SSC-333

ADVANCED METHODS FOR SHIP MOTION AND WAVE LOAD PREDICTION

SHIP STRUCTURE COMMITTEE

1990

SHIP STRUCTURE COMMITTEE

THE SHIP STRUCTURE COMMITTEE is constituted to prosecute a research program to improve the hull structure of ships and other marine structures by an extension of knowledge pertaining to design, materials and methods of construction.

RADM J. D. Sipes, USCG, (Chairman)
Chief, Office of Marine Safety,
Security and Environmental Protection
U. S. Coast Guard

Mr. Alexander Malakhoff
Director, Structural Integrity
Subgroup (SEA 55Y)
Naval Sea Systems Command

Dr. Donald Liu
Senior Vice President
American Bureau of Shipping

Mr. H. T. Haller
Associate Administrator for Shipbuilding and Ship Operations
Maritime Administration

Mr. Thomas W. Allen
Engineering Officer (N7)
Military Sealift Command

CDR Michael K. Parmelee, USCG,
Secretary, Ship Structure Committee
U. S. Coast Guard

CONTRACTING OFFICER TECHNICAL REPRESENTATIVES

Mr. William J. Siekierka
SEA 55Y3
Naval Sea Systems Command

Mr. Greg D. Woods
SEA 55Y3
Naval Sea Systems Command

SHIP STRUCTURE SUBCOMMITTEE

THE SHIP STRUCTURE SUBCOMMITTEE acts for the Ship Structure Committee on technical matters by providing technical coordinating for the determination of goals and objectives of the program, and by evaluating and interpreting the results in terms of structural design, construction and operation.

U. S. COAST GUARD

Dr. John S. Spencer (Chairman)
CAPT T. E. Thompson
Mr. David L. Motherway
CDR Mark E. Noll

NAVAL SEA SYSTEMS COMMAND

Mr. Robert A. Sielski
Mr. Charles L. Null
Mr. W. Thomas Packard
Mr. Allen H. Engle

MARITIME ADMINISTRATION

Mr. Frederick Seibold
Mr. Norman O. Hammer
Mr. Chao H. Lin
Dr. Walter M. Maclean

MILITARY SEALIFT COMMAND

Mr. Glenn M. Ashe
Mr. Michael W. Touma
Mr. Albert J. Attermeyer
Mr. Jeffery E. Beach

AMERICAN BUREAU OF SHIPPING

Mr. John F. Conlon
Mr. Stephen G. Arntson
Mr. William M. Hanzalek
Mr. Philip G. Rynn

SHIP STRUCTURE SUBCOMMITTEE LIAISON MEMBERS

U. S. COAST GUARD ACADEMY

LT Bruce Mustain

U. S. MERCHANT MARINE ACADEMY

Dr. C. B. Kim

U. S. NAVAL ACADEMY

Dr. Ramswar Bhattacharyya

STATE UNIVERSITY OF NEW YORK MARITIME COLLEGE

Dr. W. R. Porter

WELDING RESEARCH COUNCIL

Dr. Glen W. Oyler

NATIONAL ACADEMY OF SCIENCES MARINE BOARD

Mr. Alexander B. Stavovy

NATIONAL ACADEMY OF SCIENCES COMMITTEE ON MARINE STRUCTURES

Mr. Stanley G. Stiansen

SOCIETY OF NAVAL ARCHITECTS AND MARINE ENGINEERS-HYDRODYNAMICS COMMITTEE

Dr. William Sandberg

AMERICAN IRON AND STEEL INSTITUTE

Mr. Alexander D. Wilson

Member Agencies:

United States Coast Guard
Naval Sea Systems Command
Maritime Administration
American Bureau of Shipping
Military Sealift Command

Ship Structure Committee

An Interagency Advisory Committee
Dedicated to the Improvement of Marine Structures

Address Correspondence to:

Secretary, Ship Structure Committee
U.S. Coast Guard (G-MTH)
2100 Second Street S.W.
Washington, D.C. 20593-0001
PH: (202) 267-0003
FAX: (202) 267-0025

August 2, 1990

SSC-333
SR-1277

ADVANCED METHODS FOR SHIP MOTION AND WAVE LOAD PREDICTION

Advanced numerical methods are needed by ship designers to better predict and simulate ship motions and hull girder loads. Complex structural loading problems such as bottom slamming, bow flare impact, and green water on deck cannot be satisfactorily analyzed using linear strip theory.

This report provides a numerical method for predicting transient three-dimensional hydrodynamic pressures and resulting loads. This work is based on an initial level of investigation and development, and will require further testing, validation, and refinement of the numerical methods and computer programs.

J. D. SIPES
Rear Admiral, U. S. Coast Guard
Chairman, Ship Structure Committee

(THIS PAGE INTENTIONALLY LEFT BLANK)

Technical Report Documentation Page

1. Report No. SSC-333
2. Government Accession No.
3. Recipient's Catalog No.

4. Title and Subtitle

A TIME DOMAIN SIMULATION METHOD FOR SHIP MOTION AND LOAD PREDICTION

5. Report Date February 1985

6. Performing Organization Code SHIP STRUCTURE COMMITTEE

7. Author(s) JAMES C. OLIVER

8. Performing Organization Report No. 80-083-001

9. Performing Organization Name and Address

GIANNOTTI & ASSOCIATES INC.
703 Giddings Avenue, Suite U-3
Annapolis, Maryland 21401

10. Work Unit No. (TRAIS)

11. Contract or Grant No. DTCG23-80-C-20032

12. Sponsoring Agency Name and Address

U.S. COAST GUARD
WASHINGTON, D.C. 20593

13. Type of Report and Period Covered

FINAL REPORT
10/80 - 2/85

14. Sponsoring Agency Code G-M

15. Supplementary Notes

CONTRACT MONITORED BY: SHIP RESEARCH COMMITTEE
NATIONAL ACADEMY OF SCIENCES, WASHINGTON, D.C.

16. Abstract

A numerical method for the simulation of ship motions, hull girder loads and transient three-dimensional hydrodynamic pressure is developed. A time-step integration of the equations of motion is performed with sectional hydrodynamic forces computed using strip theory and buoyancy forces evaluated over the instantaneous immersed hull. In addition to hydrostatic and dynamic buoyancy forces, ship motion induced and wave diffraction forces, the method can also account for "flare" force and quadratic damping forces. The numerical model also has several features which may provide for more realistic simulations. Linear wave theory is assumed for the representation of the incident wave field, and regular and irregular, unidirectional or multidirectional wave systems can be generated.

A method for the prediction of transient three-dimensional hydrodynamic pressures extending the technique developed by R.B. Chapman is presented. Initial and boundary values for the pressure simulation are simultaneously generated from the solution of the equations of motion using sectional force computations. An option exists in the program to substitute the three-dimensional pressures for the two-dimensional hydrodynamic forces.

The work presented herein represents an initial level of investigation and development. The numerical methods and computer programs require further testing, validation and improvement.

17. Key Words

Ship Motions
Hydrodynamics
Simulation
Computer

18. Distribution Statement

19. Security Classif. (of this report) UNCLASSIFIED
20. Security Classif. (of this page) UNCLASSIFIED
21. No. of Pages 196
22. Price

Form DOT F 1700.7 (8-72)

Reproduction of completed page authorized

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

A numerical method for the simulation of ship motions, hull girder loads and transient three-dimensional hydrodynamic pressures is developed. A time-step integration of the equations of motion is performed with sectional hydrodynamic forces computed using strip theory and buoyancy forces evaluated over the instantaneous immersed hull. In addition to hydrostatic and dynamic buoyancy forces, ship motion induced and wave diffraction forces, the method can also account for "flare" force and quadratic damping forces. The numerical model has several features which may provide for more realistic simulations, including the ability to evaluate sectional hydrodynamic coefficients at the instantaneous sectional draft, and a crude scheme for adjusting the frequency at which sectional coefficients are evaluated for use in ship motion induced force calculations. Linear wave theory is assumed for the representation of the incident wave field, and regular and irregular, unidirectional or multidirectional wave systems can be generated.

A method for the prediction of transient three-dimensional hydrodynamic pressures extending the technique developed by R.B. Chapman is presented. Initial and boundary values for the pressure simulation are simultaneously generated from the solution of the equations of motion using sectional force computations. An option exists in the program to substitute the three-dimensional pressures for the two-dimensional hydrodynamic forces.

Computer programs have been developed to perform the computations -- the main simulator is called SSX. There are also two preprocessors -- HYDREX2 and HYDREX3. A Frank close-fit computation of sectional hydrodynamic coefficients is performed by HYDREX2. HYDREX3 performs computations to derive geometric data and several large arrays necessary for the transient pressure simulation.

The work presented herein represents an initial level of investigation and development. The numerical methods and computer programs require further testing, validation and improvement.

NOMENCLATURE

Equations of Motion and Strip Theory

A_{jk} = components of added mass matrix

B_{km} = transformation matrix for Eulerian angles

$BM_y\ (x_o)$ = vertical bending moment at X_o

$BM_z(x_o)$ = horizontal bending moment at X_o

F_j = external forces

I_{jk} = components of moment and product of inertia matrix

K = roll moment

M = pitch moment

M_{jk} = components of mass matrix

M_s = ship's mass

N = yaw moment

R_G = vector position of ship c.g. in space system

$SF_y\ (x_o)$ = vertical shear force at X_o

$SF_z(x_o)$ = horizontal shear force at X_o

NOMENCLATURE (Continued)

S_i, C_i = sine and cosine components of force fluctuation for wave component i, the relative magnitudes of which provide the proper phase relationship of all wave-related forces

T_j = external moments

$TM_x(x_o)$ = torsional moment at X_o

U = ship's speed

V_{G_k} = translational velocities of center of mass in space fixed system

= $\frac{d}{dt}(R_G)$

X = longitudinal force

Y = vertical force

Z = lateral force

$X_b^s \ldots M_b^s$ = hydrostatic force components of the Froude-Krylov force

$X_{3D} \ldots M_{3D}$ = force components calculated from the three-dimensional pressure computations

a_i = amplitude of i^{th} wave component

a_{22} = heave added mass

a_{33} = sway added mass

NOMENCLATURE (Continued)

a_{34} = section roll added mass moment of inertia due to sway motion

a_{44} = sectional roll added mass moment of inertia

b_{22} = heave damping coefficient

b_{33} = sway damping coefficient

b_{34} = sectional roll damping moment coefficient due to sway motion

b_{44} = sectional roll damping coefficient

b = Froude-Krylov force subscript

$\frac{dY_h}{dx}, \frac{dZ_h}{dx}, \frac{dK_h}{dx}$ = sectional ship motion-induced hydrodynamic force in heave sway and roll

$\frac{dY_w}{dx}, \frac{dZ_w}{dx}, \frac{dK_w}{dx}$ = sectional diffraction force in heave, sway and roll

f = "flare" force subscript

h = ship motion-induced force subscript

h_{cg} = vertical distance from waterline to ship c.g. (+ up).

i, N = wave index, total number of wave components

i_x = local section's mass moment of inertia

$\vec{k}_i$ = horizontal vector wave number

NOMENCLATURE (Continued)

$\vec{k}_i \vec{x}_i$ = $xk_i \cos\beta - zk_i \sin\beta$

k_i' = projection of the wave number onto the x-axis of ship

q = quadratic damping force subscript

t = total force

w = wave diffraction force subscript

x = longitudinal distance to section from ship's c.g.

$\vec{x}_i$ = horizontal cartesian coordinates

$\bar{y}$ = local section's center of gravity relative to ship c.g. (positive up).

y_G , z_G = heave and sway velocity of center of mass of ship

β = direction of wave propogation

ω_i = circular frequency

ε_i = phase angle

ω_k = rotational velocities = $B_{km} \frac{d\alpha_m}{dt}$

ψ, $\dot{\psi}$ = Eulerian pitch angle and velocity in ship system

ϕ, $\dot{\phi}$ = Eulerian yaw angle and velocity in ship system

NOMENCLATURE (Continued)

$\theta, \dot{\theta}$ = Eulerian roll angle and roll velocity in ship system

$\xi_i, \dot{\xi}_i$ = average vertical water velocity, acceleration

$\zeta_i, \dot{\zeta}_i$ = average lateral water velocity, acceleration

$\kappa_i, \dot{\kappa}_i$ = average roll water velocity, acceleration

$\xi_i, \dot{\xi}_i$ = average vertical water velocity, acceleration

$\zeta_i, \dot{\zeta}_i$ = average lateral water velocity, acceleration

$\kappa_i, \dot{\kappa}_i$ = average roll water velocity, acceleration

μ = vertical sectional added mass gradient

Three-dimensional Analysis

A_{nm}, B_{nm} = complex functions of wave numbers and time

A^*_{nm}, B^*_{nm} = complex conjugates

$A_i(t)$ = area of panel i

E_{ij} = source - velocity influence matrix

L_X, L_Y, l_x, l_y, T = parameters for computing spectral free-surface representation wave numbers and spacing

$\hat{P}_{ij}$ = source - potential influence matrix

NOMENCLATURE (Continued)

p_T, p_B, p_{FS}, p_A = pressure at panel center, total, body, free-surface, ambient

X_{ij} = source - x velocity influence matrix

a_T, a_B, a_{FS}, a_A = acceleration vector

k_{nm} = $k_{nm}^2 = kx_n^2 + ky_m^2$

kx_n, ky_m = wave numbers in x and y; e.g., $\frac{\omega_n^2}{g}$

$s_i(t)$ = strength of source at panel i

v_T, v_B, v_{FS}, v_A = velocities vector at panel centers, total, body-induced, free-surface and ambient wave

v_X, v_Z, a_X, a_Z = x and z components of ambient wave-induced velocity and acceleration at panel center

ϕ_T = total time varying potential

ϕ_B = instantaneous effect on the body

ϕ_{FS} = free surface disturbance due to previously radiated and diffracted waves

ϕ_A = ambient wave potential

σ_i = source density of panel i

Δkx_n, Δky_m = wave number spacing; e.g., $\Delta kx_n = kx_{n+1} - kx_n$

NOMENCLATURE (Continued)

ω_{nm} $= \omega_{nm}^{2} = g\,[k{x_n}^{2} + k{y_m}^{2}]^{1/2}$

$\Delta A_{nm}^{BODY}, \Delta A_{nm}^{BODY*}, \Delta B_{nm}^{BODY}, \Delta B_{nm}^{BODY*}$

= influence of body on free surface spectral component

(THIS PAGE INTENTIONALLY LEFT BLANK)

1.0 INTRODUCTION

While the established ship motion and loading methods currently used in design practice remain indispensable, there is need for a more advanced tool that can carry the ship designer a step closer toward a fully rational ship design procedure. Although the application of linear strip theory in the frequency and probabilistic domains has met with remarkable success, there are a number of practical seakeeping and structural loading problems not satisfactorily analyzed by such techniques. Green water, bow flare impact and bottom slamming are particularly obvious examples. Reported large differences in dynamic sagging and hogging moments in some type of ships is a consequence of nonlinear loading [1,2]. The accurate prediction of rolling amplitudes requires the use of a nonlinear restoring moment at a large angle and nonlinear viscous damping estimates. These examples tend to highlight the generally nonlinear character of the ship motion and loads problem. An equally important part of the problem is its three-dimensionality. For example, the prediction of hydrodynamic pressures on the ship's hull below the mean waterline using linear two- and three-dimensional methods is not at a consistent level of accuracy. Clearly, there are a number of areas where improvements can be made in ship motion and loading prediction.

Based on an extensive review of the reported advanced methods in body-wave hydrodynamics and ship motion/loads prediction, a time-domain simulation of ship motion, hull girder loads and hydrodynamic pressures has been developed. Strip theory with certain nonlinear enhancements is used to predict sectional hydrodynamic forces for motion and hull girder loading prediction, and a three-dimensional source distribution of the hull, combined with a spectral representation of the free surface is used for the evaluation of hydrodynamic hull pressures. Linear wave theory is used and some optional numerical features have been included that may improve the predictions in some cases. The time-domain numerical simulation technique using strip theory-derived forces expands upon the capsizing simulation work of Oakley, Paulling, Wood and others [3]. In that method, the hydrodynamic forces from ship motion and diffraction are assumed to be small due to low encounter frequency in following and quartering waves -- allowing for approximate formulations. On the other hand, the Froude-Krylov forces (hydrostatic and dynamic buoyancy) are computed very accurately for the instantaneous immersed hull. In the present model, this accurate approach to Froude-Krylov forces has been used, while the ship motion and diffraction-induced forces are formulated according to standard strip theory equations [4].

Three-dimensional hull pressures and resultant loads are computed by a separate method based on Chapman's techniques for the simulation of arbitrary linearized motions of a floating body [5,6]. The present method extends Chapman's formulation to include ambient wave field and associated diffraction effects. Although the three-dimensional technique can be generalized to include a time-dependent hull shape, the computational method has been developed here in as simple a form as possible. Thus, as presently established, the hull is specified by surface panels describing the hull portion directly beneath the static-waterline only. The computational method can be easily expanded to include the entire hull; however, at present the computational effort to run a program with such an extension would be prohibitively expensive.

Several optional features have been incorporated into the numerical simulation procedures. Sectional added mass and damping coefficients may be evaluated for the instantaneous average sectional draft at each time step. Also the frequencies at which the added mass and damping coefficients are evaluated (for ship motion-induced forces) can be selected to correspond to either (a) the peak frequencies of motion response spectra, or (b) allowed to vary based on the measured response of the vessel during the previous two cycles of motion. Additionally, a vertical velocity-squared term is included in the strip formulations to partially account for the effect of flare impact loading. Finally, the simulation can incorporate the effects of velocity-squared damping using quadratic damping coefficients selected by the user.

It should be pointed out that because the majority of the testing and validation efforts described in this report use regular waves, the reader may tend to form the impression that the method is limited to regular waves. This is not the case. The incident wave system may be irregular and multidirectional. It is constructed from the superposition of regular waves, the amplitudes, frequencies, directions and phases of which may be specified by the user. They can thus be selected to represent the time history associated with any given sea spectrum.

This document (Volume I) describes the method and presents some test and validation results. Several computer programs were written to implement the approach and a user's manual (Volume II) is provided as a companion report.

1.1 BACKGROUND

In recent years, a number of impressive advances have been made in body-wave hydrodynamics. In particular, the fast-growing field of numerical ship hydrodynamics is producing a wide range of promising techniques with the potential to handle difficult nonlinear three-dimensional hydrodynamic phenomena. It is enlightening to follow the progress made by various investigators as presented in the First, Second and Third International Conferences on Numerical Ship Hydrodynamics in 1975, 1977 and 1981, respectively. Some of those efforts and others pertinent to the ship motion problem will be noted.

Efforts towards quasi-analytical solutions to Laplace's equation using singularity distributions for the ship motion problem have been reported by Chapman [6] and Chang [7]. M.S. Chang represents the ship hull by source/dipole distributions and solves the fluid domain and free surface equations using a mean hull boundary condition. It is a frequency-domain solution and appears to be an excellent way to find the correct linear wave excitation load distribution along the ship for any heading, speed and wave length. Solutions in the frequency domain assume that the response of the body to excitations of varied frequency content may be represented as the linear summation of responses to a series of discrete frequencies. Thus the equations of motion need to be solved only for the component frequencies. In contrast, in a time-domain solution, a time-history of the excitation is decomposed into a series of time steps with the time history of the response constructed by re-solving the equations of motion at each time step. R.B. Chapman's method is a time-domain computation somewhat analagous to Chang's approach. However, it does not use Green's functions representing solutions of the free surface equations for oscillating singularities moving with uniform speed as does Chang. Simple sources and their image are used to represent the hull. Although the problem is linearized, the method is theoretically capable of accounting for a time-dependent hull shape. It thus provides the flexibility of arbitrary motions.

Another effort using singularity distributions includes that of Pettersen [8]. Pettersen's work is particulary interesting. As the problem is now formulated, he solves the integral equation at zero frequency for three-dimensional sources over the bow and stern and two-dimensional sources using strip theory for the middlebody. The approach has applications for ship maneuvering analysis but some work is being done for $\omega \neq 0$. Methods of finite difference potential field solutions to a free surface and wave problems have been reported by von Kerczek [9] and McCormick and J.W. Thomas [10], to mention only a few.

Bai [11] has continued to develop a finite element method applied to a three-dimensional harmonic ship-motion problem. The computational domain is reduced to a very small local domain using an eight-node three-dimensional element. The work is a direct extension of the earlier work by Bai and Yeung [12]. Other studies of finite element method applied to time harmonic ship motion problems have been presented by Chen and Mei [13], Seto and Yamamoto [14], Yue, Chen and Mei [15] and Chowdbury [16]. Emphasis in these cases is on the solution of regular harmonic motions, although finite element techniques have also been applied to transient problems by J.T. Beale [17] and C. Licht [18]. A. Jami and M. Lenoir [19] present a method for coupling finite elements and an integral representation.

Limited improvements to strip theory in the area of wave excitation and diffraction effects have been reported by S.O. Skjordal [20], and Troesch [21].

Research in the area of incompressible viscous flow has yielded several interesting techniques. Bourianoff and Penumalli [22] have developed a three-dimensional time-dependent ship motion simulation using a method they call the Inertial Marker Particle (IMP) technique. It is a Eulerian approach. A Eulerian approach uses a space-fixed reference system through which the fluid flows. In the alternate Lagrangian description, the reference frame moves with the fluid. A Lagrangian approach to transient-free surface flows was reported by Fritts and Boris [23]. Nichols and Hirt [24] report on progress being made with variants and improvements of the original Marker-and-Cell method in free-surface computations.

In the area of ship motion and loads prediction, the more advanced methods incorporate a time-simulation approach using some form of strip theory. The primary motivation for the development of this class of simulation programs is to account for nonlinearities from various sources. Paulling, Oakley and Wood [3] were interested in the capsizing problem with emphasis on accurate computation of buoyancy forces. Chang [25] developed a motion simulator emphasizing non-rigid body structural response to regular head seas. Meyerhoff and Schlacter [26] emphasized the dynamic whipping response due to bottom and flare impact in unidirectional irregular head seas. Borresen and

Telsgard [27] emphasized the nonlinear effects of bow flare on motions and rigid body hull girder loads. In the last three references, the relative motion hypothesis is employed in each case, and the frequencies at which the hydrodynamic coefficients are evaluated are usually the frequency of encounter (ω_e) or infinite frequency (ω_∞). Also, the methods in the last three references treat only pitch and heave in unidirectional regular waves.

In view of these reported successes with the time-step solution to the equations of motion and the method's general capability of incorporating loads of a nonlinear character, this approach will be used as the basis of the present model. It is also intended that some type of three-dimensional hydrodynamic formulation cast as an initial value problem would be integrated with the motion simulator from which initial and boundary values would be specified. As noted before, a finite difference solution to the Euler equation was initially chosen based on the reported successes of Bourianoff and Penumalli [22]. The technique developed to apply this finite difference solution is described in [28]. The fundamental difficulty with using this technique for ship motion and transient-free surface simulation was the conflicting demands of numerical stability vs. realistic fluid dynamics. The finite difference Euler equation solution method was abandoned after it became apparent that some of the numerical difficulties could not be overcome within the scope of this project. Alternative methods were investigated, and the technique developed by R.B. Chapman reported in [6] was selected because of its ability to predict three-dimensional flow and handle arbitrary motions and time-dependent hull shapes. Its principal disadvantage is the fact that it is computationally demanding, requiring long run times.* Nevertheless, in view of the ever-increasing capabilities of modern computational hardware, the use of the Chapman method and others like it should become practical in the near future.

---

* Prescribed motion of a 120-panel body requires 7 CPU minutes per time step (0.1 sec $< t < 0.25$ sec) on a DEC VAX 11/750 or roughly 20-25 CPU secs on a so-called supercomputer (e.g., CRAY or CDC STAR).

2.0 THEORETICAL MODEL

2.1 GENERAL FORMULATION OF EQUATIONS OF MOTION

The equations of motion for the ship can be derived from Newton's second law and formulated as four first-order ordinary differential equations, the solution of which gives translational and rotational displacements and velocities. Solution is accomplished in a time-step fashion using a fourth-order Runge-Kutta integration scheme. The external forces and moments called for in the equations of motion are calculated using an extension of basic strip theory to derive ship motion-induced and diffraction forces. Froude-Krylov forces are computed using a volume integral approach over the instantaneous immersed hull. The incident wave system is represented by one or more linear waves of arbitrary amplitude, direction, frequency and phase relationship.

The ship is considered to be advancing with forward speed with an arbitrary heading in a system of sinusoidal waves of various amplitudes, frequencies and directions. Let (x,y,z) be a Cartesian coordinate system fixed in space. Let (x',y',z') be a Cartesian coordinate system fixed with respect to the ship, whose axes coincide with the principal axes of inertia of the ship and whose origin corresponds to the ship's center of mass. The direction of the axes are illustrated in Figure 1a.

The position of the ship in space is completely characterized by the position of the ship's mass center and the angular displacement of the ship. Eulerian angles will be used to describe angular displacement. Figure 1b shows the convention for the Eulerian angles.

The motion of the ship in an inertial frame of reference, according to Newton's second law, can be formulated using subscript notation as

$$\frac{d}{dt} M_{jk} V_{G_k} = F_j \qquad (j = 1,2,3) \tag{1}$$

$$\frac{d}{dt} I_{jk} \omega_k = T_j \qquad (j = 1,2,3) \tag{2}$$

in which

M_{jk} = components of mass matrix

I_{jk} = components of moment and product of inertia matrix

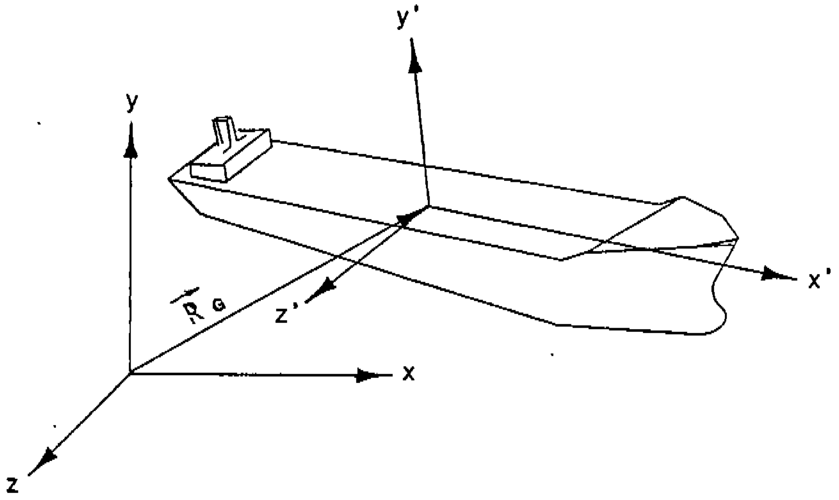

Figure 1a   Axes for Solution of Equations of Motion

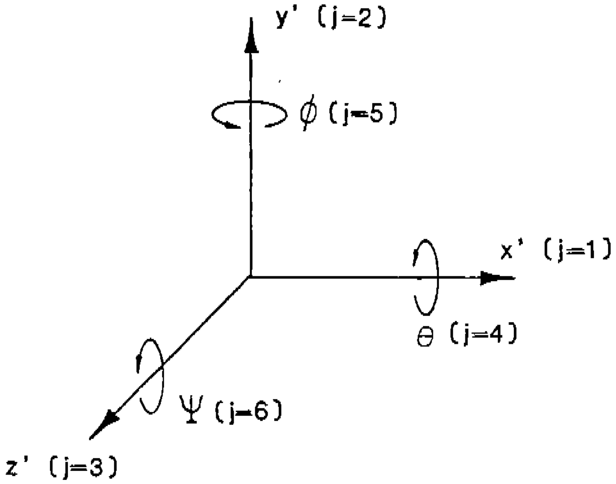

Figure 1b   Sign Convention for Translational and Rotational Motions

V_{G_k} = translational velocities of center of mass in space fixed system

$$= \frac{d'}{dt} R_G$$

ω_k = rotational velocities

$$= B_{km} \frac{d\alpha_m}{dt}$$

and

$$B_{km} = \begin{bmatrix} 1 & \sin\psi & \theta \\ 0 & \cos\theta\ \cos\psi & \sin\theta \\ 0 & -\sin\theta\ \cos\psi & \cos\theta \end{bmatrix}$$

α_m = { θ , ϕ , ψ }

θ = Eulerian roll angle

ϕ = Eulerian yaw angle

ψ = Eulerian pitch angle

F_j = external forces

T_j = external moments

In the ship coordinate system, the left side of equation (2) takes the following form:

$$\frac{d}{dt} (I_{jk}\ \omega_k) = I_{jk} \frac{d}{dt} \omega_k + \omega_k \times I_{jk}\ \omega_k \qquad (3)$$

This is the Eulerian equation of motion whose application makes possible the use of constant moments and products of inertia. In the inertial frame of reference, these values are continuously changing.

Equations (1), (2) and (3) can be manipulated to yield the following two sets of two first-order ordinary differential equations:

$$\frac{d}{dt} X_{G_k} = V_{G_k} \qquad (k = 1,2,3) \tag{4}$$

$$\frac{d}{dt} V_{G_k} = (M_{jk})^{-1} F_j \qquad (k = 1,2,3) \tag{5}$$

$$\frac{d}{dt} \alpha_m = (B_{km})^{-1} \omega_k \qquad (m = 1,2,3) \tag{6}$$

$$\frac{d}{dt} \omega_k = (I_{jk})^{-1} \{T_j - \omega_k \times I_{jk} \omega_k\} \qquad (k = 1,2,3) \tag{7}$$

Assuming that the ship has lateral symmetry, the mass matrix is given by

$$M_{jk} = \begin{bmatrix} M_s & 0 & 0 \\ 0 & M_s & 0 \\ 0 & 0 & M_s \end{bmatrix} + A_{jk} \qquad (j = 1,2,3) \tag{8}$$

where A_{jk} represents added mass. The ship's mass is shown as M_s within the brackets.

The moment and product of inertia matrix is given as

$$I_{jk} = \begin{bmatrix} I_{11} & I_{21} & I_{31} \\ I_{12} & I_{22} & I_{32} \\ I_{13} & I_{23} & I_{33} \end{bmatrix} + A_{mn} \qquad \begin{matrix} (j = 1,2,3) \\ (m = 4,5,6) \end{matrix} \tag{9}$$

where A_{mn} represents added mass coefficients, and the values within the brackets represent ship moment and products of inertia.

If we can evaluate the external forces (F_j) and moments (T_j) required by expressions (5) and (7), we can solve for the ship translational (V_G) and rotational (ω) velocities. These, in turn, are used in equations (4) and (6) to yield the ship translational (X_G) and rotational (α) displacements. The integration of these four first-order ordinary differential equations is performed numerically with the fourth-order Runge-Kutta scheme.

Thus, the prediction of ship motions by this method then becomes a matter of predicting the external forces and moments. The various types of forces and moments which comprise the total force and moment used in equations (5) and (7) are outlined in the following section.

2.2 EVALUATION OF EXTERNAL FORCES AND MOMENTS

The basic method used to evaluate the external forces will be based on strip theory. This approach is used to evaluate forces throughout the simulation of ship motions and loads. An additional and important feature is the prediction of pressures on the hull using a three-dimensional hull source representation. The resultant force on the ship from these pressures can be substituted for the hydrodynamic forces predicted from the strip approach into the equations of motion for a selected interval of time. The interval of time for which this force substitution may be made must be limited in order to minimize the considerable computing effort associated with a three-dimensional solution in the time domain. Figure 2 depicts this procedure. It should be emphasized that the formulations for the two methods (strip and three-dimensional) are separate and distinct and each can provide force information for the solution of the equations of motion.

A brief outline of the force components that comprise the total force used in the ship motion equations follows. We first examine the strip method followed by a brief discussion of the three-dimensional method.

2.2.1 Strip Approach Forces

Following strip theory, we will assume that the hydrodynamic force acting on the ship can be decomposed into three parts:

(a) Froude-Krylov Force - The resultant force from the pressure in the waves which acts on the hull surface with the assumption that the pressure field is not affected by the presence of the ship.

(b) Diffraction Force - A correction to the Froude-Krylov force that accounts for the disturbance of the ambient wave field by the presence of the ship.

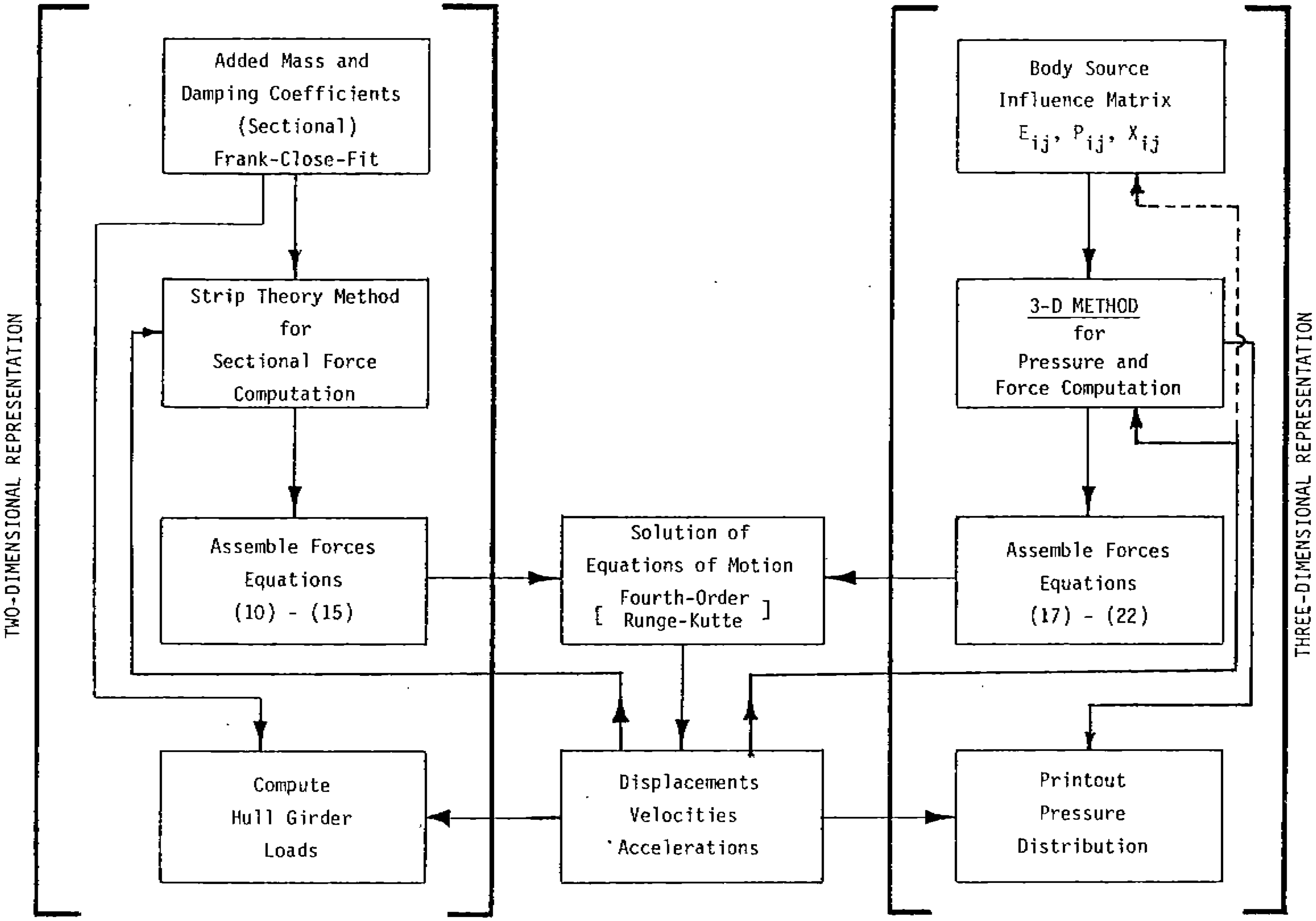

Figure 2 Diagram of Computational Scheme

(c) Ship Motion-Induced Force - The force resulting from the motion of the ship with the assumption that it undergoes this motion in calm water.

The evaluation of the Froude-Krylov forces is obtained by integrating the pressure existing in the undisturbed waves over the wetted surface of the hull. The evaluation of the diffraction force and ship motion-induced force will generally be according to the elementary strip theory (see Raff [4]). Linear wave theory is employed and it will be assumed that the Froude-Krylov pressure and diffraction force evaluated in an irregular wave system is equivalent to the summation of the pressures and forces from the individual waves that comprise the irregular system.

The calculation of added mass and damping coefficients used in the evaluation of diffraction and ship motion-induced forces is by the Frank close-fit method. Because these coefficients are frequency dependent, a problem exists as to which frequency to use in the ship motion-induced force calculations since the ship motion may be irregular (non-harmonic). One approach taken by several investigators [26, 27] is to select a constant set of coefficients associated with one frequency.

We propose two other approaches and briefly examine the effect of each in Section 3.3. The first and simpler method is for a user-specified fixed value associated with the highest peak of the response spectra obtained from a frequency domain analysis or sea spectrum. The second approach is based on a scheme that samples the previous two zero-crossing periods of motion in sway, heave and roll. The two values are averaged, and three characteristic frequencies are derived for evaluation of the sectional hydrodynamic coefficients. The sway-roll or roll-sway coefficients are (arbitrarily) linked to the sway characteristic frequency.

In frequency domain strip theory formulations, the added mass and damping coefficients are evaluated at the still waterline. In the present formulation, at each time step, two-dimensional hydrodynamic coefficients can be evaluated for each section at either the mean still water draft or at the instantaneous depth of immersion. When it is desired that the hydrodynamic coefficients be draft dependent, the coefficients are obtained by linear interpolation from an array of six values corresponding to six drafts covering a range from near the baseline to the deck edge. Since the hydrodynamic coefficients are also frequency dependent, they are selected using linear interpolation from an array of twelve values associated with a range of twelve frequencies specified by the user. Thus, when the two-dimensional sectional coefficients are to be both draft and frequency dependent, two-dimensional linear interpolation is performed.

phenomena as bottom slamming, flare slamming, and the shipping of green water; as well as wind forces, mooring forces, rudder and propeller forces, viscous and appendage related damping, etc. The incorporation of methods to evaluate most of these forces will be left as the subject of subsequent investigations. We will, however, include here a flare force prediction method and also the means to incorporate a quadratic damping force. For flare impact force, we will use the approach described by Meyerhoff [26].

2.2.2 Three-Dimensional Pressure and Forces

The ship hull is modeled by quadrilateral panels. At the center of each panel, the pressure is evaluated, and resultant forces may be obtained. The pressure computed by this method includes the Froude-Krylov pressure (excluding the hydrostatic part), the diffraction pressure and the ship motion-induced pressure. Further description of this method will be left to Section 2.5, where it is discussed in detail.

2.2.3 Total Force Equations

The total force equations are given using the notation shown below. The force for each degree of freedom is indicated by:

X = longitudinal force
Y = vertical force
Z = lateral force
K = roll moment
N = yaw moment
M = pitch moment

Subscripts to these force symbols specify the type of force according to:

b = Froude-Krylov
h = ship motion-induced
w = wave diffraction
f = "flare" force
q = quadratic damping force
t = total force

Two sets of equations will be given; the first showing the force components normally used during the simulation, and the second showing the force components when the three-dimensional method is invoked. The external forces (F) and moments (M) used in the equations of motion (5) and (7) can be assembled from various sectional force components according to:

$$F_1 = \int \frac{X_b}{dx}\, dx \qquad (10)$$

$$F_2 = \int \frac{dY_h}{dx}\, dx + \int \frac{dY_w}{dx} + \int \frac{dY_f}{dx}\, dx + \int \frac{dY_b}{dx}\, dx + Y_q \tag{11}$$

$$F_3 = \int \frac{dZ_h}{dx}\, dx + \int \frac{dZ_w}{dx}\, dx + \int \frac{dZ_b}{dx}\, dx + Z_q \tag{12}$$

$$T_1 = \int \frac{dK_h}{dx}\, dx + \int \frac{dK_w}{dx}\, dx + \int \frac{dK_b}{dx}\, dx + K_q \tag{13}$$

$$T_2 = \int x \frac{dZ_h}{dx}\, dx + \int x \frac{dZ_w}{dx}\, dx + \int \frac{dN_b}{dx}\, dx + N_q \tag{14}$$

$$T_3 = \int x \frac{dY_h}{dx}\, dx + \int x \frac{dY_w}{dx}\, dx + \int x \frac{dY_f}{dx}\, dx + \int \frac{dM_b}{dx}\, dx + M_q \tag{15}$$

All integrations are performed over the length of the ship. The strip theory equations used to evaluate the sectional forces are given in Section 2.4. As will be seen in that section, in the equations for the sectional ship motion-induced forces there is a part related to acceleration. These acceleration-related forces cannot be computed directly at any particular time step, since the acceleration values at that time are not *a priori* known. Consequently, the acceleration force terms in those equations are extracted and transferred to the mass moment and product of inertia matrices as the second terms in the right-hand side of expressions (8) and (9) given earlier. These matrices are assembled according to the following expressions:

$$\begin{aligned}
A_{22} &= \int a_{22}\, dx \\
A_{33} &= \int a_{33}\, dx \\
A_{43} &= \int a_{34}\, dx \\
A_{44} &= \int a_{44}\, dx \\
A_{45} &= \int x a_{34}\, dx
\end{aligned} \tag{16}$$

$$A_{53} = \int x a_{33} \, dx$$

$$A_{55} = \int x^2 a_{33} \, dx$$

$$A_{62} = \int x a_{22} \, dx$$

$$A_{66} = \int x^2 a_{22} \, dx$$

with the remaining values of $A_{jk} = 0$. All integrations are over the length of the ship.

When the three-dimensional pressure calculational procedure is run concurrently with the stripwise force calculations, three-dimensional pressure forces may be substituted for the stripwise hydrodynamic forces. During such an interval, the force expressions (10) - (15) would instead appear as:

$$X_T = X_{3D} + X_b^s + X_q \tag{17}$$

$$Y_T = Y_{3D} + Y_b^s + \int \frac{dY_f}{dx} dx + Y_q \tag{18}$$

$$Z_T = Z_{3D} + Z_b^s + Y_q \tag{19}$$

$$K_T = K_{3D} + K_b^s + K_q \tag{20}$$

$$N_T = N_{3D} + N_b^s + N_q \tag{21}$$

$$M_T = M_{3D} + M_b^s + \int x \frac{dY_f}{dx} dx + M_q \tag{22}$$

where X_b^s ... M_b^s are the hydrostatic force components of the Froude-Krylov forces, and X_{3D} ... M_{3D} are the force components calculated from the three-dimensional pressure computations.

This completes the formulation of the equations of motion and the force equations. The next several sections will present the computation of the Froude-Krylov force, the sectional diffraction and ship motion-induced forces, the flare impact force, and the three-dimensional pressures and resultant forces.

2.3 FROUDE-KRYLOV FORCES

We will assume that the elevation of the sea surface $\eta(\vec{x},t)$ can be described as the superposition of a number of sinusoids of the form:

$$\eta(\vec{x},t) = \sum_{i=1}^{N} a_i \cos(\vec{k}_i \vec{x} - \omega_i t + \varepsilon_i) = \sum_{i=1}^{N} \eta_i \tag{23}$$

The expression for hydrostatic and dynamic pressure can be derived from Bernoulli's equation and the definition of velocity potential for the incident wave field. If we include only the linear terms, and consider only deep water, the expression for pressure may be given as

$$p(t,x,y,z) = -\rho g \left[y - \sum_{i=1}^{N} e^{k_i y} \eta_i \right] \tag{24}$$

The Froude-Krylov force may be obtained by integrating the pressure over the immersed surface of the ship. This surface integral may be replaced by a volume integral,* so that the expressions for total Froude-Krylov forces and moments can be given by the integral of the pressure gradients. Sectional forces can be evaluated according to:

$$\frac{dX_b}{dx} = -\iint \frac{\partial p}{\partial x}\, dydz$$

---

* Gauss Theorem -

$$\mathbf{F} = \iint \mathbf{n} p \, dA = \iiint \nabla p \, dV$$

$$\frac{dY_b}{dx} = - \iint \frac{\partial p}{\partial y} \, dydz$$

$$\frac{dZ_b}{dx} = - \iint \frac{\partial p}{\partial z} \, dydz$$

$$\frac{dK_b}{dx} = \iint \left(z \frac{\partial p}{\partial y} - y \frac{\partial p}{\partial z}\right) dydz$$

$$\frac{dN_b}{dx} = \iint \left(x \frac{\partial p}{\partial z} - z \frac{\partial p}{\partial x}\right) dydz \qquad (25)$$

$$\frac{dM_b}{dx} = \iint \left(y \frac{\partial p}{\partial x} - x \frac{\partial p}{\partial y}\right) dydz$$

For each section, vertical integration extends to the wave surface rather than the mean water level. This fact necessitates some sort of approximation to evaluate pressures in the free surface zone since linear wave theory implies infinitesimal amplitudes which do not extend measurably above or below the mean water level.

From a review of the relevant literature, there appear to be three basic approaches for approximating the velocities and pressures in the free surface zone:

(a) Unmodified Formula - Use the same formulas (for pressure) above mean sea level that apply below sea level. (Eq. 26 with no restrictions on y.)

(b) Stretching Correction - Stretch the still water level in the formula to the sea surface.

(c) Hydrostatic - Assume the pressure in the wave above the mean water level is hydrostatic.

There is not sufficient experimental data to support the use of one scheme over the next. The CAPSIZE program [29] uses the unmodified formula. There are also some limited experimental data obtained by Chakrabarti [30] to indicate that a hydrostatic pressure assumption is accurate in regular waves. If we select the hydrostatic assumption, the pressure equation may then be written as

$$p\,(t,x,y,z) = -\rho g \left[y - \sum_{i=1}^{N} e^{k_i y} \eta_i \right] \qquad y \leq 0 \qquad (26)$$

$$p\ (t,x,y,z) = -\rho g\ (y - \sum_{i=1}^{N} \eta_i) \qquad y > o \qquad (27)$$

The differences in ship response as a result of using the unmodified pressure formula above the mean waterline were investigated. Comparisons for heave and pitch response of the SL-7 containership in 15-foot waves indicated that the differences were negligible. In any event, both methods are approximations. Their use, however, is necessitated by the employment of linear wave theory, and we therefore cannot escape the use of such simplifications. The computer code, as presently written, allows the user to select either approximation.

2.4 SECTIONAL HYDRODYNAMIC FORCES

Following elementary strip theory, local sectional hydrodynamic forces induced by the ship's motion and by diffraction force for heave, sway and roll are given in this section. The equations given below are equivalent to the basic equations given by Raff [4] in SSC-230. Flare impact force and quadratic roll damping are also discussed.

Vertical Sectional Ship Motion-Induced Force

$$\begin{aligned} \frac{dY_h}{dx} &= -\ a_{22}\ (\ddot{y}_G + x\ddot{\psi} - 2U\dot{\psi}\ ) \\ &\quad -\ b_{22}\ (\dot{y}_G + x\dot{\psi} - U\psi\ ) \qquad (28) \\ &\quad +\ U\frac{da_{22}}{dx}\ (\dot{y}_G + x\dot{\psi} - U\psi\ ) \end{aligned}$$

Vertical Sectional Diffraction Force

$$\frac{dy_w}{dx} = \sum_{i=1}^{N} [\ \dot{\xi}_i\ a_{22} + \xi_i (b_{22} - U\frac{da_{22}}{dx})\ ] \qquad (29)$$

Lateral Sectional Ship Motion-Induced Force

$$\begin{aligned}\frac{dZ_h}{dx} = & - a_{33}\ (\ddot{z} - x\ \ddot{\phi} + 2U\ \dot{\phi}\ ) \\ & + (U \frac{da_{33}}{dx} - b_{33})\ (\dot{z} - x\dot{\phi} + U\phi) \\ & + (a_{43} + h_{cg}\ a_{33})\ \ddot{\theta} \\ & + [b_{43} + h_{cg}\ b_{33} - U\ (\frac{da_{43}}{dx} + h_{cg} \frac{da_{33}}{dx}\ )]\ \dot{\theta}\end{aligned} \tag{30}$$

Lateral Sectional Diffraction Force

$$\begin{aligned}\frac{dZ_w}{dx} = & \sum_{i=1}^{N} \{\ [\dot{\zeta}_i a_{33} - \zeta_i (U \frac{da_{33}}{dx} - b_{33})] \\ & - [\dot{\kappa}_i h_{cg} a_{34} - \kappa_i h_{cg} (U \frac{da_{34}}{dx} - b_{34})]\}\end{aligned} \tag{31}$$

Roll Sectional Ship Motion-Induced Moment

$$\begin{aligned}\frac{dK_h}{dx} = & - (a_{44} + h_{cg}\ a_{34})\ \ddot{\theta} \\ & - [\ b_{44} + h_{cg}\ b_{34} - U\ (\frac{da_{44}}{dx} + h_{cg} \frac{da_{34}}{dx}\ )\ ]\ \dot{\theta} \\ & + a_{34}\ (\ddot{z} - x\ddot{\phi} + 2U\ \dot{\phi}) \\ & + (b_{34} - U \frac{da_{34}}{dx})(\dot{z} - x\ \dot{\phi} + U\ \phi) - h_{cg} \frac{dZ_h}{dx}\end{aligned} \tag{32}$$

Roll Sectional Diffraction Force

$$\frac{dK_w}{dx} = \sum_{i=1}^{N} \{\dot{\kappa}_i a_{44} - \kappa_i (U \frac{da_{44}}{dx} - b_{44})\} - [\dot{\zeta}_i h_{cg} a_{34} - \zeta_i (U \frac{da_{34}}{dx} + b_{34})]\} - h_{cg} \frac{dZ_w}{dx} . \tag{33}$$

Following the procedure used in the program CAPSIZE [29], each ship section is defined as a closed polygon made up of straight line segments between offsets. The integrals of vertical, horizontal and "roll" wave velocities and acccelerations, as well as pressure gradients over each section, are evaluated exactly at each time step. Longitudinal integrations of sectional diffraction forces and Froude-Krylov forces (excluding hydrostatic component) are performed assuming the sectional forces are functions of the following form:

$$f(x) = \sum_{i=1}^{N} [C_i \cos (k_i' x) + S_i \sin (k_i' x)]$$

where

S_i, C_i = sine and cosine components of force fluctuation for wave component i, the relative magnitudes of which provide the proper phase relationship of all wave-related forces.

k_i' = projection of the wave number onto the x-axis of ship.

Longitudinal integrations are performed assuming C_i and S_i vary linearly along the ship's length.

Flare Force

The approach for evaluating flare force is actually an extension of the strip theory expressions. Its incorporation into the present methodology serves to highlight the capability of the time domain method to account for the effects of the above-waterline hull on motions and loads. Furthermore, earlier work by Kaplan [31], Borreson and Tellsgaard [27], and Meyerhoff [22] have already provided the basic method to compute flare forces.

Following the approach described by Meyerhoff, the time derivative of the relative wave elevation for any strip may be given as

$$\dot{\eta}_r = \frac{\omega_o}{\omega_e}\dot{\eta} - \dot{y}_G - x\dot{\psi} + \psi U \tag{34}$$

where $\dot{\eta}$ denotes the time-derivative of the wave elevation relative to the ship-fixed coordinate system, ω_o is wave frequency and ω_e is frequency of encounter. The convective derivative of relative velocity is accounted for by the factor ω_o / ω_e. In the derivation of the expression for hydrodynamic inertia forces in the strip method, it can be shown that a nonlinear term results which represents an impact term. It contains the square of the relative velocity, and according to Meyerhoff [26] appears as:

$$\frac{dY_f}{dx} = \mu\,(\dot{\eta}_r)^2$$

where

$$\begin{aligned} \mu &= \left\{ \frac{\partial a_{22}(\infty)}{\partial \eta_r} \right\} && \text{if } \dot{\eta}_r > 0 \\ &= 0 && \text{if } \dot{\eta}_r < 0 \end{aligned}$$

The limiting value of added mass at infinite frequency is used in accordance with the theory of normal-symmetric impact upon a calm water surface. (It should also be pointed out that this formulation does not account for diffracted free surface motion, and the name "Froude-Krylov flare force" might more accurately represent the effect.)

Quadratic Damping

The expressions for total forces and moments (10) - (15) and (17) - (22) each contain a term which represents a force or moment which is proportional to velocity-squared. The user of the computer program may specify, as input, his own values for quadratic damping coefficients. Accurate roll prediction often requires some quadratic roll damping. The damping moment associated with roll, for example, may be written in the form

$$K_q = B_{44}^{*}\,|\dot{\theta}|\,\dot{\theta}$$

where B_{44}^{*} is the quadratic roll damping coefficient for the ship, and $\dot{\theta}$ is roll angle velocity.

2.5 THREE-DIMENSIONAL DYNAMIC PRESSURES AND FORCES

2.5.1 Introduction

A numerical method for calculating the transient three-dimensional flow induced by the motion in waves of a floating body of arbitrary shape with forward speed is presented in this section. Dynamic pressures and the resultant forces and moments acting on the hull surface can be evaluated in the time domain using this technique. The solution is generated in terms of a source distribution representation of the body and a spectral representation of the free surface. The approach presented here expands upon the work of R.B. Chapman [5,6]. Although Chapman had provided the basic method to solve the problem of ship motions in waves, his formulation did not explicitly include the ambient wave field effects and associated diffraction flow.

Following Chapman, the general formulation is linearized, assuming that the individual potentials and their derivatives associated with forward speed, body motion, free surface disturbance caused by radiated and diffracted waves, and the ambient wave flow are each sufficiently small to ignore higher order terms and cross-products. The free surface equations are also linearized. The formulation partially accounts for the "exact" hull boundary by evaluating hydrodynamic pressures on the actual immersed portion of the body surface at each time step below the still waterline. Thus, the effect of large-amplitude motions is included to some extent.

It should be emphasized that this three-dimensional approach is a separate procedure quite distinct from the strip formulation and equations of motion solution techniques that form the basic framework. It is best viewed as a "parallel process," interfacing with the main solution procedures in the following ways:

(a) It requires, as input, the specification of body position and velocities at each time step.

(b) It provides as output forces and moments on the hull, as well as additional information regarding distributed dynamic pressures.

In the formulation that follows, for the convenience of the reader who wishes to refer to Chapman's work, we will use Chapman's coordinate system throughout Section 2.5.

2.5.2 Formulation

A Cartesian coordinate system is fixed in space with the x-axis in the direction of the ship's forward advance, positive forward; the z-axis positive downward and y-axis positive to starboard. The still mean water level corresponds to z=0. The free surface elevation is specified by (x,y,t) and the ship's hull is defined by $S(x,y,z,t) = 0$ for $z \geqslant 0$. If the fluid is assumed to be inviscid, incompressible and homogeneous and the flow irrotational, then the fluid velocity $\vec{v}\,(\vec{x}\,,t)$ can be represented by the gradient of the velocity potential $\Phi(\vec{x},t)$. If we linearize the free surface equations, the velocity potential must satisfy the following conditions:

$$\nabla^2 \Phi = 0 \qquad z > 0 \tag{35}$$

$$\frac{\partial \Phi}{\partial t} = -g\,\eta \qquad z = 0 \tag{36}$$

$$\frac{\partial \Phi}{\partial z} = \frac{\partial \eta}{\partial t} \qquad z = 0 \tag{37}$$

$$\frac{\partial \Phi}{\partial \vec{n}} - \vec{V}_s \cdot \vec{n} = 0 \qquad \text{- on } S(x,y,z,t) = 0 \tag{38}$$

where $\vec{V}_s$ is the velocity at any point on the hull surface S, and $\vec{n}$ is the unit normal to the boundary surface at the point, pointing outward. The velocity potential can be decomposed into two parts:

$$\Phi\,(x,y,z,t) = [-\,Ux + \phi_S\,(x,y,z) + \phi_T\,(x,y,z,t)] \tag{39}$$

where $-Ux + \phi_S$ is the steady part and U is the forward speed of the ship, and $\phi_T\,(x,y,z,t)$ is the time-varying potential. In the present formulation, we will neglect the steady part and consider only the time-varying potential.

Separating the time-varying potential into three parts, we obtain

$$\phi_T = \phi_B + \phi_{FS} + \phi_A \tag{40}$$

where ϕ_B represents the instantaneous effect of the body, ϕ_{FS} represents the existing free surface disturbance due to previously radiated and diffracted waves, and ϕ_A is the ambient wave potential.

The body potential will be represented by a distribution of simple sources over the wetted hull and its negative image. A spectral representation of the free surface potential will be used so that the diffracted and radiated wave fields are represented by two series, harmonic in space and time. The potential for the ambient wave system will be given according to linear theory for progressive waves. It should be noted that, although the method can be extended to include sources distributed over the entire body, the present procedure only distributes sources over the body below the mean waterline.

The objective of this analysis is to compute the hydrodynamic pressure on the hull, as determined from the potential field ϕ_T (x,y,z,t). The dynamic pressure can be obtained from Bernoulli's equation applied to ϕ_T (x,y,z,t). If the flow is assumed to be slow enough that the nonlinear term can be neglected, Bernoulli's equation is:

$$p_T = -\rho \frac{\partial \phi_T}{\partial t} = -\rho \frac{\partial}{\partial t} [\phi_B + \phi_{FS} + \phi_A] \quad (41)$$

By linearizing the problem, we can evaluate the total pressure at any time or point as the sum of the individual pressure components, as given by:

$$p_B (x,y,z,t) = -\rho \left[\frac{\partial \phi_B}{\partial t} - \vec{V}_s \cdot \vec{\nabla}\phi_B\right] \quad (42)$$

$$p_{FS} (x,y,z,t) = -\rho \frac{\partial \phi_{FS}}{\partial t} \quad (43)$$

$$p_A (x,y,z,t) = -\rho \frac{\partial \phi_A}{\partial t} \quad (44)$$

The convective derivative for the body pressure expression is included because the Bernoulli equation given here is for fixed points in space, yet the body potential is associated with the moving hull surface. We will assume, however, that products of the body velocity and potential gradients are negligible with the exception of $U \frac{\partial \phi_B}{\partial x}$. Equation (42) then becomes:

$$p_B(x,y,z,t) = -\rho \left[\frac{\partial\phi_B}{\partial t} - U \frac{\partial\phi_B}{\partial x} \right] . \qquad (45)$$

Prior to describing the individual potential representations in detail, it will be advantageous to first describe the numerical solution procedure on a step-by-step basis.

2.5.3 Solution Procedure

This is an initial value problem that ideally should start from a condition at rest. Prior to starting the simulation, the following arrays relating to the body source representation are calculated:

(a) E_{ij} - gives the normal velocity component at the center point of panel i induced by a uniform source distributed over panel j and its image. (As will be explained, the hull surface is modeled by quadrilateral panels; at the center of each is a simple source.)

(b) $\hat{P}_{ij}$ - gives the potential at the center of panel i induced by uniform source density of unit strength acting over panel j.

(c) X_{ij} - gives the x-direction velocity component at the center point of panel i induced by a uniform source distributed over panel j and its image.

The simulation progresses by a series of small time steps, according to the following sequence:

Step 1.

As we begin each time step, we will have already computed the velocities, accelerations and pressures induced by the free-surface disturbance and ambient wave field at the center of each panel. (These will be known from steps (5) and (6) of the previous time cycle.) The body velocity and acceleration will have been obtained from the solution to the equations of motion.

First, for time $t = t_n$, compute the resultant total normal velocity $(\vec{v}_T \cdot \vec{n})$ * at the center of each panel.* This resultant velocity represents the sum of the velocity from the body motion, the velocity induced by the existing free surface disturbance, and the velocity induced by the ambient wave field:

$$\vec{v}_T \cdot \vec{n} = (\vec{v}_B + \vec{v}_{FS} + \vec{v}_A) \cdot \vec{n} \text{ at panel centers.} \tag{46}$$

According to Chapman [6], it is more convenient to use accelerations instead of velocities. Equation (46) can then be expressed as:

$$\vec{a}_T \cdot \vec{n} = (\vec{a}_B + \vec{a}_{FS} - U \frac{\partial}{\partial x} \vec{v}_{FS} + a_A - U \frac{\partial}{\partial x} V_A) \cdot \vec{n} \text{ .} \tag{47}$$

(The convective derivatives are required in this case for a_{FS} and a_I because we are in a hull-fixed system when computing the resultant normal acceleration on the body.)

Step 2.

Knowing $\vec{a}_T$ from step (1) and having precalculated E_{ij}, determine the time derivative of the source strengths $\dot{\sigma}_j$ according to

$$\sum_{i=1}^{N_B} E_{ij} \dot{\sigma}_j = \vec{a}_T \cdot \vec{n} \qquad \text{at } t = t_n. \tag{48}$$

Step 3.

Calculate the body-induced component of pressure at the center point of each panel induced by the computed time derivative of the source strength in a hull fixed system according to

---

* In the equations that follow, it is understood that all normal vectors, areas, positions or coordinates on the hull are functions of time, even though they may not be so specified.

$$p_B = -\rho \frac{\partial \phi_B}{\partial t} + \rho U \frac{\partial \phi_B}{\partial x} \tag{49}$$

or

$$p_B = -\rho \sum_{j=1}^{N_B} \hat{P}_{ij} \dot{\sigma}_j (t_n) + \rho U X_{ij} \sigma_j (t_n) \tag{50}$$

Step 4.

Compute total pressure at each panel

$$p_T (t_n) = p_B (t_n) + p_{FS} (t_n) + p_A(t_n) .$$

Step 5.

Update the source strengths for this time step (t_n) according to

$$\sigma_i (t_n) \approx \sigma_i(t_n - \Delta t) + \Delta t \, \dot{\sigma}_i (t_n) \tag{51}$$

Step 6.

As described in Section 2.5.5, update the free surface representations. Modify the free surface to account for the effect of the hull source strengths on the free surface. Calculate acceleration induced by the free surface at the center of each panel $\vec{a}_{FS} (t_n + \Delta t)$ and the pressure induced by the free surface disturbance at the center of each panel $p_{FS} (t_n + \Delta t)$.

Step 7.

Update the ambient wave field. Calculate the acceleration $\vec{a}_A (t_n + \Delta t)$ and pressure $p_A (t_n + \Delta t)$ induced by the ambient wave field at the center of each panel.

Step 8.

Go to Step 1 with the new $\vec{a}_{FS}$, $\vec{a}_A$ and newly prescribed $\vec{a}_B$.

2.5.4 Body Representation

As shown in Figure 3, the hull is modeled by a set of N_B quadrilateral panels. A simple source and its image is distributed over each panel. The expression for the body potential can then be expressed as:

$$\phi_B\,(x,y,z) = \sum_{i=1}^{N_B} \sigma_i \iint_{S_i} G(x,y,z,x',y',z')\, dS' \qquad (53)$$

in which

$$G(x,y,z,x',y',z') = [(x-x')^2 + (y-y')^2 + (z-z')^2]^{-1/2} - [(x-x')^2 + (y-y')^2 + (z+z')^2]^{-1/2} \qquad (54)$$

σ_i is the strength on the i^{th} source panel

S_i is the i^{th} panel surface.

Panel source strengths are evaluated at each time step using a linear system of equations relating resulting total normal velocities $(\vec{v}_{T_i} \cdot \vec{n})$ at each panel center to the source strength σ_j at every other panel center and by the satisfaction of the hull boundary condition at the center of each panel. The exact method of Hess and Smith [32] is used here to determine the normal velocity component at the center point of panel i induced by a plane quadrilateral source element with a unit value of source density at the center point of panel j. A body coefficient scalar matrix E_{ij} is thus computed which can be used to determine unknown source densities when given panel center normal velocities, according to

$$\vec{v}_{T_i} \cdot \vec{n} = \sum_{j=1}^{N_B} E_{ij}\, \sigma_j \qquad (55)$$

As was shown in equation (46), the panel center total normal velocity is the sum of components from the hull motion and induced velocities from radiated and diffracted free surface disturbances and ambient wave field.

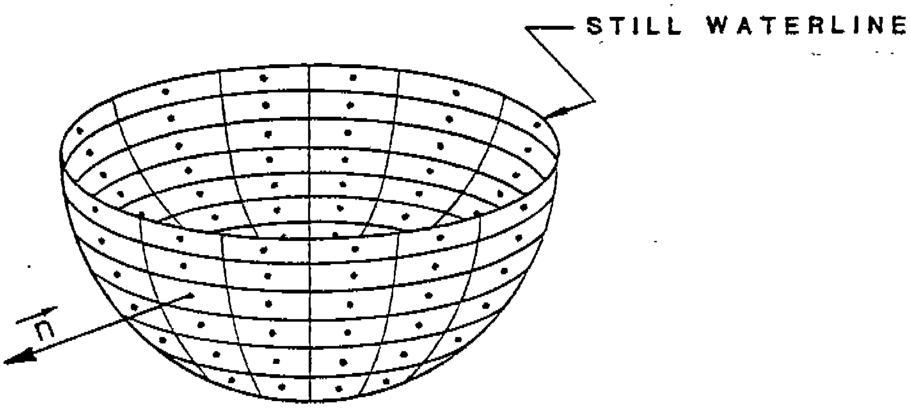

Figure 3 Quadrilateral panels and source representation of semisubmerged sphere.

Also, in Step 3 of the computational sequence, equations (49) and (50) show that the calculation of body-induced pressure at panel centers requires the use of two other arrays $\hat{P}_{ij}$ and X_{ij}. The X_{ij} array gives the x-direction velocity component at the center point of panel i induced by a uniform source (of unit strength) distributed over panel j and its image. It is thus very similar to the E_{ij} array, relating the x-component instead of the normal component of panel velocities. In practice, it is evaluated directly from the information used to evaluate E_{ij}.

The $\hat{P}_{ij}$ array gives the potential at the center of panel i induced by a uniform source (of unit strength) distributed over panel j and its image, or

$$\phi_{B_i} = \sum_{j=1}^{N_B} \hat{P}_{ij}\,\sigma_j \tag{56}$$

Looking back at equation (53) for a moment, it can be seen the $\hat{P}_{ij}$ can be computed by relating each panel center to every other panel center using the expression for G(x,y,z,x',y',z'), or in the indexed notation,

$$\hat{P}_{ij} = \iint_{S_i} dS_i \{ [(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2]^{-1/2} - [(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j + z_i)^2]^{-1/2} \} \tag{57}$$

These integrals are evaluated numerically by dividing the surface S_i into many small elements and evaluating the above integrand at the center of each smaller element.

2.5.5 Free Surface Representation

A spectral representation of the free surface will be used. Let the free surface elevation and velocity potential be represented by $\eta(x,y,t)$ and $\phi_{FS}(x,y,z,t)$, respectively. The elevation and potential satisfy the dynamic and kinematic linearized free-surface conditions at z = 0.

$$\frac{\partial}{\partial t}\,\phi(x,y,0,t) = -g\,\eta(x,y,t) \tag{58}$$

$$\frac{\partial \phi}{\partial z}\ \eta(x,y,t) = -\frac{\partial \eta}{\partial t}(x,y,o,t) \tag{59}$$

Now assume that both the elevation and deep water potential can be represented by the real part of a finite summation of harmonics in the following forms:

$$\eta\ (x,y,t) = \sum_{n=1}^{N_{kx}} \Delta kx_n \sum_{m=1}^{N_{ky}} \Delta\ ky_m$$

$$\left[A_{nm}\ (t)\ e^{i(kx_n \cdot x + ky_m \cdot y)} + A^*_{nm}\ (t)\ e^{i(kx_n \cdot x - ky_m \cdot y)}\right] \tag{60}$$

$$\phi_{FS}\ (x,y,z,t) = \sum_{n=1}^{N_{kx}} \Delta kx_n \sum_{m=1}^{N_{ky}} \Delta ky_m \left[\ B_{nm}\ (t)\ \omega_{nm}\ k_{nm}^{-1}\ e^{i(kx_n \cdot\ x + ky_m \cdot y)}\right.$$

$$\left. + B^*_{nm}\ (t)\ \omega_{nm}\ k_{nm}^{-i}\ e^{i(kx_n\ \cdot\ x - ky_m\ \cdot\ y)}\ \right] e^{-k_{nm} \cdot z} \tag{61}$$

where

A_{nm}, B_{nm} - complex functions of wave numbers and time (to be further defined below)

A^*_{nm}, B^*_{nm} - complex conjugates

kx_n, ky_m - wave numbers in x and y; e.g., $\frac{\omega_n^2}{g}$

Δkx_n, Δky_m - wave number spacing; e.g., $\Delta kx_n = kx_{n+1} - kx_n$

k_{nm} - $k_{nm}^2 = kx_n^2 + ky_m^2$

ω_{nm} - $\omega_{nm}^2 = g\ [kx_n^2 + ky_m^2]^{1/2}$

Substitution of equations (60) and (61) into the free surface equations (58) and (59) would confirm that the above elevation and potential expressions do indeed satisfy the linearized free surface conditions. We would also obtain, from those substitutions, the following relationships between the complex amplitude functions:

$$\frac{d}{dt} A_{nm}(t) = \omega_{nm} B_{nm}(t) \tag{62}$$

$$\frac{d}{dt} B_{nm}(t) = -\omega_{nm} A_{nm}(t) \tag{63}$$

(The complex conjugates are described by identical expressions).

As part of the free surface elevation representation, we further define the complex amplitude function A_{nm} as

$$A_{nm}(t) = a_{nm} \cos \omega_{nm} t + b_{nm} \sin \omega_{nm} t \tag{64}$$

where a_{nm} and b_{nm} are complex constants. Substitution of (64) into (62) yields the expression for $B_{nm}(t)$.

$$B_{nm}(t) = b_{nm} \cos \omega_{nm} t - a_{nm} \sin \omega_{nm} t \tag{65}$$

Similar expressions also describe the complex conjugates.

Updating the Free Surface

Equations (62), (63), (64) and (65) allow us to develop the expressions for the time evolution of $A_{nm}(t)$ and $B_{nm}(t)$. Consider $A_{nm}(t)$ first. If we increment the time in equation (64) by Δt, we obtain

$$A_{nm}(t + \Delta t) = a_{nm} \cos \omega_{nm} (t + \Delta t) + b_{nm} \sin \omega_{nm} (t + \Delta t) \tag{66}$$

Using the following trigonometric identities,

$$\cos (x \pm y) = \cos x \cos y \mp \sin x \sin y \tag{67}$$

$$\sin (x \pm y) = \sin x \cos y \pm \cos x \sin y$$

we can rewrite (66) as

$$\begin{aligned} A_{nm} (t + \Delta t) &= a_{nm} \cos \omega_{nm} t \cos \omega_{nm} (\Delta t) - a_{nm} \sin \omega_{nm} t \sin \omega_{nm} (\Delta t) \\ &\quad + b_{nm} \sin \omega_{nm} t \cos \omega_{nm} (\Delta t) + b_{nm} \cos \omega_{nm} t \sin \omega_{nm} (\Delta t) \end{aligned} \tag{68}$$

Rearranging (68), we can write

$$\begin{aligned} A_{nm}(t + \Delta t) &= [a_{nm} \cos \omega_{nm} t + b_{nm} \sin \omega_{nm} t] \cos \omega_{nm} (\Delta t) \\ &\quad + [b_{nm} \cos \omega_{nm} t - a_{nm} \sin \omega_{nm} t] \sin \omega_{nm} (\Delta t) \end{aligned} \tag{69}$$

It can be seen that the quantities in the brackets are equivalent to the right-hand sides of equations (64) and (65), so that we can now obtain

$$A_{nm}(t + \Delta t) = A_{nm} (t) \cos \omega_{nm} (\Delta t) + B_{nm} (t) \sin \omega_{nm} (\Delta t) \tag{70}$$

The expression for $B_{nm}(t)$ can be similarly obtained and is given as

$$B_{nm}(t + \Delta t) = B_{nm}(t) \cos \omega_{nm} (\Delta t) - A_{nm}(t) \sin \omega_{nm} (\Delta t) \tag{71}$$

Identical expressions are used for updating $A^*_{nm}(t)$ and $B^*_{nm}(t)$. Thus, expressions (70) and (71) provide the means to update the coefficients which define the wave field. To these updated values must be added any contribution to the wave field from the body motion between times t and $t + \Delta t$, as described next.

Body Effect on the Free Surface

Next we consider the effect of the body source distribution on the wave elevation field over the time interval Δt . Assume that the change in $A_{nm}(t)$ induced by the body between t_n and $t_n + \Delta t$ is expressed by the sum of the effects of the individual sources and their images. For convenience, we will first replace a uniform source density acting over a panel of area A_i by a single point source with strength

$$s_i(t) = A_i \, [\sigma_i \, (t_n) + (t-t_n) \, \dot{\sigma} \, (t_n)] \tag{72}$$

which is located at the panel center. The time, t, is defined to be $t = t_n + 1/2\Delta t$, the midpoint of the time interval.

At any given point on the mean water level plane (x,y,0), the vertical velocity induced by the source points and their images is given by

$$\frac{\partial \phi_B(t)}{\partial z} = - \sum_{i=1}^{N_B} \frac{2z_i s_i(t)}{[(x-x_i)^2 + (y-y_i)^2 + z_i{}^2]^{3/2}} \tag{73}$$

According to Chapman [6], equation (73) can also be expressed in an integral form given as

$$\frac{\partial \phi_B(t)}{\partial z} = \frac{2}{\pi} \sum_{i=1}^{N_B} s_i(t) \int_0^\infty dkx \; e^{ikx(x-x_i)} \int_{-\infty}^{\infty} dky \cos ky(y-y_i) \cdot e^{-(kx^2 + ky^2) z_i} \tag{74}$$

Substituting (74) into the linearized kinematic free surface condition (59), and integrating over the time interval Δt , we can rewrite (74) to show how the elevation changes over Δt due to the body,

$$\Delta\eta_B(x,y) = \frac{2}{\pi} \int_{t_n}^{t_n + \Delta t} dt \sum_{i=1}^{N_B} s_i(t) \int_0^\infty dkx\, e^{ikx(x-x_i)} \int_{-\infty}^{\infty} dky \cos ky(y-y_i)$$

$$\cdot\, e^{-(kx^2 + ky^2)z_i} \qquad (75)$$

We may also write $\Delta\eta_B(x,y)$ in a different form, following the same form of the original expression (60) defining the free surface elevation. In that form, we can write

$$\Delta\eta_B(x,y) = \sum_{n=1}^{N_{kx}} \Delta kx_n \sum_{m=1}^{N_{ky}} \Delta ky_m \Big[\Delta A_{nm}^{BODY}(t)\, e^{i(kx_n \cdot x + ky_m \cdot y)}$$

$$+ \Delta A_{nm}^{BODY*}(t)\, e^{i(kx_n \cdot x - ky_m \cdot y)} \Big] \qquad (76)$$

Then, if we represent the wave number integrals in (75) numerically by a finite summation, and also evaluate the time integral, it should be evident by direct comparison with (76) that the A_{nm} coefficient increments can be written as

$$\Delta A_{nm}^{BODY} = 2 \frac{\Delta t}{\pi} \sum_{i=1}^{N_B} s_i(t_n + 1/2\, \Delta t)\, e^{-i(kx_n \cdot x_i + ky_m \cdot y_i)} e^{-k_{nm} z_i} \qquad (77)$$

$$\Delta A_{nm}^{BODY*} = 2 \frac{\Delta t}{\pi} \sum_{i=1}^{N_B} s_i(t_n + 1/2\, \Delta t)\, e^{-i(ky_n \cdot x_i - ky_m \cdot y_i)} e^{-k_{nm} z_i} \qquad (78)$$

The changes in B_{nm} induced by the body can be derived from (78) and (79) using the relationship between $A_{nm}(t)$ and $B_{nm}(t)$ given in (63). Multiplying both sides of (63) by Δt and then integrating both sides over time, we obtain

$$\Delta B_{nm}^{BODY} = -\,1/2\, \frac{\partial}{\partial t} A^{BODY}(t)\, \omega_{nm} (\Delta t)^2 \qquad (79)$$

or

$$\Delta B_{nm}^{BODY} = - \frac{(\Delta t)^2}{\pi} \sum_{i=1}^{N_B} s_i(t_n)\, e^{-i(kx_n \cdot x_i + ky_m \cdot y_i)}\, e^{-k_{nm} z_i}\, \omega_{nm} \tag{80}$$

with a similar expression for ΔB_{nm}^{BODY*} .

The effect of the body on the free surface can now be easily included by adding ΔA_{nm}^{BODY}, ΔA_{nm}^{BODY*}, ΔB_{nm}^{BODY}, and ΔB_{nm}^{BODY*} to the time evolution equations given earlier as (70) and (71) and the corresponding (complex conjugate) expressions.

Free Surface-Induced Pressure and Acceleration

Finally, we need to evaluate the pressure and acceleration induced by the free surface disturbance at the center of each panel. These two quantities are required in steps (4) and (1), respectively, of the simulation procedure. The pressure at any point (x,y,z) is given by

$$p_{FS} = - \rho \frac{\partial \phi_{FS}}{\partial t}$$

$$= \rho \sum_{n=1}^{N_{kx}} \Delta kx_n \sum_{m=1}^{N_{ky}} \Delta ky_m\, e^{(ikx_n \cdot x - k_{nm} \cdot z)}$$

$$[\, A_{nm}(t)\, e^{iky_m \cdot y} + A_{nm}^{*}(t)\, e^{-iky_m \cdot y} \,] \tag{81}$$

the acceleration at any point (x,y,z) coincident with a unit normal vector (n_x, n_y, n_z) may be written as

$$\vec{a}_{FS} \cdot \vec{n} = - g \sum_{n=1}^{N_{kx}} \Delta kx_n \sum_{m=1}^{N_{ky}} \Delta ky_m e^{(ikx_n \cdot x - k_{nm} \cdot z)}$$

$$[\ A_{nm}(t) \cdot e^{iky_m \cdot y} (iky_n \cdot n_x + iky_m \cdot n_y - k_{nm} \cdot n_z)$$

$$+ A^*_{nm}\ (t) \cdot e^{-iky_m \cdot y} \cdot (ikx_n \cdot n_x - iky_m \cdot n_y - k_{nm} \cdot n_z)\ ] \tag{82}$$

2.5.6 Ambient Wave Field Representation

The ambient wave field is represented by a linear wave system of amplitude A. Consider unidirectional, regular waves and define the velocity potential as

$$\phi_A(x,z,t) = -\frac{Ag}{\omega} e^{-kz} \sin(kx-\omega t) \tag{83}$$

The x and z components of velocity can then be expressed as

$$V_x(t) = -\frac{\partial \phi_A}{\partial x} = \frac{Agk}{\omega} e^{-kz} \cos\ (kx-\omega t) \tag{84}$$

$$V_z(t) = -\frac{\partial \phi_A}{\partial z} = -\frac{Agk}{\omega} e^{-kz} \sin(kx-\omega t) \tag{85}$$

The x and z components of acceleration can be given as

$$a_x(t) = \frac{\partial V_x}{\partial t} = Agk\ e^{-kz} \sin(ky-\omega t) \tag{86}$$

$$a_z(t) = \frac{\partial V_z}{\partial t} = Agk\ e^{-kz} \cos(kx-\omega t) \tag{87}$$

The dynamic pressure may be expressed as

$$P_A(t) = -\rho \frac{\partial \phi_A}{\partial t} = -Ae^{-kz} \cos(kx-\omega t) \tag{88}$$

We can now evaluate the normal acceleration and pressure induced at the center of each panel by the ambient wave field. These quantities are used in Steps 1 and 4 of the simulation procedure. The pressure is given in (88) and the normal acceleration is simply

$$\vec{a}_A \cdot \vec{n} = \vec{a}_x \cdot \vec{n}_x + \vec{a}_z \cdot \vec{n}_z \tag{89}$$

where the velocity components are given in (84) and (85). We will assume that in an irregular wave system, the total velocity, acceleration and pressure components can be described as the superposition of individual single frequency components.

2.6 HULL GIRDER LOADS

Hull girder loads are computed in the conventional manner of strip theory where the dynamic shear force at a cross section is the difference between the inertia force and the sum of external forces acting on the portion of the hull forward of the section. The vertical shear force and bending moment at any location x_o along the ship's length is, in the original coordinate system,

$$SF_y\ (x_o) = \int_{x_o} \frac{df_y}{dx}\, dx \tag{90}$$

and

$$BM_y\ (x_o) = \int_{x_o} (x-x_o)\ \frac{df_y}{dx}\ dx \tag{91}$$

where

$$\frac{df_y}{dx} = -\ \delta m\ (\ddot{y} - x\ \ddot{\psi}\ ) + \frac{dY_h}{dx} + \frac{dY_w}{dx} + \frac{dY_f}{dx} + \frac{dY_b}{dx} + y_q \tag{92}$$

and δm = local sectional mass. Integrations are performed from the location x_o forward to the bow.

The lateral shear force, bending moment and torsional moment are given as

$$SF_z(x_o) = \int_{x_o}^{bow} \frac{df_z}{dx} dx \tag{93}$$

$$BM_z(x_o) = \int_{x_o}^{bow} (x-x_o) \frac{df_z}{dx} dx \tag{94}$$

$$TM_x(x_o) = \int_{x_o}^{bow} \frac{dm_x}{dx} dx \tag{95}$$

where

$$\frac{df_z}{dx} = - \delta m(\ddot{z} - x\ddot{\phi} - \bar{y} \ \ddot{\theta}) + \frac{dZ_h}{dx} + \frac{dZ_w}{dx} + \frac{dZ_b}{dx} + Z_q \tag{96}$$

$$\frac{dm_x}{dx} = - i_x \ddot{\theta} - \delta m\bar{y} \ (\ddot{z} - x\ddot{\phi}) + \frac{dK_h}{dx} + \frac{dK_w}{dx} + \frac{dK_b}{dx} + K_q \tag{97}$$

and $\bar{y}$ = local section's center of gravity relative to ship c.g. (positive up).

i_x = local section's mass moment of inertia.

2.7 NUMERICAL ASPECTS

2.7.1 Solution Procedures - Two-Dimensional Approach

The equations of motion as represented by expressions (4) - (7) are numerically integrated for each time to yield velocities and displacements. The fourth-order Runga-Kutta scheme is used for this purpose. At each time step, two-dimensional hydrodynamic coefficients are evaluated for each section for either the still water draft or at the instantaneous depth of immersion, at the option of the user. When it is desired that the hydrodynamic coefficients be draft dependent, the coefficients are obtained by linear interpolation from an array of six values corresponding to six drafts which cover a range specified by the user in the input data. Since the hydrodynamic coefficients are also frequency dependent, they are selected using linear interpolation from an array of twelve values associated with a range of twelve frequencies specified by the user. Thus, when the two-dimensional sectional coefficients are to be both draft and frequency dependent, two-dimensional linear interpolation is performed.

There are two methods for the selection of frequencies at which to evaluate the hydrodynamic coefficients. One is based on a scheme that samples the previous two zero-crossing periods of sectional motion in sway, heave and roll. The two values are averaged, and three characteristic frequencies are derived for evaluation of the sectional hydrodynamic coefficients. The sway-roll or roll-sway coefficients are (arbitrarily) linked to the sway characteristic frequency. The other allows the user to specify a frequency, such as the peaks of the response spectra or peak of the wave energy spectrum.

The sectional hydrodynamic coefficients are computed using the Frank close-fit method. Spatial derivatives of added mass and damping coefficients in the x-direction are approximated by center-space finite differences. In the calculation of the derivative of added mass with respect to the relative wave elevation used in the computation of vertical flare force, the added mass is evaluated at each time step at the instantaneous immersed draft of the section. For each time step interval Δt, the change in added mass is computed and divided by the change in instantaneous draft of the section. During water exit, the calculation of the impact term is not performed and the sectional flare force is set to zero.

2.7.2 Solution Procedures - Three-Dimensional Analysis

Specification of Parameters for Free Surface Harmonics

There are three parameters which must be chosen. These parameters dictate the wave numbers and wave number spacing for the free surface representation. The following parameters are to be specified for the three-dimensional transient analysis:

L_x, L_y - maximum distances in ±x and ±y from body origin defining the physical region over which the wave field is required.

T - maximum time interval for transient simulation.

l_x, l_y - minimum half-wavelength related to the smallest scale of disturbance or physical feature of the hull that needs to be represented.

There are no set rules for the selection of these parameters, and their choice must be based on an understanding of the problem. There are, however, some guidelines which generally seem to work well. The values of L_x and L_y should be 2 to 2.5 times the length and beam of the body, respectively. The values of l_x, l_y should be on the order of the average dimensions of the hull panels. The parameter T can be set to about 60% of the actual time for which a transient analysis is desired.

The wave numbers and wave number spacing are then specified according to the following relations:

(a) Maximum step sizes for kx_n and ky_m are:

$$kx_{n+1} - kx_n \leq \frac{2\pi}{L_x}$$

$$ky_{m+1} - ky_m \leq \frac{2\pi}{L_y} \tag{98}$$

(b) The minimum upper bounds of kx and ky are:

$$kx_{N_{kx}} \geq \frac{2\pi}{l_x}$$

$$ky_{N_{ky}} \geq \frac{2\pi}{l_y} \tag{99}$$

(c) Another set of conditions for wave number spacing is:

$$\sqrt{kx_{n+1}} - \sqrt{kx_n} \leq \frac{2\pi}{(T\sqrt{g})}$$

$$\sqrt{ky_{m+1}} - \sqrt{ky_m} \leq \frac{2\pi}{(T\sqrt{g})} \tag{100}$$

A physical interpretation of the wave number spacing required by the above conditions has been pointed out by Chapman [5]. The wave numbers defined by (98), (99), and (100) correspond in the physical domain to "vertical walls" which must be placed sufficiently far away from the body so that the free surface disturbances generated near the body do not reflect back over the time interval. Associated with the faster propagating, longer wavelengths are vertical walls placed further away - the shorter wavelength walls are nearer. The result is that the time interval for each wave to reflect is equal. For further discussion of the reasoning behind this method of wave number spacing and selection criteria, see [5].

Integration with Equations of Motion

At a time specified by the user, the three-dimensional transient analysis can start, using accelerations and velocities generated by the solution of the equations of motion at that time step. Then, data generated (pressures and loads) from the three-dimensional analysis is not used for a period of time specified by the user, so-called "start-up" time. This allows the free surface disturbance to fully develop since, initially, there exist no radiated or diffracted waves within the three-dimensional simulation. Based on the present level of experience with the program, the minimum start-up time for ship forms has not been fully defined; however, for a sphere, two full cycles of motion appear sufficient.

3.0 NUMERICAL RESULTS

Several computer codes have been developed according to the predictive techniques discussed in the previous sections. The main program is entitled SSX (Ship Simulator, Experimental). It performs the simulation of ship motions, hull girder loads and hull pressures. There are two preprocessing programs - HYDREX2 and HYDREX3. The program HYDREX2 computes sectional added mass and damping coefficients for an array of twelve frequencies and six drafts using the Frank close-fit technique. The program also calculates general hydrostatic data for each of the six drafts. The program HYDREX3 computes areas, normals and panel center coordinates for the panel (either quadrilateral or triangular) representation of the hull. It also evaluates three computationally demanding arrays used for the simulation of hull pressures. For further information about these programs, refer to Volume II (Program Manual).

Because the debugging and testing of SSX or any time simulation program, for that matter, is an extremely time-consuming process, the extent of the validation effort has been limited to the minimum number of cases needed to test the capabilities of the computer programs and verify the basic theory and numerical techniques when possible.

3.1 TWO-DIMENSIONAL HYDRODYNAMIC COEFFICIENTS

Sectional added mass and damping coefficients computed by Program HYDREX2 for the SL-7 containership were checked against values generated by a conformal mapping method, as given by Zielinski [33]. Agreement was excellent in most cases. As an example, Figure 4 compares added mass and damping coefficients for heave and sway at midships with a draft of 32.8 feet.

3.2 MOTION RESPONSE AND HULL GIRDER LOADS IN SMALL AMPLITUDE REGULAR WAVES

A few comparisons between computed and experimental values for heave and pitch motions in one-foot regular waves will first be presented to check the time-domain solution methods using strip theory computed forces where the added mass and damping coefficients are calculated for the mean still water sectional drafts. Figure 5 shows the heave and pitch amplitudes and phases for the Series 60 standard hull form with block coefficient 0.70 at Froude numbers 0.15 and 0.20. The points in the figures represent experimental results by Gerritsma and Beukelman [34]; the solid line is computed by the present method, and the broken line by modified strip theory (frequency domain) of Gerritsma and Beukelman. Note that the pitch amplitude is scaled by wave slope. It is seen that agreement is generally good with the exception of heave amplitude in the vicinity of $\lambda/L = 1.0$. The reason for this disagreement has not been identified.

Figure 6 gives the theoretical and experimental pitch and heave values for the SL-7 containership at 25 knots in a full load condition. The experimental values shown are by Dalzell [35]; the solid line represents the present

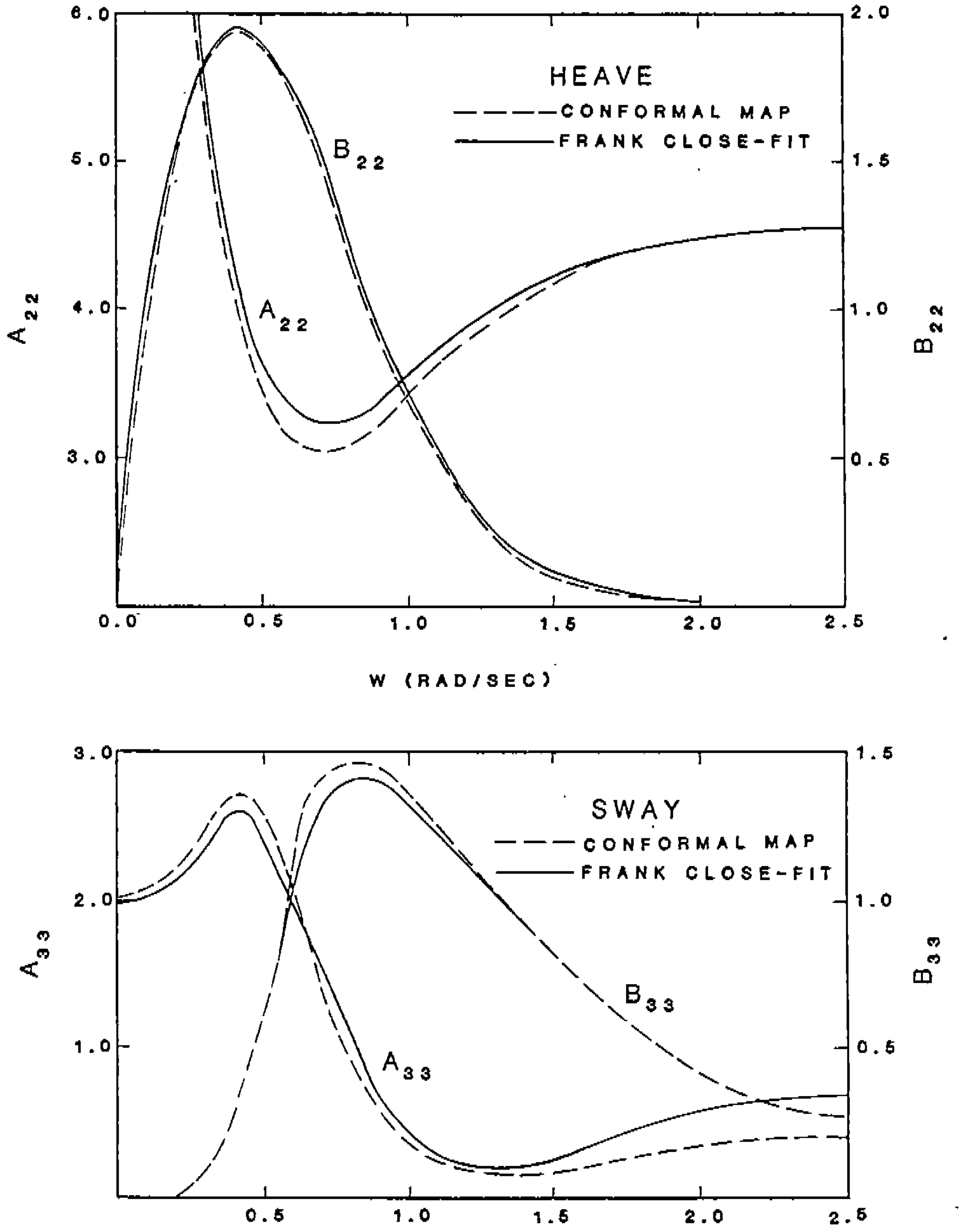

Figure 4 Added mass and damping coefficients. Comparison of HYDREX2 (Frank close-fit) with CGSCORES (Conformal Mapping). SL-7 Containerships, Midships section.

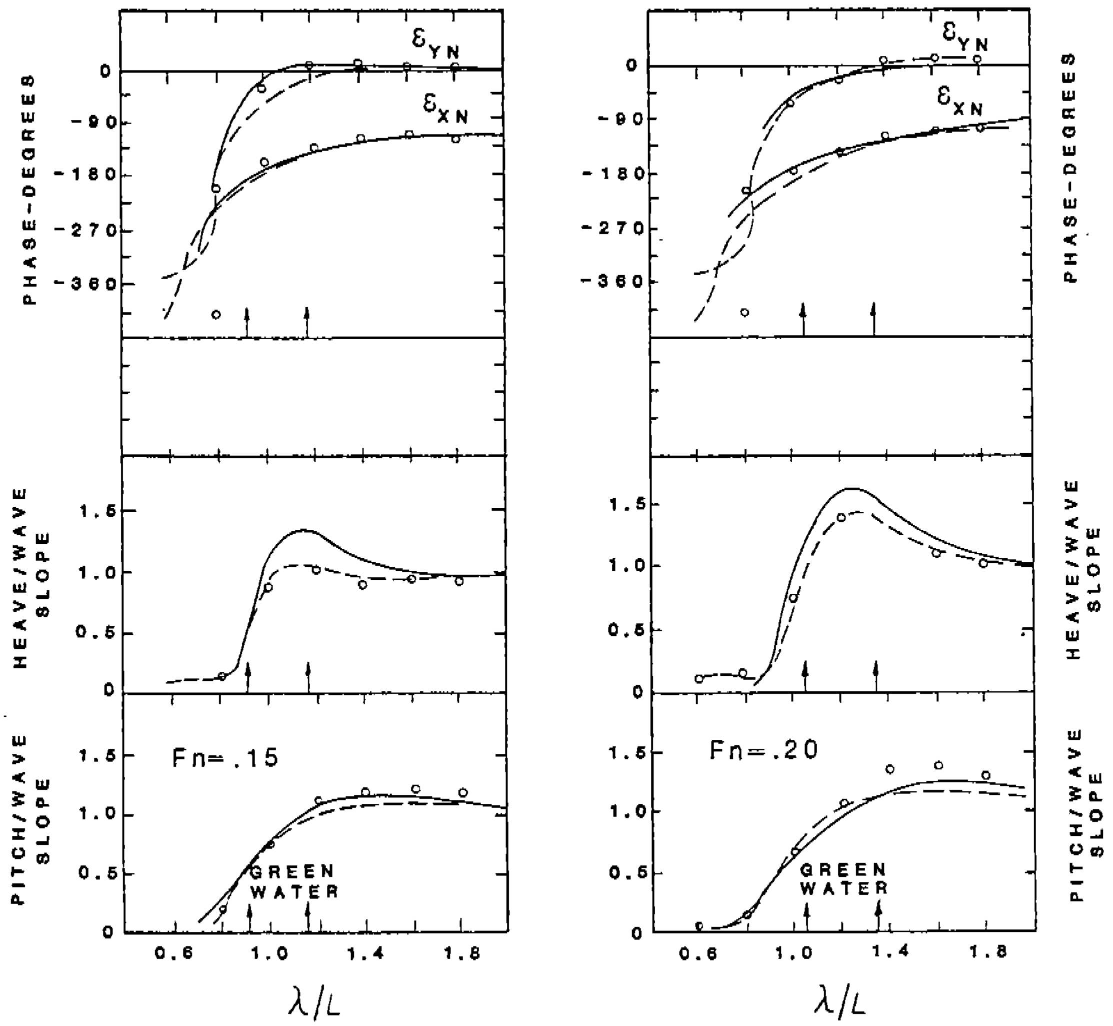

Figure 5 Calculated and measured amplitude and phase characteristics for heave and pitch. Series 60. Block coefficient 0.70. Froude Numbers 0.15 and 0.20.

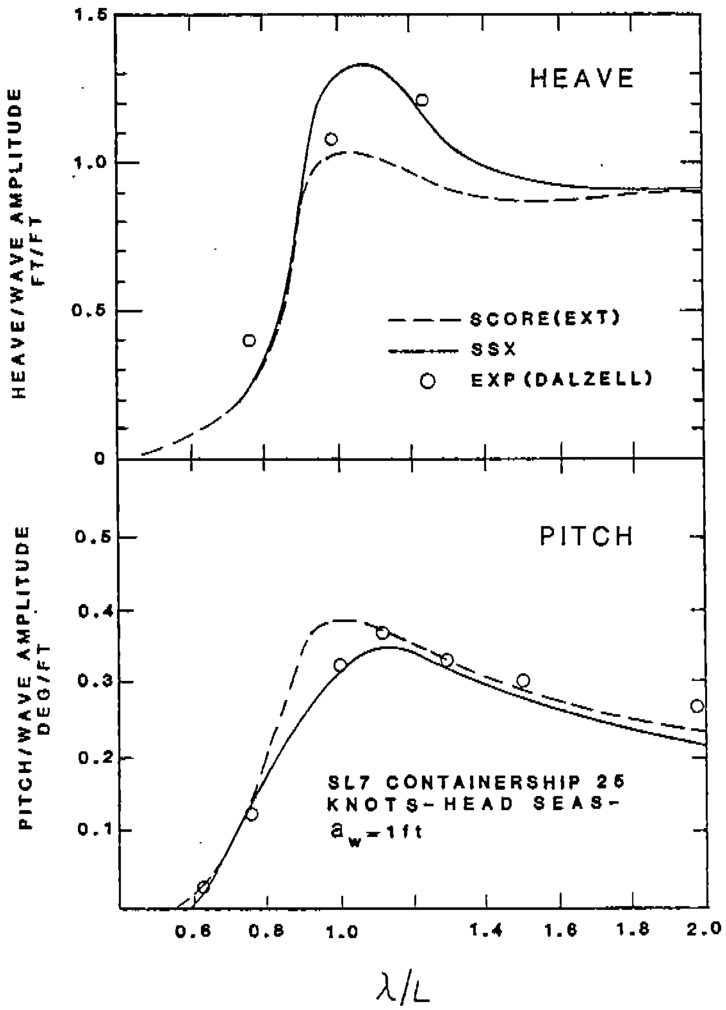

Figure 6 Calculated and measured amplitudes of heave and pitch. SL-7 Containership. 25 knots, head seas.

method, and the broken lines are strip theory results from the SCORES Program [4]. Note that the experimental values for heave at longer wave lengths have not been included because it has been suggested [35] that they were too large and are in error. (The short wavelength experimental values are assumed to be correct.) It is seen that pitch values predicted by SSX agree well with the other values. Heave response, like before, is too high around $\lambda/L = 1.0$.

The routine to compute hull girder loads in the present version of SSX is still in a premature stage. The computation of moment arms and sectional inertial forces essentially assumes a lumped mass model. There is no attempt to account for "trapezoidal" sections of the weight curve. Also, weight segments must correspond to the segment between adjacent stations. However, the routine, as written, does provide the means to evaluate the effect of including various forces or different techniques by which to evaluate such forces.

A comparison between computed and experimental midships bending moment amplitudes for the SL-7 containership at 25 knots in a full load condition is shown in Figure 7. The experimental data is from Dalzell [35]; the present theory is represented by the solid line, the broken line shows predictions from SCORES [4]. Agreement is shown to be very good.

Finally, comparison between theory and experiments for midships horizontal and vertical bending moments and pitch angle are shown in Figure 8 for the SL-7 in oblique waves (210°). Agreement is satisfactory with the exception of vertical bending moment. Further investigation is needed to identify the causes of disagreement.

3.3 IRREGULAR WAVES

A test case was run to check the program's ability to predict motion and loads response in an irregular sea. The SL-7 was subjected to an irregular head sea wave system approximately representing a Bretschneider spectral formulation with H_s = 10.8 ft and T_0 = 8 secs, where H_s is significant height and T_0 is the period associated with the peak of the sea spectrum. Forward speed was 5 knots. The frequency at which hydrodynamic coefficients used for ship motion-induced forces were to be calculated was set to .8217 - the encounter frequency associated with the peak of the sea spectrum. The sea spectrum was decomposed into ten components. Randomly generated phase angles were used to generate the irregular sea, superimposing the ten regular wave components. A real time simulation of 20 minutes was carried out. Statistics were acquired for heave, pitch and midship's vertical bending moment. Table 1 compares the statistical results of the simulation with predictions from the strip theory frequency domain program developed at MIT and described by Loukakis [36].

As can be seen, the SSX motion values are slightly higher than the MIT program values. The vertical bending moments compare very well. The measured significant RMS wave height from the SSX simulation was close to the desired value of 5.33, providing some assurance that the irregular wave spectrum is correctly being broken down into wave components.

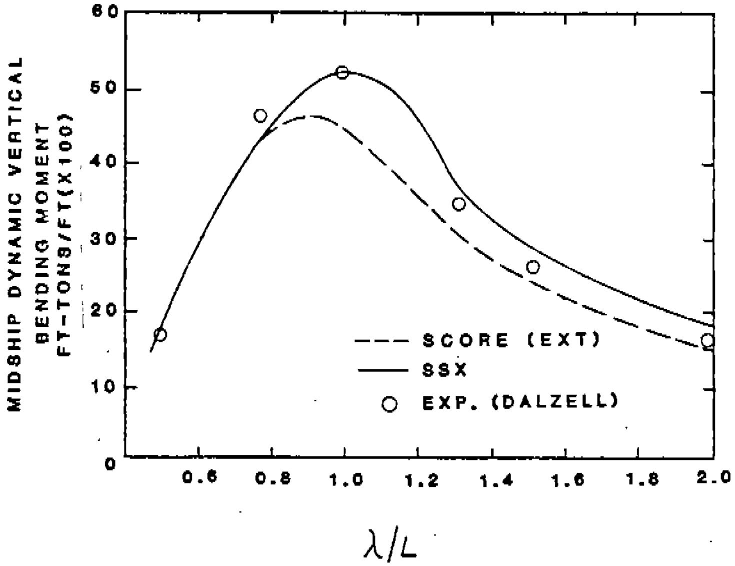

Figure 7 Midship Dynamic Vertical Bending Moment for SL-7 Containership. 25 knots, head seas, full load, regular waves, ampl. = 1 ft.

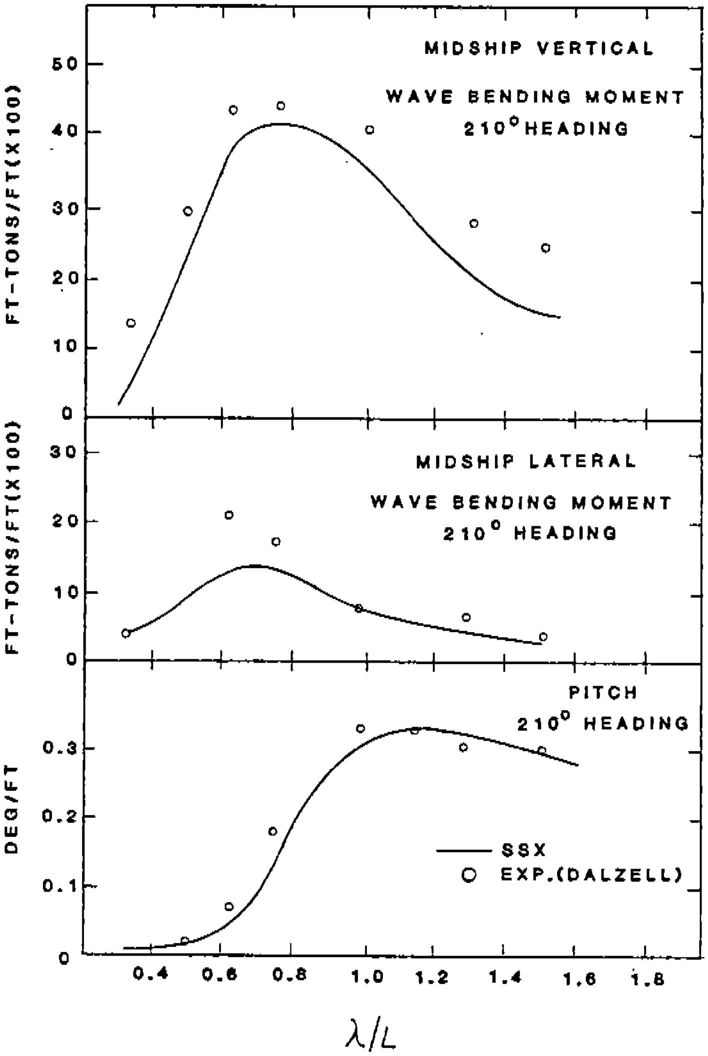

Figure 8 Midship Vertical, Horizontal Wave Bending and Pitch. SL-7 Containership, 25 knots, 210° heading.

Table 1

	SSX	MITA
RMS Pitch (deg)	.2644	.2606
RMS Heave (ft)	.5560	.5308
RMS V.B.M. (ft-tons)	56844	56415
RMS Wave Height	5.23	5.33

One feature of the SSX program is the scheme to average the two most recent zero-crossing periods and derive a "characteristic" frequency at which sectional hydrodynamic coefficients are calculated. This feature was applied to the same test case as before. The results are shown in Table 2.

Table 2

	SSX	
	with f (freq)	without f (freq)
RMS Pitch (deg)	.2619	.2644
RMS Heave (ft)	5507	.5560
RMS V.B.M. (ft.-tons)	55432	56844

The effect of invoking the characteristic frequency feature is to reduce the SSX values. The calculated sectional characteristic heave frequency during the course of the simulation generally ranged between $\omega = .35$ and .49. This reflects the actual response of the ship and is significantly different from the .8217 value used in the previous simulation. There is still relatively little difference between the RMS values of the two test cases.

3.4 NONLINEAR RESPONSE IN LARGER WAVES

As part of many model-test programs, a so-called linearity check is performed where the same test runs are made with increasing wave amplitudes to determine if responses are linearly proportional to wave amplitude. Similar experiments have been performed using SSX. The runs were made at 25 knots in head seas with wave amplitudes of 5, 10 and 15 feet. Added mass and damping coefficients were computed at the mean still water draft. Plots showing peak heave, pitch and midship's vertical bending moment are provided in Figures 9, 10 and 11, respectively.

Note that in Figures 9 and 10, the heave and pitch response has been divided by wave amplitude. In both of these figures, nonlinear behavior is exhibited; that is, normalized response is reduced at higher amplitudes, with the exception of the pitch "bow-down" response. It appears quite linear. In Figure 11, the midship's vertical wave bending moment is shown for various wave lengths and wave heights. Note that in this plot, the response has not been normalized by wave amplitude. In general, response is linear with respect to wave amplitude, with the exception of slight nonlinearities shown for hogging at 15 feet wave amplitude for $\lambda/L = 0.75, 1.0, 1.25$.

Also shown in Figure 11 is the effect of including the flare impact force. The broken lines show the bending moment when flare is included. The effect is only measurable in sagging. The values for hogging were so close to the non-flare values that, if drawn, they would be coincident with the lines shown already. (For hogging, a slight reduction in the magnitude of bending moment was the effect of including flare forces.)

The effect of including flare forces on the heave and pitch response is rather minor except around $\lambda/L = 1$, as shown in Figures 12 and 13, respectively. The heave and pitch response is shown for a ten-foot regular wave over a range of wave lengths. As shown in Figures 12 and 13, the effect of flare force inclusion is to reduce the amplitudes in both directions (heave up and down, pitch bow-up and bow-down) between $\lambda/L = 1.0$ and 1.25. It can also be seen in these figures that for this wave amplitude, the SL-7 containership pitches bow-down more than bow-up, and heave-down more than heave-up over the range of wavelengths.

Linearity checks were performed up to regular wave amplitudes of 15 feet because, at 20 feet and above, inconsistent results were sometimes obtained from the program. A check of what was physically occurring at these greater wave amplitudes was performed graphically. Figures 14 and 15 show the motion of the SL-7 containership at 25 knots for $\lambda/L = 1.5$ and 1.75, respectively. It can be seen that for $\lambda/L = 1.5$, bow submergence and emergence are quite extreme. For $\lambda/L = 1.0$ and $\lambda/L = 1.25$, behavior was equally extreme.

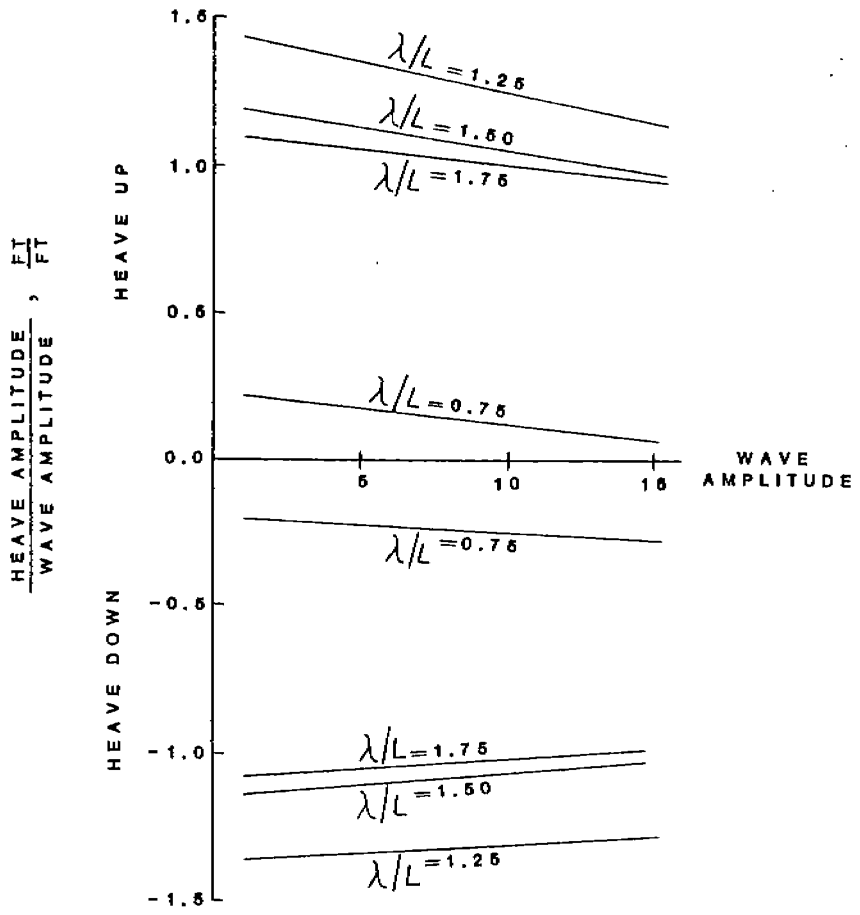

Figure 9 Heave Amplitude/Wave Amplitude vs. Wave Amplitude. SL-7 Containership, 25 knots, full load condition, Head Seas.

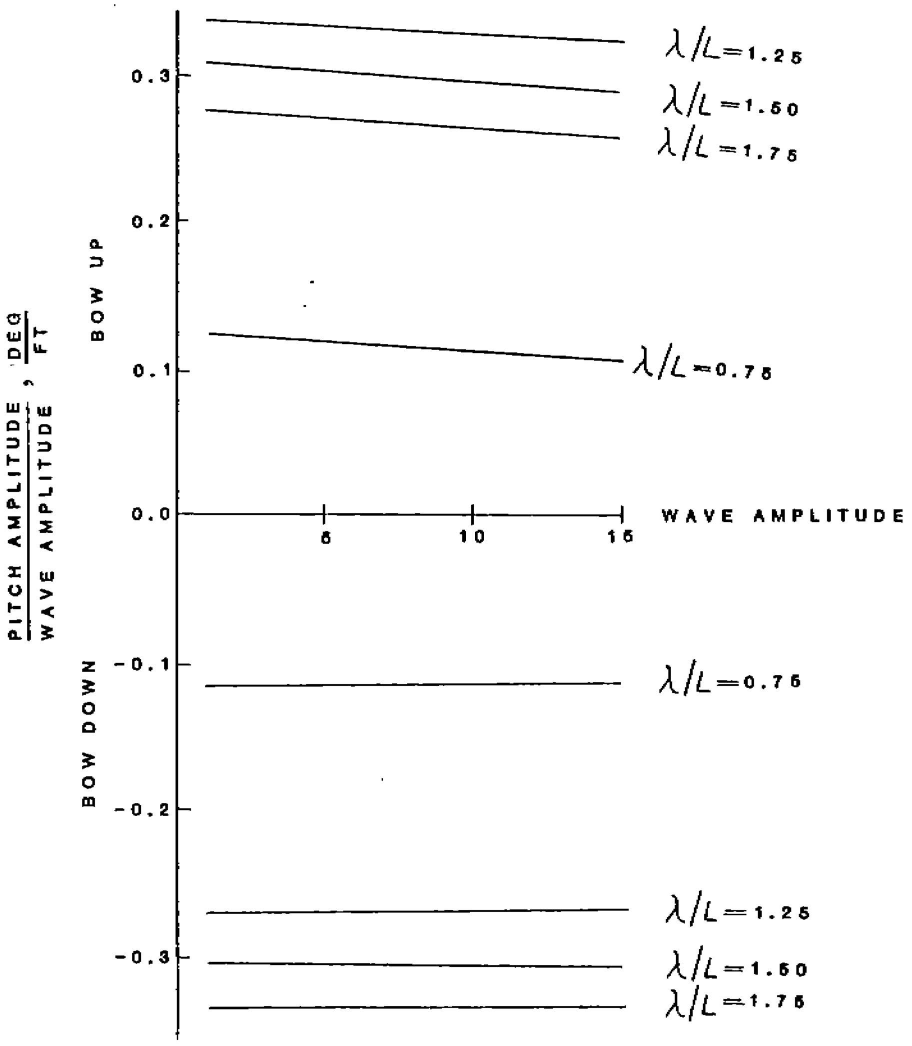

Figure 10 Pitch Amplitude/Wave Amplitude vs. Wave Amplitude. SL-7 Containership, Full Load, Head Seas, 25 knots.

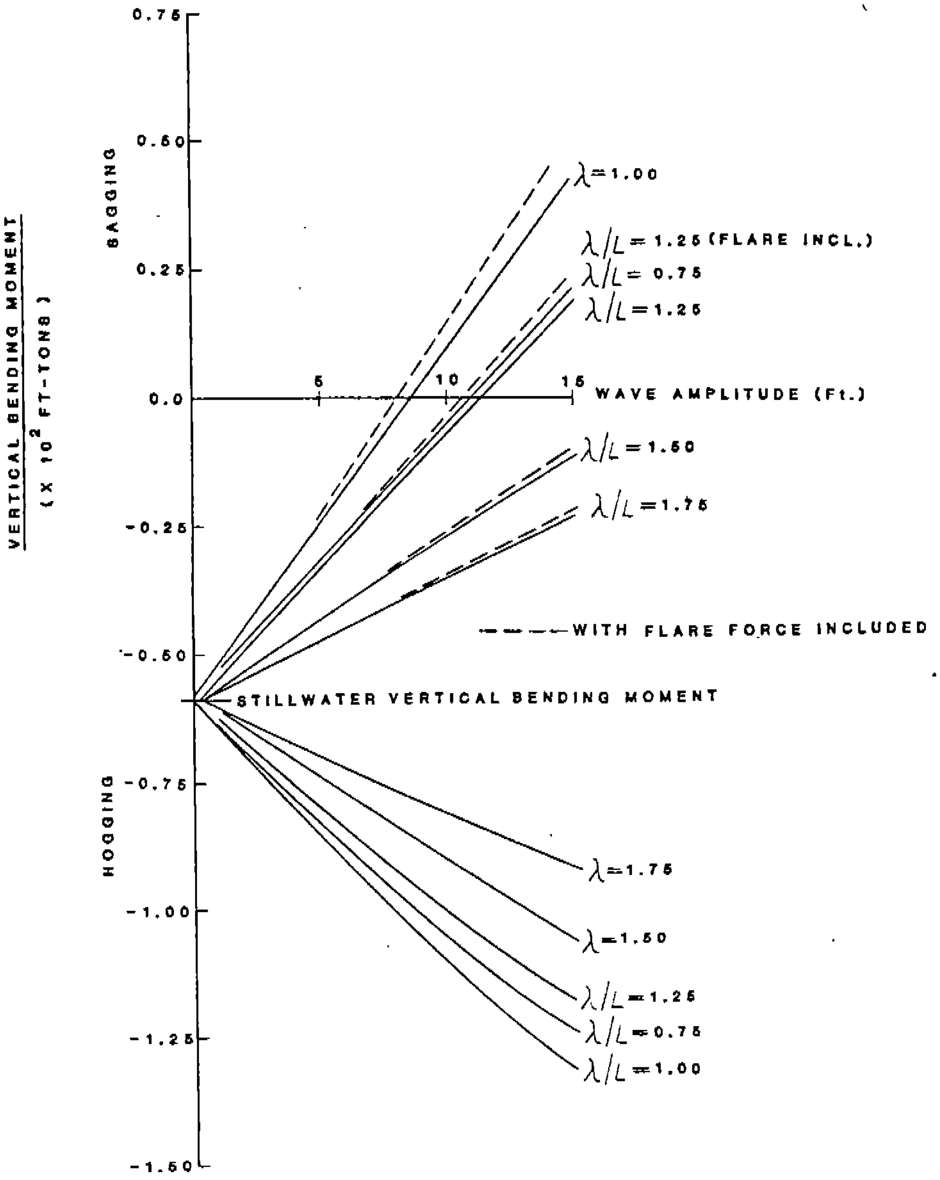

Figure 11 Midship Vertical Bending Moment, SL-7 Containership, 25 knots, head seas (effect of flare on sagging shown).

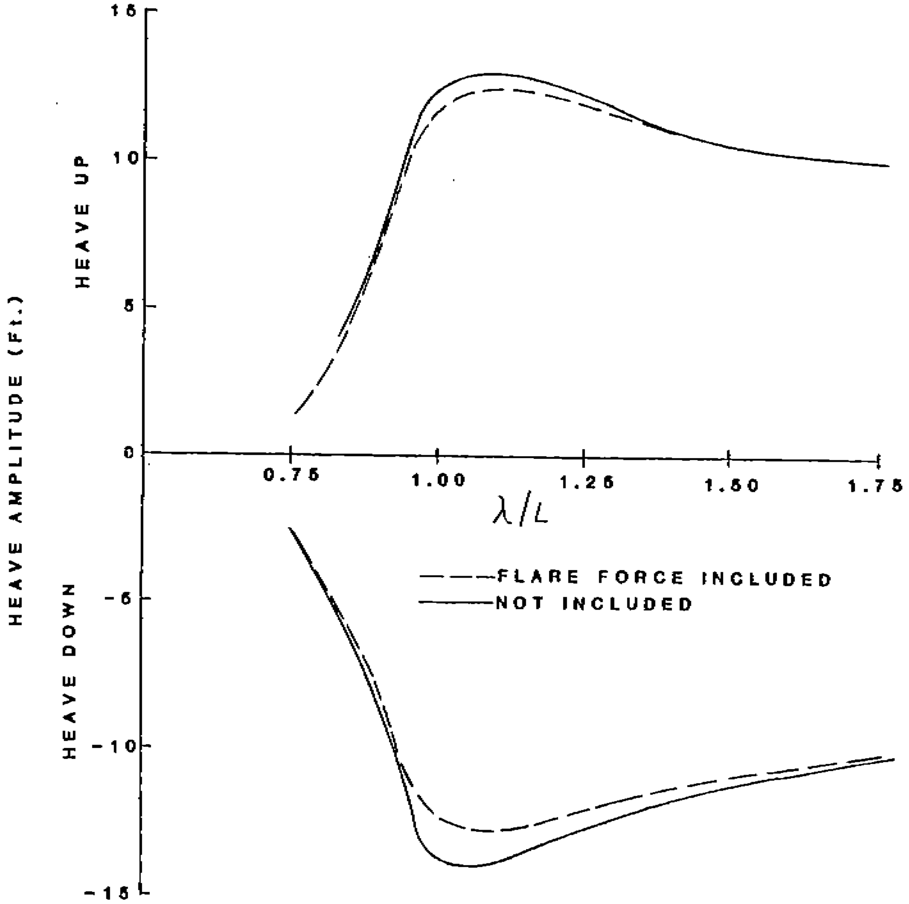

Figure 12 Heave Amplitude vs. Wavelength/Shiplength. Comparison of Theoretical Prediction with and without flare force included

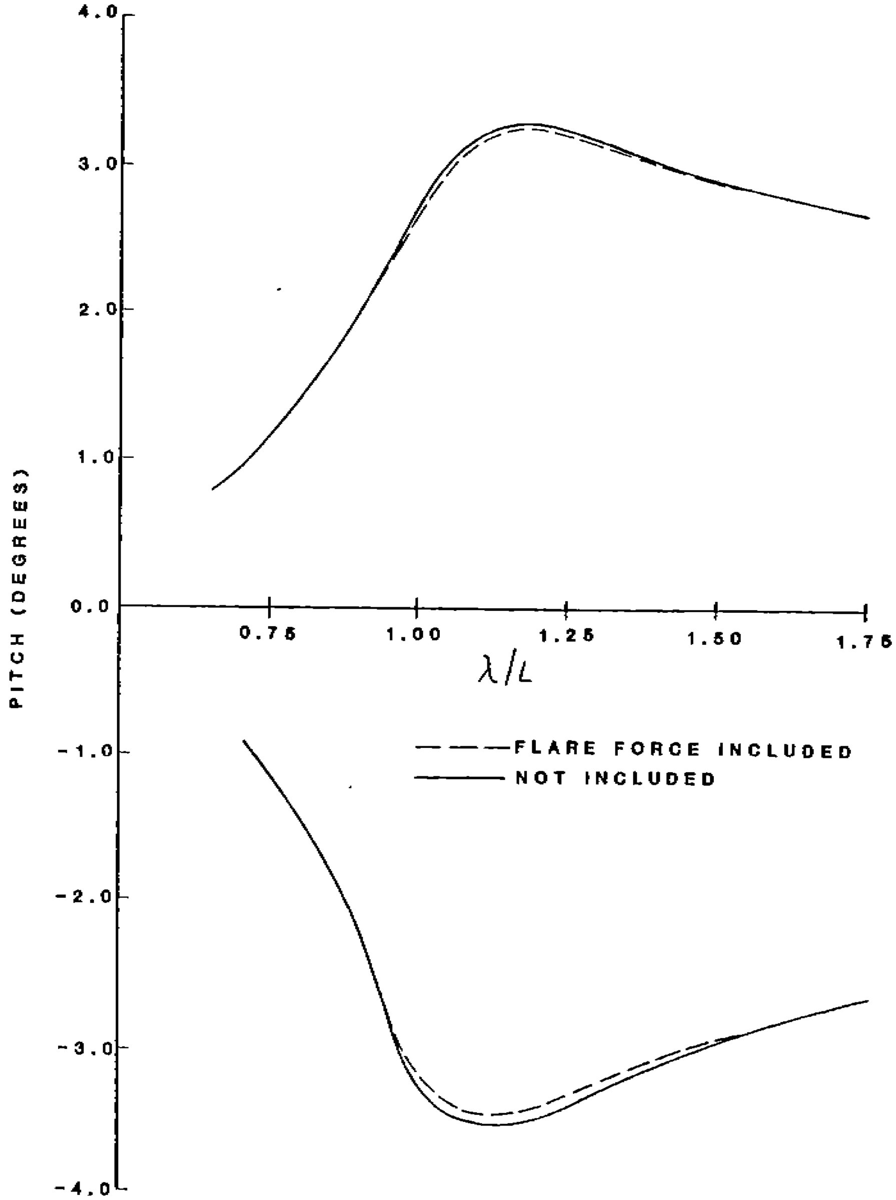

Figure 13 Pitch Amplitude vs. Wavelength/Shiplength. Comparison of Theoretical Prediction with and without flare force included.

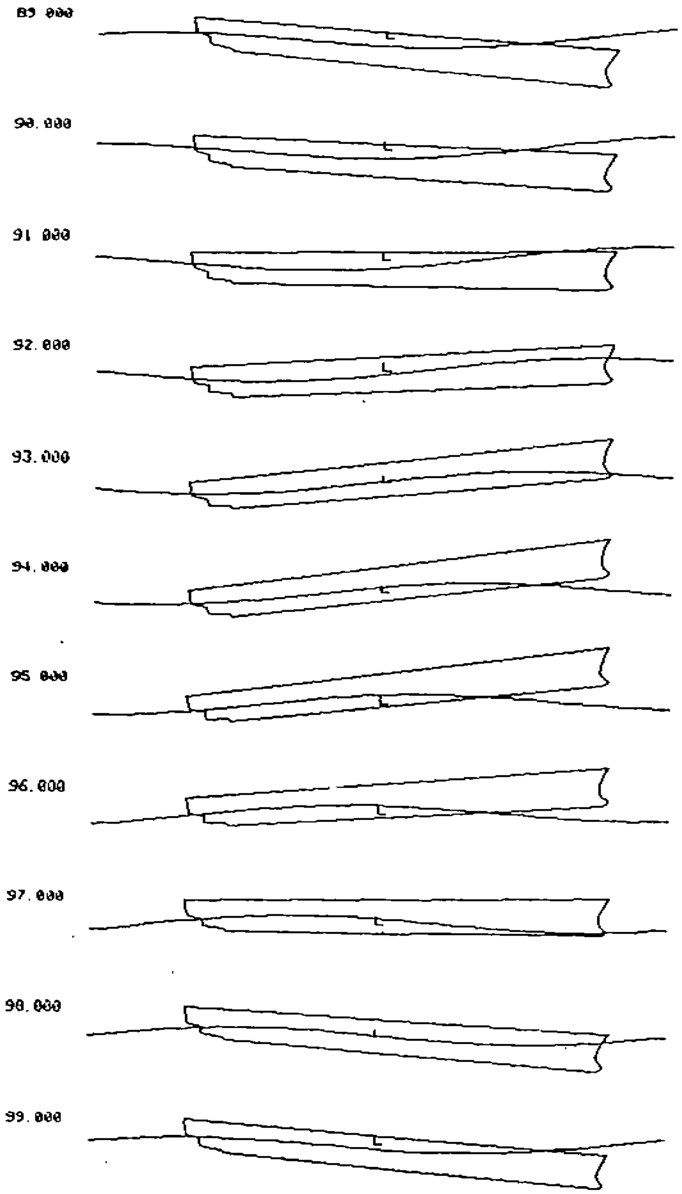

Figure 14 SL-7 Containership in Head Seas at 25 knots. Wavelength/Shiplength = 1.5.

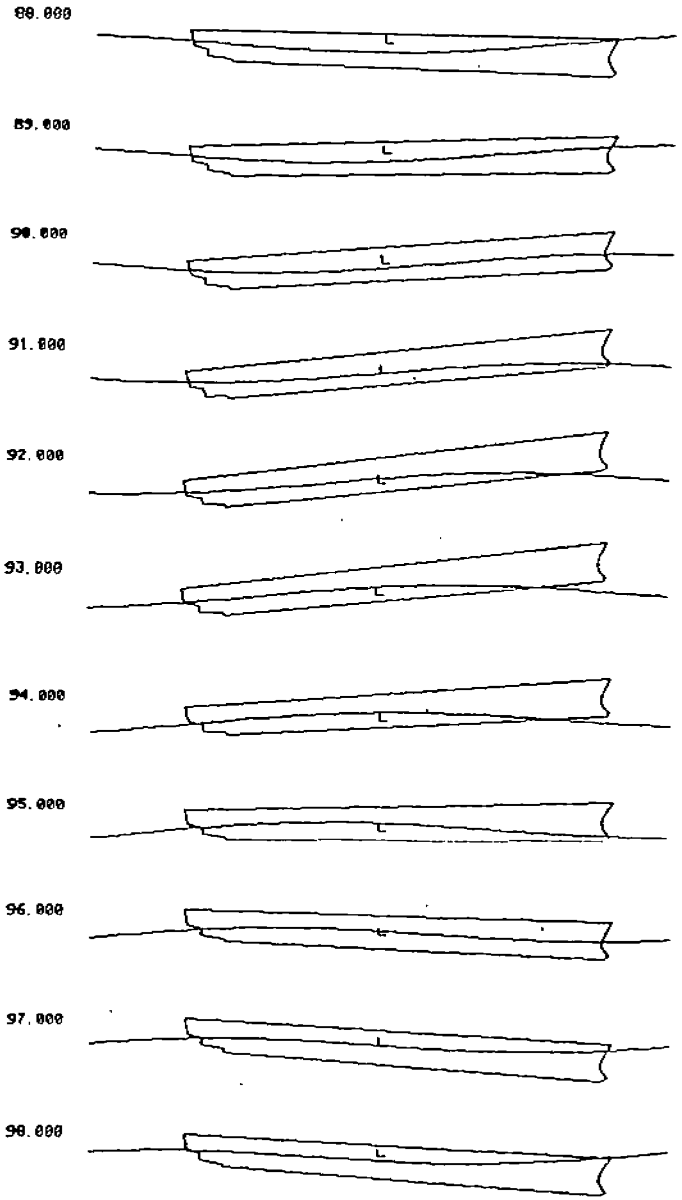

Figure 15 SL-7 Containership in Head Seas at 25 knots. Wave Amplitude = 20 ft, Wavelength/Shiplength = 1.75.

3.5 INFLUENCE OF DRAFT-DEPENDENT HYDRODYNAMIC COEFFICIENTS

In order to evaluate the influence of calculating the sectional added mass and damping coefficients at the actual draft (average) at each station for each instant, a series of runs were made at 25 knots in head seas, $\lambda/L = 2.0$. Flare forces were not included in these runs. Wave amplitudes were 5, 10 and 20 feet. It was found that within this range of conditions, the use of draft-dependent coefficients had negligible effect on pitch and heave motions and vertical wave bending moment.

3.6 FORCE PREDICTION USING THE THREE-DIMENSIONAL APPROACH

The ability of Chapman's method for the prediction of transient motions of a floating body is demonstrated in [5] with the forced oscillation and free oscillation of a two-dimensional rectangle and wedge. Chapman extends the two-dimensional model to three dimensions as reported in [6]. The technique used by Chapman to check the three-dimensional model was to subject a semisubmerged sphere represented by 60 panels to vertical impulse - a unit step in heave velocity. The heave response time history that results is used to derive added mass and damping coefficients. These coefficients are compared with classical results of Havelock [37] and results of Kim [38] with very good correlation.

The basic ability of the three-dimensional method developed here to accurately predict hydrodynamic forces will be verified by two simple tests using a semisubmerged sphere at zero speed. A more comprehensive set of tests is recommended in the future. The two tests presented here should, however, clearly demonstrate that this method has considerable possibilities.

In the first test, the sphere will be subjected to forced vertical oscillations of unit amplitude at various frequencies in otherwise calm water. From the measured pressures, the total vertical time-varying force on the sphere will be calculated, which is then used to derive added mass and damping coefficients. The added mass and damping coefficients can then be compared to the results of Havelock [37].

In the second test, the sphere will be restrained and subjected to regular unidirectional waves of various frequencies and unit amplitude. Wave exciting forces in heave and surge can be measured and compared with analytical predictions given by Garrison [39].

The numerical particulars for both tests are given in Table 3.

The total vertical force was measured over two to three full cycles, depending on the frequency, up to about 20 seconds of simulation time with time steps of 0.2 seconds. The last cycle of the simulation was evaluated to derive added mass and damping coefficients using the method described below.

Table 3
Numerical Parameters

NPAN	= number of panels	= 72
L_x	= maximum length scale in X	= 0.25
l_x	= minimum length scale in X	= 2.5
L_y	= maximum length scale in y	= 0.25
l_y	= minimum length scale in y	= 2.5
T	= maximum time scale	= 12 sec.
g	= gravitational acceleration	= 1.0
ρ	= fluid density	= 1.0
a	= sphere radius	= 1.0
Δt	= time step size	= 0.1

The vertical displacement of the sphere from its mean position is given by

$$y = A \sin \omega t.$$

The linearized equation of motion for the sphere in forced heave is

$$(m + a)\ddot{y} + b\dot{y} + c\,y = Y_T.$$

where

m = mass of the cylinder,
a = hydrodynamic or "added" mass in heave,
b = damping coefficient against vertical motion,
c = hydrostatic restoring coefficient against vertical displacement, and
Y_T = external driving force.

The total hydrodynamic force acting on the body in the vertical direction is

$$Y_h(t) = a\,\ddot{y} - b\,\dot{y}.$$

Assuming this hydrodynamic force is a harmonic function in time and can be written as

$$Y_h(t) = \gamma \sin(\omega t - \varepsilon),$$

the hydrodynamic coefficients are obtained by equating these two expressions for Y_h,

$$a = \frac{\gamma \cos \varepsilon}{\omega^2 A}$$

and

$$b = \frac{\gamma \sin \varepsilon}{\omega A} .$$

The amplitude γ and phase shift β are obtained by comparing plots of the displacement and dynamic pressure force Y_h acting on the sphere as functions of time and measuring the shift in phase. A comparison of the results obtained with Havelock's results is shown in Figure 16. The solid line is Havelock's results. The numerical predictions are shown as circles. As can be seen from Figure 16, agreement is excellent.

Figure 17 shows a time sequence of the free surface disturbance induced by the vertical motion of the sphere. The circle represents the mean still waterline of the sphere. The radius of the sphere is 1.0 and the dimensions of the free surface area shown are 8 x 8. The wave number for the sequence is k = 1.0. The internal standing waves inside of the sphere are shown. (These internal standing waves should produce no net pressure.)

Wave Excitation Forces on Semisubmerged Sphere

The purpose of the second simulation was to investigate forces on the sphere when subjected to regular waves of different frequencies. Figure 18 shows the comparison between analytical predictions from Garrison [38] for the heave and surge wave exciting forces on a semisubmerged sphere. The agreement is excellent.

Figure 19 shows the diffraction wave system that develops from regular waves (h = 1.4) imposed on the sphere. The ambient wave amplitudes have been subtracted out of the free surface plot to more clearly exhibit the diffracted waves.

Other Tests

Attempts were also made to: (a) test a semisubmerged sphere in forced vertical oscillations at a uniform forward speed; and (b) to conduct forced oscillation tests for ellipsoids with L/B ratios of 8:1 and 4:1. In each of these cases, forces quickly increase to unrealistic levels, clearly indicating

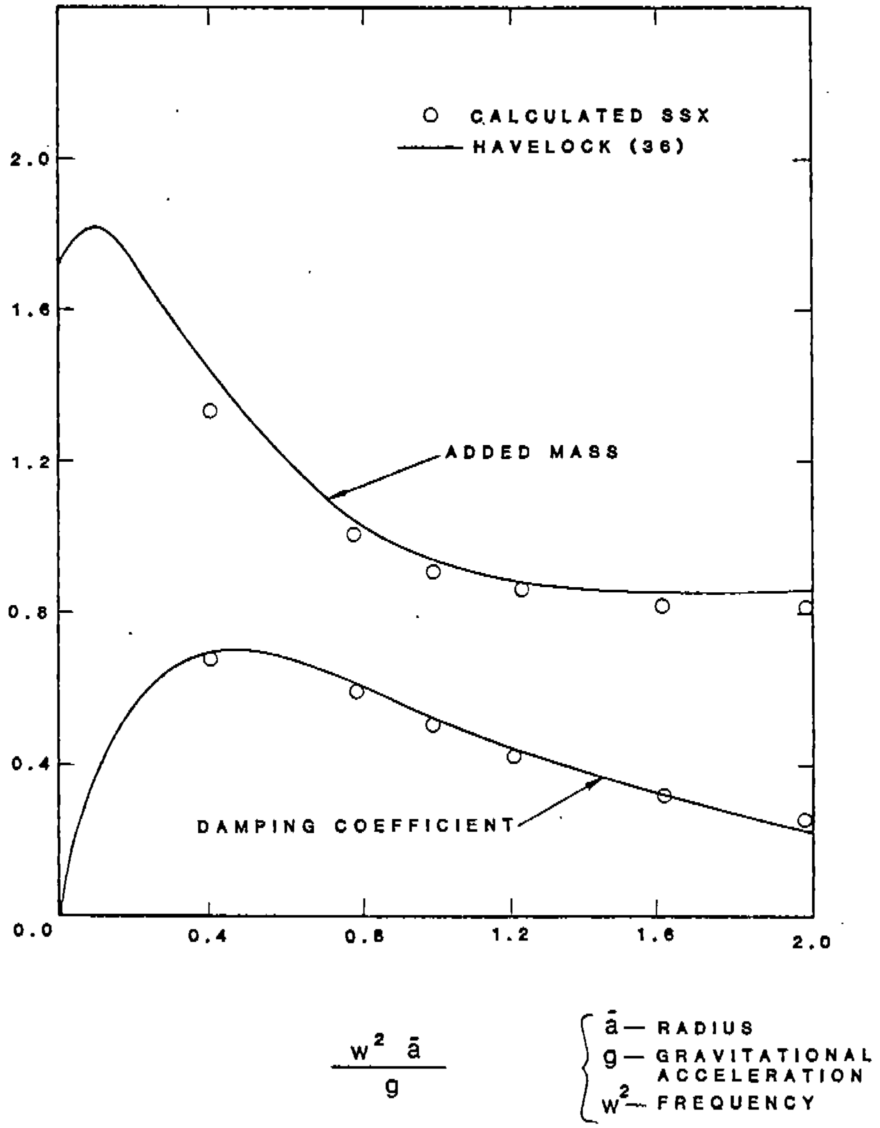

Figure 16 Added Mass and Damping Coefficients for semisubmerged sphere.

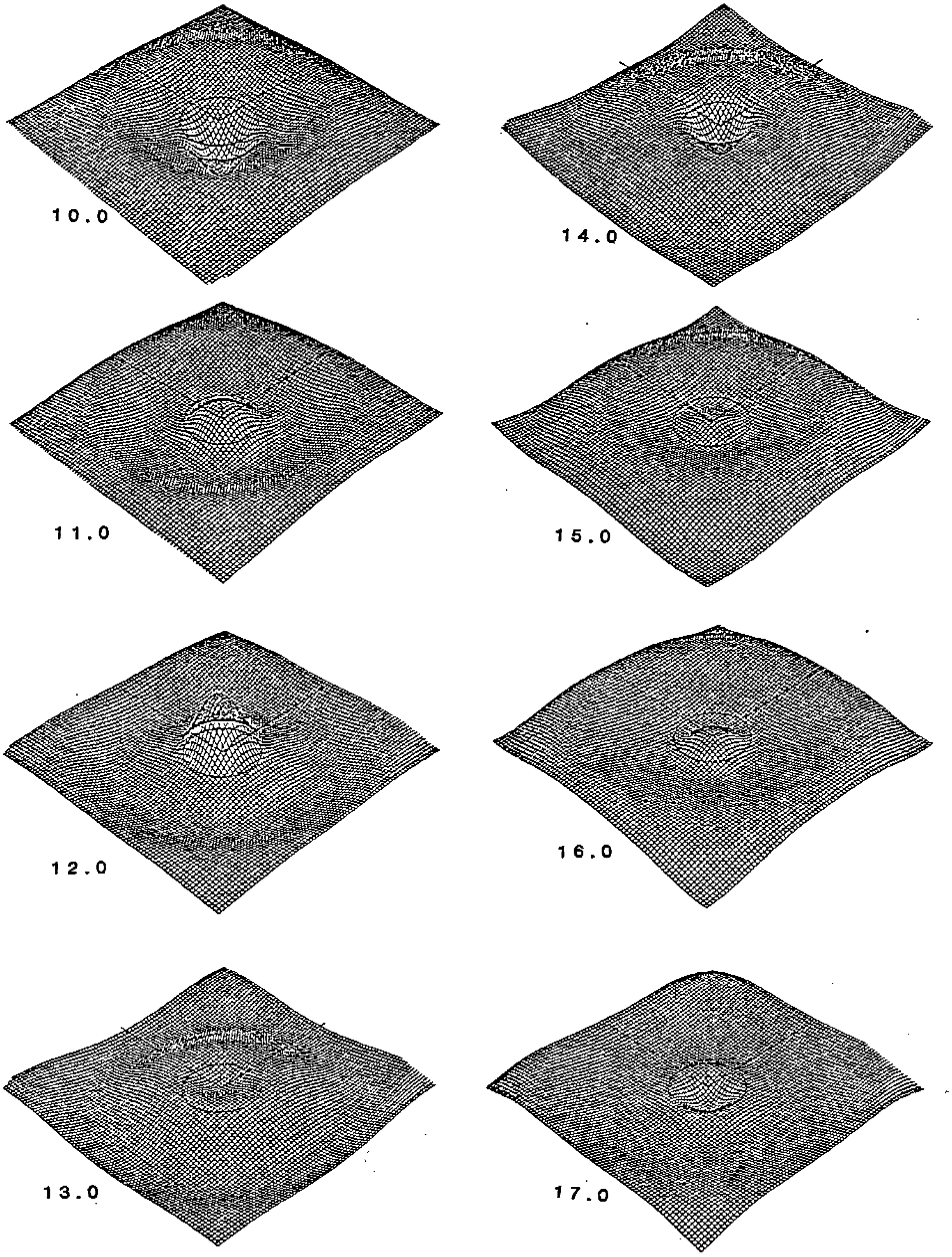

Figure 17 Radiated waves from vertically oscillating sphere at k = 1.0. Time = 10 - 17 secs after startup.

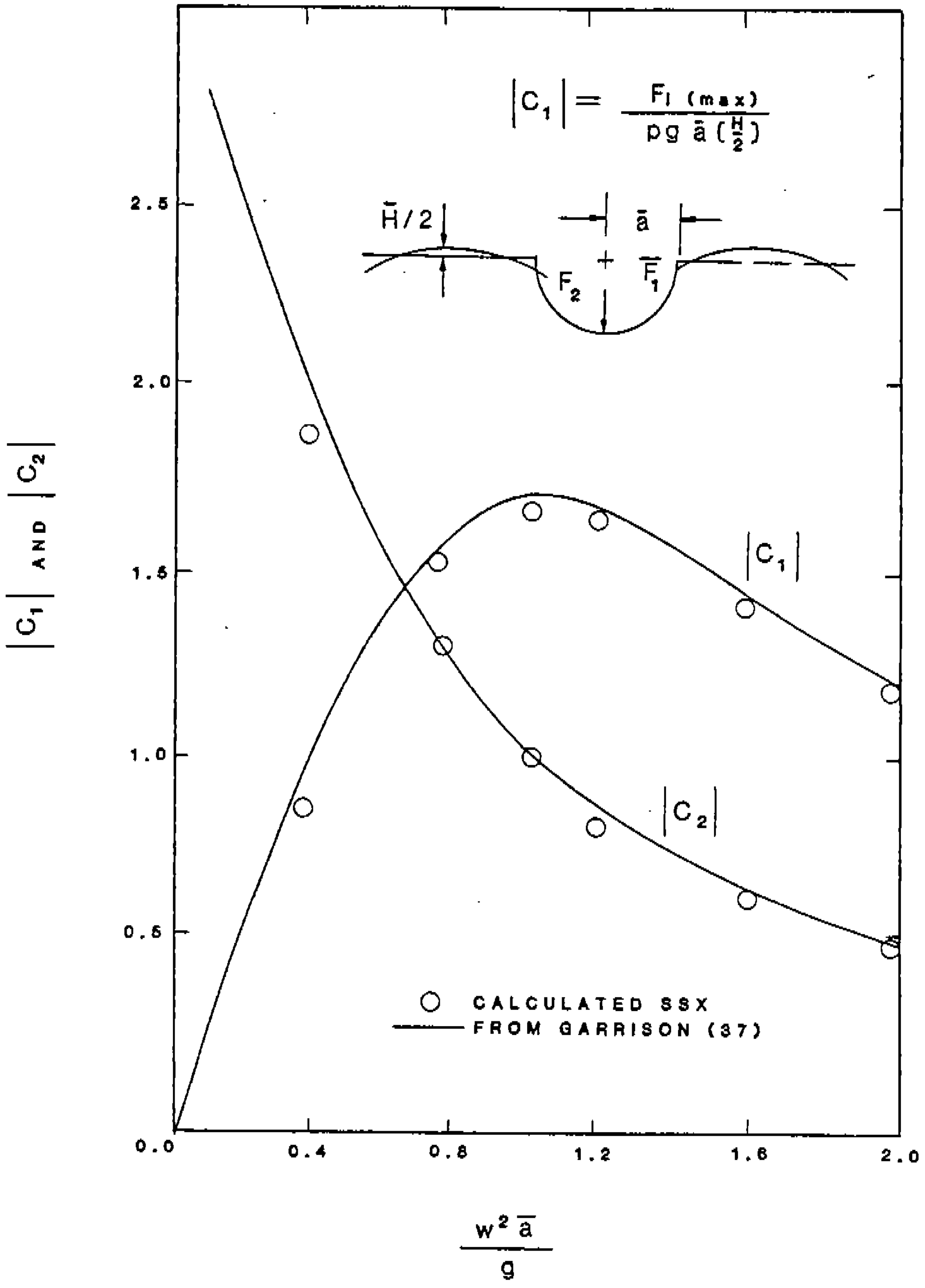

Figure 18 Wave excitation forces for semisubmerged sphere.

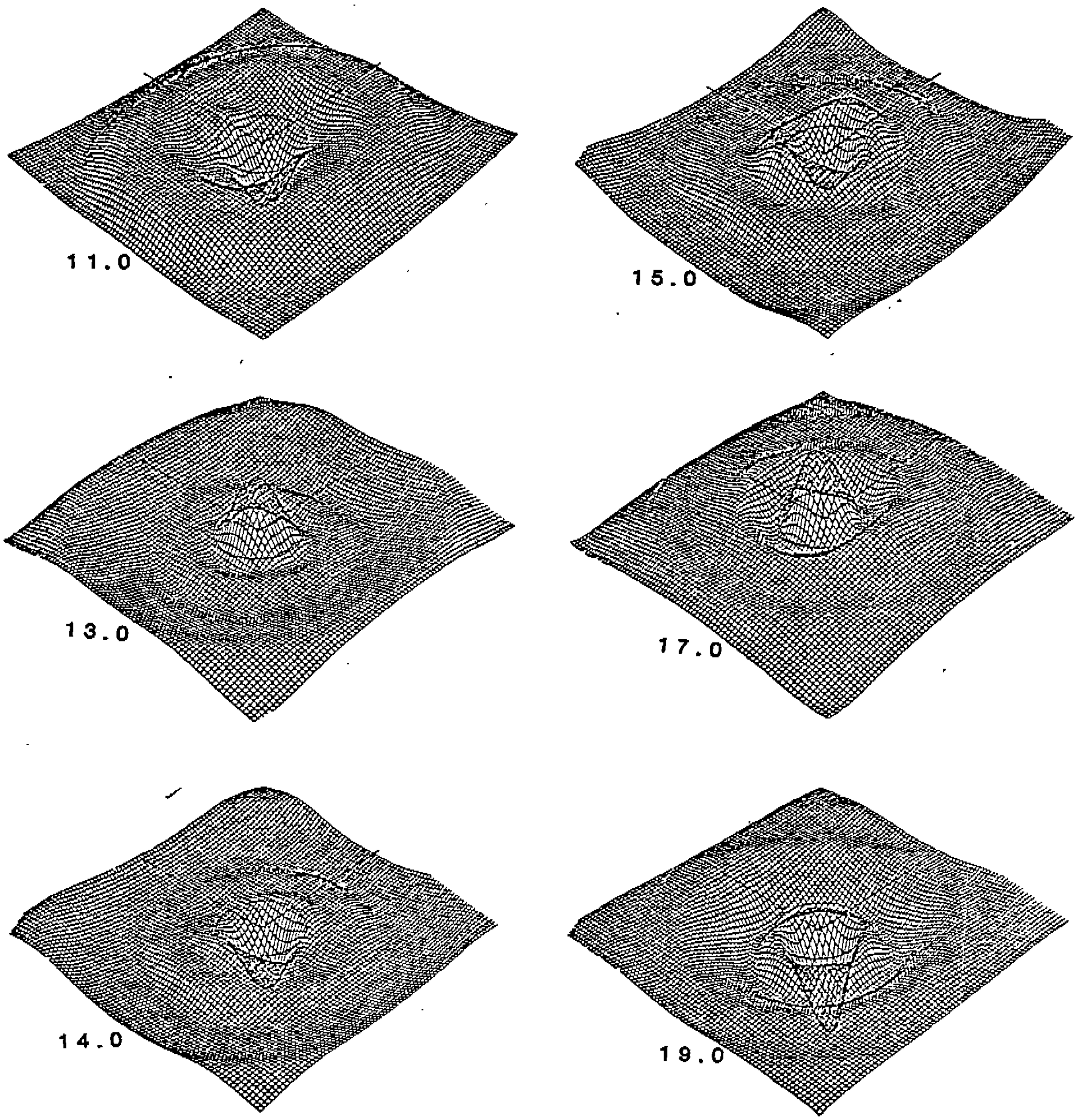

Figure 19 Diffracted waves from semisubmerged sphere, regular unidirectional waves, k=1.4. T=11-19 seconds.

that there is some problem associated with both forward speed and cases with shapes more closely representing a slender ship. The reasons for these "blow-ups" could not be readily identified.

3.7 LIMITATIONS, RUN TIMES AND COMPUTATIONAL REQUIREMENTS

Theoretical and Numerical Limitations

(1) The limitations of strip theory generally apply.

(2) The limitations of linear wave theory apply.

(3) The limitations of linearized hydrodynamic formulations (linearized free surface, separation of potentials, linearized Bernoulli equation, etc.) apply.

(4) Strip theory formulations here do not include terms related to the spatial derivative of damping coefficients (as found in "extended" SCORES).

(5) Flare force is computed using the relative free surface velocity in contrast to using a component of average subsurface wave velocity. Furthermore, free surface distortion effects are not included (diffractions, "pile-up," etc.).

(6) The ambient wave system for the three-dimensional analysis is formulated for unidirectional waves only.

(7) The P, E and X source influences matrices are computed only for the mean hull position in the present formulation. Details of the computation are presented in Volume II of this report.

(8) The use of either a pre-selected or time-varying estimate of added mass and damping coefficients for the ship motion-induced forces in irregular waves is an approximation.

(9) Pitch angle must not become large enough to cause the intersection of a station plane and the sea surface so that multiple regions or a closed contour in the station plane is defined [3].

Run Times

Typical run times for several simulations with strip theory derived forces and three-dimensionally derived forces are shown in Table 4 for a VAX 11/750 without a floating point processor.

It is estimated that reductions in run time by a factor of 5-10 would occur with any of the large mainframe CDC machines; and a reduction by a factor of 15-20 would occur with a supercomputer (CRAY or similar). It should be emphasized that the program is expensive to run, regardless of the type of machine on which it is installed.

Table 4

SIMULATION OF	VAX 11/750 CPU TIME
Motion and Hull Girder Response of 25-station ship in one regular wave. Steady state evaluated after 2 minutes real time with $\Delta t = 0.5$ sec.	1200 CPU seconds
Motion and Hull Girder Response of 25 station ship in irregular wave system of 10 components evaluated after 10 minutes of real time simulation. $\Delta t = 0.5$ sec.	16 CPU hours
Pressure prediction only of 120 panel body for 15 seconds real time $\Delta t = 0.1$ sec.	17 CPU hours

Computational Requirements

The computer programs were developed on a DEC VAX 11/750. The peak working set size was 250. The memory requirement for the completely integrated program (2D plus 3D) is 650K bytes. When the 3D subroutines are not used and only 2D simulations are used, the memory requirement is 250K bytes.

4.0 CONCLUSIONS AND RECOMMENDATIONS

A numerical method for the simulation of ship motions, hull girder loads and transient three-dimensional hydrodynamic pressures has been developed. The result is a computer program that is capable of evaluating nonlinear responses of surface ships associated with large amplitude motions. The fundamental limitation of the approach is the use of linear wave theory. The restrictions on the severity of the sea conditions that this limitation imposes is not yet known, although it is believed that the method would be appropriate for evaluating motions and hull girder loads in moderately severe sea states.

Furthermore, the three-dimensional technique developed is capable of predicting transient hydrodynamic pressures on time-varying hull shapes moving with arbitrary motions and forward speed. Further effort is required to fully realize these capabilities. An outline of the suggested work is given in the next section.

Although the method presented here and the associated computer program need additional attention to advance them from the investigative and developmental stage at which they now are, some progress has been made toward a more advanced method for ship motion and wave load prediction. There presently exists no other program for the time domain simulation of ship motions and loads in six degrees of freedom for a full range of headings. Nor is there any other program capable of predicting three-dimensional distributed hydrodynamic pressures in the time domain for arbitrary ship motions in waves.

The CAPSIZE program [3,29] and the NSUP program [6] which were used as starting points for the present method have been extended or modified in the following principal ways:

- Actual strip theory formulation used for hydrodynamic forces instead of simplified and approximate methods for added mass, damping and wave exciting forces
- Hull girder loading prediction added, including "flare force" effect
- The addition of the incident wave potential into the three-dimensional formulation
- A number of computational and numerical changes associated with the calculation of two-dimensional hydrodynamic coefficients, the evaluation of two-dimensional hydrodynamic forces, and the evaluation of pressure in the wave above the still water level.

The bulk of the effort in this project was committed to the concept development of a suitable method and the development and testing of the computer code. Because of the particularly time-consuming nature of debugging and testing simulation programs, the extent of the validation effort has been limited to the minimum number of runs needed to verify the capabilities of the computer programs and evaluate the basic theory and numerical techniques when possible. A full and extensive validation of the methods and codes would require the type of test program typically associated with actual (physical) model testing, with a level of effort representative of a major project in itself. An outline of a simulator test program is provided in Appendix A as a possible basis for the same type of studies reported in SSC-246* [40] and SSC-271** [41] to validate and further investigate SCORES after its initial development.

Recommendations

The work presented herein represents an initial level of investigation and development. The numerical methods and computer programs require further testing, validation and improvement. The following list of recommendations indicates areas where further effort is required or desirable in the near-term, mid-term or far-term.

Near-Term

1. A comprehensive test and validation program is required, an outline of which is given in Appendix A for the two-dimensional method.
2. The method or program coding for the computation of Froude-Krylov forces or strip-theory derived forces should be corrected or improved to provide better predictions, specifically heave motion.
3. The numerical methods and coding associated with the three-dimensional technique should be improved. Specifically, the following areas will be investigated and modified or improved:
    (a) Investigate the method used for computing the source panel velocity array and the source potential influence array. The so-called exact method of Smith and Hess might be improved. More efficient methods might also be incorporated. There may be reason to suspect that the problems with the ellipsoid tests attempted are due to numerical problems in this area.
    (b) Investigate the theoretical formulations that include some forward speed effects. Investigate the numerical/computational representation of the theory. Check the coding dealing with forward speed aspects. Identify any deficiencies and make the appropriate changes and additions.

---

* Kaplan, P., et al, "Theoretical Estimates of Wave Loads on the SL-7 Containership in Regular and Irregular Waves," SSC-246, 1974.

** Kaplan, P., et al, "A Correlation Study of SL-7 Containership Loads and Motions - Model Tests and Computer Simulations," SSC-271, 1977.

(c) Examine the feasibility and necessity of applying a different free-surface condition interior to the hull. As Chapman suggested in [6], instead of using the linearized free-surface condition over the entire free surface, the surface elevation interior to the hull can be constrained to be uniformly zero by placing surface panels over this region. This would suppress standing waves which are excited in the hull interior as a by-product of the potential flow method of simulation.

(d) Investigate more efficient ways of computing. For example, if we can assume that motions are sufficiently small, then certain values which are functions, sums and products of the various source influence arrays might be precomputed and stored, rather than computing them inside computational loops over kx and ky wave numbers for each panel.

(e) Demonstrate the accuracy and capabilities of the program by comparing numerical results with predictions from theory or other programs for various shapes at zero speed (sphere, ellipsoids, barge shape, ship shape) and for those shapes with forward speed. The basis for these comparisons would be added mass, damping and wave excitation coefficients.

(f) Carry out sufficient testing to develop an experience base for setting the parameters for the spectral wave representation.

(g) Investigate and demonstrate the ability of the method to properly predict the distributed hydrodynamic loads on the underwater hull.

(h) Examine the feasibility and identify the numerical techniques that would allow for the underwater hull form to change with time.

Mid-Term

4. Appropriate specific methods to predict slamming loads and green water loads should be investigated and incorporated in the present program.

5. An advanced statistical processing routine should be incorporated in the program to derive frequency response spectra, as well as probability density functions and cumulative distributions of maxima and minima.

6. A method should be investigated and appropriate subroutines incorporated to structurally model the ship hull girder in order to predict transient non-rigid body response.

7. Improved coding for the computation of sectional mass distribution properties should be written.

8. Coding for the computation of appendage damping should be written.

Far-Term

9. Linear wave theory was assumed for the present model and is consistent with the linearized free-surface formulations associated with strip theory and the three-dimensional method. It may be worthwhile, however, to pursue the use of a nonlinear wave theory to generate large amplitude deterministic unidirectional waves. Although the nonlinear wave could not be applied, in a strict sense, within the hydrodynamic formulations as they exist in the two- and three-dimensional formulations, they could be accounted for in an approximate way. Hydrostatic and dynamic buoyancy forces could be accounted for "exactly." Ship motion-induced hydrodynamic forces could be approximated as they are in the present version. Diffraction forces could be possibly estimated by using the same method as now exists but representing the nonlinear wave in the vicinity of the ship with a similarly proportioned linear wave. For the transient pressure simulation, the exact nonlinear wave kinematics could be used to prescribe part of the body boundary condition at the center of each panel.

10. The use of the impulse response function and convolution integral to predict ship motion-induced damping forces should be investigated and appropriate subroutine(s) included in the present version. This would be a partial alternative for the "characteristic frequency" scheme now in place.

The efforts listed under each of the above categories can be grouped into the three separate projects with the following approximate level of effort suggested:

Near-Term Project	3000 manhours
Mid-Term Project	2000 manhours
Far-Term Project	2000 manhours

The near-term project should be considered a necessary requirement in order that the methods and computer programs initiated here are sufficiently developed and proven for practical use.

REFERENCES

1. Smith, C.S., "Measurement of Service Stresses in Warships," Conference on Stresses in Service, Inst. of Civ. Engrs., March 1966.

2. Meek, M., et al, "The Structural Design of the O.C.L. Container Ships," Trans. RINA, Vol. 114, 1972.

3. Oakley, O.H., Paulling, J.R. and Wood, P.D., "Ship Motions and Capsizing in Astern Seas," Tenth Symposium on Naval Hydrodynamics, MIT, Cambridge, pp. 297-350, ONR Report ACR-204, June 1974.

4. Raff, A.I., "Program SCORES - Ships Structural Response in Waves," SSC-230, 1972. NTIS AD 752568.

5. Chapman, R., "Large Amplitude Transient Motion of Two-Dimensional Floating Bodies," Journal of Ship Research, Volume 23, No. 1, March 1979.

6. Chapman, R., "Time Domain Method for Computing Forces and Moments Acting on Three-Dimensional Surface Piercing Ship Hulls with Forward Speed," SAI Report No. 461-80-560-CJ, September 1980.

7. Chang, M.S., "Computations of Three-Dimensional Ship Motions with Forward Speed," Proceedings, Second International Conference on Numerical Ship Hydrodynamics, Berkeley, California, September 1977.

8. Pettersen, B., "Calculation of Potential Flow About Three-Dimensional Bodies in Shallow Water with Particular Application to Ship Maneuvering," Norwegian Institute of Technology, Trondheim, Norway, (Unpublished), 1981.

9. von Kerczek, C., "Numerical Solution of Naval Free Surface Hydrodynamics Problems," First International Conference on Numerical Ship Hydrodynamics, Bethesda, October 1975.

10. McCormick, S. and Thomas, J.W., "Multigrid Methods Applied to Water Wave Problems," Third International Conference on Numerical Ship Hydrodynamics, Paris, June 1981.

11. Bai, K.J., "A Localized Finite Element Method for Three-Dimensional Ship Motion Problems," Third International Conference on Numerical Ship Hydrodynamics, Paris, June 1981.

12. Bai, K.J. and Yeung, R.W., "Numerical Solutions to Free Surface Flow Problems," The Tenth Symposium on Naval Hydrodynamics, Office of Naval Research held at MIT, Cambridge, 1974.

13. Chen, H.S. and Mei, C.C., "Oscillations and Wave Forces in a Man-Made Harbor in the Open Sea," The Tenth Symposium on Naval Hydrodynamics, Office of Naval Research held at MIT, Cambridge, 1974.

REFERENCES (continued)

14. Seto, H. and Yamamoto, Y., "Finite Element Analysis of Surface Wave Problems by a Method of Superposition," The First International Conference on Numerical Ship Hydrodynamics, DTNSRDC, Bethesda, pp. 49-70, (1975).

15. Yue, D.K.P., Chen, H.S. and Mei, C.C., "Three Dimensional Calculations of Wave Forces by a Hybrid Element Method," The Eleventh Symposium on Naval Hydrodynamics, Office of Naval Research, pp. 325-332, (1976).

16. Chowdbury, P.C., "Fluid Finite Elements for Added-Mass Calculations," International Shipbuilding Progress, Volume 19, No. 217, pp. 302-309, (1972).

17. Beale, J.T., "Eigenfunction Expansions for Objects Floating in an Open Sea," Communications on Pure and Applied Mathematics, Volume 30, pp. 283-313, (1977).

18. Licht, C., "Time-Dependent Behavior of Floating Bodies," Third International Conference on Numerical Ship Hydrodynamics, Paris, June 1981.

19. Jami, A., "Numerical Solving of Transient Linear Hydrodynamics Problems by Coupling Finite Elements and Integral Representation," Third International Conference on Numerical Ship Hydrodynamics, Paris, June 1981.

20. Skjordal, S.O., "A Rational Strip Theory Approach for the Evaluation of Springing," Norwegian Institute of Technology, Trondheim, January 1978.

21. Troesch, A.W., "The Diffraction Potential for a Slender Ship Moving Through Oblique Waves," Department of Naval Architecture and Marine Engineering, The University of Michigan, Report No. 17, 1976.

22. Bourianoff, G.I. and Penumalli, B.R., "Final Report on Computational Prediction of Ship Motion in Confined Waters Using Inertial Marker Particles," MARAD Report MA-RD-940-77084A, November 1978.

23. Fritts, K.J. and Boris, J.P., "Transient Free-Surface Hydrodynamics," Second International Conference on Numerical Ship Hydrodynamics, Berkeley, September 1977.

24. Nichols, B.D. and Hirt, C.W., "Nonlinear Hydrodynamic Forces on Floating Bodies," Second International Conference on Numerical Ship Hydrodynamics, Berkeley, September 1977.

25. Chuang, S., Schroeder, A.E. and Wybranice, S., "Structural Seaworthiness Digital Computer Program (ROSAS)," DTNSRDC Report No. 77-0001, May 1977.

26. Meyerhoff, W.K. and Schlacter, G., "An Approach for Determination of Hull-Girder Loads in a Seaway Including Hydrodynamic Effects," Ocean Engineering, Volume 7, No. 2, 1980.

REFERENCES (continued)

27. Borresen, R. and Tellsga, R., "Nonlinear Response of Vertical Motions and Loads in Regular Head Waves, Part I (Text) and Part II (Figures)," DNV Report 79-1097, December 1979.

28. Oliver, J.C., "Development of an Advanced Method for Ship Motion and Wave Load Prediction - Phase I Report," Giannotti & Associates, Inc. Report No. 80-083-001, January 1982.

29. Fallen, W.J., Hwang, Y.L., Liquori, J.L., Paulling, J.R., Viseneau, G. and Wood, P.D., "Model Tests and Numerical Simulation of Ship Capsizing in Following Seas," USCG Report No. CG-D-08-81.

30. Chakrabarti, S.K., "Laboratory Generated Waves and Wave Theories," Journal of Waterway, Port Coastal and Ocean Engineering, Volume 106, August 1980.

31. Kaplan, P. and Sargent, T.P., "Further Studies of Computer Simulation of Slamming and Other Wave-Induced Vibratory Structural Loadings on Ships in Waves," SSC Report 231, 1972.

32. Hess, J.L. and Smith, A.M.O., "Calculation of Non-Lifting Potential Flow About Arbitrary Three-Dimensional Bodies," Douglas Aircraft Company, Report No. E.S. 40622, March 1962.

33. Zielinski, T.E., "User Manual for Program SCORES, Second Part of the USCG Ship Motion Program," HMC Report CG-M-8-80, July 1980.

34. Gerritsma, J. and Beukelman, W., "Analysis of the Modified Strip Theory for the Calculations of Ship Motions and Wave Bending Moments," International Shipbuilding Progress, Volume 14, August 1967.

35. Dalzell, T.F. and Chiocco, K.J., "Wave Loads in a Model of the SL-7 Containership Running on Oblique Headings in Regular Waves," SSC-239, 1973.

36. Loukakis, T.A., "Computer-Aided Prediction of Seakeeping Performance in Ship Design," MIT Report No. 70-3, August 1970.

37. Havelock, T., "Waves Due to a Floating Sphere Making Heaving Oscillations," Proceedings Royal Society of Architects, 231, 1955, pp. 1-7.

38. Kim, W.P., "On the Harmonic Oscillations of a Rigid Body on a Free Surface," Journal of Fluid Mechanics, 1964.

39. Garrison, C.J., "Hydrodynamics of Large Objects in the Sea, Part I - Hydrodynamic Analysis," Journal of Hydronautics, Vol. 8, January 1974, pp. 5-12.

APPENDIX A

NUMERICAL SIMULATOR TEST PROGRAM

1.0 OBJECTIVE

The objective is to fully exercise the computer program SSX, using as a test vehicle a ship for which model tests have been carried out for a range of headings, speeds and wave heights, with emphasis on lower speeds and higher wave heights.

2.0 TEST SERIES

2.1 TEST SERIES A - REGULAR WAVE TESTS

The ship is subjected to unit amplitude unidirectional regular waves at five headings, four Froude numbers, and at six frequencies corresponding to various wavelength/ship-length ratios. The critical parameters are:

Wavelength Parameters (λ/L) at $\beta = 0\ 180^0$	0.5, 0.75, 1.0, 1.25, 1.5, 2.0 (or equivalent at other headings)
Froude Numbers (F_n)	0.0, 0.05, 0.10, 0.15
Headings (β)	180, 135, 90, 45, 0
Measurements	- (a) C.G. Motions (b) Midships Hull Girder Loads (c) Sectional Forces Including Hydrostatic, Froude-Krylov, Wave Exciting and Ship Motion Induced Forces
Total Number of Runs	120

2.2 TEST SERIES B - LINEARITY CHECKS (NO FLARE FORCE)

The ship is subjected to unidirectional regular waves of four different heights, six frequencies, two speeds and five headings. The critical parameters are:

Wave Height (H_W)	0.2D, 0.4D, 0.6D, 0.8D (Where D is Ship Depth)
Wavelength Parameters (λ/L)	0.5, 0.75, 1.0, 1.25, 1.5 (With the Constraint $H/\lambda \leqslant .1$)

Froude Numbers (F_n)	0.0, 0.10
Headings	180, 135, 90, 45, 0
Measurements	(a) C.G. Motions (b) Midships Hull Girder Loads
Total Number of Runs	200

2.3 TEST SERIES C - LINEARITY CHECK WITH FLARE FORCE INCLUDED

The ship is subjected to unidirectional regular waves in head and following directions, three speeds and five frequencies and four wave heights. The critical parameters are:

Wavelength Parameters (λ/L)	0.5, 0.75, 1.0, 1.25, 1.5
Froude Numbers (F_n)	0.0, 0.10, 0.15
Headings	180, 0
Wave Heights	0.2D, 0.4D, 0.6D, 0.8D
Measurements Taken	(a) C.G. Motions (b) Midships Hull Girder Loads (c) Sectional Forces
Total Number of Runs	120

2.4 TEST SERIES D - LINEARITY CHECK WITH DRAFT-DEPENDENT HYDRODYNAMIC COEFFICIENT SELECTION

The ship is subjected to unidirectional regular waves of four different heights, six frequencies, two speeds and three headings. The critical parameters are:

Wave Height	0.2D, 0.4D, 0.6D, 0.8D
Wavelength Parameters	0.5, 0.75, 1.0, 1.25, 1.5, 2.0
Froude Numbers	0.0, 0.10
Headings	180, 135, 90
Measurements	(a) C.G. Motions (b) Midships Hull Girder Loads (c) Sectional Forces
Total Number of Runs	144

2.5 TEST SERIES E - IRREGULAR WAVE TESTS

The ship is subjected to four unidirectional irregular systems with frequency spectra corresponding to a two-parameter Bretschneider formulation, one speed, five headings. The critical parameters are:

Significant Wave Heights (H_s)	10, 20, 30, 40 ft.
Spectal Modal Frequencies	Most Probable
Headings	180, 135, 90, 45, 0
Froude Number	0.10
Measurements Taken	(a) C.G. Motion Statistics (b) Midship Hull Girder Load Statistics
Total Number of Runs	20

In Test Series E, the frequency at which the 2-D hydrodynamic coefficients are selected is fixed according to the peak of the response spectra obtained from a linear frequency domain analysis.

2.6 TEST SERIES F - IRREGULAR WAVE TESTS WITH DRAFT-DEPENDENT HYDRODYNAMIC COEFFICIENT SELECTION

The ship is subjected to three unidirectional wave systems, one speed, three headings. Two-dimensional coefficients are selected as a function of instantaneous sectional draft. The critical parameters are:

Significant Wave Heights (H_s)	20, 30, 40
Headings	180, 135, 90
Froude Number	0.10
Measurements Taken	(a) C.G. Motion Statistics (b) Midship Hull Girder Load Statistics
Total Number of Runs	6

For Test Series F, the ship motion hydrodynamic coefficients are selected as with Test Series E.

2.7 TEST SERIES G - IRREGULAR WAVE TESTS (WITH VARYING FREQUENCY)

Same as Test Series D, except the 2-D hydrodynamic coefficients associated with ship motion-induced forces are selected according to the most recent measured zero crossing periods of motion.

Additional Measurement Taken	Automatically - Selected Frequencies Used for Heave, Sway and Roll 2-D Coefficients
Total Number of Runs	20

3.0 SCOPE OF WORK

The following table is used to estimate computer costs using a VAX 11/750 with below average rates:

Test Series	No. Runs	CPU Hours Per Run	Subtotal Hours
A	120	1	120
B	200	1	200
C	120	1	120
D	144	1	144
E	20	32	640
F	6	32	192
G	20	32	640
			2056 CPU HOURS

If we assume a rate for CPU time on a commercial VAX 11/750 is at least $60/CPU Hour, the total computer charge would be at least $123,360.00. There are also a significant amount of manhours involved in preparing the input data. Analyses and evaluation of the resultant data would also require at least 1000 manhours.

Even if the number of runs could be reduced by half, a full set of validation runs for SSX requires a surprisingly large amount of effort and computational resources.

GIANNOTTI & ASSOCIATES, INC.

NAVAL ARCHITECTS
OCEAN ENGINEERS
MARINE ENGINEERS

A TIME DOMAIN SIMULATION METHOD FOR SHIP MOTION AND WAVE LOAD PREDICTION

VOLUME II

USER'S MANUAL FOR PROGRAMS HYDREX2, HYDREX3 AND SSX

James C. Oliver
GIANNOTTI & ASSOCIATES, INC.

under

Department of Transportation
United States Coast Guard
Contract No. DTCG23-80-C-20032

U.S. Coast Guard Headquarters
Washington, D.C.

May 1984

703 GIDDINGS AVENUE, SUITE U-3 • ANNAPOLIS, MARYLAND 21401 • (301) 268-0030 • D.C. 261-1031
1725 Jefferson Davis Highway, Suite 912 • Arlington, Virginia 22202 • (703) 892-2360
1847 Berkeley Way • Berkeley, California 94703 • (415) 841-5875

TABLE OF CONTENTS

Page

1. INTRODUCTION

Program SSC performs a time-domain simulation of ship motions, hull girder loads and distributed pressures. Two preprocessing programs - HYDREX2 and HYDREX3 - perform certain calculations in preparation for a simulation. Program HYDREX2 computes two-dimensional sectional hydrodynamic coefficients and HYDREX3 calculates several arrays associated with the three-dimensional source representation of the hull. Program input includes a description of the ship's hull by sectional offsets, representation of the hull by quadrilateral or triangular panels, ship geometric and inertial particulars, operational and environmental specifications, and simulation parameters and job options. Output includes a time history plus statistical summaries of heave, pitch, sway, roll, yaw, vertical and horizontal shear and bending moments, torsional moments and dynamic pressures at centers of hull surface panels. The present version of the program contains several features to facilitate experiments with alternative techniques. The program SSX can be run in a batch or interactive mode. The three programs are written in FORTRAN 77 and developed on a DEC VAX 11/750 computer.

This volume (Volume II) provides instructions to the user for performing computations with SSX, HYDREX2 and HYDREX3. A technical description of the model formulation and solution procedures are provided in Volume I.

Program structure and computational overview is first given in Section 2. Program software specifications are described in Section 3. Run procedures are given in Section 4 with input and output variable descriptions. A sample case in Section 5 illustrates code application. Program listings are contained in Appendix A.

2. GENERAL STRUCTURE AND COMPUTATIONAL OVERVIEW

A simplified schematic of the basic flow of information is provided in Figure 1. It shows the major sequence of information processing. Three data files are initially required. The basic information file [BIF] contains basic geometric and mass properties of the ship, operational and environmental parameters, and calculation specifications, simulation parameters and job options. The ship offset file [OFF] gives station offset data. The hull panel file [PAN] contains nodal points and connectivity relationships to represent the hull surface by quadrilateral and/or triangular panels.

Program HYDREX2 uses the [BIF] file and [OFF] file to calculate two-dimensional hydrodynamic added mass and damping coefficients using the Frank Close-Fit method. HYDREX2 places the results of its computation in a two-dimensional coefficient file [COF]. It also provides a printout of its results.

Program HYDREX3 uses the [PAN] file to compute panel center coordinates, areas, normals, source density matrix [E], panel pressure matrix [PP] and forward velocity pressure matrix [PX]. The results are placed in a data file [MAT]. HYDREX3 need not be run if it is anticipated that distributed pressures will not be required.

Program SSX uses the [BIF] data file, the [COF] data file, the [MAT] data file (if a pressure distribution is desired), and the [OFF] data file. SSX output includes an input data summary, time domain results and response statistics.

A simplified flow schematic of the SSX program is given in Figure 2. The main program routine SSX initially asks for the names of the [BIF], [OFF], [COF] and [MAT] data files. Subroutine READIN is called to load the necessary variables and arrays using data read in from the four data files.

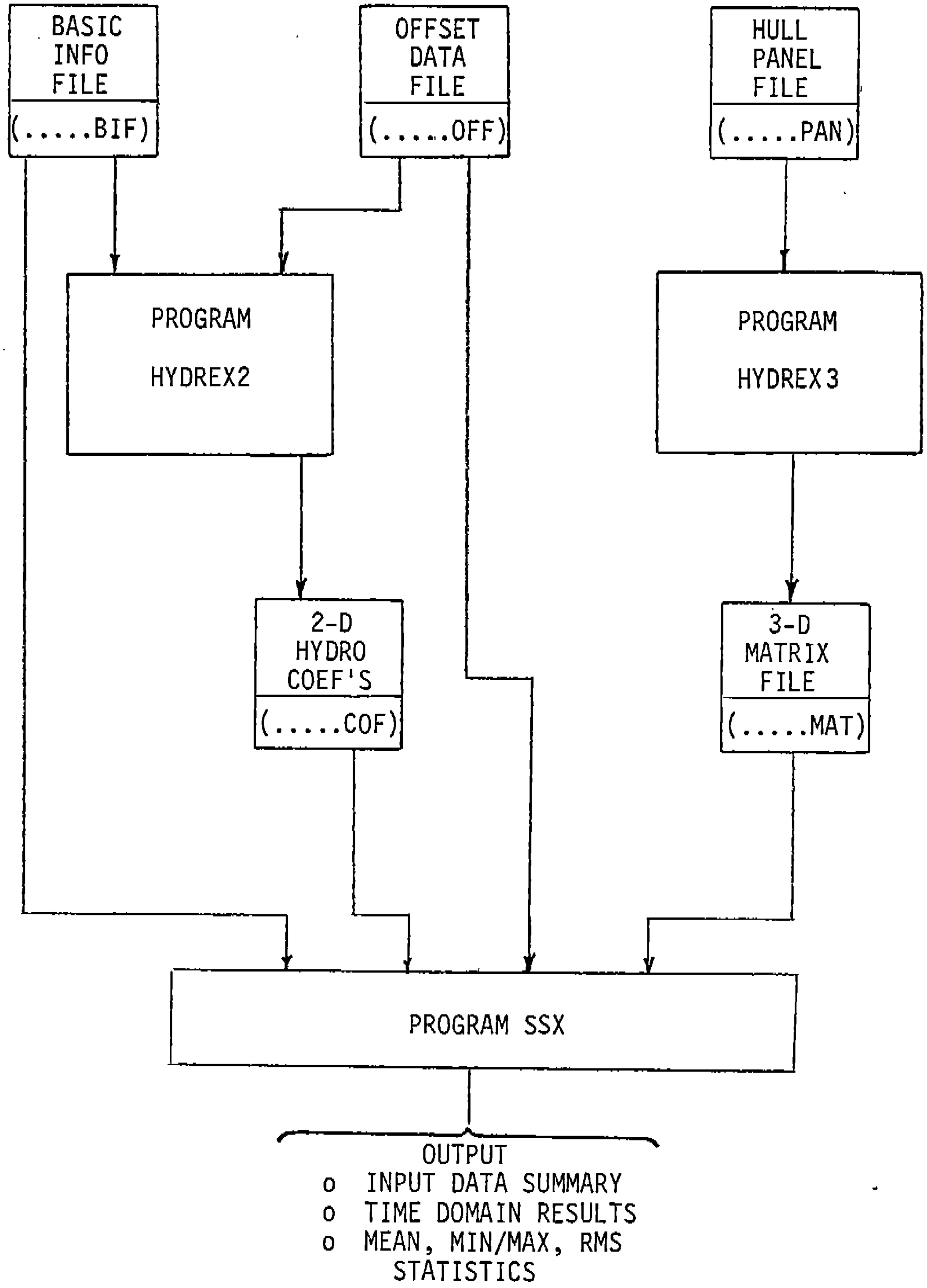

FIGURE 1 - BASIC INFORMATION FLOW

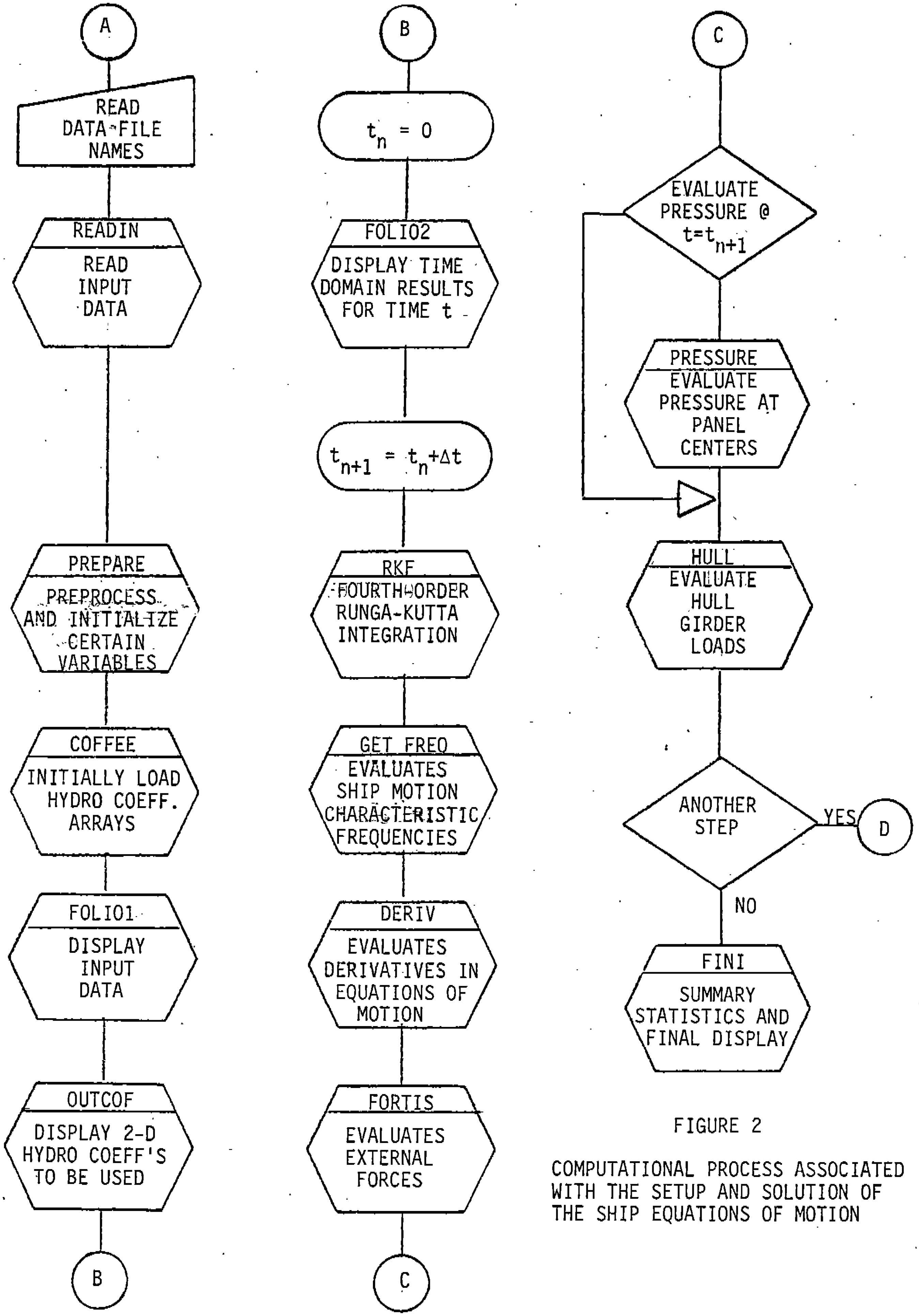

FIGURE 2

COMPUTATIONAL PROCESS ASSOCIATED WITH THE SETUP AND SOLUTION OF THE SHIP EQUATIONS OF MOTION

At this point all data has been entered, and subroutine PREPARE is called to initialize certain arrays and perform some computations in preparation for the simulation. Next, subroutine COFFEE is called to load a set of arrays that contain the initial set of two-dimensional hydrodynamic coefficients associated with the mean still water draft, the frequencies of encounter of the wave components and the predicted characteristic frequencies of ship motion.

The program then calls FOLIO1 to display the basic input data and some computed preparatory information. At the option of the user, subroutine OUTCOF can be called to display the initial set of two-dimensional hydrodynamic characteristics.

Now the actual simulation is ready to begin. At each time step, the subroutine FOLIO2 is called to display motion and loads response values for that time. Thus FOLIO2 is initially called at t = 0. The program then proceeds with the simulation. Within the time step loop in the main program, three subroutines are called - RK4, HULL and PRESSURE. The subroutine RK4 sets up and solves the equations of motions, the subroutine HULL evaluates hull girder loads, and the subroutine PRESSURE evaluates dynamic pressure at the center of each hull surface panel. PRESSURE is only called for a user-selected period of time within the simulation. Both the RK4 and PRESSURE subroutines lead into the large number of subroutines which will be described in the next two paragraphs. At the end of each time step, the time is compared to the user specified stop time. The loop is updated if the stop time has not been reached. If it has been, the simulation stops, and subroutine FINI is called to provide summary statistics.

Figure 3 provides a simplified representation of the computational process associated with the solution of the equations of motion. The subroutine RK4 calls DERIV four times for each time step according to the fourth order Runga-Kutta scheme. The subroutine DERIV sets up the equations of motion in the form of four first order ordinary differential equations. It evaluates the derivatives to be integrated by RK4. The subroutine DERIV calls subroutine FORTIS to evaluate external forces and moments, which include hydrostatic and hydrodynamic forces calculated by subroutine AQUA2D. The subroutine AQUA2D calls a number of other supporting subroutines and functions. MOCHA evaluates 2-D hydrodynamic coefficients when they are to be a function of sectional draft and/or ship motion characteristic frequencies. KRYLOV evaluates ponential and trigonometric integrals for each section. Functions ETAF and ETABAR evaluate the wave surface elevation.

The computational process associated with hull surface pressure calculations is provided in Figure 4. Subroutine PRESSURE calls ACPTR which computes the pressures and normal acceleration at panel centers induced by the free surface. Next subroutine ZBLACN determines the relative normal acceleration at each panel by subtracting the free surface induced accelerations and incident wave field accelerations from the hull kinematical acceleration. The time derivative of panel source strengths are also evaluated in ZBLACN. Next, subroutine POTB is called to obtain body-induced pressures. The body-induced pressure is added to the free-surface-induced pressure at each panel. The final subroutine is CFSR which advances the free surface representation in preparation for the next time step.

If subroutines to evaluate bottom slamming and green water loads were to be included in later modifications to the program SSX, they should be called from the subroutine AQUA2D. Any force as a function of time, displacement velocity and position on the ship can be evaluated from the information available at any instant within AQUA2D.

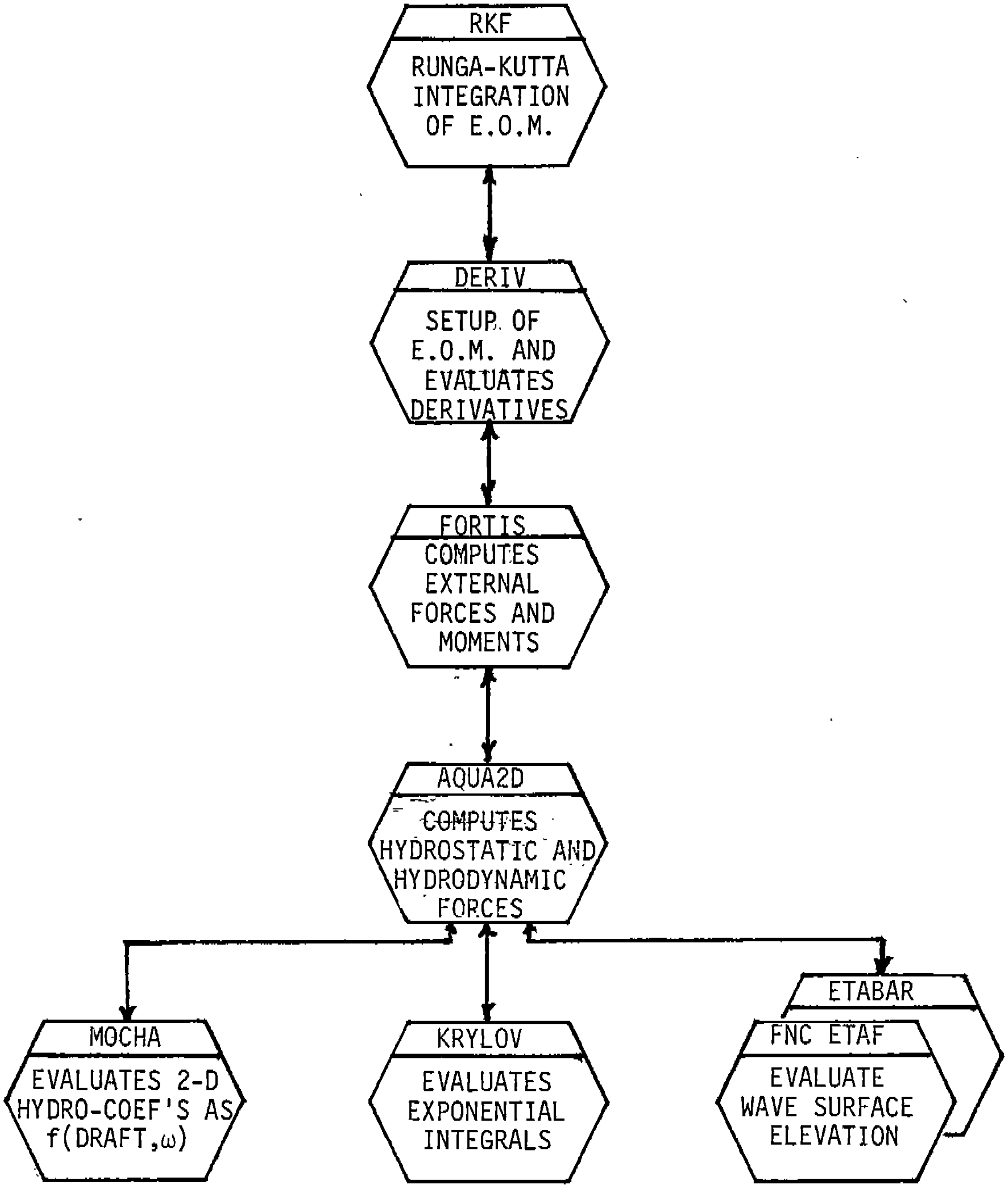

FIGURE 3

COMPUTATIONAL PROCESS ASSOCIATED WITH THE SETUP AND SOLUTION OF THE SHIP EQUATIONS OF MOTION

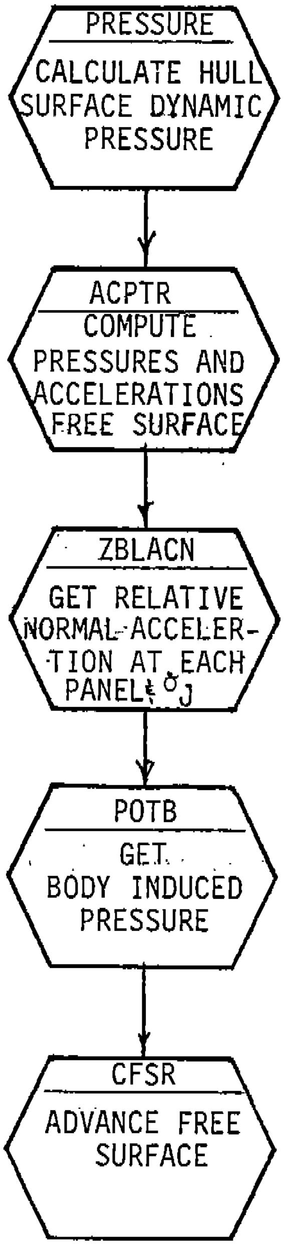

FIGURE 4

COMPUTATIONAL PROCESS ASSOCIATED WITH HULL PRESSURE CALCULATIONS

3. GENERAL SOFTWARE SPECIFICATIONS

The three codes conform to ANSI X3.9-1978 FORTRAN 77. They were developed on a Digital Equipment Corporation VAX 11/750 computer with a working set size of 250 pages. The only obvious machine depending coding are OPEN statements to open and assign files.

Rather than completely developing new codes, two previously developed and tested codes were used as a basis for the present three codes. The CAPSIZE program written at the University of California, Berkeley by Paul Wood was used as a starting point. The basic variable assignments used in that program were retained, as was the basic computational scheme for the equations of motion. For the three dimensional pressure computations, a program developed by R.B. Chapman entitled NSUP was used as a starting point. Some of the subroutines contained in NSUP and CAPSIZE were adopted directly, while others were completely rewritten. A number of new subroutines were written as well.

SSX Subroutines

The following is a list of subroutines and functions used in the SSX. Short descriptions are given here and the more complicated subroutines should be sufficiently well documented internally to clearly illustrate the computational process.

MAIN	The purpose of the main program is to accept data file names, open the appropriate files, read input data, call several subroutines to prepare data for the simulation, and then execute the time step loop which calls the several subroutines which perform the simulation of motion, hull girder loads and hull dynamic pressures. Calls READIN, SETUP, PREPARE, COFFEE, FOLIO1, OUTCOF, FOLIO2, RK4, HULL, PRESSURE and FINI.
READIN	Reads data from the basic information file [BIF], the ship offset file [OFF], the sectional hydrodynamic coefficient file [COF] and the panel pressure matrices file [MAT]. Prints an echo of data read from [BIF]. Called from MAIN.
PREPARE	Performs computations in preparation for simulation, including (a) from given offsets on the port side, assigns symmetrical offsets on starboard side, (b) recomputes offset coordinates relative to center of mass, (c) consolidates fore and aft profile coordinates into a single pair of arrays, (d) loads certain sectional mass arrays and generalized mass matrix, (e) computes wave numbers, (f) converts initial data in degrees to radians, (g) converts initial linear velocities from ship coordinates to fixed coordinates, and (h) divides all sectional hydrodynamic coefficients by fluid density, (i) initializes special free surface representation for pressure calculations. Calls AFSR. Called from MAIN.

ROTATE	Sets up coordinate system rotation matrices. Called from PREPARE, FORTIS.
FOLIO1	Displays ship motion response and hull girder loading response values for each time step. Also accumulates data for post simulation statistical analysis. Called from MAIN.
DERIV	Evaluates the right hand side of the equations of motion. These values are the ones integrated by the integration scheme. Causes external forces to be calculated, combined with inertial properties to yield translational and angular momentum in the appropriate coordinate systems for solution using Newton's second law. Called from RK4. Calls FORTIS.
FINI	Prints final execution and response value statistics - including mean, rms, maximum and minimum.
RK4	Fourth-order Runge-Kutta integration subroutine. Integrates an array YYDOT containing rotational and translational velocities and accelerations to yield array YY which contains new displacements and velocities. Called from MAIN. Calls DERIV.
ERROR	Subroutine to display a limited amount of diagnostics in the event of a runtime error, principally due to data input errors.
KRYLOV	Evaluates two-dimensional exponential and trigonometric integrals over each station of the ship. Called from AQUA2D.
COFFEE	Performs two-dimensional linear interpolation in the table of sectional hydrodynamic coefficients. Coefficients are selected for the mean design draft, and at encounter

frequencies associated with each wave component and at "characteristic" frequencies of ship motion initially assumed to be the modal frequency of the sea spectrum if irregular waves are imposed.

FXIF	Provides an index (IT) and a multiplier (TX) for linear interpolation for draft in the table of sectional hydrodynamic coefficients.
TXIT	Provides an index (TX) and a multiplier (IT) for linear interpolation for frequency in the table of sectional hydrodynamic coefficients.
FUNCTION COX	Performs the two-way linear interpolation in the table of sectional hydrodynamic coefficients using IF, FX, IT, TX obtained through FXIF and TXIT.
FUNCTION COXDXP	Same as COXDX except it uses sectional hydrodynamic coefficients that have not been divided by sectional area, and determines the x-spatial derivative.
FUNCTION ETABAR	Provides the wave elevation in fixed coordinate system above a given point of the ship's centerline along hull, stern and transom.
FUNCTION ETAF	Provides the coordinate of the water surface in yawed and pitched coordinates given a section number, distance off the centerline and relative displacement of the section.
GETFREQ	Evaluates the average of the two most recent zero crossing periods and calculates characteristic frequencies of ship motion in heave, sway and roll. Called from RK4.
OUTCOF	Prints the temporary sectional hydrodynamic coefficients currently in arrays in COMMON/COEFFX/. Normally done only at the beginning of the simulation. Called from MAIN.

FOLIO1	Displays or prints a summary of basic input data. Called from MAIN.
FOLIO2	Displays or prints ship motion and hull girder response. Called from MAIN.
FOLIO3	Displays or prints ship hull pressure reponse. Called from MAIN.
TBAR	Provides the average depth of immersion of a given section.
SPECTRA	Provides the amplitude and frequency of ten wave components derived from a two parameter Bretschneider spectrum. Called from SETUP.
PRESSURE	Provides values of dynamic pressure at the center of each hull panel for a particular instant of time. Called from MAIN. Calls ACPTR, ZBLACN, POTB and CFSR.
ZBLACN	In the pressure computations, applies the specified body accelerations for the six degrees of freedom to compute the resulting normal accelerations at the panel centers. The free-surface induced normal accelerations ACNW(J) are subtracted to obtain the net normal acceleration at each panel center, ACN(J), of the body relative to the fluid. These accelerations must be cancelled by the time derivative of the body source density distribution, ST(K). The ACN(J) vector is multiplied by he E matrix to get the necessary net rate of change of the panel source densities, ST(K). The total source strength densities are accumulated in STOLD(K). Called by PRESSURE.

ACPTR	In the pressure computations computes free-surface induced accelerations ACNW(J) and pressures PRFS(T) at panel centers. Called by PRESSURE.
POTB	In the pressure computations this subroutine is used to compute the generalized body-induced pressure BPRES generated by a known source strength distribution and its time derivative. The matrix P(J,K) is multiplied by a vector, ST(K), representing the time derivative of source densities of the panels. Similarly the term proportional to forward speed is computed from the matrix PX and a vector representing the accumulated source densities, STOLD(K). Called by PRESSURE.
CFSR	In the pressure computations advances free-surface by one time increment. Moves body relative to the free surface. Adds the changes in free-surface elevation induced by the body sources acting over the time increment to the free-surface representation. Second order effects in time are included.
AFSR	In the pressure computations sets up and initializes the free-surface representation called by PREPARE.

HYDREX2 Subroutines

The following is a list of subroutines in HYDREX2.

MAIN	Accepts data file names, opens the appropriate files, reads input data, loops over six drafts, calling FLOAT and HYDRO within the loops. Also calls INDATA to read in data from files.

FLOAT	"Floats" the ship at the specified draft. Creates a set of "wet" offsets to be used for hydrostatic calculations and for the computation of sectional hydrodynamic coefficients.
HYDRO	Applies the Frank Close-fit method to "wet" offsets. Adopted from CAPSIZE which adopted it with some modifications from the original NSRDC code. HYDRO calls GIRL, BEER, WINE and STATN.
STATN	Revises the station offsets by inserting additional points in order to optimize the hydrodynamic computations and to suppress anomolous behaviour which sometimes occurs at certain "singular frequencies". INSERT is called by STATN as part of this process.
GIRL	Computes certain frequency-independent coefficients which are used in the two-dimensional hydrodynamic computations.
BEER	Computes the two-dimensional hydrodynamic coefficients for the special cases of zero or infinite frequency. It calls the simultaneous linear equation solver LINEQT.
WINE	Performs the computation of the two-dimensional hydrodynamic coefficients for finite nonzero frequencies, calling several subroutines for special operations. WOMEN computes some of the interaction between segments of the section. SONG performs the integration of pressures around the section. ROMEO evaluates the exponential integral with complex argument. JULIET is a simultaneous linear equation solver for certain sets of equations in HYDRO.

HYDREX3 Subroutines

The following routines are contained in the program HYDREX3.

MAIN	Reads input data and calls subroutines EBD and POTST.
EBD	This subroutine reads in the (x,y,z) coordinates of each of the four corner points into a set of arrays XPT(N), YPT(N), ZPT(N). Panels are identified by a set of four integers giving the array positions of the four corner points of each panel. Panel areas, normals and center point coordinates are then computed. Finally, the E matrix giving the source time derivative distribution for a set of prescribed normal accelerations is computed. The inverse of E is computed first by the subroutine GE which gives the acceleration induced at any panel center point, J, by a uniformly distributed time derivative of source strength density of unit magnitude acting over any surface of panel, JL. Subroutine MATIN inverts E to obtain the desired form. Simultaneously, the matrix PX which gives the x component of velocity at the center of panel J induced by a source strength of unit magnitude distributed over a panel, JL, is computed.
POTST	Calculates the matrix P(N,J) giving the net force or moment for the Jth degree of freedom induced by a unit time derivative of source strength over panel N. Fundamental to this is the need to compute the potential integrated over each panel area due to a uniform source density over every other panel. For panels which are far apart relative to their dimensions this value is, for unit source density, simply proportional to the product of their areas divided by the distance between centers. The method used here is to divide each panel into a large number of small subpanels and then calculate the results numerically, adding the contributions of each subpanel

under the assumption that their separations are large relative to their dimensions.

SELF	This subroutine is called POTST to compute diagonal terms in the P matrix.
MATIN	The matrix inversion routine used by subroutine EBD to invert matrix E.
GE	A subroutine called by EBD to compute the elements of matrix E prior to inversion. It computes the velocity (acceleration) at field point (XF, YF, ZF) induced by a source density (time rate of change of source strength) of value unity distributed uniformly over panel J.
GO	A subroutine called by GE.
SOLID	Also called by subroutine GE to compute the solid angle of a panel relative to the field point.
PREP	Prepares all panels for the GE subroutine. It is called by EDB prior to using GE.

4. RUN PROCEDURES

This section presents the procedures involved with running HYDREX2, HYDREX3 and SSX. It also gives input data description and formats.

4.1 PROGRAM HYDREX2

HYDREX2 calculates two-dimensional sectional added mass and damping coefficients using the Frank Close-fit method. Two data files are required - a [BIF] file which contains basic ship and program run information and the [OFF] file which contains the ship's offsets. The program will create a third file [COF] into which it will write the results of its calculations.

When HYDREX2 is run, the only required input is the following three files names:

File Name	Format
Basic Input File	(A)*
Ship Offset File	(A)
2-D Coefficient File	(A)

Descriptions of the [BIF] and [OFF] file are given in the next two sections.

Basic Information File [BIF]

The basic information file is used for programs HYDREX3 and SSX as well as HYDREX2. The full [BIF] file will contain more information than that which is described below. However, to run HYDREX2, only the first part of the [BIF] file need be created. The remainder can be created at some other time prior to running SSX. Table 1 below summarizes the input data variables and the associated format.

---

* (A) format is alphanumeric string variable

TABLE 1

BASIC INFORMATION FILE (PARTIAL)

Data Set No.	Variables	Line Format
1	TITLE	(A)
2	TF, TA	6F10.0
3	XCG, YCG	6F10.0
4	(DRAFT (I), I-1,6)	6F10.0
5	(OM(I), I=1,12)	6F10.0
6	YMAX, ZMAX, WMAX, NWL	3F10.0, I15
7	NFWD	I5
8	(YFWD(I), XFWD(I), I=1,NFWD)	2F10.0
9	NAFT	I5
10	(YAFT(I), XAFT(I), I=1,NAFT)	2F10.0
.		
.		

A description of the variables contained in Table 1 is provided below:

TITLE - Any user-specified alphanumeric string up to 81 characters in length. This is displayed at the beginning of certain pages of output.

XCG - Distance aft of the forward perpendicular of the ship's center of gravity (ft).

YCG - Height above baseline of ship's center of gravity (ft)

TF,TA - Ships forward and after draft at FP and AP.

DRAFT(I) - An array of six drafts at which hydrodynamic coefficients are calculated. The specified drafts should cover a range beginning at approximately the height above the baseline of the top of the turn of the bilge and ending at the minimum hull depth. One of the drafts should be near the mean design draft for the displacement under consideration.

OM(I) - An array of twelve frequencies at which hydrodynamic coefficients are calculated.

YMAX, ZMAX - Specifies the maximum desired vertical (YMAX) and horizontal separation between adjacent offset points for calculation of two-dimensional hydrodynamic coefficients. If both YMAX and ZMAX are positive values, interpolated offset points (straight line) will be added before computing coefficients using the method developed by W. Frank (1967).

WMAX - A "deck" on the interior waterline has been added to the geometry of each station which is surface piercing to avoid "irregular" frequencies. The default is to use only one segment for this, but a positive WMAX will allow multiple segments each with a maximum length of WMAX.

NWL - Specifies the number of interior waterline segments to be used to avoid "irregular" frequencies. The default is to use one segment, and this is selected if NWL is zero. A negative value (not recommended) will suppress the modification to Frank's procedure which eliminates irregular frequencies.

NFWD - Number of forward profile points. This is restricted to $0 < NFWD < 25$. These points are numbered by the control variable I in a counterclockwise direction when viewed from the starboard side.

YFWD(I) - Height of point I of the forward profile measured forward of the first station.

XFWD(I) - Distance of point I of the forward profile measured forward of the first station. If NWFD=1, XFWD(1) is defined to be the forward most point of the submerged hull.

NAFT - Number of after profile points. This is restricted to $0 < NAFT < 25$. These points are numbered by the control variable I in a clockwise direction when viewed from the starboard side.

YAFT(I) - Height of point I of the after profile.

XAFT(I) - Distance of point I of the after profile measured aft of the last (MSTA) station. If NAFT=1, XAFT(1) is defined to be the after most point of the submerged hull.

The offset file [OFF] contains the ship's offsets in the SHCP* format. An actual SHCP data can be used for the [OFF] file. HYDREX2 will read only what it needs. Table 2 below summarizes the input data variables and associated formats.

TABLE 2

SHIP OFFSET FILE		
Data Set Number	Variables	Line Format
1	SHCP Card A or (blank)	(A)
2	SHCP Card B or (blank)	(A)
3	SHCP Card C or space	F10.3, (A)
4	SHCP Card D - STATN, Y1 Z1, JTEST	F6.3, 2F7.0,I6

A description of the variables contained in Table 2 is provided below.

SHCP CARD A	-	This can be left blank, or if it is an actual SHCP data file, leave it intact.
SHCP CARD B	-	Blank or actual SHCP Card B
SHCP CARD C or SPACE	-	This can be the actual SHCP Card C or simply one variable SPACE, where SPACE is the station spacing. It is actually a multiplier of the x-values of the stations given in the next card set.
STATN	-	The real distance from the station to the F.P. is the product of STATN and SPACE. The STATN values must be the same for all offsets on the same station.

---

* Ship Hull Characteristic File (U.S. Navy)

Y1	-	The half breadth of the offset point. The offsets for each station should be ordered from the bottom toward the uppermost parts of the station.
Z1	-	The height of the offset point.
JTEST	-	The "breakpoint" indicator 77777 is ignored. The last offset on each station is signified by JTEST = 88888. The last offset on the last station is signified by JTEST = 99999. JTEST values other than zero (or blank) 77777, 88888 or 99999 are illegal.

For HYDREX2, the maximum number of offset points per station is 25. However, for SSX, the maximum number of offset points is 12. It is recommended that as many offset point (up to 25) as possible are used for HYDREX2 to increase the accuracy of the close-fit calculations. However, prior to running SSX, the [OFF] file must be edited to delete extra offset points in order to reduce the total number per station down to 12. The number of stations must be limited to 25.

An example of the actual input and output of HYDREX2 will be provided in Section 5 with a sample case illustration.

4.2 PROGRAM HYDREX3

HYDREX3 calculates coordinates of hull surface panel center, areas and normal vectors of panels, body source density influence matrix [E], a body panel pressure influence matrix [PP], and a forward velocity source density influence matrix [PX].

When HYDREX3 is run, the only required input is the following two file names:

File Name	Format
Panel Description File [PAN]	(A)
Matrix and Geometry File [MAT]	(A)

Table 3 summarizes the input variables and formats for the data for the panel description file [PAN].

TABLE 3

PANEL DESCRIPTION FILE		
Data Set	Variables	Format
1	NPT, NPAN	4I5
2	XPT(N), YPT(N), ZPT(N) N=1, NPT	3F10.0
3	KK(M,1), KK(M,2), KK(M3,) KK(M,4), M=1, NPAN	4I5

The variables and their use are further described below.

NPT	- Number of nodal points used to specify coordinates of panel corners. Not to exceed 150.
NPAN	- Number of quadrilateral or triangular panels. Not to exceed 120.
XPT(N) YPT(N) ZPT(N)	- The coordinates of the nodal points used as corner points for the panels. The coordinate system used here has Z positive downwards, X positive forward and Y positive starboard. The origin is at midships at the mean design water line.
KK(M,1)	- These are four integers which identify which nodal points are used to define the corners of the panels. The convention for the sequencing of the aft panel corners is to go around the panel clockwise when viewed from outside of the ship hull along the panel normal.

4.3 PROGRAM SSX

Program SSX performs a time domain simulation of ship motions, calculating hull girder loads and distributed pressures (optional). A prerequisite of executing SSX is to have run HYDREX2 and HYDREX3 in order to perform some "preprocessing". SSX requires four data files.

(1) Basic Information Data File [BIF]
(2) Ship Offset Data File [OFF]
(3) 2-D Hydrodynamic Coefficient Data File [COF]
(4) 3-D Geometry/Matrix File [MAT]

The program initially asks for the names of these four files in the above order. It then reads the data. If the [MAT] file name is 'NONE', then no pressure computations will be performed, and data sets 21 and 22 in the [BIF] file are skipped. The first eight data groups in the [BIF] file were given in Table 1. The variable and format list for the remainder of the [BIF] file is given in Table 4.

TABLE 4

BASIC INFORMATION FILE (PARTIAL)		
Data Set	Variables	Format
9	DISPL	8F10.0
10	(RADII(I), I=1,6)	8F10.0
11	SPEED	8F10.0
12	(DAMPL(I), I=1,6)	8F10.0
13	(DAMPQ(I), I=1,6)	8F10.0
14	IXWAVE IXWAVE = 1 → Include Set 14A IXWAVE ≠ 1 → Include Set 14B	I2
14A-1	NWAVES	4F10.0
14A-2	WVAMP(N), WVFRE(N), WVDIR(N), WVPHA(N), N=1, NWAVES	4F10.0
14B-1	H13, PKFRE, HEAD	8F10.0
15	(POSIT(I), I=1,6)	8F10.0
16	(VELOC(I), I=1,6)	8F10.0
17	TSTART, TSTOP, TOUTPT, TSTEP	8F10.0
18A	(JOBFO(10),I-1,10)	10I1
18B	(JOBPO(10),I=1,10)	10I1

⋮

TABLE 4 (Cont.)

BASIC INFORMATION FILE (PARTIAL)		
Data Set	Variables	Format
18C	JOBCO	I1
19	NWTSTA	I2
20	SEGWT(I), SEGMOX(I), YBAR(I) I=1, NWTSTA	3F10.0
21	TPSTART, TPSTOP, TPRAMP	8F10.0
22	BGX, SMX, BGY, SMY, TSCALE	8F10.0

The variables and their use are further described below.

DISPL - Ship displacement (long tons). It is best to enter the displacement calculated in HYDREX2 for the design draft and printed in the hydrostatic section of the HYDREX2 printout.

RADII(1) - Radius of gyration for roll, ρ_{xx} .

RADII(2) - Radius of gyration for yaw, ρ_{yy} ,

RADII(3) - Radius of gyration for pitch, ρ_{zz} .

RADII(4) - Radius of gyration, ρ_{xy} . The products of inertia are computed as:

$$I_{xy} = I_{yx} = \rho_{xy} \cdot |\rho_{xy}| \cdot m$$

where m is the mass of the ship

RADII(5) - Radius of gyration, ρ_{xz}

RADII(6) - Radius of gyration, ρ_{yz}

SPEED - Intended speed of the ship (feet/sec)

DAMPL(1) - Linear surge damping

DAMPL(2) - Linear heave damping

DAMPL(3) - Linear sway damping

DAMPL(4) - Linear roll damping

DAMPL(5) - Linear yaw damping

DAMPL(6) - Linear pitch damping

DAMPQ(1) - Quadratic surge damping

DAMPQ(2) - Quadratic heave damping

DAMPQ(3) - Quadratic sway damping

DAMPQ(4) - Quadratic roll damping

DAMPQ(5) - Quadratic yaw damping

DAMPQ(6) - Quadratic pitch damping

IXWAVE	- Flag to indicate how the user wants to specify the wave system. IXWAVE=1 means the user will specify the amplitude, frequency, direction and phase angle of NWAVE wave components. IXWAVE ≠ 1 means the user will specify the significant height and modal frequency to be used for a two-parameter (Bretschneider) unidirectional sea spectral formulation from which 10 wave components will be calculated with random phse angles.
NWAVES	- Number of sinusoidal waves. This is restricted to $0 < \text{NWAVES} < 20$.
WVAMP(I)	- Amplitude of wave component I (ft)
WVFRE(I)	- Circular frequency of wave component I
WVPHA(I)	- Phase angle in degrees at time equal zero of wave component
H13	- Significant wave height in feet to be used in two-parameter Bretschneider unidirectional sea spectral formulation
PKFRE	- Modal frequency in cycles/sec for Bretschneider formulation
POSIT(1)	- Initial X-coordinate of mass center
POSIT(2)	- Initial Y-coordinate of mass center
POSIT(3)	- Initial Z-coordinate of mass center
POSIT(4)	- Initial roll angle in degrees
POSIT(5)	- Initial yaw angle in degrees
POSIT(6)	- Initial pitch angle in degrees
VELOC(1)	- Initial speed (ft./sec.)
VELOC(2)	- Initial heave velocity

VELOC(3) - Initial sway velocity

VELOC(4) - Initial roll rate

VELOC(5) - Initial yaw rate

VELOC(6) - Initial pitch rate

The position values are specified with respect to the wave coordinate system fixed on the earth. The velocities are with respect to the ship coordinate system. Positions are in feet: velocities in feet/sec.

TSTART - Time at which the actual simulation is to start. All forces are multiplied by a ramp funtion that increases linearly in time from a value of zero at t=0 to one at TSTART. This ramp is used to avoid transients caused by arbitrary initial conditions.

TSTOP - Time at which the simulation is to end

TOUTPT - Interval at which the ship position and velocity are to be output.

TSTEP - Integration time step

JOBCO - Option for selection of different techniques used to select the sectional hydrodynamic coefficients. The sectional coefficients may be calculated for any combination of fixed or varying draft and frequency dependence, according to the following table:

JOBCO	Frequency	Draft
1	Fix	Fix
2	Float	Fix
3	Fix	Float
4	Float	Float

The selection of fixed draft means the mean still waterline draft is always used for coefficient selection. Fixed frequency means the encounter frequency associated with the peak of the sea spectrum (for irregular waves) or the frequency of encounter for a regular wave is always used.

Floating frequency means the computed "characteristic" frequencies of motion as the simulation progresses are used. Floating draft means the actual instantaneous mean sectional draft is used at each time step.

JOBFO(10) - Option array for which terms to include in the calculation of hydrodynamic forces. In the present version only JOBFO(2) and JOBFO(3) are used. If JOBFO(2)=1 then forward speed terms are included. If JOBFO(3)=1 then flare force is included. If JOBFO(4)=1, pressures in the wave above still water are assumed to be hydrostatic.

JOBPO(10) - Printout option array. Only JOBFO(1) is used in present version. A value of 1 for JOBPO(1) will cause the initially selected values for hydrodynamic coefficients to be printed.

NWTSTA - Number of weight stations. In present version of SSX, NWTSTA must equal number of hydrodynamic stations or strips.

SEGWT(I) - Weight of weight station I (long-tons)

SEGMOX(I) - Sectional roll gyradius (ft)

YBAR(I) - Distance from section c.g. and waterline, positive up.

TPZERO - Time at which to begin 3-D hull pressure computations

TPSTART - Time at which 3-D hull pressure evaluation actually begins. Between time TPZERO and TPSTART, the exciting normal accelerations are multiplied by a ramp function that increases linearly in time from a value of zero at t = TPZERO to a value of one at t = TPSTART

TPSTOP - Time at which 3-D hull pressure evaluation ends

BGX - Maximum longitudinal length scale. Corresponds to L_x in eq. (72) of Volume I. Try 2.5 p_{pp}

SMX - Minimum longitudinal length scale corresponds to l_x in eq. (73) of Volume I. Try station spacing of ship

BGY - Maximum transverse length scale. Corresponds to L_y in eq. (72) of Volume I. Try 5B

SMY - Minimum transverse length scale. Corresponse to l_y in eq. (73) in Volume I. Try 0.1B

TSCALE - Maximum time scale. Trade off between accuracy and computational effort. Try 0.6 (TPSTOP-TPZERO)

An example of actual input and output is shown in the next section with a sample run.

5. SAMPLE CASE

This example is for the SL-7 containership. An example of the [BIF], [OFF] and [PAN] data files are given. The [BIF] file is in free format. An example of the output for HYDREX2, HYDREX3 and SSX is given.

The sample case shows the full load SL-7 containership at 25 knots in head seas in a one-foot regular wave of ω = 0.34. The basic information file [BIF] is shwon in Figure 5. The offset [OFF] file is provided as Figure 6 and the panel file [PAN] is given as Figure 7. An example of the output of HYDREX2 is shown in Figure 8. An example of output of HYDREX3 is given in Figure 9. An example of output of SSX is given in Figure 10. Pressure data output is shown in Figure 11.

```
SL7 CONTAINERSHIP
32.8,32.8
-478.05,42.3005
5.,10.,20.,32.8,40.,50.
0.,0.,60.,2
    9
0.,0.
1.5,9.6
8.25,16.88
18.0,7.04
30.,0.
42.,1.92
56.,7.92
72.,17.12
81.75,23.44
    6
0.,-71.12
6.0,-67.68
10.75,-19.44
25.0,-19.44
33.0,12.24
64.83,17.92
47760.
47.,219.,219.,0.,0.,0.
16.89
0.,0.,0.,0.,0.,0.
0.,0.,0.,0.,0.,0.
 1
 1
1.,.2,180.,0.
0.,9.4118,0.,0.,0.
16.89,0.,0.,0.,0.,0.
30.,100.,.5,.5
25
094.5500,37.31,9.5145
294.5500,37.31,9.5145
296.2025,37.31,9.5145
496.2025,37.31,9.5145
1143.8223,37.31,9.5145
1134.9353,37.31,9.5145
1681.7071,37.31,9.5145
2040.0422,37.31,9.5145
2508.4206,37.31,9.5145
2578.8111,37.31,9.5145
3053.3377,37.31,9.5145
3591.3250,37.31,9.5145
3530.8231,37.31,9.5145
3175.7759,37.31,9.5145
3725.5737,37.31,9.5145
3530.6277,37.31,9.5145
2595.3904,37.31,9.5145
3151.4265,37.31,9.5145
2534.0626,37.31,9.5145
1915.6687,37.31,9.5145
1761.7387,37.31,9.5145
715.4792,37.31,9.5145
715.4792,37.31,9.5145
747.0264,37.31,9.5145
747.0264,37.31,9.5145
0.,0.,0.,0.
0.,0.,0.,0.
```

Figure 5. Basic Information File
[BIF]

SL7 CONTAINERSHIP

```
44.0250,0.,0.,0.,0,0
00.00000.000000.0000
00.00004.480101.6404
00.00007.081508.2021
00.00003.815316.4042
00.00000.289026.2467
00.00001.011639.3701
00.00004.277852.4934
00.00008.237765.124788888
00.50000.000000.0000
00.50003.324000.8202
00.50006.358903.2808
00.50006.301113.1234
00.50003.324019.6850
00.50001.300729.5276
00.50002.861539.3701
00.50007.717452.4934
00.50013.353764.993488888
01.00000.000000.0000
01.00005.202701.6404
01.00007.659604.9213
01.00008.237708.2021
01.00007.804111.4829
01.00005.838616.4042
01.00003.179522.9659
01.00005.058239.3701
01.00018.007364.960688888
01.50000.000000.0000
01.50004.191100.8202
01.50007.399503.2808
01.50008.671206.5617
01.50008.815809.8425
01.50008.093213.1234
01.50005.347319.6850
01.50004.133326.2467
01.50007.601839.3701
01.50011.908552.4934
01.50022.545264.895088888
02.00000.000000.0000
02.00004.624700.8202
02.00008.035303.2808
02.00009.480506.5617
02.00009.682909.8425
02.00008.960313.1234
02.00006.937019.6850
02.00006.214426.2467
02.00008.006432.8084
02.00014.452145.9318
02.00023.065559.0551
02.00027.112164.796688888
03.00000.000000.0000
03.00002.890400.0656
03.00007.226001.6404
03.00010.058604.9213
03.00010.752308.2021
03.00011.330411.4829
03.00011.128116.4042
03.00011.330422.9659
03.00013.151429.5276
03.00018.094039.3701
03.00026.302752.4934
03.00034.974064.632588888
04.00000.000000.0000
04.00004.335600.1312
04.00008.757901.6404
04.00011.908504.9213
04.00013.584908.2021
04.00014.741111.4829
04.00016.330816.4042
04.00018.354122.9659
04.00021.042229.5276
04.00025.782539.3701
04.00033.528852.4934
04.00041.911064.534188888
05.00000.000
05.00004.335
05.00010.839
05.00015.319
05.00018.151
05.00020.232911.4829
05.00023.036616.4042
05.00026.013722.9659
05.00028.904129.5276
```

```
05.00040.118952.4934
05.00046.969264.370188888
06.00000.000000.0000
06.00007.226000.2625
06.00014.741101.6404
06.00020.666404.9213
06.00024.279508.2021
06.00030.002516.4042
06.00033.297522.9659
06.00036.187929.5276
06.00040.176739.3701
06.00045.379552.4934
06.00050.351064.337388888
07.00000.000000.0000
07.00011.561600.3281
07.00020.232901.6404
07.00027.025304.9213
07.00031.216408.2021
07.00034.106811.4829
07.00037.488616.4042
07.00040.407922.9659
07.00042.720329.5276
07.00045.697439.3701
07.00049.223752.4934
07.00052.027464.304588888
08.00000.000000.0000
08.00019.365800.3937
08.00027.314401.6404
08.00033.962304.9213
08.00038.297908.2021
08.00041.043811.4829
08.00043.934216.4042
08.00046.246622.9659
08.00047.836329.5276
08.00049.715139.3701
08.00051.564952.4934
08.00052.547764.304588888
09.00000.000000.0000
09.00026.302700.5906
09.00033.962301.6404
09.00040.610304.9213
09.00044.310008.2021
09.00046.477811.4829
09.00048.703416.4042
09.00050.351022.9659
09.00051.304829.5276
09.00052.027439.3701
09.00052.605552.4934
09.00052.750064.304588888
10.00000.000000.0000
10.00032.661600.6562
10.00039.309601.6404
10.00045.379504.9213
10.00048.558908.2021
10.00050.293211.4829
10.00051.911816.4042
10.00052.605522.9659
10.00052.721129.5276
10.00052.750064.304588888
11.00032.751400.6562
11.00035.649700.8202
11.00044.634603.2808
11.00048.692306.5617
```

```
98968.897688888
00026.2467
57826.2467
90732.8084
346245.9318
20.00032.084854.6260
20.00032.084868.897699999
```

Figure 6. Offset File [OFF] Partial Listing

```
120, 112
   450.00      0.00     30.00
   450.00      7.04     24.00
   450.00      6.00     18.00
   450.00      3.20     12.00
   450.00      0.88      6.00
   450.00      0.00      0.00
   360.00      0.00     30.00
   360.00      1.76     30.00
   360.00      7.60     27.00
   360.00      9.20     24.00
   360.00      9.28     18.00
   360.00      7.52     12.00
   360.00      6.40      6.00
   360.00      7.20      7.20
   450.00      3.50     27.00
   290.00      0.00     30.00
   290.00      3.44     30.00
   290.00      9.69     27.00
   290.00     11.76     24.00
   290.00     14.48     12.00
   290.00     18.32      0.00
   225.00      0.00     30.00
   225.00      5.04     30.00
   225.00     13.52     27.00
   225.00     16.88     24.00
   225.00     24.48     12.00
   225.00     30.01      0.00
   135.00      0.00     30.00
   135.00     13.12     29.75
   135.00     29.92     24.00
   135.00     38.50     12.00
   135.00     43.14      0.00
     0.00      0.00     30.00
     0.00     26.40     29.50
     0.00     43.36     27.00
     0.00     47.20     24.00
     0.00     52.24     12.00
     0.00     52.38      0.00
   -90.00      0.00     30.00
   -90.00     26.40     29.50
   -90.00     41.44     27.00
   -90.00     51.56     24.00
   -90.00     52.80     12.00
   -90.00     52.80      0.00
  -180.00      0.00     30.00
  -180.00     28.00     27.00
  -180.00     43.68     24.00
  -180.00     48.32     12.00
  -180.00     52.16      0.00
  -270.00      0.00     30.00
  -270.00     35.60     12.00
  -270.00     43.28      0.00
  -360.00      0.00     30.00
  -360.00     14.00     12.00
  -360.00     25.54      0.00
  -427.50      0.00     29.50
  -427.50      1.50     29.50
  -427.50      2.72      3.50
  -427.50      6.91      0.00
  -450.00      0.00      0.00
   450.00      0.00     30.00
   450.00     -7.04     24.00
   450.00     -6.00     18.00
   450.00     -3.20     12.00
   450.00     -0.88      6.00
   450.00      0.00      0.00
   360.00      0.00     30.00
   360.00     -1.76     30.00
   360.00     -7.60     27.00
   360.00     -9.20     24.00
   360.00     -9.28     18.00
   360.00     -7.52     12.00
   360.00     -6.40      6.00
   360.00     -7.20      7.20
   450.00     -3.50     27.00
   290.00      0.00     30.00
   290.00     -3.44     30.00
   290.00
   290.00
   290.00
   290.00
   225.00
   225.00     -5.04     30.00
```

```
   225.00    -24.48     12.00
   225.00    -30.01      0.00
   135.00      0.00     30.00
   135.00    -13.12     29.75
   135.00    -29.92     24.00
   135.00    -38.50     12.00
   135.00    -43.14      0.00
     0.00      0.00     30.00
     0.00    -26.40     29.50
     0.00    -43.36     27.00
     0.00    -47.20     24.00
     0.00    -52.24     12.00
     0.00    -52.38      0.00
   -90.00      0.00     30.00
   -90.00    -26.40     29.50
   -90.00    -41.44     27.00
   -90.00    -51.56     24.00
   -90.00    -52.80     12.00
   -90.00    -52.80      0.00
  -180.00      0.00     30.00
  -180.00    -28.00     27.00
  -180.00    -43.68     24.00
  -180.00    -48.32     12.00
  -180.00    -52.16      0.00
  -270.00      0.00     30.00
  -270.00    -35.60     12.00
  -270.00    -43.28      0.00
  -360.00      0.00     30.00
  -360.00    -14.00     12.00
  -360.00    -25.54      0.00
  -427.50      0.00     29.50
  -427.50     -1.50     29.50
  -427.50     -2.72      3.50
  -427.50     -6.91      0.00
  -450.00      0.00      0.00
```

```
    1     7     8     0
    1     8     9    15
   15     9    10     2
    2    10    11     3
    3    11    12     4
    4    12    13     5
    5    13    14     6
    7    16    17     8
    8    17    18     9
    9    18    19    10
   10    19    11     0
   11    19    20    12
   12    20    21    14
   16    22    23    17
   17    23    24    18
   18    24    25    19
   19    25    26    20
   20    26    27    21
   22    28    29    23
   23    29    24     0
   24    29    30     0
   24    30    25     0
   25    30    31    26
   26    31    32    27
   28    33    34    29
   29    34    35     0
   29    35    30     0
   30    35    36     0
   30    36    37    31
   31    37    38    32
   33    39    40    34
   34    40    41    35
   35    41    36     0
```

```
  116   115   111     0
  114   117   113     0
  118   117   114     0
  115   118   114     0
  116   118   115     0
        118   116     0
        120   118     0
```

Figure 7. Panel Description File [PAN] Partial Listing

```
== ================================================================================
SL7 CONTAINERSHIP              ADDED MASS/DAMPING COEFFICIENTS    PROGRAM HYDREX
==================================================================================
                                  Station 12
                                  ----------

    Dist.  from F.P.   -396.23                     Area                 3031.365
    DRAFT (fwd)          32.80                     Roll Ctr abv WL         0.088
    DRAFT (aft)          32.80


          -----HEAVE----       -----SWAY-----      ------ROLL------     --SWAY-ROLL---
             A22      B22         A33      B33          A44      B44       A34       B34
 F eq.

 0.00    99.0000   0.0000     2.2896   0.0000       949.04     0.00      34.4       0.0
 0.10    11.6174   0.8778     1.9201   0.0002       547.86     0.03      20.1       0.0
 0.20     7.2671   1.4760     2.0412   0.0072       561.35     1.07      21.3       0.1
 0.30     5.1586   1.8017     2.2434   0.0544       581.59     7.57      23.3       0.6
 0.40     3.9867   1.9286     2.4304   0.2150       593.28    27.55      24.8       2.4
 0.50     3.3206   1.9089     2.3837   0.5407       571.59    62.45      23.5       5.8
 0.60     2.9645   1.7834     1.9918   0.9268       515.09    94.63      18.7       9.4
 0.70     2.8127   1.5879     1.4567   1.1956       456.55   106.05      13.0      11.3
 0.90     2.8677   1.1182     0.6952   1.3047       401.19    83.61       6.3      10.4
 1.30     3.3945   0.4073     0.2841   0.9812       408.22    29.37       4.9       5.4
 1.60     3.6990   0.1707     0.2848   0.7194       423.20    11.92       6.0       2.9
 2.20     7.4603   0.0000     0.7945   0.0000       748.80     0.00      15.4       0.0




-------------------------------STATION OFFSETS-------------------------------

     Height above      Half                Submerged Offsets
         Baseline      BREADth                -Y-         -Z-

            0.000        0.000             -32.800        0.000
            0.591       26.303             -32.209       26.303
            0.820       30.060             -31.980       30.060
            1.640       33.962             -31.160       33.962
            3.281       38.298             -29.519       38.298
            4.921       40.610             -27.879       40.610
            6.562       42.778             -26.238       42.778
            8.202       44.310             -24.598       44.310
            9.842       45.524             -22.958       45.524
           11.483       46.478             -21.317       46.478
           13.123       47.345             -19.677       47.345
           16.404       48.703             -16.396       48.703
           19.685       49.715             -13.115       49.715
           26.247       50.929              -6.553       50.929
           29.528       51.305              -3.272       51.305
                                         WATERLINE       51.593
           32.808       51.594
           39.370       52.027
           52.493       52.605
           64.304       52.750
```

Figure 8. Output of HYDREX2
Partial Listing

```
        NX         NY         NZ        XP          YP         ZP        AREA
  1    0.0000     0.0000     1.0000   390.0000     0.5867    30.0000     79.2000
       0.0176     0.5404     0.8412   405.0000     3.2150    28.5000    499.6292
       0.0264     0.7592     0.6504   405.0000     6.8350    25.5000    355.6513
  4    0.0301     0.9964    -0.0797   405.0000     7.8800    21.0000    541.9710
  5    0.0394     0.9341    -0.3549   405.0000     6.5000    15.0000    578.1236
       0.0525     0.9600    -0.2752   405.0000     4.5000     9.0000    562.5251
       0.0698     0.9974    -0.0166   405.0000     3.6200     4.8000    216.5585
  8    0.0000     0.0000     1.0000   325.0000     1.3000    30.0000    182.0000
       0.0120     0.4445     0.8957   325.0000     5.6225    28.5000    472.4277
 1     0.0283     0.8527     0.5216   325.0000     9.5625    25.5000    246.2683
 11    0.0365     0.9992     0.0133   336.6667    10.0800    22.0000    210.1590
 1     0.0694     0.9962     0.0531   325.0000    10.7600    16.5000    632.4223
       0.1151     0.9722     0.2037   325.0000    11.8800     7.8000    604.7863
       0.0000     0.0000     1.0000   257.5000     2.1200    30.0000    275.6000
       0.0158     0.3772     0.9260   257.5000     7.9225    28.5000    516.9807
          510     0.7405     0.6701   257.5000    12.9625    25.5000    263.3413
7                 0.9135     0.3928   257.5000    16.9000    18.0000    853.8872
79                0.9205     0.3594   257.5000    21.8225     6.0000    847.3912
8      0.             0138     0.9999   180.0000     4.5400    29.9375    817.2777
8      0.00             34     0.9424   195.0000    10.5600    28.9167    404.9271
82     0.0125                0.9458   165.0000    18.8533    26.9167    799.3557
8      0.0532    -           0.7425   195.0000    20.1067    25.0000    203.6394
8      0.0786    -0.6           547   180.0000    27.4450    18.0000   1312.5892
85     0.1172    -0.9156              180.0000    34.0325     6.0000   1184.2212
8      0.1012    -0.9948               67.5000     9.8800    29.8125   2668.0806
8      0.0000    -0.0189     0.            0000    27.6267    28.7500   1157.2611
88     0.0000    -0.1640     0.98            0    28.8000    26.9167   1200.2803
89    -0.0131    -0.6156     0.7880               40.1600    25.0000    329.9435
9      0.0138    -0.2842     0.9587   -6           1.9650    18.0000   1871.9629
9      0.0481    -0.9936     0.1027   -60.            450     6.0000   1657.5626
92     0.0062    -1.0000     0.0000   -60.000                29.7500   2376.4260
9      0.0000    -0.0189     0.9998  -120.0000               28.2500   1457.4741
9     -0.0257    -0.1065     0.9940  -150.0000   -1             0000    219.3015
95    -0.0281    -0.1878     0.9818  -150.0000   -37.7            0    475.0337
9     -0.0249    -0.2841     0.9585  -120.0000   -45.5600              1116.7516
9     -0.0814    -0.9296     0.3594  -150.0000   -47.8533                 0.0344
98    -0.0474    -0.9514     0.3044  -150.0000   -51.0933     8.            130
9      0.0000    -0.1361     0.9907  -210.0000   -14.5600    28.000
10    -0.1575    -0.4456     0.8813  -240.0000   -26.4267    22.0000    1
101   -0.1307    -0.9247     0.3576  -210.0000   -42.5333    16.0000    58
102   -0.1182    -0.8364     0.5353  -240.0000   -42.4000     8.0000    645.6
10     0.0000    -0.7894     0.6139  -330.0000    -4.6667    24.0000   1026.1578
10    -0.1077    -0.4486     0.8872  -300.0000   -16.5333    18.0000   1805.6287
105   -0.1705    -0.7102     0.6830  -330.0000   -25.0467     8.0000    760.3083
10     0.1558     0.8949    -0.4183  -352.5000   -20.3800    13.8333   1936.0449
10     0.0130    -0.7893     0.6139  -382.5000    -5.1667    23.8333    769.6833
108   -0.1703    -0.9843     0.0462  -405.0000    -6.0733    15.0000    891.4889
10    -0.2034    -0.7057     0.6787  -382.5000   -14.0867     5.1667    573.8820
11     0.3141     0.9442    -0.0988  -405.0000    -9.4200    11.0000    929.3218
111   -1.0000     0.0000     0.0000  -427.5000    -3.2100    11.0000     49.7100
11    -0.1932    -0.6290     0.7530  -435.0000    -3.2100     1.1667     62.5984
```

Figure 9. Output of HYDREX3.
Partial Listing

```
==============================================================================
SL7 CONTAINERSHIP                                     PROGRAM SSX (Version 1.0)
==============================================================================

--------------------------Inertial Characteristics-----------------------------
                                                  ---------GYRADII--------
   WEIGHT       47760.00  L.TONS                  Kxx         47.000     ft
                                                  Kyy        219.000     ft
   XCG            -478.050     ft                 Kzz        219.000     ft
   YCG              42.300     ft                 Kxy          0.000     ft
   ZCG               0.000     ft                 Kxz          0.000     ft
                                                  Kyz          0.000     ft
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

   --------DAMPING COEFFICIENTS--------            ----OPS DATA----
          (user specified)
   MOTION      LINEAR         QUADRATIC
   ------    ----------      ----------
   SURGE    0.0000E+00      0.0000E+00            SPEED       42.23
   HEAVE    0.0000E+00      0.0000E+00
   SWAY     0.0000E+00      0.0000E+00
   ROLL     0.0000E+00      0.0000E+00
   YAW      0.0000E+00      0.0000E+00
   PITCH    0.0000E+00      0.0000E+00

--------------------------------Initial Conditions-----------------------------

   -----------POSITION-----------                 ----------VELOCITY----------
   X(CG)                      0.000   ft          SPEED        42.230   ft/sec
   Y(CG)                      9.412   ft          HEAVE         0.000   ft/sec
   Z(CG)                      0.000   ft          SWAY          0.000   ft/sec
   ROLL                       0.000 deg           ROLL          0.000   deg/sec
   YAW                        0.000 deg           YAW           0.000   deg/sec
   PITCH                      0.000 deg           PITCH         0.000   deg/sec

----------------------------------Run Parameters-------------------------------

   SIMULATION PARAMETERS                          JOB OPTIONS
   ---------------------                          -----------
   TIME(START)       30.00
   TIME(END)        100.00
   OUTPUT DT          0.50
   INTEG. DT          0.50
-------------------------------------------------------------------------------


--------------------------------MISCELLANEOUS VALUES---------------------------

   RHO        0.000889
   G         32.169998
   JOBCO 1
   JOBFO    1   1   0
```

Figure 10. Part of Output of SSX
Partial Listing

```
==========================================================================
SL7 CONTAINERSHIP                                 PROGRAM SSX (Version 1.0)
==========================================================================

-------------------------------WAVE COMPONENTS-----------------------------
   WAVE      AMPLITUDE     FREQUENCY     DIRECTION     PHASE
   ----      ---------     ---------     ---------     -----
      1           1.00          0.34        180.00      0.00
-------------------------SECTIONAL WEIGHT PROPERTIES-----------------------
   SECTION      WEIGHT        Kxx         Centroid
   -------      ------      -------       --------


        1       94.550       37.31            9.51
        2      294.550       37.31            9.51
        3      296.203       37.31            9.51
        4      496.203       37.31            9.51
        5     1143.822       37.31            9.51
        6     1134.935       37.31            9.51
        7     1681.707       37.31            9.51
        8     2040.042       37.31            9.51
        9     2508.421       37.31            9.51
       10     2578.811       37.31            9.51
       11     3053.338       37.31            9.51
       12     3591.325       37.31            9.51
       13     3530.823       37.31            9.51
       14     3175.776       37.31            9.51
       15     3725.574       37.31            9.51
       16     3530.628       37.31            9.51
       17     2595.390       37.31            9.51
       18     3151.427       37.31            9.51
       19     2534.063       37.31            9.51
       20     1915.669       37.31            9.51
       21     1761.739       37.31            9.51
       22      715.479       37.31            9.51
       23      715.479       37.31            9.51
       24      747.026       37.31            9.51
       25      747.026       37.31            9.51
```

Figure 10. Part of Output of SSX
Partial Listing

```
                                    POSITION OF CENTER OF GRAVITY
    TIME           WAVE             X              Y              Z

     0.0000        1.0000          0.0000         0.0000         0.0000
     0.5000        0.9702         21.1150         0.0002         0.0000
     1.0000        0.8825         42.2301         0.0016         0.0000
     1.5000        0.7422         63.3456         0.0048         0.0000
     2.0000        0.5576         84.4618         0.0101         0.0000
     2.5000        0.3397        105.5792         0.0169         0.0000
     3.0000        0.1016        126.6983         0.0238         0.0000
     3.5000       -0.1426        147.8197         0.0285         0.0000
     4.0000       -0.3783        168.9435         0.0284         0.0000
     4.5000       -0.5914        190.0700         0.0202         0.0000
     5.0000       -0.7693        211.1992         0.0009         0.0000
     5.5000       -0.9012        232.3307        -0.0322         0.0000
     6.0000       -0.9794        253.4639        -0.0808         0.0000
     6.5000       -0.9991        274.5980        -0.1451         0.0000
     7.0000       -0.9592        295.7319        -0.2235         0.0000
     7.5000       -0.8621        316.8644        -0.3120         0.0000
     8.0000       -0.7135        337.9943        -0.4046         0.0000
     8.5000       -0.5224        359.1200        -0.4931         0.0000
     9.0000       -0.3001        380.2405        -0.5676         0.0000

    88.0000        0.6254       3715.1992         0.9302         0.0000
    88.5000        0.7957       3736.2339         1.0866         0.0000
    89.0000        0.9187       3757.2551         1.1782         0.0000
    89.5000        0.9870       3778.2671         1.1995         0.0000
    90.0000        0.9966       3799.2751         1.1492         0.0000
    90.5000        0.9469       3820.2847         1.0303         0.0000
    91.0000        0.8409       3841.3008         0.8500         0.0000
    91.5000        0.6849       3862.3284         0.6189         0.0000
    92.0000        0.4881       3883.3718         0.3509         0.0000
    92.5000        0.2621       3904.4341         0.0621         0.0000
    93.0000        0.0205       3925.5173        -0.2304         0.0000
    93.5000       -0.2224       3946.6223        -0.5090         0.0000
    94.0000       -0.4521       3967.7488        -0.7573         0.0000
    94.5000       -0.6549       3988.8945        -0.9603         0.0000
    95.0000       -0.8187       4010.0569        -1.1060         0.0000
    95.5000       -0.9335       4031.2317        -1.1858         0.0000
    96.0000       -0.9926       4052.4143        -1.1949         0.0000
    96.5000       -0.9924       4073.5996        -1.1329         0.0000
    97.0000       -0.9328       4094.7822        -1.0034         0.0000
    97.5000       -0.8176       4115.9570        -0.8141         0.0000
    98.0000       -0.6535       4137.1196        -0.5763         0.0000
    98.5000       -0.4504       4158.2656        -0.3041         0.0000
    99.0000       -0.2205       4179.3926        -0.0138         0.0000
    99.5000        0.0225       4200.4980         0.2774         0.0000
   100.0000        0.2640       4221.5815         0.5521         0.0000


 RMS               0.7112        304.5160         0.8489         0.0000
     0.0000
 MEAN             -0.0134       3704.3413        -0.0220         0.0000
     0.0000
 MAX               1.0000       4221.5815         1.2009         0.0000
     0.0000
 MIN              -0.9991          0.0000        -1.1991         0.0000
     0.0000
```

Figure 10. Left side of 132 character width printout of SSX (partial)

```
ROTATIONS  IN DEGREES
ROLL          YAW          PITCH          MIDS. VERT        SPEED

0.0000       0.0000        0.0000       0.00000E+00       42.2300
0.0000       0.0000        0.0000      -0.57850E+06       42.2301
0.0000       0.0000       -0.0002      -0.60541E+06       42.2305
0.0000       0.0000       -0.0008      -0.63125E+06       42.2315
0.0000       0.0000       -0.0021      -0.65431E+06       42.2335
0.0000       0.0000       -0.0047      -0.67303E+06       42.2364
0.0000       0.0000       -0.0088      -0.68603E+06       42.2403
0.0000       0.0000       -0.0148      -0.69229E+06       42.2451
0.0000       0.0000       -0.0227      -0.69116E+06       42.2503
0.0000       0.0000       -0.0323      -0.68247E+06       42.2557
0.0000       0.0000       -0.0433      -0.66658E+06       42.2608
0.0000       0.0000       -0.0548      -0.64439E+06       42.2649
0.0000       0.0000       -0.0656      -0.61736E+06       42.2676
0.0000       0.0000       -0.0745      -0.58740E+06       42.2684
0.0000       0.0000       -0.0800      -0.55682E+06       42.2669
0.0000       0.0000       -0.0806      -0.52814E+06       42.2628
0.0000       0.0000       -0.0751      -0.50389E+06       42.2560
0.0000       0.0000       -0.0626      -0.48633E+06       42.2466
0.0000       0.0000       -0.0427      -0.47728E+06       42.2349

0.0000       0.0000        0.2165      -0.62122E+06       42.0858
0.0000       0.0000        0.1917      -0.62237E+06       42.0545
0.0000       0.0000        0.1554      -0.62240E+06       42.0315
0.0000       0.0000        0.1099      -0.62129E+06       42.0183
0.0000       0.0000        0.0577      -0.61914E+06       42.0157
0.0000       0.0000        0.0021      -0.61605E+06       42.0238
0.0000       0.0000       -0.0537      -0.61218E+06       42.0422
0.0000       0.0000       -0.1062      -0.60782E+06       42.0697
0.0000       0.0000       -0.1523      -0.60310E+06       42.1047
0.0000       0.0000       -0.1893      -0.59842E+06       42.1451
0.0000       0.0000       -0.2149      -0.59400E+06       42.1883
0.0000       0.0000       -0.2275      -0.59013E+06       42.2318
0.0000       0.0000       -0.2265      -0.58704E+06       42.2730
0.0000       0.0000       -0.2118      -0.58496E+06       42.3092
0.0000       0.0000       -0.1846      -0.58401E+06       42.3385
0.0000       0.0000       -0.1463      -0.58419E+06       42.3591
0.0000       0.0000       -0.0993      -0.58556E+06       42.3697
0.0000       0.0000       -0.0465      -0.58798E+06       42.3697
0.0000       0.0000        0.0091      -0.59137E+06       42.3593
0.0000       0.0000        0.0641      -0.59538E+06       42.3389
0.0000       0.0000        0.1154      -0.59986E+06       42.3098
0.0000       0.0000        0.1598      -0.60459E+06       42.2737
0.0000       0.0000        0.1948      -0.60915E+06       42.2327
0.0000       0.0000        0.2183      -0.61340E+06       42.1891
0.0000       0.0000        0.2289      -0.61708E+06       42.1456


0.0000       0.0000        0.1599       0.13537E+05        0.1269

0.0000       0.0000       -0.0053      -0.60289E+06       42.1984

0.0000       0.0000        0.2289       0.00000E+00       42.3768

0.0000       0.0000       -0.2282      -0.62252E+06        0.0000
```

Figure 10. Right side of 132 character width printout of SSX (partial)

```
PR SURES
     0.38415968      0.24316381      0.23651379      0.22607508      0.2106125
     0.19246852      0.17776074      1.14601755      1.16095650      1
     1.05791485      1.29189086      1.40567327      2.06866050
     2.15990114      2.33326197      2.64962888      3.0
     3.24160838      2.96588731      3.32186317
     4.04283953      3.78925920      3.97
     4.74622965      4.53059387
     5.02411795      5.2349
     5.02339220
     5.83894


-- -----------TOTAL PRESSURES------------------
    -1.46834481     -0.83654284     -0.78719705     -1.71809840      0.00665942
     0.23063746      1.18806934     -0.45587230     -0.17283618      0.35
    17.64413261      2.48809695      1.18732226      0.37171781
     1.17200983      1.83491194      2.47403312      0.9994
     1.39725995      1.88052142      2.80034971
     1.43834829      1.74238777      2.935439
     2.78261328      2.71100593      3
    56.71600723      3.11889052
     8.52794075      7.50


 B  DY INDUCED PRESSURES
    -1.85250449     -1.07970667     -1.02371085     -1.94417346     -0.20395313
     0.03816893      1.01030862     -1.60188985     -1.33379269     -0.83561438
    16.58621788      1.19620597     -0.21835098     -1.69694269     -1.50451446
    -0.98789138     -0.49835002     -0.17559581     -2.02371120     -1.95836079
    -1.84434843     -1.08536589     -0.52151340     -0.05880385     -2.24148512
    -2.60449123     -2.04687142     -1.03688288      0.02363856      1.00506485
    -1.96361649     -1.81958807     -0.79081982     -1.89128637      2.57368040
    51.69189072     -2.11605954     -1.65408003     -1.66934085     -1.67518842
     3.50454855      1.93148851     -1.43030024     -0.60449553      1.84978557
     0.40260637     -0.62282950     -0.58973241     -0.03671409      2.83566427
    -0.41253781      0.83666974     -0.10891942      0.55786502      0.00000000
     0.03906842     -2.58687186     -1.62076259     -0.77843022      7.79187298
    -0.35989502     -0.11869927     -0.31951666     -1.83989894     -1.40328050
    -0.73166758     11.49159718     -0.16707416     -0.29368657     -1.92022955
    -1.52942920     -1.04738855     -0.60960990     -0.22613873     -2.24740911
    -1.75089848     -1.78046894     -0.91861057     -0.41995618      0.12429215
    -2.62914538     -2.34276557     -1.72142696     -1.22348630      1.35021770
    30.93821144     -2.01162410     -1.54072320     -0.79989606     -1.55071521
    10.87392330      0.00000000     -2.08891344     -1.69489515     -1.51071918
    -1.30158806      2.16383004      1.18314898     -1.51980424     -0.68915701
     1.78912711      0.56204945     -0.96726221     -0.71865982     -0.38488629
     0.34184885     -0.67282838     -0.67956263     -0.36257684      0.79844165
     0.00000000     -0.03861484
```

Figure 11. Pressure output of SSX
(partial listing for one time step)

APPENDIX A

Program Listings

HYDREX2

HYDREX3

SSX

(Appendix A Contained Under Separate Cover)

APPENDIX A

Program Listings

HYDREX2

HYDREX3

SSX

```
      PROGRAM SSX
C
      CHARACTER*25 OFFIL,BIFIL,COFIL,MATFIL
      CHARACTER*30 TITLE
      COMMON/HEAD/TITLE
      COMMON/IOFILE/ OFFIL,BIFIL,COFIL,MATFIL
      COMMON/IO/INPUT,OUTPUT,BIF,OFF,COF,MAT
      INTEGER OUTPUT,BIF,OFF,COF
      COMMON /COEFF4 /   COEFF4(6,12,8,25),AREAN(25,6)
      COMMON/OPTION/JOBCO,JOBFO(10),JOBPO(10)
      COMMON/JOBP/JOBP
      COMMON /JOBB/ IJOB,IFORCE,IAXIS,IWANT
      COMMON /      /   NWAVES, WVSUM,
     1                  WVAMP(20), WVFRE(20), WVDIR(20), WVPHA(20),
     2                  WN(20), WNX(20), WNZ(20),
     3                  CSK(20), CCK(20), CXK(20),
     4                  CYK(20), CZK(20), XW(20)
      COMMON /      /   MSTA, NPROF, NFWD, NAFT, LPTS(25),
     1                  XOFF(25), YOFF(25,25), ZOFF(25,25),
     2                  XPROF(51), YPROF(51)
      COMMON /DAMP  /   DAMPL(6), DAMPQ(6)
      COMMON /LHS   /   Y(13)
      COMMON /MASS  /   RHO, G, GAMMA
      COMMON /MASS  /   DISPL, SMASS, XCG, YCG, ZCG,
     1                  AMX, AMY, AMZ, RADII(6)
      COMMON /MASS  /   PMI(3,3)
      COMMON /RESIST/   SPEED
      COMMON /TIME  /   TIME, RAMP
      COMMON /TIME  /   TO, TSTART, TSTOP, TOUTPT, TSTEP, ERR
      COMMON/PREXIN/TPSTART,TPSTOP,TPRAMP,BGX,SMX,BGY,SMY,TSCALE
      COMMON/DRFT1/   DRAFT1(6)
      COMMON /SIGMA /   NK, SIGMA(24), SIGMAO, ERRO, OM(12)
      COMMON /STATS /   NRHS(4), DELTA, HSUM, HMIN, HMAX, TTO, TT1
      DATA TO /0.0/
      DATA  NEQS  /12/
      LOGICAL WL,ADJUST,INTERACT
      DATA  RHO  /0.00088861607142/
      DATA ERRO/1.0E-37/
      DATA INPUT/5/,OUTPUT/6/,BIF/1/,OFF/2/,COF/9/
      DATA MAT/3/
C-------------------------------------------------------------
C
C *** 1.0  Get DATA file names and assign/open files
C
      TYPE 902
      ACCEPT 901,BIFIL
      TYPE 903
      ACCEPT 901,OFFIL
      TYPE 904
      ACCEPT 901,COFIL
      TYPE 905
      ACCEPT 901,MATFIL
C
      OPEN(UNIT=BIF,STATUS='OLD',FILE=BIFIL)
      OPEN(UNIT=OFF,STATUS='OLD',FILE=OFFIL)
      OPEN(UNIT=COF,STATUS='OLD',FORM='UNFORMATTED',FILE=COFIL)
      IF(MATFIL.NE.'NONE') OPEN(UNIT=MAT,STATUS='OLD',FORM='UNFOR
     1 MATTED',FILE=MATFIL)
C
C *** 2.0 Read in data from [BIF],[OFF] and [COF] files
      CALL READIN
C
C *** 3.0 Perform preparatory computations
      CALL PREPARE
C
      CALL COFFEE
      CALL FOLIO1
      IF(JOBPO(1).EQ.1) CALL OUTCOF
```

```
C *** 5.0 Simulation
C
      WRITE (OUTPUT,130)
      TSTEP=-ABS(TSTEP)
      TIME=TO
      TTO=TO
      TT1=TO
  110 CALL FOLIO2                              !Show time zero condition
      TNEXT=TIME+TOUTPT
C *** Fourth-order Runga-Kutta integration
      CALL RKF (TIME,Y,TNEXT,ERR,NEQS,TSTEP)
C *** Get hull girder loads
      CALL HULL
C *** Get panel pressures
      TP=TIME-TPSTART
      IF(TP.GE.0..AND.TP.LE.TPSTART) CALL PRESSURE(TP)
C     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      TIME=TNEXT
      IF (TIME.LT.TSTOP) GO TO 110
  120 CALL FOLIO2
      CALL FINI
      STOP
C
  130 FORMAT (1H1,27X,29HPOSITION OF CENTER OF GRAVITY,
     $   9X,20HROTATIONS IN DEGREES,27X,19HHORIZONTAL VELOCITY/
     $   4X,4HTIME,8X,4HWAVE,8X,1HX,11X,1HY,11X,1HZ,13X,
     $   4HROLL,9X,3HYAW,7X,5HPITCH,6X,6HRUDDER,
     $   7X,5HSPEED,5X,9HSWAY RATE/)
C
C
 901  FORMAT(A)
 902  FORMAT(' Name of Basic INPUT File [BIF] ? > '$)
 903  FORMAT(' Name of Offset File       [OFF] ? > '$)
 904  FORMAT(' Name of Coefficent File  [COF] ? > '$)
 905  FORMAT(' Name of 3-D Matrix File  [MAT] ? > '$)
C
      END
```

```
      SUBROUTINE READIN
C
      COMMON/IO/INPUT,OUTPUT,BIF,OFF,COF,MAT
      INTEGER OUTPUT,BIF,OFF,COF
      INTEGER ECHO
      COMMON /SIGMA /    NK, SIGMA(24), SIGMAO, ERRO, OM(12)
      COMMON/DRFT1/DRAFT1(6)
      COMMON/IOFILE/ OFFIL,BIFIL,COFIL,MATFIL
      COMMON/HEAD/TITLE
      CHARACTER*30 TITLE
      COMMON/COEFF4/COEFF4(6,12,8,25),AREAN(25,6)
      COMMON/OPTION/JOBCO,JOBFO(10),JOBPO(10)
      COMMON /       /   NWAVES, WVSUM,
     1                   WVAMP(20), WVFRE(20), WVDIR(20), WVPHA(20),
     2                   WN(20), WNX(20), WNZ(20),
     3                   CSK(20), CCK(20), CXK(20),
     4                   CYK(20), CZK(20), XW(20)
      COMMON /       /   MSTA, NPROF, NFWD, NAFT, LPTS(25),
     1                   XOFF(25), YOFF(25,25), ZOFF(25,25),
     2                   XPROF(51), YPROF(51)
      COMMON /DAMP   /   DAMPL(6), DAMPQ(6)
      COMMON /LHS    /   Y(13)
      COMMON /MASS   /   RHO, G, GAMMA
      COMMON /MASS   /   DISPL, SMASS, XCG, YCG, ZCG,
     1                   AMX, AMY, AMZ, RADII(6)
      COMMON /MASS   /   PMI(3,3)
      COMMON /RESIST/    SPEED
      COMMON /TIME   /   TIME, RAMP
      COMMON /TIME   /   TO, TSTART, TSTOP, TOUTPT, TSTEP, ERR
      COMMON/SXPROP/ SEGMAS(26),SEGMOX(26),STRMAS(26),STRMOM(26),
     *   STRMOX(26),XBAR(26),YBAR(26),SEGWT(26),NWTSTA
      REAL XAFT(25), YAFT(25), XFWD(25), YFWD(25)
      EQUIVALENCE (XFWD(1), XPROF(1)), (YFWD(1),YPROF(1)),
     $            (XAFT(1), XPROF(26)),(YAFT(1),YPROF(26))
      COMMON/PREXIN/TPSTART,TPSTOP,TPRAMP,BGX,SMX,BGY,SMY,TSCALE
        COMMON/BD/XPAN(120),YPAN(120),ZPAN(120),AREAP(120),
     *  ST(120),ACN(120),ACNW(120),AN(120,3),E(120),P(120,6),
     *  PRFS(120),STOLD(120),PX(120,6)
        COMMON/A/NPAN,NPT,GEE,RHOP,NKX,NKY,EYE,DT,TIM,UFWD
      COMMON/SOURCE/ EE(120,120)
      REAL POSIT(6),VELOC(6)
      COMMON/DISPLAY/TARE(6)
      EQUIVALENCE (POSIT(1),Y(1)), (VELOC(1),Y(7))
      DATA MAXSTA/25/
      DATA MAXPTS/25/
      DATA MAXFWD/25/
      DATA MAXAFT/25/
      DATA MAXWVS/20/
      DATA TPSTOP/0./
      DATA  ERRMIN/1.0E-10/
      CHARACTER*81 CARDID
      DATA  DEGREE/0.01745 32925 19943/
      DATA  G     /32.17/
      DATA NWL/1/
      DATA ECHO/4/
C *** OPEN DATA FILE ECHO PRINTOUT FILE
      OPEN(UNIT=ECHO,TYPE='NEW',NAME='ECHO.DAT')
C
      WRITE(ECHO,197)
C *** TITLE
      READ (BIF,199) TITLE
      WRITE(ECHO,196) TITLE
C *** DRAFT (fwd),DRAFT (aft), long. loc's of DRAFT marks
      READ (BIF,200) TF,TA,XFPERP,XAPERP
      WRITE(ECHO,200) TF,TA,XFPERP,XAPERP
C *** Center of Gravity (XCG aft of FP, YCG above BL)
      READ (BIF,200) XCG,YCG,ZCG
      WRITE(ECHO,200) XCG,YCG,ZCG
C *** Six DRAFTs at which hydro. coeffs are computed
      READ(BIF,200) (DRAFT1(I),I=1,6)
      WRITE(ECHO,200) (DRAFT1(I),I=1,6)
C *** Minimum segment lengths for Frank Close Fit
      READ (BIF,201) YMAX,ZMAX,WMAX,NWL
      WRITE(ECHO,201) YMAX,ZMAX,WMAX,NWL
      ADJUST=ZMAX.GT.0.0.AND.YMAX.GT.0.0
```

```
C *** Number of forward profile points
      READ (BIF,190) NFWD
      WRITE(ECHO,190) NFWD
      IF (NFWD.GT.25) CALL ERROR(15,IDUM,RDUM)
C *** Coordinates of forward profile points
      IF (NFWD.GT.0) READ (BIF,430) (YFWD(I),XFWD(I),I=1,NFWD)
      WRITE(ECHO,430) (YFWD(I),XFWD(I),I=1,NFWD)
C *** Number of aft profile points
      READ (BIF,190) NAFT
      WRITE(ECHO,190) NAFT
      IF (NAFT.GT.25) CALL ERROR(16,IDUM,RDUM)
C *** Coordinates of aft profile points
      IF (NAFT.GT.0) READ (BIF,430) (YAFT(I),XAFT(I),I=1,NAFT)
      WRITE(ECHO,430) (YAFT(I),XAFT(I),I=1,NAFT)
C *** Displacement
      READ(BIF,202) DISPL
      WRITE(ECHO,200) DISPL
C *** Radii of gyration
      READ(BIF,202) RADII
      WRITE(ECHO,200) RADII
C *** Ship speed
      READ(BIF,202) SPEED
      WRITE(ECHO,200) SPEED
C *** Linear Damping Constants
      READ(BIF,202) DAMPL
      WRITE(ECHO,200) DAMPL
C *** Quadratic Damping Constants
      READ(BIF,202) DAMPQ
      WRITE(ECHO,200) DAMPQ
C *** Number of wave components
      WRITE(ECHO,203) NWAVES
      READ(BIF,203) NWAVES
C *** Wave component specifications
      DO 140 J=1,NWAVES
      READ(BIF,202) WVAMP(J),WVFRE(J),WVDIR(J),WVPHA(J)
      WRITE(ECHO,200) WVAMP(J),WVFRE(J),WVDIR(J),WVPHA(J)
 140  CONTINUE
C *** Initial position
      READ(BIF,202) POSIT
      WRITE(ECHO,200) POSIT
C *** Initial velocities
      READ(BIF,202) VELOC
      WRITE(ECHO,200) VELOC
C *** Simulation specification
      READ(BIF,202) TSTART,TSTOP,TOUTPT,TSTEP
      WRITE(ECHO,200) TSTART,TSTOP,TOUTPT,TSTEP
C *** Number of weight stations
      READ(BIF,203) NWTSTA
      WRITE(ECHO,203) NWTSTA
C *** Segment weight, rotational gyradii , centroid
      DO 141 J=1,NWTSTA
      READ(BIF,202) SEGWT(J),SEGMOX(J),YBAR(J)
      WRITE(ECHO,200) SEGWT(J),SEGMOX(J),YBAR(J)
 141  CONTINUE
C
       IF(MATFIL.EQ.'NONE') GO TO 1000
       READ(BIF,202) TPSTART,TPSTOP,TPRAMP
       WRITE(ECHO,200) TPSTART,TPSTOP,TPRAMP
       READ(BIF,202) BGX,SMX,BGY,SMY,TSCALE
       WRITE(ECHO,200) BGX,SMX,BGY,SMY,TSCALE
C
C---------------------------------------------------------
C
C     Section 2.0 - READ OFFSET file
C     The offset file can be an actual SHCP DATA File
 1000 CONTINUE
      WRITE(ECHO,198)
C
C *** CARD TYPE A
      READ (OFF,410) CARDID
      WRITE(ECHO,410) CARDID
C *** CARD TYPE B
      READ (OFF,410)
C *** CARD TYPE C
      READ (OFF,412) SPACE,ZSCAL,YSCAL,SHIPL,NAPN,KINDO
      WRITE(ECHO,412)SPACE,ZSCAL,YSCAL,SHIPL,NAPN,KINDO
      IF (SPACE.EQ.0.0) SPACE=1.0
      ZSCAL=1.0
      YSCAL=1.0
      MSTA=0
      NFWD=0
      NAFT=0
```

```
      SUBROUTINE AFSR
C     INTIIALIZES FREE SURFACE
      COMMON/BD/XPAN(120), YPAN(120), ZPAN(120), AREA(120), ST(120),
     * ACN(120), ACNW(120), AN(120, 3), E(120), P(120, 6), PRFS(120),
     *  STOLD(120), PX(120, 6)
      COMMON/FS/AKZ(100, 100), SS(100, 100), CC(100, 100),
     * DKX(100), DKY(100), AKX(100), AKY(100)
      COMMON/FS1/A(100, 100), B(100, 100), AS(100, 100), BS(100, 100)
      COMMON/A/NPAN, NPT, GEE, RHOP, NKX, NKY, EYE, DT, TIM, UFWD
     COMMON/PREXIN/TPSTART, TPSTOP, TPRAMP, BGX, SMX, BGY, SMY, TSCALE
      COMPLEX A, B, AS, BS, EYE
      DKTT=1. 0/(GEE*TSCALE*TSCALE)
        NBX=0. 5/(BGX*DKTT)
        NBY=0. 5/(BGY*DKTT)
      AMX=1. 0/SMX
      AMY=1. 0/SMY
      N=0
      M=0
      AOLD=0. 0
1     IF(N. GE. NBX) GO TO 2
      N=N+1
      AKX(N)=N*N*DKTT
      AOLD=AKX(N)
      IF(AOLD. GE. AMX) NBX=N
      GO TO 1
2     N=N+1
      AKX(N)=AOLD+1. 0/BGX
      AOLD=AKX(N)
      IF(AOLD. LE. AMX) GO TO 2
      NKX=N
      AKX(NKX+1)=AOLD+1. 0/BGX
      AOLD=0. 00
11    IF(M. GE. NBY) GO TO 22
      M=M+1
      AKY(M)=M*M*DKTT
      AOLD=AKY(M)
      IF(AOLD. GE. AMY) NBY=M
      GO TO 11
22    M=M+1
      AKY(M)=AOLD+1. 0/BGY
      AOLD=AKY(M)
      IF(AOLD. LE. AMY) GO TO 22
      NKY=M
      AKY(M+1)=AKY(M)+1. 0/BGY
      AFF=0. 000
      DO 17 N=1, NKX
      AF=(AKX(N+1)+AKX(N))*0. 50
      DKX(N)=AF-AFF
17    AFF=AF
      BFF=0. 00
      DO 18 N=1, NKY
      BF=(AKY(N+1)+AKY(N))*0. 50
      DKY(N)=BF-BFF
18    BFF=BF
      DO 100 N=1, NKX
      DO 100 M=1, NKY
      A(N, M)=(0. 0, 0. 0)
      B(N, M)=(0. 0, 0. 0)
      AS(N, M)=(0. 0, 0. 0)
      BS(N, M)=(0. 0, 0. 0)
      AKZ(N, M)=SQRT(AKX(N)**2+AKY(M)**2)
      SIG=SQRT(GEE*AKZ(N, M))
      SS(N, M)=SIN(SIG*DT)
      CC(N, M)=COS(SIG*DT)
100   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PRESSURE(TP)
C
      COMMON/IO/INPUT,OUTPUT,BIF,OFF,COF,MAT
      INTEGER OUTPUT,BIF,OFF,COF
      INTEGER ECHO
      COMMON /SIGMA /   NK, SIGMA(24), SIGMAO, ERRO, OM(12)
      COMMON/DRFT1/DRAFT1(6)
      COMMON/IOFILE/ OFFIL,BIFIL,COFIL,MATFIL
      COMMON/HEAD/TITLE
      CHARACTER*30 TITLE
      COMMON/COEFF4/COEFF4(6,12,8,25),AREAN(25,6)
      COMMON /      /   NWAVES, WVSUM,
     1                  WVAMP(20), WVFRE(20), WVDIR(20), WVPHA(20),
     2                  WN(20), WNX(20), WNZ(20),
     3                  CSK(20), CCK(20), CXK(20),
     4                  CYK(20), CZK(20), XW(20)
      COMMON /      /   MSTA, NPROF, NFWD, NAFT, LPTS(25),
     1                  XOFF(25), YOFF(25,25), ZOFF(25,25),
     2                  XPROF(51), YPROF(51)
      COMMON /DAMP  /   DAMPL(6), DAMPQ(6)
      COMMON /LHS   /   Y(13)
      COMMON /MASS  /   RHO, G, GAMMA
      COMMON /MASS  /   DISPL, SMASS, XCG, YCG, ZCG,
     1                  AMX, AMY, AMZ, RADII(6)
      COMMON /MASS  /   PMI(3,3)
      COMMON /RESIST/   SPEED
      COMMON /TIME  /   TIME, RAMP
      COMMON /TIME  /   TO, TSTART, TSTOP, TOUTPT, TSTEP, ERR
      COMMON/SXPROP/ SEGMAS(26),SEGMOX(26),STRMAS(26),STRMOM(26),
     *   STRMOX(26),XBAR(26),YBAR(26),SEGWT(26),NWTSTA
      REAL XAFT(25), YAFT(25), XFWD(25), YFWD(25)
      EQUIVALENCE (XFWD(1), XPROF(1)), (YFWD(1),YPROF(1)),
     $            (XAFT(1), XPROF(26)),(YAFT(1),YPROF(26))
      COMMON/PREXIN/TPSTART,TPSTOP,TPRAMP,BGX,SMX,BGY,SMY,TSCALE
        COMMON/BD/XPAN(120),YPAN(120),ZPAN(120),AREA(120),
     *  ST(120),ACN(120),ACNW(120),AN(120,3),E(120),P(120,6),
     *  PRFS(120),STOLD(120),PX(120,6)
        COMMON/FS/AKZ(100,100),SS(100,100),CC(100,100),
     *  DKX(100),DKY(100),AKX(100),AKY(100)
        COMMON/BD2/XPT(150),YPT(150),ZPT(150),WRF(150),
     *  WRFR(150),KK(150,4)
        COMMON/A/NPAN,NPT,GEE,RHOP,NKX,NKY,EYE,DT,TIM,UFWD
      REAL POSIT(6),VELOC(6)
      COMMON/SOURCE/EE(120,120),BPRES(120),PT(6)
      COMMON/DISPLAY/TARE(6)
C
C-------------------------------------------------------------
C
      TIM=TP
      UFWD=YY(7)
C *** Get free surface induce component of normal acceleration
C     at panel centers ACNW(J) and pressure force array PF.
C
      CALL ACPTR
C
C
C *** Get exciting normal acceleration at panel centers
C
      CALL XNA(AXMT,AYMT,AZMT,ARLMT,APMT,AYWMT)
C
C *** Compute panel source strengths
C
      CALL ZBLACN(AXMT,AYMT,AZMT,ARLMT,APMT,AYWMT)
C
C *** Compute body induced forces and pressures
C
      CALL POTB
C
C *** Advance free surface for single time step
C
      CALL CFSR
C
C *** Forces and moments
C
      CALL PRFR(PT)
C
      RETURN
      END
```

```
      SUBROUTINE ACPTR
C     Computes wave incduced accelerations and pressures at
C     panel centers
      COMMON/BD/XPAN(120),YPAN(120),ZPAN(120),AREA(120),ST(120),
     * ACN(120),ACNW(120),AN(120,3),E(120),P(120,6),PRFS(120),
     * STOLD(120),PX(120,6)
      COMMON/FS/AKZ(100,100),SS(100,100),CC(100,100),
     * DKX(100),DKY(100),AKX(100),AKY(100)
      COMMON/FS1/A(100,100),B(100,100),AS(100,100),BS(100,100)
      COMMON/A/NPAN,NPT,GEE,RHOP,NKX,NKY,EYE,DT,TIM,UFWD
      COMPLEX EYE,SC,BX,CX,BY,CY,BYCON,SCON,SK,SKON,
     *  B1,B2,C1,C2
      COMPLEX A,B,AS,BS
      DIMENSION BX(100),CX(100),PF(6),DX(100)
      DO 1500 J=1,NPAN
      AX=AN(J,1)
      AY=AN(J,2)
      AZ=AN(J,3)
       X=XPAN(J)
      Y=YPAN(J)
      Z=ZPAN(J)
      ACT=0.00
      PRT=0.00
      DVDZ=0.00
      DO 16 N=1,NKX
      CCXX=AKX(N)*X
      S=SIN(CCXX)
      C=COS(CCXX)
      BX(N)=CMPLX(C,S)*DKX(N)
      DX(N)=-AKX(N)*AKX(N)*AX
16    CX(N)=AKX(N)*AX*EYE
      DO 163 M=1,NKY
      CCYY=AKY(M)*Y
C     CALL SINCOS(S,C,CCYY)
      S=SIN(CCYY)
      C=COS(CCYY)
      BY=CMPLX(C,S)*DKY(M)
      BYCON=CONJG(BY)
      CY=AKY(M)*AY*EYE
      DO 163 N=1,NKX
      ARGZ=AKZ(N,M)*Z
      DEP=EXP(-ARGZ)
      B1=DEP*BY*BX(N)
      B2=DEP*BYCON*BX(N)
      CZ=AKZ(N,M)*AZ
      SC=B1*A(N,M)
      SCON=B2*AS(N,M)
      SK=B1*B(N,M)
      SKON=B2*BS(N,M)
      C1=CX(N)+CY-CZ
      C2=CX(N)-CY-CZ
      ACT=ACT-C1*SC-C2*SCON
      DVZ=DVZ+EYE*AKX(N)*(C1*SK+C2*SKON)/SQRT(AKZ(N,M))
      PRT=PRT+SC+SCON
163   CONTINUE
C     NORMAL ACCELERATION INDUCED BY FREE SURFACE
      DVDZ=DVZ*UFWD*SQRT(GEE)
      ACNW(J)=ACT*GEE+DVDZ
C     PRESSURE INDUCED BY FREE SURFACE
      PRFS(J)=PRT*GEE*RHOP
1500  CONTINUE
 60   FORMAT(' FREE SURFACE INDUCE PRESSURES')
1070  FORMAT(1X,5F16.8)
      TYPE 60
      TYPE 1070,(PRFS(J),J=1,NPAN)
      RETURN
      END
```

```
        SUBROUTINE CFSR
C ***   Advance free surface wave spectra in time
        COMMON/BD/XPAN(120),YPAN(120),ZPAN(120),AREA(120),ST(120),
     *  ACN(120),ACNW(120),AN(120,3),E(120),P(120,6),PRFS(120),
     *   STOLD(120),PX(120,6)
        COMMON/FS/AKZ(100,100),SS(100,100),CC(100,100),
     *  DKX(100),DKY(100),AKX(100),AKY(100)
        COMMON/FS1/A(100,100),B(100,100),AS(100,100),BS(100,100)
        COMMON/A/NPAN, NPT, GEE, RHOP, NKX, NKY, EYE, DT, TIM, UFWD
        DIMENSION CX(100)
        COMPLEX EYE, AT, ATS, CX, CY, CXY1, CXYS, A, B, AS, BS, DFWD
        DO 100 N=1, NKX
        CXX=AKX(N)*UFWD*DT
        S=SIN(CXX)
        C=COS(CXX)
        DFWD=CMPLX(C, -S)
        DO 100 M=1, NKY
        CT=CC(N, M)
        STT=SS(N, M)
        AT=A(N, M)*CT+B(N, M)*STT
        ATS=AS(N, M)*CT+BS(N, M)*STT
        B(N, M)=B(N, M)*CT-A(N, M)*STT
        BS(N, M)=BS(N, M)*CT-AS(N, M)*STT
        A(N, M)=AT
        AS(N, M)=ATS
C       MOVE FREE SURFACE RELATIVE TO BODY WITH FWD SPEED
        A(N, M)=A(N, M)*DFWD
        B(N, M)=B(N, M)*DFWD
        AS(N, M)=AS(N, M)*DFWD
        BS(N, M)=BS(N, M)*DFWD
100     CONTINUE
  6     CONTINUE
C ***   Add effects of source panels acting over one time step
        DO 1500 J=1, NPAN
        STAR=(ST(J)*0. 50*DT+STOLD(J))*DT
C ST IS TIME RATE OF CHANGE OF SOURCE STRENGTH
C STOLD IS SOURCE STRENGTH AT START OF TIME STEP
C STAR IS AVERAGE VALUE OF STRENGTH OVER THE TIME STEP
        STAR2=STOLD(J)*DT
        STAR=STAR*AREA(J)*0. 6366197724
        STAR2=STAR2*AREA(J)*0. 31831
        X=XPAN(J)
        IF(JTM. GT. 0) X=X-UFWD*DT*. 5
        Y=YPAN(J)
        Z=ZPAN(J)
        DO 93 N=1, NKX
        CXX=AKX(N)*X
        S=SIN(CXX)
        C=COS(CXX)
93      CX(N)=CMPLX(C, -S)
        DO 94 M=1, NKY
        CYY=AKY(M)*Y
        S=SIN(CYY)
        C=COS(CYY)
        CY=CMPLX(C, -S)
        DO 94 N=1, NKX
        ARGZ=AKZ(N, M)*Z
        DEP=EXP(-ARGZ)
        CXY1=CX(N)*CY*DEP
        CXYS=CX(N)*DEP*CONJG(CY)
        A(N, M)=A(N, M)+STAR*CXY1
        AS(N, M)=AS(N, M)+STAR*CXYS
C B(N, M) AND BS(N, M) INCREMENTED NEGLECTING CHANGES IN
C SOURCE STRENGTHS
C OVER TIME INTERVAL ARE SECOND ORDER IN DT
        B(N, M)=B(N, M)-STAR2*CXY1*SS(N, M)
        BS(N, M)=BS(N, M)-STAR2*CXYS*SS(N, M)
94      CONTINUE
1500    CONTINUE
        RETURN
        END
```

```
        SUBROUTINE POTB
C ***   Pressures induced by time rate of change of source
C       strength of panel J in space fixed coordinates
        COMMON/BD/XPAN(120),YPAN(120),ZPAN(120),AREA(120),ST(120),
     *  ACN(120),ACNW(120),AN(120,3),E(120),P(120,6),PRFS(120),
     *   STOLD(120),PX(120,6)
        COMMON/A/NPAN,NPT,GEE,RHOP,NKX,NKY,EYE,DT,TIM,UFWD
        COMMON/SOURCE/EE(120,120),BPRES(120)
        COMPLEX EYE
        DIMENSION PP(6)
        PP(1)=0.00
        PP(2)=0.00
        PP(3)=0.00
        PP(4)=0.00
        PP(5)=0.00
        PP(6)=0.00
        DO 1500 J=1,NPAN
C ST(J) IS TIME RATE OF CHANGE IN HULL FIXED SYSTEM OF SOURCE STRENGTH
C OF PANEL J
C STAV IS AVERAGE SOURCE STRENGTH OVER TIME STEP AT CENTER OF PANEL J
        STAV=STOLD(J)+0.5*DT*ST(J)
C TIME DERIVATIVE IN SPACE FIXED SYSTEM
        DO 1200 K=1,6
        PP(K)=PP(K)+(ST(J)*P(J,K)-STAV*UFWD*PX(J,K))*RHOP
1200    CONTINUE
1500    CONTINUE
        DO 1800 J=1,NPAN
        BPRES(J)=(ST(J)*P(J,3)-STAV*UFWD*PX(J,3))*RHOP
        DENOM=AN(J,3)*AREA(J)
        IF(DENOM.EQ.0.) THEN
          BPRES(J)=0.
        ELSE
          BPRES(J)=BPRES(J)/DENOM
        ENDIF
 1800   CONTINUE
        TYPE *,' BODY INDUCED PRESSURES'
        TYPE 1070,(BPRES(J),J=1,NPAN)
 1070   FORMAT(1X,5F16.8)
        TYPE *
        TYPE 1870
1870    FORMAT(' BODY INDUCED FORCES=')
        TYPE 2020,PP(1),PP(2),PP(3),PP(4),PP(5),PP(6)
2020    FORMAT(3X,3F20.6)
        RETURN
        END
```

```
C
        SUBROUTINE PRFR(PT)
C ***   Computes forces and moments
        COMMON/BD/XPAN(120),YPAN(120),ZPAN(120),AREA(120),ST(120),
     *  ACN(120),ACNW(120),AN(120,3),E(120),P(120,6),PRFS(120),
     *   STOLD(120),PX(120,6)
        COMMON/A/NPAN,NPT,GEE,RHOP,NKX,NKY,EYE,DT,TIM,UFWD
        DIMENSION PF(6)
        COMMON/SOURCE/EE(120,120),BPRES(120)
      COMMON/MASS/RHO,G,GAMMA,DISPL,SMASS,XCG,YCG,ZCG
      COMMON /       /    MSTA,  NPROF,  NFWD,  NAFT,  LPTS(25),
     1                    XOFF(25),  YOFF(25,25),  ZOFF(25,25),
     2                    XPROF(51),  YPROF(51)
        COMPLEX EYE
       COMMON/DISPLAY/TARE(6)
        X1=0.0
        X2=0.0
        X3=0.0
        X4=0.0
        X5=0.0
        X6=0.0
        XCGAM=XCG-(.5*(XOFF(1)-XOFF(MSTA)))
        DO 725 J=1,NPAN
        TPRES=BPRES(J)+PRFS(J)
        PRS=TPRES(J)*AREA(J)
        FRX=-AN(J,1)*PRS
        FRY=-AN(J,2)*PRS
        FRZ=-AN(J,3)*PRS
        XF=XPAN(J+XCGAM
        YF=YPAN(J)
        ZF=ZPAN(J)+TARE(2)
        X1=X1+FRX
        X2=X2+FRY
        X3=X3+FRZ
        X4=X4+YF*FRZ-ZF*FRY
        X5=X5+ZF*FRX-XF*FRZ
        X6=X6+XF*FRY-YF*FRX
725     CONTINUE
        TYPE 80
80      FORMAT(' PRESSURE INDUCED FORCES---')
        TYPE 40,X1,X2,X3,X4,X5,X6
        PT(1)=X1
        PT(2)=X2
        PT(3)=X3
        PT(4)=X4
        PT(5)=X5
        PT(6)=X6
40      FORMAT(5X,3F15.7)
        RETURN
        END
```

```
        SUBROUTINE ZBLACN(ACBX,ACBY,ACBZ,ACBRL,ACBP,ACBYW)
C ***   Computes source strengths satisfying body b.c.
        COMMON/BD/XPAN(120),YPAN(120),ZPAN(120),AREA(120),ST(120)
     *, ACN(120),ACNW(120),AN(120,3),E(120),P(120,6),PRFS(120),
     * STOLD(120),PX(120,6)
        COMMON/A/NPAN,NPT,GEE,RHOP,NKX,NKY,EYE,DT,TIM,UFWD
        COMMON/SOURCE/EE(120,120),BPRES(120)
        COMPLEX EYE
        DO 1800 J=1,NPAN
        X=XPAN(J)
        Y=YPAN(J)
        Z=ZPAN(J)
        ACX=ACBX+ACBP*Z-ACBYW*Y
        ACY=ACBY+ACBYW*X-ACBRL*Z
        ACZ=ACBZ+ACBRL*Y-ACBP*X
C ***   Dot product of body acc. + nrml free-surface indcuded
C       normal acceleration
        ACN(J)=-ACNW(J)+ACX*AN(J,1)+ACY*AN(J,2)+ACZ*AN(J,3)
C ***   Integrate time derivative for total source strenght
C       at start of time step
        STOLD(J)=STOLD(J)+DT*ST(J)
        ST(J)=0.00
 1800   CONTINUE
        DO 1500 J=1,NPAN
        AC=ACN(J)
        DO 1400 K=1,NPAN
        ST(K)=ST(K)+EE(J,K)*AC
1400    CONTINUE
1500    CONTINUE
        RETURN
        END
```

```
C
C *** CARD TYPE D
   30 N=1
      MSTA=MSTA+1
      IF (MSTA.GT.25) CALL ERROR(10,IDUM,RDUM)
      READ (OFF,416) STATNO,Y11,Z1,JTEST
      WRITE(ECHO,417) STATNO,Y11,Z1,JTEST
      XOFF(MSTA)=STATNO*SPACE
      GO TO 50
   40 CONTINUE                         !loop within each station
      N=N+1
      IF (N.GT.25) CALL ERROR(11,MSTA,RDUM)
      READ (OFF,416) S,Y11,Z1,JTEST
      WRITE(ECHO,417) S,Y11,Z1,JTEST
      IF (S.NE.STATNO) CALL ERROR(12,MSTA,RDUM)
   50 YOFF(N,MSTA)=Z1*ZSCAL
      ZOFF(N,MSTA)=Y11*YSCAL
      IF (JTEST.EQ.0 .OR.  JTEST.EQ.77777) GO TO 40
      LPTS(MSTA)=N                               !No.  of points- MSTA
      IF (N.LT.2) CALL ERROR(13,MSTA,RDUM)
      IF (JTEST.EQ.88888) GO TO 30        !Go onto next station
      IF (JTEST.NE.99999) CALL ERROR(14,JTEST,RDUM)
C
C
C
C------------------------------------------------------------------
C
C
C     Section 3.0 READ COEFFICIENT FILE
C
      READ (COF) MSTA                 !Number of Stations
      READ (COF) (OM(I),I=1,12)              !Frequency (rad/sec)
      READ (COF) (DRAFT1(I),I=1,6)            !DRAFTs
      DO 300 L=1,MSTA    !Station index
      DO 300 K=1,6        !DRAFT index
      DO 300 J=1,12       !Frequency index
      READ(COF) (COEFF4(K,J,I,L),I=1,8)
  300 CONTINUE
      READ (COF) ((AREAN(L,K),K=1,6),L=1,MSTA)   !Section areas
C
C     DO 688 L=1,MSTA
C     TYPE 687,  (AREAN(L,K),K=1,6)
C688  CONTINUE
      DO 675 J=1,6
      TARE(J)=POSIT(J)
 675  CONTINUE
 687  FORMAT(1X,6F10.2)
C
C-------------------------------------------------------------
C
C     Section 4.0 READ 3D ARRAY FILE [MAT]
      IF(MATFIL.EQ.'NONE') RETURN
C
      READ(MAT) NPT,NPAN
      DO 500 I=1,3
      READ(MAT) (AN(J,I),J=1,NPAN)
 500  CONTINUE
      READ(MAT) (XPAN(J),J=1,NPAN)
      READ(MAT) (YPAN(J),J=1,NPAN)
      READ(MAT) (ZPAN(J),J=1,NPAN)
      READ(MAT) (AREAP(J),J=1,NPAN)
      DO 505 K=1,6
      READ(MAT) (PX(JL,K),JL=1,NPAN)
 505  CONTINUE
      DO 510 J=1,NPAN
      READ(MAT) (EE(J,K),K=1,NPAN)
 510  CONTINUE
      DO 520 K=1,6
      READ(MAT) (P(J,K),J=1,NPAN)
 520  CONTINUE
C
      CLOSE(BIF)
      CLOSE(OFF)
      CLOSE(COF)
      CLOSE(MAT)
      CLOSE(4)
      RETURN
  180 FORMAT (5X,I5)
  190 FORMAT (I5)
  197 FORMAT(1H1/,81(1H*)/,' INPUT DATA ECHO ',T64,
```

```
     *'PROGRAM SSX'/,81(1H*)//,33(1H-),
     *'[BIF] DATA FILE',32(1H-)/)
  198 FORMAT(1H1/,81(1H*)/,' INPUT DATA ECHO ',T64,
     *'PROGRAM SSX'/,81(1H*)//,33(1H-),
     *'[OFF] DATA FILE',32(1H-)/)
  196 FORMAT(1X,A)
  199 FORMAT(A)
  200 FORMAT (6F10.2)
  201 FORMAT (3F10.2,I5)
  202 FORMAT (8F10.0)
  203 FORMAT (I2)
  210 FORMAT (F10.2,I5,5X,F10.2)
C
  410 FORMAT (A)
  412 FORMAT (4F10.3,13X,I2,4X,I1)
  414 FORMAT (5X,I5,5X,'INPUT OF SHCP TYPE D OFFSET DATA')
  416 FORMAT (F6.3,2F7.0,I6)
  417 FORMAT (F7.3,2F10.2,I6)
  420 FORMAT (2I5,F10.2)
  430 FORMAT (2F10.2)
C
      END
```

```
      SUBROUTINE FINI
C
      COMMON/IO/INPUT,OUTPUT,BIF,OFF,COF
      INTEGER OUTPUT,BIF,OFF,COF
      COMMON /STATS /   NRHS(4), DELTA, HSUM, HMIN, HMAX, TTO, TT1
      COMMON /TIME  /   TIME, RAMP
      COMMON /TIME  /   TO, TSTART, TSTOP, TOUTPT, TSTEP, ERR
      COMMON/SUMMARY/ NT,YYBAR(10),YYRMS(10),YYMAX(10),YYMIN(10)
      IF (TT1.LE.TTO) GO TO 100
      DELTA=TT1-TTO
      NRHS(1)=NRHS(1)+1
      HMIN=AMIN1(DELTA,HMIN)
      HMAX=AMAX1(DELTA,HMAX)
  100 HSUM=HSUM/FLOAT(NRHS(1))
C *** Print execution statistics
      WRITE (OUTPUT,110) NRHS,DELTA,HMIN,HSUM,HMAX,TSTEP,ERR
C *** Compute response statistics
      DO 200 I=1,10
      YYBAR(I)=YYBAR(I)/FLOAT(NT)
      ARGI=(YYRMS(I)/FLOAT(NT)-YYBAR(I)**2)
      IF(ARGI.LT.0) YYRMS(I)=0.
      IF(ARGI.GE.0.) YYRMS(I)=SQRT(ARGI)
C     YYRMS(I)=SQRT(YYRMS(I)/NT-YYBAR(I)**2)
 200  CONTINUE
C *** Print response statistics
      WRITE(OUTPUT,201) ' RMS    ',YYRMS
      WRITE(OUTPUT,201) ' MEAN   ',YYBAR
      WRITE(OUTPUT,201) ' MAX    ',YYMAX
      WRITE(OUTPUT,201) ' MIN    ',YYMIN
C
      RETURN
C
  110 FORMAT (76H0***                     TIME INCREMENT :        FWD   NO STE
     $P       BACK      TOTAL/
     $       36H *** NUMBER OF COMPUTATIONS OF RHS :,4I10/36H *** FINAL V
     1ALUE OF TIME INCREMENT :,1PG13.4/38H *** MINIMUM VALUE OF TIME INC
     2REMENT :,G13.4/38H *** AVERAGE VALUE OF TIME INCREMENT :,G13.4/38H
     3 *** MAXIMUM VALUE OF TIME INCREMENT :,G13.4/42H *** FLAG VALUE, F
     4I, (PLUS 1.0 EXPECTED) :,G13.4/33H *** FINAL VALUE OF ERROR LIMIT
     5:,G13.4)
C
C
 201  FORMAT(A8,4X,4(1X,F11.4),3F12.4,2X,E15.5,2X,F12.4)
C
      END
```

```
      SUBROUTINE PREPARE
C-----------------------------------------------------------------------
      COMMON /C      /   C(3,3)
      COMMON/IO/INPUT,OUTPUT,BIF,OFF,COF
      INTEGER OUTPUT,BIF,OFF,COF
      COMMON /       /   NWAVES, WVSUM,
     1                   WVAMP(20), WVFRE(20), WVDIR(20), WVPHA(20),
     2                   WN(20), WNX(20), WNZ(20),
     3                   CSK(20), CCK(20), CXK(20),
     4                   CYK(20), CZK(20), XW(20)
      COMMON /       /   MSTA, NPROF, NFWD, NAFT, LPTS(25),
     1                   XOFF(25), YOFF(25,25), ZOFF(25,25),
     2                   XPROF(51), YPROF(51)
      COMMON/COEFF4/COEFF4(6,12,8,25),AREAN(25,6)
      COMMON /FORCE /    XF, YF, ZF, XM, YM, ZM
      REAL  FORCE(6)
      EQUIVALENCE (FORCE(1),XF)
      COMMON /LHS   /    Y(13)
      COMMON /MASS  /    RHO, G, GAMMA
      COMMON /MASS  /    DISPL, SMASS, XCG, YCG, ZCG,
     1                   AMX, AMY, AMZ, RADII(6)
      COMMON /MASS  /    PMI(3,3)
      COMMON /RESIST/    SPEED
      COMMON /TIME  /    TIME, RAMP
      COMMON /TIME  /    TO, TSTART, TSTOP, TOUTPT, TSTEP, ERR
      COMMON /V     /    VX, VY, VZ
      COMMON/PREXIN/TPSTART,TPSTOP,TPRAMP,BGX,SMX,BGY,SMY,TSCALE
      REAL  XAFT(25), YAFT(25), XFWD(25), YFWD(25)
      EQUIVALENCE (XFWD(1),XPROF(1)), (YFWD(1),YPROF(1))
      EQUIVALENCE (XAFT(1),XPROF(26)),(YAFT(1),YPROF(26))
      EQUIVALENCE (XXFWD,XPROF(51)),   (XXAFT,YPROF(51))
      COMMON/SXPROP/ SEGMAS(26),SEGMOX(26),STRMAS(26),STRMOM(26),
     *               STRMOX(26),XBAR(26),YBAR(26),
     *               SEGWT(26),NWTSTA
      DATA MAXSTA/25/
      DATA MAXPTS/25/
      DATA AMX/0./,AMY/0./,AMZ/0./
      DATA  RAD   /0.01745 32925 19943/
      GAMMA=RHO*G
      SMASS=DISPL/G
      DISPL=DISPL/GAMMA
      RAMP=TSTART-TO
      AMX=1.0+AMX
      AMY=1.0+AMY
      AMZ=1.0+AMZ
C--------------------------------------------------------------
C *** From given offsets on port side, assign symmetrical
C     offsets on starboard side
      DO 10 J=1,MSTA
         N=LPTS(J)
         NP=N
         DO 50 I=1,N
            NP=NP+1
            NM=N-I+1
            YOFF(NP,J)=YOFF(NM,J)
            ZOFF(NP,J)=-ZOFF(NM,J)
  50    CONTINUE
  60    IF (ZOFF(NP,J).NE.ZOFF(1,J)) GO TO 70
         IF (YOFF(NP,J).EQ.YOFF(1,J)) GO TO 80
  70    NP=NP+1
         YOFF(NP,J)=YOFF(1,J)
         ZOFF(NP,J)=ZOFF(1,J)
  80    IF (NP.GT.MAXPTS) GO TO 90
         IF (NP.GT.1) GO TO 20
  90    CONTINUE
      STOP 'ERROR'
  20     LPTS(J)=NP
  10  CONTINUE
C
C *** Recompute offset coordinates relative to the center of mass
      DO 130 J=1,MSTA
         XOFF(J)=-XOFF(J)-XCG
         N=LPTS(J)
         DO 120 I=1,N
            YOFF(I,J)=YOFF(I,J)-YCG
            ZOFF(I,J)=ZOFF(I,J)-ZCG
 120     CONTINUE
 130  CONTINUE
```

```
C
C *** Consolidate fore & aft profile coordinates into a single
C     pair of arrays
      NPROF=NFWD
      IF (NFWD.EQ.0) GO TO 150
      X=XOFF(1)
      DO 140 I=1,NFWD
         XFWD(I)=XFWD(I)+X
         YFWD(I)=YFWD(I)-YCG
  140 CONTINUE
      IF (NFWD.NE.1) GO TO 150
      NPROF=0
      XXFWD=XFWD(1)
  150 IF (NAFT.EQ.0) GO TO 190
      X=XOFF(MSTA)
         XXAFT=X-XAFT(1)
      IF (NAFT.EQ.1) GO TO 190
      J=1
      IF (NPROF.EQ.0) GO TO 160
      IF (XXAFT.NE.XFWD(NPROF)) GO TO 160
      IF (YPROF(NPROF).EQ.(YAFT(1)-YCG)) J=2
  160 DO 170 I=J,NAFT
         NPROF=NPROF+1
         XPROF(NPROF)=X-XAFT(I)
         YPROF(NPROF)=YAFT(I)-YCG
  170 CONTINUE
      IF (XFWD(1).NE.XFWD(NPROF)) GO TO 180
      IF (YFWD(1).EQ.YFWD(NPROF)) GO TO 190
  180 NPROF=NPROF+1
      XFWD(NPROF)=XFWD(1)
      YFWD(NPROF)=YFWD(1)
 190   CONTINUE
C
C *** This section loads certain strip-associated mass and inertial
C     arrays using mass/inertial data from segment-associated arrays
C *** Load strip & segment associated inertial arrays
      DO 106 I=1,NWTSTA
      SEGMAS(I)=SEGWT(I)/G
 106  CONTINUE
      VX=SPEED
C
C *** Load the inertia matrix
      PMI(1,1)=SMASS*RADII(1)**2
      PMI(2,2)=SMASS*RADII(2)**2
      PMI(3,3)=SMASS*RADII(3)**2
      PMI(2,1)=-ABS(RADII(4))*RADII(4)*SMASS
      PMI(1,2)=PMI(2,1)
      PMI(3,1)=-ABS(RADII(5))*RADII(5)*SMASS
      PMI(1,3)=PMI(3,1)
      PMI(2,3)=-ABS(RADII(6))*RADII(6)*SMASS
      PMI(3,2)=PMI(2,3)
C
C *** Compute wave no's and max wave amplitude
      WVSUM=0.0
      IF (NWAVES.EQ.0) GO TO 260
      DO 250 K=1,NWAVES
         WVPHA(K)=RAD*WVPHA(K)
         WN(K)=WVFRE(K)**2/G
         ARG=WVDIR(K)*RAD
         WNX(K)=WN(K)*COS(ARG)
         WNZ(K)=WN(K)*SIN(ARG)
         WVSUM=WVSUM+ABS(WVAMP(K))
  250 CONTINUE
  260 CONTINUE
C
C *** Convert degrees to radians
      DO 270 I=4,6
         Y(I)=Y(I)*RAD
         Y(I+6)=Y(I+6)*RAD
  270 CONTINUE
C
C *** Convert initial linear velocities from ship coordinates
C     fixed coordinates.
      CALL ROTATE (Y)
      VX=Y(7)
      VY=Y(8)
      VZ=Y(9)
      Y(7)=C(1,1)*VX+C(2,1)*VY+C(3,1)*VZ
      Y(8)=C(1,2)*VX+C(2,2)*VY+C(3,2)*VZ
      Y(9)=C(1,3)*VX+C(2,3)*VY+C(3,3)*VZ
C
```

```
C ***  Divide all 2-D hydro. coeffs by RHO
       DO 110 L=1,MSTA
       DO 110 K=1,6
       DO 110 J=1,12
       DO 110 I=1,8
       COEFF4(K,J,I,L)=COEFF4(K,J,I,L)/RHO
 110   CONTINUE
C
C *** Initialize the free surface for special pressure
C     computations
C
C     IF(TPSTOP.GT.5.) CALL AFSR
C
      RETURN
       END
```

```
       SUBROUTINE COFFEE
CCC
       COMMON/MASS/RHO,G,GAMMA
       COMMON/MASS/DISPL,SMASS,XCG,YCG,ZCG,
     *             AMX,AMY,AMZ,RADII(6)
       COMMON/LHS/Y(13)
C5    WAVE PROPERTIES..
      COMMON /       . /   NWAVES,  WVSUM,
     1                     WVAMP(20),  WVFRE(20),  WVDIR(20),  WVPHA(20),
     2                     WN(20),  WNX(20),  WNZ(20),
     3                     CSK(20),  CCK(20),  CXK(20),
     4                     CYK(20),  CZK(20),  XW(20)
C5
C5
C5    TABLE OF OFFSETS..
      COMMON /        /    MSTA,  NPROF,  NFWD,  NAFT,  LPTS(25),
     1                     XOFF(25),  YOFF(25,25),  ZOFF(25,25),
     2                     XPROF(51),  YPROF(51)
       COMMON/RESIST/SPEED
       COMMON/SIGMA /  NK,SIGMA(24),SIGMAO,  ERRO,  OM(12)
C
C
       COMMON/COEFF4/COEFF4(6,12,8,25),AREAN(25,6)
       COMMON/COEFFX/THAH(25,21),TSAS(25,21),TRAR(25,21),TCCA(25,21),
     *             THVH(25,21),TSVS(25,21),TRVR(25,21),TCCV(25,21),
     *             THACX(25,21),THVCX(25,21),TSACX(25,21),TSVCX(25,21),
     *             TRACX(25,21),TCACX(25,21),THVCY(25,21),
     *             ITP(25),TXP(25),IFP(21),FXP(21)
       COMMON/OPTION/JOBCO,JOBFO(10)
       COMMON/SXOMEG/OMEGAX(3)
       COMMON/DRFT1/ DRAFT1(6)
C
       DIMENSION LIX(21),FMUX(21)
        DATA DEGREE/0.01745 32925/
C
CCC
CCC    (Future revision here... no trim now, but in
CCC     next revision, the options should include
CCC     specifying trim [Y(6)] in addition to draft
CCC     as an initial condition)
CCC
              TAVG=YCG-Y(2)                               !Draft
        TYPE *,' TAVG=',TAVG
        DO 20 JSTA=1,MSTA                                 !Loop over stations
CCC
              CALL TXIT(TAVG,IT,TX)                         !Get index & mult.
       ITP(JSTA)=IT
       TXP(JSTA)=TX
       TYPE *,'IT,TX',IT,TX
CCC
CCC
        DO 10 K=1,NWAVES
CCC
              FREX=ABS((1.0-COS(WVDIR(K)*DEGREE)*SPEED
     *             *WVFRE(K)/G)*WVFRE(K))                 !Encounter freq.
       TYPE *,'Encounter freq=',FREX
C
C-------------------------------------------------------------------
C
C       NOTE: We are using the wave frequency here-- not the
C             frequency of encounter.
```

```
C
C      FREX=WVFRE(K)
C
C
C-----------------------------------------------------------------------
       CALL FXIF(FREX,IF,FX)                              !Get index & mult.
CCC
       IFP(K)=IF
       FXP(K)=FX
       THAH(JSTA,K)=COX(IT,TX,IF,FX,1,JSTA)
       TSAS(JSTA,K)=COX(IT,TX,IF,FX,2,JSTA)
       TRAR(JSTA,K)=COX(IT,TX,IF,FX,3,JSTA)
       TCCA(JSTA,K)=COX(IT,TX,IF,FX,4,JSTA)
C
       THVH(JSTA,K)=COX(IT,TX,IF,FX,5,JSTA)
       TSVS(JSTA,K)=COX(IT,TX,IF,FX,6,JSTA)
       TRVR(JSTA,K)=COX(IT,TX,IF,FX,7,JSTA)
       TCCV(JSTA,K)=COX(IT,TX,IF,FX,8,JSTA)
C
C
C      Spatial derivatives in x-direction
C
       THACX(JSTA,K)=COXDXP(IT,TX,IF,FX,1,JSTA)
       TSACX(JSTA,K)=COXDXP(IT,TX,IF,FX,2,JSTA)
       TRACX(JSTA,K)=COXDXP(IT,TX,IF,FX,3,JSTA)
       TCACX(JSTA,K)=COXDXP(IT,TX,IF,FX,4,JSTA)
C
       THVCX(JSTA,K)=COXDXP(IT,TX,IF,FX,5,JSTA)
       TSVCX(JSTA,K)=COXDXP(IT,TX,IF,FX,6,JSTA)
C      TRVCX(JSTA,K)=COXDXP(IT,TX,IF,FX,7,JSTA)
C      TCVCX(JSTA,K)=COXDXP(IT,TX,IF,FX,8,JSTA)
C
 10    CONTINUE
 20    CONTINUE
C
CCC
CCC    For ship motion related hydro coef's, we will
CCC    us, as a default frequency the frequency associated
CCC    with the first wave [wvfre(1)].
CCC
CCC    Future revision: the default frequencies should
CCC    be the natural period in heave, sway, and roll.
CCC
C
           FRSI=ABS((1.0-COS(WVDIR(1)*DEGREE)*SPEED
    *           *WVFRE(1)/G)*WVFRE(1))           !Encounter freq.
       OMEGAX(1)=FRSI
       OMEGAX(2)=FRSI
       OMEGAX(3)=FRSI
C
      CALL FXIF(FRSI,IF,FX)
C
      DO 30 JSTA=1,MSTA
C
       FXP(21)=FX
       IFP(21)=IF
       TXP(21)=TX
       ITP(21)=IT
C
       THAH(JSTA,21)=COX(IT,TX,IF,FX,1,JSTA)
       TSAS(JSTA,21)=COX(IT,TX,IF,FX,2,JSTA)
       TRAR(JSTA,21)=COX(IT,TX,IF,FX,3,JSTA)
```

```
          TCCA(JSTA,21)=COX(IT,TX,IF,FX,4,JSTA)
C
          THVH(JSTA,21)=COX(IT,TX,IF,FX,5,JSTA)
          TSVS(JSTA,21)=COX(IT,TX,IF,FX,6,JSTA)
          TRVR(JSTA,21)=COX(IT,TX,IF,FX,7,JSTA)
          TCCV(JSTA,21)=COX(IT,TX,IF,FX,8,JSTA)
C
          THACX(JSTA,21)=COXDXP(IT,TX,IF,FX,1,JSTA)
          TSACX(JSTA,21)=COXDXP(IT,TX,IF,FX,2,JSTA)
          THVCX(JSTA,21)=COXDXP(IT,TX,IF,FX,5,JSTA)
          TSVCX(JSTA,21)=COXDXP(IT,TX,IF,FX,6,JSTA)
C
 30     CONTINUE
        RETURN
        END
        FUNCTION COX(IT,TX,IF,FX,ITYPE,JSTA)
CCC
CCC     This function subprogram returns the 2-D hydrodynamic
CCC     coefficient of a particular TYPE (1 thru 8 where 1 is
CCC     HAH, 2 is SAS, and so on) keyed on ITYPE; for a par-
CCC     ticular station, keyed on ISTA. Provided to the function
CCC     is the draft index (IT) and draft interpolation multi-
CCC     plier, the frequency index (IF) and the freqency inter-
CCC     polation multiplier (FX).
CCC
CCC     This function performs a two-dim linear interpolation
CCC     using four values from the big 2-D coefficient array
CCC     COEFF4( , , , ,)...
CCC
        COMMON /COEFF4/ COEFF4(6,12,8,25),AREAN(25,6)
CCC
        A1=COEFF4(IT  ,IF  ,ITYPE,JSTA)
        A2=COEFF4(IT+1,IF  ,ITYPE,JSTA)
        A3=COEFF4(IT  ,IF+1,ITYPE,JSTA)
        A4=COEFF4(IT+1,IF+1,ITYPE,JSTA)
CCC
        B12=A1+(TX*(A2-A1))
        B34=A3+(TX*(A4-A3))
CCC
        COX=B12+(FX*(B34-B12))
CCC
        RETURN
        END
        SUBROUTINE TXIT(TAVG,IT,TX)
CCC
CCC     This routine determines the draft index (IT)
CCC     and draft interpolation multiplier (TX) given
CCC     a value for draft (TAVG).
CCC
        COMMON/DRFT1/ DRAFT1(6)
CCC
        IF (TAVG.GT.DRAFT1(6)) THEN         !Section immersed
           IT=5
           TX=1.0
        ELSE IF (TAVG.LT.DRAFT1(1)) THEN    !Emerged section
           IT=1
           TX=0.0
        ELSE
           DO 10 JJ=2,6
           IF(TAVG-DRAFT1(JJ)) 12,11,10
 11        CONTINUE
```

```
          TX=0.0
          IT=JJ
          GO TO 13
12        CONTINUE
          TX=(TAVG-DRAFT1(JJ-1))/(DRAFT1(JJ)-DRAFT1(JJ-1))
          IT=JJ-1
          GO TO 13
10        CONTINUE
13        CONTINUE
       END IF
       RETURN
       END
       SUBROUTINE FXIF(FREX,IF,FX)
CCC
CCC    This routine determines the freq. index (IF)
CCC    and freq. interpolation multiplier (FX) given
CCC    a value for frequency (FREX).
CCC
       COMMON /SIGMA/ NK,SIGMA(24),SIGMAO,ERRO,OM(12)
CCC
       IF (FREX.GT.OM(11)) THEN
          IF=11
          FX=1.0
       ELSE
          DO 10 JJ=2,12
          IF(FREX-OM(JJ)) 12,11,10
11        CONTINUE
          FX=0.0
          IF=JJ
          GO TO 13
12        CONTINUE
          FX=(FREX-OM(JJ-1))/(OM(JJ)-OM(JJ-1))
          IF=JJ-1
          GO TO 13
10        CONTINUE
13        CONTINUE
       END IF
       RETURN
       END
       FUNCTION COXDX(IT,TX,IF,FX,L,J)
C
      COMMON /      /   NWAVES, WVSUM,
     1                  WVAMP(20), WVFRE(20), WVDIR(20), WVPHA(20),
     2                  WN(20), WNX(20), WNZ(20),
     3                  CSK(20), CCK(20), CXK(20),
     4                  CYK(20), CZK(20), XW(20)
      COMMON /      /   MSTA, NPROF, NFWD, NAFT, LPTS(25),
     1                  XOFF(25), YOFF(25,25), ZOFF(25,25),
     2                  XPROF(51), YPROF(51)
       COMMON/OPTION/JOBCO,JOBFO(10)
       COMMON/ETA/YC,XK(20)
       COMMON/TEMPSTA/DXFWD,DXAFT,TSTA(25)
C
C
C-------------------------------------------------------
C
       IF(J.EQ.1) THEN
          CF=0.
       ELSE
          CF=COX(IT,TX,IF,FX,L,J-1)
          AF=ARX(IT,TX,J-1)
```

```
             CF=CF*AF
          END IF
C
             CM=COX(IT,TX,IF,FX,L,J)
             AM=ARX(IT,TX,J)
             CM=CM*AM
C
          IF(J.EQ.MSTA) THEN
             CA=0.
          ELSE
             CA=COX(IT,TX,IF,FX,L,J+1)
             AA=ARX(IT,TX,J+1)
             CA=CA*AA
          END IF
C
          C1=(CM-CF)/DXFWD
          C2=(CA-CM)/DXAFT
C
          COXDX=(.5/AM)*(C1+C2)
C
          RETURN
          END
          FUNCTION COXDXP(IT,TX,IF,FX,L,J)
C
         COMMON /         /    NWAVES, WVSUM,
     1                         WVAMP(20), WVFRE(20), WVDIR(20), WVPHA(20),
     2                         WN(20), WNX(20), WNZ(20),
     3                         CSK(20), CCK(20), CXK(20),
     4                         CYK(20), CZK(20), XW(20)
         COMMON /         /    MSTA, NPROF, NFWD, NAFT, LPTS(25),
     1                         XOFF(25), YOFF(25,25), ZOFF(25,25),
     2                         XPROF(51), YPROF(51)
C
C
C-----------------------------------------------------------------------
C
          IF(J.EQ.1) THEN
             CF=0.
             DXFWD=XOFF(J+1)-XOFF(J)
          ELSE
             CF=COX(IT,TX,IF,FX,L,J-1)
             AF=ARX(IT,TX,J-1)
             CF=CF*AF
             DXFWD=XOFF(J)-XOFF(J-1)
          END IF
C
             CM=COX(IT,TX,IF,FX,L,J)
             AM=ARX(IT,TX,J)
             CM=CM*AM
C
          IF(J.EQ.MSTA) THEN
             CA=0.
             DXAFT=XOFF(J)-XOFF(J-1)
          ELSE
             CA=COX(IT,TX,IF,FX,L,J+1)
             AA=ARX(IT,TX,J+1)
             CA=CA*AA
             DXAFT=XOFF(J+1)-XOFF(J)
          END IF
C
          C1=(CM-CF)/DXFWD
```

```
      C2=(CA-CM)/DXAFT
C
C     COXDXP=(.5/AM)*(C1+C2)
      TDX=DXFWD+DXAFT
      DCX=CA-CF
      COXDXP=(DCX/TDX)/AM
C
      RETURN
      END
      FUNCTION ARX(IT,TX,J)
CCC
      COMMON /COEFF4/ COEFF4(6,12,8,25),AREAN(25,6)
CCC
CCC
CCC
      B2=AREAN(J,IT+1)
      B1=AREAN(J,IT)
      ARX=B1+(TX*(B2-B1))
CCC
      RETURN
      END
```

```
      FUNCTION ETABAR(I)
C
C1    ETABAR RETURNS THE WAVE ELEVATION (ABSOLUTE COORDINATES) ABOVE
C1    A POINT IN THE SHIP COORDINATE SYSTEM.
C1
C2    POINT IS ON THE SHIP CENTERPLANE.
C2    THE COORDINATES OF THE POINT ON THE SHIP ARE..
C2       ( XPROF(I), YPROF(I), -ZCG )
C2
C5
C5    PHYSICAL CONSTANTS..
      COMMON /MASS  /   RHO, G, GAMMA
C5
C5    SHIP MASS PARAMETERS..
      COMMON /MASS  /   DISPL, SMASS, XCG, YCG, ZCG,
     1                  AMX, AMY, AMZ, RADII(6)
C5
C5    ROTATIONAL INERTIA MATRIX FOR FIXED MASS..
      COMMON /MASS  /   PMI(3,3)
C5
      COMMON /TRIG  /   CTHETA, STHETA, CPHI, SPHI, CPSI, SPSI
C5
C5    WAVE PROPERTIES..
      COMMON /      /   NWAVES, WVSUM,
     1                  WVAMP(20), WVFRE(20), WVDIR(20), WVPHA(20),
     2                  WN(20), WNX(20), WNZ(20),
     3                  CSK(20), CCK(20), CXK(20),
     4                  CYK(20), CZK(20), XW(20)
C5
C5
C5    TABLE OF OFFSETS..
      COMMON /      /   MSTA, NPROF, NFWD, NAFT, LPTS(25),
     1                  XOFF(25), YOFF(25,25), ZOFF(25,25),
     2                  XPROF(51), YPROF(51)
C5
C3
      ETABAR=0.0
      IF (NWAVES.EQ.0) RETURN
      X1=XPROF(I)
      Y1=YPROF(I)*CTHETA+ZCG*STHETA
      Z1=YPROF(I)*STHETA-ZCG*CTHETA
      DO 110 K=1,NWAVES
         ARG=(X1-XW(K))*CXK(K)+Y1*CYK(K)+Z1*CZK(K)
         ETABAR=COS(ARG)*WVAMP(K)+ETABAR
  110 CONTINUE
      RETURN
C
      END
      FUNCTION ETAF (ZT)
C
C1    FIND COORDINATE OF WAVE SURFACE.
C1
C5
C5    WAVE PROPERTIES..
      COMMON /      /   NWAVES, WVSUM,
     1                  WVAMP(20), WVFRE(20), WVDIR(20), WVPHA(20),
     2                  WN(20), WNX(20), WNZ(20),
     3                  CSK(20), CCK(20), CXK(20),
     4                  CYK(20), CZK(20), XW(20)
C5
C5
```

```
C5    DATA FOR THE COMPUTATION OF THE WAVE SURFACE ELAVATION..
      COMMON /ETA   /   YC, XK(20)
C5
      COMMON /CT    /   CT(3,3)
C3
      A=-YC
      B=CT(2,2)
      IF (NWAVES.EQ.0) GO TO 120
      DO 110 K=1,NWAVES
         ARG=ZT*CZK(K)+XK(K)
         A=COS(ARG)*WVAMP(K)+A
         B=SIN(ARG)*WVAMP(K)*CYK(K)+B
  110 CONTINUE
  120 ETAF=A/B
      RETURN
C
      END
      FUNCTION ETAY(J,K)
      COMMON /TRIG  /   CTHETA, STHETA, CPHI, SPHI, CPSI, SPSI
      COMMON /      /   NWAVES, WVSUM,
     1                  WVAMP(20), WVFRE(20), WVDIR(20), WVPHA(20),
     2                  WN(20), WNX(20), WNZ(20),
     3                  CSK(20), CCK(20), CXK(20),
     4                  CYK(20), CZK(20), XW(20)
      COMMON /      /   MSTA, NPROF, NFWD, NAFT, LPTS(25),
     1                  XOFF(25), YOFF(25,25), ZOFF(25,25),
     2                  XPROF(51), YPROF(51)
      ETAY=0.
      IF (NWAVES.EQ.0) RETURN
      X1=XOFF(J)
      Y1=0.
      Z1=0.
         ARG=(X1-XW(K))*CXK(K)+Y1*CYK(K)+Z1*CZK(K)
         ETAY=COS(ARG)*WVAMP(K)
      RETURN
C
      END
```

```
      SUBROUTINE FOLIO1
C
      COMMON/IO/INPUT,OUTPUT,BIF,OFF,COF
      INTEGER OUTPUT,BIF,OFF,COF
      COMMON /SIGMA /   NK, SIGMA(24), SIGMAO, ERRO, OM(12)
      COMMON/DRFT1/DRAFT1(6)
      COMMON/IOFILE/ OFFIL,BIFIL,COFIL
      COMMON/HEAD/TITLE
      CHARACTER*30 TITLE
      COMMON/COEFF4/COEFF4(6,12,8,25),AREAN(25,6)
      COMMON /      /   NWAVES, WVSUM,
     1                  WVAMP(20), WVFRE(20), WVDIR(20), WVPHA(20),
     2                  WN(20), WNX(20), WNZ(20),
     3                  CSK(20), CCK(20), CXK(20),
     4                  CYK(20), CZK(20), XW(20)
      COMMON /      /   MSTA, NPROF, NFWD, NAFT, LPTS(25),
     1                  XOFF(25), YOFF(25,25), ZOFF(25,25),
     2                  XPROF(51), YPROF(51)
      COMMON /DAMP  /   DAMPL(6), DAMPQ(6)
      COMMON /LHS   /   Y(13)
      COMMON /MASS  /   RHO, G, GAMMA
      COMMON /MASS  /   DISPL, SMASS, XCG, YCG, ZCG,
     1                  AMX, AMY, AMZ, RADII(6)
      COMMON /MASS  /   PMI(3,3)
      COMMON /RESIST/   SPEED
      COMMON/OPTION/JOBCO,JOBFO(10)
      COMMON /TIME  /   TIME, RAMP
      COMMON /TIME  /   TO, TSTART, TSTOP, TOUTPT, TSTEP, ERR
      COMMON/SXPROP/ SEGMAS(26),SEGMOX(26),STRMAS(26),STRMOM(26),
     *   STRMOX(26),XBAR(26),YBAR(26),SEGWT(26),NWTSTA
      REAL POSIT(6),VELOC(6)
      EQUIVALENCE (POSIT(1),Y(1)),(VELOC(1),Y(7))
C
      WEIGHT=SMASS*G
      WRITE(OUTPUT,100) TITLE
      WRITE(OUTPUT,110) WEIGHT,RADII(1),RADII(2),XCG,RADII(3),
     *                  YCG,RADII(4),ZCG,RADII(5),RADII(6)
      WRITE(OUTPUT,120) DAMPL(1),DAMPQ(1),SPEED,
     *                  (DAMPL(I),DAMPQ(I),I=2,6)
      WRITE(OUTPUT,130) (POSIT(I),VELOC(I),I=1,6)
      WRITE(OUTPUT,140) TSTART,TSTOP,TOUTPT,TSTEP
      WRITE(OUTPUT,160) RHO,G
      WRITE(OUTPUT,165) JOBCO,(JOBFO(I),I=1,3)
      WRITE(OUTPUT,100) TITLE
      WRITE(OUTPUT,150)
      DO 10 I=1,NWAVES
      WRITE(OUTPUT,155) I,WVAMP(I),WVFRE(I),WVDIR(I),WVPHA(I)
 10   CONTINUE
      WRITE(OUTPUT,170)
      DO 20 I=1,NWTSTA
      WRITE(OUTPUT,175) I,SEGWT(I),SEGMOX(I),YBAR(I)
 20   CONTINUE
      WRITE(OUTPUT,100) TITLE
C
 100  FORMAT(1H1,80(1H=)/,1X,A,T55,'PROGRAM SSX (Version 1.0)'/,
     *81(1H=)//)
 110  FORMAT(28(1H-),'Inertial Characteristics',29(1H-)/,
     *                             T48,9(1H-),'GYRADII',8(1H-)/,
     *T5,'WEIGHT',F15.2,2X,'L.TONS',  T48,'Kxx',4X,F10.3,5X,'ft'/,
     *                             T48,'Kyy',4X,F10.3,5X,'ft'/,
     *T5,'XCG',10X,F10.3,4X,'ft',  T48,'Kzz',4X,F10.3,5X,'ft'/,
```

```
     *T5,'YCG',10X,F10.3,4X,'ft',  T48,'Kxy',4X,F10.3,5X,'ft'/,
     *T5,'ZCG',10x,F10.3,4x,'ft',  T48,'Kxz',4X,F10.3,5X,'ft'/,
     *                            T48,'Kyz',4X,F10.3,5X,'ft'/,40(2H -)/)
 120  FORMAT(T5,8(1H-),'DAMPING COEFFICIENTS',8(1H-),T48,
     *                '----OPS DATA----'/,
     *13X,'(user specified)'/,
     *T5,'MOTION',7X,'LINEAR',7X,'QUADRATIC'/,
     *T5,'------',4X,'---------',5X,'----------'/,
     *T5,'SURGE',2(5X,E10.4),T48,'SPEED',5X,F6.2/,
     *T5,'HEAVE',2(5X,E10.4)/,
     *T5,'SWAY ',2(5X,E10.4)/,
     *T5,'ROLL ',2(5X,E10.4)/,
     *T5,'YAW  ',2(5X,E10.4)/,
     *T5,'PITCH',2(5X,E10.4)//)
 130  FORMAT(32(1H-),'Initial Conditions',31(1H-)//,
     *T5,11(1H-),'POSITION',11(1H-),T48,10(1H-),'VELOCITY',10(1H-)/,
     *T5,'X(CG)',11X,F10.3,2X,'ft',T48,'SPEED',5X,F10.3,2X,'ft/sec'/,
     *T5,'Y(CG)',11X,F10.3,2X,'ft',T48,'HEAVE',5X,F10.3,2X,'ft/sec'/,
     *T5,'Z(CG)',11X,F10.3,2X,'ft',T48,'SWAY ',5X,F10.3,2X,'ft/sec'/,
     *T5,'ROLL ',11X,F10.3,1X,'deg',T48,'ROLL',6X,F10.3,2X,'deg/sec'/,
     *T5,'YAW  ',11X,F10.3,1X,'deg',T48,'YAW',7X,F10.3,2X,'deg/sec'/,
     *T5,'PITCH',11X,F10.3,1X,'deg',T48,'PITCH',5X,F10.3,2X,
     *'deg/sec'//)
 140  FORMAT(34(1H-),'Run Parameters',33(1H-)//,
     *T5,'SIMULATION PARAMETERS',T48,'JOB OPTIONS'/,
     *T5,21(1H-),T48,11(1H-)/,
     *T5,'TIME(START)',3X,F8.2/,
     *T5,'TIME(END)',5X,F8.2/,
     *T5,'OUTPUT DT',5X,F8.2/,
     *T5,'INTEG.  DT',5X,F8.2/,81(1H-)//)
 150  FORMAT(33(1H-),'WAVE COMPONENTS',33(1H-)//,
     *T5,'WAVE',5X,'AMPLITUDE',5X,'FREQUENCY',5X,'DIRECTION',
     *5X,'PHASE'/,
     *T5,'----',5X,'---------',5X,'---------',5X,'---------',
     *5X,'-----')
 155  FORMAT(T7,I2,T16,F6.2,T30,F6.2,T44,F7.2,T55,F7.2)
 160  FORMAT(//31(1H-),'MISCELLANEOUS VALUES',30(1H-)//,
     *T5,'RHO',5X,F10.6/,T5,'G  ',5X,F10.6)
 165  FORMAT(T5,'JOBCO',I2/,T5,'JOBFO',4I4//)
 170  FORMAT(27(1H-),'SECTIONAL WEIGHT PROPERTIES',27(1H-)//,
     *T5,'SECTION',5X,'WEIGHT',5X,'  Kxx  ',5X,'Centroid'/,
     *T5,'-------',5X,'------',5x,'-------',5X,'--------'//)
 175  FORMAT(T5,3X,I2,T13,F10.3,T26,F8.2,T39,F8.2)
      RETURN
      END
```

```
      SUBROUTINE RKF (TIME,Y,TNEXT,ERR,NEQS,STEP)
C
C     RUNGE-KUTTA INTEGRATION FOR CAPSIZE.
C
      REAL  Y(1)
      REAL  YY(13), YA(13), YB(13)
C------------------------------------
      REAL YC(13)
      DATA YC(13)/0.0/
C
      DATA  YY(13)/0.0/
      DATA  YA(13)/0.0/
      DATA  YB(13)/0.0/
C
      STEP=ABS(STEP)
      NSTEP=(TNEXT-TIME)/STEP+0.5
      IF (NSTEP.LT.1) NSTEP=1
      TSTEP=(TNEXT-TIME)/FLOAT(NSTEP)
      HALF=TSTEP/2.0
      FACT=TSTEP/6.0
  110 CALL DERIV (TIME,Y,YA)
      DO 120 I=1,NEQS
         YY(I)=HALF*YA(I)+Y(I)
  120 CONTINUE
      TIME=HALF+TIME
      CALL DERIV (TIME,YY,YB)
      DO 130 I=1,NEQS
         YA(I)=2.0*YB(I)+YA(I)
         YY(I)=HALF*YB(I)+Y(I)
  130 CONTINUE
      CALL DERIV (TIME,YY,YB)
      DO 140 I=1,NEQS
         YA(I)=2.0*YB(I)+YA(I)
         YY(I)=TSTEP*YB(I)+Y(I)
  140 CONTINUE
      TIME=HALF+TIME
      CALL DERIV (TIME,YY,YB)
      DO 150 I=1,NEQS
         Y(I)=(YA(I)+YB(I))*FACT+Y(I)
  150 CONTINUE
C--------------------------------------------------------------
C     This call to DERIV is to get actual acceleration for the
C     new TIME.  By calling RHS this time we are loading the
C     acceleration array ACC( ) in COMMON ACCEL with correct
C     accelerations, rather than accel. computed as dictated
C     by the fourth-order RUNGA-KUTTA scheme.
C
      CALL DERIV(TIME,Y,YC)
C     CALL GETFREQ
C--------------------------------------------------------------
      IF (TIME+HALF.LT.TNEXT) GO TO 110
      TIME=TNEXT
      RETURN
      END
```

```
      SUBROUTINE DERIV (TT, YY, YYDOT)
C     Evaluates derivatives in equations of motion
C-------------------------------------------------------------
C
C        TT    -- TIME, THE INDEPENDENT VARIABLE
C
C        YY    -- DEPENDENT VARIABLE ARRAY. .
C                 YY(1)  = X-COORDINATE OF MASS CENTER
C                 YY(2)  = Y-COORDINATE OF MASS CENTER
C                 YY(3)  = Z-COORDINATE OF MASS CENTER
C                 YY(4)  = THETA ROTATION (ROLL ANGLE)
C                 YY(5)  = PHI   ROTATION (YAW ANGLE)
C                 YY(6)  = PSI   ROTATION (PITCH ANGLE)
C                 YY(7)  = X-COMPONENT OF LINEAR VELOCITY
C                 YY(8)  = Y-COMPONENT OF LINEAR VELOCITY
C                 YY(9)  = Z-COMPONENT OF LINEAR VELOCITY
C                 YY(10) = ANGULAR VELOCITY ABOUT X-AXIS (ROLL RATE)
C                 YY(11) = ANGULAR VELOCITY ABOUT Y-AXIS (YAW RATE)
C                 YY(12) = ANGULAR VELOCITY ABOUT Z-AXIS (PITCH RATE)
C                 YY(13) = TIME INTEGRAL OF YAW ANGLE (EVALUATED ONLY
C                          WHEN REQUIRED BY THE AUTOPILOT)
C
C        YYDOT -- DERIVATIVES OF YY.
C
C     TT AND YY ARE SUPPLIED TO THIS SUBROUTINE WHENEVER THE EQUATION
C     SOLVING ROUTINE REQUIRES THE VALUES OF THE DERIVATIVE OF YY--
C     YYDOT.  YYDOT IS COMPUTED BY EVALUATING THE RIGHT HAND SIDE OF
C     THE EQUATIONS OF MOTION FOR THE GIVEN TT AND YY AND THE PRE-
C     DEFINED GEOMETRY OF THE SHIP AND EQUATIONS OF THE WAVES.
C
      REAL YY(13), YYDOT(13)
      COMMON /A      /   A(6,6)
      COMMON /C      /   C(3,3)
      COMMON /CT     /   CT(3,3)
      COMMON /FORCE  /   XF, YF, ZF, XM, YM, ZM
      REAL  FORCE(6)
      EQUIVALENCE (FORCE(1),XF)
      COMMON /H      /   H(3)
      COMMON /MASS   /   RHO, G, GAMMA
      COMMON /MASS   /   DISPL, SMASS, XCG, YCG, ZCG,
     1                   AMX, AMY, AMZ, RADII(6)
      COMMON /MASS   /   PMI(3,3)
      COMMON /STATS  /   NRHS(4), DELTA, HSUM, HMIN, HMAX, TTO, TT1
      COMMON /TRIG   /   CTHETA, STHETA, CPHI, SPHI, CPSI, SPSI
      COMMON /Z      /   SCR(6)
      COMMON/ACCEL/ACC(6)
      NRHS(4)=NRHS(4)+1
      IF (TT.GT.TT1) GO TO 90
      IF (TT.EQ.TT1) GO TO 80
C *** Previous time step rejected
      NRHS(3)=NRHS(3)+1
      TTO=AMIN1(TT,TTO)
      TT1=TTO
      GO TO 100
C *** Previous time step repeated
   80 NRHS(2)=NRHS(2)+1
      GO TO 100
C
C *** Previous time step accepted
   90 DELTA=TT1-TTO
      TTO=TT1
      TT1=TT
```

```
      IF (DELTA.LE.0.0) GO TO 100
      NRHS(1)=NRHS(1)+1
      HSUM=HSUM+DELTA
      HMIN=AMIN1(HMIN,DELTA)
      HMAX=AMAX1(HMAX,DELTA)
  100 CONTINUE
C *** Compute forces on ship
      CALL FORTIS (TT,YY)
C *** Set the values of the derivatives
C *** Set up the H vector (angular momentum)
      H(1)=PMI(1,1)*YY(10)+PMI(1,2)*YY(11)+PMI(1,3)*YY(12)
      H(2)=PMI(2,1)*YY(10)+PMI(2,2)*YY(11)+PMI(2,3)*YY(12)
      H(3)=PMI(3,1)*YY(10)+PMI(3,2)*YY(11)+PMI(3,3)*YY(12)
C *** Set derivatives of position and rotation
      DO 110 I=1,3
         YYDOT(I)=YY(I+6)
  110 CONTINUE
      YYDOT(5)=(YY(11)*CTHETA-YY(12)*STHETA)/CPSI
      YYDOT(4)=YY(10)-YYDOT(5)*SPSI
      YYDOT(6)=YY(11)*STHETA+YY(12)*CTHETA
C *** Solve for derivatives of linear and angular velocity
      YYDOT(7)=XF/SMASS
      YYDOT(8)=YF
      YYDOT(9)=ZF
      YYDOT(10)=XM-YY(11)*H(3)+YY(12)*H(2)
      YYDOT(11)=YM-YY(12)*H(1)+YY(10)*H(3)
      YYDOT(12)=ZM-YY(10)*H(2)+YY(11)*H(1)
      DET=0.0
      L=LNEGF(6,5,1,A(2,2),YYDOT(8),DET,SCR)
      IF (L.NE.1) STOP 1
C *** Load the acceleration array ACC [ship coordinates]
      ACC(1)=YYDOT(7)
      ACC(2)=YYDOT(8)
      ACC(3)=YYDOT(9)
      ACC(4)=YYDOT(10)
      ACC(5)=YYDOT(11)
      ACC(6)=YYDOT(12)
C
C *** Resolve linear accelerations into fixed coordinates.
      SCR(1)=YYDOT(7)
      SCR(2)=YYDOT(8)
      SCR(3)=YYDOT(9)
      YYDOT(7)=C(1,1)*SCR(1)+C(2,1)*SCR(2)+C(3,1)*SCR(3)
      YYDOT(8)=C(1,2)*SCR(1)+C(2,2)*SCR(2)+C(3,2)*SCR(3)
      YYDOT(9)=C(1,3)*SCR(1)+C(2,3)*SCR(2)+C(3,3)*SCR(3)
      YYDOT(13)=YY(5)
      RETURN
      END
```

```
      SUBROUTINE ROTATE (YY)
C
C1    SET UP COORDINATE ROTATION MATRICES  CT  AND  C .
C1    RESOLVE VELOCITY INTO SHIP COORDINATES.
C1
      REAL YY(13)
      COMMON /C     /   C(3,3)
      COMMON /CT    /   CT(3,3)
      COMMON /TRIG  /   CTHETA, STHETA, CPHI, SPHI, CPSI, SPSI
      COMMON /V     /   VX, VY, VZ
      CTHETA=COS(YY(4))
      STHETA=SIN(YY(4))
      CPHI=COS(YY(5))
      SPHI=SIN(YY(5))
      CPSI=COS(YY(6))
      SPSI=SIN(YY(6))
C3
C3    SET UP THE   CT   MATRIX..
C3
      CT(1,1)=CPHI*CPSI
      CT(1,2)=SPSI
      CT(1,3)=-SPHI*CPSI
      CT(2,1)=-CPHI*SPSI
      CT(2,2)=CPSI
      CT(2,3)=SPHI*SPSI
      CT(3,1)=SPHI
      CT(3,2)=0
      CT(3,3)=CPHI
C3
C3    SET UP THE   C   MATRIX..
C3
      C(1,1)=CT(1,1)
      C(1,2)=CT(1,2)
      C(1,3)=CT(1,3)
      C(2,1)=CTHETA*CT(2,1)+STHETA*CT(3,1)
      C(2,2)=CTHETA*CT(2,2)
      C(2,3)=CTHETA*CT(2,3)+STHETA*CT(3,3)
      C(3,1)=CTHETA*CT(3,1)-STHETA*CT(2,1)
      C(3,2)=-STHETA*CT(2,2)
      C(3,3)=CTHETA*CT(3,3)-STHETA*CT(2,3)
C3
C3    RESOLVE THE LINEAR VELOCITY INTO SHIP COORDINATES..
C3
      VX=C(1,1)*YY(7)+C(1,2)*YY(8)+C(1,3)*YY(9)
      VY=C(2,1)*YY(7)+C(2,2)*YY(8)+C(2,3)*YY(9)
      VZ=C(3,1)*YY(7)+C(3,2)*YY(8)+C(3,3)*YY(9)
C3
      RETURN
C
      END
```

```
       SUBROUTINE GETFREQ
C
C********************************************************
C
C        Evaluates frequency of accelerations of each
C        strip in heave, sway and roll using recent
C        history. The routine finds the last several
C        peaKs or troughs in the response history. Evaluates
C        the time difference and calculates the frequency
C        (rad/sec) and loads the OMEGAX( , ) array.
C
C***********************************************************
C
C5     WAVE PROPERTIES..
       COMMON /        /    NWAVES, WVSUM,
     1                      WVAMP(20), WVFRE(20), WVDIR(20), WVPHA(20),
     2                      WN(20), WNX(20), WNZ(20),
     3                      CSK(20), CCK(20), CXK(20),
     4                      CYK(20), CZK(20), XW(20)
C5
C5
C5     TABLE OF OFFSETS..
       COMMON /        /    MSTA, NPROF, NFWD, NAFT, LPTS(25),
     1                      XOFF(25), YOFF(25,25), ZOFF(25,25),
     2                      XPROF(51), YPROF(51)
        COMMON/ACCEL/ ACC(6)
        COMMON/SXGEOM/AREAS(25),DXS(25),DX1S(25),DX2S(25),
     *                YMOMS(25),ZMOMS(25),ISTA,LSTA
        COMMON/SXOMEG/OMEGAX(3)
        COMMON/HIST/AHIST(3,30),TIMTK(30)
        COMMON/TIME/TIME,RAMP
        COMMON/TIME/TO,TSTART,TSTOP,TOUTPT,TSTEP,ERR
        REAL TPK(10)
C
C-----------------------------------------------------------
C       SECTION 1.0 - Move values up one index in the AHIST
C                     array
C
        DO 100 J=30,2,-1
        JM1=J-1
        AHIST(1,J)=AHIST(1,JM1)
        AHIST(2,J)=AHIST(2,JM1)
        AHIST(3,J)=AHIST(3,JM1)
C100    CONTINUE
C
        DO 110 I=30,2,-1
        IM=I-1
        TIMTK(I)=TIMTK(IM)
110     CONTINUE
C-----------------------------------------------------------
C
C      SECTION 2.0 - Calculate acceleration of a strip J in
C                    heave,sway and roll using the ACC(I)
C                    array giving ship accelerations in ship
C                    coordinate system.
C
C
       XDUM=10.   !Arbitrary moment arm to get better
C                   defined history
       AH=ACC(2)+ACC(6)*XDUM
       AS=ACC(3)-ACC(5)*XDUM
```

```
      AR=ACC(4)
C
      AHIST(1,1)=AH
      AHIST(2,1)=AS
      AHIST(3,1)=AR
C
 200  CONTINUE
C
       TIMTK(1)=TIME
C
      IF(TIME.LE.RAMP) GO TO 400
C
C------------------------------------------------------------
C
C     SECTION 3.0 - Find peaks and troughs, evaluate freq.
C                   [Note: Here we are evaluating frequency
C                   using only the last quarter of a "cycle".
C                   If we wanted to use a half cycle, we
C                   would get out of the GOTO310 loop when
C                   KNT=3 instead of KNT=2 and change
C                   the 'TQ=...' line to
C                   TQ=TPK(3)-TPK(1) followed by
C                   OMEGAX(I)=2.*PI/(2.*TQ)
C
C------------------------------------------------------------
C
       DO 300 I=1,3
       L=0
       KNT=0
       TSIGN=1.
 310   CONTINUE
       L=L+1
       IF(L.EQ.30) GO TO 325
       TSIGN2=TSIGN
       DIF=AHIST(I,L)-AHIST(I,L+1)
       TSIGN=SIGN(1.,DIF)
       IF(L.EQ.1) GO TO 310
       IF(TSIGN2.EQ.TSIGN) GO TO 310
       KNT=KNT+1
       TPK(KNT)=TIMTK(L)
       IF(KNT.EQ.2) GO TO 320
       GO TO 310
C
 320    CONTINUE
        TQ=TPK(1)-TPK(2)
        OMEGAX(I)=6.2831853/(4.*TQ)
        GO TO 300
 325    OMEGAX(I)=0.
C
 300    CONTINUE
C
 400    CONTINUE
        RETURN
        END
```

```
       SUBROUTINE OUTCOF
CCC
       COMMON/IO/INPUT,OUTPUT,BIF,OFF,COF
       INTEGER OUTPUT,BIF,OFF,COF
       COMMON/MASS/RHO,G,GAMMA
       COMMON/MASS/DISPL,SMASS,XCG,YCG,ZCG,
     *              AMX,AMY,AMZ,RADII(6)
       COMMON/LHS/Y(13)
      COMMON /      /    NWAVES,  WVSUM,
     1                   WVAMP(20),  WVFRE(20),  WVDIR(20),  WVPHA(20),
     2                   WN(20),  WNX(20),  WNZ(20),
     3                   CSK(20),  CCK(20),  CXK(20),
     4                   CYK(20),  CZK(20),  XW(20)
      COMMON /      /    MSTA,  NPROF,  NFWD,  NAFT,  LPTS(25),
     1                   XOFF(25),  YOFF(25,25),  ZOFF(25,25),
     2                   XPROF(51),  YPROF(51)
       COMMON/RESIST/SPEED
       COMMON/SIGMA /  NK,SIGMA(24),SIGMAO,  ERRO,  OM(12)
C
C
       COMMON/COEFF4/COEFF4(6,12,8,25),AREAN(25,6)
       COMMON/COEFFX/THAH(25,21),TSAS(25,21),TRAR(25,21),TCCA(25,21),
     *             THVH(25,21),TSVS(25,21),TRVR(25,21),TCCV(25,21),
     *             THACX(25,21),THVCX(25,21),TSACX(25,21),TSVCX(25,21),
     *             TRACX(25,21),TCACX(25,21),THVCY(25,21),
     *             ITP(25),TXP(25),IFP(21),FXP(21)
       COMMON/OPTION/JOBCO,JOBFO(10)
       COMMON/SXOMEG/OMEGAX(3)
       COMMON/DRFT1/ DRAFT1(6)
C
        DATA DEGREE/0.01745 32925/
C
C-------------------------------------------------------------
       WRITE(OUTPUT,2)
C
       WRITE(OUTPUT,3)
       DO 10 J=1,MSTA
       WRITE(OUTPUT,1) J,  THAH(J,21),TSAS(J,21),TRAR(J,21),TCCA(J,21)
10     CONTINUE
C
       WRITE(OUTPUT,4)
       DO 20 J=1,MSTA
       WRITE(OUTPUT,1) J,  THVH(J,21),TSVS(J,21),TRVR(J,21),TCCV(J,21)
20     CONTINUE
C
       WRITE(OUTPUT,5)
       DO 30 J=1,MSTA
       WRITE(OUTPUT,1) J,THACX(J,21),TSACX(J,21),THVCX(J,21),
     * TSVCX(J,21)
30     CONTINUE
C
       DO 40 K=1,NWAVES
       WRITE(OUTPUT,6) K
       WRITE(OUTPUT,3)
       DO 110 J=1,MSTA
       WRITE(OUTPUT,1) J,  THAH(J,K),TSAS(J,K),TRAR(J,K),TCCA(J,K)
110    CONTINUE
C
       WRITE(OUTPUT,4)
       DO 120 J=1,MSTA
       WRITE(OUTPUT,1) J,  THVH(J,K),TSVS(J,K),TRVR(J,K),TCCV(J,K)
```

```
120     CONTINUE
C
        WRITE(OUTPUT,5)
        DO 130 J=1,MSTA
        WRITE(OUTPUT,1) J,  THACX(J,K),TSACX(J,K),THVCX(J,K),TSVCX(J,K)
130     CONTINUE
40      CONTINUE
 1      FORMAT(1X,I2,5X,4E15.5)
 2      FORMAT(/'*****TEMPORARY HYDRODYNAMIC COEFFICIENTS****'//)
 3      FORMAT(1X,'STA',4X,'HAH',12X,'SAS',12X,'RAR',12X,'CCA'/)
 4      FORMAT(1X,'STA',4X,'HVH',12X,'SVS',12X,'RVR',12X,'CCV'/)
 5      FORMAT(1X,'STA',4X,'HACX',11X,'SACX',11X,'HVCX',11X,'SVCX',/)
 6      FORMAT(/1X,'WAVE NO.  - ',I2/)
        RETURN
        END
```

```
      SUBROUTINE FOLIO2
      COMMON/IO/INPUT,OUTPUT,BIF,OFF,COF
      INTEGER OUTPUT,BIF,OFF,COF
      COMMON /LHS   /   Y(13)
      COMMON /TIME  /   TIME, RAMP
      COMMON /TIME  /   TO, TSTART, TSTOP, TOUTPT, TSTEP, ERR
      COMMON /      /   NWAVES, WVSUM,
     1                  WVAMP(20), WVFRE(20), WVDIR(20), WVPHA(20),
     2                  WN(20), WNX(20), WNZ(20),
     3                  CSK(20), CCK(20), CXK(20),
     4                  CYK(20), CZK(20), XW(20)
      COMMON/SUMMARY/ NT,YYBAR(10),YYRMS(10),YYMAX(10),YYMIN(10)
      COMMON/LOADS/ SIN2(25),SIN3(25),SIN4(25),SIN5(25),SIN6(25),
     1              V2(25),V3(25),V4(25),V5(25),V6(25),
     2              RED(6,25)
      COMMON/DISPLAY/TARE(10)
      REAL YS(10)
      EQUIVALENCE (ETA,YS(1))
      DATA  RAD   /0.01745 32925 19943/
      ETA=0.0
      IF (NWAVES.EQ.0) GO TO 120
C *** Wave amplitude @ c.g.
      DO 110 K=1,NWAVES
         ARG=Y(1)*WNX(K)-Y(3)*WNZ(K)-TIME*WVFRE(K)+WVPHA(K)
         ETA=COS(ARG)*WVAMP(K)+ETA
  110 CONTINUE
C *** Position vector
  120 DO 130 I=1,3
         YS(I+1)=Y(I)
         YS(I+4)=Y(I+3)/RAD
  130 CONTINUE
C *** Display relative heave -- subtract c.g. height
C     above baseline to bring mean value to zero.
      YS(3)=YS(3)-TARE(2)
C *** Speed and sway rate
      CPHI=COS(Y(5))
      SPHI=SIN(Y(5))
      YS(8)=0.                          !NO RUDDER
      YS(8)=V6(14)      !BENDING MOMENT AT STATION J=14
      YS(9)=Y(7)*CPHI - Y(9)*SPHI
      YS(10)=Y(9)*CPHI+ Y(7)*SPHI
C *** Load mean and rms information
      TINIT=.75*TSTOP
      IF(TIME.LE.TINIT) GO TO 199
      NT=NT+1
      IF(NT.NE.1) GO TO 198
      DO 184 I=1,10
      YYMAX(I)=0.0
      YYMIN(I)=0.0
      YYBAR(I)=0.
      YYRMS(I)=0.
 184  CONTINUE
C
 198  CONTINUE
C

      DO 185 I=1,10
      IF(YS(I).GT.YYMAX(I)) YYMAX(I)=YS(I)
      IF(YS(I).LT.YYMIN(I)) YYMIN(I)=YS(I)
C
      YYBAR(I)=YYBAR(I)+YS(I)
      YYRMS(I)=YYRMS(I)+YS(I)**2



 185  CONTINUE
C
 199  CONTINUE
C *** Write to history file
      WRITE(10) YS(1),YS(3),YS(7),YS(5),YS(6),YS(4)
C *** Print YS vector
      WRITE (OUTPUT,190) TIME,(YS(I),I=1,9)
C
      RETURN
C
 190  FORMAT(5(1X,F11.4),3F12.4,2X,E15.5,2X,F12.4)
      END
```

```
      SUBROUTINE FORTIS (TT, YY)
C        TT     -- TIME, THE INDEPENDENT VARIABLE
C
C        YY     -- DEPENDENT VARIABLE ARRAY..
C                 YY(1)  = X-COORDINATE OF MASS CENTER
C                 YY(2)  = Y-COORDINATE OF MASS CENTER
C                 YY(3)  = Z-COORDINATE OF MASS CENTER
C                 YY(4)  = THETA ROTATION (ROLL ANGLE)
C                 YY(5)  = PHI   ROTATION (YAW ANGLE)
C                 YY(6)  = PSI   ROTATION (PITCH ANGLE)
C                 YY(7)  = X-COMPONENT OF LINEAR VELOCITY
C                 YY(8)  = Y-COMPONENT OF LINEAR VELOCITY
C                 YY(9)  = Z-COMPONENT OF LINEAR VELOCITY
C                 YY(10) = ANGULAR VELOCITY ABOUT X-AXIS (ROLL RATE)
C                 YY(11) = ANGULAR VELOCITY ABOUT Y-AXIS (YAW RATE)
C                 YY(12) = ANGULAR VELOCITY ABOUT Z-AXIS (PITCH RATE)
C
      REAL YY(12)
      COMMON /C      /   C(3,3)
      COMMON /DAMP   /   DAMPL(6), DAMPQ(6)
      COMMON /FORCE  /   XF, YF, ZF, XM, YM, ZM
      REAL  FORCE(6)
      EQUIVALENCE (FORCE(1),XF)
      COMMON /MASS   /   RHO, G, GAMMA
      COMMON /MASS   /   DISPL, SMASS, XCG, YCG, ZCG,
     1                   AMX, AMY, AMZ, RADII(6)
      COMMON /MASS   /   PMI(3,3)
      COMMON /STATS  /   NRHS(4), DELTA, HSUM, HMIN, HMAX, TTO, TT1
      COMMON /TIME   /   TIME, RAMP
      COMMON /TIME   /   TO, TSTART, TSTOP, TOUTPT, TSTEP, ERR
      COMMON /TRIG   /   CTHETA, STHETA, CPHI, SPHI, CPSI, SPSI
      COMMON /V      /   VX, VY, VZ
C *** Load rotational transform matrices - transform velocity
C     into ship coordinates
      CALL ROTATE (YY)
C *** Initialize force vector
      DO 110 I=1,6
         FORCE(I)=0.0
  110 CONTINUE
C
C *** Ensure constant speed for version SSX
      VX=SPEED
C *** Compute hydrodynamic and inertial forces on ship
      CALL AQUA2D (TT,YY)
C *** Add damping forces/moments in ship coordinate system
      XF=XF-(DAMPL(1)+ABS(VX)*DAMPQ(1))*VX
      YF=YF-(DAMPL(2)+ABS(VY)*DAMPQ(2))*VY
      ZF=ZF-(DAMPL(3)+ABS(VZ)*DAMPQ(3))*VZ
      XM=XM-(DAMPL(4)+ABS(YY(10))*DAMPQ(4))*YY(10)
      YM=YM-(DAMPL(5)+ABS(YY(11))*DAMPQ(5))*YY(11)
      ZM=ZM-(DAMPL(6)+ABS(YY(12))*DAMPQ(6))*YY(12)
C *** Fixed added mass coeficients - not normally used except
C     perhaps for surge.
      XF=XF/AMX
      YF=YF/AMY
      ZF=ZF/AMZ
C
C *** Ramp function applied to force
      IF (TT.GT.TSTART) GO TO 130
      IF (RAMP.EQ.0.0) GO TO 130
      R=(TT-TO)/RAMP


      DO 120 I=1,6
         FORCE(I)=FORCE(I)*R
  120 CONTINUE
  130 CONTINUE
C
      RETURN
      END
```

```
          SUBROUTINE SPECTRA(PFRQ,TRWH13,OMGBEG,OMGEND,WAMP,PER)
          DIMENSION WAMP(10),PER(10),OMEGA(10)
          DIMENSION OMG2(101),SPEC(101)
          IC1=-8
          IC2=0
          IC3=-7
          IC4=-1
          IC5=-9
          IC6=1
          IW=0
          OMG2(1)=OMGBEG
          OMG2(101)=OMGEND
1         DELFRQ=(OMG2(101)-OMG2(1))/101.
          DO 5 I=2,101
          J=I-1
5         OMG2(I)=OMG2(1)+J*DELFRQ
          DO 6 I=1,101
6         OMG2(I)=OMG2(I)/(2.*3.14159)
7         DO 10 I=1,101
10        SPEC(I)=5.*(.25*TRWH13)**2*PFRQ*(OMG2(I)*PFRQ)**(-5)*EXP(
     1              (-5./4.)*(OMG2(I)*PFRQ)**(-4))
          ODD=0.
          EVEN=0.
          DO 20 I=2,50,2
20        EVEN=EVEN+SPEC(I)
          DO 30 I=3,99,2
30        ODD=ODD+SPEC(I)
          TAREA=(OMG2(2)-OMG2(1))/3.*(SPEC(1)+4.*EVEN+2.*ODD+SPEC(101))
40        IC1=IC1+10
          IC2=IC2+10
          IC3=IC3+10
          IC4=IC4+10
          IC5=IC5+10
          IC6=IC6+10
          ODD=0.
          EVEN=0.
          IW=IW+1
          DO 50 I=IC1,IC2,2
50        EVEN=EVEN+SPEC(I)
          DO 60 I=IC3,IC4,2
60        ODD=ODD+SPEC(I)
          WAMP(IW)=(OMG2(2)-OMG2(1))/3.*(SPEC(IC5)+4.*EVEN+2.*
     1              ODD+SPEC(IC6))
          WAMP(IW)=1.414*SQRT(WAMP(IW))
          PER(IW)=(OMG2(IC6)+OMG2(IC5))/2.
          OMEGA(IW)=2.*3.14159*PER(IW)
          PER(IW)=1./PER(IW)
          IF(IW.LT.10) GO TO 40
          PRINT 500,TRWH13,PFRQ,OMEGA,PER,WAMP
          RMS=4.*SQRT(TAREA)
          PRINT 1010,TAREA,RMS
500      FORMAT(15X,'H1/3=',F10.2,2X,'PEAK FREQ=',F10.4,//,2X,
     1            'OMG',3X,10F6.3,/,2X,'PER',3X,10F6.2,/,2X,'WAMP',2X,
     2            10F6.2)
1010      FORMAT(2X,'TOTAL AREA=',F10.2,5X,'RMS=',F10.2)
          RETURN
          END
          SUBROUTINE LINEAR(X,Y,TRWPER,Y2)
          DIMENSION X(50),Y(50),Y2(10)
          DIMENSION TRWPER(10)
          J=0
```

```
140        RA=SQRT((X-XF)**2+(Y-YF)**2+(Z-ZF)**2)
           FRA=ARM/RA
1380       RB=SQRT((X-XF)**2+(Y-YF)**2+(Z+ZF)**2)
           FRA=FRA-ARM/RB
           FRX=-AN(MJ,1)*FRA
           FRY=-AN(MJ,2)*FRA
           FRZ=-AN(MJ,3)*FRA
           P1=P1+FRX
           P2=P2+FRY
           P3=P3+FRZ
           P4=P4+YF*FRZ-ZF*FRY
           P5=P5+ZF*FRX-XF*FRZ
           P6=P6+XF*FRY-YF*FRX
           PBB(NJ,MJ)=PBB(NJ,MJ)+FRA*ARN
138        CONTINUE
           P(NJ,1)=P(NJ,1)+P1*ARN
           P(NJ,2)=P(NJ,2)+P2*ARN
           P(NJ,3)=P(NJ,3)+P3*ARN
           P(NJ,4)=P(NJ,4)+P4*ARN
           P(NJ,5)=P(NJ,5)+P5*ARN
           P(NJ,6)=P(NJ,6)+P6*ARN
128        CONTINUE
127        CONTINUE
           DO 554 KQ=1,6
554        WRITE(3) (P(NJ,KQ),NJ=1,NPAN)
           RETURN
           END
```

```
               TOPI=2.*3.14159
               DO 20 II=1,10
               IF(II.GT.1) GO TO 69
  50           J=J+1
 69            IF(TOPI/TRWPER(II)-X(J))112,111,110
 110           GOTO 50
 111           Y2(II)=Y(J)
               GOTO 20
 112           Y2(II)=Y(J-1)+(Y(J)-Y(J-1))/(X(J)-X(J-1))*
          1          (TOPI/TRWPER(II)-X(J-1))
 20            CONTINUE
                RETURN
                END
```

```
       SUBROUTINE TBAR(TT,YY)
CCC
      EXTERNAL ETAF
CCC
       COMMON/MASS/RHO,G,GAMMA
       COMMON/MASS/DISPL,SMASS,XCG,YCG,ZCG,
     *              AMX,AMY,AMZ,RADII(6)
      COMMON /             /    NWAVES,  WVSUM,
     1                      WVAMP(20),  WVFRE(20),  WVDIR(20),  WVPHA(20),
     2                          WN(20),  WNX(20),  WNZ(20),
     3                          CSK(20),  CCK(20),  CXK(20),
     4                          CYK(20),  CZK(20),  XW(20)
      COMMON /          /       MSTA,  NPROF,  NFWD,  NAFT,  LPTS(25),
     1                          XOFF(25),  YOFF(25,25),  ZOFF(25,25),
     2                          XPROF(51),  YPROF(51)
       COMMON/ETA/YC,XK(20)
       COMMON/CT/CT(3,3)
       COMMON/TEMPSTA/DXFWD,DXAFT,TSTA(25)
       COMMON/DISPLAY/TARE(6)
       REAL YY(1)
C
        DATA DEGREE/0.01745 32925/
C       SECTION 1.0    Compute average draft at each station
C                      TSTA(J).
C
C

        DO 2 K=1,NWAVES
        CXK(K)=CT(1,1)*WNX(K)-CT(1,3)*WN(K)
        XW(K)=(TT*WVFRE(K)-WVPHA(K)+YY(3)*WNZ(K)-YY(1)*WNX(K))/CXK(K)
 2    CONTINUE
C
        DO 5 J=1,MSTA
        XXSTA=XOFF(J)
        IF(NWAVES.EQ.0) GO TO 11
        DO 10 K=1,NWAVES
        XK(K)=(XXSTA-XW(K))*CXK(K)
 10     CONTINUE
 11     YC=CT(1,2)*XXSTA+YY(2)
        YKEEL=YCG-TARE(2)-YC
        ETA1=ETAF(0. )
        TSTA(J)=ETA1-YKEEL
 5      CONTINUE
C
C
C
C
CCC
       RETURN
       END
```

```
      SUBROUTINE MOCHA
CCC
CCC
CCC   Purpose: For JOBCO=2,3 or 4 ... this subroutine
CCC   will load the COMMON/COEFFX/ arrays for use in
CCC   subroutine FROUDE.
CCC
      COMMON/MASS/RHO,G,GAMMA
      COMMON/MASS/DISPL,SMASS,XCG,YCG,ZCG,
     *             AMX,AMY,AMZ,RADII(6)
      COMMON/LHS/Y(13)
C5   WAVE PROPERTIES..
     COMMON /      /   NWAVES,  WVSUM,
    1                  WVAMP(20),  WVFRE(20),  WVDIR(20),  WVPHA(20),
    2                  WN(20),  WNX(20),  WNZ(20),
    3                  CSK(20),  CCK(20),  CXK(20),
    4                  CYK(20),  CZK(20),  XW(20)
C5
C5
C5   TABLE OF OFFSETS..
     COMMON /      /   MSTA,  NPROF,  NFWD,  NAFT,  LPTS(25),
    1                  XOFF(25),  YOFF(25,25),  ZOFF(25,25),
    2                  XPROF(51),  YPROF(51)
      COMMON/RESIST/SPEED
      COMMON/SIGMA /  NK,SIGMA(24),SIGMAO,  ERRO,  OM(12)
C
C
      COMMON/COEFF4/COEFF4(6,12,8,25),AREAN(25,6)
      COMMON/COEFFX/THAH(25,21),TSAS(25,21),TRAR(25,21),TCCA(25,21),
     *             THVH(25,21),TSVS(25,21),TRVR(25,21),TCCV(25,21),
     *             THACX(25,21),THVCX(25,21),TSACX(25,21),TSVCX(25,21),
     *             TRACX(25,21),TCACX(25,21),THVCY(25,21),
     *             ITP(25),TXP(25),IFP(21),FXP(21)
      COMMON/OPTION/JOBCO,JOBFO(10)
      COMMON/ETA/YC,XK(20)
      COMMON/CT/CT(3,3)
      COMMON/SXOMEG/OMEGAX(3)
      COMMON/DRFT1/ DRAFT1(6)
      COMMON/TEMPSTA/DXFWD,DXAFT,TSTA(25)
C
       DATA DEGREE/0.01745 32925/
C      SECTION 1.0    Compute average draft at each station
C                     TSTA(J).
C
C      No longer required here. Now done in FROUDE right
C      before the call to MOCHA
C
C      SECTION 2.0    Computations for Coefficients used
C                     in Wave Exciting/Diffraction Force
C                     Calculations
C
C
       DO 10 J=1,MSTA
       DO 10 K=1,NWAVES
C
       IF (JOBCO.EQ.2.OR.JOBCO.EQ.4) THEN
           IF=IFP(K)
           FX=FXP(K)
           CALL TXIT(TSTA(J),IT,TX)
       ELSE IF (JOBCO.EQ.3) THEN
           IT=ITP(J)
```

```
                 TX=TXP(J)
                 IF=IFP(K)
                 FX=FXP(K)
             END IF
C
             THAH(J,K)=COX(IT,TX,IF,FX,1,J)
             TSAS(J,K)=COX(IT,TX,IF,FX,2,J)
             TRAR(J,K)=COX(IT,TX,IF,FX,3,J)
             TCCA(J,K)=COX(IT,TX,IF,FX,4,J)
C
             THVH(J,K)=COX(IT,TX,IF,FX,5,J)
             TSVS(J,K)=COX(IT,TX,IF,FX,6,J)
             TRVR(J,K)=COX(IT,TX,IF,FX,7,J)
             TCCV(J,K)=COX(IT,TX,IF,FX,8,J)
C
C
C            Spatial derivatives in x-direction
C
             THACX(J,K)=COXDX(IT,TX,IF,FX,1,J)
             TSACX(J,K)=COXDX(IT,TX,IF,FX,2,J)
             TRACX(J,K)=COXDX(IT,TX,IF,FX,3,J)
             TCACX(J,K)=COXDX(IT,TX,IF,FX,4,J)
C
 10          CONTINUE
C
CCC
CCC         SECTION 3.0   - This section computes coefficients
CCC                         for ship motion related force
CCC                         computations, using OMEGAX( ) as
CCC                         the frequency at which computations
CCC                         are performed.  OMEGAX(1) - HEAVE,
CCC                         OMEGAX(2) - SWAY, OMEGAX(3) - ROLL.
CCC                         These are computed in Subroutine
CCC                         GETFREQ.
CCC
CCC
            DO 20 J=1,MSTA
C
            IF(JOBCO.EQ.2) THEN
               IF=IFP(21)
               FX=FXP(21)
               IFH=IF
               IFS=IF
               IFR=IF
               FXH=FX
               FXS=FX
               FXR=FX
               CALL TXIT(TSTA(J),IT,TX)
            ELSE IF(JOBCO.EQ.3) THEN
               IT=ITP(J)
               TX=TXP(J)
               CALL FXIF(OMEGAX(1),IFH,FXH)
               CALL FXIF(OMEGAX(2),IFS,FXS)
               CALL FXIF(OMEGAX(3),IFR,FXR)
            ELSE IF(JOBCO.EQ.4) THEN
               CALL TXIT(TSTA(J),IT,TX)
               CALL FXIF(OMEGAX(1),IFH,FXH)
               CALL FXIF(OMEGAX(2),IFS,FXS)
               CALL FXIF(OMEGAX(3),IFR,FXR)
C
            END IF
```

```
           THAH(J,21)=COX(IT,TX,IFH,FXH,1,J)
           TSAS(J,21)=COX(IT,TX,IFS,FXS,2,J)
           TRAR(J,21)=COX(IT,TX,IFR,FXR,3,J)
           TCCA(J,21)=COX(IT,TX,IFS,FXS,4,J)
C
           THVH(J,21)=COX(IT,TX,IFH,FXH,5,J)
           TSVS(J,21)=COX(IT,TX,IFS,FXS,6,J)
           TRVR(J,21)=COX(IT,TX,IFR,FXR,7,J)
           TCCV(J,21)=COX(IT,TX,IFS,FXS,8,J)
C
C
C         Spatial Derivatives in x-directin
C
           THACX(J,21)=COXDX(IT,TX,IFH,FXH,1,J)
           TSACX(J,21)=COXDX(IT,TX,IFS,FXS,2,J)
           THVCX(J,21)=COXDX(IT,TX,IFH,FXH,5,J)
           TSVCX(J,21)=COXDX(IT,TX,IFS,FXS,6,J)
C
 20       CONTINUE
C
C
C         [Note: Vertical derivative of added mass computed
C          in SUBROUTINE FROUDE or COFFEE.]
C
          RETURN
          END
```

```
      SUBROUTINE AQUA2D (TT,YY)
C
C-----------------------------------------------------------------
C
C     CALCULATES SHIP-MOTION INDUCED FORCES, WAVE INDUCED FORCES,
C     AND OTHER HYDRODYNAMIC FORCES ON THE SHIP.
C
C     SECTION              DESCRIPTION
C     -------              -----------
C       1.0                INITIALIZATION
C       2.0                FIND WET OFFSET POINTS/ENDS OF HULL
C       3.0                INITIALIZATION
C       4.0                CALCULATE VARIOUS 2-D INTEGRALS/GEOMETRIC INFO. FOR
C       5.0                CALCULATE FORCES THAT ARE SHIP-MOTION RELATED
C       6.0                CALCULATE DORCES THAT ARE WAVE-INDUCED
C       7.0                CALCULATE OTHER FORCES
C       8.0                INTEGRATE FORCES OVER LENGTH
C       9.0                ADD UP FORCES AND LOAD FORCE ARRAY
C      10.0                LOAD MASS/INERTIA MATRIX A(I,J)
C-----------------------------------------------------------------
C
C     NOTE: The hydrodynamic coefficients were divided by sectional area in
c           COEFS part of this program.  They were also divided by RHO in th
C           subroutine. Thus, quantities are multiplied by DVOL
C           when they would normally be multiplied by DX, and there
C           are several places where a coefficient is divided by  G operated
C           to make sure that all the units are consistent from force to
C           force.  At any point in the force calculations, one must
C           multiply by GAMMA to obtain force in lbs.
C------------------------------------------------------------------
C
C1
      REAL YY(1)
      COMMON /A      /    A(6,6)
      COMMON /AM     /    YFYA, ZFZA, XMZA, XMXR, XMYR,
     1                    YMZA, YMYR, ZMYA, ZMZR
      COMMON/RESIST/SPEED
      REAL  VT(6), AM(9)
      EQUIVALENCE (AM(1), YFYA)
      COMMON /C      /    C(3,3)
      COMMON /CT     /    CT(3,3)
      COMMON /ETA    /    YC, XK(20)
      COMMON /FORCE /     XF, YF, ZF, XM, YM, ZM
      REAL  FORCE(6)
      EQUIVALENCE (FORCE(1),XF)
      COMMON /MASS   /    RHO, G, GAMMA
      COMMON /MASS   /    DISPL, SMASS, XCG, YCG, ZCG,
     1                    AMX, AMY, AMZ, RADII(6)
      COMMON /MASS   /    PMI(3,3)
      COMMON /TRIG   /    CTHETA, STHETA, CPHI, SPHI, CPSI, SPSI
      COMMON /V      /    VX, VY, VZ
      COMMON /       /    NWAVES, WVSUM,
     1                    WVAMP(20), WVFRE(20), WVDIR(20), WVPHA(20),
     2                    WN(20), WNX(20), WNZ(20),
     3                    CSK(20), CCK(20), CXK(20),
     4                    CYK(20), CZK(20), XW(20)
      COMMON /       /    MSTA, NPROF, NFWD, NAFT, LPTS(25),
     1                    XOFF(25), YOFF(25,25), ZOFF(25,25),
     2                    XPROF(51), YPROF(51)
      COMMON /       /    INPTS(25), IPOINT(25), YWET(25,25), ETA(25,25),
     1                    ZLAST(25,25), ZWET(25,25), ZNEXT(25,25)
```

```
      COMMON /      /   ZCO(20), ZSO(20), ZCY(20), ZSY(20),
     1                  ZCZ(20), ZSZ(20)
      COMMON /      /   DZK(20)
      COMMON /      /   JSAVE(20), COSJ(20), SINJ(20),
     $                  COSJJ(20), SINJJ(20),
     1                  DWC(20), DWS(20), XCI(20), XSI(20)
      COMMON /AREA  /   AREA, YMOM, ZMOM
      EQUIVALENCE (FWD,XPROF(51)), (AFT,YPROF(51))
      COMMON/SXFOR/ FBUOY(6,25),FDAMP(6,25),FADMA(6,25),FWVEX(6,25),
     *              UDAMP(6,25),UADMA(6,25),UWVEX(6,25),FFLAR(6,25)
      COMMON/SXGEOM/AREAS(25),DXS(25),DX1S(25),DX2S(25),
     *              YMOMS(25),ZMOMS(25),ISTA,LSTA
      COMMON/SXOMEG/OMEGAX(3)
      COMMON/TEMPSTA/DXFWD,DXAFT,TSTA(25)
      COMMON/OPTION/JOBCO,JOBFD(10)
      DIMENSION COLD(25),CNEW(25),TOLD(25),TNEW(25)
      DIMENSION TTOLD(25),TTNEW(25)
      REAL ZEROF(1200)
      EQUIVALENCE (ZEROF(1),FBUOY(1,1))
      COMMON/FCOMP/XFB,YFB,ZFB,XMB,YMB,ZMB,
     1             XFD,YFD,ZFD,XMD,YMD,ZMD,
     2             XFX,YFX,ZFX,XMX,YMX,ZMX,
     3             XFN,YFN,ZFN,XMN,YMN,ZMN,
     4             XFU,YFU,ZFU,XMU,YMU,ZMU,
     5             YFF,ZMF
      REAL  ZEROC(32)
      EQUIVALENCE (ZEROC(1),XFB)
       COMMON/COEFFX/THAH(25,21),TSAS(25,21),TRAR(25,21),TCCA(25,21),
     *              THVH(25,21),TSVS(25,21),TRVR(25,21),TCCV(25,21),
     *              THACX(25,21),THVCX(25,21),TSACX(25,21),TSVCX(25,21),
     *              TRACX(25,21),TCACX(25,21),THVCY(25,21),
     *              ITP(25),TXP(25),IFP(21),FXP(21)
C *** Initialization
      IF (NWAVES.EQ.0) GO TO 120
C------------------------------------------------------------
C
C     SECTION 1.0 - INITIALIATION
C
C     Project wave numbers into calculation coordinate system
C     (yaw and pitch but not roll).  Compute x-coord of wave
C     crest.
C
C-------------------------------------------------------------
      DO 110 K=1,NWAVES
         CSK(K)=CT(1,2)*WN(K)
         CCK(K)=CT(2,2)*WN(K)
         CXK(K)=CT(1,1)*WNX(K)-CT(1,3)*WNZ(K)
         CYK(K)=CT(2,1)*WNX(K)-CT(2,3)*WNZ(K)
         CZK(K)=CT(3,1)*WNX(K)-CT(3,3)*WNZ(K)
         DZK(K)=2.0/CZK(K)
         XW(K)=(TT*WVFRE(K)-WVPHA(K)+YY(3)*WNZ(K)-YY(1)*WNX(K))/CXK(K)
         JSAVE(K)=0
  110 CONTINUE
  120 CONTINUE
C4
C-------------------------------------------------------------
C
C     1.1 Zero the acceleration force coefficients AM(I) and
C         zero the sectional force arrays in common SXFOR.
C-------------------------------------------------------------
C4
```

```
      DO 130 I=1,9
         AM(I)=0.0
  130 CONTINUE
C*
      DO 131 I=1,1200
      ZEROF(I)=0.0    ! Equivalenced with COMMON/SXFOR/
131   CONTINUE
C*
C
      DO 800 I = 1,32
      ZEROC(I)=0.     !ZEROC is EQUIVALENCEd with COMMON/FCOMP/
800   CONTINUE
C
C*
C-------------------------------------------------------------
C
C     1.2 Load COMMON/COEFFX/ if JOBCO=2,3 or 4
C
C-------------------------------------------------------------
C
      CALL TBAR(TT,YY)        !Get average draft at every sta.
      TYPE 1000, (TSTA(J),J=1,MSTA)
1000  FORMAT(1X,25F5.1)
C     DO 1000 J=1,MSTA
C     TSTA(J)=32.8
C1000  CONTINUE
C
      IF(JOBCO.NE.1.AND.TT.GT.0.) CALL MOCHA
C
      ISTA=0
C--------------------------------------------------------------
C
C     SECTION 2.0 - FIND WET OFFSET POINTS AND HULL ENDS
C
C
C--------------------------------------------------------------
C
C3
C3    FIND WET OFFSET POINTS.
C3
C3       ETA(L)   = Y-COORDINATE OF WAVE SURFACE,
C3       YWET(L)  = Y-COORDINATE OF WET OFFSET POINT.
C3       ZWET(L)  = Z-COORDINATE OF WET OFFSET POINT.
C3       ZLAST(L) = Z-COORDINATE OF EITHER THE PREVIOUS WET OFFSET
C3                  POINT OR THE Z-COORDINATE OF THE WATER SURFACE,
C3       ZNEXT(L) = Z-COORDINATE OF EITHER THE NEXT WET OFFSET POINT
C3                  OR THE Z-COORDINATE OF THE WATER SURFACE.
C3
C4
C4    KEY INDICATES THE STATUS OF THE SEARCH:
C4
C4       KEY = 1  --  THE FIRST OFFSET POINT IS BEING TESTED,
C4       KEY = 2  --  THE LAST OFFSET POINT WAS WET,
C4       KEY = 3  --  THE LAST OFFSET POINT WAS DRY (ETA COMPUTED),
C4       KEY = 4  --  THE LAST OFFSET POINT WAS DRY AND ETAMAX WAS
C4                    USED (ETA WAS NOT COMPUTED).
C4
C
C--------------------------------------------------------------
C
         DO 260 J=1,MSTA
         KEY=1
```

```
          I=1
          L=0
          LWET=0
          N=LPTS(J)
          XXSTA=XOFF(J)
          IF (NWAVES.EQ.0) GO TO 150
          DO 140 K=1,NWAVES
             XK(K)=(XXSTA-XW(K))*CXK(K)
  140     CONTINUE
  150     YC=CT(1,2)*XXSTA+YY(2)
          ETAMAX=(WVSUM-YC)/CT(2,2)
  160     Y1=YOFF(I,J)*CTHETA-ZOFF(I,J)*STHETA
          Z1=ZOFF(I,J)*CTHETA+YOFF(I,J)*STHETA
          IF (KEY.LT.3) GO TO 170
C5
C5        THE PREVIOUS OFFSET POINT WAS DRY.  AN ESTIMATE OF ETA WILL BE
C5        USED TO SEE IF THIS POINT CAN BE WET.
C5
          IF (Y1.LT.ETAMAX) GO TO 170
          KEY=4
          GO TO 230
C4
C4        COMPUTE THE Y-COORDINATE OF THE WATER SURFACE..
C4
  170     ETA1=ETAF(Z1)
          IF (KEY.GT.1) GO TO 180
C5
C5        SAVE INITIAL CALCULATIONS FOR USE WITH FINAL OFFSET POINT.
C5        THE INITIAL AND FINAL OFFSETS REFER TO THE SAME POINT.
C5
          EFIRST=ETA1
          YFIRST=Y1
          YO=Y1
          ZFIRST=Z1
          ZO=Z1
C4
C4        TEST THE POSITION OF THE CURRENT OFFSET POINT RELATIVE TO THE
C4        WATER SURFACE.  THE   IF   TEST IS SATISFIED IF THE POINT IS
C4        BELOW THE WATER SURFACE.
C4
  180     IF (Y1.LT.ETA1) GO TO (212,210,200,190),KEY
C5
C5        THE CURRENT OFFSET POINT IS DRY.  IF KEY=2, THE PREVIOUS POINT
C5        WAS WET, AND THE Z-COORDINATE OF THE INTERSECTION OF THE HULL
C5        AND THE WATER SURFACE MUST BE FOUND BY INTERPOLATION.
C5
          IF (KEY.NE.2) GO TO 181
          ZNEXT(L,J)=(Z1-ZO)/(Y1-ETA1+EY)*EY+ZO
          LWET=L
  181     KEY=3
          GO TO 220
C5
C5        IF  KEY = 4,  THE ETA VALUE FOR THE LAST OFFSET POINT WAS NOT
C5        COMPUTED.  COMPUTE ETA..
C5
  190     EY=ETAF(ZO)-YO
C5
C5        THE CURRENT OFFSET POINT IS WET, AND THE PREVIOUS POINT IS DRY.
C5        INTERPOLATE TO FIND THE INTERSECTION OF THE HULL AND THE WATER
C5        SURFACE..
C5
```

```
  200     ZO=ZO+(Z1-ZO)/(Y1-ETA1+EY)*EY
          GO TO 211
C4
C4        A WET OFFSET POINT HAS BEEN LOCATED.  STORE THE REQURED
C4        COORDINATE VALUES..
C4
  210     ZNEXT(L,J)=Z1
  211     IF (I.EQ.N) GO TO 240
  212     L=L+1
          ETA(L,J)=ETA1
          YWET(L,J)=Y1
          ZWET(L,J)=Z1
          ZLAST(L,J)=ZO
          KEY=2
C5
C5        SAVE THE COMPUTED COORDINATE OF THE WATER SURFACE..
C5
  220     EY=ETA1-Y1
C5
C5        SAVE THE COORDINATES OF THE OFFSET POINT..
C5
  230     YO=Y1
          ZO=Z1
C4
C4        INCREMENT THE INDEX OF THE OFFSET POINT FOR THIS STATION.  IF
C4        THE INDEX DOES NOT REFER TO THE LAST POINT, REPEAT THE LOOP.
C4
          I=I+1
          IF (I.LT.N) GO TO 160
          IF (I.GT.N) GO TO 250
C5
C5        FOR THE LAST OFFSET POINT, THE VALUES OF ETA1, X1, AND Y1 ARE
C5        THE SAME AS FOR THE FIRST OFFSET POINT.  REPEAT THE LOOP USING
C5        THESE VALUES..
C5
          ETA1=EFIRST
          Y1=YFIRST
          Z1=ZFIRST
          GO TO 180
C4
C4        A COMPLETE SEARCH OVER ALL THE OFFSET POINTS FOR THE CURRENT
C4        STATION HAS OBTAINED   L   WET OFFSET POINTS.  RESET ZLAST(1)
C4        IF THE LAST (AND ALSO THE FIRST) OFFSET POINT IS WET..
C4
  240     ZLAST(1,J)=ZO
  250     INPTS(J)=L
          IF (L.EQ.0) GO TO 260
          IF (LWET.EQ.L) LWET=0
          IPOINT(J)=LWET+1
          LSTA=J
          IF (ISTA.EQ.0) ISTA=J
C3
  260 CONTINUE
      IF (ISTA.EQ.0) GO TO 520
C3
C3    FIND ENDS OF WETTED HULL..
C3
C3        XXF      = X-COORDINATE FORWARD
C3        XXA      = X-COORDINATE AFT
C3
      XXF=XOFF(ISTA)
```

```
      XXA=XOFF(LSTA)
      IF (NPROF.EQ.0) GO TO 410
C4
C4    SEARCH FOR ENDS USING THE PROFILE DATA..
C4
C4       XLAST    --  X-COORDINATE OF PREVIOUS PROFILE POINT
C4       XXSTA    --  X-COORDINATE OF PRESENT PROFILE POINT
C4       ETABAR   --  WAVE ELEVATION IN ABSOLUTE COORDINATES
C4       YO       --  SUBMERGENCE OF PREVIOUS PROFILE POINT (LAST.EQ.1)
C4       Y1       --  SUBMERGENCE OF PRESENT PROFILE POINT
C4       IEND=0   --  FORWARD END OF SHIP
C4       IEND=1   --  AFT END OF SHIP
C4       KEY=-1   --  SUBMERGENCE OF PRESENT PROFILE POINT UNKNOWN
C4       KEY=0    --  WAVE ELEVATION NOT COMPUTED
C4       KEY=1    --  WAVE ELEVATION COMPUTED
C4       LAST     --  KEY  VALUE FOR PREVIOUS PROFILE POINT
C4       LWET=0   --  PREVIOUS PROFILE POINT WAS DRY (LAST.GE.0)
C4       LWET=1   --  PREVIOUS PROFILE POINT WAS SUBMERGED (LAST.GE.0)
C4
      XLAST=XPROF(1)
      LAST=-1
      DO 400 I=2,NPROF
         KEY=-1
         IEND=0
         XXSTA=XPROF(I)
         IF (XXSTA.GT.XXF) GO TO 270
         IEND=1
         IF (XXSTA.LT.XXA) GO TO 270
         IF (XLAST.GE.XXA) GO TO 390
  270    KEY=0
         Y1=YPROF(I)*C(2,2)+XXSTA*C(1,2)-YY(2)
         IF (Y1.LT.-WVSUM) GO TO 310
         IF (Y1.GT.WVSUM) GO TO 280
         Y1=ETABAR(I)-Y1
         KEY=1
         IF (Y1.GE.0.0) GO TO 310
C5
C5       POINT IS DRY.
C5
  280    IF (IEND.NE.0) GO TO 300
         IF (LAST.LT.0) GO TO 290
         IF (LWET.EQ.0) GO TO 300
         IF (LAST.NE.0) GO TO 340
  290    YO=YPROF(I-1)*C(2,2)+XLAST*C(1,2)-YY(2)
         IF (YO.GT.WVSUM) GO TO 300
         YO=ETABAR(I-1)-YO
         IF (YO.GE.0.0) GO TO 340
  300    LWET=0
         GO TO 380
C5
C5       POINT IS WET.
C5
  310    IF (IEND.NE.0) GO TO 320
         XXF=XXSTA
         LWET=1
         GO TO 380
  320    IF (LAST.LT.0) GO TO 330
         IF (LWET.NE.0) GO TO 360
         IF (LAST.NE.0) GO TO 340
  330    YO=YPROF(I-1)*C(2,2)+XLAST*C(1,2)-YY(2)
         IF (YO.LE.-WVSUM) GO TO 360
```

```
         YO=ETABAR(I-1)-YO
         IF (YO.GE.0.0) GO TO 360
C5
C5       INTERPOLATE FOR WATERLINE.
C5
  340    IF (KEY.NE.0) GO TO 350
         Y1=ETABAR(I)-Y1
         KEY=1
  350    XXSTA=(XPROF(I-1)-XXSTA)/(Y1-YO)+XXSTA
         IF (IEND.NE.0) GO TO 360
         IF (XXSTA.GT.XXF) XXF=XXSTA
         GO TO 370
  360    IF (XXSTA.LT.XXA) XXA=XXSTA
  370    LWET=IEND
  380    YO=Y1
  390    LAST=KEY
         XLAST=XXSTA
  400 CONTINUE
  410 IF (NFWD.NE.1) GO TO 420
      IF (XXF.LT.FWD) XXF=FWD
  420 IF (NAFT.NE.1) GO TO 430
      IF (XXA.GT.AFT) XXA=AFT
C3
C-------------------------------------------------------------------
C
C     SECTION 3.0  ANOTHER INITIALIZATION
C
C     Resolve velocity vector into yawed and pitched coord's.
C     NOTE: Forces are calculated in local (ship) coordinate
C           system.
C
C-------------------------------------------------------------------
C
  430 VT(1)=VX
      VT(2)=VY*CTHETA-VZ*STHETA
      VT(3)=VZ*CTHETA+VY*STHETA
      VT(4)=YY(10)
      VT(5)=YY(11)*CTHETA-YY(12)*STHETA
      VT(6)=YY(12)*CTHETA+YY(11)*STHETA
C-------------------------------------------------------------
C
C      Loop over wet stations ISTA to LSTA.
C      Also check to see if ship is out of water (SMALL)
C
C-------------------------------------------------------------
      LSAVE=0
C
      SMALL=(XXF-XXA)*1.0E-06
      IF (SMALL.LE.0.0) GO TO 520
      XXSTA=XXF
      DO 510 J=ISTA,LSTA
C------------------------------------------------------------------
C
C     Section 4.1 - Get geometric station data, e.g. DX.
C
C------------------------------------------------------------------
         XXFWD=XXSTA
         XXSTA=XOFF(J)
         IF (INPTS(J).EQ.0) GO TO 510
         YC=YY(2)+CT(1,2)*XXSTA
         IF (J.NE.LSTA) GO TO 331
```

```
         XXAFT=XXA
         GO TO 332
  331    XXAFT=XOFF(J+1)
  332    CONTINUE
         DXFWD=XXFWD-XXSTA
         DXAFT=XXSTA-XXAFT
         DX=(DXFWD+DXAFT)/2.0
         XA=XXSTA-DXAFT/3.0
         XB=XXSTA+DXFWD/3.0
         DX1=(DXFWD*XB+DXAFT*XA)/2.0
         DX2=(DXFWD*XB**2+DXAFT*XA**2)/2.0+(DXFWD**3+DXAFT**3)/36.0
C--------------------------------------------------------------
C
C        Section 4.2 - Call KRYLOV by section (j) and get 2-D
C                      integrals for that station.
C
C
C--------------------------------------------------------------
         CALL KRYLOV (J)
C--------------------------------------------------------------
C
C       Section 4.3 - Compute the sectional volumes and moments
C                     about center of mass.
C
C--------------------------------------------------------------
C
C
         DVOL=AREA*DX
         ADX1=AREA*DX1
         ADX2=AREA*DX2
         YDX=YMOM*DX
         ZDX=ZMOM*DX
C
C
C--------------------------------------------------------------
C        Load COMMON/SXGEOM/
         AREAS(J)=AREA
         DXS(J)=DX
         DX1S(J)=DX1
         DX2S(J)=DX2
         YMOMS(J)=YMOM
         ZMOMS(J)=ZMOM
C--------------------------------------------------------------
C
C        SECTION 5.0 - SHIP MOTION RELATED FORCES
C
C        FBUOY(I,J) - Buoy force
C        FDAMP(I,J)   Damping force
C        FADMA(I,J) - Added Mass Force for purposes of analysis
C                     only since the added mass actually goes
C                     on the left hand side of the Equations.
C
C--------------------------------------------------------------
C
C        Section 5.1 - Buoyancy Force
C--------------------------------------------------------------
       FBUOY(1,J)=CT(1,2)*DVOL
       FBUOY(2,J)=CT(2,2)*DVOL
       FBUOY(3,J)=0.
C            ???? ^ ????
       FBUOY(4,J)=-CT(2,2)*ZDX
       FBUOY(5,J)=CT(1,2)*ZDX
```

```
       FBUOY(6,J)= CT(2,2)*ADX1-CT(1,2)*YDX
C
C-------------------------------------------------------------------
C
C      Section 5.2 - Damping force
C
C      Get damping coefficients ...
C              This section will be changed to reflect the
C              dependency on frequency of ship motion.
C
C        HVH=COEFF(1,2,J)/G
C        SVS=COEFF(2,2,J)/G
C        RVR=COEFF(3,2,J)/G
C        CCV=COEFF(4,2,J)/G
C
         HVH=THVH(J,21)/G
         SVS=TSVS(J,21)/G
         RVR=TRVR(J,21)/G
         CCV=TCCV(J,21)/G
C
         HAH=THAH(J,21)
         SAS=TSAS(J,21)
         RAR=TRAR(J,21)
         CCA=TCCA(J,21)
C
C
         VS=VT(5)*ADX1-VT(3)*DVOL
         VR=VT(4)*DVOL
         FDAMP(1,J)=0.
         FDAMP(2,J)=(VT(4)*ZDX-VT(2)*DVOL-VT(6)*ADX1)*HVH
         FDAMP(3,J)=VS*SVS-VR*CCV
         FDAMP(4,J)=VS*CCV-VR*RVR
         FDAMP(5,J)=(VT(3)*ADX1-VT(5)*ADX2)*SVS+VT(4)*ADX1*CCV
         FDAMP(6,J)=(VT(4)*ZMOM*DX1-VT(2)*ADX1-VT(6)*ADX2)*HVH
C
C
C-------------------------------------------------------------------
C
C        Section 5.3 - Speed terms
C        5.3.1 - Speed terms associated with damping
C
         UDAMP(1,J)=0.
C
         UDAMP(2,J)=(2.*SPEED*YY(6))*HVH*DVOL
         UDAMP(3,J)=-(2.*SPEED*YY(5))*SVS*DVOL
         UDAMP(4,J)=0.
         UDAMP(5,J)=(2.*SPEED*YY(5))*SVS*ADX1
         UDAMP(6,J)=(2.*SPEED*YY(6))*HVH*ADX1
C
C
C *** Extra speed terms from extended strip (X SCORES)
C        U2TEMP=(YY(2)*DVOL+YY(6)*ADX1+SPEED*DVOL*
C    *                (VT(6)/OMEGAX(1)**2))*(THVCX(J,21)/G)*
C    *                 SPEED
C        UDAMP(2,J)=UDAMP(2,J)+U2TEMP
C        UDAMP(6,J)=UDAMP(6,J)+U2TEMP*(ADX1/DVOL)
C
C        5.3.2  Speed terms associated with added mass
C
         UADMA(1,J)=0.
         UA21=(2.*SPEED*VT(6))*(HAH/G)*DVOL
```

```
         UA22=(VT(4)*ZDX-VT(2)*DVOL-VT(6)*ADX1+SPEED*YY(6)
     1          *DVOL)*SPEED*(THACX(J,21)/G)
         UADMA(2,J)=UA21-UA22
         UA31=(-2.*SPEED*VT(5))*(SAS/G)*DVOL
         UA32=(VS-SPEED*YY(5))*(TSACX(J,21)/G)*SPEED
         UADMA(3,J)=UA31+UA32
         UADMA(4,J)=-VR*SPEED*TRACX(J,21)/G
         UA51=(2.*SPEED*VT(5))*(SAS/G)*ADX1
         UA52=((SPEED*YY(5)*ADX1)+(VT(3)*ADX1-VT(5)*ADX2))
     1         *SPEED*TSACX(J,21)/G
         UADMA(5,J)=UA51+UA52
         UA61=(2.*SPEED*VT(6))*(HAH/G)*ADX1
         UA62=((SPEED*YY(6)*ADX1)+(VT(4)*ZMOM*DX1-VT(2)*
     1         ADX1-VT(6)*ADX2))*SPEED*THACX(J,21)/G
         UADMA(6,J)=UA61-UA62
C
C

         YFYA=HAH*DVOL+YFYA
         ZFZA=SAS*DVOL+ZFZA
         XMZA=CCA*DVOL+XMZA
         XMXR=RAR*DVOL+XMXR
         XMYR=XMYR-CCA*ADX1
         YMZA=YMZA-SAS*ADX1
         YMYR=SAS*ADX2+YMYR
         ZMYA=HAH*ADX1+ZMYA
         ZMZR=HAH*ADX2+ZMZR
C
C------------------------------------------------------------
C
C
C--------------------------------------------------------------
C
C     Section 5.4 - Force due to weight of ship
C--------------------------------------------------------------
       XFW=-CT(1,2)*DISPL
       YFW=-CT(2,2)*DISPL
C
C--------------------------------------------------------------
C
C
C     Section 7.2 - So called "flare" force
C
C

      IF(TT.EQ.TTOLD(J)) GO TO 721
      TNEW(J)=TSTA(J)
      TAVG=TSTA(J)
      CALL TXIT(TAVG,IT,TX)
      CNEW(J)=COX(IT,TX,11,1,1,J)*ARX(IT,TX,J)/G
      TTNEW(J)=TT
      DC=CNEW(J)-COLD(J)
      DT=TNEW(J)-TOLD(J)
      DTT=TTNEW(J)-TTOLD(J)
      TOLD(J)=TNEW(J)
      TTOLD(J)=TTNEW(J)
      COLD(J)=CNEW(J)
C
      IF (DT.GT.O..AND.DTT.NE.O.) THEN !Relative Velocity indicates immersio
      FFLAR(2,J)=(DT/DTT)**2*(DC/DT)*DX
      FFLAR(6,J)=(DT/DTT)**2*(DC/DT)*DX1
      ELSE ! Relative velocity indicates emergence
      FFLAR(2,J)=O.
      FFLAR(6,J)=O.
```

```
      END IF
C
 721  CONTINUE                     !Skip to here when TT=TTOLD
C
C------------------------------------------------------------
C        If there are no waves, skip over the wave loop -
C        [Do 500] and go to 510 (the other side of the section
C        loop).
         IF (NWAVES.EQ.0) GO TO 510
C--------------------------------------------------------------
C           SECTION 6.0 - WAVE-INDUCED FORCES
C--------------------------------------------------------------
C
         IF (LSAVE.NE.0) GO TO 450
         IF (DXFWD.LT.SMALL) GO TO 441
         DDXFWD=1.0/DXFWD
         GO TO 460
  441    DDXFWD=0.0
         GO TO 460
  450    DDXFWD=DDXAFT
  460    IF (DXAFT.LT.SMALL) GO TO 461
         DDXAFT=1.0/DXAFT
         GO TO 470
  461    DDXAFT=0.0
C
  470    LSAVE=LSTA-J
         IF (LSAVE.NE.0) LSAVE=INPTS(J+1)
C
C
C-----------------------------------------------------------
C        Loop over waves (frequency index K)
C-----------------------------------------------------------
C
         DO 500 K=1,NWAVES
C--------------------------------------------------------------
C
C           Section 6.1 - Compute integration multipliers...
C                         DXCOS,  DXSIN,DX1COS,DX1SIN.
C
C--------------------------------------------------------------
C
            XWC=XW(K)
            XWSTA=XXSTA-XWC
            SQSTA=XWSTA**2
            XWAFT=XXAFT-XWC
            SQAFT=XWAFT**2
            XWFWD=XXFWD-XWC
            SQFWD=XWFWD**2
            DXWAFT=XWAFT*DDXAFT
            DXWFWD=XWFWD*DDXFWD
            WCOSB=CXK(K)
C
            IF (JSAVE(K).EQ.0) GO TO 472
            COSSTA=COSJ(K)
            SINSTA=SINJ(K)
            GO TO 473
  472       CONTINUE
            ARGSTA=WCOSB*XWSTA
            COSSTA=COS(ARGSTA)
            SINSTA=SIN(ARGSTA)
  473       CONTINUE
```

```
C
      IF (ABS(WCOSB*DX).GT.0.01) GO TO 480
C
      DXCOS=DX*COSSTA
      DXSIN=DX*SINSTA
      DX1COS=DX1*COSSTA
      DX1SIN=DX1*SINSTA
      JSAVE(K)=0
      GO TO 490
C
 480  WCOSB2=2.0/WCOSB
      XCSTA=COSSTA*XWSTA
      XSSTA=SINSTA*XWSTA
C
      IF (JSAVE(K).EQ.0) GO TO 481
      COSFWD=COSJJ(K)
      SINFWD=SINJJ(K)
      DWCFWD=DWC(K)
      DWSFWD=DWS(K)
      XCIFWD=XCI(K)
      XSIFWD=XSI(K)
      GO TO 482
C
 481  ARGFWD=WCOSB*XWFWD
      COSFWD=COS(ARGFWD)
      SINFWD=SIN(ARGFWD)
      DWCFWD=(COSFWD-COSSTA)/WCOSB
      DWSFWD=(SINFWD-SINSTA)/WCOSB
      XCIFWD=(SINFWD*XWFWD-XSSTA+DWCFWD)*DDXFWD
      XSIFWD=(XCSTA-COSFWD*XWFWD+DWSFWD)*DDXFWD
C
 482  ARGAFT=WCOSB*XWAFT
      COSAFT=COS(ARGAFT)
      SINAFT=SIN(ARGAFT)
      DWCAFT=(COSSTA-COSAFT)/WCOSB
      DWSAFT=(SINSTA-SINAFT)/WCOSB
      XCIAFT=(XSSTA-SINAFT*XWAFT+DWCAFT)*DDXAFT
      XSIAFT=(COSAFT*XWAFT-XCSTA+DWSAFT)*DDXAFT
C
      DXCOS=(XCIAFT-XCIFWD)/WCOSB-DXWAFT*DWSAFT+DXWFWD*DWSFWD
      DXSIN=(XSIAFT-XSIFWD)/WCOSB+DXWAFT*DWCAFT-DXWFWD*DWCFWD
      DX1COS=XWC*DXCOS+((SQSTA*SINSTA-SQAFT*SINAFT)*DDXAFT+(SQSTA*
     1SINSTA-SQFWD*SINFWD)*DDXFWD+(XSIFWD-XSIAFT)*WCOSB2+XWFWD*XCI
     2FWD-XWAFT*XCIAFT)/WCOSB
      DX1SIN=XWC*DXSIN+((SQFWD*COSFWD-SQSTA*COSSTA)*DDXFWD+(SQAFT*
     1COSAFT-SQSTA*COSSTA)*DDXAFT+(XCIAFT-XCIFWD)*WCOSB2+XWFWD*XSI
     2FWD-XWAFT*XSIAFT)/WCOSB
C
      JSAVE(K)=LSAVE
      IF (LSAVE.EQ.0) GO TO 490
      COSJ(K)=COSAFT
      SINJ(K)=SINAFT
      COSJJ(K)=COSSTA
      SINJJ(K)=SINSTA
      DWC(K)=DWCAFT
      DWS(K)=DWSAFT
      XCI(K)=XCIAFT
      XSI(K)=XSIAFT
C
 490  CONTINUE
      EXPSTA=EXP(WN(K)*YC)*WVAMP(K)
```

```
            DXCOS=DXCOS*EXPSTA
            DXSIN=DXSIN*EXPSTA
            DX1COS=DX1COS*EXPSTA
            DX1SIN=DX1SIN*EXPSTA
C3
C
C---------------------------------------------------------------
C
C      Section 6.2 - Compute exciting forces (wave dynamic
C                    pressure plus diffraction force)
C
C---------------------------------------------------------------
C
C           HVH=COEFF(1,2,J)/WVFRE(K)
C           SVS=COEFF(2,2,J)/WVFRE(K)
C           RVR=COEFF(3,2,J)/WVFRE(K)
C           CCV=COEFF(4,2,J)/WVFRE(K)
C
C-----------------------------------------------------
C
            HAH=THAH(J,K)
            HAH=HAH+1.
            SAS=TSAS(J,K)
            SAS=SAS+1.
            RAR=TRAR(J,K)
            CCA=TCCA(J,K)
C
C-----------------------------------------------------
C
C
            HVH=THVH(J,K)/WVFRE(K)
            SVS=TSVS(J,K)/WVFRE(K)
            RVR=TRVR(J,K)/WVFRE(K)
            CCV=TCCV(J,K)/WVFRE(K)
C
C
C------------------------------------------------------
C
            AH1=CYK(K)*ZSO(K)-CCK(K)*ZCO(K)
            AH2=CCK(K)*ZSO(K)+CYK(K)*ZCO(K)
            AS1=CZK(K)*ZSO(K)
            AS2=CZK(K)*ZCO(K)
            AR1=CZK(K)*ZSY(K)+CCK(K)*ZCZ(K)-CYK(K)*ZSZ(K)
            AR2=CZK(K)*ZCY(K)-CCK(K)*ZSZ(K)-CYK(K)*ZCZ(K)
            FWVEX(1,J)=(CXK(K)*ZSO(K)-CSK(K)*ZCO(K))*DXCOS
     1      +(CXK(K)*ZCO(K)+CSK(K)*ZSO(K))*DXSIN+FWVEX(1,J)
            YF1=AH1*HAH+AH2*HVH
            YF2=AH2*HAH-AH1*HVH
            FWVEX(2,J)=YF1*DXCOS+YF2*DXSIN+FWVEX(2,J)
            ZF1=AS1*SAS+AS2*SVS+(AS1*CCA+AS2*CCV)*WN(K)
            ZF2=AS2*SAS-AS1*SVS+(AS2*CCA-AS1*CCV)*WN(K)
            FWVEX(3,J)=ZF1*DXCOS+ZF2*DXSIN+FWVEX(3,J)
            FWVEX(4,J)=(AR1+(AS1*RAR+AS2*RVR)*WN(K)+AS1*CCA+
     1      AS2*CCV)*DXCOS+(AR2+(AS2*RAR-AS1*RVR)*WN(K)+AS2*CCA
     2      -AS1*CCV)*DXSIN+FWVEX(4,J)
            FWVEX(5,J)=(CXK(K)*ZSZ(K)-CSK(K)*ZCZ(K))*DXCOS+
     1      (CXK(K)*ZCZ(K)+CSK(K)*ZSZ(K))*DXSIN-ZF1*DX1COS-ZF2
     2      *DX1SIN+FWVEX(5,J)
            FWVEX(6,J)=(CSK(K)*ZCY(K)-CXK(K)*ZSY(K))*DXCOS-
     1      (CSK(K)*ZSY(K)+CXK(K)
     2      *ZCY(K))*DXSIN+YF1*DX1COS+YF2*DX1SIN+FWVEX(6,J)
```

```
C3
C
C
C-----------------------------------------------------------------
C          6.3   Speed dependent wave-exciting terms
C-----------------------------------------------------------------
C
       DADX=SPEED*THACX(J,K)/WVFRE(K)   !Turns acc.'s into Vel's
       UWVEX(2,J)=AH1*DADX*DXSIN-AH2*DADX*DXCOS+UWVEX(2,J)
C      Another speed term - spatial derivative of damping in x
C      DCDX=SPEED*(THVCX(J,K)/G)*DVOL*WVFRE(K)/OMEGAX(1)
C      UWVEX(2,J)=UWVEX(2,J)+DCDX*ETAY(J,K)
       UZF1=AS1*TSACX(J,K)+AS2*TSVCX(J,K)
       UZF2=AS2*TSACX(J,K)-AS1*TSVCX(J,K)
       UWVEX(3,J)=(-UZF1*DXSIN+UZF2*DXCOS)*(SPEED/WVFRE(K))
     1          +UWVEX(3,J)

       UWVEX(4,J)=0.        ! Modify this
       UWVEX(5,J)=(-UZF1*DX1SIN+UZF2*DX1COS)*(SPEED/WVFRE(K))
     1          + UWVEX(5,J)
       UWVEX(6,J)=AH1*DADX*DX1SIN-AH2*DADX*DX1COS
C      DCDX1=SPEED*(THVCX(J,K)/G)*ADX1*WVFRE(K)/OMEGAX(1)
C      UWVEX(6,J)=DCDX1*ETAY(J,K)+UWVEX(6,J)
C
C----------------------------------------------------------------------
C
  500    CONTINUE
  510 CONTINUE
C---------------------------------------------------------------------
C
C
C      SECTION 7.0 - OTHER FORCES
C
C      Force resulting from acceleration caused by the rotating
C      coordinates.
C
C---------------------------------------------------------------------
      YA=VT(6)*VT(1)-VT(4)*VT(3)
      YA=YA/G
      ZA=VT(4)*VT(2)-VT(5)*VT(1)
      ZA=ZA/G
      XFN=0.
      YFN=YFYA*YA
      ZFN=ZFZA*ZA
      XMN=XMZA*ZA
      YMN=YMZA*ZA
      ZMN=ZMYA*YA
C
C--------------------------------------------------------------------
C
C
C
C      SECTION 8.0 - INTEGRATE FORCES ALONG LENGTH OF HULL
C
C----------------------------------------------------------------------
C      Section 8.1 - Integrate buoyancy force
C----------------------------------------------------------------------
C
      DO 801 J=ISTA,LSTA
      XFB=FBUOY(1,J)+XFB
      YFB=FBUOY(2,J)+YFB
      ZFB=FBUOY(3,J)+ZFB
      XMB=FBUOY(4,J)+XMB
```

```
      YMB=FBUOY(5,J)+YMB
      ZMB=FBUOY(6,J)+ZMB
801   CONTINUE
C
C-----------------------------------------------------------------
C     Section 8.2 - Integrate damping force
C-----------------------------------------------------------------
C
      DO 802 J=ISTA,LSTA
      XFD=FDAMP(1,J)+XFD
      YFD=FDAMP(2,J)+YFD
      ZFD=FDAMP(3,J)+ZFD
      XMD=FDAMP(4,J)+XMD
      YMD=FDAMP(5,J)+YMD
      ZMD=FDAMP(6,J)+ZMD
802   CONTINUE
C
C----------------------------------------------------------------
C     Section 8.3 - Integrate exciting force
C----------------------------------------------------------------
C
      DO 803 J=ISTA,LSTA
      XFX=FWVEX(1,J)+XFX
      YFX=FWVEX(2,J)+YFX
      ZFX=FWVEX(3,J)+ZFX
      XMX=FWVEX(4,J)+XMX
      YMX=FWVEX(5,J)+YMX
      ZMX=FWVEX(6,J)+ZMX
803   CONTINUE
C
C------------------------------------------------------------------
C       Section 8.4 - Integrate Speed Related forces
C------------------------------------------------------------------
C
C
       IF(JOBFO(2).NE.1) GO TO 814
       DO 804 J=ISTA,LSTA
       XFU=UDAMP(1,J)+UADMA(1,J)+UWVEX(1,J)+XFU
       YFU=UDAMP(2,J)+UADMA(2,J)+UWVEX(2,J)+YFU
       ZFU=UDAMP(3,J)+UADMA(3,J)+UWVEX(3,J)+ZFU
       XMU=UDAMP(4,J)+UADMA(4,J)+UWVEX(4,J)+XMU
       YMU=UDAMP(5,J)+UADMA(5,J)+UWVEX(5,J)+YMU
       ZMU=UDAMP(6,J)+UADMA(6,J)+UWVEX(6,J)+ZMU
804    CONTINUE
814    CONTINUE                !Go directly here to skip over
C
C-----------------------------------------------------------------
C     Section 8.5 - Integrate flare force
C-----------------------------------------------------------------
C
      IF(JOBFO(3).NE.1) GO TO 815
      DO 805 J=ISTA,LSTA
      YFF=FFLAR(2,J)+YFF
      ZMF=FFLAR(6,J)+ZMF
805   CONTINUE
815   CONTINUE              !Directly to here to skip over
C
C------------------------------------------------------------------
C
C
C4    SET UP MATRIX OF FORCE COEFFICIENTS FOR FORCES AND ACCELERATIONS
```

```
C4     IN SHIP COORDINATE DIRECTIONS..
C4
  520 CONTINUE
C4
C
C-------------------------------------------------------------------
C
C
C      SECTION 9.0 - TOTAL FORCES
C
C
C      Add up forces.  Then resolve forces and moments into the
C      ship offset coordinate system.
C
C
C-------------------------------------------------------------------
C
C
C
      XFT=XFB+XFD+XFX+XFN+XFW+XFU
      YFT=YFB+YFD+YFX+YFN+YFW+YFU+YFF
      ZFT=ZFB+ZFD+ZFX+ZFN+ZFU
      XMT=XMB+XMD+XMX+XMN+XMU
      YMT=YMB+YMD+YMX+YMN+YMU
      ZMT=ZMB+ZMD+ZMX+ZMN+ZMU+ZMF
C
      XF=XFT*GAMMA
      YF=YFT*GAMMA
      TEMP=YF
      ZF=ZFT*GAMMA
      YF=YF*CTHETA+ZF*STHETA
      ZF=ZF*CTHETA-TEMP*STHETA
      XM=XMT*GAMMA
      TEMP=YMT
      YM=(YMT*CTHETA+ZMT*STHETA)*GAMMA
      ZM=(ZMT*CTHETA-TEMP*STHETA)*GAMMA
C
C-------------------------------------------------------------------
C
C
C      SECTION 10.0 - LOAD MASS/INERTIA MATRIX A(I,J)
C
C
C------------------------------------------------------------------
C
      A(2,4)=0.0
      DO 530 I=1,9
         AM(I)=RHO*AM(I)
  530 CONTINUE
      CC=CTHETA**2
      CS=CTHETA*STHETA
      SS=STHETA**2
      A(2,2)=YFYA*CC+ZFZA*SS+SMASS
      A(2,3)=(ZFZA-YFYA)*CS
      A(3,2)=A(2,3)
      A(2,5)=ZMYA*CS
      A(2,6)=ZMYA*CC
      A(3,3)=ZFZA*CC+YFYA*SS+SMASS
      A(3,4)=XMZA*CTHETA
      A(4,3)=A(3,4)
      A(4,2)=XMZA*STHETA
      A(4,4)=XMXR+PMI(1,1)
      A(4,5)=XMYR*CTHETA+PMI(1,2)
      A(5,4)=A(4,5)
      A(4,6)=PMI(1,3)-XMYR*STHETA
      A(6,4)=A(4,6)
      A(5,2)=(ZMYA+YMZA)*CS
```

```
      A(3,6)=-A(5,2)
      A(6,3)=A(3,6)
      A(5,3)=YMZA*CC-ZMYA*SS
      A(3,5)=A(5,3)
      A(5,5)=YMYR*CC+ZMZR*SS+PMI(2,2)
      A(5,6)=(ZMZR-YMYR)*CS+PMI(2,3)
      A(6,5)=A(5,6)
      A(6,2)=ZMYA*CC-YMZA*SS
      A(6,6)=ZMZR*CC+YMYR*SS+PMI(3,3)
      RETURN
C
      END
```

```
       SUBROUTINE HULL
C     This subroutine calculates hull girder loads
C-----------------------------------------------------------------
C
      COMMON /C      /   C(3,3)
      COMMON/IO/INPUT,OUTPUT,BIF,OFF,COF
      INTEGER OUTPUT,BIF,OFF,COF
      COMMON /       /   NWAVES, WVSUM,
     1                  WVAMP(20), WVFRE(20), WVDIR(20), WVPHA(20),
     2                  WN(20), WNX(20), WNZ(20),
     3                  CSK(20), CCK(20), CXK(20),
     4                  CYK(20), CZK(20), XW(20)
      COMMON /       /   MSTA, NPROF, NFWD, NAFT, LPTS(25),
     1                  XOFF(25), YOFF(25,25), ZOFF(25,25),
     2                  XPROF(51), YPROF(51)
       COMMON/COEFFX/THAH(25,21),TSAS(25,21),TRAR(25,21),TCCA(25,21),
     *              THVH(25,21),TSVS(25,21),TRVR(25,21),TCCV(25,21),
     *              THACX(25,21),THVCX(25,21),TSACX(25,21),TSVCX(25,21),
     *              TRACX(25,21),TCACX(25,21),THVCY(25,21),
     *              ITP(25),TXP(25),IFP(21),FXP(21)
      COMMON /FORCE /   XF, YF, ZF, XM, YM, ZM
      REAL  FORCE(6)
      EQUIVALENCE (FORCE(1),XF)
      COMMON /LHS   /   Y(13)
      COMMON /MASS  /   RHO, G, GAMMA
      COMMON /MASS  /   DISPL, SMASS, XCG, YCG, ZCG,
     1                  AMX, AMY, AMZ, RADII(6)
      COMMON /MASS  /   PMI(3,3)
      COMMON/RESIST/SPEED
      COMMON /V     /   VX, VY, VZ
      COMMON/SXPROP/ SEGMAS(26),SEGMOX(26),STRMAS(26),STRMOM(26),
     *   STRMOX(26),XBAR(26),YBAR(26),SEGWT(26),NWTSTA
      COMMON/SXFOR/ FBUOY(6,25),FDAMP(6,25),FADMA(6,25),FWVEX(6,25),
     *              UDAMP(6,25),UADMA(6,25),UWVEX(6,25),FFLAR(6,25)
      COMMON/SXGEOM/ AREAS(25),DXS(25),DX1S(25),DX2S(25),
     *               YMOMS(25),ZMOMS(25),ISTA,LSTA
      COMMON/ACCEL/ACC(6)
      COMMON/LOADS/ SIN2(25),SIN3(25),SIN4(25),SIN5(25),SIN6(25),
     1              V2(25),V3(25),V4(25),V5(25),V6(25),
     2              RED(6,25)
      COMMON/OPTION/JOBCO,JOBFO(10)
C
C
C      SECTION 1.0  GET SECTIONAL INERTIA/MOMENT OF INERTIAS
C
C
C-----------------------------------------------------------------------
       DO 90 J=1,MSTA
       XBAR(J)=XOFF(J)
       STRMAS(J)=SEGMAS(J)
       SIN2(J)=0.
       SIN3(J)=0.
       SIN4(J)=0.
       SIN5(J)=0.
       SIN6(J)=0.
       DO 91  I=1,6
       RED(I,J)=0.
91     CONTINUE
90     CONTINUE
C
       DO 100 J=1,MSTA
       DSTRPI=XOFF(J)
       DO 101 JJ=1,J
```

```
C
      TVACCL=ACC(2)+XBAR(JJ)*ACC(6)
      THACCL=ACC(3)-XBAR(JJ)*ACC(5)+YBAR(JJ)*ACC(4)
C------NOTE!!! We are adding in weight of the section
C      by including gravitational acceleration.  This
C      is different from standard strip theory load
C      computations. (They compute wave induce dynamic
C      loads)  We are computing total loads at each time
C      step... We have the TOTAL HYDROSTATIC RESTORING
C      force not just the change due to unit wave/motion
C      [see eq. 70  STF METHOD/TRANS SNAME '70]
C
      GVACCL=-G*COS(Y(4))
      GHACCL=G*SIN(Y(4))
C
      HACCL=-THACCL+GHACCL
      VACCL=-TVACCL+GVACCL
C
C
      SIN2(J)=STRMAS(JJ)*VACCL+SIN2(J)
      SIN3(J)=STRMAS(JJ)*HACCL+SIN3(J)
      SIN4(J)=STRMOX(JJ)*ACC(4)-((STRMAS(JJ)*YBAR(JJ))*
     1      (ACC(3)+XBAR(JJ)*ACC(5)))+SIN4(J)
C     DSTRPI=XOFF(J)-(.5*DXS(J))
      ZETMX=XBAR(JJ)-DSTRPI
      SIN5(J)=ZETMX*STRMAS(JJ)*HACCL+SIN5(J)
      SIN6(J)=ZETMX*STRMAS(JJ)*VACCL+SIN6(J)
C
 101  CONTINUE
C
C---------------------------------------------------------
C
C-----------------------------------------------------------
C
 100  CONTINUE
C
C-----------------------------------------------------------------
C
C     SECTION 2.0  GET ADDED MASS FORCES FADMA( , )
C
C----------------------------------------------------------------
C
      DO 200 J=1,MSTA
      DO 200 I=1,6
      FADMA(I,J)=0.
200   CONTINUE
C
      DO 201 J=ISTA,LSTA
C
      AREA=AREAS(J)
      DX=DXS(J)
      DX1=DX1S(J)
      DX2=DX2S(J)
      YMOM=YMOMS(J)
      ZMOM=ZMOMS(J)
C
      DVOL=AREA*DX
      ADX1=AREA*DX1
      ADX2=AREA*DX2
      YDX=YMOM*DX
      ZDX=ZMOM*DX
```

```
C
      HAH=THAH(J,21)/G
      SAS=TSAS(J,21)/G
      RAR=TRAR(J,21)/G
      CCA=TCCA(J,21)/G
C
C
      AS=ACC(5)*ADX1-ACC(3)*DVOL
      AR=ACC(4)*DVOL
      FADMA(1,J)=0.
      FADMA(2,J)=(ACC(4)*ZDX-ACC(2)*DVOL-ACC(6)*ADX1)*HAH
      FADMA(3,J)=AS*SAS-AR*CCA
      FADMA(4,J)=AS*CCA-AR*RAR
      FADMA(5,J)=(ACC(3)*ADX1-ACC(5)*ADX2)*SAS+ACC(4)
     1            *ADX1*CCA
      FADMA(6,J)=(ACC(4)*ZMOM*DX1-ACC(2)*ADX1-
     1            ACC(6)*ADX2)*HAH
C
C
 201  CONTINUE
C
C    SECTION 3.0  GET HYDRO FORCES (CALL THEM RED) AND THEN
C                 GET LOADS
C-------------------------------------------------------------------
C
      DO 900 J=1,MSTA
900   CONTINUE
C
      DO 300 I=2,3
      DO 301 J=1,MSTA
      REDSUM=0.
      DO 302 JJ=1,J
      REDSUM=-FBUOY(I,JJ)-FDAMP(I,JJ)-FADMA(I,JJ)-FWVEX(I,JJ)
     *         +REDSUM
      UTERMS=-UDAMP(I,JJ)-UADMA(I,JJ)-UWVEX(I,JJ)
      FTERMS=-FFLAR(I,JJ)
      IF(JOBFO(2).EQ.1) REDSUM=REDSUM+UTERMS
      IF(JOBFO(3).EQ.1) REDSUM=REDSUM+FTERMS
302   CONTINUE
      RED(I,J)=REDSUM*GAMMA
301   CONTINUE
300   CONTINUE
C
C----------------------------------------------------
C
C
C     Now do moments
C
      DO 310 IX=5,6
      IF(IX.EQ.5) I=3
      IF(IX.EQ.6) I=2
      DO 311 J=1,MSTA
      DSTRPI=XOFF(J)
C     DSTRPI=XOFF(J)-(.5*DXS(J))
      REDSUM=0.
C
      DO 312 JJ=1,J
C
      ARMJJ=XBAR(JJ)
      ARMX=ARMJJ-DSTRPI
      FDARM=FDAMP(I,JJ)*ARMX
      FWARM=FWVEX(I,JJ)*ARMX
```

```
      FBARM=FBUOY(I,JJ)*ARMX
      FAARM=FADMA(I,JJ)*ARMX
      FUARM=(UADMA(I,JJ)+UDAMP(I,JJ)+UWVEX(I,JJ))*ARMX
      FFARM=FFLAR(I,JJ)*ARMX
C
      REDSUM=-FBARM-FDARM-FAARM-FWARM+REDSUM
      IF(JOBFO(2).EQ.1) REDSUM=REDSUM-FUARM
      IF(JOBFO(3).EQ.1) REDSUM=REDSUM-FFARM
 312  CONTINUE
C
      RED(IX,J)=REDSUM*GAMMA
C
 311  CONTINUE
 310  CONTINUE
C
C---------------------------------------------------------
C
C
      DO 305 J=1,MSTA
      V2(J)=SIN2(J)-RED(2,J)
      V3(J)=SIN3(J)-RED(3,J)
      V4(J)=SIN4(J)-RED(4,J)
      V5(J)=SIN5(J)-RED(5,J)
      V6(J)=SIN6(J)-RED(6,J)
C
305   CONTINUE
C
C-----OUTPUT FORCES FOR EACH SECTION
      RETURN
69     FORMAT(1X,I2,1X,6F15.4)
      END
```

```
      SUBROUTINE KRYLOV (J)
C
      COMMON /      /   NWAVES, WVSUM,
     1                  WVAMP(20), WVFRE(20), WVDIR(20), WVPHA(20),
     2                  WN(20), WNX(20), WNZ(20),
     3                  CSK(20), CCK(20), CXK(20),
     4                  CYK(20), CZK(20), XW(20)
      COMMON /      /   MSTA, NPROF, NFWD, NAFT, LPTS(25),
     1                  XOFF(25), YOFF(25,25), ZOFF(25,25),
     2                  XPROF(51), YPROF(51)
      COMMON /      /   INPTS(25), IPOINT(25), YWET(25,25), ETA(25,25),
     1                  ZLAST(25,25), ZWET(25,25), ZNEXT(25,25)
      COMMON /      /   ZCO(20), ZSO(20), ZCY(20), ZSY(20),
     1                  ZCZ(20), ZSZ(20)
      COMMON /      /   DZK(20)
      COMMON /      /   JSAVE(20), COSJ(20), SINJ(20),
     $                  COSJJ(20), SINJJ(20),
     1                  DWC(20), DWS(20), XCI(20), XSI(20)
      COMMON /      /   ISAVE(20), CSI(20), SNI(20),
     $                  CSII(20), SNII(20),
     1                  DKC(20), DKS(20), ZCI(20), ZSI(20)
      COMMON /AREA  /   AREA, YMOM, ZMOM
      AREA=0.0
      YMOM=0.0
      ZMOM=0.0
      KSAVE=0
      DO 110 K=1,NWAVES
         ZCO(K)=0.0
         ZSO(K)=0.0
         ZCY(K)=0.0
         ZSY(K)=0.0
         ZCZ(K)=0.0
         ZSZ(K)=0.0
      ISAVE(K)=0
  110 CONTINUE
      LI=INPTS(J)
      II=IPOINT(J)
      LAST=II-1
      IF (LAST.EQ.0) LAST=LI
  111 DO 190 I=II,LI
         ETA1=ETA(I,J)
         Y1=YWET(I,J)
         Z0=ZLAST(I,J)
         Z1=ZWET(I,J)
         Z2=ZNEXT(I,J)
C3
         DZ0=Z1-Z0
         DZ2=Z2-Z1
         DZ=(Z2-Z0)/2.0
         ZA=Z1-DZ0/3.0
         ZB=Z1+DZ2/3.0
         DZ1=(DZ2*ZB+DZ0*ZA)/2.0
         AREA=AREA+(ETA1-Y1)*DZ
         ZMOM=ZMOM+(ETA1-Y1)*DZ1
         YMOM=(ETA1**2-Y1**2)/2.0*DZ+YMOM
         IF (NWAVES.EQ.0) GO TO 190
C
         IF (KSAVE.EQ.0) GO TO 112
         SQ0=SQ1
         SQ1=SQ2
         DDZ0=DDZ2
```

```
      DZZ0=DZZ2
      GO TO 130
 112  SQ0=Z0**2
      SQ1=Z1**2
      IF (DZ0.EQ.0.0) GO TO 120
      IF ((DZ2/DZ0).GT.1.0E+06) GO TO 120
      DDZ0=1.0/DZ0
      DZZ0=Z0*DDZ0
      GO TO 130
 120  DDZ0=0.0
      DZZ0=0.0
 130  IF (DZ2.EQ.0.0) GO TO 140
      IF ((DZ0/DZ2).GT.1.0E+06) GO TO 140
      DDZ2=1.0/DZ2
      DZZ2=Z2*DDZ2
      GO TO 150
 140  DDZ2=0.0
      DZZ2=0.0
 150  SQ2=Z2**2
      KSAVE=0
      IF (I.EQ.LAST) GO TO 151
      IN=I
      IF (IN.EQ.LI) IN=0
      IF(Z1.EQ.ZLAST(IN+1,J) .AND. Z2.EQ.ZWET(IN+1,J)) KSAVE=1
 151  CONTINUE
      IF(ETA1.GT.0..AND.Y1.GT.0.) ETA1=Y1
      DO 180 K=1,NWAVES
         CK=CCK(K)
         YK=CYK(K)
         ZK=CZK(K)
         EE=EXP(CK*ETA1)
         EY=EXP(CK*Y1)
         ARG=YK*ETA1
         CE=COS(ARG)
         SE=SIN(ARG)
         ARG=YK*Y1
         CY=COS(ARG)
         SY=SIN(ARG)
         D=CK**2+YK**2
         CK=CK/D
         YK=YK/D
         COSE=(CK*CE+YK*SE)*EE
         SINE=(CK*SE-YK*CE)*EE
         COSY=(CK*CY+YK*SY)*EY
         SINY=(CK*SY-YK*CY)*EY
         YC0=COSE-COSY
         YS0=SINE-SINY
         YCY=ETA1*COSE-Y1*COSY-CK*YC0-YK*YS0
         YSY=ETA1*SINE-Y1*SINY-CK*YS0+YK*YC0
C
         IF (ISAVE(K).EQ.0) GO TO 152
         COS1=CSI(K)
         SIN1=SNI(K)
         GO TO 153
 152     ARG1=ZK*Z1
         COS1=COS(ARG1)
         SIN1=SIN(ARG1)
 153     CONTINUE
         IF (ABS(ZK*DZ).GT.0.01) GO TO 160
C
         DZCOS=DZ*COS1
```

```
            DZSIN=DZ*SIN1
            DZ1COS=DZ1*COS1
            DZ1SIN=DZ1*SIN1
            ISAVE(K)=0
            GO TO 170
C
  160       ZK2=DZK(K)
            ZCOS1=COS1*Z1
            ZSIN1=SIN1*Z1
C
            IF (ISAVE(K).EQ.0) GO TO 161
            COS0=CSII(K)
            SIN0=SNII(K)
            DKC0=DKC(K)
            DKS0=DKS(K)
            ZCI0=ZCI(K)
            ZSI0=ZSI(K)
            GO TO 162
C
  161       ARG0=ZK*Z0
            COS0=COS(ARG0)
            SIN0=SIN(ARG0)
            DKC0=(COS1-COS0)/ZK
            DKS0=(SIN1-SIN0)/ZK
            ZCI0=(ZSIN1-SIN0*Z0+DKC0)*DDZ0
            ZSI0=(COS0*Z0-ZCOS1+DKS0)*DDZ0
C
  162       ARG2=ZK*Z2
            COS2=COS(ARG2)
            SIN2=SIN(ARG2)
            DKC2=(COS2-COS1)/ZK
            DKS2=(SIN2-SIN1)/ZK
            ZCI2=(SIN2*Z2-ZSIN1+DKC2)*DDZ2
            ZSI2=(ZCOS1-COS2*Z2+DKS2)*DDZ2
C
            DZCOS=(ZCI0-ZCI2)/ZK-DZZ0*DKS0+DZZ2*DKS2
            DZSIN=(ZSI0-ZSI2)/ZK+DZZ0*DKC0-DZZ2*DKC2
            DZ1COS=((SQ1*SIN1-SQ0*SIN0)*DDZ0+(SQ1*SIN1-SQ2*SIN2)*DDZ2+(Z
     1      SI2-ZSI0)*ZK2+Z2*ZCI2-Z0*ZCI0)/ZK
            DZ1SIN=((SQ2*COS2-SQ1*COS1)*DDZ2+(SQ0*COS0-SQ1*COS1)*DDZ0+(Z
     1      CI0-ZCI2)*ZK2+Z2*ZSI2-Z0*ZSI0)/ZK
C
            ISAVE(K)=KSAVE
            IF (KSAVE.EQ.0) GO TO 170
            CSI(K)=COS2
            SNI(K)=SIN2
            CSII(K)=COS1
            SNII(K)=SIN1
            DKC(K)=DKC2
            DKS(K)=DKS2
            ZCI(K)=ZCI2
            ZSI(K)=ZSI2
C
  170       CONTINUE
C
            ZC0(K)=ZC0(K)+DZCOS*YC0-DZSIN*YS0
            ZS0(K)=ZS0(K)+DZCOS*YS0+DZSIN*YC0
            ZCY(K)=ZCY(K)+DZCOS*YCY-DZSIN*YSY
            ZSY(K)=ZSY(K)+DZCOS*YSY+DZSIN*YCY
            ZCZ(K)=ZCZ(K)+DZ1COS*YC0-DZ1SIN*YS0
            ZSZ(K)=ZSZ(K)+DZ1COS*YS0+DZ1SIN*YC0
```

```
  180     CONTINUE
  190 CONTINUE
      IF (LI.EQ.LAST) RETURN
      LI=LAST
      II=1
      GO TO 111
C
      END
      FUNCTION LNEQF(M,N,N1,A,B,DTRMNT,Z)
C.. SOLVES SIMULTANEOUS LINEAR EQUATIONS BY GAUSSIAN REDUCTION.
C.. FORTRAN IV EQUIVALENT OF LNEQS.
      REAL A(M,M),B(M,M) ,Z(M),DTRMNT,RMAX,RNEXT,W,DOV
      NM1=N-1
      DO 200 J=1,NM1
         J1=J+1
C.. FIND ELEMENT OF COL J, ROWS J-N, WHICH HAS MAX ABSOLUTE VALUE.
         LMAX=J
         RMAX=ABS(A(J,J))
         DO 110 K=J1,N
            RNEXT=ABS(A(K,J))
            IF (RMAX.GE.RNEXT) GO TO 110
            RMAX=RNEXT
            LMAX=K
  110    CONTINUE
         IF (LMAX.NE.J) GO TO 120
C.. MAX ELEMENT IN COLUMN IS ON DIAGONAL
         IF (A(J,J)) 150,290,150
C.. MAX ELEMENT IS NOT ON DIAGONAL.  EXCHANGE ROWS J AND LMAX.
  120    DO 130 L=J,N
            W=A(J,L)
            A(J,L)=A(LMAX,L)
  130    A(LMAX,L)=W
         DO 140 L=1,N1
            W=B(J,L)
            B(J,L)=B(LMAX,L)
  140    B(LMAX,L)=W
         DTRMNT=-DTRMNT
C.. ZERO COLUMN J BELOW THE DIAGONAL.
  150    Z(J)=1./A(J,J)
         DO 190 K=J1,N
            IF (A(K,J)) 160,190,160
  160       W=-Z(J)*A(K,J)
            DO 170 L=J1,N
  170       A(K,L)=W*A(J,L)+A(K,L)
            DO 180 L=1,N1
  180       B(K,L)=W*B(J,L)+B(K,L)
  190    CONTINUE
  200 CONTINUE
      IF (A(N,N)) 210,290,210
  210 Z(N)=1./A(N,N)
C.. OBTAIN SOLUTION BY BACK SUBSTITUTION.
      DO 220 L=1,N1
  220 B(N,L)=Z(N)*B(N,L)
      DO 250 K=1,NM1
         J=N-K
         J1=J+1
         DO 240 L=1,N1
            W=0.
            DO 230 I=J1,N
  230       W=A(J,I)*B(I,L)+W
  240    B(J,L)=(B(J,L)-W)*Z(J)
```

```
  250 CONTINUE
C..  EVALUATE DETERMINANT.
      IF (DTRMNT) 260,280,260
  260 DO 270 J=1,N
  270 DTRMNT=DTRMNT*A(J,J)
  280 LNEQF=1
      RETURN
C..  SINGULAR MATRIX, SET ERROR FLAG.
  290 LNEQF=2
      DTRMNT=0.
      RETURN
C
      END
```

PROGRAM HYDREX2

```
      PROGRAM HYDREX2
C
      CHARACTER*25 OFFIL,BIFIL,COFIL
      CHARACTER*30 TITLE
      COMMON/HEAD/TITLE
      COMMON/IOFILE/ OFFIL,BIFIL,COFIL
      COMMON/IO/INPUT,OUTPUT,BIF,OFF,COF
      INTEGER OUTPUT,BIF,OFF,COF
      COMMON /COEFF4 /    COEFF4(6,12,8,25),AREAN(25,6)
      COMMON/SHIP/ISTA,LSTA,ISWL,LSWL,TF,TA,XCG,YCG,DISPL
      COMMON /DRFT12/ DRAFT(6,2),IDRAFT
      COMMON /SIGMA /    NK, SIGMA(24), SIGMAO, ERRO, OM(12)
      COMMON /U      /    RHO, G
      COMMON/GEOMETRY/MSTA,LPTS(25),YOFF(25,25),NAFT,XAFT(25),
     *       YAFT(25),NFWD,XFWD(25),YFWD(25),XOFF(25),
     *      ZOFF(25,25),XFPERP,XAPERP,SHIPL,SHIPB,SHIPT,
     *      Y1(21,25),ZWL(25),WL(25),INPTS(25),XWLF,XWLA,XXF,
     *      XXA,TAN,NON,NOE,NWL,CR,XXFWD,XXSTA,XXAFT,DX,DX1,
     *      DX2,Z2(21),Y2(21),ZZ(20),YY(20),SNE(20),CSE(20),
     *      DEL(20),ROL(20),ADJUST,WMAX,YMAX,ZMAX,AREA,VERT
      LOGICAL WL,ADJUST
      DATA  RHO  /0.00088861607142/
      DATA ERRO/1.0E-37/
C     DATA OM/ 0.000, 0.200, 0.400, 0.600, 0.800, 1.000,
C    1         1.200, 1.600, 2.000, 2.400, 2.800, 9.999/
C     DATA OM/0.0,0.0992,.1718,.243,.3137,.3842,.45469,
C    *         .525,.59534,.49611,.7358,9.999/
      DATA OM/0.0,0.099,0.2,0.3,0.4,0.5,0.6,0.7,0.9,
     *        1.3,1.6,2.2/
      DATA INPUT/5/,OUTPUT/6/,BIF/1/,OFF/2/,COF/3/
C-----------------------------------------------------------------
C
C *** 1.0  Get DATA file names and assign/open files
C
      TYPE 902
      ACCEPT 901,BIFIL
      TYPE 903
      ACCEPT 901,OFFIL
      TYPE 904
      ACCEPT 901,COFIL
C
      OPEN(UNIT=BIF,STATUS='OLD',FILE=BIFIL)
      OPEN(UNIT=OFF,STATUS='OLD',FILE=OFFIL)
      OPEN(UNIT=COF,STATUS='NEW',FORM='UNFORMATTED',FILE=COFIL)
C
C *** 2.0 Read in data from two files
      CALL INDATA
      CLOSE(BIF)
      CLOSE(OFF)
      DO 10 J=1,6
      DRAFT(J,2)=DRAFT(J,1)                !Aft draft = fwd draft
 10   CONTINUE
C
C *** 3.0 Float ship at indexed draft then compute added mass
C           and damping coefficients using Frank Close Fit.
      DO 20 IDRAFT=1,6
           CALL FLOAT
           CALL HYDRO
   20 CONTINUE
C
C *** 4.0 Write to unformatted coefficient file containing
```

```
C         added mass and damping coefficients for each station
C         at six drafts and twelve frequencies.
C
      WRITE (COF) MSTA          !Number of Stations
      WRITE (COF) (OM(I),I=1,12)        !Frequency (rad/sec)
      WRITE (COF) (DRAFT(I,1),I=1,6)    !Drafts
      DO 30 L=1,MSTA    !Station index
      DO 30 K=1,6       !Draft index
      DO 30 J=1,12      !Frequency index
      WRITE(COF) (COEFF4(K,J,I,L),I=1,8)
   30 CONTINUE
      WRITE (COF) ((AREAN(L,K),K=1,6),L=1,MSTA)  !Section areas
      STOP
C
 901  FORMAT(A)
 902  FORMAT(' Name of Basic INPUT File [BIF] ? > '$)
 903  FORMAT(' Name of Offset File      [OFF] ? > '$)
 904  FORMAT(' Name of Coefficent File  [COF] ? > '$)
C
      END
      SUBROUTINE HYDRO
C
C     CALCULATION OF HYDRODYNAMIC FORCE COEFFICIENTS FOR THE SHIP.
C
      COMMON/IO/INPUT,OUTPUT,BIF,OFF,COF
      INTEGER OUTPUT,BIF,OFF,COF
      COMMON/HEAD/TITLE
      CHARACTER*30 TITLE
      COMMON /COEFF4/   COEFF4(6,12,8,25),AREAN(25,6)
      COMMON/DRFT12/DRAFT(6,2),IDRAFT
      COMMON/SHIP/ISTA,LSTA,ISWL,LSWL,TF,TA,XCG,YCG,DISPL
      COMMON /SIGMA /   NK, SIGMA(24), SIGMAO, ERRO, OM(12)
      COMMON /U     /   RHO, G
      COMMON/GEOMETRY/MSTA,LPTS(25),YOFF(25,25),NAFT,XAFT(25),
     *       YAFT(25),NFWD,XFWD(25),YFWD(25),XOFF(25),
     *       ZOFF(25,25),XFPERP,XAPERP,SHIPL,SHIPB,SHIPT,
     *       Y1(21,25),ZWL(25),WL(25),INPTS(25),XWLF,XWLA,XXF,
     *       XXA,TAN,NON,NOE,NWL,CR,XXFWD,XXSTA,XXAFT,DX,DX1,
     *       DX2,Z2(21),Y2(21),ZZ(20),YY(20),SNE(20),CSE(20),
     *       DEL(20),ROL(20),ADJUST,WMAX,YMAX,ZMAX,AREA,VERT
      LOGICAL ADJUST, WL
      COMMON /       /  HA1, SA1, RA1, CA1, HV1, SV1, RV1, CV1,
     1                  RHO2, RSIG, WN, W1, W2, ERR, XRI, YRI, EJT
      COMMON //HAO(24),SAO(24),RAO(24),CAO(24),
     1                  HVO(24), SVO(24), RVO(24), CVO(24)
      REAL              FA(24,8)
      EQUIVALENCE      (FA(1,1), HAO(1))
      COMMON /       /  BLOGP(20,20), YLOGP(20,20),
     1                  BLOGM(20,20), YLOGM(20,20)
      REAL COEFFO(8), COEFF1(8)
      EQUIVALENCE (COEFF1(1),HA1)
      DATA  TPI   /6.2831853072/
      NODES=NWL
      ERR=ERRO
      RHO2=RHO*2.0
      XXSTA=XXF
      XWO=XWLF
      VOLO=0.0
      VOL1=0.0
      VOLV=0.0
      WPO=0.0
```

```
      WP1=0.0
      WP2=0.0
      WPT=0.0
C     CR  IS THE CENTER OF ROLL.  ROLL IS ASSUMED TO BE ABOUT THE
C     HORIZONTAL AXIS THROUGH THE GIVEN CENTER OF GRAVITY.
      CR=YCG-DRAFT(IDRAFT,1)-(XFPERP-XCG)*TAN
C     STRIP CALCULATION PROCEDURE FOR EACH WET STATION..
C
C
      DO 240 J=ISTA,LSTA
         XXFWD=XXSTA
         XXSTA=XOFF(J)
         IF (J.EQ.LSTA) GO TO 122
         XXAFT=XOFF(J+1)
         XW2=XXAFT
         IF (J.EQ.LSWL) XW2=XWLA
         GO TO 124
  122    XXAFT=XXA
         XW2=XWLA
  124    X1=XOFF(J)-XCG
C
C        INTEGRATE OVER WATERPLANE OF THE SHIP..
C
         IF (J.LT.ISWL) GO TO 130
         DXFWD=XWO-XXSTA
         XWO=XXSTA
         IF (.NOT.WL(J)) GO TO 130
         DXAFT=XWO-XW2
         DX=DXFWD+DXAFT
         A=X1-DXAFT/3.0
         B=X1+DXFWD/3.0
         XW2=DXFWD*B*B+DXAFT*A*A+(DXFWD**3+DXAFT**3)/18.0
         XW1=DXFWD*B+DXAFT*A
         ZW=ZWL(J)
         DZW=ZW*DX
         WPO=WPO+DZW
         WP1=WP1+ZW*XW1
         WP2=WP2+ZW*XW2
         WPT=WPT+DZW*ZW**2
C
C        SET UP STATION GEOMETRY..
C
  130    NON=INPTS(J)
         IF (NON.EQ.0) GO TO 140
         NWL=0
         IF (WL(J)) NWL=NODES
         CALL STATN (ZOFF(1,J),Y1(1,J),ZWL(J),INPTS(J))
         IF (NON.LE.0) GO TO 140
C
C        INTEGRATE OVER SUBMERGED VOLUME OF THE SHIP..
C
         DXFWD=XXFWD-XXSTA
         DXAFT=XXSTA-XXAFT
         DX=(DXFWD+DXAFT)/2.0
         VOLO=VOLO+AREA*DX
         A=X1-DXAFT/3.0
         B=X1+DXFWD/3.0
         DX2=(DXFWD*B*B+DXAFT*A*A)/2.0+(DXFWD**3+DXAFT**3)/36.0
         DX1=(DXFWD*B+DXAFT*A)/2.0
         VOL1=VOL1+AREA*DX1
         VOLV=VOLV+VERT*DX
```

```
  140    CONTINUE
C
         DO 150 I=1,8
            COEFFO(I)=0.0
  150    CONTINUE
C
         CALL GIRL
C
C
         DO 210 K=1,12
            ESIG=OM(K)
            RSIG=ESIG*RHO2
            WN=ESIG*ESIG/G
C *** Zero (K=1) or Infinite (K=12) Frequency Computations
         IF(K.EQ.1.OR.K.EQ.12) THEN
            CALL BEER(K)
         ELSE
C *** Non zero and non-infinite frequency computations
            W1=TPI/WN
            W2=2.0/WN
            CALL WINE (K)
       ENDIF
C
C---------------------------------------------------------------
C
 180        CONTINUE
C
            DO 190 I=1,8
               COEFF4(IDRAFT,K,I,J)=COEFF1(I)/AREA
  190       CONTINUE
C
            AREAN(J,IDRAFT)=AREA
            HAO(K)=HA1
            SAO(K)=SA1
            RAO(K)=RA1
            CAO(K)=CA1
            HVO(K)=HV1
            SVO(K)=SV1
            RVO(K)=RV1
            CVO(K)=CV1
 210     CONTINUE
C---------------------------------------------------------------
C
C        PRINTOUT Offset and Hydrodynamic Coefficient Info.
C
         WRITE (OUTPUT,300) TITLE
         WRITE(OUTPUT,301) J,XOFF(J),AREA,DRAFT(IDRAFT,1),CR,
     *                  DRAFT(IDRAFT,2)
         WRITE (OUTPUT,302)
         WRITE (OUTPUT,310) (OM(K),HAO(K),HVO(K),SAO(K),SVO(K),RAO(K),
     1   RVO(K),CAO(K),CVO(K),K=1,12)
C *** Offset Information
         WRITE(OUTPUT,705)
               WRITE(OUTPUT,706)
         N=LPTS(J)
         M=0
         IF (J.LT.ISTA) GO TO 670
         IF (J.GT.LSTA) GO TO 670
         M=INPTS(J)
         IF(M.NE.0)WRITE(OUTPUT,710) (YOFF(I,J),ZOFF(I,J),Y1(I,J),ZOFF
     1   (I,J),I=1,M)
```

```
         IF (WL(J)) WRITE (OUTPUT,720) ZWL(J)
  670    M=M+1
         IF (M.LE.N) WRITE (OUTPUT,730) (YOFF(I,J),ZOFF(I,J),I=M,N)
C
  220    CONTINUE
  240 CONTINUE
      GAMMA=RHO*G
      DISPL=VOLO*GAMMA
C
      WPT=WPT/3.0
      YFYTO=WPO*GAMMA
      YFZRO=WP1*GAMMA
      XWA=0.0
      IF (WPO.NE.0.0) XWA=WP1/WPO
      XCB=VOL1/VOLO
      YCB=VOLV/VOLO
      XBM=WPT/VOLO
      ZBM=(WP2-WP1*XWA)/VOLO
      XGM=XBM+YCB
      ZGM=ZBM+YCB
      XWA=XWA+XCG
      XCB=XCB+XCG
      YCB=YCB+YCG
      XMXRO=XGM*DISPL
      ZMZRO=ZGM*DISPL
      WRITE (OUTPUT,270) TITLE
      WRITE (OUTPUT,281) DRAFT(IDRAFT,1),DRAFT(IDRAFT,2)
      WRITE(OUTPUT,290) WPO,XWA,WP1,WP2,WPT
      WRITE(OUTPUT,291) DISPL,VOLO,XCB,YCB,VOL1,VOLV
      WRITE(OUTPUT,292) ZBM,XBM,ZGM,XGM
      WRITE(OUTPUT,293) XMXRO,ZMZRO,YFYTO,YFZRO
      WRITE(OUTPUT,294) XXF,XXA
      WRITE (OUTPUT,280) XCG,YCG
C *** OUTPUT OF NOTES
      WRITE(OUTPUT,420)
      IF (ADJUST) WRITE( OUTPUT,430) ZMAX,YMAX
      IF (.NOT.ADJUST) WRITE (OUTPUT,440)
      IF (NWL.GT.0) GO TO 110
      GO TO 120
  110 WRITE (OUTPUT,460) NWL
      IF (WMAX.GT.0.0) WRITE (OUTPUT,470) WMAX
  120 CONTINUE
C
      RETURN
  270 FORMAT (1H1/,81(1H=)/,1X,A30,'HYDROSTATIC COEFFICIENTS'
     *,T64,'PROGRAM HYDREX',/,81(1H=)/)
  280 FORMAT (//81(1H-)/,' NOTE: All moments are about',
     *'center of gravity' /,7X,'XCG=',F11.3,' YCG=',F11.3,/)
  281 FORMAT(T5,' DRAFT (fwd) =',F8.2,/,T5,' DRAFT (aft) =',F8.2/)
  290 FORMAT(T10,'-------------- WATERPLANE ---------------',/
     *    T10,'Area                        ',T40,F13.2,T60,' units ',/
     *    T10,'LCF                         ',T40,F13.2,T60,' units ',/
     *    T10,'1st Long. Moment            ',T40,E13.7,T60,' units ',/
     *    T10,'2nd Long. Moment            ',T40,E13.7,T60,' units ',/
     *    T10,'2nd Transv. Moment          ',T40,E13.7,T60,' units ',/)
 291  FORMAT(T10,'-------------- VOLUME ----------------',/
     *    T10,'Displacement                ',T40,F13.2,T60,' units ',/
     *    T10,'Volume of Displacement      ',T40,E13.7,T60,' units ',/
     *    T10,'LCB                         ',T40,F13.2,T60,' units ',/
     *    T10,'VCB                         ',T40,F13.2,T60,' units ',/
     *    T10,'Long. Moment                ',T40,E13.7,T60,' units ',/
```

```
     *    T10,'Vert. Moment                      ',T40,E13.7,T60,' units ',/)
292   FORMAT(T10,'---------  METACENTRIC HEIGHTS ---------',/
     *    T10,'BM (longitudinal)                 ',T40,F13.2,T60,' units ',/
     *    T10,'BM (transverse)                   ',T40,F13.2,T60,' units ',/
     *    T10,'GM (longitudinal)                 ',T40,F13.2,T60,' units ',/
     *    T10,'GM (transverse)                   ',T40,F13.2,T60,' units ',/)
293   FORMAT(T10,'----------HYDROSTATIC FORCES ----------',/
     *    T10,'Roll Restoring Moment             ',T40,E13.7,T60,' units ',/
     *    T10,'Pitch Restoring Moment            ',T40,E13.7,T60,' units ',/
     *    T10,'Heave Restoring Force             ',T40,E13.7,T60,' units ',/
     *    T10,'Pitch Induced Heave Force         ',T40,E13.7,T60,' units ',/)
294   FORMAT(/,T5,' LWL begins at ',F10.2,/
     *       ,T5,' LWL ends at   ',F10.2,/)
  300 FORMAT(1H1/,81(1H=),/,1X,A30,'ADDED MASS/DAMPING COEFFICIENTS'
     *,T66,'PROGRAM HYDREX'/,81(1H=))
  301 FORMAT (T36,'Station 'I2/,T36,'----------'/,
     * T5,' Dist. from F.P. ',F8.2,T50,'Area            ',F11.3,/
     * T5,' DRAFT (fwd)      ',F8.2,T50,'Roll Ctr abv WL',F11.3/
     * T5,' DRAFT (aft)      ',F8.2,//)
  302 FORMAT(10X,' -----HEAVE----',
     *4X,'-----SWAY-----',4X,'------ROLL------',
     *4X,'--SWAY-ROLL---',/,14X,'A22',5X,'B22',
     *7X,'A33',5X,'B33',8X,'A44',6X,'B44',6X,'A34',6X,'B34',/
     *,2X,'Freq.'/)
  309 FORMAT ((1X,0PF22.4,3(5X,1P2E10.2)))
  310 FORMAT(1X,F5.2,3X,F8.4,F8.4,2X,F8.4,F8.4,
     *          2X,F9.2,F9.2,F9.1,F9.1)
C
  420 FORMAT (1H1/,33(1H*),' N O T E S ',33(1H*)/)
  430 FORMAT(T5,'Generate additional offset points.'/,
     *          T5,'Maximum segment height=',F10.3/,
     *          T5,'Maximum segment width =',F10.3/)
  440 FORMAT(/T5,'Use segments as defined by table of offsets.')
  450 FORMAT(T5,'No internal free-surface segments are used.')
  460 FORMAT(T5,I3,2X,'nodes for internal free-surface.')
  470 FORMAT(5X,'Add internal nodes if surface segment lengths',
     *' exceed',F10.3)
  706 FORMAT (6X,'Height above',5X,'Half',15X,
     *'Submerged Offsets'/,10X,'Baseline',5X,'BREADth',
     *14X,'-Y-',9X,'-Z-'/)
  705 FORMAT (/////,33(1H-),'STATION OFFSETS',32(1H-),/)
  710 FORMAT (6X,2F12.3,F18.3,F12.3)
  720 FORMAT (39X,9HWATERLINE,F12.3)
  730 FORMAT (6X,2F12.3)
      END
      SUBROUTINE INDATA
C
      COMMON/IO/INPUT,OUTPUT,BIF,OFF,COF
      INTEGER OUTPUT,BIF,OFF,COF
      COMMON /SIGMA /   NK, SIGMA(24), SIGMAO, ERRO, OM(12)
      COMMON / DRFT12/ DRAFT(6,2),IDRAFT
      COMMON/IOFILE/ OFFIL,BIFIL,COFIL
      COMMON/HEAD/TITLE
      CHARACTER*30 TITLE
      COMMON /U      /    RHO,  G
      COMMON/SHIP/ISTA,LSTA,ISWL,LSWL,TF,TA,XCG,YCG,DISPL
      COMMON/GEOMETRY/MSTA,LPTS(25),YOFF(25,25),NAFT,XAFT(25),
     *       YAFT(25),NFWD,XFWD(25),YFWD(25),XOFF(25),
     *       ZOFF(25,25),XFPERP,XAPERP,SHIPL,SHIPB,SHIPT,
     *       Y1(21,25),ZWL(25),WL(25),INPTS(25),XWLF,XWLA,XXF,
     *       XXA,TAN,NON,NOE,NWL,CR,XXFWD,XXSTA,XXAFT,DX,DX1,
```

```
     *       DX2,Z2(21),Y2(21),ZZ(20),YY(20),SNE(20),CSE(20),
     *       DEL(20),ROL(20),ADJUST,WMAX,YMAX,ZMAX,AREA,VERT
      LOGICAL ADJUST,WL
C
      CHARACTER*81 CARDID
      DATA  DEGREE/0.01745 32925 19943/
      DATA  G     /32.17/
      DATA NWL/1/
C
      WRITE(OUTPUT,197)
C *** TITLE
      READ (BIF,199) TITLE
      WRITE(OUTPUT,196) TITLE
C *** DRAFT (fwd),DRAFT (aft), long. loc's of DRAFT marks
      READ (BIF,200) TF,TA,XFPERP,XAPERP
      WRITE(OUTPUT,200) TF,TA,XFPERP,XAPERP
C *** Center of Gravity (XCG aft of FP, YCG above BL)
      READ (BIF,200) XCG,YCG,ZCG
      WRITE(OUTPUT,200) XCG,YCG,ZCG
C *** Six DRAFTs at which hydro. coeffs are computed
      READ(BIF,200) (DRAFT(I,1),I=1,6)
      WRITE(OUTPUT,200) (DRAFT(I,1),I=1,6)
C *** Minimum segment lengths for Frank Close Fit
      READ (BIF,201) YMAX,ZMAX,WMAX,NWL
      WRITE(OUTPUT,201) YMAX,ZMAX,WMAX,NWL
      ADJUST=ZMAX.GT.0.0.AND.YMAX.GT.0.0
C *** Number of forware profile points
      READ (BIF,190) NFWD
      WRITE(OUTPUT,190) NFWD
      IF (NFWD.GT.25) CALL ERROR(15,IDUM,RDUM)
C *** Coordinates of forward profile points
      IF (NFWD.GT.0) READ (BIF,430) (YFWD(I),XFWD(I),I=1,NFWD)
      WRITE(OUTPUT,430) (YFWD(I),XFWD(I),I=1,NFWD)
C *** Number of aft profile points
      READ (BIF,190) NAFT
      WRITE(OUTPUT,190) NAFT
      IF (NAFT.GT.25) CALL ERROR(16,IDUM,RDUM)
C *** Coordinates of aft profile points
      IF (NAFT.GT.0) READ (BIF,430) (YAFT(I),XAFT(I),I=1,NAFT)
      WRITE(OUTPUT,430) (YAFT(I),XAFT(I),I=1,NAFT)
C
C
C----------------------------------------------------------------
C
C     Section 2.0 - READ OFFSET file
C     The offset file can be an actual SHCP DATA File
C
      WRITE(OUTPUT,198)
C
C *** CARD TYPE A
      READ (OFF,410) CARDID
      WRITE(OUTPUT,410) CARDID
C *** CARD TYPE B
      READ (OFF,410)
C *** CARD TYPE C
      READ (OFF,412) SPACE,ZSCAL,YSCAL,SHIPL,NAPN,KINDO
      WRITE(OUTPUT,412)SPACE,ZSCAL,YSCAL,SHIPL,NAPN,KINDO
      IF (SPACE.EQ.0.0) SPACE=1.0
      ZSCAL=1.0
      YSCAL=1.0
      MSTA=0
```

```
      NFWD=0
      NAFT=0
C
C *** CARD TYPE D
   30 N=1
      MSTA=MSTA+1
      IF (MSTA.GT.25) CALL ERROR(10,IDUM,RDUM)
      READ (OFF,416) STATNO,Y11,Z1,JTEST
      WRITE(OUTPUT,417) STATNO,Y11,Z1,JTEST
      XOFF(MSTA)=STATNO*SPACE
      GO TO 50
   40 CONTINUE                   !loop within each station
      N=N+1
      IF (N.GT.25) CALL ERROR(11,MSTA,RDUM)
      READ (OFF,416) S,Y11,Z1,JTEST
      WRITE(OUTPUT,417) S,Y11,Z1,JTEST
      IF (S.NE.STATNO) CALL ERROR(12,MSTA,RDUM)
   50 YOFF(N,MSTA)=Z1*ZSCAL
      ZOFF(N,MSTA)=Y11*YSCAL
      IF (JTEST.EQ.0 .OR. JTEST.EQ.77777) GO TO 40
      LPTS(MSTA)=N                           !No. of points- MSTA
      IF (N.LT.2) CALL ERROR(13,MSTA,RDUM)
      IF (JTEST.EQ.88888) GO TO 30      !Go onto next station
      IF (JTEST.NE.99999) CALL ERROR(14,JTEST,RDUM)
C
C
      DO 220 J=1,MSTA
         XOFF(J)=-XOFF(J)
  220 CONTINUE
      IF (NFWD.EQ.0) GO TO 240
      X=XOFF(1)
      DO 230 I=1,NFWD
         XFWD(I)=XFWD(I)+X
  230 CONTINUE
  240 CONTINUE
      X=XOFF(MSTA)
      DO 250 I=1,NAFT
         XAFT(I)=X-XAFT(I)
  250 CONTINUE
      RETURN
  180 FORMAT (5X,I5)
  190 FORMAT (I5)
  197 FORMAT(1H1/,81(1H*)/,' INPUT DATA ECHO ',T64,
     *'PROGRAM HYDREX'/,81(1H*)//,33(1H-),
     *'[BIF] DATA FILE',32(1H-)/)
  198 FORMAT(1H1/,81(1H*)/,' INPUT DATA ECHO ',T64,
     *'PROGRAM HYDREX'/,81(1H*)//,33(1H-),
     *'[OFF] DATA FILE',32(1H-)/)
  196 FORMAT(1X,A)
  199 FORMAT(A)
  200 FORMAT (6F10.2)
  201 FORMAT (3F10.2,I5)
  210 FORMAT (F10.2,I5,5X,F10.2)
C
  410 FORMAT (A)
  412 FORMAT (4F10.3,13X,I2,4X,I1)
  414 FORMAT (5X,I5,5X,'INPUT OF SHCP TYPE D OFFSET DATA')
  416 FORMAT (F6.3,2F7.0,I6)
  417 FORMAT (F7.3,2F10.2,I6)
  420 FORMAT (2I5,F10.2)
  430 FORMAT (2F10.2)
```

```
C
      END
      SUBROUTINE INSERT(A1,A2,J1,L1,L2)
C     Purpose: Inserts array A2 into array A1 at location J1.
      REAL  A1(1), A2(1)
      IF (L1.LT.J1) GO TO 120
      M=L1+L2
      I=L1
      K=L1-J1+1
      DO 110 J=1,K
         A1(M)=A1(I)
         M=M-1
         I=I-1
  110 CONTINUE
  120 I=J1-1
      DO 130 K=1,L2
         M=K+I
         A1(M)=A2(K)
  130 CONTINUE
      RETURN
      END
      SUBROUTINE STATN (Z,Y,ZW,NPTS)
C
C     CALCULATION OF DATA CONCERNING STATION GEOMETRY.
C     REVISION OF OFFSETS FOR GOOD RESULTS MAY BE PERFORMED.
C
C
C     NON = NUMBER OF CALCULATED MIDPOINTS.
C     NWL = NUMBER OF WATERLINE MIDPOINTS.
C     Z   = HORIZONTAL COORDINATE OF SEGMENT ENDPOINT.
C     Y   = VERTICAL COORDINATE OF SEGMENT ENDPOINT.
C     ZZ  = HORIZONTAL COORDINATE OF SEGMENT MIDPOINT.
C     YY  = VERTICAL COORDINATE OF SEGMENT MIDPOINT.
C     SNE = HORIZONTAL COMPONENT OF UNIT NORMAL TO SEGMENT.
C     CSE = VERTICAL COMPONENT OF UNIT NORMAL.
C     DEL = LENGTH OF SEGMENT.
C     ROL = MOMENT OF UNIT NORMAL ABOUT CENTER OF ROLL (CG).
C
      REAL Y(1),Z(1)
      COMMON/IO/INPUT,OUTPUT,BIF,OFF,COF
      INTEGER OUTPUT,BIF,OFF,COF
      COMMON/GEOMETRY/MSTA,LPTS(25),YOFF(25,25),NAFT,XAFT(25),
     *       YAFT(25),NFWD,XFWD(25),YFWD(25),XOFF(25),
     *       ZOFF(25,25),XFPERP,XAPERP,SHIPL,SHIPB,SHIPT,
     *       Y1(21,25),ZWL(25),WL(25),INPTS(25),XWLF,XWLA,XXF,
     *       XXA,TAN,NON,NOE,NWL,CR,XXFWD,XXSTA,XXAFT,DX,DX1,
     *       DX2,Z2(21),Y2(21),ZZ(20),YY(20),SNE(20),CSE(20),
     *       DEL(20),ROL(20),ADJUST,WMAX,YMAX,ZMAX,AREA,VERT
      COMMON /      / HA1, SA1, RA1, CA1, HV1, SV1, RV1, CV1,
     1                RHO2, RSIG, WN, W1, W2, ERR, XRI, YRI, EJT
      COMMON /      /   HAO(24), SAO(24), RAO(24), CAO(24),
     1   HVO(24), SVO(24), RVO(24), CVO(24)
      COMMON // ZNEW(20),YNEW(20),ZZNEW(20),YYNEW(20),CNEW(20),SNEW(20),
     1DNEW(20),RNEW(20),MORE(20),I,K,ZO,YO,M1,MTOT,ZINT,YINT,NUMBER,M,N1
     2,I1,JOB,ZS,YS,D,C,S,P,Q,NUT
      LOGICAL ADJUST
      DATA LIMIT/20/
      DATA MAXPTS/20/
      AREA=0.0
      VERT=0.0
      ZO=0.0
      YO=Y(1)
```

```
      MTOT=0
      K=1
      I1=K
      IF (Z(1).EQ.0.0) I1=2
      IF (I1.GT.NPTS) GO TO 130
C
C     CALCULATION LOOP FOR SUBMERGED OFFSET POINTS..
C
      DO 120 I=I1,NPTS
         ZINT=Z(I)-Z0
         YINT=Y(I)-Y0
         D=SQRT(ZINT*ZINT+YINT*YINT)
         IF (D.EQ.0.0) GO TO 120
         IF (.NOT.ADJUST) GO TO 110
C
C        CODE INSTRUCTIONS FOR THE ADDITION OF POINTS..
C
         NUMBER=MAX0(IABS(IFIX(ZINT/ZMAX)),IABS(IFIX(YINT/YMAX)))
         MORE(K)=NUMBER
         MTOT=MTOT+NUMBER
C
C        CONTINUE STANDARD PROCEDURE..
C
  110    ZS=Z0+Z(I)
         YS=Y0+Y(I)
         AREA=AREA+YINT*ZS
         VERT=VERT+YINT*(Z0*(Y0-CR+YINT/3.0)+Z(I)*(Y0-CR+YINT/1.5))
         C=ZINT/D
         S=YINT/D
         CSE(K)=C
         SNE(K)=S
         DEL(K)=D
         ZZ(K)=0.5*ZS
         YY(K)=0.5*YS
         ROL(K)=(CR-YY(K))*S-ZZ(K)*C
         Z2(K)=Z0
         Y2(K)=Y0
         Z0=Z(I)
         Y0=Y(I)
         K=K+1
  120 CONTINUE
C
C     END OF CALCULATION LOOP FOR SUBMERGED POINTS.
C
  130 Z2(K)=Z0
      Y2(K)=Y0
      NON=K-1
C
C     ADD UPPERMOST SEGMENT..
C
      IF (NWL.NE.0) GO TO 150
C        SECTION IS SUBMERGED.
      IF (Z0.EQ.0.0) GO TO 200
      IF (.NOT.ADJUST) GO TO 140
C
C     CODE INSTRUCTIONS FOR THE ADDITION OF POINTS..
C
      NUMBER=IABS(IFIX(Z0/ZMAX))
      MORE(K)=NUMBER
      MTOT=MTOT+NUMBER
C
```

```
C     CONTINUE STANDARD PROCEDURE..
C
  140 DEL(K)=ABS(ZO)
      CSE(K)=-1.0
      SNE(K)=0.0
      ZZ(K)=0.5*ZO
      YY(K)=YO
      ROL(K)=ZZ(K)
      NON=K
      K=K+1
      Z2(K)=0.0
      Y2(K)=YO
      GO TO 200
C
C     ADD SEGMENT UP TO WATERLINE..
C
  150 ZINT=ZW-ZO
      YINT=-YO
      IF (NWL.LT.0) NWL=0
      IF (ZW.LE.0.0) NWL=0
      D=SQRT(ZINT*ZINT+YINT*YINT)
      IF (D.EQ.0.0) GO TO 170
      IF (.NOT.ADJUST) GO TO 160
C
C     CODE INSTRUCTIONS FOR THE ADDITION OF POINTS..
C
      NUMBER=MAX0(IABS(IFIX(ZINT/ZMAX)),IABS(IFIX(YINT/YMAX)))
      MORE(K)=NUMBER
      MTOT=MTOT+NUMBER
C
C     CONTINUE STANDARD PROCEDURE..
C
  160 ZS=ZO+ZW
      YS=YO
      AREA=AREA+YINT*ZS
      VERT=VERT-YINT*(ZO*(CR+YINT/1.5)+ZW*(CR+YINT/3.0))
      C=ZINT/D
      S=YINT/D
      CSE(K)=C
      SNE(K)=S
      DEL(K)=D
      ZZ(K)=0.5*ZS
      YY(K)=0.5*YS
      ROL(K)=(CR-YY(K))*S-ZZ(K)*C
      NON=K
      K=K+1
      Z2(K)=ZW
      Y2(K)=0.0
C
C     ADD DECK AT WATERLINE..
C
  170 CONTINUE
      IF (NWL.EQ.0) GO TO 200
      ZINT=ZW/FLOAT(NWL)
      IF (WMAX.EQ.0.0) GO TO 180
      IF (ZINT.LT.WMAX) GO TO 180
      NWL=IFIX(ZW/WMAX)+1
      ZINT=ZW/FLOAT(NWL)
  180 D=ZINT*0.5
      DO 190 I=1,NWL
         DEL(K)=ZINT
```

```
C
        SUBROUTINE POTST
        COMMON/BD/XPAN(120),YPAN(120),ZPAN(120),AREA(120),ST(120),
     *  ACN(120),ACNW(120),AN(120,3),E(120),P(120,6),PRFS(120),
     *   STOLD(120),PX(120,6)
        COMMON/BD2/XPT(150),YPT(150),ZPT(150),WRF(150),WRFR(150),
     *   KK(150,4)
        COMMON/A/NPAN,NPT,GEE,RHO,NKX,NKY,EYE,DT,TIM,UFWD
        COMPLEX A,B,EYE
        DIMENSION XPSL(3,4),XPSLR(3,4),PBB(120,120)
        COMMON/PTST/ARE4(200,4),X4(200,4),Y4(200,4),Z4(200,4)
     *  ,SEL(200,4)
        DO 1500 J=1,NPAN
        ARE4(J,4)=-1.0
        JT=4
        IF(KK(J,4).EQ.0) JT=3
        DO 1500 JJ=1,JT
        J2=1
        IF(JJ.LT.JT) J2=JJ+1
        KF=KK(J,JJ)
        KG=KK(J,J2)
        X4(J,JJ)=(XPT(KF)+XPT(KG)+XPAN(J))/3.0
        Y4(J,JJ)=(YPT(KF)+YPT(KG)+YPAN(J))/3.0
        Z4(J,JJ)=(ZPT(KF)+ZPT(KG)+ZPAN(J))/3.0
        AF=XPT(KF)-XPAN(J)
        BF=YPT(KF)-YPAN(J)
        CF=ZPT(KF)-ZPAN(J)
        AG=XPT(KG)-XPAN(J)
        BG=YPT(KG)-YPAN(J)
        CG=ZPT(KG)-ZPAN(J)
        CALL SELF(AF,BF,CF,AG,BG,CG,FEE)
        SEL(J,JJ)=FEE
        CR=AF*BG-BF*AG
        AR=BF*CG-CF*BG
        BR=CF*AG-AF*CG
        ARE4(J,JJ)=0.5*SQRT(AR*AR+BR*BR+CR*CR)
1500    CONTINUE
        DO 127 NJ=1,NPAN
        DO 1277 MJ=1,NPAN
1277    PBB(NJ,MJ)=0.00
        P(NJ,1)=0.00
        P(NJ,2)=0.00
        P(NJ,3)=0.00
        P(NJ,4)=0.00
        P(NJ,5)=0.00
        P(NJ,6)=0.00
        DO 128 NK=1,4
        ARN=ARE4(NJ,NK)
        IF(ARN.LT.0.0) GO TO 128
        P1=0.0
        P2=0.0
        P3=0.0
        P4=0.0
        P5=0.0
        P6=0.0
        X=X4(NJ,NK)
        Y=Y4(NJ,NK)
        Z=Z4(NJ,NK)
        DO 138 MJ=1,NPAN
        DO 138 MK=1,4
        XF=X4(MJ,MK)
        YF=Y4(MJ,MK)
        ZF=Z4(MJ,MK)
        ARM=ARE4(MJ,MK)
        IF(ARM.LT.0.00) GO TO 138
        IF(NJ.NE.MJ) GO TO 140
        IF(MK.NE.NK) GO TO 140
        FRA=SEL(MJ,MK)/ARM
        GO TO 1380
```

```
         CSE(K)=-1.0
         SNE(K)=0.0
         YY(K)=0.0
         ZZ(K)=Z2(K)-D
         ROL(K)=ZZ(K)
         NON=K
         K=K+1
         Y2(K)=0.0
         Z2(K)=Z2(K-1)-ZINT
  190 CONTINUE
      Z2(K)=0.0
      NON=K-1
C
C     END OF FIRST PASS.  ADD ADDITIONAL SEGMENTS IF REQUIRED..
C
  200 IF (NON.GT.LIMIT) GO TO 290
      IF (.NOT.ADJUST) GO TO 280
      IF (NON.GT.MAXPTS) GO TO 280
      IF (MTOT.EQ.0) GO TO 280
      MTOT=MTOT+NON
      M1=NON-NWL
      IF (MTOT.LE.MAXPTS) GO TO 230
C
C     DECREASE  MORE  UNTIL  MTOT  IS EQUAL  MAXPTS ..
C
  210 DO 220 K=1,M1
         IF (MORE(K).LE.0) GO TO 220
         MORE(K)=MORE(K)-1
         MTOT=MTOT-1
         IF (MTOT.LE.MAXPTS) GO TO 230
  220 CONTINUE
      GO TO 210
C
C     INSERT ADDITIONAL SEGMENTS AS INDICATED BY  MORE ..
C
  230 I=1
      DO 270 M=1,M1
         I1=I+1
         NUMBER=MORE(M)
         IF (NUMBER.LE.0) GO TO 260
         N1=NUMBER+1
         Z0=Z2(I)
         Y0=Y2(I)
         ZINT=Z2(I1)-Z0
         YINT=Y2(I1)-Y0
         ZINT=ZINT/FLOAT(2*N1)
         YINT=YINT/FLOAT(2*N1)
         D=DEL(I)/FLOAT(N1)
         C=CSE(I)
         S=SNE(I)
         DO 240 K=1,N1
            P=(K-1)*2
            Q=P+1.0
            ZNEW(K)=Z0+P*ZINT
            YNEW(K)=Y0+P*YINT
            ZZNEW(K)=Z0+Q*ZINT
            YYNEW(K)=Y0+Q*YINT
            RNEW(K)=(CR-YYNEW(K))*S-ZZNEW(K)*C
  240    CONTINUE
         DO 250 K=1,NUMBER
            DNEW(K)=D
```

```
          CNEW(K)=C
  250     SNEW(K)=S
          ZZ(I)=ZZNEW(N1)
          YY(I)=YYNEW(N1)
          ROL(I)=RNEW(N1)
          DEL(I)=D
          NUT=NON+1
          CALL  INSERT  (Z2, ZNEW(2), I1, NUT, NUMBER)
          CALL  INSERT  (Y2, YNEW(2), I1, NUT, NUMBER)
          CALL  INSERT  (ZZ, ZZNEW, I, NON, NUMBER)
          CALL  INSERT  (YY, YYNEW, I, NON, NUMBER)
          CALL  INSERT  (DEL, DNEW, I, NON, NUMBER)
          CALL  INSERT  (ROL, RNEW, I, NON, NUMBER)
          CALL  INSERT  (CSE, CNEW, I, NON, NUMBER)
          CALL  INSERT  (SNE, SNEW, I, NON, NUMBER)
  260     I=I1+NUMBER
          NON=NON+NUMBER
  270 CONTINUE
  280 NOE=NON+NON
C
      RETURN
C
C     ERROR DIAGNOSTICS..
C
C     TOO MANY WET SEGMENTS..
C
  290 WRITE (OUTPUT,300) XXSTA
      STOP 7
C
  300 FORMAT(' More than 20 wet segments for station at X=',F13.5)
C
      END
      SUBROUTINE FLOAT
C
C     THIS SUBROUTINE APPLIES THE GIVEN DRAFT TO THE ORIGINAL TABLE
C        OF OFFSETS.
      COMMON/IO/INPUT,OUTPUT,BIF,OFF,COF
      INTEGER OUTPUT,BIF,OFF,COF
C
      COMMON/SHIP/ISTA,LSTA,ISWL,LSWL,TF,TA,XCG,YCG,DISPL
       COMMON/DRFT12/ DRAFT(6,2),IDRAFT
      COMMON/GEOMETRY/MSTA,LPTS(25),YOFF(25,25),NAFT,XAFT(25),
     *         YAFT(25),NFWD,XFWD(25),YFWD(25),XOFF(25),
     *         ZOFF(25,25),XFPERP,XAPERP,SHIPL,SHIPB,SHIPT,
     *         Y1(21,25),ZWL(25),WL(25),INPTS(25),XWLF,XWLA,XXF,
     *         XXA,TAN,NON,NOE,NWL,CR,XXFWD,XXSTA,XXAFT,DX,DX1,
     *         DX2,Z2(21),Y2(21),ZZ(20),YY(20),SNE(20),CSE(20),
     *         DEL(20),ROL(20),ADJUST,WMAX,YMAX,ZMAX,AREA,VERT
      LOGICAL WL, ADJUST
C
C     PLACE SHIP AT GIVEN DRAFT..
C
C-----------------------------------------------------------
C
      TF=DRAFT(IDRAFT,1)
      TA=DRAFT(IDRAFT,2)
C
C-----------------------------------------------------------
C
C
      IF (XFPERP.NE.XAPERP) GO TO 130
```

```
      XFPERP=XOFF(1)
      XAPERP=XOFF(MSTA)
      IF (XFPERP.NE.XAPERP) GO TO 130
      TAN=0.0
      IF (TF.EQ.TA) GO TO 140
      WRITE(OUTPUT,570) XFPERP,TF,TA
      STOP 5
  130 TAN=(TA-TF)/(XFPERP-XAPERP)
  140 ISTA=0
      ISWL=0
      LSWL=0
      XX=XOFF(1)
      DO 190 J=1,MSTA
         IF (XX.GE.XOFF(J)) GO TO 142
         WRITE (OUTPUT,575) J,XOFF(J),XX
         STOP 6
C
C        TJ IS DRAFT OF STATION J.
C
  142    XX=XOFF(J)
         TJ=(XFPERP-XX)*TAN+TF
         YNEXT=YOFF(1,J)-TJ
         N=0
         IF (YNEXT.GT.0.0) GO TO 152
         LSTA=J
         IF (ISTA.EQ.0) ISTA=J
         Y1(1,J)=YNEXT
         N=LPTS(J)
         DO 150 I=2,N
            YNEXT=YOFF(I,J)-TJ
            IF (YNEXT.GT.0.0) GO TO 160
            Y1(I,J)=YNEXT
  150    CONTINUE
         I=N+1
         IF (YNEXT.EQ.0.0) GO TO 170
C
C        SECTION IS NOT SURFACE PIERCING..
C
  152    INPTS(J)=N
         WL(J)=.FALSE.
         GO TO 190
C
C        SECTION IS SURFACE PIERCING.  FIND WATERLINE COORDINATES..
C
  160    Y0=YOFF(I-1,J)-TJ
         IF (Y0.EQ.0.0) GO TO 170
         INPTS(J)=I-1
         Z0=ZOFF(I-1,J)
         ZWL(J)=Z0-Y0*(ZOFF(I,J)-Z0)/(YNEXT-Y0)
         GO TO 180
C
  170    INPTS(J)=I-2
         ZWL(J)=ZOFF(I-1,J)
C
  180    WL(J)=.TRUE.
         LSWL=J
         IF (ISWL.EQ.0) ISWL=J
  190 CONTINUE
      IF (ISTA.NE.0) GO TO 200
      WRITE (OUTPUT,580)
      STOP 77
```

```
C
C     FIND FORWARD AND AFTER ENDS OF WETTED HULL..
C
  200 XWLA=1.0E32
      XWLF=-XWLA
      IF (ISWL.EQ.0) GO TO 210
      XWLF=XOFF(ISWL)
      XWLA=XOFF(LSWL)
  210 XXF=XOFF(ISTA)
      IF (NFWD.EQ.0) GO TO 250
      IF (NFWD.GT.1) GO TO 220
C
C     XFWD(1) DEFINED AS FORWARD END OF WATERLINE..
C
      XWLF=AMAX1(XFWD(1),XWLF)
      XXF=AMAX1(XWLF,XXF)
      GO TO 250
C
C     FIND FORWARD END OF WATERLINE AND FORWARD END OF WETTED HULL..
C
  220 YO=(XFWD(1)-XFPERP)*TAN+YFWD(1)-TF
      DO 230 J=2,NFWD
         XX=XFWD(J)
         YNEXT=(XX-XFPERP)*TAN+YFWD(J)-TF
         IF (YNEXT.LT.0.0) GO TO 228
         IF (YO.GT.0.0) GO TO 229
         IF (YNEXT.EQ.0.0) GO TO 227
C
C        INTERPOLATE FOR WATERLINE..
C
         XO=XFWD(J-1)
         XX=(XX-XO)/(YO-YNEXT)*YO+XO
  227    IF (XX.LE.XWLF) GO TO 229
         XWLF=XX
C
  228    XXF=AMAX1(XX,XXF)
  229    YO=YNEXT
  230 CONTINUE
C
C     FIND AFTER END OF WATERLINE AND AFTER END OF WETTED HULL..
C
  250 XXA=XOFF(LSTA)
      IF (NAFT.EQ.0) GO TO 290
      IF (NAFT.GT.1) GO TO 260
C
C     XAFT(1) DEFINED AS AFTER END OF WATER LINE..
C
      XWLA=AMIN1(XAFT(1),XWLA)
      XXA=AMIN1(XWLA,XXA)
      GO TO 290
C
C     FIND AFTER END OF WATERLINE AND AFTER END OF WETTED HULL..
C
  260 YO=(XAFT(1)-XFPERP)*TAN+YAFT(1)-TF
      DO 270 J=2,NAFT
         XX=XAFT(J)
         YNEXT=(XX-XFPERP)*TAN+YAFT(J)-TF
         IF (YNEXT.LT.0.0) GO TO 268
         IF (YO.GT.0.0) GO TO 268
         IF (YNEXT.EQ.0.0) GO TO 267
C
```

```
C      INTERPOLATE FOR WATERLINE..
C
       XO=XAFT(J-1)
       XX=(XX-XO)/(YO-YNEXT)*YO+XO
 267   IF (XX.GE.XWLA) GO TO 269
       XWLA=XX
C
 268   XXA=AMIN1(XX,XXA)
 269   YO=YNEXT
 270 CONTINUE
 290 IF (ISWL.EQ.0) ISWL=26
C
 420 FORMAT (32HOS T A T I O N   G E O M E T R Y)
 480 FORMAT (22HO     DRAFT FWD (AT X =,F10.3,3H) =,F10.3/5X,17HDRAFT AF
    1T (AT X =,F10.3,3H) =,F10.3)
 570 FORMAT (61HO*** TWO DRAFTS SPECIFIED, AND LENGTH BETWEEN PERPS. IS
    1 ZERO./31H *** BOTH PERPENDICULARS AT X =,F12.4/16H *** DRAFT FWD
    2=,F12.4/16H *** DRAFT AFT =,F12.4)
 575 FORMAT (12HO*** STATION,I3,5H (X =,F12.4,18H) IS OUT OF ORDER./29H
    1 *** PREVIOUS STATION HAS X =,F12.4)
 580 FORMAT (25HO*** SHIP IS ABOVE WATER.)
C
     END
     SUBROUTINE ERROR(NO,IDUM,RDUM)
     COMMON/IO/INPUT,OUTPUT,BIF,OFF,COF
     INTEGER OUTPUT,BIF,OFF,COF
     WRITE(OUTPUT,10) NO
     WRITE(OUTPUT,11) IDUM,RDUM
 10  FORMAT(' STOPPED DUE TO ERROR NO.  ',I2,//)
 11  FORMAT(1X,I3,5X,F10.3)
     STOP
     END
     SUBROUTINE BEER (K)
C
C1   TWO-DIMENSIONAL HYDRODYNAMIC CALCULATION FOR THE SPECIAL CASE
C1   OF ZERO OR INFINITE FREQUENCY.
C1
C5
     COMMON/GEOMETRY/MSTA,LPTS(25),YOFF(25,25),NAFT,XAFT(25),
    *         YAFT(25),NFWD,XFWD(25),YFWD(25),XOFF(25),
    *         ZOFF(25,25),XFPERP,XAPERP,SHIPL,SHIPB,SHIPT,
    *         Y1(21,25),ZWL(25),WL(25),INPTS(25),XWLF,XWLA,XXF,
    *         XXA,TAN,NON,NOE,NWL,CR,XXFWD,XXSTA,XXAFT,DX,DX1,
    *         DX2,Z2(21),Y2(21),ZZ(20),YY(20),SNE(20),CSE(20),
    *         DEL(20),ROL(20),ADJUST,WMAX,YMAX,ZMAX,AREA,VERT
     COMMON // HAH,SAS,RAR,CCA,HVH,SVS,RVR,CCV,RHO2,RSIG,WN,W1,W2,ERR,Z
    1RI,YRI,EJT
     COMMON /     /   HAO(24,8),BLOGP(20,20),YLOGP(20,20),BLOGM(20
    1,20),YLOGM(20,20),CONH(40),CONR(40,2),CROLL(40,40),CHEAV(40,40),HE
    2AVI(20,20),HEAVT(20,20),ROLLI(20,20),ROLLT(20,20),EJI,CZRI,CZLI,SZ
    3RI,SZLI,RARI,RALI,RBRI,RBLI,CLI,CRI,SLI,SRI,I,IPESO,J,NJ
C5
C5   INPUT AND OUTPUT LOGICAL UNITS..
     COMMON/IO/INPUT,OUTPUT,BIF,OFF,COF
     INTEGER OUTPUT,BIF,OFF,COF
C5
C5   OUTPUT LISTING PAGE HEADING DATA..
C5
     HAH=0.0
     SAS=0.0
     RAR=0.0
```

```
      CCA=0. 0
      HVH=0. 0
      SVS=0. 0
      RVR=0. 0
      CCV=0. 0
      NN=NON-NWL
C3
C-----------------------------------------------------
C
       IF (WN. NE. 0. ) GO TO 130
C3    ZERO FREQUENCY CASE. .
C3
      DO 120 I=1, NN
         XM1=ZZ(I)-Z2(1)
         XP1=ZZ(I)+Z2(1)
         YP1=YY(I)+Y2(1)
         FCR1=. 5*ALOG(XM1**2+YP1**2)
         FCL1=. 5*ALOG(XP1**2+YP1**2)
         ACR1=ATAN2(YP1, XM1)
         ACL1=ATAN2(YP1, XP1)
         DO 110 J=1, NN
            XM2=ZZ(I)-Z2(J+1)
            XP2=ZZ(I)+Z2(J+1)
            YP2=YY(I)+Y2(J+1)
            FCR2=. 5*ALOG(XM2**2+YP2**2)
            FCL2=. 5*ALOG(XP2**2+YP2**2)
            ACR2=ATAN2(YP2, XM2)
            ACL2=ATAN2(YP2, XP2)
            SIMJ=SNE(I)*CSE(J)-SNE(J)*CSE(I)
            CIMJ=CSE(I)*CSE(J)+SNE(I)*SNE(J)
            SIPJ=SNE(I)*CSE(J)+SNE(J)*CSE(I)
            CIPJ=CSE(I)*CSE(J)-SNE(I)*SNE(J)
            DCNR=SIPJ*(FCR1-FCR2)+CIPJ*(ACR1-ACR2)
            PCR=CSE(J)*(XM1*FCR1-YP1*ACR1-XM1-XM2*FCR2+YP2*ACR2+XM2)+SNE
     1      (J)*(YP2*FCR2+XM2*ACR2+YP1-YP1*FCR1-XM1*ACR1-YP2)
            DCNL=SIMJ*(FCL2-FCL1)+CIMJ*(ACL2-ACL1)
            PCL=CSE(J)*(XP2*FCL2-YP2*ACL2-XP2-XP1*FCL1+YP1*ACL1+XP1)+SNE
     1      (J)*(YP2*FCL2+XP2*ACL2-YP2-YP1*FCL1-XP1*ACL1+YP1)
            CROLL(I, J)=BLOGM(J, I)+2. 0*(DCNR-DCNL)
            CHEAV(I, J)=BLOGP(J, I)+2. 0*(DCNR+DCNL)
            ROLLT(I, J)=-YLOGM(J, I)-2. 0*(PCR-PCL)
            HEAVT(I, J)=-YLOGP(J, I)-2. 0*(PCR+PCL)
            IF (J. EQ. NN) GO TO 110
            XM1=XM2
            XP1=XP2
            YP1=YP2
            FCR1=FCR2
            FCL1=FCL2
            ACR1=ACR2
            ACL1=ACL2
  110    CONTINUE
  120 CONTINUE
      GO TO 160
C3
C3    INFINITE FREQUENCY CASE. .
C3
  130 CONTINUE
      DO 150 I=1, NN
         DO 140 J=1, NN
            CROLL(I, J)=BLOGM(J, I)
            CHEAV(I, J)=BLOGP(J, I)
```

```
            ROLLLT(I,J)=-YLOGM(J,I)
            HEAVT(I,J)=-YLOGP(J,I)
  140    CONTINUE
  150 CONTINUE
C3
C3    SOLUTION FOR EITHER THE ZERO OR INFINITE FREQUENCY CASE..
C3
  160 CONTINUE
      DO 170 I=1,NN
         CONH(I)=CSE(I)
         CONR(I,1)=-SNE(I)
         CONR(I,2)=ROL(I)
  170 CONTINUE
      IT=LNEGT(40,NN,1,CHEAV,CONH,ERR,HEAVI)
      IF (IT.EQ.0) GO TO 180
      IF (WN.EQ.0.0) WRITE (OUTPUT,230)
      IF (WN.NE.0.0) WRITE (OUTPUT,240)
      WRITE (OUTPUT,270) XXSTA
      IF (IT.NE.0) GO TO 190
  180 CONTINUE
      IT=LNEGT(40,NN,2,CROLL,CONR,ERR,ROLLI)
      IF (IT.EQ.0) GO TO 200
      IF (WN.EQ.0.0) WRITE (OUTPUT,250)
      IF (WN.NE.0.0) WRITE (OUTPUT,260)
      WRITE (OUTPUT,270) XXSTA
      IF (IT.EQ.1) GO TO 200
  190 WRITE (OUTPUT,280)
      STOP 10
C3
C3    EVALUATE VELOCITY POTENTIALS AND FORCE COEFFICIENTS..
C3
  200 DO 220 I=1,NN
         PAH=0.0
         PAS=0.0
         PAR=0.0
         DO 210 J=1,NN
            PAH=PAH+CONH(J)*HEAVT(J,I)
            PAS=PAS+CONR(J,1)*ROLLLT(J,I)
            PAR=PAR+CONR(J,2)*ROLLLT(J,I)
  210    CONTINUE
C6
C6    THE PRESSURES IN PHASE WITH THE SINUSOIDAL DISPLACEMENT ARE..
C6
C6    HEAVE -- PAH = PAH*RHO*ESIG*ESIG
C6    SWAY  -- PAS = PAS*RHO*ESIG*ESIG
C6    ROLL  -- PAR = PAR*RHO*ESIG*ESIG
C6
C6       THE ACCELERATION COMPONENTS OF THE FORCE ARE EQUAL
C6       IN MAGNITUDE TO THE ABOVE, BUT HAVE THE OPPOSITE SIGN.
C6
         DDD=DEL(I)
         DCI=CSE(I)*DDD
         DSI=-SNE(I)*DDD
         DFR=ROL(I)*DDD
C6
C6       INTEGRATION TO OBTAIN FORCE ACCELERATION COEFFICIENTS..
C6
         HAH=HAH+PAH*DCI
         SAS=SAS+PAS*DSI
         RAR=RAR+PAR*DFR
         CCA=PAR*DSI+PAS*DFR+CCA
```

```
C6
  220 CONTINUE
      HAH=HAH*RHO2
      SAS=SAS*RHO2
      RAR=RAR*RHO2
      CCA=CCA*RHO2/2.0
      IF(WN.EQ.0.) HAH=99.   ! INFINITE AT ZERO FREQ
      RETURN
C
  230 FORMAT (43H *** HEAVE MATRIX, ZERO ENCOUNTER FREQUENCY)
  240 FORMAT (37H *** HEAVE MATRIX, INFINITE FREQUENCY)
  250 FORMAT (47H *** SWAY-ROLL MATRIX, ZERO ENCOUNTER FREQUENCY)
  260 FORMAT (41H *** SWAY-ROLL MATRIX, INFINITE FREQUENCY)
  270 FORMAT (31H *** COEFFS. FOR STATION AT X =,F13.5)
  280 FORMAT (26H *** EXECUTION TERMINATED.)
C
      END
      SUBROUTINE GIRL
C
C1    CALCULATION OF FREQUENCY INDEPENDENT TERMS TO BE USED IN THE
C1    TWO-DIMENSIONAL HYDRODYNAMIC CALCULATIONS.
C1
C6    THIS SUBROUTINE IS CALLED ONCE FOR EACH STATION OF THE SHIP
C6    WHEN THE HYDRODYNAMIC COEFFICIENTS ARE BEING GENERATED.
C6
C5
      COMMON/GEOMETRY/MSTA,LPTS(25),YOFF(25,25),NAFT,XAFT(25),
     *       YAFT(25),NFWD,XFWD(25),YFWD(25),XOFF(25),
     *       ZOFF(25,25),XFPERP,XAPERP,SHIPL,SHIPB,SHIPT,
     *       Y1(21,25),ZWL(25),WL(25),INPTS(25),XWLF,XWLA,XXF,
     *       XXA,TAN,NON,NOE,NWL,CR,XXFWD,XXSTA,XXAFT,DX,DX1,
     *       DX2,Z2(21),Y2(21),ZZ(20),YY(20),SNE(20),CSE(20),
     *       DEL(20),ROL(20),ADJUST,WMAX,YMAX,ZMAX,AREA,VERT
      COMMON // HA1,SA1,RA1,CA1,HV1,SV1,RV1,CV1,RHO2,RSIG,WN,W1,W2,ERR,Z
     1RI,YRI,EJT
      COMMON /      /   HA0(24,8),BLOGP(20,20),YLOGP(20,20),BLOGM(20,
     120),YLOGM(20,20),I,J,ACLI,ACLT,ACRI,ACRT,APLI,APLT,APRI,APRT,CIMJ,
     2CIPJ,DCNL,DCNR,DPNL,DPNR,FCLI,FCLT,FCRI,FCRT,FPLI,FPLT,FPRI,FPRT,P
     3CL,PCR,PPL,PPR,SIMJ,SIPJ,ZMI,ZMT,ZPI,ZPT,YMI,YMT,YPI,YPT
      DATA PIN/-3.14159265358979/
      DATA TPI/6.28318530717958/
      DO 160 I=1,NON
         ZMI=ZZ(I)
         ZPI=ZMI
         YMI=YY(I)-Y2(1)
         YPI=YY(I)+Y2(1)
         FPRI=ALOG(ZMI*ZMI+YMI*YMI)/2.0
         FPLI=FPRI
         FCRI=ALOG(ZMI*ZMI+YPI*YPI)/2.0
         FCLI=FCRI
         APRI=ATAN2(YMI,ZMI)
         APLI=APRI
         ACRI=ATAN2(YPI,ZMI)
         ACLI=ACRI
         DO 150 J=1,NON
            J1=J+1
            YMT=YY(I)-Y2(J1)
            YPT=YY(I)+Y2(J1)
            ZMT=ZZ(I)-Z2(J1)
            ZPT=ZZ(I)+Z2(J1)
C           CALCULATE ANGLES (MEASURED OUTSIDE SECTION)..
```

```
            APRT=ATAN2(YMT, ZMT)
            IF (ZMT.GE.0.0) GO TO 130
            IF (J1.GT.I) GO TO 110
            IF (YMT.LT.0.0) APRT=APRT+TPI
            GO TO 120
  110       IF (YMT.GE.0.0) APRT=APRT-TPI
  120       IF (YPT.LT.0.0) GO TO 130
            ACRT=PIN
            GO TO 140
  130       ACRT=ATAN2(YPT, ZMT)
  140       ACLT=ATAN2(YPT, ZPT)
            APLT=ATAN2(YMT, ZPT)
            FPRT=ALOG(ZMT*ZMT+YMT*YMT)/2.0
            FPLT=ALOG(ZPT*ZPT+YMT*YMT)/2.0
            FCRT=ALOG(ZMT*ZMT+YPT*YPT)/2.0
            FCLT=ALOG(ZPT*ZPT+YPT*YPT)/2.0
            SIMJ=SNE(I)*CSE(J)-SNE(J)*CSE(I)
            CIMJ=CSE(I)*CSE(J)+SNE(I)*SNE(J)
            SIPJ=SNE(I)*CSE(J)+SNE(J)*CSE(I)
            CIPJ=CSE(I)*CSE(J)-SNE(I)*SNE(J)
            DPNR=SIMJ*(FPRI-FPRT)+CIMJ*(APRI-APRT)
            PPR=CSE(J)*(ZMI*FPRI-YMI*APRI-ZMI-ZMT*FPRT+YMT*APRT+ZMT)+SNE
     1      (J)*(YMI*FPRI+ZMI*APRI-YMI-YMT*FPRT-ZMT*APRT+YMT)
            DPNL=SIPJ*(FPLT-FPLI)+CIPJ*(APLT-APLI)
            PPL=CSE(J)*(ZPT*FPLT-YMT*APLT-ZPT-ZPI*FPLI+YMI*APLI+ZPI)+SNE
     1      (J)*(YMI*FPLI+ZPI*APLI+YMT-YMT*FPLT-ZPT*APLT-YMI)
            DCNR=SIPJ*(FCRI-FCRT)+CIPJ*(ACRI-ACRT)
            PCR=CSE(J)*(ZMI*FCRI-YPI*ACRI-ZMI-ZMT*FCRT+YPT*ACRT+ZMT)+SNE
     1      (J)*(YPT*FCRT+ZMT*ACRT+YPI-YPI*FCRI-ZMI*ACRI-YPT)
            DCNL=SIMJ*(FCLT-FCLI)+CIMJ*(ACLT-ACLI)
            PCL=CSE(J)*(ZPT*FCLT-YPT*ACLT-ZPT-ZPI*FCLI+YPI*ACLI+ZPI)+SNE
     1      (J)*(YPT*FCLT+ZPT*ACLT-YPT-YPI*FCLI-ZPI*ACLI+YPI)
            BLOGP(J, I)=DPNR+DPNL-DCNR-DCNL
            YLOGP(J, I)=PPR+PPL-PCR-PCL
            BLOGM(J, I)=DPNR-DPNL-DCNR+DCNL
            YLOGM(J, I)=PPR-PPL-PCR+PCL
            IF (J.EQ.NON) GO TO 150
            FPRI=FPRT
            FPLI=FPLT
            FCRI=FCRT
            FCLI=FCLT
            APRI=APRT
            APLI=APLT
            ACRI=ACRT
            ACLI=ACLT
            ZMI=ZMT
            YMI=YMT
            ZPI=ZPT
            YPI=YPT
  150    CONTINUE
  160 CONTINUE
      RETURN
C
      END
      SUBROUTINE WINE (K)
C
C1    TWO-DIMENSIONAL HYDRODYNAMIC CALCULATION FOR NON-ZERO FREQUENCIES.
C1
C6    THIS SUBROUTINE IS CALLED FOR EACH STATION AND ALL NON-ZERO
C6    FREQUENCIES WHEN THE HYDRODYNAMIC COEFFICIENTS ARE BEING
C6    GENERATED.
```

```
C6
C5
C5    INPUT AND OUTPUT LOGICAL UNITS..
      COMMON/IO/INPUT,OUTPUT,BIF,OFF,COF
      INTEGER OUTPUT,BIF,OFF,COF
C5
C5
      COMMON/GEOMETRY/MSTA,LPTS(25),YOFF(25,25),NAFT,XAFT(25),
     *       YAFT(25),NFWD,XFWD(25),YFWD(25),XOFF(25),
     *       ZOFF(25,25),XFPERP,XAPERP,SHIPL,SHIPB,SHIPT,
     *       Y1(21,25),ZWL(25),WL(25),INPTS(25),XWLF,XWLA,XXF,
     *       XXA,TAN,NON,NOE,NWL,CR,XXFWD,XXSTA,XXAFT,DX,DX1,
     *       DX2,Z2(21),Y2(21),ZZ(20),YY(20),SNE(20),CSE(20),
     *       DEL(20),ROL(20),ADJUST,WMAX,YMAX,ZMAX,AREA,VERT
      COMMON // HAH,SAS,RAR,CCA,HVH,SVS,RVR,CCV,RHO2,RSIG,WN,W1,W2,ERR,Z
     1RI,YRI,EJT
      COMMON /     /   HAO(24,8),BLOGP(20,20),YLOGP(20,20),BLOGM(20
     1,20),YLOGM(20,20),CONH(40),CONR(40,2),CROLL(40,40),CHEAV(40,40),HE
     2AVI(20,20),HEAVT(20,20),ROLLI(20,20),ROLLT(20,20),EJI,CZRI,CZLI,SZ
     3RI,SZLI,RARI,RALI,RBRI,RBLI,CLI,CRI,SLI,SRI,I,IPESO,NI
      DO 110 I=1,NON
         NI=NON+I
         CONH(I)=0.0
         CONR(I,1)=0.0
         CONR(I,2)=0.0
         CONH(NI)=CSE(I)
         CONR(NI,1)=-SNE(I)
         CONR(NI,2)=ROL(I)
         ZRI=WN*ZZ(I)
         YRI=-WN*(YY(I)+Y2(1))
         EJT=EXP(-YRI)
         EJI=EJT
         CALL ROMEO (CZRI,SZRI,RARI,RBRI,CRI,SRI)
         CZLI=CZRI
         SZLI=SZRI
         RALI=RARI
         RBLI=RBRI
         CLI=CRI
         SLI=SRI
         CALL WOMEN (I,BLOGP(1,I),YLOGP(1,I),BLOGM(1,I),YLOGM(1,I),CHEAV
     1   (1,I),CROLL(1,I),HEAVI(1,I),HEAVT(1,I),ROLLI(1,I),ROLLT(1,I))
  110 CONTINUE
      IF (NWL.EQ.0) GO TO 130
      I=NOE-NWL+1
      DO 120 I=I,NOE
         CONH(I)=0.0
         CONR(I,1)=0.0
         CONR(I,2)=0.0
  120 CONTINUE
  130 IT=JULIET(1,CHEAV,CONH)
      IF (IT.EQ.0) GO TO 140
      WRITE (OUTPUT,180) K,WN,XXSTA
      IF (IT.NE.1) GO TO 150
  140 IT=JULIET(2,CROLL,CONR)
      IF (IT.EQ.0) GO TO 160
      WRITE (OUTPUT,190) K,WN,XXSTA
      IF (IT.EQ.1) GO TO 160
  150 WRITE (OUTPUT,200)
      STOP 11
C
  160 HAH=0.0
```

```
      SAS=0.0
      RAR=0.0
      CCA=0.0
      HVH=0.0
      SVS=0.0
      RVR=0.0
      CCV=0.0
C9
C9    SLIGHT INCREASE IN SPEED IF THE FINAL INTEGRATION AVOIDS THE
C9    INTERIOR SURFACE SEGMENTS..
C9    NI = NON - NWL
C9    DO *** I=1,NI
C9
      DO 170 I=1,NON
         CALL SONG (HEAVI(1,I),HEAVT(1,I),ROLLI(1,I),ROLLT(1,I),I)
  170 CONTINUE
C6
C6    FORCE COEFFICIENTS.
C6    FORCE IS THAT WHICH MUST BE APPLIED TO THE CYLINDER (PER UNIT
C6       LENGTH) TO CAUSE SINUSOIDAL OSCILLATIONS AT THE GIVEN FREQUENCY
C6       AND UNIT AMPLITUDE.
C6    COEFFICIENTS ARE THE PARTIAL DERIVATIVES OF THE FORCE BY THE
C6       ACCELERATION OR VELOCITY COMPONENT OF THE GIVEN MOTION.
C6
C6    ACCELERATION TERMS..
C6
      HAH=HAH*RHO2
      SAS=SAS*RHO2
      RAR=RAR*RHO2
      CCA=CCA*RHO2/2.0
C6
C6    VELOCITY TERMS..
C6
      HVH=HVH*RSIG
      SVS=SVS*RSIG
      RVR=RVR*RSIG
      CCV=CCV*RSIG/2.0
C6
      RETURN
C3
  180 FORMAT (36H *** HEAVE MATRIX, FREQUENCY INDEX =,I3,15H, WAVE NUMBE
     1R =,1PE13.5/31H *** COEFFS. FOR STATION AT X =,0PF13.5)
  190 FORMAT (40H *** SWAY-ROLL MATRIX, FREQUENCY INDEX =,I3,15H, WAVE N
     1UMBER =,1PE13.5/31H *** COEFFS. FOR STATION AT X =,0PF13.5)
  200 FORMAT (26H *** EXECUTION TERMINATED.)
C
      END
      SUBROUTINE WOMEN
     1   (I,BLOGP,YLOGP,BLOGM,YLOGM,CHI,CRI,HII,HTI,RII,RTI)
C
      REAL BLOGP(1),BLOGM(1),YLOGP(1),YLOGM(1),CHI(1),HII(1),RII(1),CRI(
     11),HTI(1),RTI(1)
C5
      COMMON/GEOMETRY/MSTA,LPTS(25),YOFF(25,25),NAFT,XAFT(25),
     *       YAFT(25),NFWD,XFWD(25),YFWD(25),XOFF(25),
     *       ZOFF(25,25),XFPERP,XAPERP,SHIPL,SHIPB,SHIPT,
     *       Y1(21,25),ZWL(25),WL(25),INPTS(25),XWLF,XWLA,XXF,
     *       XXA,TAN,NON,NOE,NWL,CR,XXFWD,XXSTA,XXAFT,DX,DX1,
     *       DX2,Z2(21),Y2(21),ZZ(20),YY(20),SNE(20),CSE(20),
     *       DEL(20),ROL(20),ADJUST,WMAX,YMAX,ZMAX,AREA,VERT
      COMMON // HA1,SA1,RA1,CA1,HV1,SV1,RV1,CV1,RHO2,RSIG,WN,W1,UU,ERR,X
```

```
     1RT,YRT,EJT
      COMMON /      /   HAO(24,8),BP(20,20),YP(20,20),BM(20,20),YM(2
     10,20),CONH(40),CONR(40,2),CROLL(40,40),CHEAV(40,40),HEAVI(20,20),H
     2EAVT(20,20),ROLLI(20,20),ROLLT(20,20),EJI,CZRI,CZLI,SZRI,SZLI,RARI
     3,RALI,RBRI,RBLI,CLI,CRQ,SLI,SRI
      DATA TPI/6.28318530717958/
      YYI=YY(I)
      ZZI=ZZ(I)
      SI=SNE(I)
      CI=CSE(I)
      DO 110 J=1,NON
         XRT=WN*(ZZI-Z2(J+1))
         YRT=-WN*(YYI+Y2(J+1))
         EJT=EXP(-YRT)
         CALL ROMEO (CZRT,SZRT,RART,RBRT,CRT,SRT)
         XRT=WN*(ZZI+Z2(J+1))
         CALL ROMEO (CZLT,SZLT,RALT,RBLT,CLT,SLT)
         CJ=CSE(J)
         SJ=SNE(J)
         SSS=SI*CJ
         TTT=SJ*CI
         UUU=CI*CJ
         VVV=SI*SJ
         CIPJ=UUU-VVV
         SIPJ=SSS+TTT
         SIMJ=SSS-TTT
         CIMJ=UUU+VVV
         SSS=SIMJ*(CLI-CLT)-CIMJ*(SLI-SLT)
         TTT=SIPJ*(CRQ-CRT)-CIPJ*(SRI-SRT)
         UUU=SJ*(RALI-RALT)+CJ*(RBLT-RBLI)
         VVV=SJ*(RARI-RART)+CJ*(RBRI-RBRT)
         WWW=EJT*(SZRT*CIPJ-CZRT*SIPJ)-EJI*(SZRI*CIPJ-CZRI*SIPJ)
         RRR=EJT*(SZLT*CIMJ-CZLT*SIMJ)-EJI*(SZLI*CIMJ-CZLI*SIMJ)
         QQQ=EJI*(SZRI*CJ-CZRI*SJ)-EJT*(SZRT*CJ-CZRT*SJ)
         PPP=EJI*(SZLI*CJ+CZLI*SJ)-EJT*(SZLT*CJ+CZLT*SJ)
         CHI(J)=BLOGP(J)+2.0*(TTT-SSS)
         CRI(J)=BLOGM(J)+2.0*(TTT+SSS)
         HII(J)=YLOGP(J)+UU*(VVV+UUU)
         RII(J)=YLOGM(J)+UU*(VVV-UUU)
C        CHN(NJ) = CHI(J)
C        CRN(NJ) = CRI(J)
         NJ=NON+J
         CHI(NJ)=TPI*(WWW-RRR)
         CRI(NJ)=TPI*(WWW+RRR)
         HTI(J)=W1*(QQQ-PPP)
         RTI(J)=W1*(QQQ+PPP)
C        CHN(J) = -CHI(NJ)
C        CRN(J) = -CRI(NJ)
         IF (J.EQ.NON) GO TO 110
         EJI=EJT
         CRQ=CRT
         SRI=SRT
         CLI=CLT
         SLI=SLT
         RARI=RART
         RBRI=RBRT
         RALI=RALT
         RBLI=RBLT
         CZRI=CZRT
         SZRI=SZRT
         CZLI=CZLT
```

```
          SZLI=SZLT
  110 CONTINUE
      RETURN
C
      END
      SUBROUTINE SONG (HAI,HOT,RAI,ROT,I)
C
      DIMENSION HAI(1), HOT(1), RAI(1), ROT(1), PP(6)
C5
      COMMON/GEOMETRY/MSTA,LPTS(25),YOFF(25,25),NAFT,XAFT(25),
     *       YAFT(25),NFWD,XFWD(25),YFWD(25),XOFF(25),
     *       ZOFF(25,25),XFPERP,XAPERP,SHIPL,SHIPB,SHIPT,
     *       Y1(21,25),ZWL(25),WL(25),INPTS(25),XWLF,XWLA,XXF,
     *       XXA,TAN,NON,NOE,NWL,CR,XXFWD,XXSTA,XXAFT,DX,DX1,
     *       DX2,Z2(21),Y2(21),ZZ(20),YY(20),SNE(20),CSE(20),
     *       DEL(20),ROL(20),ADJUST,WMAX,YMAX,ZMAX,AREA,VERT
      COMMON // HAH,SAS,RAR,CCA,HVH,SVS,RVR,CCV,RHO2,RSIG,WN,W1,W2,ERR,Z
     1RI,YRI,EJT
      COMMON /      /   HAO(24,8),BLOGP(20,20),YLOGP(20,20),BLOGM(20
     1,20),YLOGM(20,20),CONH(40),CONR(40,2)
      PAH=0.0
      PAS=0.0
      PAR=0.0
      PVH=0.0
      PVS=0.0
      PVR=0.0
      DO 110 J=1,NON
         NJ=NON+J
         PAH=PAH+CONH(J)*HOT(J)-CONH(NJ)*HAI(J)
         PAS=PAS+CONR(J,1)*ROT(J)-CONR(NJ,1)*RAI(J)
         PAR=PAR+CONR(J,2)*ROT(J)-CONR(NJ,2)*RAI(J)
         PVH=PVH+CONH(J)*HAI(J)+CONH(NJ)*HOT(J)
         PVS=PVS+CONR(J,1)*RAI(J)+CONR(NJ,1)*ROT(J)
         PVR=PVR+CONR(J,2)*RAI(J)+CONR(NJ,2)*ROT(J)
  110 CONTINUE
      DDD=DEL(I)
      DCI=CSE(I)*DDD
      DSI=-SNE(I)*DDD
      DFR=ROL(I)*DDD
C6
C6    THE PRESSURES ON THIS SEGMENT OF THE CYLINDER MAY BE CALCULATED.
C6
C6    THE PRESSURES IN PHASE WITH THE SINUSOIDAL DISPLACEMENT ARE..
C6
C6    HEAVE -- PAH = PAH*RHO*ESIG*ESIG
C6    SWAY  -- PAS = PAS*RHO*ESIG*ESIG
C6    ROLL  -- PAR = PAR*RHO*ESIG*ESIG
C6
C6       OF COURSE THE ACCELERATION COMPONENTS OF THE FORCE ARE EQUAL
C6       IN MAGNITUDE TO THE ABOVE, BUT HAVE THE OPPOSITE SIGN.
C6
C6    THE PRESSURES IN PHASE WITH THE SINUSOIDAL VELOCITY ARE..
C6
C6    HEAVE -- PVH = PVH*RHO*ESIG*ESIG
C6    SWAY  -- PVS = PVS*RHO*ESIG*ESIG
C6    ROLL  -- PVR = PVR*RHO*ESIG*ESIG
C6
C6
C6    INTEGRATION TO OBTAIN FORCE ACCELERATION COEFFICIENTS..
C6
      HAH=HAH+PAH*DCI
```

```
      SAS=SAS+PAS*DSI
      RAR=RAR+PAR*DFR
C6
      CCA=PAR*DSI+PAS*DFR+CCA
C6    INTEGRATION TO OBTAIN FORCE VELOCITY COEFFICIENTS..
C6
      HVH=HVH+PVH*DCI
      SVS=SVS+PVS*DSI
      RVR=RVR+PVR*DFR
      CCV=PVR*DSI+PVS*DFR+CCV
C6
C6    AT THIS POINT THE PRESSURES REQUIRE THE MODIFICATIONS NOTED
C6       ABOVE TO GIVE THE DIMENSIONAL VALUES.
C6       THE INTEGRATIONS OF THE PRESSURES ARE COMPLETED ELSEWHERE
C6       IN THE PROGRAM TO GIVE DIMENSIONAL FORCE COEFFICIENTS.
C6
      RETURN
C
      END
      SUBROUTINE ROMEO (C,S,RA,RB,CIN,SON)
C
C1    EXPONENTIAL INTEGRAL WITH COMPLEX ARGUMENT.
C1
C2    THE ARGUMENT IS SUPPLIED THROUGH BLANK COMMON AS THE VARIABLES
C2    X   AND   Y .
C2
C3    PARAMETERS AND VARIABLES..
C3       X        -- REAL PART OF ARGUMENT
C3       Y        -- IMAGINARY PART OF ARGUMENT
C3       E        -- EXP( -Y )
C3       C        -- COS( X )
C3       S        -- SIN( X )
C3       CIN      -- REAL RESULT
C3       SON      -- IMAGINARY RESULT
C3       RA       -- ALOG( X**2 + Y**2 )/2.0 - CIN
C3       RB       -- ATAN( X/Y ) - PI/2.0 + SON
C3
C5
      COMMON/GEOMETRY/MSTA,LPTS(25),YOFF(25,25),NAFT,XAFT(25),
     *       YAFT(25),NFWD,XFWD(25),YFWD(25),XOFF(25),
     *       ZOFF(25,25),XFPERP,XAPERP,SHIPL,SHIPB,SHIPT,
     *       Y1(21,25),ZWL(25),WL(25),INPTS(25),XWLF,XWLA,XXF,
     *       XXA,TAN,NON,NOE,NWL,CR,XXFWD,XXSTA,XXAFT,DX,DX1,
     *       DX2,Z2(21),Y2(21),ZZ(20),YY(20),SNE(20),CSE(20),
     *       DEL(20),ROL(20),ADJUST,WMAX,YMAX,ZMAX,AREA,VERT
      COMMON /      / HA1, SA1, RA1, CA1, HV1, SV1, RV1, CV1,
     1   RHO2, RSIG, WN, W1, W2, ERR, X, Y, E
C5
C5    INPUT AND OUTPUT LOGICAL UNITS..
      COMMON/IO/INPUT,OUTPUT,BIF,OFF,COF
      INTEGER OUTPUT,BIF,OFF,COF
      DATA  GAMMA /0.5772 15664 90153 28606 06512/
      DATA  HALFPI/1.5707 96326 79489 66192 31322/
C
      AT=ATAN2(X,Y)
      ARG=AT-HALFPI
      C=COS(X)
      S=SIN(X)
      R=X*X+Y*Y
      AL=0.5*ALOG(R)
      TEST=0.00001
```

```
      IF (R.LT.1.) GO TO 130
      TEST=.1*TEST
      IF (R.LT.2.) GO TO 130
      TEST=.1*TEST
      IF (R.LT.4.) GO TO 130
      TEST=.1*TEST
  130 SUMC=GAMMA+AL+Y
      SUMS=AT+X
      TC=Y
      TS=X
      COX=1.
      DO 140 K=2,501
         TO=TC
         FACT=COX/FLOAT(K)**2
         COX=K
         TC=FACT*(Y*TC-X*TS)
         TS=FACT*(Y*TS+X*TO)
         SUMC=SUMC+TC
         SUMS=SUMS+TS
         IF ((ABS(TC)+ABS(TS)).LE.TEST) GO TO 150
  140 CONTINUE
      WRITE (OUTPUT,190) XXSTA,X,Y,WN
      STOP 12
  150 CIN=E*(C*SUMC+S*SUMS)
      SON=E*(S*SUMC-C*SUMS)
      RA=AL-CIN
      RB=ARG+SON
      RETURN
C
  190 FORMAT (59H0*** NON-CONVERGENT EXPONENTIAL INTEGRAL FOR STATION AT
     1 X =,F13.5/22H *** PARAMETERS -- X =,1PE13.5,5H, Y =,E13.3,15H, WA
     2VE NUMBER =,E13.5)
C
      END
      FUNCTION JULIET (N1,A,B)
C
C..  SOLVES SIMULTANEOUS LINEAR EQUATIONS BY GAUSSIAN REDUCTION.
C..  FOR THE SPECIALIZED MATRICES IN THE SUBROUTINE WINE.
C
      REAL A(40,1),B(40,1)
C2
C2    THE A MATRIX MUST BE DIMENSIONED WITH EXACTLY 40 ROWS AND
C2    AT LEAST 40 COLUMNS.  THE B MATRIX MUST ALSO BE DIMENSIONED
C2    WITH EXACTLY 40 ROWS AND AT LEAST N1 COLUMNS.
C2
C5
C5    INPUT AND OUTPUT LOGICAL UNITS..
      COMMON/IO/INPUT,OUTPUT,BIF,OFF,COF
      INTEGER OUTPUT,BIF,OFF,COF
C5
C5
      COMMON/GEOMETRY/MSTA,LPTS(25),YOFF(25,25),NAFT,XAFT(25),
     *       YAFT(25),NFWD,XFWD(25),YFWD(25),XOFF(25),
     *       ZOFF(25,25),XFPERP,XAPERP,SHIPL,SHIPB,SHIPT,
     *       Y1(21,25),ZWL(25),WL(25),INPTS(25),XWLF,XWLA,XXF,
     *       XXA,TAN,NON,NOE,NWL,CR,XXFWD,XXSTA,XXAFT,DX,DX1,
     *       DX2,Z2(21),Y2(21),ZZ(20),YY(20),SNE(20),CSE(20),
     *       DEL(20),ROL(20),ADJUST,WMAX,YMAX,ZMAX,AREA,VERT
      COMMON // HA1,SA1,RA1,CA1,HV1,SV1,RV1,CV1,RHO2,RSIG,WN,W1,W2,ERR,Z
     1RI,YRI,EJT
C
```

```
C
      COMMON /       /   HA0(24,8),BLOGP(20,20,4),CONH(40,3),CROLL(40
     1,40,2),HEAVI(20,20,4),EJI,CZRI,CZLI,SZRI,SZLI,RARI,RALI,RBRI,RBLI,
     2CLI,CRQ,SLI,SRI,Z(40)
C
      N2=NOE/2
      D=1.0
C
C.. COMPLETE THE MATRIX A.
C
      DO 120 J=1,N2
         L=N2+J
         DO 110 I=1,N2
            K=N2+I
            A(I,L)=-A(K,J)
            A(K,L)=A(I,J)
  110    CONTINUE
  120 CONTINUE
C
      NM1=NOE-1
C
C.. SOLVE   AT*X=B   FOR   X , WHERE   AT   IS THE TRANSPOSE OF THE
C   MATRIX   A .  STORE THE   X   VECTOR(S)  IN   B .
C
      DO 210 J=1,NM1
         J1=J+1
C
C.. FIND ELEMENT OF ROW J, COLS J--N, WHICH HAS MAX ABSOLUTE VALUE.
C
         LMAX=J
         RMAX=ABS(A(J,J))
         DO 130 K=J1,NOE
            RNEXT=ABS(A(J,K))
            IF (RMAX.GE.RNEXT) GO TO 130
            RMAX=RNEXT
            LMAX=K
  130    CONTINUE
         IF (LMAX.NE.J) GO TO 140
C
C.. MAX ELEMENT IS ON DIAGONAL.
C
         IF (A(J,J)) 170,260,170
C
C.. MAX ELEMENT IS NOT ON DIAGONAL.
C.. EXCHANGE COLUMNS J AND LMAX.
C
  140    DO 150 L=J,NOE
            W=A(L,J)
            A(L,J)=A(L,LMAX)
            A(L,LMAX)=W
  150    CONTINUE
C
C.. EXCHANGE ROWS J AND LMAX.
C
         DO 160 L=1,N1
            W=B(J,L)
            B(J,L)=B(LMAX,L)
            B(LMAX,L)=W
  160    CONTINUE
         D=-D
C
```

```
C.. ZERO ROW J TO RIGHT OF DIAGONAL.
C
  170     D=A(J,J)*D
          V=1.0/A(J,J)
          Z(J)=V
          DO 200 K=J1,NOE
             IF (A(J,K).EQ.0.0) GO TO 200
             W=-V*A(J,K)
             DO 180 L=J1,NOE
                A(L,K)=W*A(L,J)+A(L,K)
  180        CONTINUE
             DO 190 L=1,N1
                B(K,L)=W*B(J,L)+B(K,L)
  190        CONTINUE
  200     CONTINUE
  210 CONTINUE
      D=A(NOE,NOE)*D
      IF (A(NOE,NOE).EQ.0.0) GO TO 260
      Z(NOE)=1.0/A(NOE,NOE)
C
C.. OBTAIN SOLUTION BY BACK SUBSTITUTION.
C
      DO 220 L=1,N1
         B(NOE,L)=Z(NOE)*B(NOE,L)
  220 CONTINUE
      DO 250 K=1,NM1
         J=NOE-K
         J1=J+1
         DO 240 L=1,N1
            W=0.0
            DO 230 I=J1,NOE
               W=A(I,J)*B(I,L)+W
  230       CONTINUE
            B(J,L)=(B(J,L)-W)*Z(J)
  240    CONTINUE
  250 CONTINUE
      IF (ABS(D).LT.ERR) GO TO 270
C
      JULIET=0
C
C.. NO PROBLEMS DURRING THIS EXECUTION.
C
      RETURN
C
C.. SINGULAR MATRIX--MAXIMUM ELEMENT IN ROW IS ZERO.
C
  260 JULIET=3
      WRITE (OUTPUT,280)
      RETURN
C
C.. ABSOLUTE DETERMINANT VALUE LESS THAN ERROR VALUE..
C
  270 JULIET=1
      WRITE (OUTPUT,290) D,ERR
      RETURN
C
  280 FORMAT (20H0*** SINGULAR MATRIX)
  290 FORMAT (18H0*** DETERMINANT =,1PE13.5,24H,   ERROR SPECIFICATION =,
     1E13.5)
C
      END
```

```
      FUNCTION LNEQT(M,N,N1,A,B,ERROR ,Z)
C
C..  SOLVES SIMULTANEOUS LINEAR EQUATIONS BY GAUSSIAN REDUCTION.
C..  SOLVES   A*X=B   FOR   X ,  AND STORES THE   X   VECTOR(S) IN   B .
C
      REAL A(M,M),B(M,M) ,Z(M),ERROR ,RMAX,RNEXT,W
C
C5
C5    INPUT AND OUTPUT LOGICAL UNITS..
      COMMON/IO/INPUT,OUTPUT,BIF,OFF,COF
      INTEGER OUTPUT,BIF,OFF,COF
C5
C
      D=1.0
      NM1=N-1
      IF (NM1.EQ.0) GO TO 210
      DO 200 J=1,NM1
         J1=J+1
C
C..  FIND ELEMENT OF COL J, ROWS J-N, WHICH HAS MAX ABSOLUTE VALUE.
C
         LMAX=J
         RMAX=ABS(A(J,J))
         DO 110 K=J1,N
            RNEXT=ABS(A(K,J))
            IF (RMAX.GE.RNEXT) GO TO 110
            RMAX=RNEXT
            LMAX=K
  110    CONTINUE
         IF (LMAX.NE.J) GO TO 120
C
C..  MAX ELEMENT IN COLUMN IS ON DIAGONAL
C
         IF (A(J,J)) 150,270,150
C
C..  MAX ELEMENT IS NOT ON DIAGONAL.  EXCHANGE ROWS J AND LMAX.
C
  120    DO 130 L=J,N
            W=A(J,L)
            A(J,L)=A(LMAX,L)
            A(LMAX,L)=W
  130    CONTINUE
         DO 140 L=1,N1
            W=B(J,L)
            B(J,L)=B(LMAX,L)
            B(LMAX,L)=W
  140    CONTINUE
         D=-D
C
C..  ZERO COLUMN J BELOW THE DIAGONAL.
C
  150    D=A(J,J)*D
         Z(J)=1.0/A(J,J)
         DO 190 K=J1,N
            IF (A(K,J)) 160,190,160
  160       W=-Z(J)*A(K,J)
            DO 170 L=J1,N
               A(K,L)=W*A(J,L)+A(K,L)
  170       CONTINUE
            DO 180 L=1,N1
               B(K,L)=W*B(J,L)+B(K,L)
```

```
  180          CONTINUE
  190       CONTINUE
  200 CONTINUE
  210 D=A(N,N)*D
      IF (A(N,N).EQ.0.0) GO TO 270
      Z(N)=1./A(N,N)
C
C.. OBTAIN SOLUTION BY BACK SUBSTITUTION.
C
      DO 220 L=1,N1
         B(N,L)=Z(N)*B(N,L)
  220 CONTINUE
      IF (NM1.EQ.0) GO TO 260
      DO 250 K=1,NM1
         J=N-K
         J1=J+1
         DO 240 L=1,N1
            W=0.
            DO 230 I=J1,N
               W=A(J,I)*B(I,L)+W
  230       CONTINUE
            B(J,L)=(B(J,L)-W)*Z(J)
  240    CONTINUE
  250 CONTINUE
  260 LNEQT=0
      IF (ABS(D).GE.ERROR) RETURN
      LNEQT=1
      WRITE (OUTPUT,280) D,ERROR
      RETURN
C
C.. SINGULAR MATRIX--MAXIMUM ELEMENT IN COLUMN IS ZERO.
C
  270 LNEQT=3
      WRITE (OUTPUT,290)
      RETURN
C
  280 FORMAT (18H0*** DETERMINANT =,1PE13.5,24H,   ERROR SPECIFICATION =,
     1E13.5)
  290 FORMAT (20H0*** SINGULAR MATRIX)
C
      END
```

PROGRAM HYDREX3

```
      PROGRAM HYDREX3
C
      CHARACTER*25 PANFIL,MATFIL
       COMMON/BD/XPAN(120),YPAN(120),ZPAN(120),AREA(120),
     * ST(120),ACN(120),ACNW(120),AN(120,3),E(120),P(120,6),
     * PRFS(120),STOLD(120),PX(120,6)
       COMMON/FS/AKZ(100,100),SS(100,100),CC(100,100),
     * DKX(100),DKY(100),AKX(100),AKY(100)
       COMMON/BD2/XPT(150),YPT(150),ZPT(150),WRF(150),
     * WRFR(150),KK(150,4)
       COMMON/A/NPAN,NPT,GEE,RHO,NKX,NKY,EYE,DT,TIM,UFWD
       DIMENSION PF(6),PB(6),PT(6)
       COMMON/WAVEX/OMEGA
       COMMON/BP/BPRES(120),TPRES(120)
       COMPLEX EYE
       EYE=(0.0,1.0)
C--------------------------------------------------------------
       TYPE 1
       ACCEPT 4,PANFIL
       TYPE 2
       ACCEPT 4,MATFIL
       OPEN(UNIT=2,FILE=PANFIL,TYPE='OLD')
       OPEN(UNIT=3,FILE=MATFIL,FORM='UNFORMATTED',TYPE='NEW')
       OPEN(UNIT=99,FILE='X.DAT',FORM='UNFORMATTED',TYPE ='NEW')
       CALL EBD
       CALL POTST
       WRITE(6,6)
       WRITE(6,7) J,AN(J,1),AN(J,2),AN(J,3),
     *              XPAN(J),YPAN(J),ZPAN(J),AREA(J)
150    CONTINUE
       STOP
 1     FORMAT(' Input name of [PAN] file >'$)
 2     FORMAT(' Input name of [MAT] file >'$)
  4    FORMAT(A)
  6    FORMAT( 'J',9X,'NX',9X,'NY',9X,'NZ',9X,'XP',9X,
     * 'YP',9X,'ZP',9X,'AREA')
  7    FORMAT(1X,I5,7F11.4)
       END
```

```
       SUBROUTINE PREP(J)
       COMMON/BD/XPAN(120),YPAN(120),ZPAN(120),AREA(120),ST(120),
    *  ACN(120),ACNW(120),AN(120,3),E(120),P(120,6),PRFS(120),
    *   STOLD(120),PX(120,6)
       COMMON/BD2/XPT(150),YPT(150),ZPT(150),WRF(150),WRFR(150),
    *           KK(150,4)
       COMMON/ARE/RR(500),XZJ(200),YXJ(200),ZYJ(200)
       ZYT=0.0
       YXT=0.00
       XZT=0.00
       J4=J*4
       JT=4
       IF(KK(J,4).EQ.0) JT=3
       DO 20 JJ=1,JT
       J4=J4+1
       J2=1
       IF(JJ.LT.JT) J2=JJ+1
       KF=KK(J,JJ)
       KG=KK(J,J2)
       AG=XPT(KG)
       BG=YPT(KG)
       CG=ZPT(KG)
       AF=XPT(KF)
       BF=YPT(KF)
       CF=ZPT(KF)
       R=SQRT((AF-AG)**2+(BF-BG)**2+(CF-CG)**2)
       XT=AF-XPAN(J)
       YT=BF-YPAN(J)
       ZT=CF-ZPAN(J)
       ANX=(AF-AG)/R
       ANY=(BF-BG)/R
       ANZ=(CF-CG)/R
       DOT=ANX*XT+ANY*YT+ANZ*ZT
       XT=XT-DOT*ANX
       YT=YT-DOT*ANY
       ZT=ZT-DOT*ANZ
       ZYT=ZYT+ZT*ANY-ANZ*YT
       YXT=YXT+YT*ANX-ANY*XT
       XZT=XZT+XT*ANZ-ANX*ZT
       RR(J4)=R
20     CONTINUE
       XZJ(J)=SIGN(AN(J,2),XZT)
       YXJ(J)=SIGN(AN(J,3),YXT)
       ZYJ(J)=SIGN(AN(J,1),ZYT)
       RETURN
       END


       SUBROUTINE SELF(AF,BF,CF,AG,BG,CG,FEE)
       REAL LB21,LA21
       ASQ=AF*AF+BF*BF+CF*CF
       BSQ=AG*AG+BG*BG+CG*CG
       ADB=AF*AG+BF*BG+CF*CG
       ADB2=ADB+ADB
       ASAS=(AF*BG-BF*AG)**2+(CF*BG-BF*CG)**2+(AF*CG-BF*AG)**2
       FF=0.00
       DO 15 MK=1,10
       DO 15 NK=1,MK
       LA21=FLOAT(NK-MK)
       A2SQ=ASQ*LA21*LA21
       DO 15 ML=1,11-MK
       DO 15 NL=1,11-ML
       LB21=FLOAT(NL-ML)
       IF(LA21.NE.0.) GO TO 5
       IF(LB21.LT.0.) GO TO 5
       GO TO 15
5      R=SQRT(A2SQ+ABD2*LA21*LB21+BSQ*LB21*LB21)
       FF=FF+1.0/R
15     CONTINUE
       FEE=FF*ASAS*0.002
       RETURN
       END
```

```
        SUBROUTINE SOLID(XPN,G,NSIDE)
        DIMENSION CS(4),SN(4),Z(4),XPN(3,4)
        G=-6.283185308
        ACR12=XPN(1,1)*XPN(1,2)+XPN(2,1)*XPN(2,2)+XPN(3,1)*XPN(3,2)
        ACR13=XPN(1,1)*XPN(1,3)+XPN(2,1)*XPN(2,3)+XPN(3,1)*XPN(3,3)
        ACR23=XPN(1,2)*XPN(1,3)+XPN(2,2)*XPN(2,3)+XPN(3,2)*XPN(3,3)
        IF(NSIDE.EQ.4) GO TO 40
        G=-3.141592659
        CS(1)=ACR23-ACR13*ACR12
        CS(2)=ACR13-ACR12*ACR23
        CS(3)=ACR12-ACR23*ACR13
        SN(1)=XPN(1,1)*(XPN(2,2)*XPN(3,3)-XPN(3,2)*XPN(2,3))+
     +          XPN(2,1)*(XPN(3,2)*XPN(1,3)-XPN(1,2)*XPN(3,3))+
     +          XPN(3,1)*(XPN(1,2)*XPN(2,3)-XPN(2,2)*XPN(1,3))
        SN(2)=SN(1)
        SN(3)=SN(1)
        SN(4)=0.
        GO TO 50
40      ACR14=XPN(1,1)*XPN(1,4)+XPN(2,1)*XPN(2,4)+XPN(3,1)*XPN(3,4)
        ACR24=XPN(1,2)*XPN(1,4)+XPN(2,2)*XPN(2,4)+XPN(3,2)*XPN(3,4)
        ACR34=XPN(1,3)*XPN(1,4)+XPN(2,3)*XPN(2,4)+XPN(3,3)*XPN(3,4)
        CS(1)=ACR24-ACR14*ACR12
        CS(2)=ACR13-ACR23*ACR12
        CS(3)=ACR24-ACR34*ACR23
        CS(4)=ACR13-ACR34*ACR14
        B241=XPN(2,2)*XPN(3,4)-XPN(3,2)*XPN(2,4)
        B242=XPN(3,2)*XPN(1,4)-XPN(1,2)*XPN(3,4)
        B243=XPN(1,2)*XPN(2,4)-XPN(2,2)*XPN(1,4)
        B131=XPN(2,1)*XPN(3,3)-XPN(3,1)*XPN(2,3)
        B132=XPN(3,1)*XPN(1,3)-XPN(1,1)*XPN(3,3)
        B133=XPN(1,1)*XPN(2,3)-XPN(2,1)*XPN(1,3)
        SN(1)=XPN(1,1)*B241+XPN(2,1)*B242+XPN(3,1)*B243
        SN(2)=-(XPN(1,2)*B131+XPN(2,2)*B132+XPN(3,2)*B133)
        SN(3)=-(XPN(1,3)*B241+XPN(2,3)*B242+XPN(3,3)*B243)
        SN(4)=XPN(1,4)*B131+XPN(2,4)*B132+XPN(3,4)*B133
50      CONTINUE
D       TYPE 8844,SN(1),CS(1)
D       TYPE 8844,SN(2),CS(2)
D       TYPE 8844,SN(3),CS(3)
D       TYPE 8844,SN(4),CS(4)
8844    FORMAT('      SN,CS=',2F15.8)
        SUM=SN(1)+SN(2)+SN(3)+SN(4)
        IF(ABS(SUM).GT.0.01) GO TO 25
        IF(ABS(CS(1)).GT.ABS(SN(1))) GO TO 25
        IF(ABS(CS(2)).GT.ABS(SN(2))) GO TO 25
        IF(ABS(CS(3)).GT.ABS(SN(3))) GO TO 25
1090    G=SUM*.25
        IF(NSIDE.EQ.3) G=SUM*.1666666667
        RETURN
 25     ST=SN(NSIDE)
        DO 30 I=1,NSIDE
        IF((ABS(CS(I)).LT.9E-8).AND.(ABS(SN(I)).LT..9E-05))
     +  GO TO 1090
        IF(ST*SN(I).LT.0.) GO TO 1090
        ST=SN(I)
        C2=CS(I)/SQRT(SN(I)**2+CS(I)**2)
        G=G+ACOS(C2)
30      CONTINUE
        RETURN
        END
```

```
        SUBROUTINE EBD
C INITIALIZE PANELS AND COMPUTE BODY MATRIX
C
        COMMON/BD/XPAN(120),YPAN(120),ZPAN(120),AREA(120),ST(120),
     *  ACN(120),ACNW(120),AN(120,3),E(120),P(120,6),PRFS(120),
     *   STOLD(120),PX(120,6)
        COMMON/BD2/XPT(150),YPT(150),ZPT(150),WRF(150),WRFR(150),
     *   KK(150,4)
        COMMON/A/NPAN,NPT,GEE,RHO,NKX,NKY,EYE,DT,TIM,UFWD
        DIMENSION EP(120),EPP(120)
        COMPLEX EYE
C READ IN BODY PANEL PARAMETERS
101     FORMAT(4I5)
100     FORMAT(3F10.0)
103     FORMAT(3F10.2)
104     FORMAT(I3)
C NUMBER OF POINTS AND PANELS
        READ(2,101) NPT,NPAN
        TYPE 101, NPT,NPAN
C COORDIANTES OF POINTS
        READ(2,100) (XPT(N),YPT(N),ZPT(N),N=1,NPT)
        DO 7777 N=1,NPT
        TYPE 103, XPT(N),YPT(N),ZPT(N)
7777    CONTINUE
C DEFINE CORNER PINTS OF EACH PANEL
        READ(2,101) (KK(N,1),KK(N,2),KK(N,3),KK(N,4),N=1,NPAN)
C COMPUTE PANEL AREAS
        DO 150 J=1,NPAN
        K1=KK(J,1)
        K2=KK(J,2)
        K3=KK(J,3)
        K4=KK(J,4)
        IF(K4.EQ.0) GO TO 8
        XPAN(J)=(XPT(K1)+XPT(K2)+XPT(K3)+XPT(K4))*0.25
        YPAN(J)=(YPT(K1)+YPT(K2)+YPT(K3)+YPT(K4))*0.25
        ZPAN(J)=(ZPT(K1)+ZPT(K2)+ZPT(K3)+ZPT(K4))*0.25
        GO TO 9
C TRIANGULAR PANELS
8       XPAN(J)=(XPT(K1)+XPT(K2)+XPT(K3))/3.0
        YPAN(J)=(YPT(K1)+YPT(K2)+YPT(K3))/3.0
        ZPAN(J)=(ZPT(K1)+ZPT(K2)+ZPT(K3))/3.0
        K4=K3
9       XA=XPT(K3)-XPT(K1)
        XB=XPT(K4)-XPT(K2)
        YA=YPT(K3)-YPT(K1)
        YB=YPT(K4)-YPT(K2)
        ZA=ZPT(K3)-ZPT(K1)
        ZB=ZPT(K4)-ZPT(K2)
C COMPUTE PANEL AREAS
        AZ=XA*YB-YA*XB
        AX=YA*ZB-ZA*YB
        AY=ZA*XB-XA*ZB
        ARE=SQRT(AX*AX+AY*AY+AZ*AZ)
        AREA(J)=ARE*0.50
        AN(J,1)=-AX/ARE
        AN(J,2)=-AY/ARE
        AN(J,3)=-AZ/ARE
150     CONTINUE
808     FORMAT(1X,I5,5F11.4)
        DO 1308 J=1,NPAN
        JJJ=J
        CALL PREP(JJJ)
        ST(J)=0.00
        DO 1308 K=1,6
        PX(J,K)=0.00
1308    CONTINUE
        DO 308 J=1,NPAN
        JJJ=J
        AX=-AN(J,1)
        AY=-AN(J,2)
        AZ=-AN(J,3)
        XF=XPAN(J)
        YF=YPAN(J)
        ZF=ZPAN(J)
        DO 157 L=1,NPT
        WRF(L)=SQRT((XPT(L)-XF)**2+(YPT(L)-YF)**2+(ZPT(L)-ZF)**2)
157     WRFR(L)=SQRT((XPT(L)-XF)**2+(YPT(L)-YF)**2+(ZPT(L)+ZF)**2)
C
```

```
        DO 500 I=1,3
        WRITE(3) (AN(J,I),J=1,NPAN)
 500    CONTINUE
        WRITE(3) (XPAN(J),J=1,NPAN)
        WRITE(3) (YPAN(J),J=1,NPAN)
        WRITE(3) (ZPAN(J),J=1,NPAN)
        WRITE(3) (AREA(J),J=1,NPAN)
        DO 309 JL=1,NPAN
        JLJ=JL
        CALL GE(XF,YF,ZF,JLJ,VX,VY,VZ,VXR,VYR,VZR,JJJ)
        VX=VX+VXR
        VY=VY+VYR
        VZ=VZ+VZR
C       COMPUTE NORMAL VELOCITY AT PANEL J DUE TO PANEL JL
        E(JL)=AX*VX+AY*VY+AZ*VZ
C INCREMENT PX MATIRX
        FR1=-AREA(J)*VX*AN(J,1)
        FR2=-AREA(J)*VX*AN(J,2)
        FR3=-AREA(J)*VX*AN(J,3)
        PX(JL,1)=PX(JL,1)+FR1
        PX(JL,2)=PX(JL,2)+FR2
        PX(JL,3)=PX(JL,3)+FR3
        PX(JL,4)=PX(JL,4)+YF*FR3-ZF*FR2
        PX(JL,5)=PX(JL,5)+ZF*FR1-XF*FR3
        PX(JL,6)=PX(JL,6)+XF*FR2-YF*FR1
309     CONTINUE
        WRITE(99) (E(JL),JL=1,NPAN)
308     CONTINUE
        DO 2424 K=1,6
2424    WRITE(3) (PX(JL,K),JL=1,NPAN)
        CLOSE (UNIT=99)
C       INVERT E MATRIX
        CALL MATIN(NPAN)
        RETURN
        END
```

```
        SUBROUTINE MATIN(NPAN)
C INVERST MATRIX
        DIMENSION E(120,120),BB(120),EST(120)
        OPEN(UNIT=99,FILE='SCR',FORM='UNFORMATTED',TYPE ='OLD')
        DO 120 J=1,NPAN
120     READ(99) (E(J,I),I=1,NPAN)
        DO 130 J=1,NPAN
        DO 11 MM=1,NPAN
        EST(MM)=0.00
11      BB(MM)=0.00
        BB(J)=1.0
        EST(J)=1.0/E(J,J)
        DO 17 NIT=1,6
        DO 17 K=1,NPAN
        B=BB(K)
        DO 15 I=1,NPAN
15      IF(I.NE.K) B=B-E(K,I)*EST(I)
        EST(K)=B/E(K,K)
17      CONTINUE
        WRITE(3) (EST(K),K=1,NPAN)
130     CONTINUE
        RETURN
        END
```

```
C
       SUBROUTINE GE(XF, YF, ZF, J, V1, V2, V3, V1R, V2R, V3R, NBT)
       COMMON/BD/XPAN(120), YPAN(120), ZPAN(120), AREA(120), ST(120),
     * ACN(120), ACNW(120), AN(120, 3), E(120), P(120, 6), PRFS(120),
     *  STOLD(120), PX(120, 6)
       COMMON/BD2/XPT(150), YPT(150), ZPT(150), WRF(150), WRFR(150)
     *    , KK(150, 4)
       COMMON/ARE/RR(500), XZJ(200), YXJ(200), ZYJ(200)
       DIMENSION XSA(3, 4), XFA(3), XSAR(3, 4)
       J4=J*4
       V1=0. 00
       V2=0. 00
       V3=0. 00
       V1R=0. 00
       V2R=0. 00
       V3R=0. 00
       XNJ=AN(J, 1)
       YNJ=AN(J, 2)
       ZNJ=AN(J, 3)
       NSIDE=4
       IF(KK(J, 4). EQ. 0) NSIDE=3
       DO 20 JJ=1, NSIDE
       J2=1
       IF(JJ. LT. NSIDE) J2=JJ+1
       J4=J4+1
       KF=KK(J, JJ)
       AF=XPT(KF)
       BF=YPT(KF)
       CF=ZPT(KF)
       R=RR(J4)
       KG=KK(J, J2)
       ANX=(AF-XPT(KG))/R
       ANY=(BF-YPT(KG))/R
       ANZ=(CF-ZPT(KG))/R
       A=AF-XF
       B=BF-YF
       C=CF-ZF
       TX=XZJ(J)*ANZ-YXJ(J)*ANY
       TY=YXJ(J)*ANX-ZYJ(J)*ANZ
       TZ=ZYJ(J)*ANY-XZJ(J)*ANX
       EX1=A*ANX+B*ANY+C*ANZ
       CALL GO(EX1, R, FF, WRF(KF), WRF(KG))
       V1=V1+FF*TX
       V2=V2+FF*TY
       V3=V3+FF*TZ
        XSA(1, JJ)=-A/WRF(KF)
        XSA(2, JJ)=-B/WRF(KF)
        XSA(3, JJ)=-C/WRF(KF)
       EX1R=EX1+2. 0*ZF*ANZ
       CR=-CF-ZF
       CALL GO(EX1R, R, FR, WRFR(KF), WRFR(KG))
       V1R=V1R-FR*TX
       V2R=V2R-FR*TY
       V3R=V3R+FR*TZ
       XSAR(1, JJ)=-A/WRFR(KF)
       XSAR(2, JJ)=-B/WRFR(KF)
       XSAR(3, JJ)=CR/WRFR(KF)
20     CONTINUE
       G=6. 283185307
       IF(J. EQ. NBT) GO TO 84
       CALL SOLID(XSA, G, NSIDE)
       AGG=A*XNJ+B*YNJ+C*ZNJ
       G=-SIGN(G, AGG)
84     CONTINUE
       CALL SOLID(XSAR, GR, NSIDE)
       AGGR=A*XNJ+B*YNJ-CR*ZNJ
       GR=SIGN(GR, AGGR)
85     CONTINUE
7371   FORMAT(' G, GR=',  2F15. 5)
       V1=V1+XNJ*G
       V2=V2+YNJ*G
       V3=V3+ZNJ*G
       V1R=V1R+XNJ*GR
       V2R=V2R+YNJ*GR
       V3R=V3R-ZNJ*GR
5590   FORMAT(' V1, V2, V3=', 3F15. 5)
5591   FORMAT(' V1R, V2R, V3R=', 3F15. 5)
       RETURN
       END
```

```
C
      SUBROUTINE POTST
      COMMON/BD/XPAN(120),YPAN(120),ZPAN(120),AREA(120),ST(120),
     *ACN(120),ACNW(120),AN(120,3),E(120),P(120,6),PRFS(120),
     * STOLD(120),PX(120,6)
      COMMON/BD2/XPT(150),YPT(150),ZPT(150),WRF(150),WRFR(150),
     * KK(150,4)
      COMMON/A/NPAN,NPT,GEE,RHO,NKX,NKY,EYE,DT,TIM,UFWD
      COMPLEX A,B,EYE
      DIMENSION XPSL(3,4),XPSLR(3,4),PBB(120,120)
      COMMON/PTST/ARE4(200,4),X4(200,4),Y4(200,4),Z4(200,4)
     *,SEL(200,4)
      DO 1500 J=1,NPAN
      ARE4(J,4)=-1.0
      JT=4
      IF(KK(J,4).EQ.0) JT=3
      DO 1500 JJ=1,JT
      J2=1
      IF(JJ.LT.JT) J2=JJ+1
      KF=KK(J,JJ)
      KG=KK(J,J2)
      X4(J,JJ)=(XPT(KF)+XPT(KG)+XPAN(J))/3.0
      Y4(J,JJ)=(YPT(KF)+YPT(KG)+YPAN(J))/3.0
      Z4(J,JJ)=(ZPT(KF)+ZPT(KG)+ZPAN(J))/3.0
      AF=XPT(KF)-XPAN(J)
      BF=YPT(KF)-YPAN(J)
      CF=ZPT(KF)-ZPAN(J)
      AG=XPT(KG)-XPAN(J)
      BG=YPT(KG)-YPAN(J)
      CG=ZPT(KG)-ZPAN(J)
      CALL SELF(AF,BF,CF,AG,BG,CG,FEE)
      SEL(J,JJ)=FEE
      CR=AF*BG-BF*AG
      AR=BF*CG-CF*BG
      BR=CF*AG-AF*CG
      ARE4(J,JJ)=0.5*SQRT(AR*AR+BR*BR+CR*CR)
1500  CONTINUE
      DO 127 NJ=1,NPAN
      DO 1277 MJ=1,NPAN
1277  PBB(NJ,MJ)=0.00
      P(NJ,1)=0.00
      P(NJ,2)=0.00
      P(NJ,3)=0.00
      P(NJ,4)=0.00
      P(NJ,5)=0.00
      P(NJ,6)=0.00
      DO 128 NK=1,4
      ARN=ARE4(NJ,NK)
      IF(ARN.LT.0.0) GO TO 128
      P1=0.0
      P2=0.0
      P3=0.0
      P4=0.0
      P5=0.0
      P6=0.0
      X=X4(NJ,NK)
      Y=Y4(NJ,NK)
      Z=Z4(NJ,NK)
      DO 138 MJ=1,NPAN
      DO 138 MK=1,4
      XF=X4(MJ,MK)
      YF=Y4(MJ,MK)
      ZF=Z4(MJ,MK)
      ARM=ARE4(MJ,MK)
      IF(ARM.LT.0.00) GO TO 138
      IF(NJ.NE.MJ) GO TO 140
      IF(MK.NE.NK) GO TO 140
      FRA=SEL(MJ,MK)/ARM
      GO TO 1380
```

COMMITTEE ON MARINE STRUCTURES
Commission on Engineering and Technical Systems
National Academy of Sciences - National Research Council

The COMMITTEE ON MARINE STRUCTURES has technical cognizance of the interagency Ship Structure Committee's research program.

Mr. Stanley G. Stiansen, Chairman, Riverhead, NY
Prof. C. Allin Cornell, Stanford Univesity, Stanford, CA
Mr. Peter A. Gale, Webb Institute of Naval Architecture, Glen Cove, NY
Mr. Griff C. Lee, Griff C. Lee, Inc., New Orleans, LA
Prof. David L. Olson, Colorado School of Mines, Goldon, CO
Mr. Paul H. Wirsching, University of Arizona, Tucson, AZ
Mr. Alexander B. Stavovy, Staff Officer, National Research Council, Washington, DC
CDR Michael K. Parmelee, Secretary, Ship Structure Committee, Washington, DC

LOADS WORK GROUP

Mr. Paul H. Wirsching, Chairman, University of Arizona, Tucson, AZ
Prof. Keith D. Hjelmstad, University of Illinois, Urbana, IL
Dr. Hsien Yun Jan, President of Martech Inc., Neshanic Station, NJ
Prof. Jack Y. K. Lou ,Texas A & M University, College Station, TX
Mr. Edward K. Moll, Bath Iron Works Corp., Bath, MA
Mr. Naresh Maniar, M. Rosenblatt & Son, Inc., New York , NY
Prof. Anastassios N. Perakis, The University of Michigan, Ann Arbor, MI

MATERIALS WORK GROUP

Prof. David L. Olson, Chairman, Colorado School of Mines, Golden,CO
Prof. William H. Hartt, Vice Chairman, Florida Atlantic University, Boca Raton, FL
Dr. Santiago Ibarra Jr., Amoco Corporation, Naperville, IL
Mr. Paul A. Lagace, Massachusetts Institute of Tech., Cambridge, MA
Mr. Mamdouh M. Salama, Conoco Inc., Ponca City, OK
Mr. James M. Sawhill, Jr., Newport News Shipbuilding, Newport News, VA
Mr. Thomas A. Siewert, National Bureau of Standards, Boulder, CO

SHIP STRUCTURE COMMITTEE PUBLICATIONS

SSC-321	Survey of Experience Using Reinforced Concrete in Floating Marine Structures by O. H. Burnside and D. J. Pomerening, 1984
SSC-322	Analysis and Assessment of Major Uncertainties Associated With Ship Hull Ultimate Failure by P. Kaplan, M. Benatar, J. Bentson and T. A. Achtarides, 1984
SSC-323	Updating of Fillet Weld Strength Parameters for Commercial Shipbuilding by R. P. Krumpen, Jr., and C. R. Jordan, 1984
SSC-324	Analytical Techniques for Predicting Grounded Ship Response by J. D. Porricelli and J. H. Boyd, 1984
SSC-325	Correlation of Theoretical and Measured Hydrodynamic Pressures for the SL-7 Containership and the Great Lakes Bulk Carrier S. J. Cort by H. H. Chen, Y. S. Shin & I. S. Aulakh, 1984
SSC-326	Long-Term Corrosion Fatigue of Welded Marine Steels by O. H. Burnside, S. J. Hudak, E. Oelkers, K. B. Chan, and R. J. Dexter, 1984
SSC-327	Investigation of Steels for Improved Weldability in Ship Construction by L. J. Cuddy, J. S. Lally and L. F. Porter 1985
SSC-328	Fracture Control for Fixed Offshore Structures by P. M. Besuner, K. Ortiz, J. M. Thomas and S. D. Adams 1985
SSC-329	Ice Loads and Ship Response to Ice by J. W. St. John, C. Daley, and H. Blount, 1985
SSC-330	Practical Guide for Shipboard Vibration Control by E. F. Noonan, G. P. Antonides and W. A. Woods, 1985
SSC-331	Design Guide for Ship Structural Details by C. R. Jordan and R. P. Krumpen, Jr., 1985
SSC-332	Guide for Ship Structural Inspections by Nedret S. Basar & Victor W. Jovino, 1985
SSC-333	Advance Methods For Ship Motion And Wave Load Prediction by William J. Walsh, Brian N. Leis, and J.Y. Yung, 1989
None	Ship Structure Committee Publications - A Special Bibliography, AD-A140339