

DTNSRDC/SPD-0867-01

LEVEL



**DAVID W. TAYLOR NAVAL SHIP
RESEARCH AND DEVELOPMENT CENTER**

Bethesda, Md. 20084



ADA 078605

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

by

ERNEST E. ZARNICK

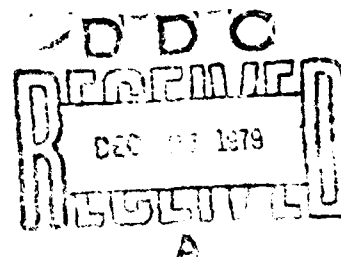
THIS DOCUMENT IS BEST QUALITY PRACTICABLE.
THE COPY FURNISHED TO DDC CONTAINED A
SIGNIFICANT NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

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

DDC FILE COPY

SHIP PERFORMANCE DEPARTMENT

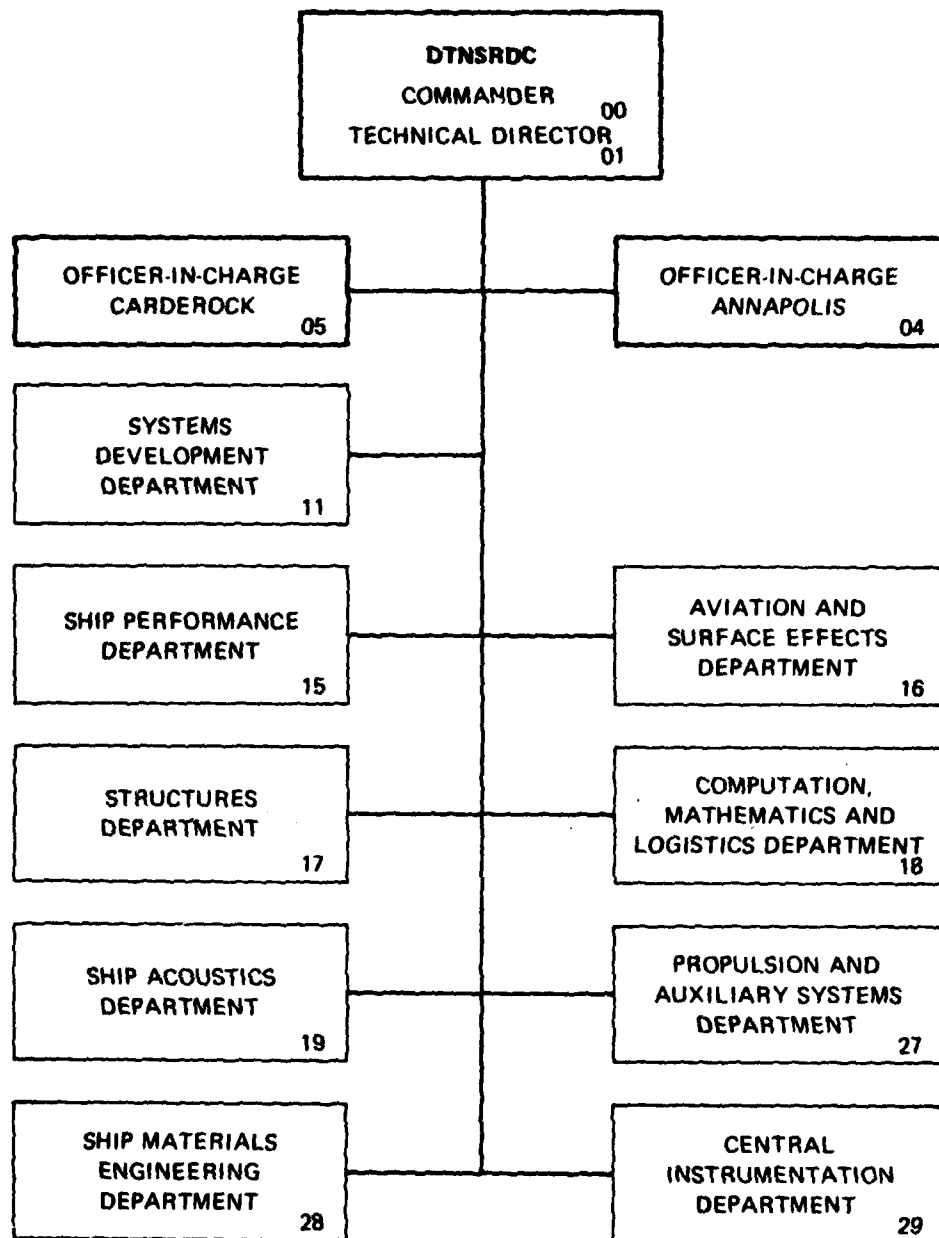


SEPTEMBER 1979

DTNSRDC/SPD-0867-01

79 12 19 256

MAJOR DTNSRDC ORGANIZATIONAL COMPONENTS



DISCLAIMER NOTICE

**THIS DOCUMENT IS BEST QUALITY
PRACTICABLE. THE COPY FURNISHED
TO DDC CONTAINED A SIGNIFICANT
NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.**

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER DTNSRDC/SPD-0867-01	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A NONLINEAR MATHEMATICAL MODEL OF MOTIONS OF A PLANING BOAT IN IRREGULAR WAVES.		5. TYPE OF REPORT & PERIOD COVERED
7. AUTHOR(s) Ernest E. Zarnick		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS David W. Taylor Naval Ship R&D Center Bethesda, Maryland		8. CONTRACT OR GRANT NUMBER(s) 1261
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Material Command (08T2) Washington, DC 20362		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Program Element 625 43 N Task Area ZF43-421-001 Work Unit No. 1500-102
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) F43401		12. REPORT DATE Sept 1979
16. DISTRIBUTION STATEMENT (of this Report) APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED		13. NUMBER OF PAGES 55
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
18. SUPPLEMENTARY NOTES		16. DECLASSIFICATION/DOWNGRADING SCHEDULE
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Planing Boat, Hydrodynamic Impact, Small Boat Seaworthiness, Nonlinear Ship Motions in Waves		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A computer program previously developed to estimate the motions and accelerations of a planing craft in regular waves was modified and extended to compute the corresponding motions in random or irregular waves. Ten regular waves with random phase were combined to represent the random seaway. The amplitudes and frequencies that were selected represent the energy distribution of a Pierson-Moskowitz spectrum for a fully developed sea. A comparison of computed results with experiment indicates that the computer program can predict craft behavior with reasonable quantitative		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-314-6601

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

389 694

50

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

✓ accuracy in moderate operating conditions. In severe operating conditions, however, the amplitudes of the computed vertical accelerations, which include impacts, are one-half of the experimental values. 

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

TABLE OF CONTENTS

TABLE OF CONTENTS	iii
LIST OF FIGURES	iv
NOTATION.	v
ABSTRACT.	1
ADMINISTRATIVE INFORMATION.	1
INTRODUCTION.	2
MATHEMATICAL FORMULATION.	2
EQUATION OF MOTION.	2
REPRESENTATION OF SEAWAY.	8
COMPARISON OF COMPUTER RESULTS WITH EXPERIMENTS	9
SUMMARY AND CONCLUSIONS	14
ACKNOWLEDGMENT.	16
APPENDIX - LISTING OF COMPUTER PROGRAM FOR MOTION COMPUTATIONS. .	31

Accession For	
NTIS GRA&I	☒
DDC TAB	☐
Unannounced	☐
Justification	
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or special
A	23

LIST OF FIGURES

1	- Coordinate System	22
2	- Lines of Prismatic Models	23
3	- Sample of Computed Pitch and Heave.	24
4	- Sample of Computed Bow and CG Acceleration.	25
5	- Example of Pitch Motion Correlation with Generalized Rayleigh Distribution	26
6	- Example of Heave Motion Correlation with Generalized Rayleigh Distribution	27
7	- Example of Bow and CG Acceleration Correlation with Exponential Distribution.	27
8	- Comparison of Computed and Experimental Pitch Variation with Significant Wave Height.	28
9	- Comparison of Computed and Experimental Heave Variation with Significant Wave Height.	28
10	- Comparison of Computed and Experimental Bow and CG Acceleration Variation with Significant Wave Height	29
11	- Comparison of Computed and Experimental Pitch Variation with Dead Rise Angle.	29
12	- Comparison of Computed and Experimental Heave Variation with Dead Rise Angle.	30
13	- Comparison of Computed and Experimental Bow and CG Acceleration Variation with Dead Rise Angle	30

LIST OF TABLES

1	- Wave Amplitude Representing Discrete Spectrum	19
2	- Model Characteristics and Wave Conditions for Computations.	20
3	- Comparison of Computed Results with Experiments	21

NOTATION

A_R	Section Area
a	Correction factor for buoyancy force
b	Half-beam of craft
c_i	Wave celerity of i th component
$C_{D,c}$	Crossflow drag coefficient
CG	Center of Gravity of boat
C_Δ	Load coefficient $\Delta/pg(2b)^3$
C_λ	Wavelength coefficient $L/\lambda C_\Delta/(L/2b)^2$ ^{1/3}
D	Friction drag force
F_x	Total hydrodynamic force in x direction
F_θ	Total hydrodynamic moment about pitch axis
f	Two-dimensional hydrodynamic force
g, G	Acceleration of gravity
H	Wave height, crest to trough
h	Vertical submergence of point below free surface
h_{50}	Heave crest or trough value corresponding to 50% probability point
h_{90}	Heave crest or trough value correspond to 90% probability point
I	Pitch moment of inertia
I_a	Added pitch, moment of inertia
k_a	Two-dimensioal added-mass coefficient
k_i	Wave number of i th component
L	Hull length
LCG	Longitudinal center of gravity percent of L
M	Mass of craft

NOTATION (CONT)

M_a	Added mass of craft
m_a	Sectional (two-dimensional) added mass
N	Hydrodynamic force normal to baseline
r^-	Ratio of negative maximum to total maximum
r	Wave elevation, positive down, feet
r_i	Wave amplitude of i th component
U	Relative fluid velocity parallel to baseline
V	Relative fluid velocity normal to baseline
$V/\sqrt{L}$	Speed-to-length ratio in knots/ft ^{1/2}
W	Weight of craft
w_z	Vertical component of wave orbital velocity, positive down
x	Fixed longitudinal axis; also the coordinate of a point relative to the origin of body axis.
$\dot{x}_{CG}$	Surge velocity
$\ddot{x}_{CG}$	Surge acceleration
x_{CG}	Longitudinal distance from origin of fixed axis to CG of the body
z	Fixed vertical normal axis; also the coordinate of a point relative to the origin of body axis, positive down
$\ddot{z}_{CG}$	Heave acceleration of the CG
z_{CG}	Heave displacement of the CG, positive down
β	Deadrise angle
Δ	Hull displacement W
ζ	Normal body axis; also the coordinate of a point relative to the origin of body axis
η	Vertical acceleration (i.e., in direction of ζ axis)

NOTATION (CONT)

θ	Pitch angle
$\dot{\theta}$	Pitch angular velocity
$\ddot{\theta}$	Pitch angular acceleration
θ_{50}	Pitch crest or trough corresponding to 50% probability point, degrees
θ_{90}	Pitch crest or trough corresponding to 90% probability point, degrees
ξ	Longitudinal body axis; also the coordinate of a point relative to the origin of body axis
ρ	Density of water
σ_i	Phase angle of i th component
ω	Wave frequency
ω_p	Peak frequency of wave spectrum
L	Wetted length
Ω	Nondimensional frequency, ω/ω_p
ϵ	Spectral width parameter

ABSTRACT

A computer program previously developed to estimate the motions and accelerations of a planing craft in regular waves was modified and extended to compute the corresponding motions in random or irregular waves. Ten regular waves with random phase were combined to represent the random seaway. The amplitudes and frequencies that were selected represent the energy distribution of a Pierson-Moskowitz spectrum for a fully developed sea. A comparison of computed results with experiments indicate that the computer program can predict craft behavior with reasonable quantitative accuracy in moderate operating conditions. In severe operating conditions, however, the amplitudes of the computed vertical accelerations, which include impacts, are one-half of the experimental value.

ADMINISTRATIVE INFORMATION

This work has been authorized by the Naval Material Command (08T2); under Program Element 625 43 N, Task Area ZF43-421-001, administered by the Ship Performance Department, High Performance Vehicle Program Office, Code 1512.

INTRODUCTION

In a previous study¹ a computer program was developed to estimate the motions and accelerations of a planing craft in regular waves. As a logical extension of this work, the program was modified to compute the motion of the craft in random or irregular waves.

Since the mathematical model is nonlinear, the computations are made in the time domain. Ten regular waves are combined with random phase to represent the random seaway. The amplitudes and frequencies are adjusted to conform to the energy distribution in a Pierson-Moskowitz fully developed sea.

The mathematical model was developed for a V-shaped prismatic-body with hard chines and constant deadrise planing at constant speed. The thrust and the friction drag forces are assumed to act through the center of gravity. The vertical components of the thrust and friction drag are also assumed to be negligible in comparison to the hydrodynamic forces.

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.

MATHEMATICAL FORMULATION

EQUATIONS OF MOTIONS

The equations of motion for a planing craft restricted to pitch θ , heave z_{CG} , and surge x_{CG} can be written as

$$\begin{aligned}\ddot{Mx}_{CG} &= T_x - N \sin \theta - D \cos \theta \\ \ddot{Mz}_{CG} &= T_z - N \cos \theta + D \sin \theta + W \\ I\ddot{\theta} &= Nx_c - Dx_d + Tx_p\end{aligned}$$

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
- x_c is distance from center of gravity (CG) to center of pressure for normal force, positive forward
- x_d is distance from CG to line of action for friction drag force
- x_p is moment arm of thrust about CG.

Motions are measured relative to a fixed coordinate system with the x axis located in the undisturbed free surface pointing in the direction of travel and the z axis pointing downward.

Since the perturbation velocities in the forward direction are small in comparison to the speed of the craft, the equations of motion can be simplified by neglecting them and by setting the forward velocity equal to a constant, i.e.,

$$\dot{x}_{CG} = \text{CONSTANT}$$

Furthermore, if it is assumed that the vertical components of the thrust and friction drag forces are small in comparison to the hydrodynamic forces

and that the total thrust and friction drag forces are acting through the center of gravity (so as to produce no moments) the equations of motion can be written as

$$\begin{aligned}\ddot{x}_{CG} &= 0 \\ \ddot{Mz}_{CG} &= -N\cos\theta + W \\ \ddot{I\theta} &= Nx_C\end{aligned}$$

A so called "strip theory" is used to obtain the hydrodynamic force acting on the body by integrating the 2-D hydrodynamic forces normal to the baseline over the wetted length of the body. A body coordinate system (ξ, ζ) with its origin at the CG and the ξ axis pointing forward parallel to the baseline of the body as shown in Figure 1 is used to facilitate this integration.

The normal hydrodynamic force per unit length f , acting at a section, is assumed to be proportional to the rate of change of momentum associated with an added mass term and the cross flow drag, i.e.

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

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

m_a is the added mass associate with the section form

$C_{D,c}$ is the crossflow drag coefficient

ρ is the density of the fluid

b is the half beam

Expanding the momentum term results is

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

where ξ is the body coordinate parallel to the baseline; see Figure 1.

The last term on the right-hand side of the above equation 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.

For a V-shaped wedge the 2-D added mass is defined as

$$m_a = k_a \frac{\pi}{2\rho} b^2 (\xi, \zeta, t)$$

where k_a is the added mass coefficient (assumed to be 1 in this study) and b is the wetted half beam. Once the chine becomes wetted the beam is assumed constant regardless of depth of penetration.

Cross-flow drag for a V section with separation at the chine is assumed equal to the drag of a flat plate ($C_{D,c} = 1.0$) corrected by the Bobyleff flow coefficient approximated by $\cos \beta$, i.e.,

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

The Bobyleff flow coefficient is the theoretical ratio of the pressure on a V-section to that experienced by a flat plate for 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 here in to act in the vertical direction and to be equal to the static buoyancy force multiplied by a correction factor,

i.e.,

$$f_B = a \rho g A_R$$

where A_R 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. Shuford² 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 moments, 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.

Integrating the 2-D hydrodynamic force over the wetted length of the craft (ℓ) and taking the component in the z direction results in

$$\begin{aligned} -N \cos \theta &= F_z(t) = \int_{\ell} f \cos \theta d\xi + \int_{\ell} f_B d\xi \\ &= - \left[\int_{\ell} \left\{ m_a(\xi, t) \dot{V}(\xi, t) + \dot{m}_a(\xi, t) V(\xi, t) \right. \right. \\ &\quad \left. \left. - U(\xi, t) \frac{\partial}{\partial \xi} [m_a(\xi, t) V(\xi, t)] \right. \right. \\ &\quad \left. \left. + C_{D,c}(\xi, t) \rho b(\xi, t) V^2(\xi, t) \right\} \cos \theta d\xi \right. \\ &\quad \left. + a \rho g A_R d\xi \right] \end{aligned}$$

Similarly, the hydrodynamic moment about the CG is obtained by integrating the product of the normal force per unit length by the corresponding moment arm.

$$\begin{aligned}
F_\theta &= - \int_L f(\xi, t) \xi d\xi - \int_L f_b \cos \theta \xi d\xi \\
&= \int_L \left\{ m_a(\xi, t) \dot{V}(\xi, t) + \dot{m}_a(\xi, t) V(\xi, t) \right. \\
&\quad - U(\xi, t) \frac{\partial}{\partial \xi} (m_a(\xi, t) V(\xi, t)) + C_{D,c}(\xi, t) \rho b(\xi, t) V^2(\xi, t) \\
&\quad \left. + \frac{\rho g A_R}{2} \cos \theta \right\} \xi d\xi
\end{aligned}$$

Wave excitation enters into the above equations through 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 . Diffraction has been neglected.

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

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

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

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

where r is the instantaneous value of the wave elevation directly above the point measured positive downward.

A more detailed derivation of the above integrals for the hydrodynamic force and moment is presented in reference (1). Although the hydrodynamic

forces and moment require integration over the wetted length, which may vary with time, the resulting equations of motion can be integrated in the time domain using numerical method such as the Runge-Kutta Merson integration routine used in these studies.

REPRESENTATION OF THE SEAWAY

The seaway in general can be represented by an infinite sum of sine waves with random phase. In these studies, for the sake of computational economy, the seaway is represented by the discrete sum of ten harmonic waves with random phase

$$r = \sum_{i=1}^{10} r_i \cos[k_i (x + c_i t) + \sigma_i]$$

where r_i is the wave amplitude

k_i is the wave number

c_i is the wave celerity

and σ_i is the random phase angle of the i th wave component.

Note that at point $P(\xi, \zeta)$ on the craft

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

and $x_{CG} = \int \dot{x}_{CG} dt$

Each frequency and wave amplitude is weighted in accordance with the energy distribution in a Pierson-Moskowitz spectrum for a fully developed sea. The Pierson-Moskowitz formulation for a continuous spectrum can be written as

$$S(\omega) = \frac{Ag^2}{\omega^5} e^{-B/\omega^4}$$

where $A = 8.1 \times 10^{-3}$

g = gravitational constant

$$B = \frac{4Ag^2}{(H/3)^2}$$

The constant B is also related to the peak frequency (ω_p) of the spectrum

by

$$B = \frac{5}{4} \omega_p^4$$

which can be confirmed by differentiating the spectrum formulation and setting it equal to zero. Normalizing the frequencies by the peak frequency leads to a nondimensional spectrum $\bar{S}$ which is related to the dimensional spectrum by

$$\bar{S}(\Omega) = \frac{16\omega_p}{H_{1/3}^2} S(\omega = \Omega\omega_p) = \frac{5}{\Omega^5} e^{-5/4\Omega^4}$$

where $\Omega = \omega/\omega_p$

and where

$$\int_0^\infty \bar{S}(\Omega) d\Omega = 1$$

The discrete frequencies representing the spectrum varied from $\Omega = .80$ to 2.6 in nearly equal increments $\Delta\Omega=0.2$. A slight random perturbation is given to each frequency to avoid precise integer multiple frequencies, thereby increasing the fundamental repetition period of the computed time history. Each discrete amplitude is adjusted so that its energy corresponds to that contained in a band width ($\Delta\Omega$) centered about its frequency in the continuous spectrum i.e.,

$$r_i^2 = \frac{H_{1/3}^2}{8} \cdot \int_{\Omega_i - \Delta\Omega/2}^{\Omega_i + \Delta\Omega/2} \bar{S}(\Omega) d\Omega$$

The band widths are equally spaced between frequencies except for the first and last frequencies which lump all of the remaining energy at the beginning and end of the spectrum. Table 1 presents a list of the amplitude for each nondimensional frequency in terms of the significant wave height.

COMPARISON OF COMPUTED RESULTS WITH EXPERIMENTS

Computations of pitch and heave motions and bow and CG vertical

accelerations were made using the computer program (see Appendix) for comparison with the model experiments of Fridsma³. Fridsma tested a series of constant-deadrise models of various lengths in irregular waves to determine the effects of deadrise, trim, load, speed and sea state on the added resistance, heave and pitch motions and vertical accelerations at the bow and CG. Figure 2 shows the lines of the prismatic models. The computations were made with the Centers' Control Data Corporation 6700 computer system. A listing of the computer program is presented in the Appendix.

Table 2 presents the characteristics of the model craft for those conditions selected for the comparison. The number of computer runs was kept to a minimum for economic reasons. Approximately one minute of central processor time was required for every second of data using model scale dimensions, and approximately 40 seconds of model scale data was required to obtain 100 cycles of amplitude data.

The output of the program is the time histories of the pitch and heave motions and the bow and CG accelerations. Sample plots of the outputs are shown in Figures 3 and 4. Procedures required for processing and analysing the data are not a part of this study. In order to facilitate the comparison, the analysis procedures followed in this report are those used by Fridsma³ for his experimental data. The amplitudes (maxima or minima) of the pitch and heave motions about the mean are assumed to be described by the so called "Generalized Rayleigh Distribution." i.e.,

$$p(y) = \frac{1}{\sqrt{2\pi}} \epsilon e^{-1/2y^2/\epsilon^2} + (1 - \epsilon^2)^{1/2} y e^{-1/2y^2} \int_{-\infty}^{y(1-\epsilon^2)\epsilon} e^{-1/2x^2} dx$$

where y = maximum or minimum (absolute values) normalized by the standard deviation,

$$\epsilon^2 = 1 - (1 - 2r')^2$$

and r' = ratio of negative maxima to total maxima or positive minima to total minima.

To fit the data to this distribution the crests (maxima) and troughs (minima) relative to the mean value are first determined from time histories of the motions. The mean value is defined as halfway between the average crest and average trough value. The crests or trough data (X_i) are sorted in ascending order and grouped into fifteen intervals. At the same time, the proportion r' of negative maxima to total maxima or positive minima to total minima is determined. The cumulative frequency and corresponding probability that a crest or trough is less than or equal to the interval value (X_i) is then computed. From the probability and r' values, the theoretical value of the normalized amplitude (y) is calculated. A plot of X_i versus corresponding y values is compared with a line drawn through $x = y = 0$, and the point, $x = \bar{x}$, $y = \bar{y} = \sqrt{\pi/2}(1-2r')$ which is indicative of the fit of the theoretical distribution function to the data. The values x and y are the observed average value of the first moment and the theoretical average value (normalized) respectively.

Figures 5 and 6 show typical examples of such plots for the pitch and heave crests and troughs. As can be seen in these figures, the data fit the assumed probability function reasonably well, but it is also quite possible that some other distribution might fit the data better.

The acceleration data were assumed to follow a simple exponential distribution. For this distribution, the probability, P of the acceleration peak n being less than a given value is

$$P(n) = 1 - e^{-n/\bar{n}}$$

where $\bar{a}$ = average peak acceleration.

Only the negative peak accelerations (impact spikes as well as wave induced) were analyzed. The data were sorted and grouped into fifteen intervals similar to the motion data analysis and the probability was plotted with respect to a on inverted semilog paper. For a good fit, the data should follow a straight line through the point ($P = 0.368$, $a = \bar{a}$) and the origin. Figure 7 shows a sample of the acceleration data plotted in the above manner. An exponential probability function appears to be a good fit to the data.

Table 3 presents a comparison of the computed motions and accelerations with the corresponding experimental results. The computations were made for a craft with a length to beam ratio (L/b) of 5, a deadrise angle (β) of 20 degrees and a speed length ratio ($V/\sqrt{L}$) of 6 for several sea states with significant wave height to beam ratios ($H_{1/3}/2b$) of .222, .444 and .666 which would correspond to Sea states 2, 3, and 4 respectively for a 50 foot craft. Computations were also made in sea condition with $H_{1/3}/2b$ of .444 for a speed length ratio of 4 at 10, 20 and 30 degrees deadrise angles.

The tabulated values for the heave and pitch are those with a 50 percent and those with a 90 percent probability of not being exceeded. Heave is nondimensionalized by the beam. The values for the bow and CG accelerations are the average values of the negative peaks.

Other statistical variables such as the 1/3 or 1/10 highest values can be computed from the specific probability distribution. For the assumed distribution of the motion amplitudes, the 1/10 highest value is related to the 90% probability value by the ratio of $y_{1/10}/y_{90}$, which is

approximately 1.33 over the r' value range measured. The 1/10 highest accelerations from the exponential distribution is $3.30 \bar{n}$.

Plots of the data in Table 3 are also presented in Figure 8 through 13. Figures 8 through 10 show respectively a comparison between the computed and experimental results of the variation in the 90 percent probability values for the pitch and heave motions, and the average values for the bow and CG accelerations, with significant wave height to beam ratio. Figures 11 through 13 show similar plots for the variation with deadrise angle.

The pitch data in the figures show that while the computed troughs (bow down) are in reasonable agreement with the experiment, the computer crests are lower than the experiment. The heave exhibits the same trend with the computed crests (CG down) being in reasonable agreement with the experiment and troughs being lower than the experiment. Furthermore, the 90 percent probability values for the pitch and heave crests and troughs for the computer model are about equal in magnitude; whereas, the experimental model values in the pitch bow up and heave CG up direction are greater. It appears that the experiment model exhibits more nonlinearity than the computer model.

The computed acceleration data for the bow and CG are generally lower than the corresponding experiment data. Figures 10 and 13 show that the computed accelerations differ by 15 percent to 50 percent of the corresponding experiment values with the largest differences occurring in the more severe conditions where the accelerations are extremely high.

For example, at the ten degree deadrise angle condition presented in Table 2, the average 1/10 highest bow acceleration corresponds to about

24 g's for the experiment and 11 g's for the computed value. It is doubtful that any operational boat would be driven to such extreme conditions.

The characteristics of the experiment model motions appear to be slightly different from those of the computer model in very extreme conditions. The experiment model experienced larger pitch bow up and heave motion than the computer model, which probably resulted in larger impact accelerations. This does not completely explain the differences between the experiment and computed vertical accelerations which may also reflect a deficiency in the mathematical representation of the impact phenomenon, and perhaps the seaway as well.

In less severe conditions the agreement between computed and experiment results is better. Good results can be expected for speed length ratios up to approximately 6 in a seaway with significant wave height to beam ratio of 0.222 (State 2 sea for a 50 foot craft). For a significant wave height to beam ratio of 0.444 (State 3 for a 50 foot craft), the calculations could probably be used to predict reasonable quantitative results up to a speed length ratio of 4. In more extreme conditions the computed results are less accurate quantitatively, but are still indicative of gross trends.

SUMMARY AND CONCLUSIONS

A computer program was developed to compute the motions and accelerations of simple prismatic planing craft in head irregular waves. This was achieved by incorporating irregular waves into an existing program for computing the motions in regular waves. The irregular waves were synthesized by combining ten regular waves with random phase and with frequencies and amplitudes weighted to represent a Pierson-Moskowitz spectrum for fully developed seas.

Computations were made for a craft with a length to beam ratio of 5, a deadrise angle of 20 degrees and a speed length ratio of 6 for several sea states, and in a single sea state for a speed length ratio of 4 with 10, 20 and 30 degrees deadrise angles. The results were compared with the experiment results of Fridsma. First the probability distributions of the crests and troughs of the motions and accelerations were examined to determine whether or not they were the same as those obtained by the experiments. It was found that the fit of the pitch and heave crests and troughs to the "Generalized Rayleigh Distribution" which was used in the experiment data analysis, was acceptable for the computed data, but no attempt was made to fit the data to other types of distribution which might have fitted better. The computed acceleration data, fitted an exponential type of distribution.

A comparison of the motions showed that the computed pitch troughs were in good agreement with the experiment, but the crests (bow up) were lower than the experiment values. The heave exhibited the same trend, with the troughs (CG heave up) for the experiment being higher than the computed values while the crests were in good agreement.

Computed vertical accelerations at the bow and CG were, for certain conditions, much lower than the experiment values. At some conditions examined, the computed accelerations were about half of the comparable experiment values. This occurred at the more extreme operating condition where very large accelerations were experienced. For example, a value of 24 g's was obtained for the average of the 1/10 highest bow acceleration from the experiment for a craft with 10 degrees deadrise angle as compared to 11 g's from the computed results. These values were obtained in a seaway

with a significant wave height to beam ratio of .666 and a speed length ratio of 6 which represent an operating condition far more severe than that in which the boat would be expected to operate (42 knots in a State 4 sea for a 50 foot craft).

In summary it appears that the computer program can predict craft behavior quantitatively most effectively in moderate operating conditions. In severe operating conditions, probably beyond that in which the craft would be expected to operate, the computed vertical accelerations which include impacts are roughly one half the experiment values; however, despite the deficiency of the computer program in predicting quantitative results, the predictions are still indicative of gross trends.

Additional work is required to improve the prediction of the impact accelerations especially during the more severe conditions. Towards this end, it would be desirable to compare the time histories of experimental motion and acceleration data with the corresponding computed values. It may be possible to modify the hydrodynamic coefficients in the mathematical model on the basis of experimental results, specifically those affecting impact acceleration, and greatly improve the correlation. It is recommended that if additional model experiments are conducted, time histories of the motions and accelerations along with the wave height be made available for the above studies.

ACKNOWLEDGMENT

Acknowledgment is given to Ms. Dana Gentily of ORI, Inc., who prepared the computer programs for statistical analysis of the data.

REFERENCES

1. Zarnick, E.E., "A Nonlinear Mathematical Model of a Planing Boat in Regular Waves," DTNSRDC Report 78/032 (March 1978).
2. 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).
3. Fridsma, G., "A Systematic Study of the Rough Water Performance of Planing Boats (Irregular Waves - Part II)," Davidson Laboratory Stevens Institute of Technology Report R1495 (March 1971).

TABLE 1
WAVE AMPLITUDE REPRESENTING
DISCRETE SPECTRUM

Ω	$r/H_{1/3}$
0.795	0.1364
1.0	0.1861
1.183	0.1657
1.403	0.1302
1.602	0.0999
1.795	0.0771
2.004	0.0604
2.194	0.0482
2.392	0.0390
2.612	0.0626

TABLE 2

MODEL CHARACTERISTICS AND WAVE

CONDITIONS FOR COMPUTATIONS

Model Length = 114.3 cm (3.75 ft)
 Length/Beam = 5; $C_A = 0.600$

COMPUTER RUN	SYMBOL	β DEGREE	LCG PERCENT L	RADIUS OF GYRATION PERCENT L	$V/\sqrt{L}$	$H_{1/3}/2b$
1	M	20	64.0	24.8	6	0.222
2	M	20	64.0	24.8	6	0.444
3	M	20	64.0	24.8	6	0.666
4	O	20	66.8	25.0	4	0.666
5	C	10	68.0	25.0	6	0.444
6	G	30	62.1	25.0	6	0.444

TABLE 3 - COMPARISON OF COMPUTED RESULTS WITH EXPERIMENT

	PITCH (DEGREES)						HEAVE				ACCELERATION (Gs)	
	Crest			Trough			Crest		Trough		$\bar{n}_{bow}$	$\bar{n}_{CG}$
	r	θ_{90}	θ_{90}	r	θ_{90}	θ_{90}	r	$h_{90}/2b$	$h_{90}/2b$	r	$h_{90}/2b$	$h_{90}/2b$
Computed	0.169	0.69	1.67	0.108	0.82	1.75	0.122	0.027	0.060	0.158	0.025	0.059
Experiment	0.168	0.90	2.13	0.080	0.88	1.78	0.148	0.034	0.067	0.114	0.036	0.076
Computed	0.105	1.83	3.93	0.139	1.72	3.85	0.196	0.082	0.213	0.140	0.099	0.223
Experiment	0.182	2.25	5.51	0.064	2.20	4.34	0.033	0.140	0.265	0.222	0.145	0.393
Computed	0.129	2.47	5.53	0.151	2.38	5.45	0.083	0.212	0.414	0.167	0.165	0.392
Experiment	0.106	3.14	6.71	0.200	2.45	6.27	0.192	0.142	0.365	0.118	0.179	0.396
Computed	0.101	3.92	8.18	0.500	3.86	7.47	0.140	0.199	0.447	0.084	0.197	0.402
Experiment	0.126	1.60	3.66	0.097	1.75	3.63	0.146	0.090	0.209	0.089	0.106	0.261
Computed	0.143	1.76	3.98	0.134	1.76	3.91	0.044	0.118	0.226	0.121	0.120	0.261
Experiment	0.121	2.15	4.72	0.131	2.13	4.70	0.132	0.113	0.253	0.132	0.115	0.253
Computed	0.177	2.39	5.78	0.094	2.36	4.87	0.062	0.132	0.259	0.272	0.135	0.424
Experiment												

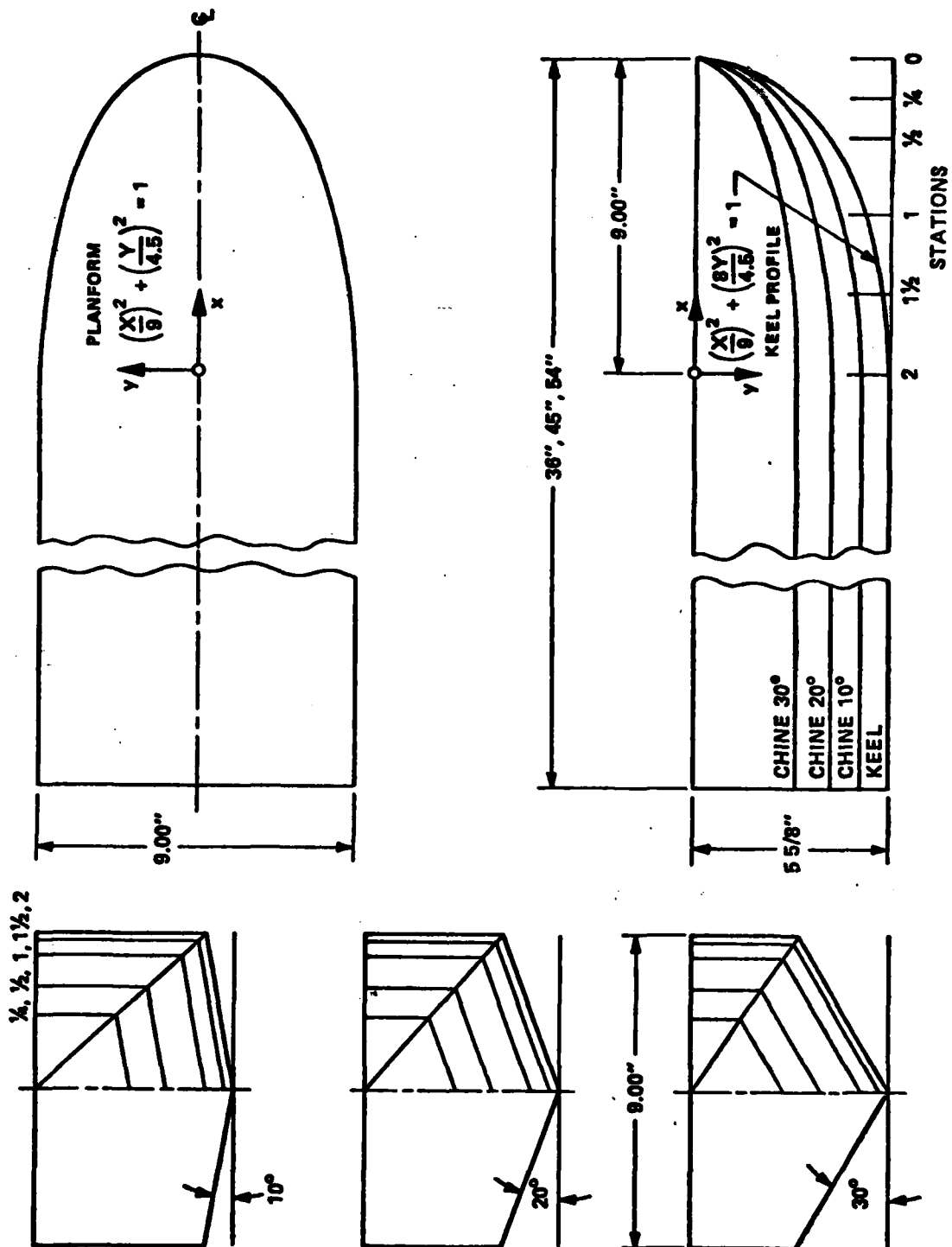


Figure 2 - Lines of Prismatic Models
(From Reference 3)

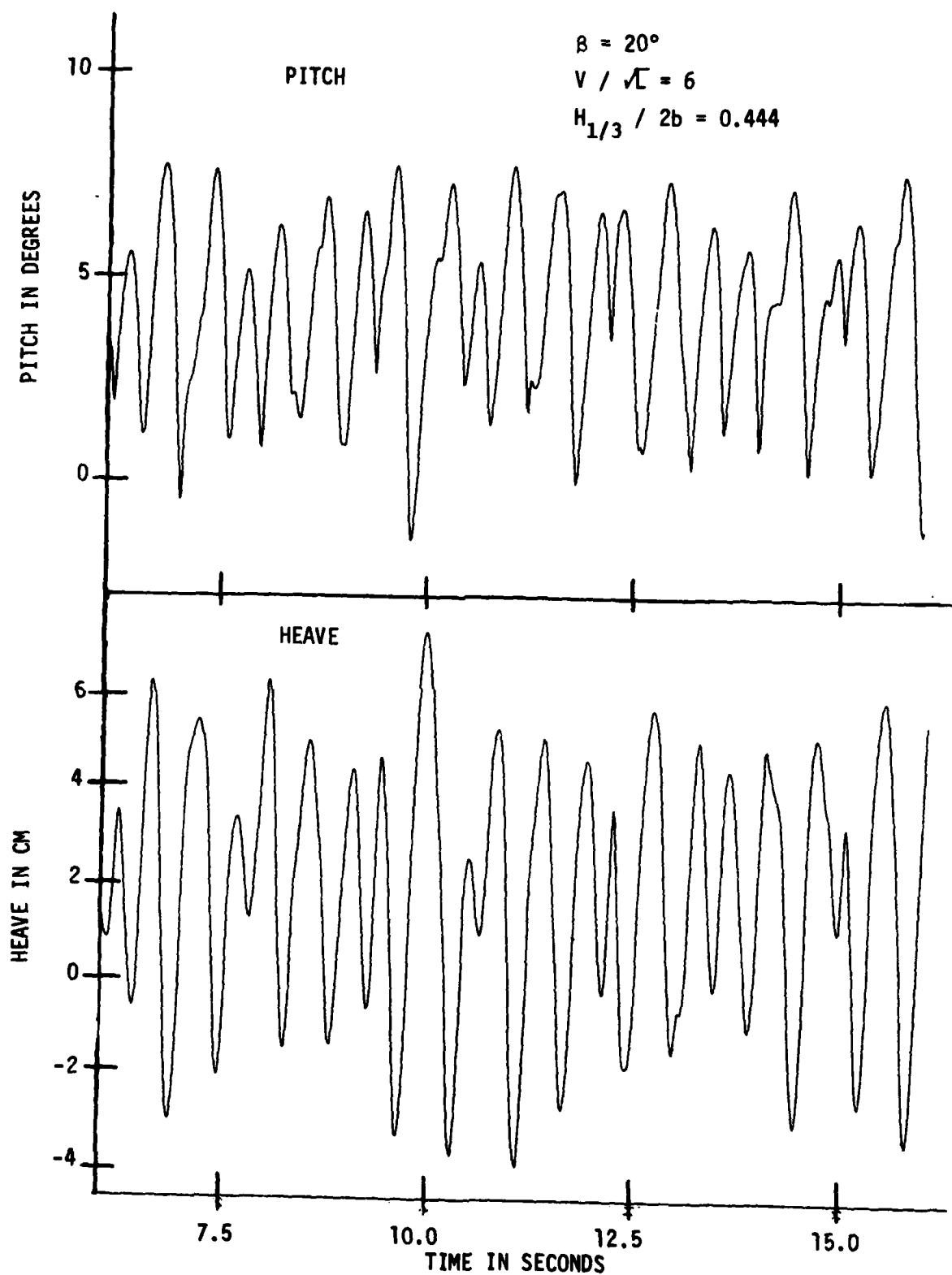


Figure 3 - Sample of Computed Pitch and Heave Motion

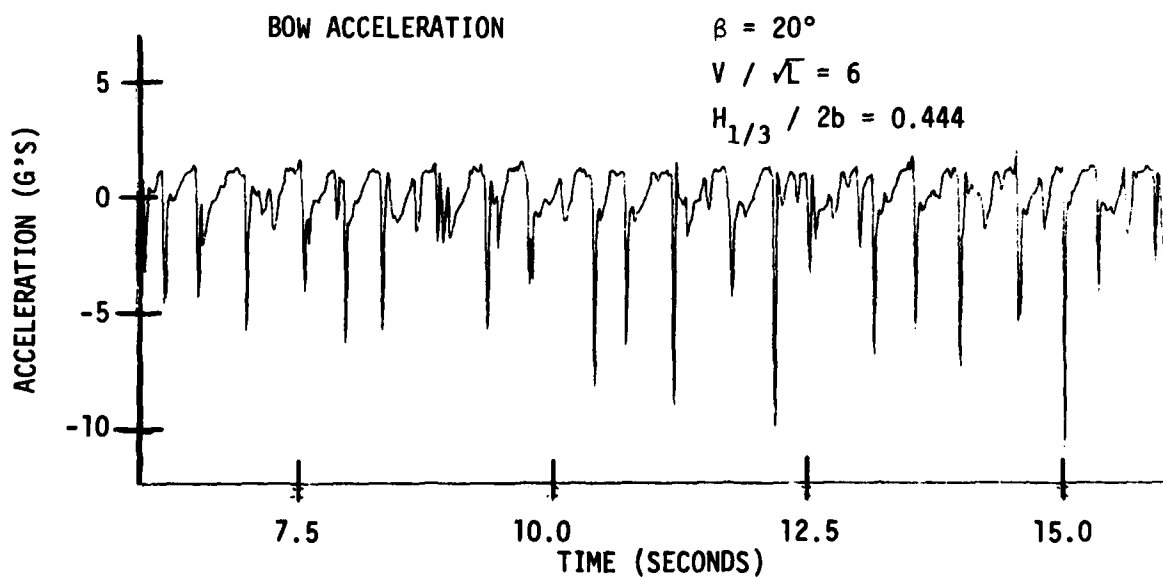
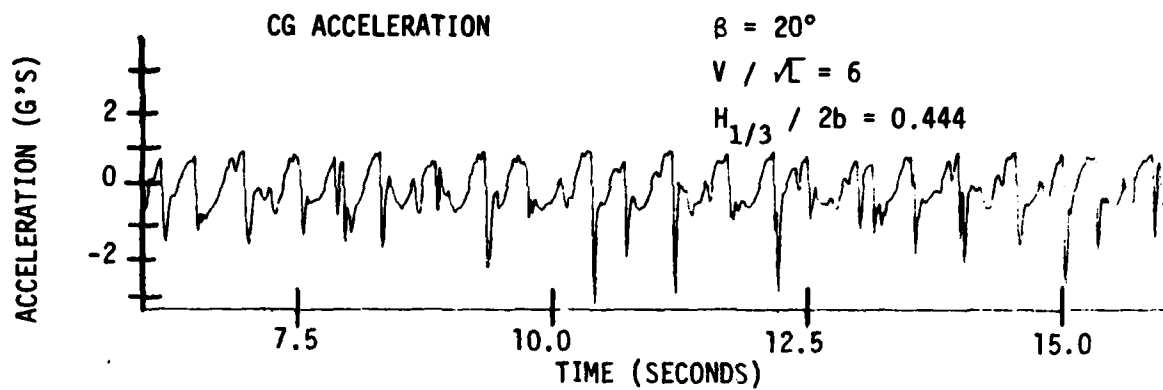


Figure 4 - Sample of Computed Bow and CG Acceleration

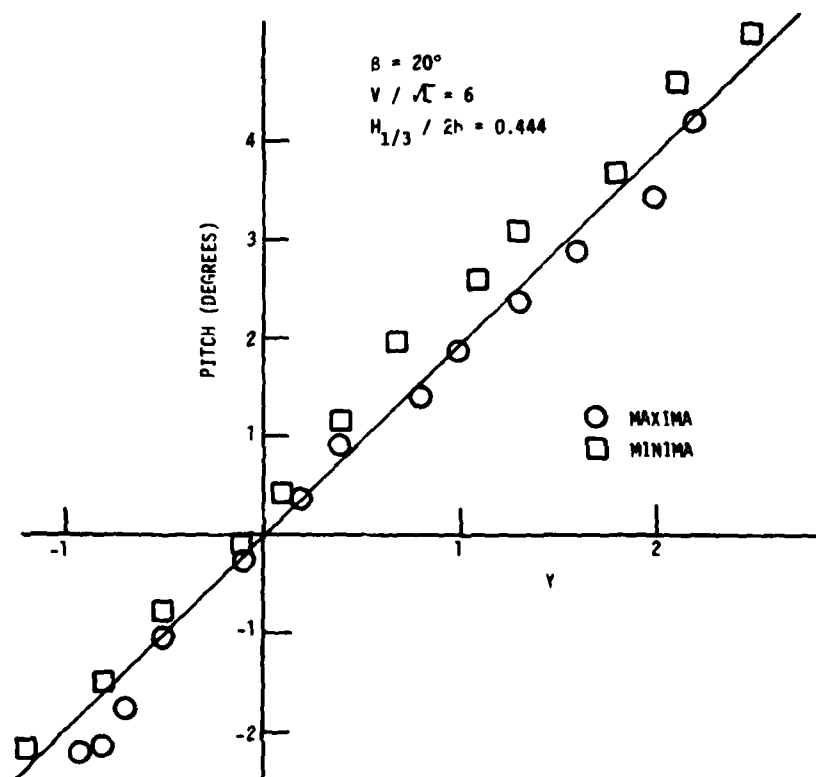


Figure 5 - Example of Pitch Motion Correlation
with Generalized Rayleigh Distribution

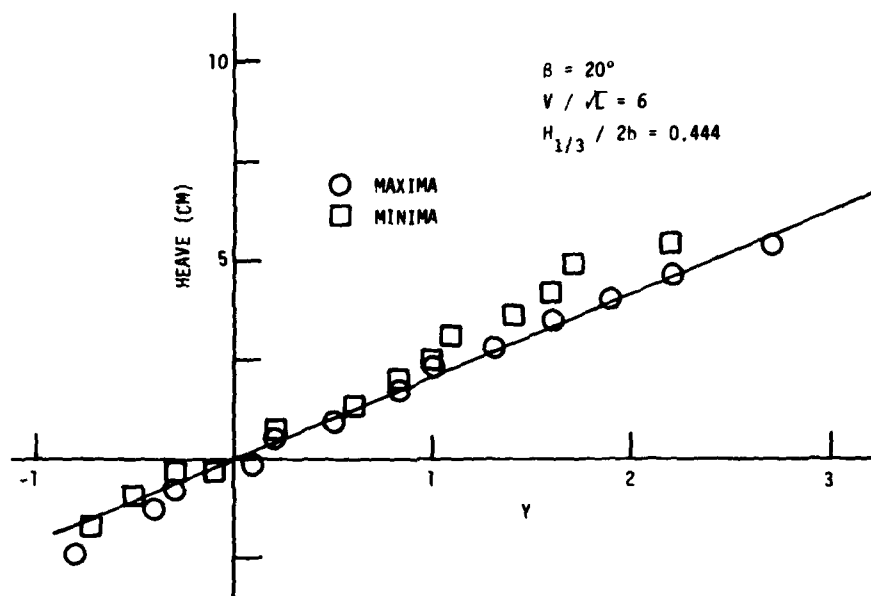


Figure 6 - Example of Heave Motion Correlation with Generalized Rayleigh Distribution

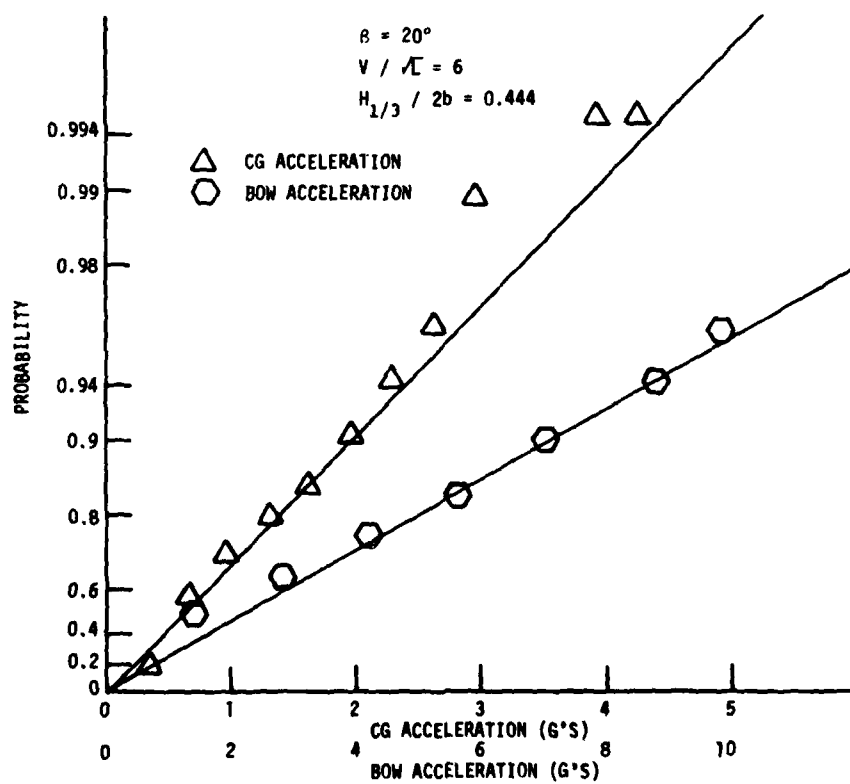


Figure 7 - Example of Bow and CG Acceleration Correlation with Exponential Distribution

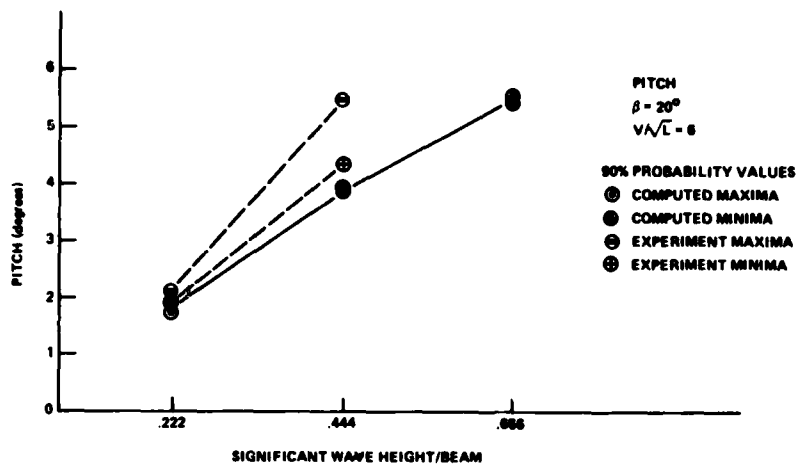


Figure 8 - Comparison of Computed and Experimental Pitch Variation with Significant Wave Height

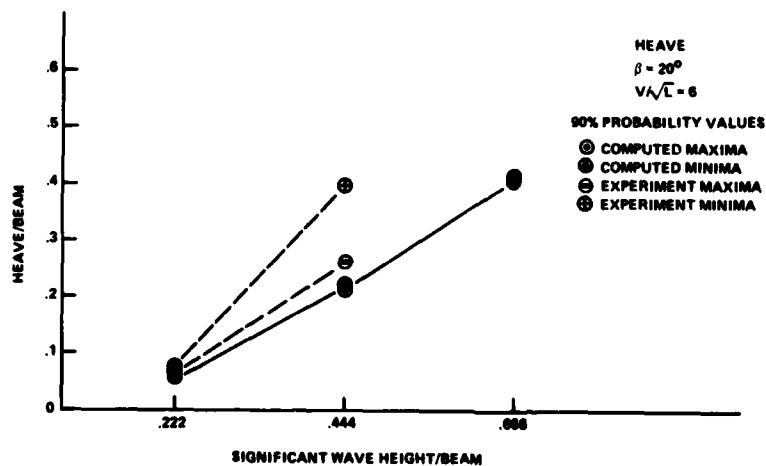


Figure 9 - Comparison of Computed and Experimental Heave Variation with Significant Wave Height

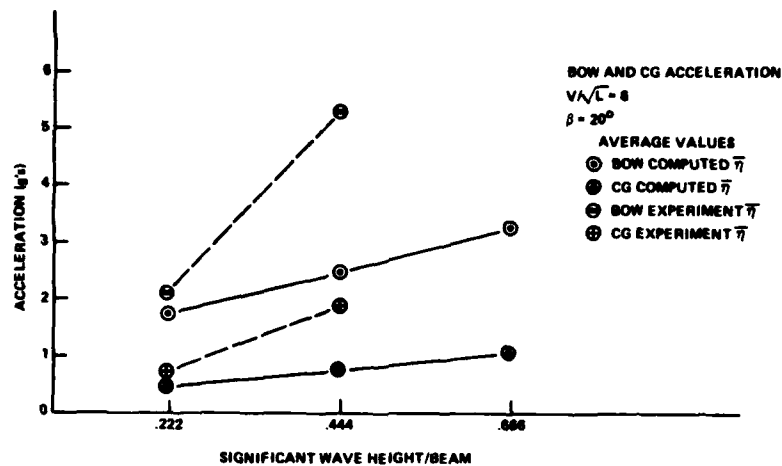


Figure 10 - Comparison of Computed and Experimental Bow and CG Acceleration Variation with Significant Wave Height

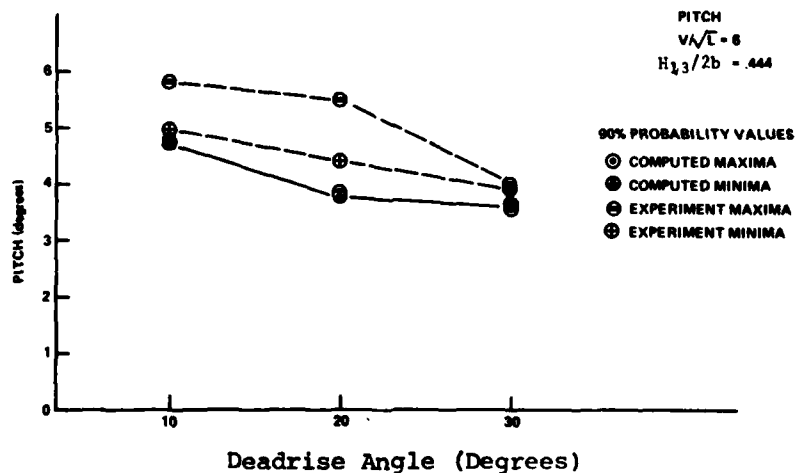


Figure 11 - Comparison of Computed and Experimental Pitch Variation with Deadrise Angle

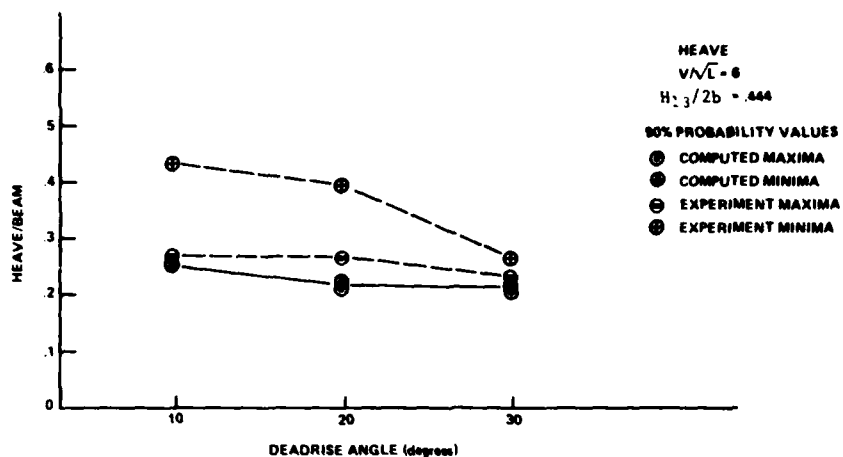


Figure 12 - Comparison of Computed and Experimental Heave Variation with Deadrise Angle

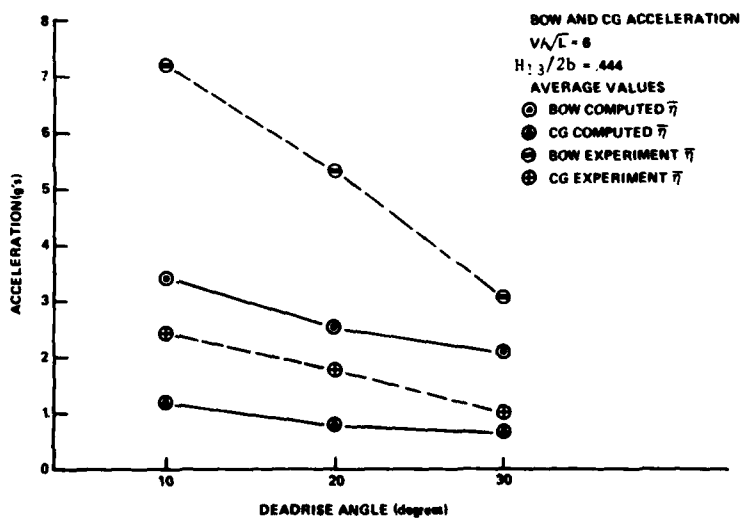


Figure 13 - Comparison of Computed and Experimental Bow and CG Acceleration Variation with Deadrise Angle

APPENDIX

LISTING OF COMPUTER PROGRAM FOR MOTION COMPUTATIONS

	PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE3=512,	MAIN	2
	TAPE2=512,TAPE4=512,TAPE4)	MAIN	3
C		MAIN	4
	REAL IT,M,MMAX,N,N0,NL,KAR	MAIN	5
	INTEGER END)	MAIN	6
C		MAIN	7
	DIMENSION X(6),FX(2,2000)	MAIN	8
C		MAIN	9
	COMMON /CONST/ NCG,ECG,PI,DPR,RPD,GRAVITY,RMU,NUM,MA(120),CD,TA,	MAIN	10
	B(120),BETA,HM(120),T2,URAG,W,XD,T,XP,M,IT,	MAIN	11
1	DELTAS,TA,EST(120),KAR,MMAX(120),TEST(120),	MAIN	12
	N(120),PHALF	MAIN	13
	COMMON /SHIP/ MASS,CINT,QA,CE,CE2,CE3,OMU,EUMU,E20MU,E30MU,BF,BMM,	MAIN	14
	NL,FL,IA,E(120)	MAIN	15
	COMMON /IN/ BM(120),RI(120),VELIN	MAIN	16
	COMMON/OUT/NPRINT,NPLOT,END	MAIN	17
	COMMON/TRANS/ T1,T2,T3,T4,T5,T6,T7,T8	MAIN	18
	COMMON /TRANS/ START,RISE,RAMP	MAIN	19
	COMMON /INTER/ II,KTT(10),DIFF(10)	MAIN	20
	COMMON /IN2/ NG(120),XA,XE,MMAX,MMIN,A(6),EPSE(6)	MAIN	21
	COMMON /ACCEL / XACCL,BWACL,CGACL,BL	MAIN	22
C		MAIN	23
	CALL INPUT	MAIN	24
C		MAIN	25
C	COMPUTE INTEGRATION INTERVAL INFORMATION	MAIN	26
C		MAIN	27
	NLESS = NUM-1	MAIN	28
	I = 1	MAIN	29
	II = 1	MAIN	30
	DIFFER = EST(I+1)-EST(I)	MAIN	31
	KTT(II) = 1	MAIN	32
	DIFF(II) = DIFFER	MAIN	33
	DO 25 I=2,NLESS	MAIN	34
	DIFFER= EST(I+1)-EST(I)	MAIN	35
	KTT(II) = KTT(II)+1	MAIN	36
	IF(DIFFER.NE.DIFF(II))GO TO 24	MAIN	37
	GO TO 25	MAIN	38
24	II = II+1	MAIN	39
	KTT(II) = 1	MAIN	40
	DIFF(II) = DIFFER	MAIN	41
25	CONTINUE	MAIN	42
	KTT(II) = KTT(II)+1	MAIN	43
C	*** CHECK IF NUMBER OF INTERVALS EXCEEDS DIMENSION	MAIN	44
	IF (II.GT.10) WRITE(6,28) (KTT(I),DIFF(I),I=1,II)	MAIN	45
	IF(II.GT.10) STOP 4	MAIN	46
C	*** POINT AT WHICH MULTIPLE RUNS START	MAIN	47
	8 CONTINUE	MAIN	48
	CALL SEAWAY	MAIN	49
	CALL TABLE	MAIN	50
	TIME=XA	MAIN	51
	KOUNT=1	MAIN	52
	END=END-1	MAIN	53
	WRITE(6,39)	MAIN	54
	39 FORMAT(1H)	MAIN	55
C	*** READ IN INITIAL CONDITIONS	MAIN	56
C	X(1) = VELOCITY, X(2) = Z OUT, X(3) = THETA DOT	MAIN	57
C	X(4) = X, X(5) = Z, X(6) = THETA	MAIN	58
C	THETA IS DEAD IN DEGREES THEN CONVERTED TO RADIANS IN PROGRAM	MAIN	59
C		MAIN	60

C	READ(5,10) (X(I), I=1,6)	MAIN 61
C	DATA . USED IN RAMP FUNCTION, TO TURN ON WAVE	MAIN 62
C	READ(5,10) START,RISE	MAIN 63
C	10 FORMAT(8F10.4)	MAIN 64
C	* * * * * WRITE OUT THE INPUT VALUES	MAIN 65
C	WRITE(6,19) START,RISE,KAR	MAIN 66
C	19 FORMAT(" START = ",F10.4,"/," RISE = ",F10.4,"/," KAR = ",F10.4)	MAIN 67
C		MAIN 68
C	TME IS THE TIME AT WHICH THE INTEGRATION INTERVAL IS	MAIN 69
C	TO BE CHANGED	MAIN 70
C	HMX IS THE NEW MAXIMUM INTERVAL SIZE AFTER TIME TME	MAIN 71
C	HMN IS THE NEW MINIMUM INTERVAL SIZE FOR KUTMER TO SUB-DIVIDE	MAIN 72
C	THE MAXIMUM INTERVAL UP TO	MAIN 73
C	IF THIS OPTION IS NOT USED SET TME TO THE STOP TIME OF THE RUN	MAIN 74
C		MAIN 75
C	READ(5,10) TME,HMX,HMN	MAIN 76
C	WRITE(6,11) TME,HMX,HMN,HMIN,HMN	MAIN 77
C	11 FORMAT(* AT TIME *,F7.2,* THE MAXIMUM INTERVAL SIZE FOR INTEGRATION	MAIN 78
C	WILL BE CHANGED FROM *,F10.4,* TO *,F10.4,/*	MAIN 79
C	* AND THE MINIMUM SIZE FOR HALVING CHANGES FROM *,F10.4,	MAIN 80
C	* TO *,F10.4)	MAIN 81
C	ADJUST THE TIME FOR CHANGE OF INTEGRATION INTERVAL	MAIN 82
C	FOR CHECK AGAINST TIME IN THE INTEGRATION LOOP	MAIN 83
C	TM = TME-(HMX/2.)	MAIN 84
C		MAIN 85
C	IPT = 0	MAIN 86
C	IF(TME.EQ.XE) IPT = 1	MAIN 87
C		MAIN 88
C	READ(5,10) PERCNT	MAIN 89
C	XACCL = ECG-PERCNT*BL	MAIN 90
C	WRITE(6,12) PERCNT,XACCL	MAIN 91
C	12 FORMAT(* THE X USED FOR THE BOW AND CG ACCELERATION COMPUTATIONS	MAIN 92
C	IS EQUAL TO ECG=*,F10.4,7H*HL OR *,F10.4)	MAIN 93
C		MAIN 94
C	WRITE(6,23)	MAIN 95
C	WRITE(6,27)	MAIN 96
C	23 FORMAT(1H,///)	MAIN 97
C	47 FORMAT(" STATION NO.",3X,"DEAD RISE",8X,"EST",8X,"NU",	MAIN 98
C	* 10X,"BEA")	MAIN 99
C	WRITE(6,55) ((1,BETA,EST(I),NU(I),BM(I)), I=1,NUM)	MAIN 100
C	55 FORMAT(6X,12,5X,F10.4,4X,F10.4,4X,F10.4,3X,F10.4)	MAIN 101
C	WRITE(6,23)	MAIN 102
C	WRITE(6,54) (X(I), I=1,6)	MAIN 103
C	56 FORMAT(" X VALUES",4X,6(F10.4,2X))	MAIN 104
C	* * * * * CHANGE INPUT FROM DEGREES TO RADIANS	MAIN 105
C	X(3) = X(3)*RPD	MAIN 106
C	X(6) = X(6)*RPD	MAIN 107
C		MAIN 108
C	* * * * * WRITE OUT COMPUTED ARRAYS	MAIN 109
C	WRITE(6,57) M,IT,PMHALF,PI,GRAVITY	MAIN 110
C	IF(NPRINT.LT.4) GO TO 62	MAIN 111
C	WRITE(6,58) (E(I), I=1,NUM)	MAIN 112
C	WRITE(6,59) (N(I), I=1,NUM)	MAIN 113
C	WRITE(6,60) (HMX(I), I=1,NUM)	MAIN 114
C	WRITE(6,61) (EST(I), I=1,NUM)	MAIN 115
C		MAIN 116
C		MAIN 117
C		MAIN 118
C		MAIN 119

62 CONTINUE	MAIN 120
WRITE(6,28) (KTY(I),DIFF(I),I=1,11)	MAIN 121
28 FORMAT(1X,KTY,DIFF,110.2X,F10.4)	MAIN 122
57 FORMAT(4H M=,F10.4,4H I=,F10.4,11H P1*RHU/2=,F10.4,	MAIN 123
1 SH P1=,F10.4,10H GRAVITY=,F10.4)	MAIN 124
58 FORMAT(1X,E(1),10F10.4)	MAIN 125
59 FORMAT(1X,N(1),10F10.4)	MAIN 126
64 FORMAT(1X,HMAX(1),10F10.4)	MAIN 127
66 FORMAT(1X,TEST(1),10F10.4)	MAIN 128
IB = 1	MAIN 129
IPRINT = NPRINT	MAIN 130
WRITE(4,91)	MAIN 131
C * * * * * WRITE HEADINGS AND CONDITIONS AT TIME = 0.	MAIN 132
91 FORMAT(1H1,2X,"TIME",9X,"XOUT",9X,"ZDOT",9X,"THETA DOT",6X,	MAIN 133
1HX,9X,1HZ,9X,5HTHETA,9X,2HNL,9X,2HFL,	MAIN 134
4X,8HBDW ACCL,4X,7HCG ACCL,77)	MAIN 135
WRITE(4,92) TIME,(X(I),I=1,6),NL,FL,BWACL,CGACL	MAIN 136
WRITE(9) TIME,(X(I),I=4,6),BWACL,CGACL	MAIN 137
KOUNT = KOUNT+1	MAIN 138
FX(1,18)=X(5)	MAIN 139
FX(2,18)=X(6)	MAIN 140
IKUTH = (TIME-XA)/HMAX + (XE-TME)/HMX + .05	MAIN 141
FIRST=0.0	MAIN 142
NEQS=6	MAIN 143
IKUTS=0	MAIN 144
C	MAIN 145
C START OF INTEGRATION LOOP	MAIN 146
C	MAIN 147
851 CONTINUE	MAIN 148
NPRINT = IPRINT	MAIN 149
C * * * * * CHECK PITCH ,GT. .5236 RADIANS	MAIN 150
IF(X(6),GT.,.5236) GO TO 853	MAIN 151
C * * * * * PERFORM INTEGRATIONS	MAIN 152
IF(TIME,LT,TIME,UR,TIME,EQ,XE) GO TO 98	MAIN 153
IF(IPT,EQ,1) GO TO 98	MAIN 154
HMIN = HMX	MAIN 155
HMAX = HMX	MAIN 156
FIRST = 0.0	MAIN 157
98 CONTINUE	MAIN 158
CALL KUTMER(NEQS,TIME,HMAX,X,EPSE,A,HMIN,FIRST)	MAIN 159
IKUTS=IKUTS+1	MAIN 160
IF(FIRST,EQ,2) GO TO 861	MAIN 161
IF(KOUNT,NE,1,AND,KOUNT,NE,41) GO TO 99	MAIN 162
WRITE(4,91)	MAIN 163
KOUNT=1	MAIN 164
C * * * * * WRITE OUT TIME INTERVAL RESULTS	MAIN 165
99 WRITE(4,92) TIME,(X(I),I=1,6),NL,FL,BWACL,CGACL	MAIN 166
WRITE(6,93) T1,T2,T3,T4,T5,T6,T7,T8,BMM,BF	MAIN 167
WRITE(9) TIME,(X(I),I=4,6),BWACL,CGACL	MAIN 168
IF(TIME,LT,TIME,UR,TIME,EQ,XE) GO TO 200	MAIN 169
IF(IPT,EQ,1) GO TO 200	MAIN 170
CALL PLOTER (FX,XA,HMAX,18,IPT)	MAIN 171
IPT = 1	MAIN 172
IB = 0	MAIN 173
XA = TIME	MAIN 174
FIRST = 0.0	MAIN 175
HMIN = HMX	MAIN 176
HMAX = HMX	MAIN 177
200 CONTINUE	MAIN 178

10=10+1	MAIN 179
FX(1,10)=X(5)	MAIN 180
FX(2,10)=X(6)	MAIN 181
93 FORMAT(" ",10E10.4)	MAIN 182
92 FORMAT(1X,11(F10.4,2X))	MAIN 183
100 CONTINUE	MAIN 184
KOUNT=KOUNT+1	MAIN 185
21 CONTINUE	MAIN 186
IF(TIME.LE.XE.AND.IKUTS.LT.IKUTM)GO TO 851	MAIN 187
WRITE(2,852)	MAIN 188
854 CONTINUE	MAIN 189
852 FORMAT(" END OF KUTMER")	MAIN 190
853 CONTINUE	MAIN 191
CALL PLOT2(FX,XA,HMAX,10,1PT)	MAIN 192
C * * * * * CHECK FOR LAST RUN IF NOT CYCLE BACK TO READ	MAIN 193
C NEW DATA FOR NEXT RUN	MAIN 194
IF(END.NE.1)GO TO 4	MAIN 195
GO TO 999	MAIN 196
C * * * * KUTMER ERROR MESSAGES	MAIN 197
861 WRITE(6,862)	MAIN 198
862 FORMAT(" ERROR CRITERION IN KUTMER CAN NOT BE MET")	MAIN 199
WRITE(6,863) (X(I),I=1,6)	MAIN 200
WRITE(6,864) TIME	MAIN 201
86 FORMAT(" TIME =",F10.4)	MAIN 202
IF(END.NE.1)GO TO 8	MAIN 203
GO TO 853	MAIN 204
999 CONTINUE	MAIN 205
END FILE 9	MAIN 206
END	MAIN 207
SUBROUTINE PLOT2(F,FMIN,FMAX,NVAR,NFUN,N1,N,X0,DELX)	PLOT2 2
C	PLOT2 3
C PLOT FIRST N POINTS OF UP TO 26 FUNCTIONS F(X)	PLOT2 4
C F(I,J) CONTAINS THE VALUE FOR THE JTH POINT OF THE ITH FUNCTION	PLOT2 5
C FMIN(I) AND FMAX(I) CONTAIN THE MIN AND MAX ORDINATE VALUES FOR	PLOT2 6
C THE ITH FUNCTION.	PLOT2 7
C NVAR(I) AN ARRAY OF TITLES FOR THE VARIOUS FUNCTIONS	PLOT2 8
C TO BE PLOTTED AGAINST THE ABSCISSA	PLOT2 9
C NFUN NUMBER OF FUNCTIONS TO BE PLOTTED - DIMENSION OF	PLOT2 10
C NVAR, FMIN, FMAX	PLOT2 11
C N1 USED ONLY IN F(N1,1) AS PASSED DIMENSION	PLOT2 12
C N NUMBER OF POINTS IN A SINGLE PLOT FRAME	PLOT2 13
C X0 FIRST ABSCISSA VALUE	PLOT2 14
C DELX ABSCISSA INCREMENT	PLOT2 15
C	PLOT2 16
DIMENSION JSTEP(26),F(N1,N),FMIN(NFUN),FMAX(NFUN),VLAST(26),	PLOT2 17
1 VFIRST(26),HEAD(6),STEP(26)	PLOT2 18
INTEGER CH(26),NVAR(NFUN),OUT,ASTER,PLUS,BLANK	PLOT2 19
INTEGER C	PLOT2 20
INTEGER A(101)	PLOT2 21
C	PLOT2 22
DATA BLANK,OUT,ASTER,PLUS/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)	PLOT2 24
2 / 1HA,1HH,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ /	PLOT2 25
DATA CH(11),CH(12),CH(13),CH(14),CH(15),CH(16),CH(17),CH(18)	PLOT2 26
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.OR.N.LE.0) RETURN	PLOT2 31

```

C PRINT HEADINGS.
WRITE(6,46)
46 FORMAT (///)
DO 40 I=1,NFUN
30 TENM=ABS(FMAX(I)-FMIN(I))
EXP=1.
IF (TENM.EQ.0.) GO TO 2
C BRING TENM TO A VALUE BETWEEN 1 AND 10
IF(TENM.LT.1.) GO TO 1
3 IF(TENM.LT.10.) GO TO 2
EXP=EXP*10.
TENM=TENM*.1
GO TO 3
1 EXP=EXP*.1
TENM=TENM*10.
IF(TENM.GT.1.) GO TO 2
GO TO 1
C SET UP VALUE BETWEEN GRID LINES, RSTEP.
2 PSTEP=5.
IF(TENM.GE.5.) PSTEP=10.
IF(TENM.LT.2.) PSTEP=2.
5 RSTEP(I)=PSTEP*EXP*.1
C COMPUTE VALUE OF STARTING LINE, VFIRST.
FIRST=FMIN(I)/RSTEP(I)
IF(FMIN(I).LT.0.) FIRST=FIRST-1.
FIRST=AINT(FIRST)
VFIRST(I)=FIRST*RSTEP(I)
C CHECK END LINE VALUE, VLAST.
VLAST(I)=VFIRST(I)+10.*RSTEP(I)
IF(VLAST(I).GT.FMAX(I)) GO TO 4
C IF GRAPH IS TOO SMALL TAKE NEXT LARGER STEP.
AA=PSSTEP
IF(AA.LT.5.) PSTEP=5.
IF(AA.EQ.5.) PSTEP=10.
IF(AA.LT.10.) GO TO 5
PSTEP=2.
EXP=10.*EXP
GO TO 5
C COMPUTE VALUE BETWEEN POINTS, STEP.
4 STEP(I)=RSTEP(I)*.1
RK=0.
DO 6 KK=1,6
HEAD(KK)=VFIRST(I)+2.*RK*RSTEP(I)
6 RK=RK+1.
40 WRITE (6,45) CH(I), NVAR(I), (HEAD(KK),KK=1,6)
45 FORMAT(1X,41,3H = ,A10,5X,1PE12.4,5(8X,1PE12.4))
DO 50 J=1,101
A(J)=BLANK
IF(MOD(J,10).EQ.1) A(J)=DOT
50 CONTINUE
WRITE(6,55) A,A
55 FORMAT (25X,101A1/15X,4H TIME,6X,101A1)
C PLOT EACH POINT
DO 100 J=1,N
B=X0+FLOAT(J-1)*DELX
DO 70 K=1,101
A(K)=HLAYK
IF(MOD(K,10).EQ.1) A(K)=DOT
IF(MOD(J,5).EQ.1) A(K)=DOT

```

```

PLOT2 32
PLOT2 33
PLOT2 34
PLOT2 35
PLOT2 36
PLOT2 37
PLOT2 38
PLOT2 39
PLOT2 40
PLOT2 41
PLOT2 42
PLOT2 43
PLOT2 44
PLOT2 45
PLOT2 46
PLOT2 47
PLOT2 48
PLOT2 49
PLOT2 50
PLOT2 51
PLOT2 52
PLOT2 53
PLOT2 54
PLOT2 55
PLOT2 56
PLOT2 57
PLOT2 58
PLOT2 59
PLOT2 60
PLOT2 61
PLOT2 62
PLOT2 63
PLOT2 64
PLOT2 65
PLOT2 66
PLOT2 67
PLOT2 68
PLOT2 69
PLOT2 70
PLOT2 71
PLOT2 72
PLOT2 73
PLOT2 74
PLOT2 75
PLOT2 76
PLOT2 77
PLOT2 78
PLOT2 79
PLOT2 80
PLOT2 81
PLOT2 82
PLOT2 83
PLOT2 84
PLOT2 85
PLOT2 86
PLOT2 87
PLOT2 88
PLOT2 89
PLOT2 90

```

THIS DOCUMENT IS BEST QUALITY AVAILABLE
THE COPY FURNISHED TO DDC CONTAINED A
SIGNIFICANT NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.

70 CONTINUE	PLOT2 91
DO 40 I=1,NFUN	PLOT2 92
LOC=((F(I,J)-VFIRST(I))/STEP(I)+1.5)	PLOT2 93
C=A(LOC)	PLOT2 94
A(LOC)=CH(I)	PLOT2 95
IF(C.NE.BLANK.AND.C.NE.DOT) A(LOC)=ASTER	PLOT2 96
80 CONTINUE	PLOT2 97
IF(MOD(J,10).EQ.1)GO TO 95	PLOT2 98
WRITE(6,85) A	PLOT2 99
85 FORMAT (25X,101A1)	PLOT2100
GO TO 100	PLOT2101
95 WRITE(6,15)B,A	PLOT2102
15 FORMAT (12X,1PE12.4,1X,101A1)	PLOT2103
100 CONTINUE	PLOT2104
RETURN	PLOT2105
END	PLOT2106
SUBROUTINE KUTMER(N0,T,H,Y0,EPSE,A,MCX,FIRST)	KUTMER 2
DIMENSION Y0(6),Y1(6),Y2(6),FU(6),F1(6),F2(5),EPSE(6),A(6)	KUTMER 3
COMMON/OUT/NPRINT,NPLOT,END	KUTMER 4
COMMON /ACCL / XACCL,BWACL,CGACL,DL	KUTMER 5
DATA NAM1,NAM2 /2HY1,2HY2 /	KUTMER 6
C	KUTMER 7
C NO = NUMBER OF EQUATIONS. NO. OF COMPONENTS OF Y0	KUTMER 8
C T = INDEPENDENT VARIABLE	KUTMER 9
C H = INCREMENT FOR WHICH SOLUTION IS TO BE RETURNED. OR -	KUTMER10
C Y0 = THE VECTOR OF DEPENDENT VARIABLES. ENTER WITH INITIAL	KUTMER11
C VALUES AT T AND RETURN WITH VALUES AT T+H	KUTMER12
C EPSE = RELATIVE ERROR CRITERION FOR COMPONENTS OF Y0 .GT. ABS(A)	KUTMER13
C A = ABSOLUTE ERROR CRITERION FOR COMPONENTS OF Y0 .LT. ABS(A)	KUTMER14
C NOTE-- EPSE AND A MUST BE SPECIFIED FOR EACH COMPONENT OF THE SYSTEM	KUTMER15
C MCX = THE SMALLEST STEP SIZE USED IN THE INTEGRATION	KUTMER16
C FIRST SHOULD BE 0 WHEN KUTMER IS ENTERED FOR THE FIRST TIME	KUTMER17
C AFTER THAT FIRST IS 1 IF KUTMER IS ENTERED WITH THE SAME H OR	KUTMER18
C IF IT IS ENTERED WITH A CHANGED H	KUTMER19
C IF FIRST IS 2 THE ERROR CRITERIA CANNOT BE MET AND THE STEP SIZE	KUTMER20
C REDUCED TO H/128.	KUTMER21
C	KUTMER22
C IF (FIRST) 20,10,20	KUTMER23
C - - - - - FIRST ENTRY	KUTMER24
10 MC = H	KUTMER25
IPLUC = 1	KUTMER26
FIRST = 1.	KUTMER27
C - - - - - OTHER ENTRY	KUTMER28
20 LOC = 0	KUTMER29
MCX = MC	KUTMER30
IF (MC.NE.0.) GO TO 30	KUTMER31
WRITE(6,800)	KUTMER32
800 FORMAT(5X,45MKUTMER ENTERED WITH ZERO INTEGRATION INTERVAL)	KUTMER33
FIRST = 2.	KUTMER34
RETURN	KUTMER35
C - - - - - 5 CALLS TO DAUX	KUTMER36
30 CALL DAUX(T,Y0,F0)	KUTMER37
IF(NPRINT.EQ.3)WRITE(6,400)Y0,T,F0	KUTMER38
400 FORMAT(6(2X,F10.4),4HTIME,2X,F10.4)	KUTMER39
IF(NPRINT.EQ.3)WRITE(6,400)MC	KUTMER40
30 DO 40 I=1,ND	KUTMER41
40 Y1(I) = Y0(I)+(MC/J.)*FU(I)	KUTMER42
IF(NPRINT.EQ.3)WRITE(6,400)Y1,T	KUTMER43
C	KUTMER44

CALL DAUX(T+MC/3.,Y1,F1)	KUTMER45
IF(NPRINT.EQ.5)WRITE(6,400)F1,T	KUTMER46
DO 50 I=1,ND	KUTMER47
50 Y1(I) = Y0(I)+(MC/6.)*F0(I)+(MC/6.)*F1(I)	KUTMER48
IF(NPRINT.EQ.5)WRITE(6,400)Y1,T	KUTMER49
C	KUTMER50
CALL DAUX(T+MC/3.,Y1,F1)	KUTMER51
IF(NPRINT.EQ.5)WRITE(6,400)F1,T	KUTMER52
DO 60 I=1,ND	KUTMER53
60 Y1(I) = Y0(I)+(MC/8.)*F0(I)+.375*MC*F1(I)	KUTMER54
IF(NPRINT.EQ.5)WRITE(6,400)Y1,T	KUTMER55
C	KUTMER56
CALL DAUX(T+MC/2.,Y1,F2)	KUTMER57
IF(NPRINT.EQ.5)WRITE(6,400)F2,T	KUTMER58
DO 70 I=1,ND	KUTMER59
70 Y1(I) = Y0(I)+(MC/2.)*F0(I)-1.5*MC*F1(I)+2.*MC*F2(I)	KUTMER60
IF(NPRINT.EQ.5)WRITE(6,400)Y1,T	KUTMER61
C	KUTMER62
CALL DAUX(T+MC,Y1,F1)	KUTMER63
IF(NPRINT.EQ.5)WRITE(6,400)F1,T	KUTMER64
DO 80 I=1,ND	KUTMER65
80 Y2(I) = Y0(I)+MC/6.*F0(I)+(2./3.)*MC*F2(I)+(MC/6.)*F1(I)	KUTMER66
IF(NPRINT.EQ.5)WRITE(6,400)Y2,T	KUTMER67
INC = 0	KUTMER68
C - - - - - CHECK ERROR CRITERIA	KUTMER69
DO 110 I=1,ND	KUTMER70
ZZZ = ABS(Y1(I))-A(I)	KUTMER71
IF (ZZZ) 85,87,87	KUTMER72
C - - - - - ABSOLUTE ERROR	KUTMER73
85 ERROR = ABS(.2*(Y1(I)-Y2(I)))	KUTMER74
IF (ERROR-A(I)) 100,100,90	KUTMER75
C - - - - - RELATIVE ERROR	KUTMER76
87 ERROR = ABS(.2-.2*Y2(I)/Y1(I))	KUTMER77
IF (ERROR-EMSE(I)) 100,100,90	KUTMER78
C - - - - - SINCE ERROR .GT. ERROR CRITERIA CHECK IF MC.GT.H/KUTMER79	KUTMER79
C - - - - - IF YES THEN HALVE INTERVAL. OTHERWISE STOP.	KUTMER80
90 X = 128.*ABS(MC)-ABS(H)	KUTMER81
IF (X) 91,95,95	KUTMER82
C - - - - - ERROR TOO LARGE	KUTMER83
91 WRITE(6,92)I,T,ERROR,MC	KUTMER84
92 FORMAT(/18H FOR EQUATION NO. 12,2/H, THE RELATIVE ERROR AT T = ,	KUTMER85
E15.8, 4H IS .E15.8,13H STEP SIZE = .E15.8)	KUTMER86
FIRST = 2.	KUTMER87
RETURN	KUTMER88
C - - - - - HALVE INTERVAL	KUTMER89
95 MC = MC/2.	KUTMER90
IPLOC = 2*IPLOC	KUTMER91
LLOC = 2*LLOC	KUTMER92
MCX = MC	KUTMER93
WRITE(2,710)T,1,ERROR,MC	KUTMER94
710 FORMAT(/8H TIME = .F10.3,5X,20HHALVE INTERVAL. EQUATION .13,	KUTMER95
13H HAS ERROR = .E16.8,6X,17H STEP SIZE NOW = .E15.8)	KUTMER96
WRITE(2,720) 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
C - - - - - TEST IF INTERVAL LENGTH CAN BE DOUBLED	KUTME101
100 IF (ERROR*.54.-EMSE(I)) 110,110,101	KUTME102
101 INC = 1	KUTME103

110	CONTINUE	KUTME104
C	----- UPDATE T AND SOLUTION	KUTME105
111	T = T+MC	KUTME106
	DO 112 I=1,N0	KUTME107
112	Y0(I) = Y2(I)	KUTME108
C	----- GET SOLUTION IN NEXT INTERVAL	KUTME109
	LOC = LOC+1	KUTME110
	IF (LOC-IPLOC) 120,210,210	KUTME111
120	IF (INC) 210,130,210	KUTME112
130	IF (LOC-(LOC/2)*2) 210,140,210	KUTME113
140	IF (IPLOC-1) 210,210,200	KUTME114
C	----- DOUBLE INTERVAL LENGTH	KUTME115
200	MC = 2.*MC	KUTME116
	LOC = LOC /2	KUTME117
	IPLOC = IPLOC/2	KUTME118
210	IF (IPLOC-LOC) 30,329,30	KUTME119
329	BWACL = F0(2)-XACCL*F0(3)	KUTME120
	CGACL = F0(2)	KUTME121
	RETURN	KUTME122
	END	KUTME123
	SUBROUTINE DAUX(TIME,X,RHS)	DAUX 2
C	TIME	DAUX 3
C	TIME AT WHICH SYSTEM IS TO BE EVALUATED	DAUX 4
C	X	DAUX 5
C	STATE VECTOR	DAUX 6
C	RHS	DAUX 7
	THE RIGHT HAND SIDE OF THE EQUATION S = F A	DAUX 8
	REAL KAR	DAUX 9
	REAL IA,IT,M,K,MA,MASS,NL,N,MMAX	DAUX 10
	1 DIMENSION X(6),RHS(6),F(3,1),A(3,3),INDEX(3,3),	DAUX 11
	1 Q(120),U(120),V(120),WOTIME(20)	DAUX 12
C	COMMON /SHIP/ MASS,CINT,QA,CE,CE2,CE3,DMU,EDMU,E2DMU,E3DMU,BF,BMM,	DAUX 13
	NL,FL,IA,E(120)	DAUX 14
	COMMON /CONST/ NCG,ECG,PI,DPH,MPD,GRAVITY,RMU,NUM,MA(120),CD,TA,	DAUX 15
	B(120),HETA,HW(120),IZ,URAG,W,XD,T,XP,M,IT,	DAUX 16
	1 U,LTAS,TA,EST(120),KAR,MMAX(120),TEST(120),	DAUX 17
	N(120),PMALF	DAUX 18
	COMMON /IN/ BM(120),BI(120),VELIN	DAUX 19
	COMMON/OUT/NPRINT,NPLOT,END	DAUX 20
	COMMON /TRANS/ START,RISE,RAMP	DAUX 21
	COMMON /WAVE1/ ZMA,ZWMA,EMAS,ZZWMA,ZWEMA,ZZWMA,E2MAZ,	DAUX 22
	1 ZWOUT(120),ASINPT(120,20),ACOSPT(120,20),CX6, SX6	DAUX 23
	COMMON /WAVE2/ W0(20),K(20),C(20),RW0(20),RW02(20),RK(20),	DAUX 24
	1 RO(20),RWK(20),H,PHS(20)	DAUX 25
C	COMMON/SINE/POINT(1000)	DAUX 26
	RAMP = RMP/TIME,START,RISE)	DAUX 27
	PIH = PI/2.	DAUX 28
	DO 5 JJ =1,10	DAUX 29
	WOTIME(JJ) = W0(JJ)*TIME+PHS(JJ)	DAUX 30
	PNTRAD = 319.3098861	DAUX 31
5	CONTINUE	DAUX 32
	CX6 = COS(X(6))	DAUX 33
	SX6 = SIN(X(6))	DAUX 34
C****	SET VALUES OF MA AND B	DAUX 35
	DO 75 I=1,NUM	DAUX 36
	RI = 0.0	DAUX 37
	PTO = X(4)*E(1)*CX6*N(I)*SX6	DAUX 38
	VI = 0.0	DAUX 39
		DAUX 40

77 FORMAT (" R(I)",10F10.4)	DAUX 100
78 FORMAT (" W(I)",10F10.4)	DAUX 101
79 FORMAT (" H(I)",10F10.4)	DAUX 102
80 FORMAT (" V(I)",10F10.4)	DAUX 103
81 FORMAT (" D(I)",10F10.4)	DAUX 104
82 FORMAT (" MA(I) ",10F10.4)	DAUX 105
83 CONTINUE	DAUX 106
C	DAUX 107
C * * * * * COMPUTES NL AND FL AND THE ASSOCIATED INTEGRALS	DAUX 108
CALL FUNCT(X)	DAUX 109
C	DAUX 110
IF(NPRINT.LT.4)GO TO 17	DAUX 111
WRITE(6,15) TX,FL,DRAG,TZ,W,NL,XD,T,XP	DAUX 112
15 FORMAT(" ",10E12.0)	DAUX 113
17 CONTINUE	DAUX 114
C * * * * * COMPUTE THE F VECTOR	DAUX 115
F(1,1) = TX*FL*SX6-DRAG*CX6	DAUX 116
F(1,1)=0.0	DAUX 117
F(2,1) = T7*FL*CX6+DRAG*SX6-W	DAUX 118
F(3,1)=NL-DRAG*XD-T*XP	DAUX 119
IF(NPRINT.LT.3)GO TO 18	DAUX 120
WRITE(6,10) (F(1,1),I=1,3)	DAUX 121
18 CONTINUE	DAUX 122
C * * * * * COMPUTE THE A MATRIX	DAUX 123
A(1,1) = M*MASS*SX6*CX6	DAUX 124
A(1,2) = MASS*SA6*CX6	DAUX 125
A(1,3) = -DA*SA6	DAUX 126
A(1,2) = 0.	DAUX 127
A(1,3) = 0.	DAUX 128
A(2,1)=A(1,2)	DAUX 129
A(2,2) = M*MASS*CX6*CX6	DAUX 130
A(2,3) = -DA*CA6	DAUX 131
A(3,1)=A(1,3)	DAUX 132
A(3,2)=A(2,3)	DAUX 133
A(3,3)=IT*IA	DAUX 134
IF(NPRINT.LT.3)GO TO 25	DAUX 135
WRITE(6,12) (A(1,1),I=1,3)	DAUX 136
WRITE(6,13) (A(1,2),I=1,3)	DAUX 137
WRITE(6,14) (A(1,3),I=1,3)	DAUX 138
C * * * * * INVERT THE A MATRIX	DAUX 139
25 CALL MATINS(A,J,3,F,1,1,DETERM,ID,INDEX)	DAUX 140
IF(ID.EQ.2)WRITE(6,26)	DAUX 141
26 FORMAT(" MATRIX IS SINGULAR ")	DAUX 142
C*****A ON RETURN WILL CONTAIN THE INVERSE MATRIX	DAUX 143
C ID=2 MATRIX IS SINGULAR	DAUX 144
C =1 INVERSE WAS FOUND	DAUX 145
C	DAUX 146
C * * * * * COMPUTE THE RIGHT HAND SIDE	DAUX 147
RHS(1) = F(1,1)	DAUX 148
RHS(2) = F(2,1)	DAUX 149
RHS(3) = F(3,1)	DAUX 150
RHS(1) = 0.0	DAUX 151
RHS(4) = X(1)	DAUX 152
RHS(5) = X(2)	DAUX 153
RHS(6) = X(3)	DAUX 154
10 FORMAT(" F(1,1) ",3(2X,E12.4))	DAUX 155
12 FORMAT(" A(1,1) ",3(2X,E12.4))	DAUX 156
13 FORMAT(" A(1,2) ",3(2X,E12.4))	DAUX 157
14 FORMAT(" A(1,3) ",3(2X,E12.4))	DAUX 158

30 IF(NPRINT.LT.2) GO TO 40	DAUX 159
WRITE(6,12) (A(1,1),I=1,3)	DAUX 160
WRITE(6,13) (A(1,2),I=1,3)	DAUX 161
WRITE(6,14) (A(1,3),I=1,3)	DAUX 162
WRITE(6,35) (RMS(1),I=1,6)	DAUX 163
35 FORMAT(" RMS(1) ",6(2X,E12.6))	DAUX 164
40 CONTINUE	DAUX 165
RETURN	DAUX 166
END	DAUX 167
SUBROUTINE FUNCT(X)	FUNCT 2
REAL KAR	FUNCT 3
REAL IA,IAA,IPART,KPI,MA,MASS,N	FUNCT 4
INTEGER EN	FUNCT 5
DIMENSION IPART(120),C1(120),C2(120),	FUNCT 6
U1(120),U2(120),U3(120),U4(120),U5(120),U6(120),	FUNCT 7
OPART(120),Z1(120),Z2(120),Z3(120),Z4(120),Z5(120),	FUNCT 8
Z6(120),Z7(120)	FUNCT 9
X(6),VMAA(120)	FUNCT 10
C	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 /CONST/ NCG,ECG,PI,OPR,RPD,GHVTY,RHU,NUM,MA(120),CD,TA,	FUNCT 14
B(120),BETA,HW(120),TZ,DRAG,W,XD,T,XP,M,IT,	FUNCT 15
DELTA,TA,EST(120),KAR,MMA(120),TEST(120),	FUNCT 16
N(120),PHALF	FUNCT 17
COMMON /IN/ HM(120),B1(120),VELIN	FUNCT 18
COMMON/OUT/NPRINT,NPLOT,END	FUNCT 19
COMMON /WAVE1/ ZMA,ZWMA,EMAS,ZZWMA,ZWEMA,ZZEMA,E2MAZ,	FUNCT 20
ZWDOT(120),ASINPT(120,20),ACOSPT(120,20),CX6, SX6	FUNCT 21
COMMON /WAVE2/ WU(20),K(20),C(20),RWU(20),RW2(20),RK(20),	FUNCT 22
RO(20),RWK(20),M,PHS(20)	FUNCT 23
COMMON /INTER/ II,KTT(10),DIFF(10)	FUNCT 24
COMMON/TRANS/START,RISE,RAMP	FUNCT 25
COMMON /TEST/ VMA	FUNCT 26
C * * * * * INITIALIZE INTEGRAL SUMS	FUNCT 27
MASS = 0.0	FUNCT 28
QA = 0.0	FUNCT 29
IA = 0.0	FUNCT 30
CE = 0.0	FUNCT 31
CE2 = 0.0	FUNCT 32
DMU = 0.0	FUNCT 33
EDMU=0.0	FUNCT 34
E2DMU = 0.0	FUNCT 35
E3DMU = 0.0	FUNCT 36
BF = 0.0	FUNCT 37
BMM = 0.0	FUNCT 38
ZMA = 0.0	FUNCT 39
ZWMA = 0.0	FUNCT 40
EMAS = 0.0	FUNCT 41
ZZWMA = 0.0	FUNCT 42
ZWEMA = 0.0	FUNCT 43
ZZEMA = 0.0	FUNCT 44
E2MAZ = 0.0	FUNCT 45
VPART = X(1)*SIN(X(6))+X(2)*COS(X(6))	FUNCT 46
C * * * * * SET UP FUNCTIONS FOR INTEGRALS * * * * *	FUNCT 47
DO 90 I=1,NUM	FUNCT 48
IPART(I)=C(I)*E(I)*MA(I)	FUNCT 49
OPART(I)=E(I)*MA(I)	FUNCT 50
ZWDOT = 0.0	FUNCT 51

THIS DOCUMENT CONTAINS NEITHER RECOMMENDATIONS NOR
 CONCLUSIONS OF THE NATIONAL BUREAU OF STANDARDS
 AND IS NOT INTENDED TO BE USED IN LEGAL PROCEEDINGS

UWDT = 0.0	FUNCT 52
URDT = 0.0	FUNCT 53
UWDE = 0.0	FUNCT 54
XNE = X(J)*(N(I)*CX6-E(I)*SX6)	FUNCT 55
XNEO = X(J)*(E(I)*CX6+N(I)*SX6)	FUNCT 56
DO 15 J = 1,10	FUNCT 57
ZWDTT = -RWDT(J)*ASINPT(I,J)	FUNCT 58
ZWDI = ZWDI+ZWDTT	FUNCT 59
UWDTT = -RWDT(J)*ACOSPT(I,J)	FUNCT 60
UWDT = DWDT+UWDTT	FUNCT 61
URDTT = -RWDT(J)*ASINPT(I,J)*(X(I)+C(J)*XNE)	FUNCT 62
URDT = DWDT+URDTT	FUNCT 63
DWDEE = -RWDT(J)*ACUSPT(I,J)	FUNCT 64
UWDE = DWDE+DWDEE	FUNCT 65
15 CONTINUE	FUNCT 66
ZWDOT(I) = ZWDI+RAMP	FUNCT 67
UWDE = DWDE+RAMP	FUNCT 68
URDT = DWDT+RAMP	FUNCT 69
U = X(I)*CX6-X(2)*SX6+ZWDOT(I)*SX6	FUNCT 70
VEL = VPART-X(J)*E(I)-ZWDOT(I)*CX6	FUNCT 71
Z1(I) = MA(I)*ZWDOT(I)	FUNCT 72
Z2(I) = MA(I)*UWDT+RAMP	FUNCT 73
Z3(I) = E(I)*Z2(I)	FUNCT 74
Z4(I) = E(I)*Z1(I)	FUNCT 75
Z5(I) = MA(I)*U*DWDE	FUNCT 76
Z6(I) = E(I)*Z5(I)	FUNCT 77
Z7(I) = MA(I)*VEL*U	FUNCT 78
IF (VEL.LE.0.) GO TO 60	FUNCT 79
IF (B1(I).LE.0.0) GO TO 50	FUNCT 80
U1(I) = VEL*B1(I)*(X(2)-XNEO-URDT)	FUNCT 81
GO TO 51	FUNCT 82
50 U1(I) = 0.	FUNCT 83
51 CONTINUE	FUNCT 84
U2(I) = E(I)*U1(I)	FUNCT 85
C1(I) = VEL*VEL*U1(I)	FUNCT 86
C2(I) = E(I)*C1(I)	FUNCT 87
GO TO 61	FUNCT 88
60 U1(I) = 0.	FUNCT 89
U2(I) = 0.	FUNCT 90
C1(I) = 0.	FUNCT 91
C2(I) = 0.	FUNCT 92
61 CONTINUE	FUNCT 93
U3(I) = MA(I)*UWDE*VEL	FUNCT 94
U4(I) = E(I)*U3(I)	FUNCT 95
PIH = PI/2.	FUNCT 96
U5(I) = B(I)*(MW(I)-B(I)*TA/2.)	FUNCT 97
65 U6(I) = U5(I)*C(I)*.5	FUNCT 98
90 CONTINUE	FUNCT 99
RMOG=RM0*GRAVITY	FUNCT100
PIH = PI/2.	FUNCT101
KPI = KAP*PI	FUNCT102
C EVALUATE INTEGRALS USING TRAP METHOD	FUNCT103
I = 1	FUNCT104
INDEX = 1	FUNCT105
91 CALL TRAP(1A(INDEX),DIFF(I),KIT(I),IMASS)	FUNCT106
CALL TRAP(1PART(INDEX),DIFF(I),KIT(I),OAI)	FUNCT107
CALL TRAP(C1(INDEX),DIFF(I),KIT(I),CEA)	FUNCT108
CALL TRAP(C2(INDEX),DIFF(I),KIT(I),CE2A)	FUNCT109
CALL TRAP(1PART(INDEX),DIFF(I),KIT(I),IAA)	FUNCT110

```

CALL TRAP(D1(INDEX),DIFF(I),KTT(I),DMUA)
CALL TRAP(D2(INDEX),DIFF(I),KTT(I),EDMUA)
CALL TRAP(D3(INDEX),DIFF(I),KTT(I),E2DMUA)
CALL TRAP(D4(INDEX),DIFF(I),KTT(I),E3DMUA)
CALL TRAP(D5(INDEX),DIFF(I),KTT(I),BFA)
CALL TRAP(D6(INDEX),DIFF(I),KTT(I),BMM)
CALL TRAP(D7(INDEX),DIFF(I),KTT(I),ZMA)
CALL TRAP(D8(INDEX),DIFF(I),KTT(I),ZWMA)
CALL TRAP(D9(INDEX),DIFF(I),KTT(I),EMAS)
CALL TRAP(D10(INDEX),DIFF(I),KTT(I),ZWMMA)
CALL TRAP(D11(INDEX),DIFF(I),KTT(I),ZWMMA)
CALL TRAP(D12(INDEX),DIFF(I),KTT(I),ZWMMA)
CALL TRAP(D13(INDEX),DIFF(I),KTT(I),ZWMMA)
CALL TRAP(D14(INDEX),DIFF(I),KTT(I),ZWMMA)
CALL TRAP(D15(INDEX),DIFF(I),KTT(I),ZWMMA)
CALL TRAP(D16(INDEX),DIFF(I),KTT(I),ZWMMA)
CALL TRAP(D17(INDEX),DIFF(I),KTT(I),ZWMMA)

```

C

93 CONTINUE

```

MASS = MASS + TMASS
QA = QA + QA1
IA = IA + IAA
CE = CE + CEA
CE2 = CE2 + CE2A
UMU = DMU + DMUA
EDMU = EDMU + EDMUA
E2DMU = E2DMU + E2DMUA
E3DMU = E3DMU + E3DMUA
BF = BF + DMUG*BFA
BMM = BMM + RMUG*BMMMA
ZMA = ZMA + ZMAA
ZWMA = ZWMA + ZWMAA
EMAS = EMAS + EMASA
ZZWMA = ZZWMA + ZZWMAA
ZWEMA = ZWEMA + ZWEMA
ZZWMA = ZZWMA + ZZWMAA
EZMAZ = EZMAZ + EZMAZA

```

94 CONTINUE

```

IF (I,GE,II) GO TO 92
INDEX = INDEX + KTT(I) - 1
I = I + 1
GO TO 91

```

92 CONTINUE

C

C * * * * * CALL COMPUT TO FIND THE VALUE OF NL AND FL USING
C THE VALUES OF THE ABOVE INTEGRALS
CALL COMPUT(X)

C

```

IF(NPRINT,LT,3) GO TO 111
IF(NPRINT,EQ,3) GO TO 108
IF(NPRINT,GT,3) GO TO 108
WRITE(6,97) (IPART(I),I=1,NUM)
WRITE(6,98) (OPART(I),I=1,NUM)
WRITE(6,99) (C1(I),I=1,NUM)
WRITE(6,100) (C2(I),I=1,NUM)
WRITE(6,102) (U1(I),I=1,NUM)
WRITE(6,103) (U2(I),I=1,NUM)
WRITE(6,104) (U3(I),I=1,NUM)
WRITE(6,105) (U4(I),I=1,NUM)
WRITE(6,106) (U5(I),I=1,NUM)
WRITE(6,112) (D6(I),I=1,NUM)
WRITE(6,113) (Z1(I),I=1,NUM)
WRITE(6,114) (Z2(I),I=1,NUM)

```

FUNCT111
 FUNCT112
 FUNCT113
 FUNCT114
 FUNCT115
 FUNCT116
 FUNCT117
 FUNCT118
 FUNCT119
 FUNCT120
 FUNCT121
 FUNCT122
 FUNCT123
 FUNCT124
 FUNCT125
 FUNCT126
 FUNCT127
 FUNCT128
 FUNCT129
 FUNCT130
 FUNCT131
 FUNCT132
 FUNCT133
 FUNCT134
 FUNCT135
 FUNCT136
 FUNCT137
 FUNCT138
 FUNCT139
 FUNCT140
 FUNCT141
 FUNCT142
 FUNCT143
 FUNCT144
 FUNCT145
 FUNCT146
 FUNCT147
 FUNCT148
 FUNCT149
 FUNCT150
 FUNCT151
 FUNCT152
 FUNCT153
 FUNCT154
 FUNCT155
 FUNCT156
 FUNCT157
 FUNCT158
 FUNCT159
 FUNCT160
 FUNCT161
 FUNCT162
 FUNCT163
 FUNCT164
 FUNCT165
 FUNCT166
 FUNCT167
 FUNCT168
 FUNCT169

BEST QUALITY PRACTICE
 COPY FURNISHED TO DDC CONTAINING A
 SIGNIFICANT NUMBER OF PAGES WHICH DO NOT
 REPRODUCE LIBRILY

WRITE(6,115) (Z3(I),I=1,NUM)	FUNCT170
WRITE(6,116) (Z4(I),I=1,NUM)	FUNCT171
WRITE(6,117) (Z5(I),I=1,NUM)	FUNCT172
WRITE(6,118) (Z6(I),I=1,NUM)	FUNCT173
WRITE(6,120) (Z7(I),I=1,NUM)	FUNCT174
WRITE(6,107) KPI,RHUG,PHI	FUNCT175
108 WRITE(6,109) MASS,CINT,QA,CE,CE2,CE3	FUNCT176
WRITE(6,121) IA	FUNCT177
121 FORMAT(* IA *,E10.4)	FUNCT178
WRITE(6,110) DMU,EDMU,E2DMU,E3DMU,BF,BMM	FUNCT179
WRITE(6,117) ZMA,ZMA,EMAS,ZZWMA,ZWEMA,ZZWMA,E2MAZ	FUNCT180
96 FORMAT(" CPART(I)",10(2X,E10.4))	FUNCT181
97 FORMAT(" IPART(I)",10(2X,E10.4))	FUNCT182
98 FORMAT(" QPART(I)",10(2X,E10.4))	FUNCT183
99 FORMAT(" C1 ",10(2X,E10.4))	FUNCT184
100 FORMAT(" C2 ",10(2X,E10.4))	FUNCT185
101 FORMAT(" C3 ",10(2X,E10.4))	FUNCT186
102 FORMAT(" D1 ",10(2X,E10.4))	FUNCT187
103 FORMAT(" D2 ",10(2X,E10.4))	FUNCT188
104 FORMAT(" D3 ",10(2X,E10.4))	FUNCT189
105 FORMAT(" D4 ",10(2X,E10.4))	FUNCT190
106 FORMAT(" D5 ",10(2X,E10.4))	FUNCT191
112 FORMAT(" D6 ",10(2X,E10.4))	FUNCT192
107 FORMAT(" KPI ",E10.4," RHUG ",E10.4," PHI ",E10.4)	FUNCT193
109 FORMAT(" MASS ",E10.4," CINT ",E10.4," QA ",E10.4," CE ",E10.4,	FUNCT194
"CE2 ",E10.4," CE3 ",E10.4)	FUNCT195
110 FORMAT(" DMU ",E10.4," EDMU ",E10.4," E2DMU ",E10.4," E3DMU ",	FUNCT196
"E10.4," BF ",E10.4," BMM ",E10.4)	FUNCT197
113 FORMAT(4H 71 ,10(2X,E10.4))	FUNCT198
114 FORMAT(4H 72 ,10(2X,E10.4))	FUNCT199
115 FORMAT(4H 73 ,10(2X,E10.4))	FUNCT200
116 FORMAT(4H 74 ,10(2X,E10.4))	FUNCT201
118 FORMAT(4H 75 ,10(2X,E10.4))	FUNCT202
119 FORMAT(4H 76 ,10(2X,E10.4))	FUNCT203
120 FORMAT(4H 77 ,10(2X,E10.4))	FUNCT204
117 FORMAT(5H 7MA ,E10.4,6H ZWMA ,E10.4,6H EMAS ,E10.4,	FUNCT205
" 7H ZZWMA ,E10.4,7H ZWEMA ,E10.4,7H ZZWMA ,E10.4,	FUNCT206
" 7H E2MAZ ,E10.4)	FUNCT207
111 CONTINUE	FUNCT208
RETURN	FUNCT209
END	FUNCT210
SUBROUTINE COMPUT(X)	COMPUT 2
DIMENSION X(6)	COMPUT 3
REAL KAR,KPI	COMPUT 4
REAL NL,MASS,MA	COMPUT 5
C	COMPUT 6
COMMON /SHIP/ MASS,CINT,QA,CE,CE2,CE3,DMU,EDMU,E2DMU,E3DMU,BF,BMM,	COMPUT 7
NL,FL,IA,E(120)	COMPUT 8
COMMON /CONST/ NCG,ECG,PI,DPR,KPD,GRAVITY,RMU,NUM,MA(120),CD,TA,	COMPUT 9
H(120),BETA,MW(120),TZ,DMAG,w,XD,T,XP,M,IT,	COMPUT 10
1 DELTAS,TX,EST(120),KAR,MMA(120),TEST(120),	COMPUT 11
N(120),PHALF	COMPUT 12
COMMON/OUT/NPRINT,NPLOT,END	COMPUT 13
COMMON /TEAMS/ T1,T2,T3,T4,T5,T6,T7,T8	COMPUT 14
COMMON /WAVE1/ ZMA,ZWMA,EMAS,ZZWMA,ZWEMA,ZZWMA,E2MAZ,	COMPUT 15
1 ZWOUT(120),ASINPT(120,20),ACUSPT(120,20),CAB, SX6	COMPUT 16
COMMON /TEST/ VMA	COMPUT 17
C	COMPUT 18
C * * * * * SEE PAGES 7,8, AND 9 OF NOTES	COMPUT 19

C		COMPUT20
	CX6 = COS(X(6))	COMPUT21
	SX6 = SIN(X(6))	COMPUT22
	PIH = PI/2.0	COMPUT23
	KPI = KAH*PI	COMPUT24
	CONS1 = CX6	COMPUT25
	CONS2 = (KPI*RHO*PIH/TA)/CX6	COMPUT26
	CONS3 = CX6*SX6	COMPUT27
	CONS4 = CX6*CX6	COMPUT28
	TERM1 = X(1)*CX6	COMPUT29
	TERM2 = X(2)*SX6	COMPUT30
	UVNUM = (X(1)*CX6 - (X(2) - ZWDOT(NUM))*SX6)*	COMPUT31
	(X(1)*SX6 - X(3)*E(NUM) + (X(2) - ZWDOT(NUM))*CX6)	COMPUT32
C		COMPUT33
	ZMA = ZMA*X(3)*SX6	COMPUT34
	ZZWMA = ZZWMA*X(3)*SX6	COMPUT35
	ZWMA = ZWMA*CONS1	COMPUT36
	EMAS = EMAS*CONS1	COMPUT37
	DMU = DMU*CONS2	COMPUT38
	EDMU = EDMU*CONS2	COMPUT39
	CE = CE*CU*RHO	COMPUT40
	CE2 = CE2*CD*RHO	COMPUT41
	E2DMU = E2DMU*CONS3	COMPUT42
	E3DMU = E3DMU*CONS3	COMPUT43
	ZWEMA = ZWEMA*CONS4	COMPUT44
	ZZWMA = ZZWMA*CONS4	COMPUT45
C		COMPUT46
20	T1 = QA*X(3)*(TERM1-TERM2)	COMPUT47
	T1 = T1 + ZZWMA - EMAS	COMPUT48
	T2 = EDMU	COMPUT49
	T3 = CE2	COMPUT50
	T4 = MA(NUM)*E(NUM)*UVNUM + E2MAZ + E3DMU - ZZWMA + BMM	COMPUT51
	NL = T1 + T2 + T3 + T4 + BMM	COMPUT52
	T5 = MASS*X(3)*(TERM2-TERM1)	COMPUT53
	T5 = T5 + ZWMA - ZMA	COMPUT54
	T6 = -DMU	COMPUT55
	T7 = -CE	COMPUT56
	T8 = -MA(NUM)*UVNUM - E2DMU + ZWEMA	COMPUT57
	BF = BF/CX6	COMPUT58
C		COMPUT59
	FL = T5 + T6 + T7 + T8 - BF	COMPUT60
C		COMPUT61
	IF(NPRINT.LT.3)GO TO 30	COMPUT62
25	CONTINUE	COMPUT63
	WRITE(6,10)NL,FL	COMPUT64
10	FORMAT(" NL = ",E12.6," FL = ",E12.6)	COMPUT65
30	RETURN	COMPUT66
	END	COMPUT67
	SUBROUTINE INPUT	INPUT 2
C	***** DEFINITION OF INPUT VARIABLES	INPUT 3
C	XA = INITIAL TIME	INPUT 4
C	XE = FINAL TIME	INPUT 5
C	HMIN = MINIMUM STEP SIZE	INPUT 6
C	HMAX = MAXIMUM STEP SIZE	INPUT 7
C	EPSE = RELATIVE ERROR CRITERION USED FOR VALUES OF Y GT A	INPUT 8
C	EPS = ERROR CRITERION IN KUTMER	INPUT 9
C	A = ABSOLUTE ERROR CRITERIA USED IN KUTMER	INPUT 10
C	NPRINT = 1 FINAL PRINTOUT	INPUT 11
C	= 2 MATRIX INVERSE MATRIX, F COLUMN MATRIX, AND KUTMER	INPUT 12

C	RESULTS	INPUT 13
C	= 3 INTEGRAL VALUES	INPUT 14
C	= 4 CALCULATED VALUES--CONSTANT FOR GIVEN INPUT VALUES	INPUT 15
C	NPLOT = 0 NO PLOT	INPUT 16
C	= 1 PRINTER PLOT	INPUT 17
C	END = NUMBER OF RUNS	INPUT 18
C		INPUT 19
C	M = MASS OF CRAFT	INPUT 20
C	W = WEIGHT OF CRAFT	INPUT 21
C	TZ = THRUST COMPONENT IN Z DIRECTION	INPUT 22
C	TX = THRUST COMPONENT IN X DIRECTION	INPUT 23
C	XECG = DISTANCE FROM CG TO CENTER OF PRESSURE FOR NORMAL FORCE	INPUT 24
C	XP = MOMENT ARM OF PROPELLER THRUST	INPUT 25
C	XD = DISTANCE FROM CG TO CENTER OF PRESSURE FOR DRAG FORCE	INPUT 26
C	KAI(I) = ADDED MASS COEFFICIENT	INPUT 27
C	AN ARRAY GIVEN THE VALUE KAR WHICH IS READ IN	INPUT 28
C	BM(I) = BEAM AT FREE SURFACE OR AT CHINE	INPUT 29
C	URAG = FRICTION DRAG	INPUT 30
C	K = WAVE NUMBER	INPUT 31
C	RO = WAVE HEIGHT	INPUT 32
C	NU = WAVE SLOPE	INPUT 33
C	NUM = NUMBER OF STATIONS	INPUT 34
C	BL = BOAT LENGTH	INPUT 35
C	LAMBDA = WAVE LENGTH	INPUT 36
C	RG = RADIUS OF GENERATION IN FEET	INPUT 37
C	T = PROPELLED THRUST IN LBS	INPUT 38
C	GAMMA = PROPELLER THRUST ANGLE IN DEGREES	INPUT 39
C	DELTAS = STATION SPACING IN FEET	INPUT 40
C	ECG = LONGITUDINAL CENTER OF GRAVITY	INPUT 41
C	NCG = VERTICAL CG	INPUT 42
C	BETA(I) = DEAD RISE	INPUT 43
C	NO(I) = HEIGHT OF MEAN BUTTOK	INPUT 44
C	RHO = DENSITY OF WATER	INPUT 45
C	GRAVITY = GRAVITY FT/SEC**2	INPUT 46
C	DPR = DEGREES PER Radian	INPUT 47
C	RPD = RADIAN PER DEGREE	INPUT 48
C	PI = 3.14159	INPUT 49
C	EST(I) = STATION POSITION	INPUT 50
C	START = START TIME OF THE RAMP FUNCTION FOR SEA WAVE	INPUT 51
C	RISE = DURATION OF THE RISE FROM ZERO TO ONE OF THE RAMP	INPUT 52
C		INPUT 53
C	***** IC OPTIONS	INPUT 54
C		INPUT 55
C	IC(1) = 1 USE WAVE Z DISTANCE IN COMPUTING LIFT COMPONENT	INPUT 56
C	OF NL AND FL	INPUT 57
C		INPUT 58
C		INPUT 59
C	REAL IT,K,M,MA,MMA,NU,N,NCG,NO,MASS,NL,IA,KAR	INPUT 60
C	INTEGER END	INPUT 61
C		INPUT 62
C		INPUT 63
C	COMMON /CONST/ NCG,ECG,PI,DPR,RPD,GRAVITY,RHO,NUM,MA(120),CD,TA,	INPUT 64
C	B(120),BETA,MW(120),TZ,URAG,W,XD,T,XP,M,IT,	INPUT 65
C	DELTAS,TA,EST(120),KAR,MMA(120),TEST(120),	INPUT 66
C	N(120),PHALF	INPUT 67
C	COMMON /SHIP/ MASS,CINT,QA,CE,CE2,CE3,DMU,EDMU,E2DMU,E3DMU,BF,BMM,	INPUT 68
C	NL,FL,IA,E(120)	INPUT 69
C	COMMON /IN/ HM(120),BI(120),VELIN	INPUT 70
C	COMMON /IN2/ NC(120),XA,XE,MMA,MMIN,A(6),EPSL(6)	INPUT 71

	COMMON/OUT/NPRINT,NPLOT,END	INPUT 72
	COMMON /ACCEL/ XACCL,BWACL,CGACL,BL	INPUT 73
C	NAMelist/MSP/A,NPRINT,NPLOT,END,W,HL,TZ,IX,XECG,XP,XD,	INPUT 74
	DRAG,RG,T,GAMMA,ECG,NCG,KAR,NUM,BETA,EST	INPUT 75
	,XA,XE,HMIN,HMAX,EPS,VELIN	INPUT 76
C		INPUT 77
	DATA A /.01,.0001,.00001,.1,.0001,.00001/	INPUT 78
	DATA NPRINT,NPLOT,END/1,1,1/	INPUT 79
	DATA W,HL,TZ,IX,XECG,XP,XD,DRAG,RG,T,GAMMA,	INPUT 80
	ECG,NCG,KAR /16.,3.75,0.0,0.0,0.0,2.0,0.0,	INPUT 81
	2.325,0.0,1.0/	INPUT 82
	DATA NUM,BETA,EST /77,20.0,	INPUT 83
	0.0000,.03125,.06250,.09375,.12500,.15625,.18750,.21875,	INPUT 84
	.25000,.28125,.31250,.34375,.37500,.40625,.43750,.46875,	INPUT 85
	.50000,.53125,.56250,.59375,.62500,.65625,.68750,.71875,	INPUT 86
	.75000,.78125,.81250,.84375,.87500,.90625,.93750,.96875,1.000,	INPUT 87
	1.06250,1.12500,1.18750,1.25000,1.3125,1.37500,1.4375,	INPUT 88
	1.500,1.5625,1.625,1.6875,1.75,1.8125,1.875,1.9375,2.0,	INPUT 89
	2.0625,2.125,2.1875,2.25,2.3125,2.375,2.4375,2.5,2.5625,2.625,	INPUT 90
	2.6875,2.75,2.8125,2.875,2.9375,3.0,3.0625,3.125,3.1875,	INPUT 91
	3.2500,3.3125,3.375,3.4375,3.5,3.5625,3.625,3.6875,3.75 /	INPUT 92
	DATA XA,XE,HMIN,HMAX,EPS /0.0,20.0,.025,.1,.15/	INPUT 93
	DATA VELIN /19.62/	INPUT 94
C		INPUT 95
C	***** READ IN AND WRITE OUT KUTHER PARAMETERS AND PROGRAM	INPUT 96
C	OPTIONS	INPUT 97
	HEAD(5,MSP)	INPUT 98
	WRITE(6,MSP)	INPUT 99
	DO 10 I=1,5	INPUT 100
	10 EPS(I) = FPS	INPUT 101
C		INPUT 102
C	***** SET UP CONSTANTS	INPUT 103
	PI = 3.141592653589	INPUT 104
	GRAVITY=32.18	INPUT 105
	UPR=57.24577951308	INPUT 106
	RPD=.017453292519	INPUT 107
	IF (EST(NUM).LT.BL) STOP J	INPUT 108
C		INPUT 109
C	COMPUTE NU AND HM ARRAYS	INPUT 110
C	THIS IS FOR SPECIAL BOX FORM ONLY. CHANGE PROGRAM	INPUT 111
C	THRU STATEMENT 32 FOR NEW BOX SHAPE	INPUT 112
C		INPUT 113
	DO 32 I=1,NUM	INPUT 114
	IF(EST(I).GE.0.75) GO TO 30	INPUT 115
	NU(I)=-0.46875*(1.0-SQRT(EST(I)/0.375-(EST(I)/0.75)**2.0))	INPUT 116
	HM(I)=.375*SQRT(1.0-(EST(I)/.75-1.0)**2.0)	INPUT 117
	GO TO 32	INPUT 118
	30 NU(I)=0.0	INPUT 119
	HM(I) = 0.375	INPUT 120
	32 CONTINUE	INPUT 121
C	*****COMPUTE CONSTANTS AND INITIALIZE ARRAYS	INPUT 122
	M=W/GRAVITY	INPUT 123
	RHO=1.99	INPUT 124
	IT=M*RG*RG	INPUT 125
	PHALF = (PI/2.)*RHO	INPUT 126
C		INPUT 127
	BETA = BETA*RMU	INPUT 128
	CU = COS(BETA)	INPUT 129
		INPUT 130

```

      TA = TAN(BETA)
      DO 60 I=1,NUM
      E(I) = ECG-EST(I)
      N(I) = NCG.NO(I)
      MMAX(I) = KAR*PHALF*BM(I)*BM(I)
      TEST(I) = (2.*M(I)*TA)/PI
60  CONTINUE
      END=END+1
      RETURN
      END
      SUBROUTINE PLUTER (FX,XA,MMAX,IB,IMT)

C
C  INPUT:
C      FX      A TWO DIMENSIONAL ARRAY CONTAINING PITCH AND
C              HEAVE VALUES AT EACH TIME STEP
C      XA      INITIAL TIME
C      MMAX     TIME INTERVAL, MTIME*MMAX = INTERVAL BETWEEN
C              FX VALUES
C      IB      NUMBER OF FX VALUES
C
      REAL IT,K,LAMBDA,M,MA,MMAX,N,NCG
      INTEGER ENI

      DIMENSION FX(2,2000),FMIN(2),FMAX(2),NVAR(2)

      COMMON /CONST/ NCG,ECG,PI,DPH,NPD,GRAVITY,RHO,NUM,MA(120),CD,TA,
      .             B(120),BETA,MW(120),TZ,DRAG,X,XD,T,XP,M,IT,
      .             DELTA,TX,EST(120),KAR,MMAX(120),TEST(120),
      .             N(120),PHALF
      COMMON/OUT/NPRINT,NPLOT,END

C
C  * * * * * SET UP VALUES FOR PLOT AND CREATE PLOT
      NFUN=2
C  * * * * * SET UP MIN AND MAX LIMITS FOR PLOT
      FMIN(1)=FX(1,1)
      FMIN(2)=FX(2,1)
      FMAX(1)=FX(1,1)
      FMAX(2)=FX(2,1)

C
      DO 200 I=1,IB
      IF(FX(1,I).LT.FMIN(1))FMIN(1)=FX(1,I)
      IF(FX(1,I).GT.FMAX(1))FMAX(1)=FX(1,I)
      IF(FX(2,I).LT.FMIN(2))FMIN(2)=FX(2,I)
      IF(FX(2,I).GT.FMAX(2))FMAX(2)=FX(2,I)
200  CONTINUE

C
      800 CONTINUE
      NVAR(1)=10H HEAVE
      NVAR(2)=10H PITCH
      N1=2
      X0=XA
      DELX = MMAX
      IF(NPLOT.EQ.1)CALL PLOT2(FX,FMIN,FMAX,NVAR,NFUN,N1,IB,X0,DELX)
      RETURN
      END
      SUBROUTINE TRAP (F,UX,NPTS,ANS)

C
C  INPUT:

```

```

INPUT131
INPUT132
INPUT133
INPUT134
INPUT135
INPUT136
INPUT137
INPUT138
INPUT139
INPUT140
PLOTEN 2
PLOTEN 3
PLOTEN 4
PLOTEN 5
PLOTEN 6
PLOTEN 7
PLOTEN 8
PLOTEN 9
PLOTEN10
PLOTEN11
PLOTEN12
PLOTEN13
PLOTEN14
PLOTEN15
PLOTEN16
PLOTEN17
PLOTEN18
PLOTEN19
PLOTEN20
PLOTEN21
PLOTEN22
PLOTEN23
PLOTEN24
PLOTEN25
PLOTEN26
PLOTEN27
PLOTEN28
PLOTEN29
PLOTEN30
PLOTEN31
PLOTEN32
PLOTEN33
PLOTEN34
PLOTEN35
PLOTEN36
PLOTEN37
PLOTEN38
PLOTEN39
PLOTEN40
PLOTEN41
PLOTEN42
PLOTEN43
PLOTEN44
PLOTEN45
PLOTEN46
PLOTEN47
TRAP 2
TRAP 3
TRAP 4

```

C	F	ARRAY OF FUNCTIONAL VALUES OF THE INTEGRAND	TRAP	5
C	DX	THE X INTERVAL BETWEEN VALUES	TRAP	6
C	NPTS	THE NUMBER OF VALUES GIVEN	TRAP	7
C	OUTPUT:		TRAP	8
C	ANS	THE VALUE OF THE INTEGRAL	TRAP	9
C			TRAP	10
	DIMENSION F(NPTS)		TRAP	11
	ANS=0.0		TRAP	12
	IF(NPTS.LT.2)GO TO 999		TRAP	13
	DO 1 I=1,NPTS		TRAP	14
	1 ANS=ANS+F(I)		TRAP	15
	ANS=DX*(ANS-0.5*(F(1)+F(NPTS)))		TRAP	16
	999 CONTINUE		TRAP	17
	RETURN		TRAP	18
	END		TRAP	19
	FUNCTION RMP(T,START,RISE)		RMP	2
C	***** THIS FUNCTION IS USED TO GRADUALLY IMPLIMENT THE WAVE		RMP	3
C			RMP	4
C	T	CURRENT TIME	RMP	5
C	START	TIME TO START HAMP FROM 0.0 TO 1.0	RMP	6
C	RISE	THE LENGTH OF THE RISE FROM 0.0 TO 1.0	RMP	7
C			RMP	8
	M=0.0		RMP	9
	IF(T.LT.START)GO TO 99		RMP	10
	IF(RISE.EQ.0.0)GO TO 80		RMP	11
	TOP=T-START		RMP	12
	M=1.0		RMP	13
	IF(TOP.LT.RISE)M=TOP/RISE		RMP	14
	GO TO 99		RMP	15
	80 M=1.		RMP	16
	IF(T.EQ.START)M=0.5		RMP	17
	99 RMP=M		RMP	18
	RETURN		RMP	19
	END		RMP	20
	SUBROUTINE SEAWAY		SEAWAY	2
	REAL K		SEAWAY	3
	COMMON /WAVE2/ W0(20),K(20),C(20),RW0(20),RW02(20),RK(20),		SEAWAY	4
	RO(20),HKK(20),M,PHS(20)		SEAWAY	5
	COMMON /CONST/ NCG,ECG,PI,DPR,RPU,GRAVITY,RHU,NUM,MA(120),CD,TA,		SEAWAY	6
	B(120),BETA,MW(120),TZ,DHAG,W,XD,T,XP,M,IT,		SEAWAY	7
	1 DELTA5,TX,EST(120),KAR,MMA(120),TEST(120),		SEAWAY	8
	N(120),PHALF		SEAWAY	9
	HEAD (5,80) M		SEAWAY	10
	WRITE (6,97) M		SEAWAY	11
	80 FORMAT (MF10.4)		SEAWAY	12
	90 FORMAT (274 SIGNIFICANT WAVE HEIGHT = ,E10.4,2X,5H FEET,//)		SEAWAY	13
	MROOT = SQRT(M)		SEAWAY	14
	WN = 2.276/MROOT		SEAWAY	15
	W0(1) = .7454**WN		SEAWAY	16
	W0(2) = WN		SEAWAY	17
	W0(3) = 1.163**WN		SEAWAY	18
	W0(4) = 1.4036**WN		SEAWAY	19
	W0(5) = 1.6015**WN		SEAWAY	20
	W0(6) = 1.7953**WN		SEAWAY	21
	W0(7) = 2.0035**WN		SEAWAY	22
	W0(8) = 2.1941**WN		SEAWAY	23
	W0(9) = 2.3419**WN		SEAWAY	24
	W0(10) = 2.612**WN		SEAWAY	25
	RO(1) = 0.1364**M		SEAWAY	26

```

NO(2) = 0.1861*M
NO(3) = 0.1657*M
NO(4) = 0.1302*M
NO(5) = 0.0999*M
NO(6) = 0.0771*M
NO(7) = 0.0604*M
NO(8) = 0.0482*M
NO(9) = 0.0390*M
NO(10) = 0.0626*M
PHS(1) = .005
PHS(2) = 2.41
PHS(3) = 5.20
PHS(4) = 4.00
PHS(5) = 0.00
PHS(6) = 1.27
PHS(7) = 3.11
PHS(8) = 2.92
PHS(9) = 3.55
PHS(10) = 0.70
DO 50 J=1,10
K(J) = NO(J)*NO(J)/GRAVITY
C(J) = GRAVITY/NO(J)
RWO(J) = RO(J)*NO(J)
RWO2(J) = RWO(J)*NO(J)
RWK(J) = RWO(J)*K(J)
RK(J) = NO(J)*K(J)
50 CONTINUE
RETURN
END
SUBROUTINE TABLE
COMMON/SINE/POINT(1000)
UX = .003141592654
X = 1.570796327
DO 100 J=1,501
POINT(J) = SIN(X)
K = 1002-J
POINT(K) = -POINT(J)
X = X+DX
100 CONTINUE
RETURN
END

```

```

SEAWAY27
SEAWAY28
SEAWAY29
SEAWAY30
SEAWAY31
SEAWAY32
SEAWAY33
SEAWAY34
SEAWAY35
SEAWAY36
SEAWAY37
SEAWAY38
SEAWAY39
SEAWAY40
SEAWAY41
SEAWAY42
SEAWAY43
SEAWAY44
SEAWAY45
SEAWAY46
SEAWAY47
SEAWAY48
SEAWAY49
SEAWAY50
SEAWAY51
SEAWAY52
SEAWAY53
SEAWAY54
SEAWAY55
TABLE 2
TABLE 3
TABLE 4
TABLE 5
TABLE 6
TABLE 7
TABLE 8
TABLE 9
TABLE 10
TABLE 11
TABLE 12
TABLE 13

```

DTNSRDC ISSUES THREE TYPES OF REPORTS

- 1. DTNSRDC REPORTS, A FORMAL SERIES, CONTAIN INFORMATION OF PERMANENT TECHNICAL VALUE. THEY CARRY A CONSECUTIVE NUMERICAL IDENTIFICATION REGARDLESS OF THEIR CLASSIFICATION OR THE ORIGINATING DEPARTMENT.**
- 2. DEPARTMENTAL REPORTS, A SEMIFORMAL SERIES, CONTAIN INFORMATION OF A PRELIMINARY, TEMPORARY, OR PROPRIETARY NATURE OR OF LIMITED INTEREST OR SIGNIFICANCE. THEY CARRY A DEPARTMENTAL ALPHANUMERICAL IDENTIFICATION.**
- 3. TECHNICAL MEMORANDA, AN INFORMAL SERIES, CONTAIN TECHNICAL DOCUMENTATION OF LIMITED USE AND INTEREST. THEY ARE PRIMARILY WORKING PAPERS INTENDED FOR INTERNAL USE. THEY CARRY AN IDENTIFYING NUMBER WHICH INDICATES THEIR TYPE AND THE NUMERICAL CODE OF THE ORIGINATING DEPARTMENT. ANY DISTRIBUTION OUTSIDE DTNSRDC MUST BE APPROVED BY THE HEAD OF THE ORIGINATING DEPARTMENT ON A CASE-BY-CASE BASIS.**