
RK=0.	PLOT2 72
DO 6 KK=1,6	PLOT2 73
HEAD(KK)=VFIRST(I)+2,*RK*RSTEP(I)	PLOT2 74
6 RK=RK+1.	PLOT2 75
40 WRITE (6,45) CH(I), NVAR(I), (HEAD(KK),KK=1,6)	PLOT2 76
45 FORMAT(1X,A1,3H = ,A10,5X,1PE12.4,5(8X,1PE12.4))	PLOT2 77
$D_0 = 50 J = 1 + 101$	PLOT2 78
A (J) = BLANK	PLOT2 79
IF (MOD (J+10) .EQ.1) A (J)=DOT	PLOT2 80
50 CONTINUE	PLOT2 81
WRITE (6,55) A,A	PLOT2 82
55 FORMAT (25x,101A1/15x,4HTIME,6x,101A1)	PLOT2 83
C PLUT EACH PUINT	
D0 100 J=1.N	PLOT2 84
	PLOT2 85
B=X0+FLUAT(J-1)*DELX	PLOT2 86
D0 70 K=1+101	PLOT2 87

			_
		A(K)=BLANK	PLOT2 88
		IF (MOD( $K_{10}$ ) $\in$ Qol) A(K)=DUT	PL0T2 89
		IF (MOD (J.5 ).EQ.1) A(K)=DUT	PLOT2 90
	70	CONTINUE	PLOT2 91
		DO 80 I=1,NFUN	PL0T2 92
		LOC=((F(I,J)-VFIRST(I))/STEP(I)+1.5)	PLOT2 93
		C=A(LOC)	PLOT2 94
		A(LOC)=CH(I)	PL0T2 95
		IF(C.NE.BLANK.AND.C.NE.DOT) A(LOC)=ASTER	PL0T2 96
	80	CONTINUE	PL0T2 97
	-	IF (MOD (J,10) .EQ.1) GO TO 95	PL0T2 98
		WRITE (6,85) A	PLOT2 99
	85	FORMAT (25X,101A1)	PLOT2100
	00	GO TO 100	PL072101
	0E		
		WRITE (6,15) B,A	PLOT2102
		FORMAT (12X, 1PE12, 4, 1X, 101A1)	PL0T2103
	100	CONTINUE	PLOT2104
		RETURN	PL0T2105
		END	PLOT2106
		SUBROUTINE KUTMER(ND,T,H,YO,EPSE,A,HCX,FIRST)	KUTMER 2
		DIMENSION Y0(6), Y1(6), Y2(6), F0(6), F1(6), F2(6), EPSE(6), A(6)	KUTMER 3
		COMMON/UUT/NPRINT,NPLOT,END	KUTMER 4
		COMMON /ACCEL / XACCL, BWACL, CGACL, BL	KUTHER 5
		DATA NAM1.NAM2 /2HY1,2HY2 /	KUTMER 6
~		DATA NAPIINAPE /ENTINETIE /	
c			KUTMER 7
С		ND = NUMBER OF EQUATIONS, NO. OF COMPONENTS OF YO	KUTMER 8
С		T = INDEPENDENT VARIABLE	KUTMER 9
С		H ⊐ INCREMENT FOR WHICH SULUTION IS TO BE RETURNED + OR -	KUTMER10
С		YO = THE VECTOR OF DEPENDENT VARIABLES. ENTER WITH INITIAL	KUTMER11
С		VALUES AT T AND RETURN WITH VALUES AT T+H	KUTMER12
С		EPSE = RELATIVE ERROR CRITERION FOR COMPONENTS OF Y0 .GT ABS(A)	KUTMER13
č		A = ABSOLUTE ERROR SRITERION FUR COMPONENTS OF YO .LT. ABS(A)	
č	NU	TE EPSE AND A MUST BE SPECIFIED FOR EACH COMPONENT OF THE SYSTEM	
	140	HCX = THE SMALLEST STEP SIZE USED IN THE INTEGRATION	KUTMER16
с с		FIRST SHOULD BE O WHEN KUTHER IS ENTERED FOR THE FIRST TIME	KUTMER17
c			
	'	AFTER THAT FIRST IS 1 IF KUTMER IS ENTERED WITH THE SAME H OR	KUTMER18
С		IF IT IS ENTERED WITH A CHANGED H	KUTMER19
С		IF FIRST IS 2 THE ERROR SRITERIA CANNOT BE MEET AND THE STEP SIZE	
С	1	REDUCED TO H/128.	KUTMER21
С			KUTMER22
		IF (FIRST) 20,10,20	KUTMER23
с		FIRST ENTRY	KUTMER24
-	10	HC = H	KUTMER25
		IPLOC = 1	KUTMER26
		FIRST = 1	KUTMER27
с		OTHER ENTRY	KUTMER28
C			
	20		KUTMER29
		HCX = HC	KUTMER30
		IF (HC,NE,0,) GO TO 30	KUTMER31
		WRITE(6+800)	KUTMER32
	800	FORMAT(5X,45HKUTMER ENTERED WITH ZERO INTEGRATION INTERVAL )	KUTMER33
	-	$FIRST = 2_{\bullet}$	KUTMER34
		RETURN	KUTMER35
С			KUTMER36
	30	CALL DAUX (T,Y0,FO)	KUTMER37
	22	IF (NPRINT.EQ.5) WRITE (6,400) Y0, T,F0	KUTMER38
	460	FORMAT (6 (2x, F10, 4), 4HTINE, 2x, F10, 4)	KUTMER39
	-44	IF (NPRINT_EQ.5) WRITE (6.400) HC	KUTMER40
	20		KUTHER41
	22	DO 40 I=1,ND	NUTWER41

	40	$Y1(I) = Y0(I) + (HC/3_{o}) + F0(I)$	KUTMER42
	-	IF (NPRINT_EQ.5) WRITE (6,400) Y1,T	KUTMER43
с			KUTMER44
Ũ		CALL DAUX(T+HC/3.,Y1,F1)	KUTMER45
		IF (NPRINT.EQ.5) WRITE (6,400) F1+T	KUTHER46
		DO 50 I=1.ND	KUTHER47
	50	$Y1(I) = Y0(I) + (HC/6_{\bullet}) + F0(I) + (HC/6_{\bullet}) + F1(I)$	KUTHER48
	J.	IF (NPRINT_FQ.5) WRITE (6,400) Y1,T	KUTHER49
с		IT THE REAL OF THE TO T	KUTMER50
C		CALL DAUX(T+HC/3.,Y1,F1)	KUTMER51
		IF (NPRINT.EQ.5) WRITE (6,400)F1+T	KUTMER52
		DO 60 $I=1$ ND	
	6.0	$Y1(I) = Y0(I) + (HC/8_{\bullet}) + F0(I) + 375 + HC + F1(I)$	KUTMER53 KUTMER54
	0.0	IF (NPRINT_EQ.5) WRITE (6,400) Y1,T	KUTMER55
с			
C		CALL DAUX (T+HC/2.,Y1,F2)	KUTHER56
		IF (NPRINT.=FQ.5) WRITE (6,400)F2+T	KUTMER57
		DO 70 I=1.ND	KUTMER58
	70		KUTMER59
	19	$Y1(I) = Y0(I) + (HC/2_{\circ}) + F0(I) - 1_{\circ} + HC + F1(I) + 2_{\circ} + HC + F2(I)$	KUTMER60
с		IF (NPRINT. = Q.5) WRITE (6,400) Y1,T	KUTMER61
C		CALL DAMY (THE YE CT)	KUTMER62
		CALL DAUX(T+HC,Y1,F1)	KUTMER63
		IF (NPRINT.EQ.5) WRITE (6,400) F1+T	KUTMER64
	00	D0 80 I=1,ND Y2(I) = Y0(I)+HC/6,*F0(I)+(2,/3,)*HC*F2(I)+(HC/6,)*F1(I)	KUTHER65
	٥Ÿ	IF (NPRINT_EQ.5) WRITE (6,400) Y2,T	KUTMER66 KUTMER67
		INC = 0	
с			KUTMER68 KUTMER69
C	-	DO 110 1=1.ND	KUTMER70
		ZZZ = ABS(Y1(I)) - A(I)	KUTMER71
		IF (ZZZ) 85,87,87	KUTMER72
с		ABSOLUTE ERRUR	KUTMER73
C		ERROR = ABS(.2*(Y1(I)-Y2(I)))	KUTMER74
	00	IF (ERROR-A(I)) 100,100,90	KUTMER75
c		RELATIVÉ ERKOR	KUTMER76
č		ERROR = ABS(.22*Y2(I)/YI(I))	KUTMER77
	٠.	IF (ERROR-EPSE(I)) 100,100,90	KUTMER78
с		SINCE ERROR .GT. ERROR CRITERIA CHECK IF HC.GT.H.	
		IF YES THEN HALVE INTERVAL. OTHERWISE STOP.	KUTMER80
-		X = 128 * ABS(HC) - ABS(H)	KUTMER81
	-	IF(X) 91,95,95	KUTMER82
С		ERROR TOU LARGE	KUTMER83
-	91	WRITE(6992)I+T+ERROR+HC	KUTMER84
		FORMAT (/18H FUR EQUATION NO. 12,27H, THE RELATIVE ERROR AT T = ,	KUTMER85
		E15.8, 4H IS ,E15.8,13H STEP SIZE = ,E15.8)	KUTMER86
		FIRST = 2.	KUTMER87
		RETURN	KUTMER88
С		HALVE INTERVAL	KUTMER89
	95	$HC = HC/2_{\bullet}$	KUTMER90
		IPLOC = 2*IPLUC	KUTMER91
		LOC = 2*LOC	KUTMER92
		HCX = HC	KUTMER93
		WRITE(2,71)T,I,ERROR,HC	KUTMER94
	710	FORMAT(/8H TIME = ,F10.3,5X,26HHALVE INTERVAL. EQUATION ,I3,	KUTMER95
		13H HAS ERPOR = ,E16.8,6X,17H STEP SIZE NOW = ,E15.8)	KUTMER96
		WRITE(2,72) NAM2, (Y2(J), J=1,ND)	KUTMER97
		WRITE(2,720) NAM1, (Y1(J), J=1,ND)	KUTMER98
	720	FORMAT( 2X+A2 / 3(10E13+5/))	KUTMER99
		GO TO 30	KUTME100

```
TEST IF INTERVAL LENGTH CAN BE DOUBLED
                                                                             KUTME101
c
 ------
  100 IF (ERRUR*64,-EPSE(I)) 110,110,101
                                                                             KUTME102
  101 INC = 1
                                                                              KUTME 103
  110 CONTINUE
                                                                              KUTME104
                        UPDATE T AND SOLUTION
c
        - - - -
                                                                             KUTME105
  111 T = T + HC
                                                                             KUTME106
      DO 112 I=1.ND
                                                                             KUTME 107
  112 YO(1) = Y2(1)
                                                                             KUTME108
      ---- GET SOLUTION IN NEXT INTERVAL
                                                                             KUTME109
c
      LOC = LUC+1
                                                                              KUTME110
      IF (LUC-IPLOC) 120,210,210
                                                                              KUTME111
  120 IF (INC) 210,130,210
                                                                             KUTME112
  130 IF (LUC-(LUC/2)+2) 210,140,210
                                                                              KUTME113
  140 IF (IPLOC-1) 210,210,200
                                                                             KUTME114
      - - - - - - DOUBLE INTERVAL LENGTH
                                                                             KUTME115
c
  KIITME116
  200 HC = 2. +HC
      LOC = LUC /2
                                                                              KUTME117
      IPLOC = IPLOC/2
                                                                             KUTME118
  210 IF(IPLOC-LOC) 30.329.30
                                                                             KUTME119
  329 BWACL = F0(2)-XACCL+F0(3)
                                                                             KUTME 120
      CGACL = FO(2)
                                                                             KUTME121
      RETURN
                                                                             KUTME122
      END
                                                                             KUTME123
      END
                                                                             KUTME124
      SUBROUTINE DAUX (TIME . X . RHS)
                                                                             DAUX
                                                                                     2
с
                                                                             DAUX
                                                                                     3
                                                                                     4
č
          TIME
                    TIME AT WHICH SYSTEM IS TO BE EVALUATED.
                                                                             DAIIX
с
         х
                    STATE VECTOR
                                                                             DAUX
                                                                                     5
ċ
         RHS
                    THE RIGHT HAND SIDE UF THE EQUATION S = F A
                                                                             DALIX
                                                                                     6
С
                                                                             DAUX
                                                                                     7
      REAL KAR
                                                                             DAHX
                                                                                     8
      REAL IA+IT+M+K+MA+MASS+NCG+NL+N+MMAX
                                                                             DAUX
                                                                                     9
      INTEGER END.PTIME
                                                                             DAUX
                                                                                    10
      DIMENSIUN x (6) . RHS (6) . F (3.1) . A (3.3) . INDEX (3.3) .
                                                                             DALIX
                                                                                    11
                    R(120) .V(120) .D(120)
                                                                             DAUX
                                                                                    12
     .
С
                                                                             DAUX
                                                                                    13
      COMMON /SHIP/ MASS+CINT+QA+CE+CE2+CE3+DMU+EDMU+E2DMU+E3DMU+BF+BMM+DAUX
                                                                                    14
                    NL,FL, IA, E(120)
                                                                             DAUX
                                                                                    15
      COMMON /CUNST/ NCG.ECG.PI.DPR.RPD.GRAVTY.RHO.K.NUM.MA(120).CD.TA. DAUX
                                                                                    16
                B(120), BETA, HW(120), TZ, DRAG, W, XD, T, XP, M, IT,
                                                                             DAUX
                                                                                    17
     .
                DELTAS.TX.EST(120).C.RO.KAR.MMAX(1:0).TEST(120).
                                                                             DAUX
                                                                                    18
     .
                     N(120) PHALF
                                                                             DAUX
                                                                                    19
      COMMON /IN/ BM(120) .B1(120) .VELIN
                                                                             DAUX
                                                                                    20
      COMMON/UUT/NPRINT,NPLOT,END
                                                                             DAUX
                                                                                    21
      COMMON /SEAWAVE/ START.RISE.RAMP
                                                                             DAUX
                                                                                    22
      COMMON /WAVE/ R.PT(120), ZMA, ZWMA, EMAS, ZZWMA, ZWEMA, Z2WMA, EZMAZ,
                                                                             DAUX
                                                                                    23
                    ZwDOT (120)
                                                                             DAUX
                                                                                    24
     .
                                                                                    25
С
                                                                             DAUX
      RAMP = RMP(TIME.START.RISE)
                                                                             DAUX
                                                                                    26
      PIH = PI/2.
                                                                             DAUX
                                                                                    27
      CT = C*TIME
                                                                             DAUX
                                                                                    28
      Cx6 = CUS(x(6))
                                                                             DAUX
                                                                                    29
      SX6 = SIN(X(6))
                                                                             DAUX
                                                                                    30
C*******SET VALUES OF MA AND B
                                                                             DAUX
                                                                                    31
      DO 75
             I=1.NUM
                                                                             DAUX
                                                                                    32
      PT(I) = (X(4) + E(I) + CX6 + N(I) + SX6 + CT) + K
                                                                             DAUX
                                                                                    33
      R(I) = RU*COS(PT(I))*RAMP
                                                                             DAUX
                                                                                    34
    -
        * * * * * COMPUTE HW SUBMERGENCE OF A POINT AND R THE WAVE
                                                                                    35
С
                                                                             DAUX
C
              HW(I) IS IN THE FIXED COORDINATE SYSTEM
                                                                             DAUX
                                                                                    36
```

		HW(I) = X(5) - E(I) + SX6 + N(I) + CX6 - R(I)	DAUX	37
		IF(HW(I).GT.0) GO TO 65	DAUX	38
С		CRAFT IS NOT SUBMERGED	DAUX	39
		$MA(I) = 0_{e}$	DAUX	40
		B1(I)=0	DAUX	41
		B(I) = 0,	DAUX	42
		GO TO 75	DAUX	43
	45	V(I) = -RO*K*SIN(PT(I))*RAMP	DAUX	44
	05	D(I) = HW(I)/(CX6-V(I)*SX6)	DAUX	45
с		D(I) IS IN THE BODY AXIS SYSTEM AND IS THE SUBMERGENCE	DAUX	46
C		IF (D(I) .GE TEST(I)) GO TU 70		
~		CRAFT IS PARTLY SUBMERGED	DAUX	47
С			DAUX	48
		$B(I) = D(I) * (I_{\bullet}/TA) * PIH$	DAUX	49
		$Bl(I) = D(I)*(1_{\bullet}/TA)*PIH$	DAUX	50
		MA(I) = KAR*PHALF*B(I)*B(I)	DAUX	51
		GO TO 75	DAUX	52
C		CHINE IS IMMERSED	DAUX	53
С		BI ARRAY IS USED FOR THE INTEGRALS OVER THE PORTION	DAUX	54
С		OF THE HULL FOR WHICH THE CHINE IS NOT IMMERSED	DAUX	55
	10	MA(I) = MMAX(I)	DAUX	56
		B(I)=BM(I)	DAUX	57
		B1(I)=0.	DAUX	58
	75	CONTINUE	DAUX	59
		IF (NPRINT.LT.4) GO TO 85	DAUX	60
		WRITE(6,74)TIME	DAUX	61
	74	FORMAT(" TIME = ".F10.4)	DAUX	62
		WRITE(6,76) (X(I),I=1,6)	DAUX	63
		WRITE(6,77) (R(I),I=1,NUM)	DAUX	64
		WRITE (6,78) (HW(I), I=1, NUM)	DAUX	65
		WRITE (6,79) ( B(I),I=1,NUM)	DAUX	66
		WRITE (6,8^) (V(I),I=1,NUM)	DAUX	67
		WRITE (6,81) (D(I),I=1,NUM)	DAUX	68
		WRITE(6+82)(MA(I)+I=1+NUM)	DAUX	69
		FORMAT(" X(I) "+6(2X+E12+6))	DAUX	70
		FORMAT (" R(I)", 10F10.4)	DAUX	71
		FORMAT (" HW(I)",10F10,4)	DAUX	72
		FORMAT (" 8(I)",10F10,4)	DAUX	73
		FORMAT (" V(I)", 10F10.4)	DAUX	74
		FORMAT (" D(I)",10F10,4)	DAUX	75
		FORMAT(" MA(I) "+10F10.4)	DAUX	76
	85	CONTINUE	DAUX	77
С			DAUX	78
С	4 <b>4</b>	* * * * * * COMPUTES NL AND FL AND THE ASSOCIATED INTERGALS	DAUX	79
		CALL FUNCT(X)	DAUX	80
С			DAUX	81
		IF (NPRINT_LT_4) GO TO 17	DAUX	82
		WRITE(6,15) TX,FL,DRAG,TZ,W,NL,XD,T,XP	DAUX	83
		FORMAT(" ",10E12.6)	DAUX	84
		CONTINUE	DAUX	85
С	* *	* * * * COMPUTE THE F VECTOR	DAUX	86
		F(1,1) = TX + FL + SX6 - DRAG + CX6	DAUX	87
		F(1,1)=0.0	DAUX	88
		$F(2,1) = T_{Z}+FL*C_{X}6+D_{R}AG*S_{X}6+W$	DAUX	89
		F(3,1)=NL-DRAG*XD+T*XP	DAUX	90
		IF (NPRINT.LT.3)GO TO 18	DAUX	91
		WRITE(6,10)(F(I,1),I=1,3)	DAUX	92
		CONTINUE	DAUX	93
С	* *	* * * * COMPUTE THE A MATRIX	DAUX	94
		A(1,1) = M+MASS*SX6*SX6	DAUX	95

```
A(1,2) = MASS*SX6*CX6
                                                                                  DAIIX
                                                                                         96
      A(1+3) = -0A#SX6
                                                                                  DALIX
                                                                                         97
      A(1,2) = 0.
                                                                                  DAILY
                                                                                         98
      A(1 \cdot 3) = 0
                                                                                  XIIAG
                                                                                         99
      A(2.1)=A(1.2)
                                                                                  DALLX 100
      A(2*2) = M + MASS + CX6 + CX6
                                                                                  DAILX 101
                                                                                  DALLY 102
      A(2,3) = -0A * C \times 6
       A(3.1) = A(1.3)
                                                                                  DAILX 103
       A(3,2) = A(2,3)
                                                                                  DAILY 104
       A(3 \cdot 3) = IT + TA
                                                                                  DAUX 105
       IF (NPRINT .| T. 3) GO TO 25
                                                                                  DAUX 106
       WRITE (6 + 12) (A (I + 1) + I = 1 + 3)
                                                                                  DAILX 107
       WRITE (6+13) (A(1+2)+1=1+3)
                                                                                  DAUX 108
                                                                                  DALLX 109
      WRITE (6+14) (A(I+3)+I=1+3)
  * * * * * * * INVERT THE A MATRIX
                                                                                  DAUX 110
c
                                                                                  DAUX 111
   25 CALL MATINS (A+3+3+F+1+1+DETERM+ID+INDEX)
                                                                                  DAUX 112
       IF (ID. EQ. 2) WRITE (6.26)
   26 FORMAT (" MATRIX IS SINGULAR
                                                                                  DAUX 113
                                                   C******* ON RETURN WILL CONTAIN THE INVERSE MATRIX
                                                                                  DAUX 114
С
          ID=2 MATRIX IS SINGULAR
                                                                                  DALLX 115
Ċ
            =1 INVERSE WAS FOUND
                                                                                  DAILX 116
С
                                                                                  DALLY 117
č
    * * * * * * COMPUTE THE RIGHT HAND SIDE
                                                                                  DAILX 118
  RHS(1) = F(1,1)
                                                                                  DALIX 119
      RHS(2) = F(2.1)
                                                                                  DAUX 120
      RHS(3) = F(3 \cdot 1)
                                                                                  DAUX 121
      RHS(1) = 0.0
                                                                                  DAUX 122
                                                                                  DAUX 123
      RHS(4) = \dot{X}(1)
      RHS(5) = X(2)
                                                                                  DAUX 124
       RHS(6) = X(3)
                                                                                  DAUX 125
   10 FORMAT ("
                   F(I.1) ".3(2X.E12.4))
                                                                                  DAUX 126
   12 FORMAT("
                   A(I.1) ".3(2X.E12.4))
                                                                                  DAUX 127
   13 FORMAT ("
                   A(I.2) ".3(2X,E12.4))
                                                                                  DAHX 128
   14 FORMAT ("
                   A(I.3) ".3(2X.E12.4))
                                                                                  DAUX 129
   30 IF (NPRINT.LT.2) GO TO 40
                                                                                  DAUX 130
       WRITE(6,12) (A(I,1),I=1,3)
                                                                                  DAUX 131
       WRITE(6.13) (A(I.2).I=1.3)
                                                                                  DAUX 132
       WRITE(6,14) (A(I,3),I=1,3)
                                                                                  DAUX 133
       WRITE(6,35) (RHS(I), I=1,6)
                                                                                  DAUX 134
   35 FORMAT ("
                  RHS(I) ",6(2X,E12.6))
                                                                                  DAUX 135
   40 CONTINUE
                                                                                  DAUX 136
       RETURN
                                                                                  DAUX 137
       END
                                                                                  DAUX 138
       SUBROUTINE FUNCT(X)
                                                                                  FUNCT
                                                                                          2
       REAL KAR
                                                                                  FUNCT
                                                                                          3
       REAL IA, IAA, IPART, K, KPI, MA, MASS, NL, NCG, IT, M, MMAX, N
                                                                                  FUNCT
                                                                                          4
                                                                                          5
       INTEGER END
                                                                                  FUNCT
       DIMENSION IPART(120), C1(120), C2(120),
                                                                                  FUNCT
                                                                                          6
                  01(120),D2(120),D3(120),U4(120),D5(120),D6(120),
                                                                                  FUNCT
                                                                                          7
                  0PART (120) . 21 (120) . 22 (120) . 23 (120) . 24 (120) . 25 (120) .
                                                                                  FUNCT
                                                                                          я
                      26(120).77(120)
                                                                                  FUNCT
                                                                                          q
      .
                  .X(6) .VMAA(120)
                                                                                  FUNCT
                                                                                         10
С
                                                                                  FUNCT
                                                                                         11
       COMMON /SHIP/ MASS, CINT, QA, CE, CE2, CE3, DMU, EDMU, E2DMU, E3DMU, BF, BMM, FUNCT
                                                                                         12
                     NL+FL+IA+E(120)
                                                                                  FUNCT
                                                                                         13
      COMMON /CUNST/ NCG, ECG, PI, DPR, RPD, GRAVTY, RHO, K, NUM, MA (120), CD, TA,
                                                                                  FUNCT
                                                                                         14
                 B(120), BETA, HW(120), TZ, DRAG, W, XD, T, XP, M, IT,
                                                                                  FUNCT 15
      ٠
                 DELTAS, TX, EST(120), C, RO, KAR, MMAX(1 0), TEST(120),
                                                                                  FUNCT 16
      .
                       N(120), PHALF
                                                                                  FUNCT 17
      .
```

		CONNON (IN & BM (120) B1 (120) VEL IN	FUNCT	10
		COMMON /IN/ BM(120),B1(120),VLLIN COMMON/UUT/NPRINT,NPLOT,END	FUNCT I	-
		COMMON /WAVE/ R(120),PT(120),ZMA,ZWMA,EMAS,ZZWMA,ZWEMA,Z2WMA,E2P		
		2WDOT(120)	FUNCT 2	
	•	COMMON /INTER/ II,KTT(10),DIFF(10)	FUNCT 2	
		COMMON /SEAWAVE/ START, RISE, RAMP	FUNCT 2	
		COMMON /TEST/ VMA	FUNCT 2	
С #		* * * * * * * INITIALIZE INTEGRAL SUMS	FUNCT a	25
		MASS = 0.0	FUNCT a	
		$QA = 0_0 \tilde{Q}$	FUNCT a	
		IA = 0.0	FUNCT a	
		CE = 0.0	FUNCT 2	
		CE2 = 0.0	FUNCT C	
		DMU = 0.0	FUNCT C	
		EDMU=0.0 E2DMU=0.0	FUNCT C	
		$E3DMU = 0_0 \eta$	FUNCT 3	
		BF = 0.0	FUNCT 3	
		BMM = 0.0	FUNCT	
		ZMA = 0.0	FUNCT 3	
		ZWMA = 0.0	FUNCT 3	38
		$EMAS = \hat{V}_{\bullet}\hat{O}$	FUNCT 3	
		ZZWMA = 0.0	FUNCT 4	
		ZWEMA = 0.0	FUNCT 4	
			FUNCT 4	
		E2MAZ = 0.0 VPART = X(1)*SIN(X(6))*X(2)*CUS(X(6))	FUNCT 4	
		$S_{X6} = SIN(x(6))$	FUNCT 4	
		CX6 = CUS(X(6))	FUNCT 4	
		WO = K * C	FUNCT 4	
с :		* * * * * * SET UP THE FUNCTIONS FOR THE INTEGRALS (PAGE 4 OF		
		D0 90 I=1,NUM	FUNCT 4	49
		IPART(I)=E(I)*E(I)*MA(I)	FUNCT 5	
		QPART(I) = E(I) + MA(I)	FUNCT	
		ZWDOT(I) = -RU*WO*SIN(PT(I))*RAMP	FUNCT	
		U = X(1)*CX6-X(2)*SX6+ZWDUT(I)*SX6 VEL = VPART-X(3)*E(I)-ZWDUT(I)*CX6	FUNCT S	
		$V_{L} = V_{A} (I) + Z_{A} (J) + Z_{A} (J$	FUNCT	
		$Z_2(I) = -M_4(I) * COS(PT(I)) * RAMP$	FUNCT	
		$Z_3(I) = E(I) * Z_2(I)$	FUNCT S	
		Z4(I) = E(I)*Z1(I)	FUNCT 5	58
		Z5(1) = 0*22(1)	FUNCT S	59
		$Z_6(I) = E(I) * Z_5(I)$	FUNCT (	
		Z7(I) = MA(I)*VEL*U	FUNCT (	
		IF (VEL.LE.0.) GO TO 60	FUNCT (	
		IF (B1(I).LE.0.0) GO TO 50 DRDT = ZWDOT(I)*(X(I)+C+X(3)*(N(I)*CX6-E(I)*SX6))/C	FUNCT 6	
		D1(I) = VEL*B1(I)*(X(2)-X(3)*(CX6*E(I)*SX6*N(I)) - DRDT)	FUNCT	
			FUNCT 6	
	50	D1(I) = 0	FUNCT 6	
	51	CONTINUE	FUNCT 6	68
		D2(I) = E(I) * D1(I)	FUNCT 6	
		Cl(I) = VEL*VEL*B(I)	FUNCT	
		$C_2(I) = E(I) * C_1(I)$	FUNCT 1	
	40	GO TO 61	FUNCT	
	٥ů	D1(I) = 0, D2(I) = 0,	FUNCT	
		$C_{1}(I) = 0.$	FUNCT	
		$C_{2}(1) = 0.$	FUNCT	

	61	CONTINUE	FUNCT 77
		D3(I) = Z2(I)*VEL	FUNCT 78
		D4(I) = E(I) * D3(I)	FUNCT 79
		PIH = PI/2	FUNCT 80
•			
		D5(I) = B(I)*(Hw(I)-B(I)*TA/2.)	FUNCT 81
	66	D6(I) = D5(I) + E(I) + 5	FUNCT 82
	90	CONTINUE	FUNCT 83
		RHOG=RHÜ*GRAVTY	FUNCT 84
C ·		* * * * * SET UP THE FUNCTIONS FOR THE INTEGRALS (PAGE 5 OF NO	TES) FUNCT 85
•		PIH = PI/2.	FUNCT 86
			FUNCT 87
	<b>.</b>	KPI = KAR*PI	
С	÷١	VALUATE INTEGRALS USING TRAP METHOD	FUNCT 88
		I = 1	FUNCT 89
		INDEX = 1	FUNCT 90
	91	CALL TRAP(MA(INDEX), DIFF(I), KTT(I), TMASS)	FUNCT 91
		CALL TRAP (OPART (INDEX) DIFF (I) +KTT(I) +QA1)	FUNCT 92
		CALL TRAP(c1(INDEX),DIFF(I),KTT(I),CEA)	FUNCT 93
		CALL TRAP(C2(INDEX),DIFF(I),KTT(I),CE2A)	FUNCT 94
			FUNCT 95
		CALL TRAP(IPART(INDEX),DIFF(I),KTT(I),IAA)	
		CALL TRAP(D1(INDEX), DIFF(I), KTT(I), DMUA)	FUNCT 96
		CALL TRAP(D2(INDEX),DIFF(I),KTT(I),EDMUA)	FUNCT 97
		CALL TRAP(D3(INDEX), DIFF(I), KTT(I), E2DMUA)	FUNCT 98
		CALL TRAP(D4(INDEX), DIFF(I), KTT(I), E3DMUA)	FUNCT 99
		CALL TRAP(D5(INDEX),DIFF(I),KTT(I), BFA)	FUNCT100
		CALL TRAP (D6 (INDEX) , DIFF (I) , KTT (I) , BMMA)	FUNCT101
			FUNCT102
		CALL TRAP(Z1(INDEX),DIFF(I),KTT(I),ZMAA)	
		CALL TRAP(Z2(INDEX),DIFF(I),KTT(I),ZWMAA)	FUNCT103
		CALL TRAP(Z3(INDEX),DIFF(I),KTT(I),EMASA)	FUNCT104
		CALL TRAP(74(INDEX),DIFF(I),KTT(I),ZZWMAA)	FUNCT105
		CALL TRAP(75(INDEX),DIFF(I),KTT(I),ZWEMAA)	FUNCT106
		CALL TRAP(76(INDEX).DIFF(I).KTT(I).Z2WMAA)	FUNCT107
		CALL TRAP (77 (INDEX) DIFF (I) KTT (I) E2MAZA)	FUNCT108
с			FUNCT109
č	60	CONTINUE	FUNCT110
	,,,	MASS = MASS + TMASS	FUNCT111
		QA = QA + QA1	FUNCT112
		IA = IA + IAA	FUNCT113
		CE = CE + CEA	FUNCT114
		CE2 = CE2 + CE2A	FUNCT115
		DMU = DMU + DMUA	FUNCT116
		EDMU = EDMU + EDMUA	FUNCT117
		E2DMU = E2DMU + E2DMUA	FUNCT118
		E3DMU = E3DMU + E3DMUA	FUNCT119
		BF = BF + DHUG#BFA	FUNCT120
		BMM = BMM + RHOG+BMMA	FUNCT121
		$ZMA \approx ZMA + ZMA$	FUNCT122
		ZWMA = ZWMA + ZWMAA	FUNCT123
		EMAS = EMAS+EMASA	FUNCT124
		ZZWMA = ZZWMA+ZZWMAA	FUNCT125
		ZWEMA = ZWEMA+ZWEMAA	FUNCT126
		Z2WMA = Z2WMA+Z2WMAA	FUNCT127
		E2MAZ = E2MAZ+E2MAZA	FUNCT128
	94	CONTINUE	FUNCT129
		IF ( I.GE.II)GO TO 92	FUNCT130
			FUNCT131
		INDEX = INDEX+KTT(I)-1	
		I = I + 1	FUNCT132
		GO TO 91	FUNCT133
	92	CONTINUE	FUNCT134
С			FUNCT135

С	* * * * * * CALL COMPUT TO FIND THE VALUE OF NL AND FL USING	FUNCT136
Ċ	THE VALUES OF THE ABOVE INTEGRALS	FUNCT137
•	CALL COMPUT(X)	
~	CALL CONFOL(X)	FUNCT138
С		FUNCT139
	IF (NPRINT.LT.3) GO TO 111	FUNCT140
	IF(NPRINT_EQ.3) GO TO 108	FUNCT141
	IF (NPRINT_EQ_4)GO TO 108	FUNCT142
	WRITE(6,97) (IPART(I),I=1,NUM)	FUNCT143
	WRITE (6,98) (QPART(I), $I=1$ , NUM)	FUNCT144
	WRITE(6,99) (C1(I),I=1,NUM)	FUNCT145
	WRITE(6,100) (C2(I),I=1,NUM)	FUNCT146
	WRITE(6,101) (C3(1), I=1, NUM)	
		FUNCT147
	WRITE(6,102) (D1(I),I=1,NUM)	FUNCT148
	WRITE (6,103) (D2(I),I=1,NUM)	FUNCT149
	WRITE(6,104) (D3(I),I=1,NUM)	FUNCT150
	WRITE(6,105) (D4(I),I=1,NUM)	FUNCT151
	WRITE(6,106) (D5(I),I=1,NUM)	FUNCT152
	WRITE(6+112) (D6(I)+I=1+NUM)	FUNCT153
	WRITE (6,113) (Z1(I), I=1,NUM)	FUNCT154
	WRITE (6,114) (Z2(I), I=1, NUM)	FUNCT155
	WRITE (6,115) (23(I), I=1, NUM)	
	WRITE (6,116) (24(I), I=1, NUM)	FUNCT156
		FUNCT157
	WRITE(6.118)(Z5(I),I=1,NUM)	FUNCT158
	WRITE(6,119)(26(I),I=1,NUM)	FUNCT159
	WRITE(6+120)(27(I)+I=1+NUM)	FUNCT160
	WRITE(6,107)KPI,RHOG,PIH	FUNCT161
	108 WRITE(6,109) MASS,CINT,QA,CE,CE2,CE3	FUNCT162
	WRITE (6,121) IA	FUNCT163
	121 FORMAT(* IA *,E10,4)	FUNCT164
	WRITE(6,110)DMU,EDMU,E2DMU,E3DMU,BF,BMM	FUNCT165
	WRITE (6,117) ZMA, ZWMA, EMAS, ZZWMA, ZWEMA, ZZWMA, E2MAZ	FUNCT166
С	* * * * * * * * * * FORMATS * * * * * * * * * * * *	FUNCT167
Ŭ	96 FORMAT(" CPART(I)", 10(2x, E10.4))	FUNCT168
	97 FORMAT (" IPART (I)", 10 (2x, E10.4))	
	98 FORMAT(" OPART(I)",10(2X,E10.4))	FUNCT169
		FUNCT170
	99 FORMAT(" C1 ",10(2X,E10.4))	FUNCT171
	100 FORMAT(" C2 ",10(2X,E10.4))	FUNCT172
	101 FORMAT(" C3 ",10(2X,E10.4))	FUNCT173
	102 FORMAT(" D1 "+10(2X+E10+4))	FUNCT174
	103 FORMAT(" D2 "+10(2X+E10+4))	FUNCT175
	104 FORMAT(" D3 ",10(2X,E10.4))	FUNCT176
	105 FORMAT(" D4 ",10(2X,E10.4))	FUNCT177
	106 FORMAT(" D5 ",10(2X,E10.4))	FUNCT178
	112 FORMAT(" D6 ",10(2X,E10.4))	FUNCT179
	107 FORMAT(" KPHI "+E10.4, "RHUG "+E10.4," PHIH "+E10.4)	FUNCT180
	109 FORMAT (" MASS ",E10.4," CINT ",E10.4," QA ",E10.4," CE ",E10.4,	FUNCT181
	*"CE2 ",E10.4," CE3 ",E10.4)	FUNCT182
	110 FORMAT (" DMU ",E10,4," EDMU ",E10,4," E2DMU ",E10,4," E3DMU ",	FUNCT183
	*E10,49" BF ",E10,49" BMM ",E10,4)	FUNCT184
	113 FORMAT (4H Z1 )10(2X,E10.4))	FUNCT185
	114 FORMAT(4H Z2 +10(2X+E10+4))	FUNCT186
	115 FORMAT(4H Z3 ,10(2X,E10.4))	
		FUNCT187
	116 FORMAT(4H Z4 ,10(2X,E10.4))	FUNCT188
	118 FORMAT(4H Z5 ,10(2x,E10,4))	FUNCT189
	119 FORMAT (4H Z6 ,10(2X,E10.4))	FUNCT190
	120 FORHAT (4H Z7 ,10(2X,E10.4))	FUNCT191
	117 FORMAT (5H ZMA , E10.4,6H ZWMA , E10.4,6H EMAS , E10.4,	FUNCT192
	TH ZZWMA JE10.4, TH ZWEMA JE10.4, TH ZZWMA JE10.4,	FUNCT193
	• 7H E2MAZ ,E10.4)	FUNCT194

```
111 CONTINUE
                                                                               FUNCT195
      RETURN
                                                                               FUNCT196
                                                                               FUNCT197
      END
      SUBPOUTINE COMPUT(X)
                                                                               COMPLIT 2
      DIMENSION X(6)
                                                                               COMPLIT
                                                                                       7
      REAL KAR . KPT
                                                                               COMPLIT 4
      REAL NE MASS-NCG-M-IT-IA-K-MA-MMAX-N
                                                                               COMPLIT 5
      INTEGER END
                                                                               COMPLIT 6
c
                                                                               COMPLIT 7
      COMMON /SHIP/ MASS.CINT.QA.CE.CE2.CE3.DMU.EDMU.E2DMU.E3DMU.BF.BMM.COMPUT 8
                    NL .FL . IA.F (120)
                                                                               COMPLIT 9
      COMMON /CONST/ NCG.ECG.PI.DPR.RPD.GRAVTY.RHO.K.NUM.MA(120).CD.TA. COMPUTIO
                B(120) .BETA.HW(120) .TZ.DRAG.W.XD.T.XP.M.IT.
                                                                               COMPUT11
     .
                DELTAS, TX, EST (120) . C. RO, KAR, MMAX (1 0) . TEST (120) .
                                                                               COMPUT12
     .
                     N(120) .PHALF
                                                                               COMPUT13
     4
      COMMON/OUT/NPRINT .NPLOT .END
                                                                               COMPUT14
      COMMON /TEPMS/ T1.T2.T3.T4.T5.T6.T7.T8
                                                                               COMPLIT15
      COMMON /WAVE/ R(120), PT(120), ZMA, ZWMA, EMAS, ZZWMA, ZWEMA, Z2WMA,
                                                                               COMPUT16
                    E2MA7 . 7WDOT (120)
                                                                               COMPUT17
      COMMON /TEST/ VMA
                                                                               COMPUT18
с
                                                                               COMPLIT19
c
                                                                               COMPLIT20
                                                                               COMPLIT21
      Cx6 = C05(x(6))
      SX6 = SIN(X(6))
                                                                               COMPLIT22
      WO = K * C
                                                                               COMPUT23
      PIH = PI/2.0
                                                                               COMPUT24
      KPI = KAR#PI
                                                                               COMPUT25
      CONS1 = RO * WO * WO * CX6
                                                                               COMPUT26
      CONS2 = (KPI#RHU#PIH/TA)/CX6
                                                                               COMPUT27
      CONS3 = RU#WU#K#CX6#SX6
                                                                               COMPUT28
      CONS4 = R0+W0+K+CX6+CX6
                                                                               COMPLIT29
      TERM1 = X(1) * CX6
                                                                               COMPLIT 30
      TERM2 = X(2) + SX6
                                                                               COMPUT31
      UVNUM = (X(1)*CX6-(X(2)-ZWD0T(NUM))*SX6)*
                                                                               COMPUT32
               (X(1) *SX6-X(3) *E(NUM) + (X(2) -ZWDOT(NUM)) *CX6)
                                                                               COMPUT33
c
                                                                               COMPUT34
      ZMA = ZMA + x (3) + SX6
                                                                               COMPUT35
      77WMA = 77WMA#x(3)#5x6
                                                                               COMPLIT 36
      ZWMA = ZWMA+CUNS1
                                                                               COMPUT37
      EMAS = EMAS+CUNS1
                                                                               COMPUT38
      DMU = DMU+CONS2
                                                                               COMPUT39
      EDMU = EDMU#CONS2
                                                                               COMPLIT40
                                                                               COMPUT41
      CE = CE + CD + RHO
      CE2 = CE2 + CD + RHO
                                                                               COMPUT42
      E2DMU = E2DMU*CONS3
                                                                               COMPUT43
      E3DMU = E3DMU*CONS3
                                                                               COMPUT44
      ZWEMA = ZWEMA*CONS4
                                                                               COMPUT45
                                                                               COMPUT46
      Z2WMA = 72WMA#CONS4
С
                                                                               COMPUT47
   20 T1 = QA*X(3)*(TERM1-TERM2)
                                                                               COMPUT48
      T1 = T1 + ZZWMA - EMAS
                                                                               COMPUT49
      T2 = EDMU
                                                                               COMPLIT50
      T3 = CE2
                                                                               COMPUT51
      T4 = MA(NUM) *E(NUM) *UVNUM + E2MAZ + E3DMU - Z2WMA + BMM
                                                                               COMPLIT52
      NL = T1 + T2 + T3 + T4 + BMM
                                                                               COMPUT53
      T5 = MASS*x(3)*(TERM2-TERM1)
                                                                               COMPUT54
      T5 = T5 + ZWMA - ZMA
                                                                               COMPUT55
      T6 = -DMU
                                                                               COMPUT56
      T7 = -CE
                                                                               COMPUT57
```

```
63
```

```
TR = -MA (NUM) #UVNUM - F20MU + 7WEMA
                                                                             COMPLIT58
      BE = BE/CX6
                                                                             COMPLITES
                                                                             COMPUT60
C
                                                                             COMPUT61
      FI = T5+T6+T7+T8-BF
                                                                             COMPUT62
c
                                                                             COMPLITE3
      IF (NPRINT_LT_3)G0 TO 30
                                                                             COMPLIT64
      CONTINUE
   25
      WRITE (6 . 10) NL .FL
                                                                             COMPLIT65
                                                                             COMPLIT66
   10 FORMAT(" NL = ",E12.6," FL = ",E12.6)
                                                                             COMPUT67
   30 RETURN
                                                                             COMPLIT68
      END
      SUBROUTINE INPUT
                                                                             INPUT
                                                                                    2
         # # # DEFINITION OF INPUT VARIABLES
                                                                             INPUT
                                                                                    3
C#
         XA = INITIAL TIME
                                                                             INPUT
                                                                                    4
С
                                                                             INPIIT
                                                                                    5
Ċ
         XE = FINAL TIME
      HMIN = MINIMUM STEP SIZE
č
                                                                             INPIIT
                                                                                    6
č
      HMAX = MAXIMUM STEP SIZE
                                                                             INPUT
                                                                                     7
      FPSE = RELATIVE ERROR CRITERIUN USED FOR VALUES OF Y GT A
                                                                             INPUT
C
                                                                                    8
č
     FPS = FRRUP CRITERION IN KUTMER
                                                                             INPUT
                                                                                    ā
č
       A = ABSOLUTE ERROR CRITERIA USED IN KUTMER
                                                                             INPUT 10
                                                                             INPUT 11
ċ
        NPRINT = 1 FINAL PRINTUUT
č
                = 2 MATRIX INVERSE MATRIX.F COLUMN MATRIX.AND KUTHER
                                                                             INPUT 12
ċ
                      RESULTS
                                                                             INPUT 13
                                                                             INPUT 14
ē
                = 3 INTEGRAL VALUES
                = 4 CALCULATED VALUES-CONSTANT FOR GIVEN INPUT VALUES
                                                                             INPUT 15
ē
                                                                             INPUT 16
c
       NPLOT = 0
                    NO PLOT
                                                                             INPUT 17
С
              = 1
                   PRINTER PLOT
                                                                             INPUT 18
Ċ
         FND = NUMBER OF RUNS
                                                                             INPUT 19
с
č
         M = MASS OF CRAFT
                                                                             INPUT 20
                                                                             INPUT 21
с
          W = WEIGHT OF CRAFT
                                                                             INPUT 22
        TZ = THRUST COMPONENT IN Z DIRECTION
С
                                                                             INPUT 23
         TX = THRUST COMPONENT IN X DIRECTION
С
                                                                             INPUT 24
     XECG = DISTANCE FROM CG TO CENTER OF PRESSURE FOR NORMAL FORCE
С
         XP = MOMENT ARM OF PROPELLER THRUST
                                                                             INPUT 25
c
                                                                             INPUT 26
         XD = DISTANCE FROM CG TO CENTER OF PRESSURE FOR DRAG FURCE
č
      KA(I) = ADDED MASS COEFFICIENT
                                                                             INPUT 27
č
č
          AN ARRAY GIVEN THE VALUE KAR WHICH IS READ IN
                                                                             INPUT 28
č
      BM(I) = BEAM AT FREE SURFACE UR AT CHINE
                                                                              INPUT 29
С
      DRAG = FRICTION DRAG
                                                                             INPUT 30
                                                                              INPUT
ċ
          K = WAVE NUMBER
                                                                                    31
                                                                              INPUT 32
         RO = WAVE HEIGHT
с
                                                                              INPUT
                                                                                    33
С
        NU = WAVF SLOPE
                                                                              INPUT 34
Ċ
       NUM = NUMBER OF STATIONS
                                                                             INPUT 35
с
         BL = BOAT LENGTH
                                                                             INPUT 36
      LAMBDA = WAVE LENGTH
C
                                                                             INPUT 37
          RG = RADIUS OF GENERATION IN FEET
С
                                                                             INPUT 38
          T = PROPELLED THRUST IN LBS
С
                                                                             INPUT 39
С
      GAMMA = PROPELLER THRUST ANGLE IN DEGREES
                                                                             INPUT 40
Ċ
      DELTAS=STATION SPACING IN FEET
                                                                             INPUT 41
с
         ECG = LONGITUDINAL CENTER OF GRAVITY
                                                                             INPUT 42
с
         NCG = VEDTICAL CG
                                                                             INPUT 43
č
      BETA(I) = DEAD RISE
                                                                              INPUT 44
      NO(I) = HEIGHT OF MEAN BUTTOCK
С
                                                                              INPUT 45
         RHO = DENSITY OF WATER
с
č
                                                                              INPUT 46
      GRAVTY = GPAVITY FT/SEC*#2
                                                                              INPUT 47
с
       DPR = DEGREES PER RADIAN
         RPD = RADIANS PER DEGREE
                                                                              INPUT 48
с
                                                                             INPUT 49
С
          PI = 3.14159 . . . .
```

```
c
    EST(I) = STATION POSITION
                                                                             INPLIT 50
    START = START TIME OF THE RAMP FUNCTION FOR SEA WAVE
                                                                             INPUT 51
C
č
      RISE = DURATION OF THE RISE FROM ZERO TO ONE OF THE RAMP
                                                                             INPLIT
                                                                                    52
                                                                             INPUT
                                                                                   53
Ċ
        * * * * * TC OPTIONS
č
    . .
                                                                             INPUT
                                                                                   54
  -
Ċ
                                                                             INPUT
                                                                                   55
                =1 USE WAVE Z DISTANCE IN COMPUTING LIFT COMPONENT
č
         TC(1)
                                                                             INPUT
                                                                                   56
                                                                             INPUT
Ċ
                    OF NE AND FL
                                                                                   57
                                                                             INPUT
                                                                                   58
C
c
                                                                             INPUT
                                                                                   59
      REAL IT .K .I AMBDA .M. MA. MMAX .NII.N. NCG.NU. MASS.NL. IA.KAR
                                                                             INPUT
                                                                                   60
      INTEGER END
                                                                             INPUT 61
                                                                             INPUT 62
C
                                                                             INPUT 63
ċ
      COMMON /CONST/ NCG.ECG.PI.DPR.RPD.GRAVTY.RHO.K.NUM.MA(120).CD.TA. INPUT 64
                B(120), BETA, HW(120), TZ, DRAG, W, XD. T. XP. M. IT.
                                                                             INPUT 65
                DELTAS.TX.EST(120).C.RO.KAR.MMAX(120).TEST(120).
                                                                             INPUIT 66
     a.
                     N(120) . PHALF
                                                                             INPUT 67
      COMMON /SHIP/ MASS.CINT.QA.CE.CE2.CE3.DMU.EDMU.E2DMU.E3DMU.RF.RMM.INPUT 68
                    NL +FL + TA +F(120)
                                                                             INPLIT 69
      COMMON /IN/ BM(120) .B1(120) .VELIN
                                                                             INPUT 70
      COMMON /IN2/ NO(120) *XA*XE*HMAX*HMIN*A(6)*EPSE(6)*LAMBDA
                                                                             INPUT 71
                                                                             INPUT 72
      COMMON/QUIT/NPRINT • NPLOT • END
      COMMON /ACCEL/ XACCL, BWACL, CGACL, BL
                                                                             INPUT 73
                                                                             INPUT 74
С
      NAMELIST/HSP/A.NPRINT.NPLUT.END.W.HL.TZ.TX.XECG.XP.XD.
                                                                             INPUT 75
                      DRAG.RG.T.GAMMA.ECG.NCG.KAR.RO.LAMBDA.NUM.BETA.EST INPUT 76
     .
                      .XA.XE.HMIN.HMAX.EPS.VELIN
                                                                             INPUT 77
с
                                                                             INPUT 78
      DATA A /.01..0001..00001..1..0001.00001/
                                                                             INPUT
                                                                                   79
      DATA NPRINT NPL OT FND/1.1.1.1/
                                                                             INPUT
                                                                                   80
      DATA W.BL.TZ.TX.XECG.XP.XD.DRAG.RU.LAMBDA.RG.T.GAMMA.
                                                                             INPUIT
                                                                                   81
            ECG.NCG.KAR /16..3.75.6+0.0..0416.22.5.9562.2+0.0.
                                                                             INPUT 82
     ×
                       2.325,0.0,1.0/
                                                                             INPUT 83
      DATA NUM.BETA.EST /77.20.0.
                                                                             INPUT
                                                                                   84
         0.0000.03125.06250.09375.12500.15625.18750.21875.
                                                                             INPUT
                                                                                   85
     -
          .25000..28125..31250..34375..37500..40625..43750..46875.
                                                                             INPLIT
                                                                                   86
          .50000,.53125,.56250,.59375,.62500,.65625,.6875,.71875,
                                                                             INPUT
                                                                                   87
     di
          -75000, 78125, 81250, 84375, 87500, 90625, 93750, 96875, 1,000, INPUT 88
     4
                                                                             INPUT
         1,06250,1,12500,1,18750,1,25000,1,3125,1,37500,1,4375,
                                                                                   89
         1,500,1,5625,1,625,1,6875,1,75,1,8125,1,875,1,9375,2,0,
                                                                             INPUT
                                                                                   90
         2,0625,2,125,2,1875,2,25,2,3125,2,375,2,4375,2,5,2,5625,2,625, INPUT
                                                                                   91
     ×
         2,6875,2,75,2,8125,2,8750,2,9375,3,0,3,0625,3,125,3,1875,
                                                                             INPUT
                                                                                   92
         3,2500,3,3125 ,3,375,3,4375,3,5,3,5625,3,625,3,6875,3,75 /
                                                                             INPLIT
                                                                                   93
                                                                             INPUT
      DATA XA, XE, HMIN, HMAX, EPS /0.0, 20.0, 025, 1, 15/
                                                                                   94
                                                                             INPUT
                                                                                   95
      DATA VELIN /19.62/
                                                                             INPUT
                                                                                   96
С
                                                                             INPUT 97
Ċ
    * * * * * * READ IN AND WRITE OUT KUTMER PARAMETERS AND PROGRAM
  .....
       OPTIONS
                                                                             INPUT 98
C
      READ (5,HSP)
                                                                             INPLIT 99
      WRITE (6.HSP)
                                                                             INPUT100
      00 10 I=1.6
                                                                             INPUT101
   10 EPSE(I) = FPS
                                                                             INPUT102
                                                                             INPUT103
С
  - 14
    - 6
      * * * * * * SET UP CONSTANTS
                                                                             INPUT104
      PI = 3.141592653589
                                                                             INPUT105
                                                                             INPUT106
      GRAVTY=32.18
      DPR=57.29577951308
                                                                             INPUT107
      RPD=.017453292519
                                                                             INPUT108
```

```
IE (EST (NUM) - LT. 3.75) STOP 3
                                                                                INPUT109
c
c
                                                                               INPUT110
          COMPUTE NO AND BM APPAYS
                                                                               INPUT111
č
                                                                               INPUT112
      DO 32 1=1.NUM
                                                                                [NPUT113
       IF (EST(I) .GE.0.75) GO TO 30
                                                                               INPUT114
      NO(T) = -0.46875*(1.0-S0RT(EST(1)/0.375-(EST(1)/0.75)**2.0))
                                                                               INPUT115
      BM(T) = .375 * SQRT(1, 0 - (FST(1)/.75 - 1, ) * *2.0)
                                                                               INPUT116
                                                                               INPUT117
      60 10 32
   30 NO(1)=0.0
                                                                               INPUT118
       BM(I) = 0.375
                                                                               INPUT119
   32 CONTINUE
                                                                               INPUT120
C*******COMPUTE CONSTANTS AND INITIALIZE ARRAYS
                                                                               INPUT121
       M=W/GRAVTY
                                                                               INPUT122
                                                                               INPUT123
       RH0=1.99
       IT=M#RG#RG
                                                                               INPUT124
       K = 2. +PI/LAMBDA
                                                                               INPUT125
       C=SORT (GRAVIY/K)
                                                                               INPUT 126
      NU=R0#K
                                                                               INPUT127
      PHALF = (PI/2_{\bullet}) * RH0
                                                                               INPUT128
с
                                                                               INPUT129
      BETA = HETA*RPD
                                                                               INPUT130
       CD = COS(BFTA)
                                                                               INPUT131
       TA = TAN (BETA)
                                                                               INPUT132
                                                                                INPUT133
       DO 60
             I=1.NUM
       E(I) = ECG - EST(I)
                                                                               INPUT134
       N(I) = NCG+NO(I)
                                                                               INPUT135
       MMAX(I) = KAR*PHALF*BM(I)*BM(I)
                                                                               INPUT136
                                                                                INPLIT137
       TEST(I) = (2*BM(I)*TA)/PI
   60 CONTINUE
                                                                               INPUT138
                                                                                INPUT139
       END=END+1
                                                                                INPUT140
       RETURN
       END
                                                                               INPUT141
       SUBROUTINE PLUTER (EX.XA.HMAX.LAMBUA.IB.NWAVE.IPT)
                                                                               PLOTER 2
                                                                               PLOTER 3
с
    INPUT:
                                                                               PLOTER 4
00000000000
                     A TWO DIMENSIONAL ARRAY CONTAINING PITCH AND
                                                                               PLOTER 5
          FX
                                                                               PLOTER 6
                     HEAVE VALUES AT EACH TIME STEP
                                                                               PLOTER 7
          ΧA
                     INITIAL TIME
          HMAX
                     TIME INTERVAL. PTIME * HMAX = INTERVAL BETWEEN
                                                                               PLOTER 8
                                                                               PLOTER 9
                     EX VALUES
                     WAVELENGTH USED IN CALCULATING PITCH AND
                                                                               PLOTER10
          LAMBDA
                     HEAVE RATIOES
                                                                               PLOTER11
          TR
                     NUMBER OF FX VALUES
                                                                               PLOTER12
          NWAVE
                     START OF VALUES AFTER WAVE IS COMPLETELY ON
                                                                               PLOTER13
с
                                                                               PLOTER14
č
                                                                               PLOTER15
                                                                               PLOTER16
       REAL IT .K .L AMBDA .M.MA .MMAX .N.NCG
                                                                               PLOTER17
       INTEGER END
                                                                               PLOTER18
С
       DIMENSION FX(2.400) .FMIN(2) .FMAX(2) .NVAR(2)
                                                                               PLOTER19
С
                                                                               PLOTER20
                                                                               PLOTER21
       COMMON /CONST/ NCG, ECG, PI, DPR, RPD, GRAVTY, RHO, K, NUM, MA(120), CD, TA,
                                                                               PLOTER22
                       B(120), BETA, HW(120), TZ, DRAG, W, XD, T, XP, M, IT,
      ٠
                       DELTAS.TX.EST(120).C.RU.KA.MMAX(120).TEST(120).
                                                                               PLOTER23
      .
                      N(120) .PHALF
                                                                               PLOTER24
      .
                                                                               PLOTER25
       COMMON/OUT/NPRINT,NPLOT,END
                                                                               PLOTER26
С
č
           * * * * SET UP VALUES FOR PLOT AND CREATE PLOT
                                                                               PLOTER27
    8
```

	NFUN=2		PLOTE	R28
C # #	* * * * * * *	SET UP MIN AND MAX LIMITS FOR PLOT	PLOTE	R29
	FMIN(1)=FX	(1+1)	PLOTE	R30
	FNIN(2)=FX	(2.1)	PLOTE	R31
	FMAX(1)=FX	(1,1)	PLOTE	R32
	FMAX(2)=FX	(2,1)	PLOTE	R33
C + +		SET UP MIN AND MAX LIMIMTS FOR FITCH AND HEAVE RATIO	PLOTE	R34
•	FMNP=FX(2)		PLOTE	R35
	FMXP=FX(2.		PLOTE	
	FMNH=FX(1)		PLOTE	
	FMXH=FX(1)		PLOTE	
с	1 1001-1 0 111		PLOTE	
C	DO 200 I=1	18	PLOTE	
		LT.FMIN(1))FMIN(1)=FX(1,I)	PLOTE	
		.GT.FMAX(1))FMAX(1)=FX(1,1)	PLOTE	
		LT.FMIN(2))FMIN(2)=FX(2,1)	PLOTE	
		GT • FMAX (2) > FMAX (2) = FX (2) 1)	PLOTE	
		AVE)GO TO 200	PLOTE	
		LT.FMNH)FMNH=FX(1,1)	PLOTE	
		GT.FMXH)FMXH=FX(1,I)	PLOTE	
		LT.FMNP)FMNP=FX(2,1)	PLOTE	
		GT.FMXP)FMXP=FX(2,I)	PLOTE	
200	CONTINUE		PLOTE	
		0) GO TO 800	PLOTE	
C * *		COMPUTE RATIOES	PLOTE	
		xH-FMNH)/(2,*R0)	PLOTE	
		xP-FHNP)/((4.*PI*RO)/LAMBDA)	PLOTE	
		0) COL3,COL4	PLOTE	
700		" HEAVE AMPLITUDE/WAVEHEIGHT = ",E12.6,/,2X,	PLOTE	
	• • PI1	TCH AMPLITUDE/(2.*PI*WAVEHEIGHT/LAMBDA) = ".E12.6)	PLOTE	
С			PLOTE	
800	CONTINUE		PLOTE	R59
-	NVAR(1)=10	H HEAVE	PLOTE	R60
	NVAR(2)=10	H PITCH	PLOTE	R61
	N1=2		PLOTE	R62
	XO=XA		PLOTE	R63
	DELX = HMA	X	PLOTE	R64
	IF (NPLOT .E	<pre>Deltain and a second seco</pre>	PLOTE	R65
	RETURN		PLOTE	R66
	END		PLOTE	R67
	SUBROUTINE	TRAP (F+DX+NPTS+ANS)	TRAP	2
С			TRAP	3
	NPUT:		TRAP	4
С	F	ARRAY OF FUNCTIONAL VALUES OF THE INTEGRAND	TRAP	5
С	DX	THE X INTERVAL BETWEEN VALUES	TRAP	6
С	NPTS	THE NUMBER OF VALUES GIVEN	TRAP	7
	UTPUTI		TRAP	8
С	ANS	THE VALUE OF THE INTEGRAL	TRAP	9
с			TRAP	10
	DIMENSION P	F (NPTS)	TRAP	11
	ANS=0.0		TRAP	12
	IF (NPTS.LT.	2)GO TO 999	TRAP	13
	DO 1 I=1.N	PTS	TRAP	14
1	ANS=ANS+F (	I)	TRAP	15
		5-0+5#(F(1)+F(NPTS)))	TRAP	16
999	CONTINUE		TRAP	17
	RETURN		TRAP	18
	END		TRAP	19
	FUNCTION R	MP(T,START,RISE)	RMP	2

C + +		THIS FUNCTION IS USED TO GRADUALLY IMPLIMENT THE WAVE	RMP	3	
С			RMP	4	
Ċ	т	CURRENT TIME	RMP	5	
С	START	TIME TO START RAMP FRUM 0.0 TO 1.0	RMP	6	
C	RISE	THE LENGTH OF THE RISE FROM 0.0 TO 1.0	RMP	7	
Ċ			RMP	8	
-	H=0.0		RMP	9	
	IF (T.LT.STA	RT)GO TO 99	RMP	10	
	IF (RISE - EQ. 0.0) GO TO 80				
	TOP=T-START			12	
	H=1.0		RMP	13	
	IF (TOP.LT.PISE) H=TOP/RISE			14	
	GO TO 99		RMP	15	
80	) H=1.		RMP	16	
	IF (T.EQ.START) H=0.5			17	
99			RMP	18	
	RETURN		RMP	19	
	END		RMP	20	

# LISTING OF COMPUTER PROGRAM FOR CALCOMP PLOTS

PROGRAM PLTHSP(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE7,TAPE9)	MAIN	2
ITAPE = 7	MAIN	3
CALL CALPLT (ITAPE)	MAIN	4
	MAIN	ŝ
STOP		6
END	MAIN	
SUBROUTINE CALPLT(ITAPE)	CALP	2
DIMENSIUN TIMĖ(4003),PITCH(4003),HEAVE(4003)	CALP	3
• , IBUF (1000) , BWACL (4003) , CGACL (4003)	CALP	4
LOGICAL ACCEL	CALP	5
	CALP	6
CAL CUMP PLOT OF PITCH AND HEAVE VERSUS TIME	CALP	7
CAE COMP FEUT OF FILCH AND HEAVE VERSUS TIME	CALP	8
10540 - 5	CALP	9
IREAD = 5	CALP	10
READ(IREAD.10) XAXIS,YAXISP,YAXISH,HT		
10 FORMAT(8F10.0)	CALP	11
ACCEL = .FALSE.	CALP	12
READ(IRÉAD.20) IA	CALP	13
29 FORMAT(110)	CALP	14
IF (IA, EQ.1) ACCEL = TRUE.	CALP	15
IF (ACCEL) DEAD (IREAD, 10) YAXISB, YAXISC	CALP	16
CALL READT (TIME, HEAVE, PITCH, BWACL, CGACL, NPTS)	CALP	17
	CALP	18
CALL PLOTS(IBUF, 1000,7)		
CALL PLUT (0.5,1.0,-3)	CALP	19
CALL ESCALE (TIME, XAXIS, NPTS, 1)	CALP	20
CALL ESCALE (HEAVE, YAXISH, NPTS, 1)	CALP	21
CALL ESCALE (PITCH, YAXISP, NPTS, 1)	CALP	22
IF (ACCEL) CALL ESCALE (BWACL, YAXISH, NPTS, 1)	CALP	23
IF (ACCEL) CALL ESCALE (CGACL, YAXISC, NPTS, 1)	CALP	24
NI = NPTS+1	CALP	25
N2 = NPTS+2	CALP	26
$N_2 = N_1 + 3 + 3$ N3 = NPTS+3	CALP	27
	CALP	28
CALL EAXIS(0.0,0.0,15HTIME IN SECUNDS,-15,XAXIS,0.0,	CALP	29
• TIME (N1) , TIME (N2) , TIME (N3) , HT)		
CALL EAXIS(0.0,0.0,13HHEAVE IN FELT,13,YAXISH,90.0,	CALP	30
HEAVE (N1) + EAVE (N2) + EAVE (N3) + HT)	CALP	31
TEMP = TIME(N2)	CALP	32
TIME(N2) = TIME(N2)/TIME(N3)	CALP	33
HEAVE (N2) = HEAVE (N2)/HEAVE (N3)	CALP	34
CALL LINE (TIME, HEAVE, NPTS, 1,0,0)	CALP	35
TIME(N2) = TEMP	CALP	36
XNEW = XAXIS+3.	CALP	37
YNEW = 1.0	CALP	38
CALL PLUT (XNEW,0.0,-3)	CALP	39
CALL EAXIS(0.0,0.0,15HTIME IN SECUNDS,-15,XAXIS,0.0,	CALP	40
CALL CAALS(U. U. U. THE AND TH	CALP	41
• TIME (N1) +TIME (N2) +TIME (N3) +HT)	CALP	42
CALL EAXIS(0.0,0.0.13HPITCH IN RAD.,13,YAXISP,90.0,		
<ul> <li>PITCH(N1),PITCH(N2),PITCH(N3),HT)</li> </ul>	CALP	43
TIME(N2) = TIME(N2)/TIME(N3)	CALP	44
PITCH(N2) = PITCH(N2)/PITCH(N3)	CALP	45
CALL LINE(TIME,PITCH,NPTS,1,0,0)	CALP	46
IF(.NUT.ACCEL) GO TO 30	CALP	47
TIME(N2) = TEMP	CALP	48
CALL PLUT (XNEW,0.0,-3)	CALP	49
CALL EAXIS(0.0,0.0,15HTIME IN SECUNDS,-15,XAXIS,0.0,TIME(N1),	CALP	50
. TIMÉ (N2) • TIME (N3) • HT)	CALP	51
CALL EAXIS(0.0,0.0,16HBUW ACCELERATION,16,YAXISB,90.0,BWACL(N1)		52
	CALP	53
	CALP	54
TIME(N2) = TIME(N2)/TIME(N3)		55
BWACL(N2) = BWACL(N2)/BWACL(N3)	CALP	22

CALL LINE(TIME,BWACL,NPTS,1,0,0)	CALP	56
	CALP	57
TIME(N2) = TEMP	CALP	58
CALL PLUT(XNEW,0.0,-3)	CALP	59
CALL EAXIS(0.0,0.0,15HTIME IN SECONDS,-15,XAXIS,0.0,TIME(N1),	CALP	60
• TIME (N2) • TIME (N3) • HT)	CALP	61
CALL EAXIS(0.0,0.0,15HCG ACCELERATION,15,YAXISC,90.0,CGACL(N1),		62
ČGACL (N2) + CGACL (N3) + HT)	CALP	63
TIME(N2) = TIME(N2)/TIME(N3)	CALP	64
CGACL(N2) = CGACL(N2)/CGACL(N3)	CALP	65
CALL LINE (TIME, CGACL, NPTS, 1, 0, 0)	CALP	66
30 CONTINUE	CALP	67
CALL PLUT (30.0,0,999)	CALP	68
RETURN	CALP	69
END	CALP	70
SUBROUTINE READT(TIME,HEAVE,PITCH,BWACL,CGACL,NPTS)	READ	2
DIMENSION X(6)+HEAVE(1)+PITCH(1)	READ	3
+ ,TIME(1),BWACL(1),CGACL(1)	READ	4
I = 0	READ	5
5 CONTINUE	READ	6
I = I + I	READ	7
READ(9) TIME(I),(X(I),I=4,6),BWACL(I),CGACL(I)	READ	8
IF (EOF (9))10,15	READ	9
15 CONTINUE	READ	10
WRITE(6+20) TIME(I)+(X(J)+J=4+6)+BWACL(I)+CGACL(I)	READ	11
20 FORMAT(1H +6(F7+2+2X))	READ	12
HEAVE(I) = X(5)	READ	13
$PITCH(I) = \chi(6)$	READ	14
IF(1.6E.4000) GO TO 10	READ	15
GO TO 5	READ	16
10 CONTINUE	READ	17
NPTS = $I-1$	READ	18
RETURN	READ	19
END	READ	20
SUBROUTINE EAXIS (XPAGE, YPAGE, IBCD, NCHAR, AXLEN, ANGLE, FIRSTV,	EAXIS	
DELTAV, DELTAV, HT)	EAXIS	
DIMENSION IBCD(1)	EAXIS	
THIS RUDTINE WORKS LIKE THE CALCOMP AXIS WITH THE	EAXIS	
EXCEPTION THAT THE TICK MARKS ARE NOT NECCESSARILY	EAXIS	
EVERY INCH AND THE HEIGHT OF THE CHARACTERS IS INPUTTED	EAXIS	
EVERT INCH AND THE HETOTY OF THE CHARACTERS IS THEOTHER	EAXIS	
CALL PLUT (XPAGE, YPAGE, 3)	EAXIS	-
ISN = ISIGN(1, NCHAR)	EAXIS	
ISGN = SIGN(1., DELTAV)	EAXIS	
AMIN = FIRSTV	EAXIS	
X = XPAGE	EAXIS	
Y = YPAGE	EAXIS	
XNUM = FIRSTV-DELTAV	EAXIS	16
N = AXLEN/DELTAU	EAXIS	17
IF (N+DELTAU.LT.AXLEN) N=N+1	EAXIS	18
AMAX = AMIN+(N*DELTAV)	EAXIS	19
NDIG = NDIGIT (AMIN, AMAX, DELTAU, ND)	EAXIS	
10 CONTINUE	EAXIS	
TEST = (NDIG*HT) + HT	EAXIS	
IF (TEST.GT.DELTAU) HT=HT/2.	EAXIS	
IF(TEST.GT.DELTAU) GO TO 10	EAXIS	
AYN = (1.5  PHT)	EAXIS	
BYN = (((NDIG-2)*HT)/2.+.5*HT)	EAXIS	26

с

```
N = N+1
                                                                             EAXIS 27
     TANG = (90.+ANGLE)/57.295A
                                                                             EAXIS 28
     ANG = ANGLE/57.2958
                                                                             EAXIS 29
           = SIN(TANG)
     ST
                                                                             EAXIS 30
     ĊŤ
           = COS(TANG)
                                                                             FAXIS 31
     c
           = STN(ANG)
                                                                             EAXIS 32
     č
           = COS(ANG)
                                                                             EAXIS 33
   00 30 I=1.N
                                                                             FAXIS 34
     IF (I.EQ.1) GO TO 20
                                                                             EAXIS 35
     X = X + OFI TAU#C
                                                                             EAXIS 36
     Y = Y+DEL TAU*S
                                                                             EAXIS 37
     CALL PLOT (X+Y+2)
                                                                            EAXIS 38
   IF(I.EQ.N) GU TO 20
                                                                             EAXIS 39
     XT = X+(.)+CT+ISN)
                                                                            EAXIS 40
     YT = Y + (.1*ST*ISN)
                                                                             EAXIS 41
     CALL PLUT (XT.YT.2)
                                                                             FAXIS 42
20 XN = X+AYN+CT*ISN-BYN+C
                                                                             FAXIS 43
   YN = Y+AYN*ST*ISN-BYN*S
                                                                            EAXIS 44
   XNUM = XNUM+DELTAV
                                                                            FAXIS 45
     CALL NUMBER (XN. YN. HT. XNUM. ANGLE .ND)
                                                                            EAXIS 46
     CALL PLUT (X,Y,3)
                                                                            EAXIS 47
30 CONTINUE
                                                                            EAXIS 48
   XSP = (((AXLEN/HT)/2_{\bullet}) - (IABS(NCHAR)/2_{\bullet})) * HT
                                                                            EAXIS 49
   YSP = 3.5*HT
                                                                            EAXIS 50
     XT = XPAGE + XSP*C + ISN*YSP*CT
                                                                            EAXIS 51
     YT = YPAGE + XSP#S + ISN#YSP#ST
                                                                            FAXIS 52
                                                                            EAXIS 53
     CALL SYMBOL (XT.YT.HT.IBCD.ANGLE.IABS(NCHAR))
   RETURN
                                                                            FAXIS 54
   END
                                                                            EAXIS 55
   FUNCTION NDIGIT (AMIN, AMAX, ANUM, ND)
                                                                            NDIG
                                                                                    2
                                                                            NDIG
                                                                                    з
      FINDS THE NUMBER OF DIGITS NECCESSARY TO PRINT
                                                                            NDIG
                                                                                    4
      EVEN INCREMENT OF THE FUNCTION ON THE AXIS
                                                                                    5
                                                                            NDIG
                                                                            NOTG
                                                                                    6
      NDIGIT
                 THE NUMBER OF PLACES IN THE ENTIRE NUMBER
                                                                            NDIG
                                                                                    7
      ND
                 THE NUMBER OF DECIMAL PLACES
                                                                            NDIG
                                                                                    8
      ΔΝΠΜ
                 THE VALUE GIVEN TO EACH INCREMENT ON THE AXIS
                                                                            NDIG
                                                                                    ġ
                                                                            NDIG
                                                                                   10
   IF (ABS(AMIN) .LT. ABS(AMAX)) GO TO 20
                                                                            NDIG
                                                                                   11
   IF (AHS (AMIN) . EQ. ABS (AMAX) . AND . AMAX . NE. 0) GO TU 20
                                                                            NDIG
                                                                                   12
   IF (ABS(AMIN).GT.ABS(AMAX)) GO TO 10
                                                                            NDIG
                                                                                   13
     AMAX = 1.
                                                                            NDIG
                                                                                   14
     AMIN = -1
                                                                            NDIG
                                                                                   15
     GO TO 20
                                                                            NDIG
                                                                                   16
10
       AMAX = ABS(AMIN)
                                                                            NDIG
                                                                                   17
20 IF (AMAX.LE.1.) GU TO 50
                                                                            NDIG
                                                                                   18
     NDIV = 10
                                                                            NDIG
                                                                                   19
          = 1
     T
                                                                            NDIG
                                                                                   20
30
       IF (AMAX/NDIV.LT.1) GO TO 40
                                                                            NDIG
                                                                                   21
       I = I + 1
                                                                            NDIG
                                                                                   22
       NDIV = NDIV*10
                                                                            NDIG
                                                                                   23
       GU TU 30
                                                                            NDIG
                                                                                   24
40
     NDIGIT = I+3
                                                                            NDIG
                                                                                   25
   ND = 2
                                                                            NDIG
                                                                                   26
   GO TO 80
                                                                            NDIG
                                                                                   27
50 NDIV = 10
                                                                            NDIG
                                                                                   28
        = 1
                                                                            NDIG
   а.
                                                                                   29
60
     IF (AMAX*NDIV.GT.1.) GO TO 70
                                                                                   30
                                                                            NDIG
     I = I + 1
                                                                            NDIG
                                                                                   31
```

```
0000000
```

ć

```
NOTV = NOTV#10
                                                                           NDIG
                                                                                  32
                                                                           NDIG
                                                                                  33
     GO TO 60
                                                                           NDIG
                                                                                  34
70 NDIGIT = I+2
                                                                           NOTO
   ND = I
                                                                                  35
                                                                           NDIG
                                                                                  36
80 DD = FLUAT(ND)
                                                                           NDIG
                                                                                  37
   X = ANUM*(10**DD)
                                                                           NOTO
                                                                                  38
   \mathbf{1}\mathbf{X} = \mathbf{X}
                                                                           NDIG
                                                                                  30
   IF (X-FLUAT (IX) .LT .. 0001) GO TU 90
                                                                           NOTO
                                                                                  40
   DD = DD+1
                                                                           NDIG
                                                                                  41
   ND = ND+1
                                                                           NDIG
                                                                                  42
   NDIGIT = NDIGIT+1
                                                                                  43
                                                                           NDIG
   GO TO 80
                                                                           NDIG
                                                                                  44
90 CONTINUÉ
                                                                           NDIG
                                                                                  45
   RETURN
                                                                           NDTG
                                                                                  46
   END
                                                                           ESCAL
                                                                                   2
   SUBROUTINE ESCALE (ARRAY+AXLEN+NPTS+INC)
                                                                                   3
                                                                           ESCAL
      FINDS THE SCALE TO BE USED ON THE AXIS -
                                                                           FSCAL
                                                                                   4
                                                                                   5
      ARRAY MUST HAS THREE UNUSED POSITIONS
                                                                           ESCAL
                                                                           ESCAL
            ARPAY (NPTS+1) = FIRSTV
                                                                                   6
                                      (THE INCREMENT BETWEEN TICK MARKS ESCAL
            ARPAY(NPTS+2) = DELTAV
                                                                                   7
                                          VALUES - NUMBERS)
                                                                           ESCAL
                                                                                   8
                                                                                   9
            APDAY (NPTS+3) = DELTAU (THE INCREMENT IN INCHES
                                                                           ESCAL
                                      BETWEEN TICK MARKS )
                                                                            ESCAL 10
                                                                            ESCAL 11
                                                                            ESCAL 12
                                                                            ESCAL 13
   DIMENSION APPAY(1)
                                                                            ESCAL 14
   AMIN = ARRAY(1)
                                                                            ESCAL 15
   AMAX = ARRAY(1)
                                                                            ESCAL 16
   ISGN = ISIGN(1 \cdot INC)
   INC = IABS(INC)
                                                                            ESCAL 17
                                                                            ESCAL 18
       DO 10 T=1+NPTS+INC
          IF (ADRAY(1).LT.AMIN) AMIN=ARRAY(1)
                                                                            ESCAL 19
          IF (ARRAY (I) . GT. AMAX) AMAX=ARRAY (I)
                                                                            ESCAL 20
10
        CONTINUE
                                                                            ESCAL 21
      AUNIT = UNIT (AMIN.AMAX.AXLEN.N.ANUM)
                                                                           ESCAL 22
20
    CALL AUJUST (AMIN, AMAX, AUNIT, AXLEN, N, ANUM)
                                                                            ESCAL 23
                                                                            ESCAL 24
   ARRAY(NPTS+1) = AMIN
                                                                            ESCAL 25
   ARRAY (NPTS+2) = ANUM#ISGN
                                                                            ESCAL 26
      IF (ISGN.FQ.-1) ARRAY (NPTS+1) = AMAX
                                                                            ESCAL 27
   ARRAY(NPTS+3) = AUNIT
   IF (ABS(ANUM).EQ.AUNIT) ARRAY(NPTS+2) = 1.*ISGN
                                                                            ESCAL 28
                                                                           ESCAL 29
   IF (ABS(ANUM) .EQ.AUNIT) ARRAY(NPTS+3) = 1.
                                                                            ESCAL 30
   RETURN
                                                                            ESCAL 31
   END
                                                                            JUST
                                                                                   2
   SUBROUTINE ADJUST (AMIN, AMAX, AUNIT, AXLEN, N, ANUM)
                                                                            JUST
                                                                                   3
       GIVEN AMIN AND AMAX WHICH ARE DISTINCT VALUES, ADJUST
                                                                            JUST
                                                                                   4
                                                                            JUST
                                                                                   5
      THEM SO THAT THEY ARE EVEN MULTIPLES OF AUNIT
                                                                                   6
                                                                            JUST
                                                                                    7
                                                                            JUST
   K = 1
                                                                            JUST
                                                                                   8
   MIN = AMIN/ANUM
   IF (AMIN.LT.MIN*ANUM) MIN = MIN-1
                                                                            JUST
                                                                                   9
   AMIN = MIN*ANUM
                                                                            JUST
                                                                                  10
   MAX = AMAX/ANUM
                                                                            JUST
                                                                                  11
    IF (AMAX.GT.MAX#ANUM) MAX = MAX+1
                                                                            JUST
                                                                                  12
    AMAX = MAX+ANUM
                                                                                  13
                                                                            JUST
                                                                            JUST
                                                                                  14
10 TERM = AMIN+ (N-K) *ANUM
                                                                                  15
                                                                            JUST
    IF (TERM.LT.AMAX) GO TO 20
```

```
00000000000
```

с с с с

-	K = K+1 GO TO 10 AUNIT = AXLEN/(N-K+1) N = AXLEN/AUNIT+1 RETURN END FUNCTION UNIT(AMIN,AMAX,AXLEN,N,ANUM) FINDS THE INCREMENT BETWEEN VALUES TO BE USED ON THE AXIS IN 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	נ נ נ ט ט ט ט ט ט ט ט ט ט ט ט ט ט ט ט ט	UST UST UST UST UST NIT NIT NIT NIT	16 17 18 19 20 21 2 3 4 5 6 7
	<pre>IF(AMIN.NE.AMAX) GO TO 10 AMIN = AMIN-1 AMAX = AMAX+1 IF(AMAX.LT.L.AND.AMIN.GT1)GU TO 110 MIN = AMIN MAX = AMAX IF(AMAX.GT.MAX) MAX=MAX+1 IF(AMIN.LT.MIN) MIN=MIN-1 IF(MIN.LT.0) NWID = MAX+IABS(MIN)</pre>	ບ ບ ບ ບ ບ ບ ບ	NIT NIT NIT NIT NIT NIT NIT NIT	8 9 10 11 12 13 14 15 16 17
40	IF (MIN_GE_0) NWID = MAX-MIN NUM = 10 IF (NWID_LT_NUM) GO TO 60 NUM = NUM*10 GO TO 40	u u	NIT NIT NIT NIT NIT	18 19 20 21 22
60	N = NWID/(NUM/10) IF(MIN_LT_0_AND_MAX_GT_0) GO TO 70 IF(N*(NUM/10)_LT_NWID) N=N+1 ANUM = NUM/10_ AUNIT = AXLEN/N GO TO 160		NIT NIT NIT NIT NIT NIT	23 24 25 26 27 28
7 <u>0</u>	N = IABS(MIN)/(NUM/10) IF(NN*(NUM/10)_LT_IABS(MIN)) NN = NN+1 N = MAX/(NIM/10) IF(N*(NUM/10)_LT_MAX) N = N+1 N = N+NN ANUM = NUM/10_ AUNIT = AXLEN/N GO TO 160		NIT NIT NIT NIT NIT NIT NIT	29 30 31 32 33 34 35 36
	NUM=10 IF(AMAX*NUM_GT_1) GO TO 130 NUM = NUM*10 GO TO 120		NIT NIT NIT	37 38 39 40
130 140	UNITT = 1./NUM N1 = AMIN*NUM N2 = AMAX*NUM IF(AMIN*NUM.LT.N1) N1=N1-1 IF(AMAX*NUM.GT.N2) N2=N2+1 IF(N1.NE.N2) GO TO 150 AMIN = AMIN-UNITT AMAX = AMAX-UNITT GU TO 140	יט יט יט יט יט יט יט יט	NIT NIT NIT NIT NIT NIT NIT	41 42 43 44 45 46 47 48 49
150	N = N2-N1 ANUM = UNITT IF(AMIN.LT.O.AND.AMAX.LT.O) N=N1-N2 IF(AMIN.LT.O.AND.AMAX.GE.O) N=N2-N1 AUNIT = AXLEN/N	U U U	NIT NIT NIT	50 51 52 53 54

160 170		UNIT UNIT UNIT	55 56 57 58 59 60 61
	RETURN	UNIT	61
	END	UNIT	62

# INITIAL DISTRIBUTION

## Copies

### Copies

Copies		Copies	
1	WES	2	NAVSHIPYD MARE
1	CHONR/438 Cooper		1 Library 1 Code 250
2	NRL	1	NAVSHIPYD BREM/Lib
	1 Code 2027 1 Code 2629	1	NAVSHIPYD PEARL/Code 202.32
1	ONR/Boston	8	NAVSHIP ID FLARE/Code 202.32
1	ONR/Chicago	0	1 SEC 6034B
1	ONR/Pasadena		1 SEC 6110
1	NORDA		1 SEC 6114H
4	USNA		1 SEC 6120 1 SEC 6136
4	1 Tech Lib		1 SEC 6140B
	1 Nav Sys Eng Dept		1 SEC 6144G
	1 B. Johnson		1 SEC 6148
	1 Bhattacheryya	1	NAVSEC, NORVA/6660.03 Blount
3	NAVPGSCOL 1 Library	12	DDC
	1 T. Sarpkaya	1	AFOSR/NAM
	1 J. Miller	1	AFFOL/FYS, J. Olsen
1	NADC	1	NSF/Eng Lib
3	NOSC	1	LC/Sci and Tech
	1 Library	1	DOT/Lib TAD-491.1
	1 Fabula 1 Hoyt	1	MMA, Library
1	NCSL/712 D. Humphreys	1	U. of BRIDGEPORT/E. Uram
1	NCEL/Code L31	4	U. of CAL/Dept Naval Arch, Berkeley
1	NSWC, Dahlgren		1 Library 1 Webster
1	NUSC/Lib		1 Paulling
7	NOSC/EID		1 Wehausen
'	1 SEA 0322	2	U. of CAL, San Diego
	1 SEA 033		1 A.T. Ellis
	1 SEA 03512/Pierce	-	1 Scripps Inst Lib
	1 SEA 037 3 SEA 09G32	3	CIT 1 Aero Lib
1	NAVFAC/Code 032C		1 T.Y. Wu
1	NAVSHIPYD PTSMH/Lib		1 A. Acosta
1	NAVSHIPYD PHILA/Lib	1	CATHOLIC U. of AMER/CIVIL
1	NAVSHIPYD NORVA/Lib		MECH ENG
1	NAVSHIPYD CHASN/Lib	1	COLORADO STATE U./ENG RES CEN
1	NAVSHIPYD LBEACH/Lib	1	U. of CONNECTICUT/Scottron
1	NAVORINT D LBEACH/LID	1	CORNELL U./Sears

Copies		Copies	
2	FLORIDA ATLANTIC U. 1 Tech Lib 1 S. Dunne	2	SOUTHWEST RES INST 1 Applied Mech Rev 1 Abramson
2	HARVARD U. 1 G. Carrier 1 Gordon McKay Lib	2	STANFORD U. 1 Eng Lib 1 R. Street
1	U. of HAWAII/Bretschneider	1	STANFORD RES INST/Lib
1	U. of ILLINOIS/J. Robertson	1	U. of WASHINGTON/ARL Tech Lib
3	U. of IOWA 1 Library 1 Landweber 1 Kennedy	3	WEBB INST 1 Library 1 Lewis 1 Ward
1	JOHN HOPKINS U./Phillips	1	WOODS HOLE/Ocean Eng
1	KANSAS STATE U./Nesmith	1	WORCHESTER PI/Tech Lib
1	U. of KANSAS/Civil Eng Lib	1	SNAME/Tech Lib
1	LEHIGH U./Fritz Eng Lab Lib	1	BETHLEHEM STEEL/Sparrows Point
5	MIT	1	BETHLEHEM STEEL/New York/Lib
	1 Library	1	BOLT, BERANEK and NEWMAN/Lib
	1 Leehey 1 Mandel	1	GENERAL DYNAMICS, EB/Boatwright
	1 Abkowitz	1	GIBBS and COX/Tech Info
4	1 Newman U. of MIN/ST. ANTHONY FALLS 1 Silberman 1 Schiebe 1 Wetzel 1 Song	5	HYDRONAUTICS 1 Library 1 E. Miller 1 A. Goodman 1 V. Johnson 1 C.C. Hsu
3	U. of MICH/NAME	1	LOCKHEED, Sunnyvale/Waid
	1 Library 1 Ogilivie 1 Hammitt	2	McDONNELL DOUGLAS, Long Beach 1 J. Hess 1 T. Cebeci
2	U. of NOTRE DAME	1	NEWPORT NEWS SHIPBUILDING/Lib
	1 Eng Lib 1 Strandhagen	1	NIELSEN ENG and RES
1	PENN STATE/ARL/B, Parkin	1	OCEANICS
1	PRINCETON U./Mellor	1	ROCKWELL INTERNATIONAL/B. Ujihara
5	SIT 1 Library	1	SPERRY RAND/Tech Lib
	1 Breslin	1	SUN SHIPBUILDING/Chief Naval Arch
	1 Savitsky	1	ROBERT TAGGART
	1 P.W. Brown 1 Fridsma	1	TRACOR
1	U. of TEXAS/ARL Lib		

- 1 U. of TEXAS/ARL Lib
- UTAH STATE U./Jeppson 1

## CENTER DISTRIBUTION

Copies	Code	Name
1	1500	W.E. Cummins
1	1504	V.J. Monacella
1	1506	M.K. Ochi
1	1507	D. Cieslowski
1	1512	J.B. Hadler
1	1520	R. Wermter
1	1521	P. Pien
1 1	1524 1524	Y.T. Shen W.C. Lin
1 1	1532 1532	G. Dobay R. Roddy
1	1540	W.B. Morgan
1 1	1552 1552	J. McCarthy N. Salvesen
1 1	1560 1560	G. Hagen N. Hubble
10	1562	M. Martin
1	1564	J. Feldman
1	1568	G. Cox
1 1	1572 1572	M.D. Ochi C.M. Lee
1	1576	W.E. Smith
10	5214.1	Reports Distribution
1	522.1	Unclassified Library (C)
1	522.2	Unclassified Library (A)

.

Dept. of APR 3 197

DEMCO

#### **DTNSRDC ISSUES THREE TYPES OF REPORTS**

1. DTNSRDC REPORTS, A FORMAL SERIES, CONTAIN INFORMATION OF PERMANENT TECH-NICAL 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 PRELIM-INARY, 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 IN-TERNAL 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.

