AD-A039 510 DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/G 13/10
STRUCTURAL SEAWORTHINESS DIGITAL COMPUTER PROGRAM ROSAS (A CONV--ETC(U)
MAY 77 S CHUANG, E A SCHROEDER, S WYBRANIEC

UNCLASSIFIED DTNSRDC-77-0001 NL

1 OF 2
AD A039510



Report 77-0001

STRUCTURAL SEAWORTHINESS DIGITAL COMPUTER PROGRAM ROSAS (A CONVERSION FROM SEAWORTHINESS ANALOG COMPUTER)

AD A039510

DDC FILE COPY

DAVID W. TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CENTER

Bethesda, Md. 20084

STRUCTURAL SEAWORTHINESS DIGITAL COMPUTER PROGRAM ROSAS (A CONVERSION FROM SEAWORTHINESS ANALOG COMPUTER)

by

Sheng-Lun Chuang
Erwin A. Schroeder
Suzanne Wybraniec

DDC
RECEIVED
MAY 17 1977
C

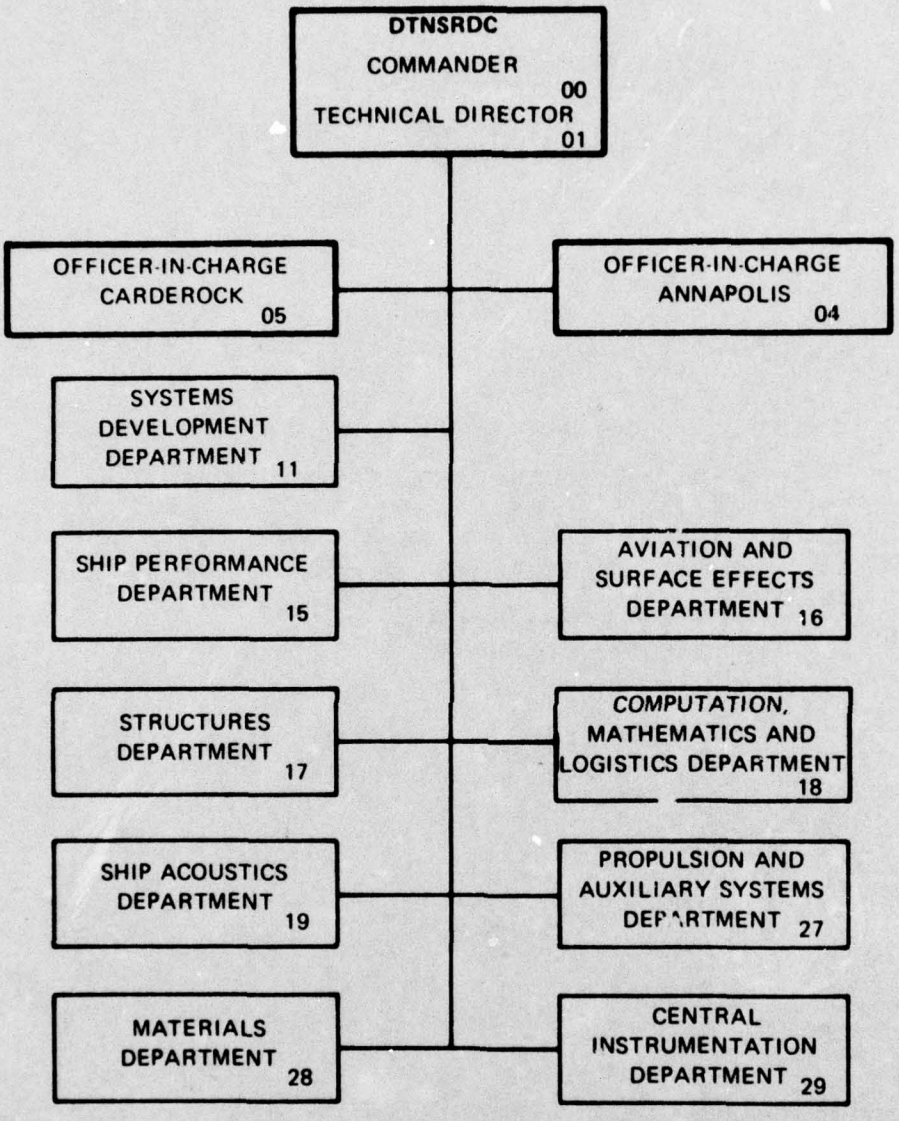
APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

STRUCTURES DEPARTMENT
AND
COMPUTATION, MATHEMATICS, AND LOGISTICS DEPARTMENT
RESEARCH AND DEVELOPMENT REPORT

May 1977

Report 77-0001

MAJOR DTNSRDC ORGANIZATIONAL COMPONENTS



GPO 905-678

DTNSRDC ISSUES THREE TYPES OF REPORTS

(1) DTNSRDC REPORTS, A FORMAL SERIES PUBLISHING INFORMATION OF PERMANENT TECHNICAL VALUE, DESIGNATED BY A SERIAL REPORT NUMBER.

(2) DEPARTMENTAL REPORTS, A SEMIFORMAL SERIES, RECORDING INFORMATION OF A PRELIMINARY OR TEMPORARY NATURE, OR OF LIMITED INTEREST OR SIGNIFICANCE, CARRYING A DEPARTMENTAL ALPHANUMERIC IDENTIFICATION.

(3) TECHNICAL MEMORANDA, AN INFORMAL SERIES, USUALLY INTERNAL WORKING PAPERS OR DIRECT REPORTS TO SPONSORS, NUMBERED AS TM SERIES REPORTS; NOT FOR GENERAL DISTRIBUTION.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: DTNSRDC [redacted]-77-0001
2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle): STRUCTURAL SEAWORTHINESS DIGITAL COMPUTER PROGRAM ROSAS (A CONVERSION FROM SEAWORTHINESS ANALOG COMPUTER)

5. TYPE OF REPORT & PERIOD COVERED

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s): Sheng-Lun Chuang, Erwin A. Schroeder, and Suzanne Wybraniec

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS: David W. Taylor Naval Ship Research and Development Center, Bethesda, Maryland 20084

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: Task Area SF 43 422 504, Work Unit 1-1730-315

11. CONTROLLING OFFICE NAME AND ADDRESS: Naval Sea Systems Command, Washington, D.C. 20362

12. REPORT DATE: May [redacted]77

13. NUMBER OF PAGES: 152

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Research and development rept.

15. SECURITY CLASS. (of this report): UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

DDC RELEASED MAY 17 1977 C

18. SUPPLEMENTARY NOTES

There is no warranty, expressed or implied, as to the correctness of the computer program or of its documentation contained in the report. The Government accepts no liability for the results of the program. While a

(Continued on reverse side)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Digital computer program
Ship dynamic response
Ship slamming and whipping
Program ROSAS

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The structural seaworthiness digital computer program ROSAS and users manual are presented in this report. The program was developed using FORTRAN computer language, and it simulates the hull girder structural response of a ship, including dynamic effects when it encounters head seas of the regular, irregular, discrete, standing or other wave forms. Response calculations include the ship rigid and elastic body motion, bending moment, →

next page

(Continued on reverse side)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

387 682

(Block 18 continued)
reasonable effort will be made to answer questions about the program, and a defective copy will be replaced on request, the Government does not undertake to provide maintenance in the future. It should be emphasized that all software, made available to other Government agencies, the private sector for domestic use, to foreign governments, or to business firms having contracts with foreign governments, will be accepted as documented, and no change in or maintenance of documentation shall be required.

(Block 20 continued)
cont → and shear. Vibratory hull girder modes can also be determined, and the effect of bow flare, bottom slamming and springing can be included. Computations have been compared with actual ship responses. The results are in good agreement with those obtained from actual ship sea trials, model experiments, a computer program developed by the Ship Structure Committee, and an earlier analog computer program developed by the David W. Taylor Naval Ship Research and Development Center. The program is a valuable tool for predicting hull girder response of new ship designs or to compare results of model or prototype data.

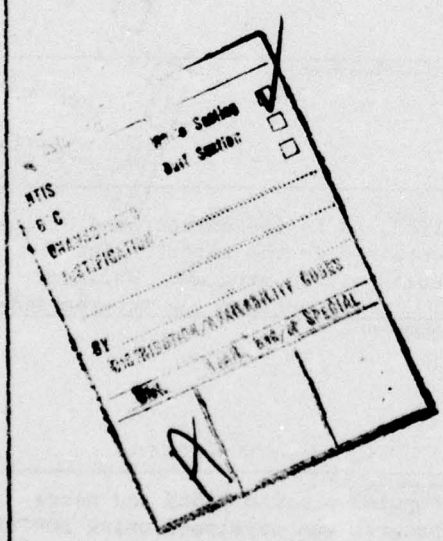


TABLE OF CONTENTS

	Page
ABSTRACT	1
ADMINISTRATIVE INFORMATION	1
INTRODUCTION	1
BACKGROUND	4
DIGITAL COMPUTER MODELING	5
HYDRODYNAMIC FORCE	6
SHIP AND SHIP RESPONSE	9
SEA GENERATION	15
BOTTOM SLAMMING	16
DISCUSSION OF RESULTS BY DIGITAL METHOD	17
SIMPLE HARMONIC EXCITATION	17
STEADY-STATE SINUSOIDAL WAVE EXCITATION	18
DISCRETE WAVE-TRAIN EXCITATION	20
RESPONSE AMPLITUDE OPERATORS	21
BOTTOM SLAMMING	25
SUMMARY AND CONCLUSIONS	30
ACKNOWLEDGMENTS	31
APPENDIX A - INPUT DATA FOR COMPUTER MODEL	33
APPENDIX B - STRUCTURAL SEAWORTHINESS DIGITAL COMPUTER PROGRAM ROSAS	53
REFERENCES	141

LIST OF FIGURES

1 - Diagram of Structural Seaworthiness Digital Computer Program ROSAS	6
2 - Method Used to Separate Linear and Nonlinear Buoyancy Forces.	8
3 - Method for Determining Added Mass for Immersion and Emersion.	9
4 - Ship Stations Used in Analysis	11

Page

5 - Actual Records for Ex-ESSEX (CVA-9) Sea Trials 20

6 - Response to Discrete Wave Train-Digital Program and Sea Trial . 22

7 - Response to Discrete Wave Train-Digital Program and Analog Computer 24

8 - Comparisons of Response Amplitude Operators Obtained by Different Methods 26

9 - Response to Bottom Slamming 29

10 - Curve for Estimating Coefficient J Used in Added Mass Evaluation 40

11 - Curves for Estimating Coefficient C_v Used in Added Mass Evaluation 40

12 - Nonlinear Buoyancy Forces 48

13 - Variation of Buoyancy Force with Draft 49

14 - Nonlinear Added Masses 50

15 - Hydrodynamic Damping Coefficients 52

16 - Structural Seaworthiness Digital Computer Program ROSAS 56

17 - Control Flow between Main Program SIMSHIP and Principal Subroutines of Structural Seaworthiness Digital Computer Program ROSAS 58

18 - Program SIMSHIP Flow Chart 59

19 - Subroutine SLAM 60

LIST OF TABLES

1 - Response to Sinusoidal Wave Train 19

2 - Ship Masses 34

3 - Determination of Sectional Vertical Area Moment of Inertia . . 36

4 - Structural Mass and Bending Rigidity 44

5 - Mass Moment of Inertia and Shear Rigidity 45

6 - Linear Added Mass, Buoyancy, and Smith Correction Factors . . . 46

Page

7 - Added Mass as a Function of Design Waterline of 28.5 Feet . . . 46

8 - Nonlinear Buoyancy Terms. 49

9 - Nonlinear Added Mass Terms 51

10 - Guide to ROSAS Subprograms. 61

11 - Functioning of Subroutines and Alternate Entries 63

12 - Structural Seaworthiness Digital Computer Program ROSAS 67

13 - Principal FORTRAN Variables for Subroutine HYD FRC 121

14 - Input Card Types 123

NOTATION

Symbol	Definition	Unit US	Unit SI
A	Cross sectional area of submerged portion of ship	ft^2	m^2
$\overline{A}$	Cross sectional area due to nonlinear portion of the buoyancy force	ft^2	m^2
A_0	Cross sectional area of ship to still waterline	ft^2	m^2
A_v	Cross sectional area of ship hull for vertical plating only	ft^2	m^2
a	Cross sectional area of structural member	ft^2	m^2
a_1, a_2, a_3	Arbitrary constant		
b	Half-width in general	ft	m
b_1	Ship beam at still waterline	ft	m
b_{1_n}	Ship beam at nth station at still waterline	ft	m
b_2, b_3	Arbitrary constant		
C	Ship structural damping coefficient		
$C(\omega)$	Real part of hydrodynamic damping coefficient per unit length of ship	$ton\text{-}s/ft^2$	$t\text{-}s/m^2$
C_V	Added mass coefficient		
c	Celerity of wave propagation	ft/s	m/s
c_1, c_2, c_3	Arbitrary constant		
d	(1) Distance in general; (2) depth of ship hull girder	ft	m
d_0	Distance of neutral axis	ft	m
E	Modulus of elasticity	ton/ft^2	t/m^2
EI	Bending rigidity	$ton\text{-}ft^2$	$t\text{-}m^2$
G	Modulus of rigidity, i.e., shear modulus of elasticity	ton/ft^2	t/m^2
GBRC	General bending response code		

Symbol	Definition	Unit US	Unit SI
g	Acceleration due to gravity	ft/s^2	m/s^2
Hz	Hertz	cps	cps
h	Double amplitude of wave	ft	m
I	Area moment of inertia	ft^4	m^4
I_0	Area moment of inertia of structural member about its own neutral axis	ft^4	m^4
I_{mz}	Mass moment of inertia about a transverse axis, through its center of mass, of a slice of ship of unit thickness	$ton\text{-}s^2$	$t\text{-}s^2$
J	Longitudinal inertia coefficient (Figure 10)		
KAG	Shear rigidity of ship hull, where K is the numerical factor depending upon the geometry of cross section $K \leq 1$, and A is the cross sectional area of side shell plating and continuous longitudinal bulkhead for carrying shear stress for vertical vibration	ton	t
K_b	buoyancy spring = $\rho g b_1 \Delta x/\Delta x$ per unit length of ship	ton/ft^2	t/m^2
L	Ship length	ft	m
M	Bending moment	ft-ton	m-t
$\bar{m}$	Time varying portion of added mass per unit length of ship	$ton\text{-}s^2/ft^2$	ts^2/m^2
m'_{0_n}	Added mass of ship half section associated with still waterline at nth station for a segment of station length	$ton\text{-}s^2/ft$	ts^2/m
m_0	Added mass associated with still waterline per unit length of ship	$ton\text{-}s^2/ft^2$	ts^2/m^2
m_s, m	Ship mass per unit length of ship	$ton\text{-}s^2/ft^2$	ts^2/m^2
m_V	Added mass of fluid (hydrodynamic mass) per unit length of ship	$ton\text{-}s^2/ft^2$	ts^2/m^2
m_1	Added mass during ship emergence per unit length of ship	$ton\text{-}s^2/ft^2$	ts^2/m^2

Symbol	Definition	Unit US	Unit SI
m_2	Added mass during ship immergence per unit length of ship	ton-s^2/ft^2	ts^2/m^2
n	Station number of ship		
P	Total hydrodynamic force per unit length of ship	ton/ft	t/m
P_1	Inertia force acting on a mass of fluid per unit length of ship	ton/ft	t/m
P_2	Dynamic portion of the buoyancy force per unit length of ship	ton/ft	t/m
P_3	Hydrodynamic damping force per unit length of ship	ton/ft	t/m
p	Bottom slamming pressure	psi	kg/cm^2
RAO	Response amplitude operator		
r	Radius of gyration	ft	m
rms	Root mean square		
SAC	Seaworthiness analog computer		
SCF	Smith correction factor		
SSDC	Structural seaworthiness digital computer program		
T	Wave period	s	s
t	Time coordinate on the ship or in the fluid lamina	s	s
U	Forward velocity of ship	ft/s	m/s
u	Forward velocity of fluid	ft/s	m/s
V	Shear force	ton	t
V_h	Relative horizontal velocity between ship and wave	ft/s	m/s
V_r	Relative vertical velocity between ship and wave surface	ft/s	m/s

Symbol	Definition	Unit US	Unit SI
v	Vertical velocity of wave surface	ft/s	m/s
W	Weight in general	ton	t
X	Space coordinate fixed in ship	ft	m
Y	Vertical translation of ship (↑+)	ft	m
Y_r	Relative vertical translation between ship and sea surface $=Y-Y_w$ = positive when ship moves out of water	ft	m
Y_w	Vertical translation of wave surface	ft	m
y	Waterline distance from baseline, i.e., ship draft	ft	m
β	Ship sectional area coefficient (Figure 8)		
γ	Angular displacement	rad	rad
ΔX	Length of one ship section	ft	m
$\Delta \ell$	Longitudinal length of bottom slamming area	ft	m
Θ_p	Pitch angle	rad	rad
Θ_w	Wave slope	rad	rad
λ	Wavelength	ft	m
μ	Mass per unit length	ton-s^2/ft	ts^2/m
ξ	Impact angle	rad	rad
ρ	Mass density of fluid	ton-s^2/ft^4	ts^2/m^4
ρ_s	Ship mass density	ton-s^2/ft^4	ts^2/m^4
Σ	Summation		
ω	Wave frequency	rad/sec	rad/s

U.S. CUSTOMARY–SI (INTERNATIONAL SYSTEM OF UNITS) CONVERSION FACTORS

The work contained in the report was performed before formal issuance of the metrication policy of the Center. Therefore, the U.S. customary units are used in the report instead of the international system of units (SI). The following list contains conversion factors to enable readers to compute the SI unit values of measurements.

1 ft = 30.48 cm
1 in = 25.40 mm

1 m = 39.37 in
= 3.281 ft

1 in^2 = 6.45 cm^2

1 cm^2 = 0.155 in^2

1 ft^3 = 28.32 liters
= 0.02832 m^3

1 m^3 = 1000 liters
= 35.31 ft^3

1 long ton = 1.016 t
= 2240 lb
1 lb = 0.454 kg (mass)
= 4.45 N (force)

1 newton (N) = (1/g) kg-m/sec^2
= (1/9.81) kg (force)
1 tonne (t) = 0.984 long ton
= 1000 kg

1 psi = 0.0703 kg/cm^2
= 6.9 kN/m^2

1 kg/cm^2 = 14.22 psi
1 N/m^2 = 1 pascal (Pa)

1 ft-lb = 0.1383 kg-m
= 1.356 N-m
1 Btu = 107.6 kg-m
= 778.3 ft-lb

1 kg-m = 7.23 ft-lb
= 9.807 N-m

1 hp = 0.746 kw
= 1.014 metric hp (ps)

1 ps = 0.735 kw
= 0.986 hp

ABSTRACT

The structural seaworthiness digital computer program ROSAS and users manual are presented in this report. The program was developed using FORTRAN computer language, and it simulates the hull girder structural response of a ship, including dynamic effects when it encounters head seas of the regular, irregular, discrete, standing or other wave forms. Response calculations include the ship rigid and elastic body motion, bending moment, and shear. Vibratory hull girder modes can also be determined, and the effect of bow flare, bottom slamming and springing can be included. Computations have been compared with actual ship responses. The results are in good agreement with those obtained from actual ship sea trials, model experiments, a computer program developed by the Ship Structure Committee, and an earlier analog computer program developed by the David W. Taylor Naval Ship Research and Development Center. The program is a valuable tool for predicting hull girder response of new ship designs or to compare results of model or prototype data.

ADMINISTRATIVE INFORMATION

This project has been funded and authorized by the Naval Ship Systems Command (035) under Subproject SF 43 422 504, Task 15939, Work Unit 1-1730-315.

INTRODUCTION

The design of a new seagoing ship has usually been based on past experience as well as rules and empirical formulas that are not too involved with higher mathematics. The rules and formulas, prepared by classification societies such as the American Bureau of Shipping, Lloyd's, and others are quite simple to apply and are the only guides needed for ship design by designing offices and shipyards. Such practice is considered reliable because, from year to year, rules and formulas are reviewed, revised, modified, and improved by a group of experienced and reputable engineers and specialists in the fields of ship operation, maintenance, repair, construction, and design.

If a ship design were to deviate from the so-called conventional type, adoption of the standard rules and formulas would be difficult. Therefore, when novel types of ships are being designed, the quasi-static balance method has been adopted to determine hull loads for the design. The technique is to put a ship on a fictitious wave-shaped sea surface and to balance the weight of the ship statically with the buoyancy force of the static wave. Calculations provide ship responses in bending so that the ship designer may determine, accordingly, the ship scantlings.

Calculation by this method is simple but cumbersome, if it is done by hand. With the aid of computers, the process becomes simple. However, a drawback to this method is the omission of dynamic effects of wave and ship motions.

To obtain more reliable information about ship responses, the present tendency is to use either a physical model tested in waves or a mathmetical model for an analytical solution. In solving a mathematical model analytically, one approach is to use an analog or a digital computer.

In the early 1960's, the David W. Taylor Naval Ship Research and Development Center (the Center) developed the seaworthiness analog computer (SAC). It is actually a complicated mathematical simulation model and consists of a sea generator, a ship analog, and a hydrodynamic force generator.[1] Output from the ship analog is fed back to the hydrodynamic force generator to produce dynamic interaction between ship and sea. Computations made to determine responses of an aircraft carrier, the Ex-USS ESSEX (CVA-9), to a specific wave train have been in good agreement with actual measurements made on the ship during sea trials. Unlike the quasi-static approach, this method includes both hydrodynamic effects and dynamic interaction in the analysis and provides as well a realistic representation of the ship response to sea waves during operations at sea.

Although SAC was considered one of the important developments toward realistic analysis in ship design, investment in a large analog computer facility for solving *only* ship response problems could not be justified

---

[1] Andrews, J.N., and S.-L. Chuang, "Seaworthiness Analog Computer," David Taylor Model Basin Report 1829 (Aug 1965). A complete listing of references is given on page 142.

economically. Since the Center has a digital computer facility, utilization of this facility is definitely a logical approach for solving this specific problem because any problem that can be solved by using the analog computer can also be solved by using the digital computer. Thus, the SAC facility never materialized.

Several unsuccessful attempts were made by others to convert SAC to a digital computer program; however, there were numerous obstacles. These obstacles have finally been overcome. Results obtained from the digital computer check very well not only with those obtained from SAC but also with those obtained from model tests and sea trials. This computer program is named the structural seaworthiness digital computer program ROSAS (i.e., response of ship at sea).

The program ROSAS in its present form has the capability of determining the following:

- Ship-hull-bending vibratory modes
- Ship rigid- and elastic-body motions, bending moment, shear, and other hull responses from a ship subjected to regular, irregular, discrete, standing, and other wave forms
- Ship RAO (response amplitude operator), rms (root-mean-square), and other statistical properties
- Effect of bow-flare-slamming in magnifying hull girder response
- Effect of bottom-slamming in magnifying hull girder response
- Effect of springing* in magnifying hull girder response

The listed capabilities are considered sufficient for use in the practical design of ship hull structures. While the present program is limited to head sea conditions, this limitation will be overcome in the near future because three-dimensional mathematical representations of sea and ship are now available but not fully developed.

The program ROSAS is presented in this report. An aircraft carrier (ESSEX) has been chosen to illustrate some of the capabilities of the program. Comparisons of the results of program ROSAS, SAC, model tests,

*Springing is a term generally applied to the pseudo-steady-state response of a ship hull in its fundamental, vertical, vibratory mode due to synchronous wave excitation. Springing is often known to generate significant hull stresses in such ships as Great Lakes bulk carriers, etc.; therefore, it requires proper accounting in design of those ships.

sea trials, and other sources are discussed and evaluated. Detailed presentations of program ROSAS are given in the appendixes. Also, methods for obtaining ship parameters used for input to the program are given in the appendixes.

In writing this report, it has been kept in mind that the program can readily be used by the readers for ship-design applications. However, because of the wide range of experience represented by individual users, it is impossible to cover every possible item needed for using the program.

BACKGROUND

A ship can develop appreciable hull stress, associated with transient vibration or "whipping" of the ship in heavy or moderate seas. This whipping may be generated by either emergence and subsidence of the bottom and subsequent impact or nonlinear buoyancy and momentum forces associated with bow-flare immergence. The substantial contributions from bow flareup that induced whipping stresses in the hull girder were vividly demonstrated during rough sea trials of ESSEX.[2]

A theoretical analysis was developed at the Center for investigating the whipping-response phenomenon. The analysis in essence utilized measured or calculated rigid body motion at each transverse section of the ship to compute the instantaneous waterline at each section as well as the velocity of the section relative to the waterline. Next, added mass for each section at each waterline was computed. Then, added mass force at each section was computed as the time rate of change of the momentum imparted by the water. This computation was added to the buoyancy and gravity forces to give total hydrodynamic force. Finally, response of the elastic ship was computed, thus giving the desired bending moments and shear forces.

---

[2]Jasper, N.H. and J.T. Birmingham, "Strains and Motions of USS ESSEX (CVA-9) During Storms Near Cape Horn," David Taylor Model Basin Report 1216 (Aug 1958).

Using this procedure, a detailed analysis of the whipping response of ESSEX was made by using a digital computer.[3] Although there were some discrepancies in details, the maximum stresses were predicted with reasonable accuracy, and the general agreement was considered good. However, changes in hull parameters could not be made easily with this method, and hand computation of the hydrodynamic forces before obtaining a computer solution was undesirable because it was time consuming. In particular, the need for having previous knowledge of ship motions severely limited the choice of mathematical model that would be suitable for design evaluations. These considerations motivated further development of a mathematical model to be computerized for easier handling of the ship dynamic response problem.

The early decision was made on the basis that the analog computer would be more suitable than the digital computer because the analog computer had greater flexibility in changing values of ship parameters and other variables. In fact, the Center possessed a passive analog computer named "network analyzer," that was used in making hull-vibration calculations.[4] Now scrapped, the network analyzer was considered excellent in its time. The abandonment of analyzer has necessarily stimulated conversion of SAC to the program ROSAS mentioned previously. Descriptions of SAC are given in Reference 1.

DIGITAL COMPUTER MODELING

Figure 1 shows the structural seaworthiness digital computer program ROSAS. Three principal elements that constitute the program are the hydrodynamic force, the ship, and the sea.

The sea subroutine is capable of simulating regular sinusoidal waves, a wave train of definite shape or simply a sinusoidal pulse at a prescribed location. The response of the ship feeds back to the hydrodynamic force

---

[3] Andrews, J.N., "A Method for Computing the Response of a Ship to a Transient Force," David Taylor Model Basin Report 1544 (Nov 1963).

[4] McGoldrick, R.T., "Ship Vibration," David Taylor Model Basin Report 1451; Figure 3-8, p. 3-18; Figures A-2 and A-3, p. A-3; Table 8-2, p. 8-10 (Dec 1960).

subroutine to produce dynamic interaction between the ship and the hydrodynamic forces. In case bow emergence occurs, a slamming subroutine computes the bottom slamming forces and adds them to the hydrodynamic forces for computing the ship responses.

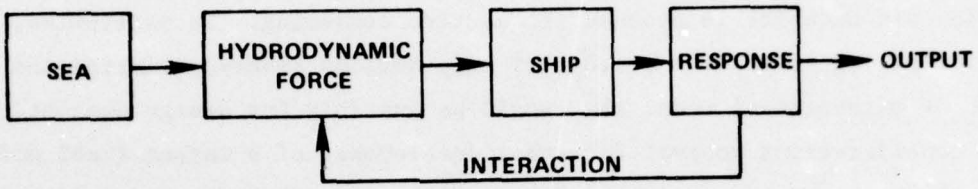


Figure 1 - Diagram of Structural Seaworthiness Digital Computer Program ROSAS

Mathematical representations of the hydrodynamic force, ship, sea, and slamming for digital computer modeling are presented in the following sections.

HYDRODYNAMIC FORCE

The hydrodynamic force consists of three types: added mass; buoyancy, including the Smith correction; and damping. These forces can be represented by the following equations:

$$P = P_1 + P_2 + P_3 \tag{1}$$

where P is the total hydrodynamic force

$$P_1 = \frac{d}{dt}\left(m_V V_r\right) \tag{2}$$

is the added mass force or the fluid inertial force

$$P_2 = \rho A\left(g + \frac{dv}{dt}\right) - \rho g A_0 \tag{3}$$

is the dynamic buoyancy force or spring force, and

$$P_3 = - C(\omega)\ V_r \tag{4}$$

is the hydrodynamic damping force.

The relative vertical velocity between the ship and the sea surface is

$$V_r = \frac{d}{dt}\ (Y - Y_w) = \frac{d}{dt}\ Y_r \tag{5}$$

The previous equations are expressed in terms of a coordinate system moving longitudinally with a fixed point in the fluid lamina. If they are expressed in terms of a coordinate system fixed in the ship, these equations can be approximated as follows, neglecting the horizontal component of the fluid velocity (Appendix A of Reference 1):

$$P_1 = - \frac{\partial}{\partial t}\ (m_V\ V_r) + U\ \frac{\partial}{\partial X}\ (m_V\ V_r) \tag{6}$$

$$P_2 = \rho g\ (A - A_0) + \rho A_0 \left(\frac{c}{c + U}\right)^2 \frac{\partial^2 Y_w}{\partial t^2} \tag{7}$$

$$P_3 = - C(\omega)\ V_r \tag{8}$$

$$V_r = \frac{\partial Y_r}{\partial t} - U\ \frac{\partial Y_r}{\partial X} \tag{9}$$

The added mass m_V and area A are separated into linear and nonlinear terms so that the effects of the nonlinearities may be examined to assess the importances of hull-form variations. These terms are

$$m_V = m_0 + \bar{m} \tag{10}$$

$$A = A_0 - b_1\ Y_r + \bar{A} \tag{11}$$

where m_0 is the added mass associated with the still waterline, and $\bar{m}$ is the time-varying portion of the added mass. The term $b_1\ Y_r$ is the rectangular area measured from the still to the instantaneous waterline, where Y_r

is the distance from the still to the actual waterline; $\bar{A}$ is the nonlinear portion of cross sectional area that produces the dynamic or nonlinear portion of buoyancy force; see Figure 2.

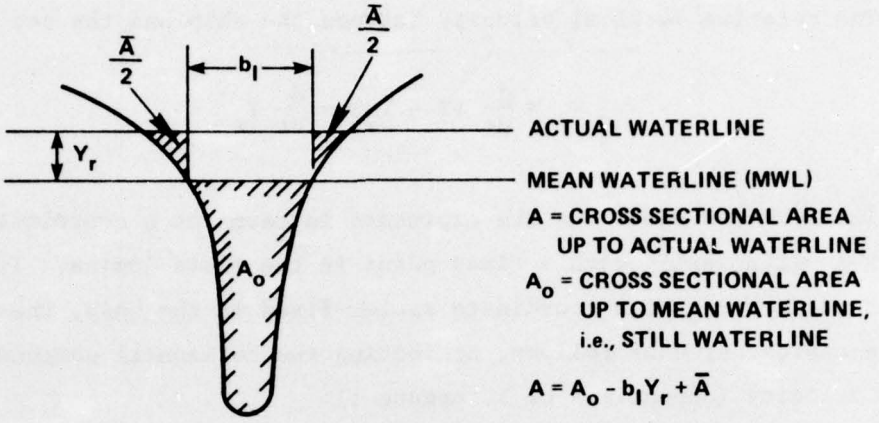


Figure 2 - Method Used to Separate Linear and Nonlinear Buoyancy Forces

The term $\bar{m}$ is defined by the following relationships

$$\bar{m} = m_1 \text{ for } V_r \geq 0 \text{ (emersion)} \tag{12}$$

$$= m_2 \text{ for } V_r < 0 \text{ (immersion)} \tag{13}$$

The two relationships are the result of the added mass being different, depending upon whether ship is immerging or emerging; see Figure 3.

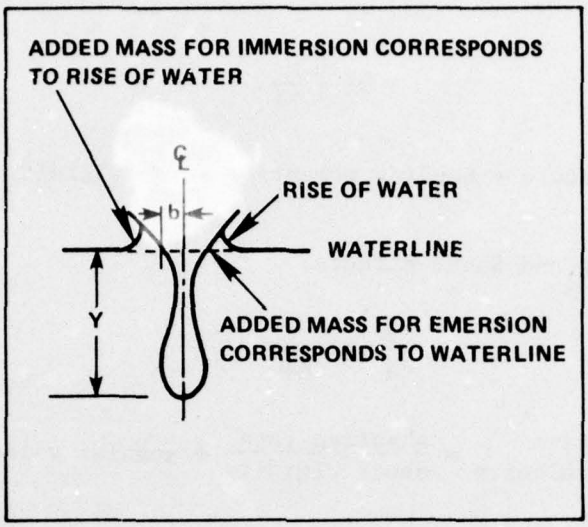


Figure 3 - Method for Determining Added Mass for Immersion and Emersion

SHIP AND SHIP RESPONSE

When the force function P (X, t) acts on the flexible ship, the equations governing the ship response comprise the following set.

Equation of Motion:

$$m \frac{\partial^2 Y}{\partial t^2} + C \frac{\partial \dot{Y}}{\partial t} + \frac{\partial V}{\partial X} = P \qquad (14a)$$

Inertia + damping + shearing = excitation

Moment Equation:

$$\frac{\partial M}{\partial X} = V + I_{mz} \frac{\partial \dot{\gamma}}{\partial t} \qquad (14b)$$

Spatial change of moment = shearing force + rotary inertia

Elastic Equation:

$$\frac{\partial \dot{Y}}{\partial X} = \frac{\dot{M}}{EI} \tag{14c}$$

Curvature = bending moment/flexural rigidity

Equation of Bending and Shear Effects:

$$\frac{\partial \dot{Y}}{\partial X} = -\frac{\dot{V}}{KAG} + \dot{\gamma} \tag{14d}$$

$$\text{Space derivative of vertical velocity} = \frac{\text{shearing rate}}{\text{shear rigidity}} + \text{angular velocity}$$

To discretize the variable X, 21 equally spaced stations are assigned along the length of the ship. The first station is assigned at the stern and the last at the bow. Twenty half-stations are also assigned, each midway between two stations. The stations are numbered from 0 to 20, and the half stations are numbered from 0.5 to 19.5; see Figure 4.

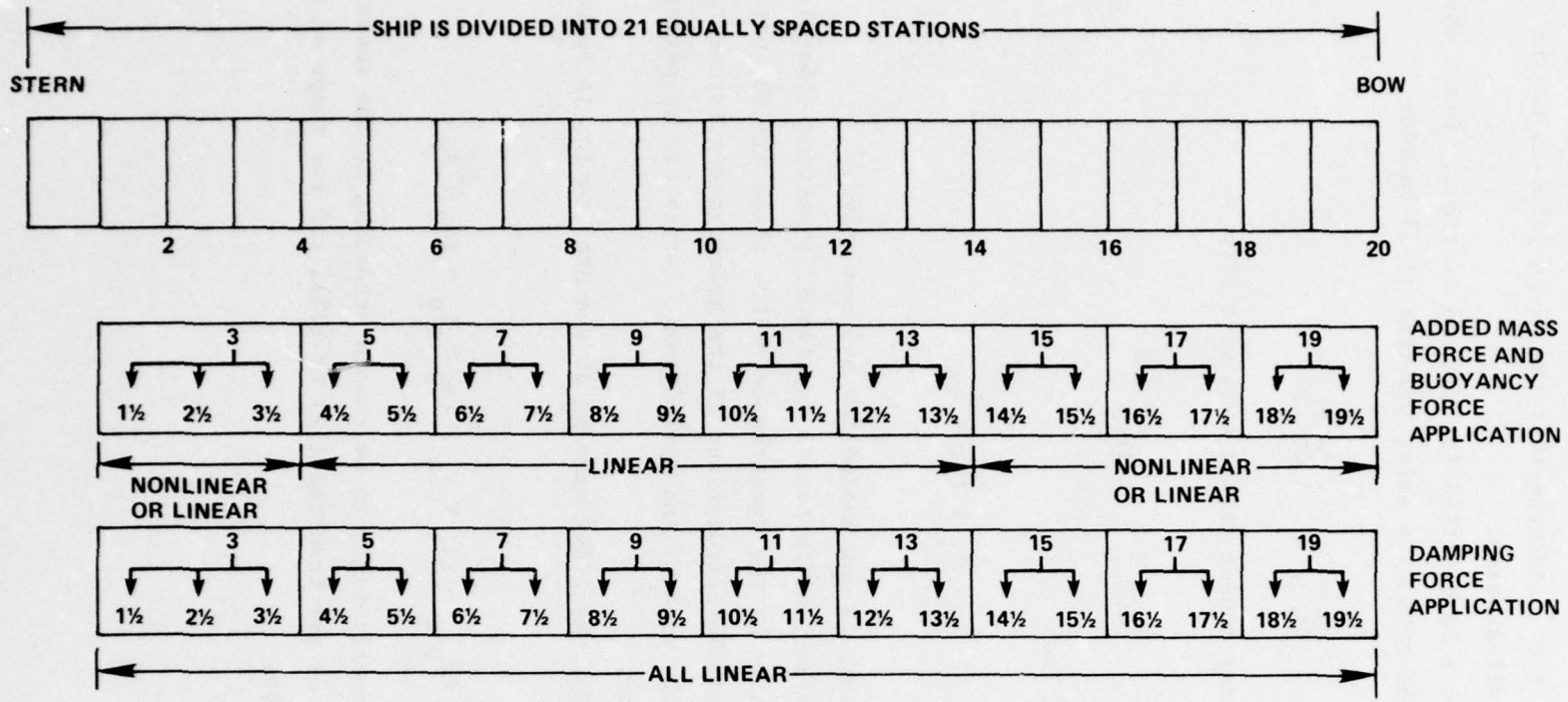
The values of $\dot{\gamma}$, V, KAG, and I_{mz} are lumped at the stations; the values of $\dot{Y}$, M, EI, m, C, and force P are lumped at the half-stations. With these quantities lumped and with the distance between stations denoted by ΔX, we can replace the derivatives with respect to X in the system of Equations (14a-d) by central difference quotients to get a system of ordinary differential equations. Thus,

$$\frac{d}{dt}\dot{Y}_{n+\frac{1}{2}} = (P_{n+\frac{1}{2}} - (C\dot{Y})_{n+\frac{1}{2}} - (V_{n+1} - V_n)/\Delta X)/m_{n+\frac{1}{2}} \tag{15a}$$

$$\frac{d}{dt}\dot{\gamma}_n = ((M_{n-\frac{1}{2}} - M_{n+\frac{1}{2}})/\Delta X - V_n)/(I_{mz})_n \tag{15b}$$

$$\frac{d}{dt}M_{n+\frac{1}{2}} = EI_{n+\frac{1}{2}}(\dot{\gamma}_{n+1} - \dot{\gamma}_n)/\Delta X \tag{15c}$$

$$\frac{d}{dt}V_n = KAG_n(\dot{\gamma}_n - (\dot{Y}_{n+\frac{1}{2}} - \dot{Y}_{n-\frac{1}{2}})/\Delta X) \tag{15d}$$



NOTE: AT STERN, HYDRODYNAMIC FORCES DISTRIBUTED EQUALLY TO STATIONS 1½, 2½, AND 3½. AT OTHER STATIONS, HYDRODYNAMIC FORCES DISTRIBUTED EQUALLY TO TWO ADJACENT HALF STATIONS.

Figure 4 - Ship Stations Used in Analysis

The subscripts n, $n+\frac{1}{2}$, etc., indicate that the given value is at the indicated station or half-station.

These equations must be modified for the end stations. Since there is no shearing force at the ends of a ship, Equation (15d) becomes

$$\frac{d}{dt} V_0 = 0 \; ; \; \frac{d}{dt} V_{20} = 0$$

Also, there is no bending moment at the ends of a ship

$$M_{0.5} \approx 0 \; ; \; M_{19.5} \approx 0$$

thus Equation (15c) implies

$$\dot{\gamma}_0 \approx \dot{\gamma}_1 \; ; \; \dot{\gamma}_{20} \approx \dot{\gamma}_{19}$$

Therefore, Equation (15b) is not needed for n=0 and n=20.

The system of ordinary differential equations, Equations (15a-d), can be integrated numerically by the Runge-Kutta method after all the parameters have been evaluated and the calculations of the hydrodynamic forces have been completed. Calculations of the hydrodynamic forces can be performed in the following manner.

From Equation (10), the added mass m_V at the nth station is approximated by the expression

$$(m_V)_n = (m_0)_n + (a_1 Y_r)_n + (a_2 Y_r^2)_n + (a_3 Y_r^3)_n \qquad (16)$$

where a_1, a_2, a_3 are coefficients to be chosen according to the emersion or immersion of the ship as per Equation (12) or (13), and the shape of ship section at that station.

In Equations (5) and (9), the vertical velocity at nth station relative to the water surface is approximated by

$$(V_r)_n \approx \left(\frac{\partial}{\partial t} Y_r\right)_n - U \left(\frac{\partial}{\partial X} Y_r\right)_n$$

$$\approx \dot{Y}_n - \left(\frac{\partial}{\partial t} Y_w\right)_n - \frac{U}{4\Delta X} ((Y_r)_{n+2} - (Y_r)_{n-2}) \qquad (17)$$

The relative displacement of the moving waterline with respect to the still waterline is given by $Y_r = Y - Y_w$. The vertical displacement of water surface Y_w and its vertical velocity $\partial Y_w/\partial t$ are provided by a subroutine that simulates the sea. The vertical displacement of ship Y is obtained by adding the equation $dY/dt = \dot{Y}$ to the system of Equations (14). The sign of Y_r is taken as positive when the ship at nth station moves out of the water and away from the water surface. The ship speed is prescribed.

At Station 3 Equation (17) is modified, since for $(Y_r)_1 = Y_1 - (Y_w)_1$, $(Y_w)_1$ is not available. The modified equation for Station 3 is

$$(V_r)_3 \approx \dot{Y}_3 - \left(\frac{\partial}{\partial t} Y_w\right)_3 - \frac{U}{2\Delta X} ((Y_r)_5 - (Y_r)_3) \qquad (18)$$

A similar modification is made for Station 19.

Combining Equations (2), (6), and (17), the hydrodynamic inertial force due to the change in momentum of the added mass $\frac{d}{dt} (m_V V_r)$ at the nth station is

$$(P_1)_n = - \frac{d}{dt} (m_V V_r)_n$$

$$= - \frac{\partial}{\partial t} (m_V V_r)_n + U \frac{\partial}{\partial X} (m_V V_r)_n$$

$$= - (m_V \frac{\partial}{\partial t} V_r)_n - (V_r \frac{\partial}{\partial t} m_V)_n + U \frac{\partial}{\partial X} (m_V V_r)_n$$

$$\approx - (m_V)_n (\ddot{Y}_n - (\ddot{Y}_w)_n - \frac{U}{4\Delta X} ((\dot{Y}_r)_{n+2} - (\dot{Y}_r)_{n-2}))$$

$$- (V_r)_n \ (a_1 + 2a_2 \ Y_r + 3a_3 \ Y_r^2)_n \ (\dot{Y}_r)_n$$

$$+ \frac{U}{4\Delta X} \left((m_V \ V_r)_{n+2} - (m_V \ V_r)_{n-2}\right) \qquad (19)$$

with the appropriate modifications for Stations 3 and 19 similar to Equation (18).

From Equation (11), the submerged cross sectional area at a station is approximated by the expression

$$A = A_0 - b_1 \ Y_r + b_2 \ Y_r^2$$

The coefficients b_1 and b_2 are chosen according to the vertical displacement of the station relative to the water surface. Substituting the previous expression into Equation (7), we obtain the expression for the dynamic buoyancy force as follows, i.e.,

$$P_2 = \rho g \ (- b_1 \ Y_r + b_2 \ Y_r^2) + \rho A_0 \left(\frac{c}{c + U}\right)^2 \frac{\partial^2 Y_w}{\partial t^2} \qquad (20)$$

The values of c and $\partial^2 Y_w / \partial t^2$ are provided by the subroutine that simulates the sea.

The terms $m \ddot{Y}$ in Equation (14a) and $m_V \ddot{Y}$ in Equation (19) can be transposed by adding m_V to m to form the term $(m + m_V) \ddot{Y}$ as part of Equation (14a), since Equation (19) is a part of Equation (14a).

At the present state of the art, there is no accurate method to determine the damping coefficients for structures and fluids. The complex structural and load distribution of a ship make determination of damping coefficients cumbersome. The best results that may be expected will show good correlation between calculated and test-determined values for the fundamental mode of ship hull only. This is also true for fluid damping. Fortunately, the inaccuracy of the damping coefficient will not much affect the maximum magnitude and frequency of the fundamental mode between the computer model and the actual ship. Therefore, only an approximation method will be adopted for the present computer program. This of course can easily be improved later when a more accurate method is available.

The total hydrodynamic force is then the sum of P_1 given by Equation (19) for inertial force, P_2 given by Equation (20) for buoyancy force, and P_3 given by Equation (4) for hydrodynamic damping. Information about the total hydrodynamic force is needed to integrate the system of Equations (15).

SEA GENERATION

Three types of seas are used for computer input. They are simple harmonic excitations, sinusoidal seas, and a specific discrete wave train. Simple harmonic excitation is a sinusoidal exciting force applied at a prescribed location of the ship, e.g., Station 10. Therefore, it does not represent an actual sea condition but is used to determine frequencies of the hull girder at various modes. The sinusoidal sea is a sinusoidal wave train moving without change in form and at a constant wave velocity from the bow to the stern of the moving ship. This is approximated by a delay function that caused the wave train to appear at points along the ship with a time delay equal to the distance from the bow, divided by the sum of ship speed and wave velocity. Sinusoidal sea excitation permits determination of RAO, an important tool for statistical analysis in ship design. The discrete wave train is composed of a series of sinusoidal waves by superposition to curve fit and wave data from the sea trials. This is used to verify dependability of program ROSAS by comparing computer output with sea-trial test results.

No random sea excitation was employed. If this is needed, it is necessary to start with statistical representation of a random sea and then to convert it to random sea excitation in the time domain. From the random sea excitation, the ship responses can be obtained and be converted into the statistical representation for the lifetime prediction of the ship.

The same results can be obtained with response amplitude operators. Of course, this is a more direct and shorter process than the method given previously.

BOTTOM SLAMMING

Bottom slamming occurs when the ship bottom is on or above water during immersion, i.e.,

$$\left.\begin{array}{l}(1)\ Y_r \geq y \\ (2)\ V_r \leq 0\end{array}\right\} \qquad (21)$$

Both conditions (1) and (2) must be satisfied to generate bottom slamming. Slamming starts at a location where $Y_r=y$, and $V_r=0$. It ends at a time where $Y_r=y$ with $V_r<0$, and $Y_r<y$ elsewhere.

At any t during slamming, the impact area can generally be assumed to be triangular in shape, and the load is estimated to be

$$\text{Load} = \Sigma p = \frac{1}{2}\, p_{max}\, \Delta\ell\, b_1\ (144/2240)$$

where $\Delta\ell$ is the longitudinal distance from keel to bilge where the water surface intersects with the ship bottom.

b_1 is the width of ship bottom at impact

p_{max} is the maximum impact pressure. The method for determining p_{max} is given in Reference 5. The slamming load is proportionally added to two adjacent stations together with the hydrodynamic forces for ship response.

---

[5] Chuang, S.-L. et al., "Experimental Investigation of Catamaran Cross-Structure Slamming," NSRDC Report 4653 (Sep 1975).

DISCUSSION OF RESULTS BY DIGITAL METHOD

Since the program ROSAS has newly been converted from SAC, it is necessary to verify its feasibility and accuracy. For comparison, ROSAS was programed with three excitations the same as those used by SAC, i.e., simple harmonic excitation, sinusoidal seas, and discrete wave trains. Slamming response is added to the program. In addition, RAO's for the ship are also obtained and are compared with other methods.

Since sea trials were made with ESSEX, which was simulated on SAC, the ship was used to test the performance of the program ROSAS also. The structural and hydrodynamic parameters that describe ESSEX are taken from Reference 1. To facilitate the use of the program, Appendix A is provided for determining these parameters.

SIMPLE HARMONIC EXCITATION

Vibration modes of the ESSEX hull were observed during sea trials. Thus it has been possible to check the program by comparing vertical bending modes obtained from simple harmonic excitations of the computer model with those of the full-scale ship. The tests also consist of comparing the frequencies of fundamental bending modes by ROSAS with those by GBRC (the general bending response code).[6] and by SAC. The frequency of a fundamental mode was determined using the program ROSAS by searching for the frequency of a point of sinusoidal exciting force that produced the largest response.

Three types of test were made, represented as follows:

Case 1 - Only ship structural mass was used; no hydrodynamic forces were applied.

Case 2 - Added mass was added to structural mass; however, no hydrodynamic buoyancy force was applied.

Case 3 - Added mass was added to structural mass, and hydrodynamic buoyancy force was applied.

---

[6] Henderson, F., "Transient Response Calculation in the Frequency Domain with General Bending Response Program (GBRP)," NSRDC Report 3613 (Feb 1971).

Neither structural nor hydrodynamic damping has been added for the three cases because the damping force does not influence or affect the frequency of fundamental mode very much in actual hull vibration of the ship.

Analog computer results are available only for Cases 2 and 3; sea trial results also are available for Case 3. For Case 3 only the linear part of the buoyancy force could be simulated with GBRC. For all three cases, the ship speed was zero, and calm seas were used. Frequencies in hertz for the fundamental mode of the ship hull obtained from various methods are compared as follows:

Case	ROSAS	GBRC	SAC	Sea Trials
1	1.043 Hz	1.043	-	-
2	0.742	0.750	0.734	-
3	0.749	0.763	0.744	0.825

Frequencies determined by digital simulation of ROSAS agree with those determined by the other simulations within 2 percent. However, the frequency determined by ROSAS is about 91 percent of that obtained from the sea trial. The value of 0.825 Hz was calculated from the record given in Figure 5a of Reference 2. This discrepancy in frequency is considered reasonable for full-scale measurement at sea.

STEADY-STATE SINUSOIDAL WAVE EXCITATION

The actual regular sea wave form can generally be represented to a close approximation by a sinusoidal wave, which is much easier to handle mathematically than the usual approximation by a trochoidal form. Therefore, for this type of test, the sea was represented by a sinusoidal wave train moving at a constant wave velocity from the bow to the stern of the ship. The theoretical wave velocity c is

$$c = g/\omega$$

which generally agrees with the values obtained from observations at sea.

The test results for this type of excitation were available only in Reference 1, i.e., by SAC. Unfortunately, only the linear analysis was performed for this type of excitation. The ship responses reported in Reference 1 were the vertical displacement of the ship at a station relative to the surface of the sea, the pitch angle of the ship, the hydrodynamic force acting at a station, and the bending moment at a station. The test results for Case 6 of Reference 1 and ROSAS are compared in Table 1. In this case, 0.35 rad/s of the wave frequency and 16 knots of the ship speed were programed.

TABLE 1 - RESPONSE TO SINUSOIDAL WAVE TRAIN*

	Station**	Phase Angle***		Amplitude	
		Digital	Analog	Digital	Analog
Wave Height in Feet	3	147	145	10	9.74
	11	71	80	10	9.97
	19	0	0	10	9.88
Relative Displacement in Feet	3	167	114	1.53	2.29
	7	247	283	3.04	1.52
	11	230	270	2.42	1.21
	15	107	87	2.72	2.58
	17	84	79	5.82	5.91
	19	72	74	9.65	9.81
Pitch Angle in Degrees	10.5	9	3	2.02	1.97
Hydrodynamic Force in Tons	3	317	285	162	281
	15	267	242	289	225
	17	248	242	289	275
	19	239	239	135	135
Bending Moment in Foot-Tons	4	247	275	$23.9 \cdot 10^3$	$37.6 \cdot 10^3$
	8	252	263	$104 \cdot 10^3$	$126 \cdot 10^3$
	10	268	257	$121 \cdot 10^3$	$135 \cdot 10^3$
	12	244	252	$112 \cdot 10^3$	$116 \cdot 10^3$
	16	237	240	$37.5 \cdot 10^3$	$35.2 \cdot 10^3$

*Linear analysis: 16 knots of ship speed, 0.35 rad/s of wave frequency.
**Station 0 is at stern; Station 20, at bow.
***Angle given is the relative phase of the measured quantities with respect to wave position at the bow.

The values obtained from ROSAS show trends similar to the values reported for SAC and are of the same order of magnitude. A trend pointed out in Reference 1, and apparent in ROSAS results, is that the relative displacement is large at the bow and small at the stern, and the phase angle of the relative displacement at the stern is close to that of the wave.

The steady-state sinusoidal wave excitation is a very important tool for analyzing ship responses. Further application of this method will be discussed and evaluated in the section about response amplitude operators.

DISCRETE WAVE-TRAIN EXCITATION

For this kind of test, the surface of the sea was represented by an approximation to a discrete wave train recorded during ESSEX sea trials; see Figure 5.[2] Wave height, pitch angle, and midship-bending stress were recorded at the same time. The discrete wave record of the sea trials was so selected that the ship produced whipping of the hull girder.

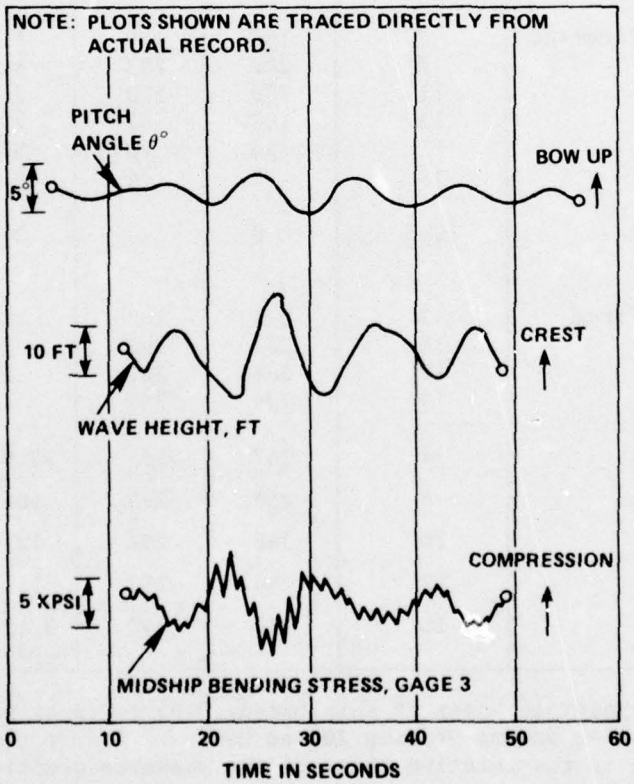


Figure 5 - Actual Records for Ex-ESSEX (CVA-9) Sea Trials (Figure 5a, Reference 2)

The approximated wave heights for both ROSAS and SAC were smoothly connected to a long period of steady sinusoidal waves to avoid extraneous whipping excitation. Thus, the approximations had the same general shape as the recorded large waves but were different in details.

Figure 6 compares wave heights, pitch angles, and midship bending moments obtained from ROSAS and sea trials; Figure 7 compares similar results obtained from ROSAS and SAC. As expected, the ship responses recorded during the sea trials and computed by ROSAS and SAC are similar in character but different in details. The characteristic common to all is that the large discrete wave generated large bow immersions to excite whipping of the hull girder. Results show good agreement in maximum magnitudes among them in ship responses. However, in the sea trial data, a previous excited whipping was not quite damped out before the next excitation began. Since the damping coefficient used in the digital program was not determined from an actual ship, the excited whippings were damped out at different rates for the ship and the computer model.

It was evident that the ship was excited when the bow was pitched steeply down to the high wave. Whipping was obviously due to the pronounced bow flare of the ship section, resulting in an impulse at the ship bow. Bow flare is not the same as bottom slamming, even though both types of impulse would generate whippings of the hull girder. Bottom slamming is more common than bow flare for most ships, and ROSAS in its present form is also programed for bottom slamming, which will be discussed later.

RESPONSE AMPLITUDE OPERATORS

The irregular sea surface may be represented by the sum of a great number of small-amplitude sine waves having different directions and periods. (Wave length and period have a fixed relation.) Ship response to an irregular seaway may be represented by the sum of the ship responses to the simple sine wave components. This linear superposition of random wave theory is known as the energy spectrum analysis and applies not only for ship responses to random sea waves but also to any form of energy such as heat, electricity, light, sound, and many mechanical phenomena.

Figure 6 - Response to Discrete Wave Train-Digital Program and Sea Trial

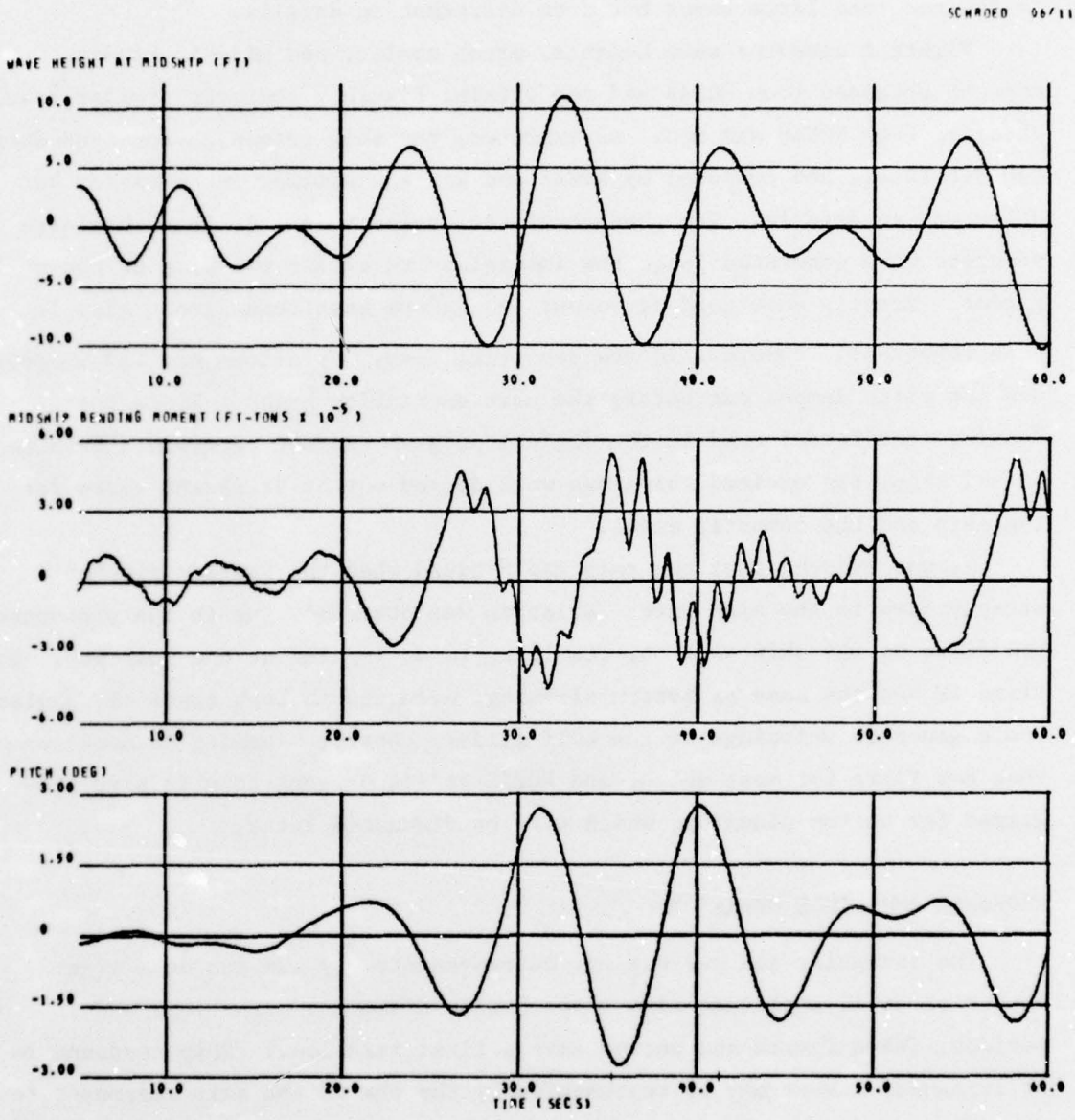


Figure 6a - Actual Output from Program ROSAS

Figure 6 (Continued)

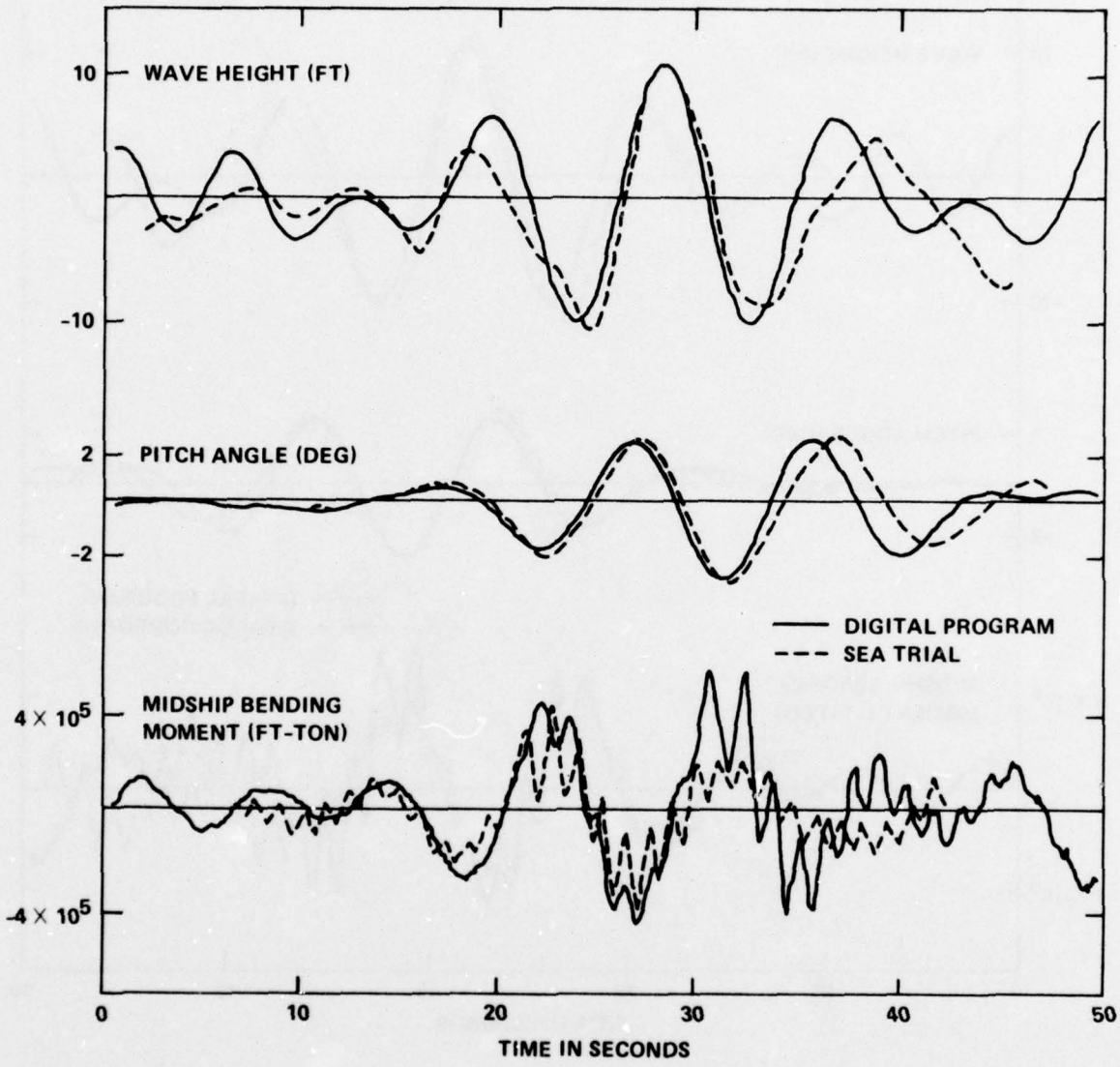


Figure 6b - Comparisons Between ROSAS Output and Sea Trial Data

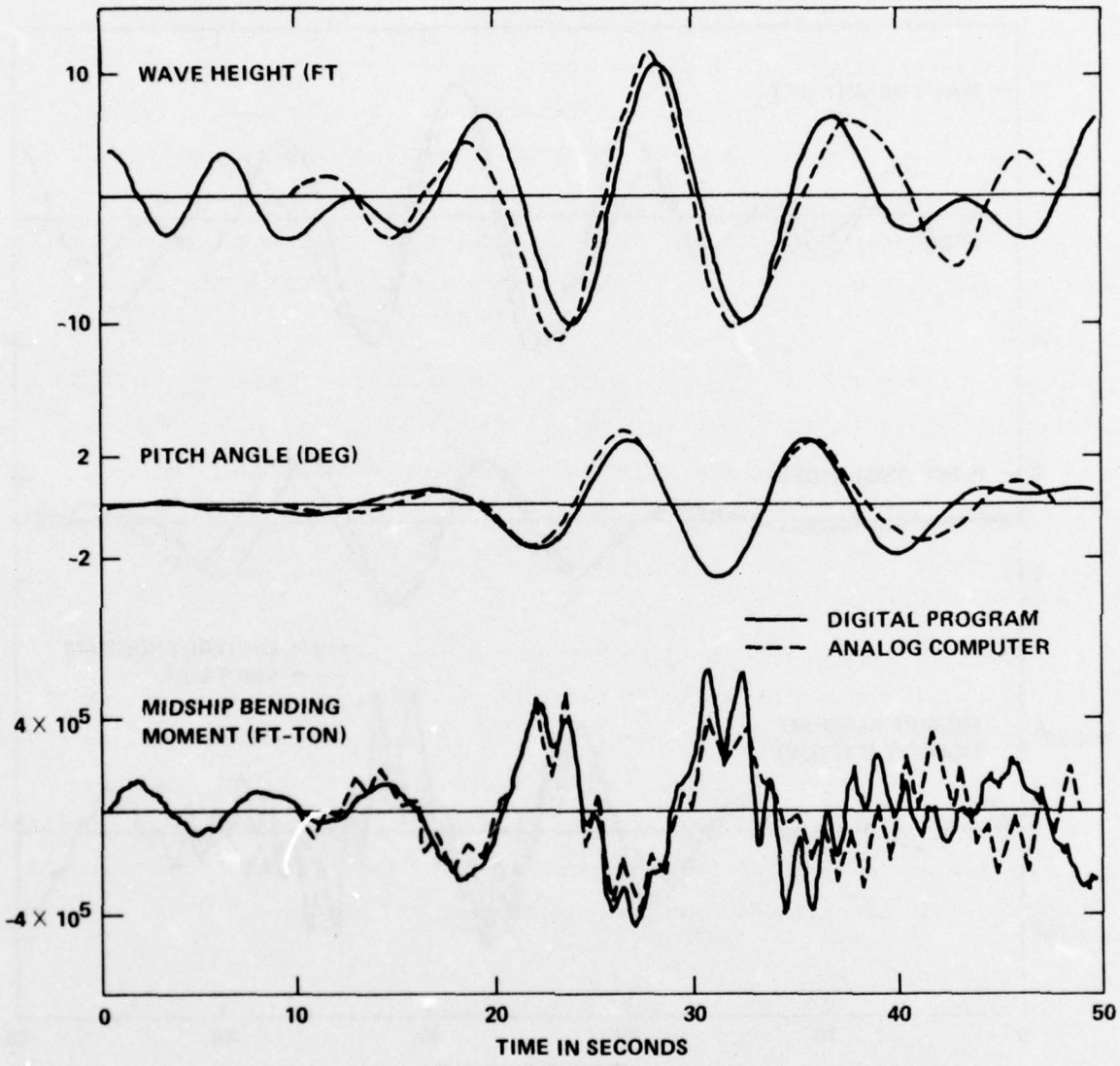


Figure 7 - Response to Discrete Wave Train-Digital Program and Analog Computer

Even though the theory seems to be complex, it may comfort the naval architect and ship designer to know that this powerful tool can be applied in ship design without a detailed understanding of the unusual mathematics. Any seaway can be characterized simply by a "wave energy spectrum." When a squared RAO curve is multiplied by the wave spectrum, a ship response spectrum is obtained. This can be pitch, heave, bending moment, or any other ship response. To obtain the response amplitude operator: First, apply one sine wave at a time to the ship to get the ship response. Second, divide each ship response by the single sine wave; the result is RAO. This operation usually requires the physical model test under regular or irregular sea conditions; now, however, it can be obtained directly by the program ROSAS.

Figure 8 compares RAO's for pitch angle and bending moment of ESSEX among the physical model tests,[7] the program SCORES,[7] and ROSAS results. Very good agreement is shown. The SAC is capable of obtaining RAO's. However, it was not included in Reference 1.

BOTTOM SLAMMING

The ship chosen for the present example has deep draft, and its bow does not come out of the water at all. As an illustration, the ship draft was drastically reduced in the program so that bottom slamming occurred during ship operations. Figure 9 shows bottom slamming output from the computer; nonlinear terms have been omitted in the program to avoid bow-flare slamming. Since no actual bottom-slamming data have been obtained from the sea trials of ESSEX, no comparison can be made at this time.

---

[7] Kaplan, P. et al., "An Investigation of the Utility of Computer Simulation to Predict Ship Structural Response in Waves," Ship Structure Committee Report SSC-197 (Jun 1969).

Figure 8 - Comparisons of Response Amplitude Operators Obtained by Different Methods

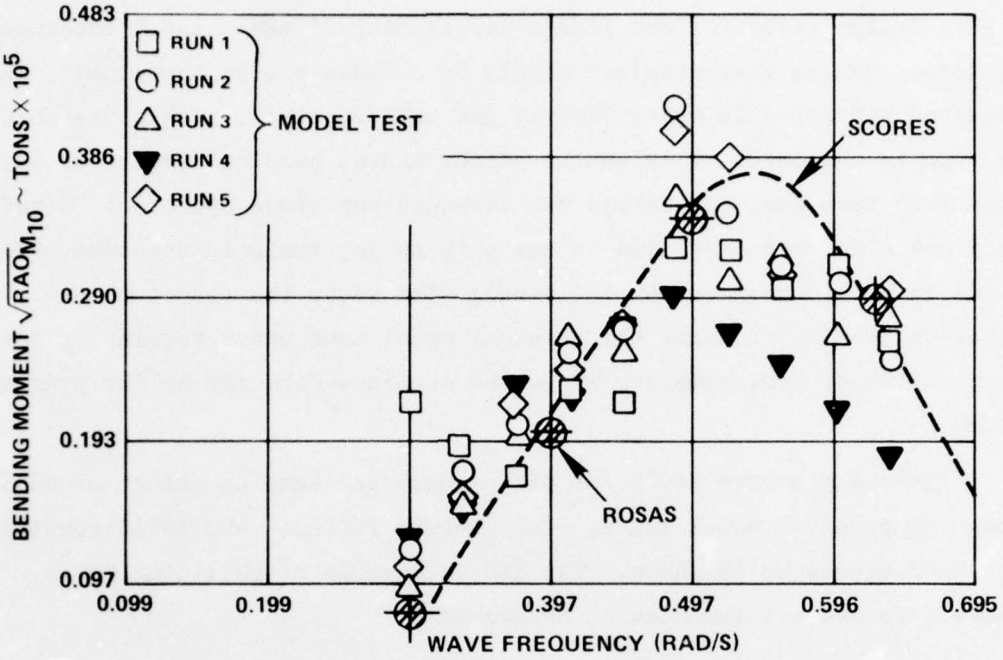


Figure 8a - Square Root of Bending Moment $RAO_{M_{10}}$, Zero Speed

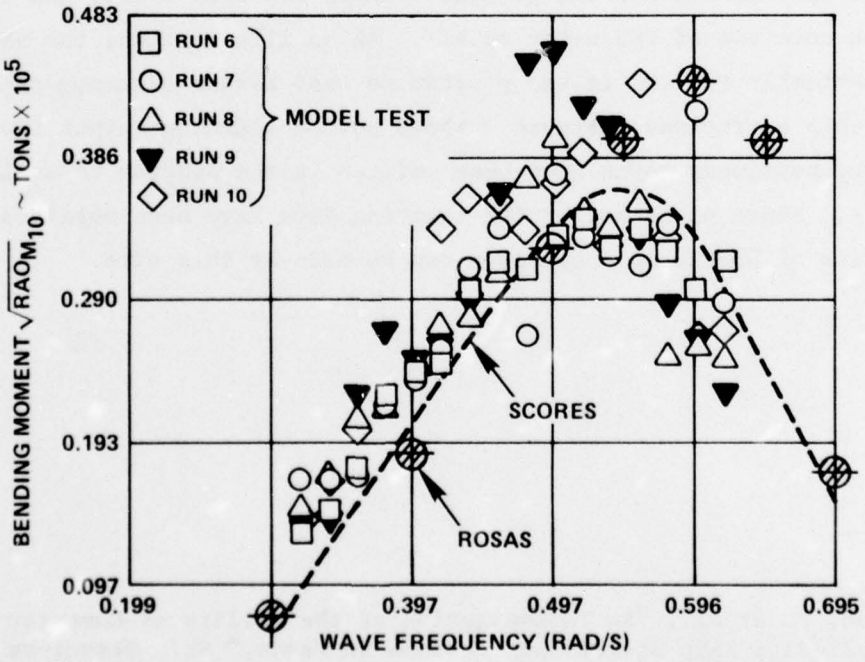


Figure 8b - Square Root of Bending Moment $RAO_{M_{10}}$, 13.8 Knots

Figure 8 (Continued)

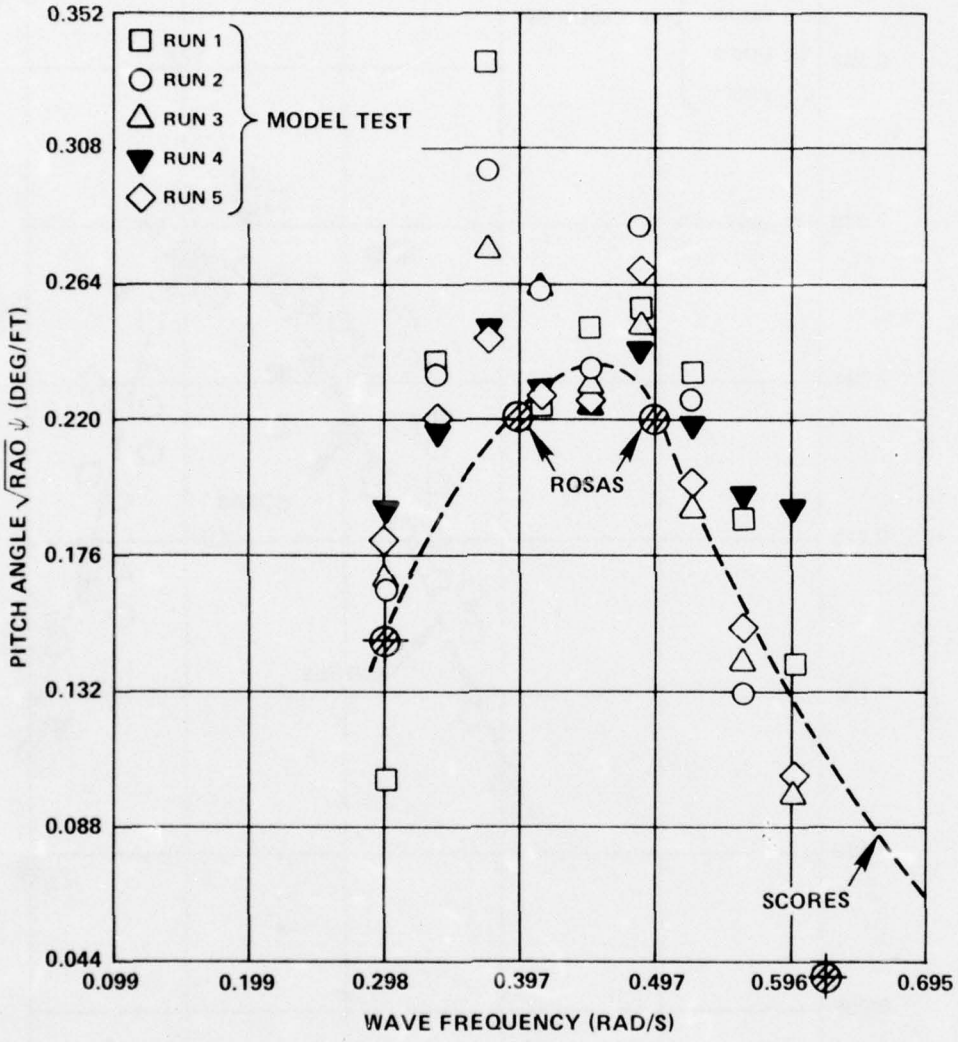


Figure 8c - Square Root of Pitch Angle RAO_{ψ}, Zero Speed

Figure 8 (Continued)

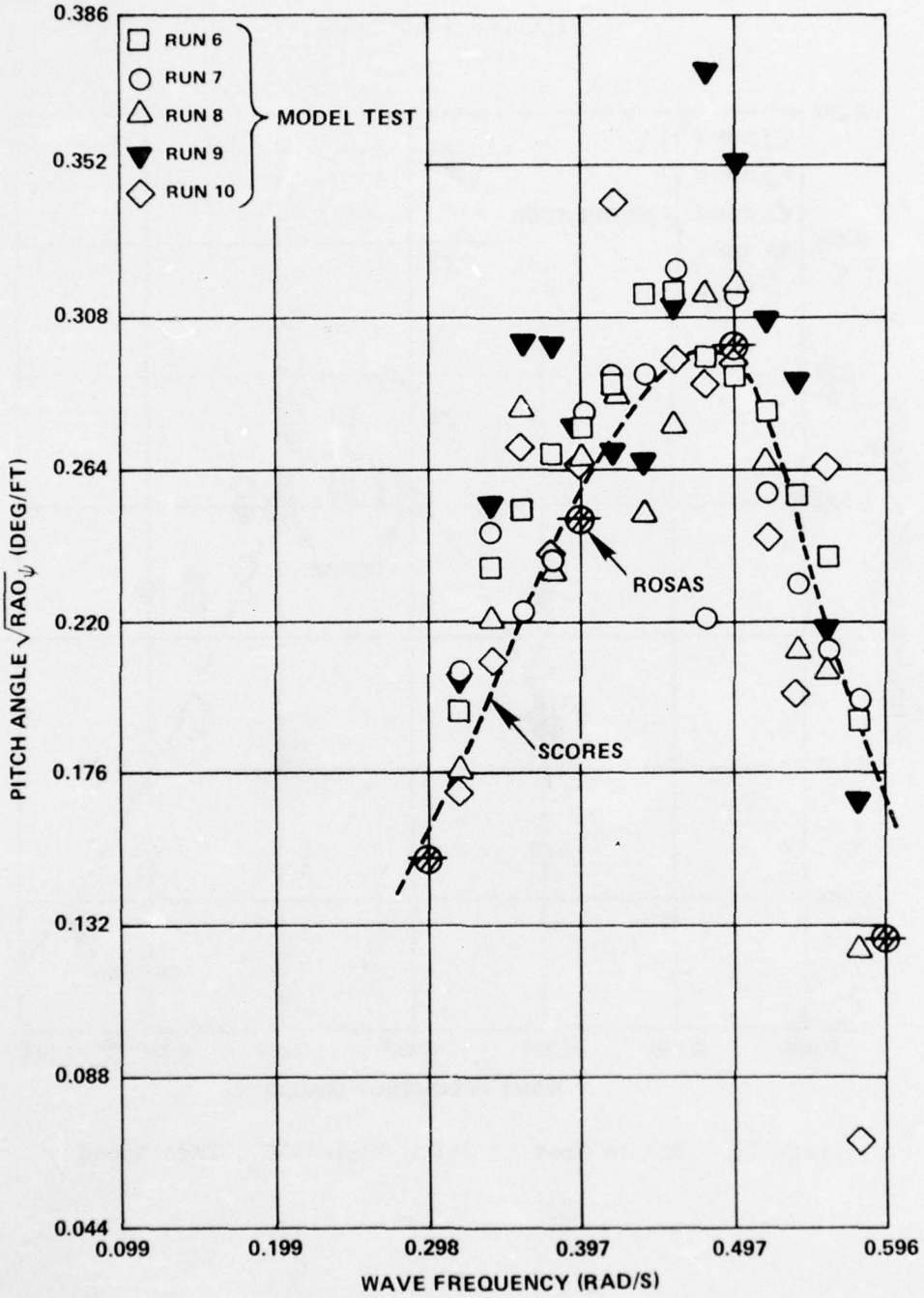


Figure 8d - Square Root of Pitch Angle RAO_ψ, 13.8 Knots

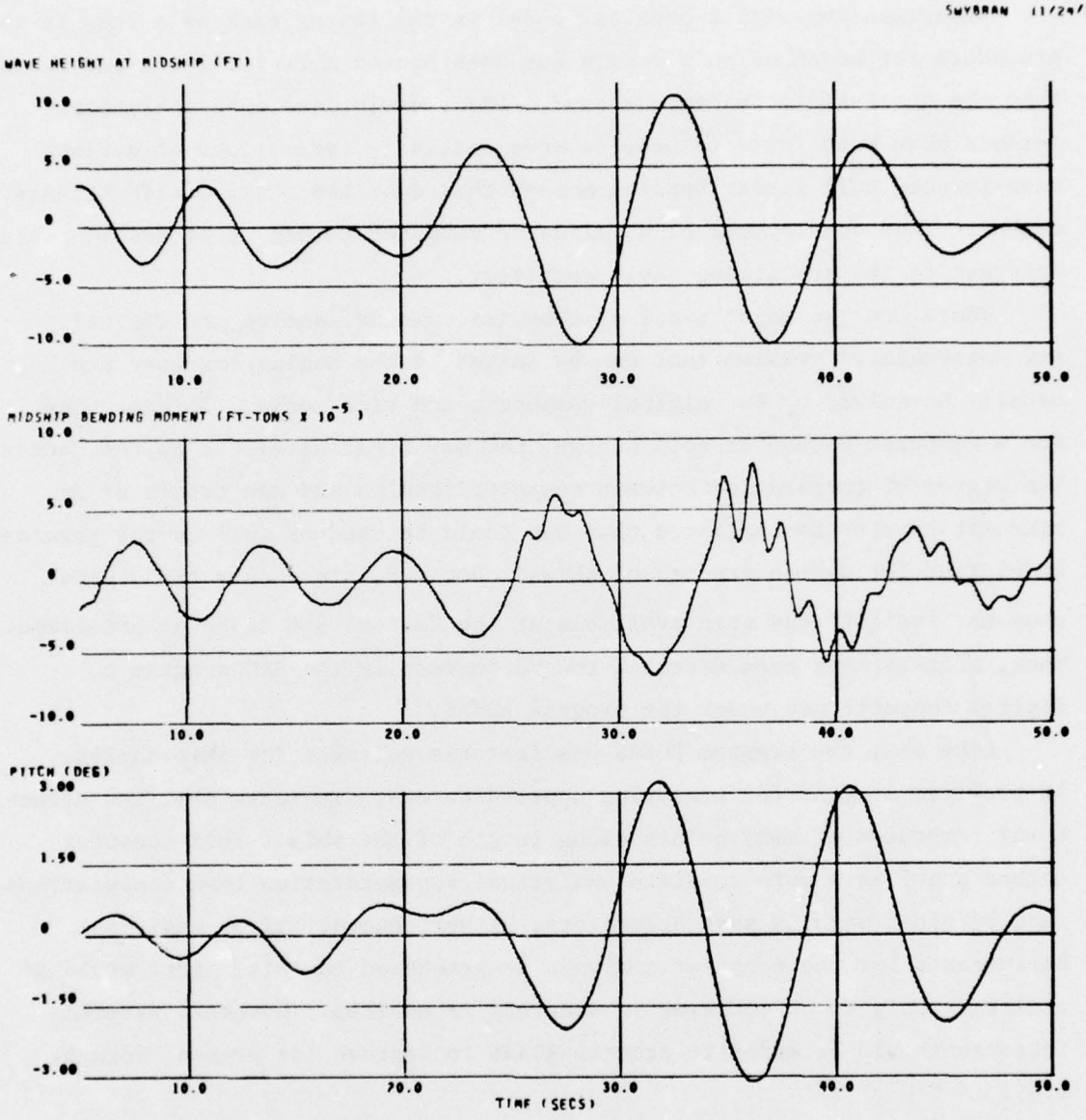


Figure 9 - Response to Bottom Slamming

SUMMARY AND CONCLUSIONS

Experimenting with a physical model in the towing tank as a step in the procedure for advanced ship design has been proved more realistic and reliable than the quasi-static balance method. Also, newly developed analytical methods have been found to provide more realistic estimations of maximum wave-induced hull girder bending moment than does the quasi-static balance method. Thus development of a validated computer method is of considerable interest to the practicing naval architect.

There are two major types of computer, namely, analog and digital. Any mathematical problem that can be solved by the analog computer can usually be solved by the digital computer, and vice versa. In searching for a computer method in ship design, SAC was first developed by the Center. The degree of correlation between computer results and sea trials of an aircraft carrier demonstrated that SAC could be used as well as the physical model test for design studies of ships. However, since only the digital computer facility has been available at the Center, SAC never materialized. Thus, efforts have been directed toward converting the SAC program to digital computer use under the program ROSAS.

Like SAC, the program ROSAS has features suitable for ship design. It provides a means for measuring applied forces, displacements, and structural responses at many points along length of the ship. This computer method provides a more complete analytical representation than computations made by other methods used heretofore, either theoretical or empirical. Refinements for the computer analysis as presented in this report would be necessary only if an increase in accuracy is desired. However, several features should be added to program ROSAS to improve its present format, i.e.,

1. An accurate method for determining both structural and hydrodynamic damping coefficients is needed. Unfortunately, one is still not available.

2. The present program is limited to the head sea condition. The program will be more flexible and like the actual operation of a ship in the seaway, if its capability can be increased from two- to three-dimensional form. This improvement can be accomplished by further development of the program.

3. Capability for investigating bottom slamming has been incorporated in the program. However, head-on wave impact, or any other form of wave impact, has not. This also needs further development.

This study has been confined to checking the feasibility and workability of the program ROSAS after its conversion from SAC. Comparisons made among results obtained from both SAC and ROSAS as well as from the sea trials, physical model, and other methods lead us to conclude confidently that the program ROSAS as it is will be an extremely useful tool for advanced ship research and design. Of course, there is room for improvement, as described previously.

The user's manual is provided in the appendixes. This includes the method for determining ship parameters and hydrodynamic forces and the complete computer program of an illustrated example used in the report.

ACKNOWLEDGMENTS

The authors express their thanks to Dr. J. C. Adamchak, Messrs. J. T. Birmingham, J. N. Andrews, and W. H. Buckley who reviewed the report and provided expert advice to the authors. The authors also gratefully acknowledge the helpful assistance by Mr. R. S. Sigman in preparing a portion of the users manual given in the appendixes. They also wish to take this opportunity to express thanks to Dr. W. W. Murray and Mr. A. B. Stavovy who originally suggested the work.

APPENDIX A - INPUT DATA FOR COMPUTER MODEL

At present, ROSAS is still in a preliminary stage of development. The original concept of this computer model was to have it supply a new ship design, specifying the lines and mass elastic parameters of a ship, which would be simulated on the computer for various environmental conditions. The hydrodynamic and buoyancy forces would then be computed automatically and supplied to various positions along the ship. This can easily be accomplished in time. Presently, the user has to either calculate all these values by longhand or write his own program.

Procedures for calculating parameters of ships are given in detail in Reference 8. The required parameters for computer input are:

1. Ship mass m
2. Bending stiffness EI
3. Mass moment of inertia I_{mz}
4. Shear stiffness KAG
5. Added mass m_v
6. Buoyancy force
7. Smith correction factor ρA_0
8. Structural damping coefficient C
9. Hydrodynamic damping $C(\omega)$. All calculations are based on 21 ship stations, with Stations 0, 1, 2, . . . , 20; Half-Stations 1/2, 1 1/2, . . . , 19 1/2; rudder post Station 0; bow Station 20. Procedures for calculating these parameters are given briefly as follows.

CALCULATION OF SHIP MASS

The total weight of each section — including hull structure, rudder, machinery, ballast, fuel, cargo, etc. — is assumed to be concentrated either at the center of that section or at the half-station point. Either weight distribution curves or a list of weights can be obtained from the shipyard or the ship design office. For a new ship, it may be obtained either by comparing the weight distribution of a sister ship or by estimation. All

---

[8]Bruck, H.A., "Procedure for Calculating Vibration Parameters of Surface Ships," NSRDC Report 2875 (Oct 1968).

PRECEDING PAGE BLANK-NOT FILMED

the weight added together shall be the displacement of a ship under any loading conditions.

The procedure for calculating ship mass distribution is as follows:

1. Obtain a weight distribution curve. This must be the total weight of the ship. If a list of weights is provided, construct a weight distribution curve.

2. Divide the weight distribution into 20 equal sections by drawing a perpendicular line at each station point; then, find the area under the curve for each section. This area is the weight of that section. The weight is assumed to be located at the center of each section, i.e., at the half-station point.

3. Make a table with three columns as indicated in Table 2. In the first column list the half-stations from 1/2 to 19 1/2, in the second list the weight just obtained from the curve, and in the third list the masses m. The mass is calculated by dividing each weight by the value of g, i.e., 32.2 ft/s^2, and then dividing the quotient by the length of each section to obtain mass per unit length of the section.

TABLE 2 - SHIP MASSES

Station	Weight W tons	Mass m $ton\text{-}s^2/ft^2$
1/2		
1 1/2		
.		
.		
.		
19 1/2		

Note: $m = W/g/\Delta X$

CALCULATION OF BENDING STIFFNESS

The value of Young's modulus E can easily be found from the materials handbook. The vertical area moment of inertia I for a ship section is calculated by summing the area moment of inertia of the deck, shell, double bottom, and continuous longitudinal members that comprise that section; it is determined about the horizontal neutral axis, which is parallel to the ship baseline through the centroid of that section, with

$$I = \Sigma\ (I_0 + ad^2) \tag{22}$$

where I_0 is the area moment of inertia of a structural member about its own neutral axis

a is the area of a structural member

d is the distance from the neutral axis of the structural member to the reference axis.

The procedure for calculating the vertical area moment of inertia I is as follows:

1. The decks and shell of a ship are usually constructed by using several plates of different thickness welded or riveted together. Find the areas of deck and shell plates, longitudinals, etc. List these areas in a table such as Table 3.

2. Use the baseline of the ship as the reference axis.

3. Measure and list the distance d of each item from its centroid to the baseline.

4. Multiply a by d and by d^2; calculate I_0, and list.

5. Calculate $d_0 = \Sigma\ ad/\Sigma\ a$, and $I = 2\ (\Sigma\ I_0 + \Sigma\ ad^2 - d_0^2\ \Sigma\ a)$; where d_0 is the distance of neutral axis of ship section from the baseline, and I is the total vertical area moment of inertia of ship section.

6. Repeat this procedure for all sections.

7. Plot all values of I against ship length and fair-in the curve. Locate the half-station points, read-off the I values at half-stations, and list.

8. The EI values are simply I multiplied by the constant E, which is 1.93×10^6 ton/ft^2 for steel.

TABLE 3 - DETERMINATION OF SECTIONAL VERTICAL AREA MOMENT OF INERTIA

Section No. _____

Item	a	d	d^2	ad	ad^2	I_0
	ft^2	ft	ft^2	ft^3	ft^4	ft^4
Deck Plate						
-						
-						
-						
Deck Long'l						
-						
-						
-						
Shell Plate						
-						
-						
-						
Shell Long'l						
-						
-						
-						
Double Bottom						
-						
-						
-						
Long'l						
-						
-						
-						
Etc.						
	Σ a			Σ ad	Σad^2	ΣI_0

$d_0 = \Sigma ad / \Sigma a$

$I = 2 (\Sigma I_0 + \Sigma ad^2 - d_0^2 \Sigma a)$

Not counted in the calculations are superstructures, discontinuous longitudinals, hatchways, deck plating between two hatches, and transverse members. However, welded intercostal members with no lightening holes should be included.

I_0 is usually omitted in calculating horizontal plates because I_0 values are very small, compared to ad^2 values.

CALCULATION OF MASS MOMENT OF INERTIA

The mass moment of inertia I_{mz} of a ΔX section of a ship about a rotating axis through its center of gravity parallel to z-axis (horizontal athwartship direction) consists of contributions from the ship mass and the added mass of fluid, i.e.,

$$\text{Total } I_{mz} = I_{mz}]_{\text{hull}} + I_{mz}]_{\text{superstructure}} + I_{mz}]_{\text{added mass}}$$

The added mass moment of inertia may be neglected.[8] For the hull, the actual mass in a length ΔX is assumed to be uniformly distributed and is bounded by the main deck and the shell. The total I_{mz} of a station is given by the equation

$$I_{mz} = m \cdot r^2 \tag{23}$$

where m is the mass per unit length of a ship section, and r is the radius of gyration given by the equation

$$r^2 = (d^2 + (\Delta X)^2)/12 \tag{24}$$

with d the depth of ship to main deck. If the ship deviates from standard, i.e., having its flight deck above the main deck or a long superstructure, detailed calculation to obtain I_{mz} is recommended.

CALCULATION OF SHIP SHEAR STIFFNESS

The ship hull is similar to a box girder with its uppermost continuous deck and bottom as flanges and its side shell as webs. In vertical vibration, the shear V is essentially carried by the side shell and any continuous longitudinal bulkheads. The shear stress is approximately uniform over the area of vertical plating A_v. This gives

$$KAG \cong A_v G \tag{25}$$

where K is the constant, depending on the shape of ship hull cross section

A is the cross sectional area of ship hull

G is the shear modulus of elasticity ($G = 7.72 \times 10^5$ ton/ft^2 for steel)

A_v is the cross sectional area of ship hull for vertical plating only.

CALCULATION OF ADDED MASS

A body moving with unsteady motion in an ideal fluid is subject to hydrodynamic pressure forces which are proportional to instantaneous acceleration. The resultant force acting on the body is directed opposite to that of the acceleration in the same manner as if an additional mass were attached to the system. It is therefore called added mass — sometimes virtual or hydrodynamic mass.

For an arbitrary cross sectional ship form, it is customary to calculate the added mass per foot of length by using the added mass values per foot of length of an infinite length of plate with width b. This value is amended by correction factors C_V and J, developed by various authors to allow for the finite length of an actual ship and the departure of its cross sectional shape from a rectangle. The formulas used for calculating added mass per unit length are as follows. For ship emergence (Figure 3)

$$m_V = m_0 + m_1 = \frac{2}{\pi} J C_V \rho b^2 \tag{26}$$

For ship immergence

$$m_V = m_0 + m_2 = \frac{\pi}{2} J C_V \rho b^2 \tag{27}$$

Here J is the longitudinal coefficient, depending on the ship length-to-beam ratio L/2b, and is given in Figure 10. The added mass coefficient is C_V, depending on the ship sectional area coefficient β and the beam-to-draft ratio 2b/d, and is given in Figure 11.[4] Equations (26) and (27) are the added mass for both port and starboard sides and do not need to be multiplied by two again.

As indicated in Figure 3, the water surfaces at the ship boundary are different during ship emergence and immergence. This causes the difference in added masses, mainly because the half-width used in the previous equations is b for the emergency and $\frac{\pi}{2}$ b for the immergence, due to rising of the water surface at the boundary.

Calculation of added mass used for program ROSAS is divided in two parts. For sections with no nonlinear terms considered, assume that they are wall sided and that only the added mass at the still waterline is calculated by using Equation (27). For sections with nonlinear terms considered, the added mass at each 1 ft of draft interval is calculated by Equation (26) for emergence and Equation (27) for immergence. The procedure used for calculating the added mass with nonlinear terms is listed as follows:

1. Obtain a body plan of the ship which shows the cross sectional profiles for 21 stations.

2. Obtain a Bonjean curve of areas for 21 stations. (If not available, make one.)

3. At each station, obtain half-breadth b and cross sectional area A for every draft d at 1-ft intervals; where b is from the body plan, and A is from the Bonjean curve.

4. Calculate $\beta = A/(2bd)$, L/2b, and 2b/d; obtain C_V and J from Figures 10 and 11.

5. Use Equations (26) and (27) to calculate the added mass $m_0 + m_1$ and $m_0 + m_2$ for ship emergence and immergence.

6. Calculate the added mass m_0 associated with a still waterline by Equation (27) as if the added mass has no nonlinear term.

7. Obtain the nonlinear added mass terms m_1 and m_2.

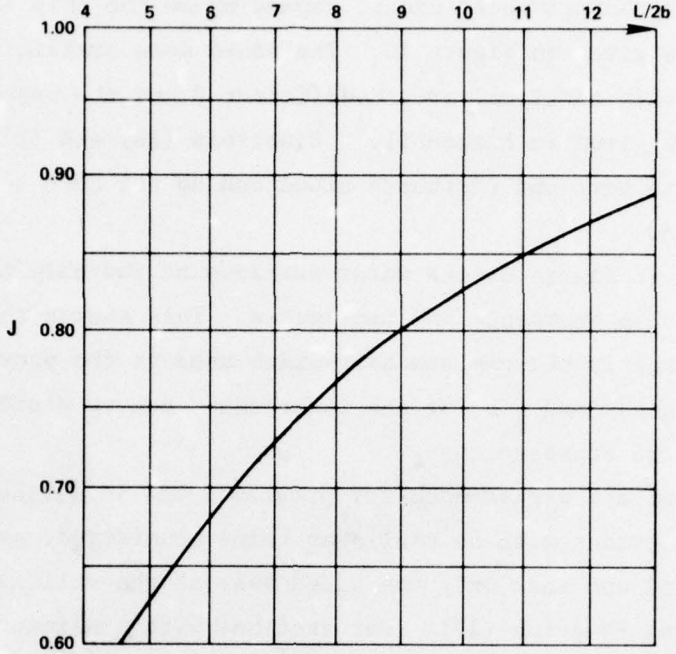


Figure 10 - Curve for Estimating Coefficient J Used in Added Mass Evaluation

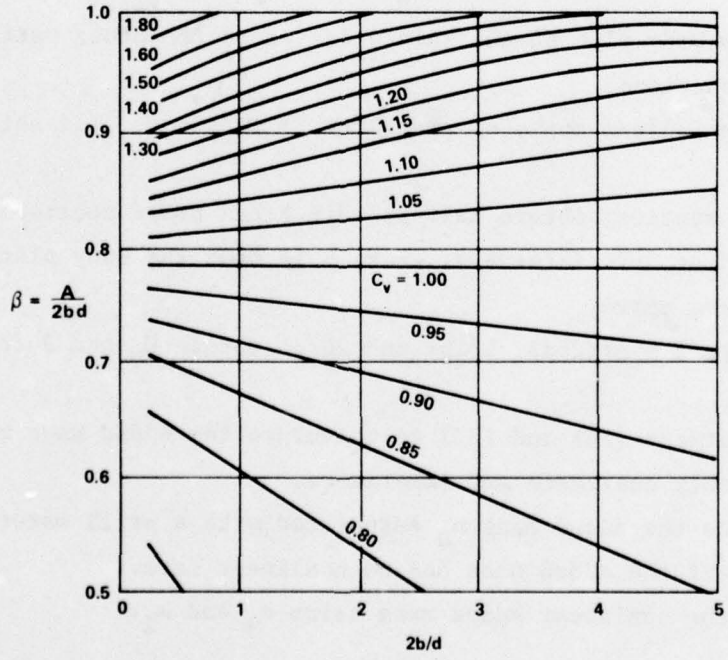


Figure 11 - Curves for Estimating Coefficient C_v Used in Added Mass Evaluation

8. Approximate m_1 and m_2 into a power series by

$$m_1 = a_1 Y_r + a_2 Y_r^2 + a_3 Y_r^3 \tag{28}$$

$$m_2 = c_1 Y_r + c_2 Y_r^2 + c_3 Y_r^3 \tag{29}$$

where a_1, a_2, a_3, c_1, c_2, and c_3 are arbitrary constants to be determined, e.g., as given in Reference 9.

9. The total added mass term for each section is given by Equation (16)

$$(m_V)_n = (m_0)_n + (a_1 Y_r)_n + (a_2 Y_r^2)_n + (a_3 Y_r^3)_n \tag{30}$$

for emersion, and

$$(m_V)_n = (m_0)_n + (c_1 Y_r)_n + (c_2 Y_r^2)_n + (c_3 Y_r^3)_n \tag{31}$$

for immersion.

CALCULATION OF BUOYANCY FORCE

The expression for the dynamic buoyancy force is given by Equation (20), in which

$$K_b = \rho g b_1 \Delta X/\Delta X \tag{32}$$

is the buoyancy spring, and b_1 is illustrated in Figure 2. The value of b_1 can be obtained either from the ship body plan or from the offset.

In calculating buoyancy force, the cross sectional area to the actual waterline is

$$A = A_0 - b_1 Y_r + \bar{A} \tag{33}$$

---

[9] Carnahan, B. et al., "Applied Numerical Methods," Chapter 1, John Wiley & Son Inc., New York (1969).

The buoyancy force per unit length of ship is then

$$\rho g A = \rho g \ (A_0 - b_1 \ Y_r + \bar{A}) \tag{34}$$

The nonlinear buoyancy force per unit length of ship becomes

$$\rho g \bar{A} = \rho g \ (A - A_0) + K_b \ Y_r \tag{35}$$

where A and A_0 are obtained from the Bonjean curve; thus, algebraic expressions can be determined in the form of

$$\rho g \bar{A} = b_2 \ Y_r^2 \ \rho g \tag{36}$$

for the nonlinear buoyancy force in Equation (20).

SMITH CORRECTION FACTOR

The Smith correction factor (SCF) ρA_0 is part of the dynamic buoyancy force given in Equation (20). The value of A_0 can be obtained as before by the Bonjean curve.

STRUCTURAL AND HYDRODYNAMIC DAMPING COEFFICIENTS

Evaluation procedures for determining damping of a ship seem uncertain at the present time. When a ship is oscillating on water, four types of damping factors are generated, namely,

1. Water Friction
2. Generation of a Pressure Wave
3. Generation of a Surface Wave
4. Structural Damping Force. The first three types generate the hydrodynamic damping force. Generation of surface waves and the structural damping force are the main sources of damping. McGoldrick (Reference 4, Table 8-2) mentioned from the analysis of many data about full-scale experimental work that damping in ship vibration appears to increase with frequency and the value of $(C + C(\omega))/m_s\omega$; where $(C + C(\omega))$ is the ship damping coefficient for the sum of hydrodynamic and structural damping, m_s is the ship

mass per unit length, and ω is the frequency in radians. He gave 0.034 as the mean value of $(C + C(\omega))/m_s\omega$ for all modes of ship vibration.

The rigid body motion of a ship mainly contains hydrodynamic damping in the low-frequency region. Determination of the hydrodynamic damping coefficient $C(\omega)$, which depends on the shape of the cross section of the ship, will be given later in the working example.

INPUT DATA FOR ESSEX

ESSEX has been used to test ROSAS because results obtained from ROSAS can be readily compared with those obtained from SAC,[1] the model test,[7] the sea trials,[2] and program SCORES.[7] The cellular divisions used in analyzing ESSEX are shown in Figure 4. Masses of the ship m are lumped at 20 equally spaced half-station points at intervals of 41 ft, and the bending rigidity EI is evaluated at the same points. These data are recorded in Table 4. The mass moment of inertia I_{mz} and the shear rigidity KAG are evaluated at 21 equally spaced stations and are recorded in Table 5.

Figure 4 also shows how the hydrodynamic forces are applied to the lumped model of the ship. The hydrodynamic forces are evaluated at nine stations — 3, 5, 7, 9, 11, 13, 15, 17, and 19. At four stations — 3, 15, 17, and 19 — nonlinear buoyancy and added mass effects are included. The hydrodynamic forces are equally distributed to either two or three half-station points in order to obtain a more uniform force distribution.

The linear hydrodynamic properties used in the analysis are listed in Table 6. Values apply to a ship in calm water at a draft of 28.5 ft.

The linear added mass m_0' for each station is equal to $\frac{\pi}{2} J C_V \rho b^2$ by Equation (27) and is listed in Table 7. The values for added mass are lumped, extending over the ship length from Station $(n - \frac{1}{2})$ to Station $(n + \frac{1}{2})$. Values for added mass listed in Table 6 come from Table 7. The value for m_{0_s}, for example, is

TABLE 4 - STRUCTURAL MASS AND BENDING RIGIDITY

Station	m	$EI \cdot 10^{-9}$
	$ton\text{-}sec^2/ft^2$	$ton\text{-}ft^2$
0.5	0.45467	5.05075
1.5	0.74401	7.76725
2.5	0.84735	12.31675
3.5	1.13281	18.30495
4.5	1.70245	27.04600
5.5	2.10287	36.95395
6.5	2.20362	44.24075
7.5	2.14550	49.67695
8.5	2.09512	55.16235
9.5	2.10158	58.75385
10.5	2.04374	59.00065
11.5	1.99308	50.67175
12.5	1.94270	37.47675
13.5	1.85357	33.20715
14.5	1.76574	34.4828
15.5	1.65465	28.64295
16.5	1.35756	19.27405
17.5	0.87060	13.65365
18.5	0.54122	9.82855
19.5	0.36426	7.4482

TABLE 5 - MASS MOMENT OF INERTIA AND SHEAR RIGIDITY

Station	$I_{mz} \cdot 10^{-3}$	$KAG \cdot 10^{-5}$
	ton-sec^2	ton
0	0.02686	13.3987
1	0.25408	16.3673
2	0.41905	24.8485
3	0.61563	35.3144
4	0.83089	47.7022
5	1.01557	60.5613
6	1.15716	71.4286
7	1.26788	77.6515
8	1.33742	78.3190
9	1.34124	75.0916
10	1.29996	71.8039
11	1.23465	69.6686
12	1.16907	70.2055
13	1.06159	75.2984
14	0.91029	81.5109
15	0.73649	73.2143
16	0.54697	40.0000
17	0.35450	24.5509
18	0.20379	18.5520
19	0.13025	14.3106
20	0.08431	12.6935

TABLE 6 - LINEAR ADDED MASS, BUOYANCY, AND SMITH CORRECTION FACTORS

Station	m_0	K_b	ρA_0
	ton-sec^2/ft^2	ton/ft^2	ton-sec^2/ft^2
3	1.2780	2.6057	0.9077
5	2.5988	3.1841	2.0060
7	3.5183	3.2951	2.6419
9	3.9610	3.3012	2.8519
11	3.7488	3.2732	2.7917
13	2.8329	3.0329	2.3768
15	1.5122	2.3220	1.5920
17	0.4963	1.2354	0.8216
19	0.1000	0.3402	0.3184

TABLE 7 - ADDED MASS AS A FUNCTION OF DESIGN WATERLINE OF 28.5 FEET

Station n	Added Mass m'_{0_n}	Station n	Added Mass m'_{0_n}	Station n	Added Mass m'_{0_n}
	ton-s^2/ft		ton-s^2/ft		ton-s^2/ft
0	0.23	7	146.06	14	89.87
1	18.12	8	157.78	15	60.58
2	37.34	9	164.27	16	36.97
3	59.43	10	163.27	17	18.19
4	84.20	11	155.58	18	7.96
5	107.44	12	140.39	19	3.48
6	127.03	13	117.14	20	0.71

Values for added mass are lumped extending over ship length from Station $(n - \frac{1}{2})$ to Station $(n + \frac{1}{2})$.

$$m_{0_5} = \left(\frac{1}{2} m'_{0_4} + m'_{0_5} + \frac{1}{2} m'_{0_6}\right)$$

$$= (42.1 + 107.4 + 63.5)$$

$$= 213.1 \text{ ton-s}^2/\text{ft of 82 ft ship length}$$

$$= 2.5988 \text{ ton-s}^2/\text{ft}^2 \text{ of 1 ft ship length}$$

The linear buoyancy spring K_b is equal to $\rho g b_1$ given in Equation (20) and repeated in Equation (32). The values for the buoyancy spring listed in Table 4 are obtained in the same manner as for the added mass illustrated previously.

The Smith correction factor is equal to $\rho A_0 \Delta X/m_0$. The values of ρA_0 are listed in Table 4 and are obtained in the same manner as for the added mass. These three linear hydrodynamic properties of added mass, buoyancy spring, and Smith correction factor are obtained from Table 4 of Reference 1 which explains how these values were obtained.

The two nonlinear hydrodynamic forces, buoyancy force and added masses, are included for Station 3 at the stern and Stations 15, 17, 19 at the bow at the present time. This can be modified if so desired.

The nonlinear buoyancy forces used in the analysis are shown in Figure 12 as functions of emersion Y_r. These curves were derived from curves of the total buoyancy force shown in Figure 13 for a 1-ft section at Stations 3, 15, 17, and 19 by subtracting the linear spring rates recorded in Table 6. Algebraic expressions for these curves are given in Table 8. These were obtained by fitting the best mathematical curve to the data points for the nonlinear buoyancy force.

The nonlinear added masses used in the analysis are shown in Figure 14, and they were obtained in the same manner as for the nonlinear buoyancy force. Algebraic expressions for these curves are given in Table 9.

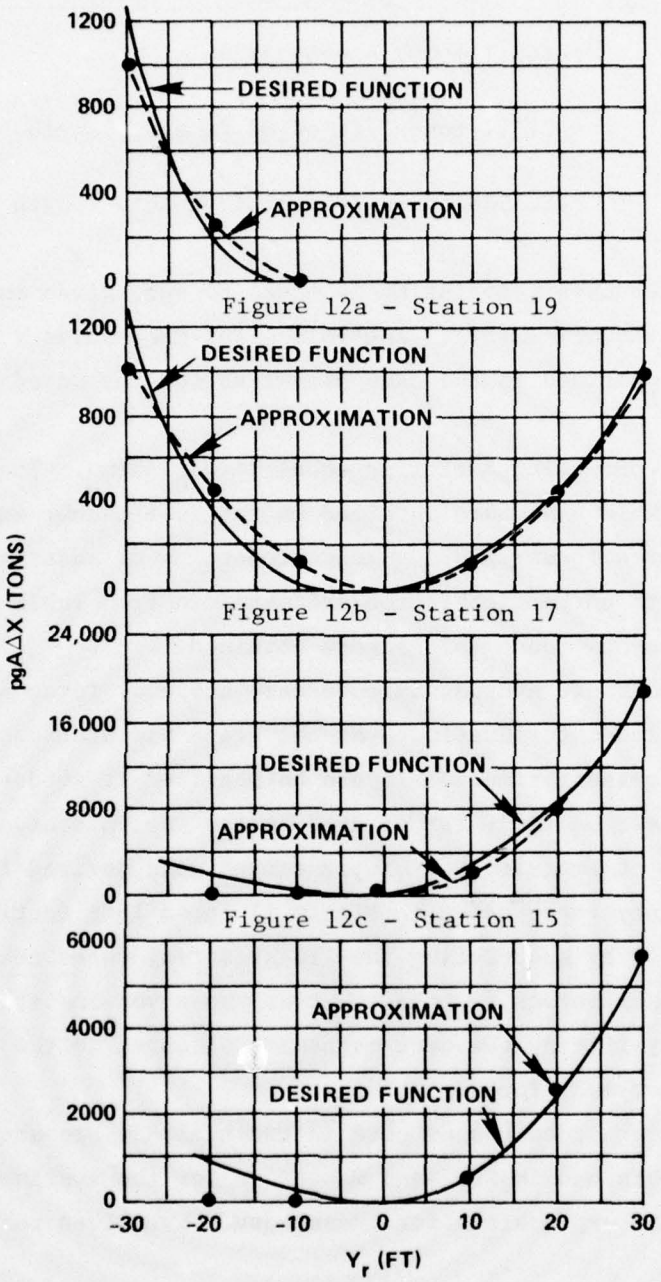


Figure 12a - Station 19

Figure 12b - Station 17

Figure 12c - Station 15

Figure 12d - Station 3

Figure 12 - Nonlinear Buoyancy Forces

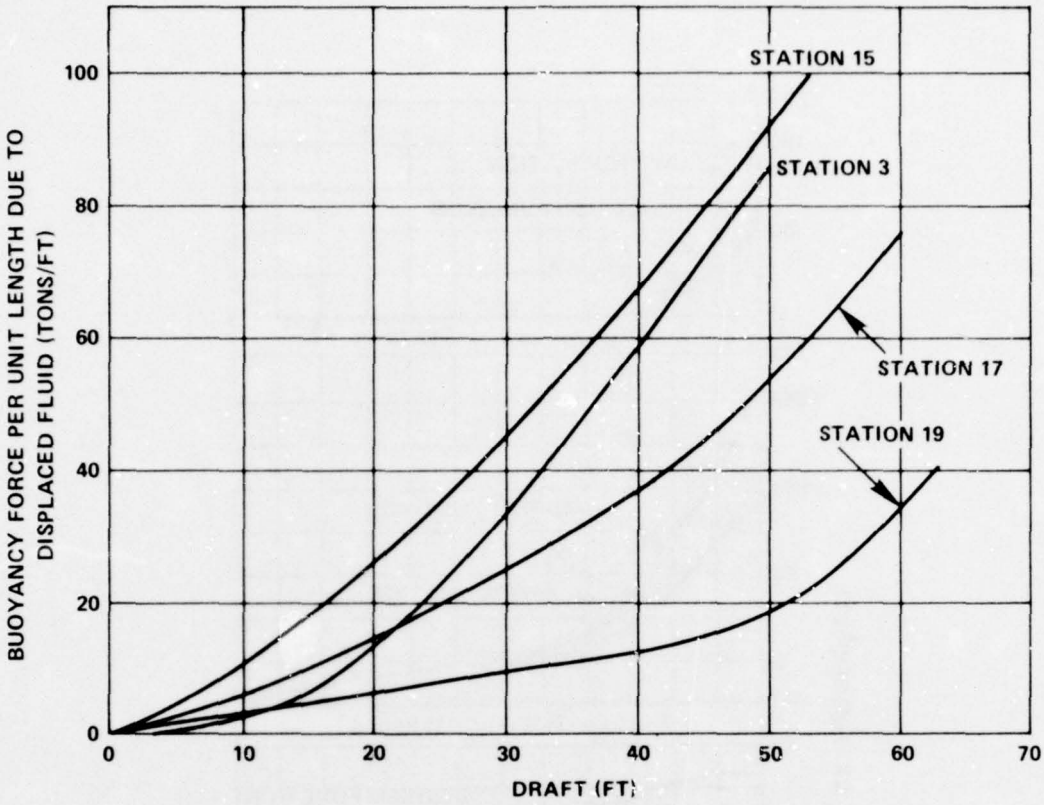


Figure 13 - Variation of Buoyancy Force with Draft

TABLE 8 - NONLINEAR BUOYANCY TERMS

Station	Buoyancy tons/ft	
19	$\rho g\bar{A} = 0.03024\ (Y_r + 10)^2$	$Y_r < -10$
	$= 0$	≥ -10
17	$\rho g\bar{A} = 0.01345\ Y_r^2$	
15	$\rho g\bar{A} = 0.02512\ Y_r^2$	$Y_r > 0$
	$= 0$	$Y_r \leq 0$
3	$\rho g\bar{A} = 0.05179\ Y_r^2$	$Y_r > 0$
	$= 0$	$Y_r \leq 0$

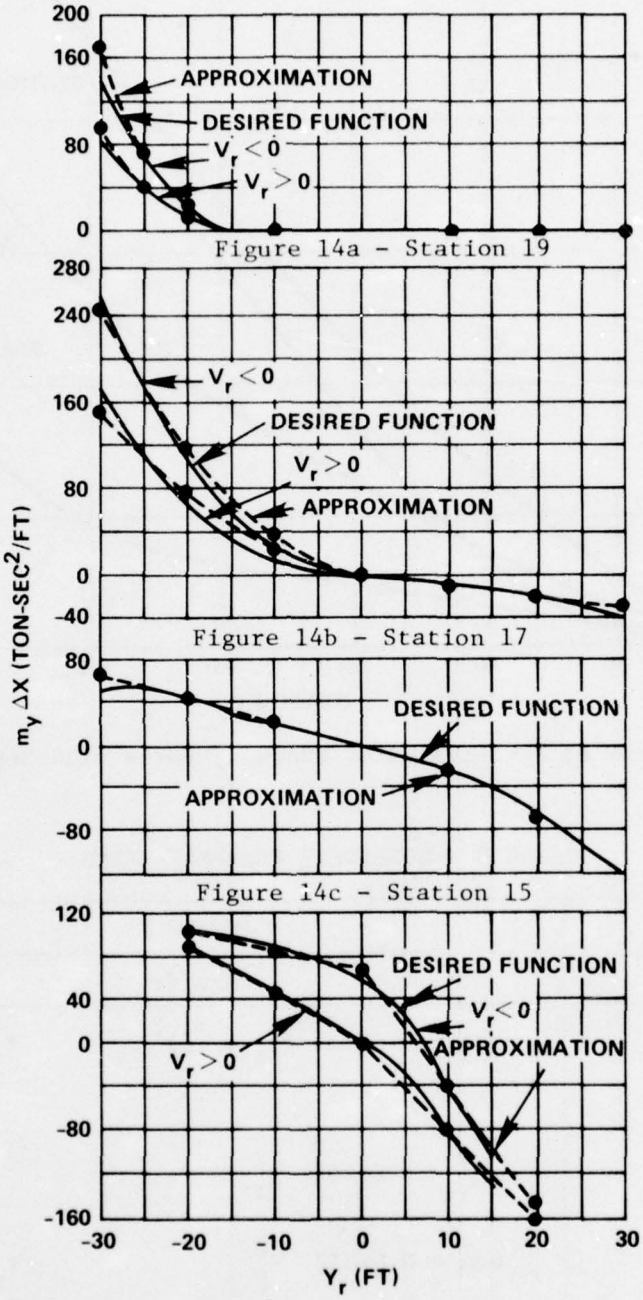


Figure 14a - Station 19

Figure 14b - Station 17

Figure 14c - Station 15

Figure 14d - Station 3

Figure 14 - Nonlinear Added Masses

TABLE 9 - NONLINEAR ADDED MASS TERMS

Station	Added Mass·10^3 ton-s^2/ft^2	
19	$\bar{m} = -0.1476\ (Y_r + 10)^3$	$Y_r < -10,\ V_r > 0$
	$= 0$	$> -10,\ > 0$
	$= -0.2658\ (Y_r + 10)^3$	$< -10,\ < 0$
	$= 0$	$> -10,\ < 0$
17	$\bar{m} = -13.17\ Y_r + 1.568\ Y_r^2$	$Y_r < 0,\ V_r > 0$
	$= -13.17\ Y_r$	$> 0,\ > 0$
	$= -13.17\ Y_r + 2.720\ Y_r^2$	$< 0,\ < 0$
	$= -13.17\ Y_r$	$> 0,\ < 0$
15	$\bar{m} = -26.13\ Y_r - 0.7805\ Y_r^2$	$Y_r > 0$
	$= -26.13\ Y_r$	< 0
3	$\bar{m} = -37.32\ Y_r$	$Y_r < 0,\ V_r > 0$
	$= -68.29\ Y_r$	$> 0,\ > 0$
	$= 560.2 - 14.74\ Y_r$	$< 0,\ < 0$
	$= 560.2 - 91.54\ Y_r$	$> 0,\ < 0$

Figure 15 shows the hydrodynamic damping coefficient for a 1-ft section. These curves were derived from References 10 and 11, and were used for the SAC given in Reference 1. Since damping force does not affect the solution given by the computer very much, the values of $C(\omega)$ used are considered sufficiently accurate for both SAC and program ROSAS.

Figure 15 can also be used to estimate $C(\omega)$ for other ships with non-dimensional factors $\omega\sqrt{B/g}$ and $C(\omega)/(\Delta/\sqrt{gL^3})$. For ship vibration, the mean value of 0.034 for $(C + C(\omega))/m_s\omega$ may be used for all modes.

The velocity of wave propagation of the sea surface has been established

$$c(\text{fps}) = g/\omega \tag{37}$$

The wave velocity relative to the ship is equal to c+U.

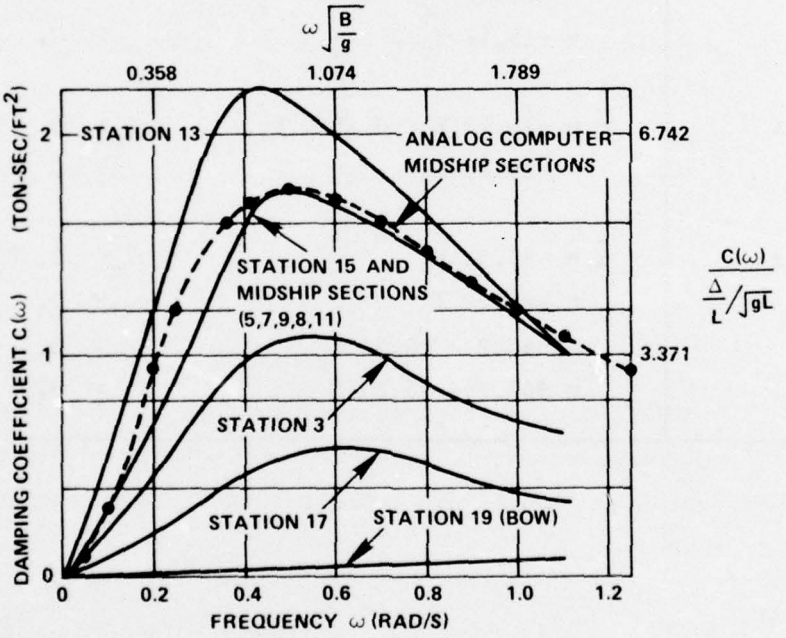


Figure 15 - Hydrodynamic Damping Coefficients

---

[10] Grim, O., "Berechnung der durch Schwingungen eines Schiffskörpers erzeugten hydrodynamischen Kräfte," Jahrbuch der Schiffbautechnischen Gesellschaft, Vol. 47 (1953).

[11] Golovato, P., "A Study of the Forces and Moments on a Heaving Surface Ship," David Taylor Model Basin Report 1074 (Sep 1957).

APPENDIX B - STRUCTURAL SEAWORTHINESS DIGITAL COMPUTER PROGRAM ROSAS

The schematic of ROSAS has been shown in Figure 1. The program consists of the main program called PROGRAM SIMSHIP and many subroutines as shown in Figure 16. Descriptions of subroutines and functions are given in Table 10. The functioning of subroutines and alternate entries is given in Table 11, which shows names of subroutines to call and to be called. Table 12 gives the complete program ROSAS which includes PROGRAM SIMSHIP, the major subroutines SUBROUTINE HYD FRC, SUBROUTINE SLAM, SUBROUTINE SEA GEN, SUBROUTINE DAUX, SUBROUTINE KUTMER, and other subroutines. Also included in Table 12 are a list of input cards and a sample of selected output print. The flow chart for the program ROSAS is shown in Figure 17.

SIMSHIP

The main program is SIMSHIP. It determines the length of the time step to the next printing or plotting time. It calls the integration subroutine to integrate to this time and then prints data or calls subroutines that store data for plotting. At the end of the problem it calls subroutines to plot the data that have been previously stored. The flow chart for SIMSHIP is given by Figure 18.

KUTMER

SUBROUTINE KUTMER is used to integrate the system of ordinary differential equations that describe the response of the ship to hydrodynamic forces. This subroutine implements a Runge-Kutta method that incorporates automatic error control.

DAUX

SUBROUTINE DAUX is called by the integration routine to compute linear derivatives with respect to time of the ship vertical position, vertical velocity, angular velocity, bending moment, and shear force. These derivatives are computed at the various stations of the ship by evaluating the

expressions on the right-hand sides of Equations (15a) to (15d). The subroutine HYD FRC is called by DAUX to compute the hydrodynamic force P acting on the ship.

HYD FRC

SUBROUTINE HYD FRC computes the hydrodynamic force P, using Equations (1) to (4). The hydrodynamic forces are computed at odd numbered Stations 3 through 19. Then they are distributed to Half-Stations 1.5 through 19.5. The sum of the linear part of the added mass and the ship mass is returned to DAUX, where it is used in Equation (15a).

SUBROUTINES SEA GEN and SLAM are called by HYD FRC to compute a kinematic description of the surface of the sea and forces due to bottom slamming of the ship. Principal FORTRAN variables for subroutine HYD FRC are listed in Table 13, and the flow chart for subroutine SLAM is shown in Figure 19.

SEA GEN

SUBROUTINE SEA GEN computes the vertical height, velocity, and acceleration of the sea. These values are computed and returned for each of the nine stations at which the hydrodynamic forces are computed. Also the velocity of the waves is computed. For sinusoidal waves, vertical height, velocity, and acceleration of the waves is computed by

$$Y_w = H \sin \omega(t+X/c)$$

$$\dot{Y}_w = \omega H \cos \omega(t+X/c)$$

$$\ddot{Y}_w = -\omega^2 H \cos \omega(t+X/c)$$

where H is the prescribed wave height in feet

ω is the prescribed wave frequency in radius/second

X is the position of the station for which the values are being computed. The wave velocity (celerity) c equals g/ω, where g is the acceleration of gravity in feet/second^2. The shape of the discrete wave train is approximated by the sum of two sinusoidal waves.

INPUT CARDS

The input deck consists of a first-case input deck and either none or more subsequent-case input decks. Case decks are separated by 7/8/9 cards, and the last-case input deck is followed by two consecutive 7/8/9 cards.

The possible card types in the first-case input deck are listed in Table 14. The consequence of omitting an input card is explained in the last column of that table. Subsequent-case input decks consist of only card types on which one or more fields differ from the preceding case.

Figure 16 - Structural Seaworthiness Digital Computer Program ROSAS

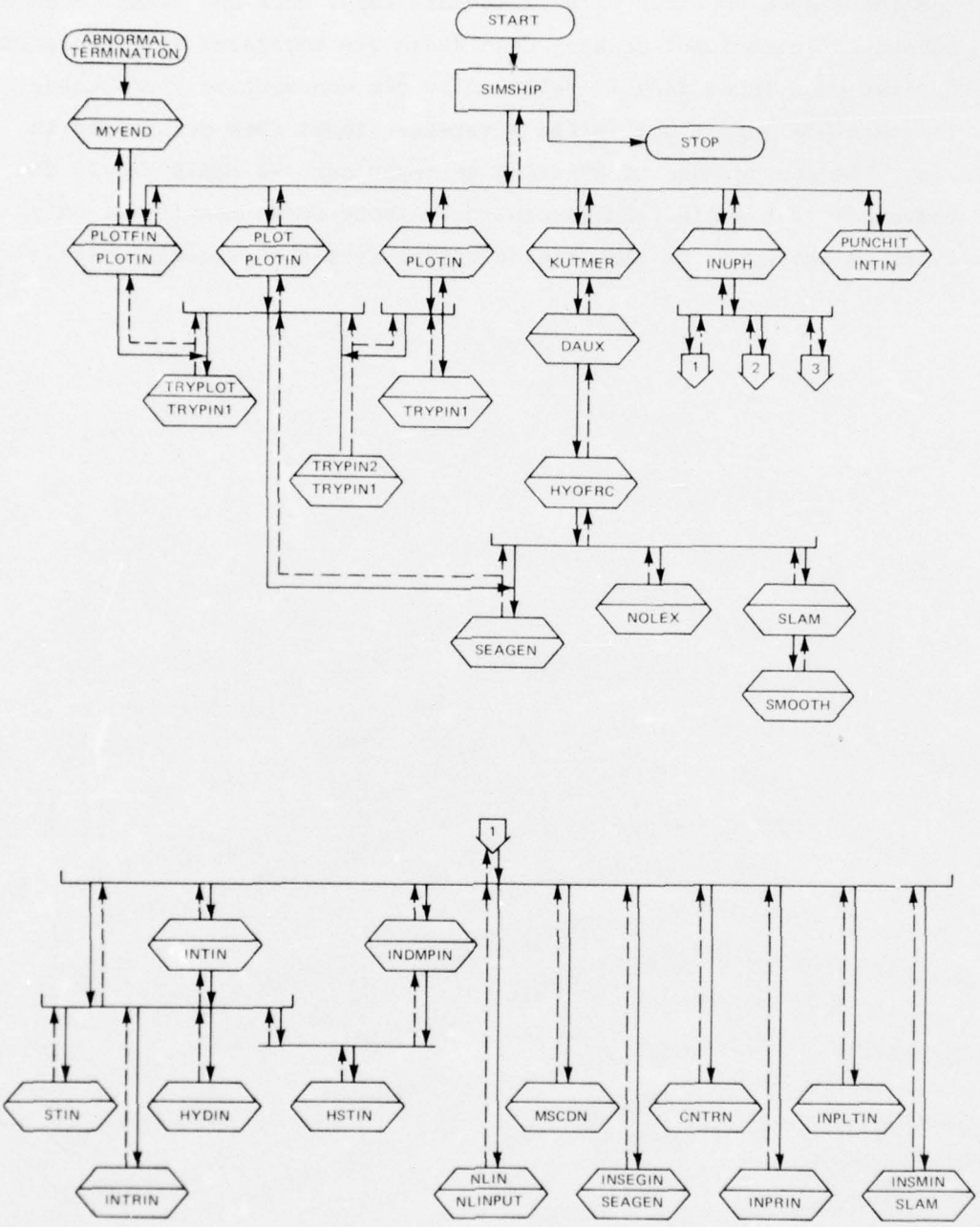
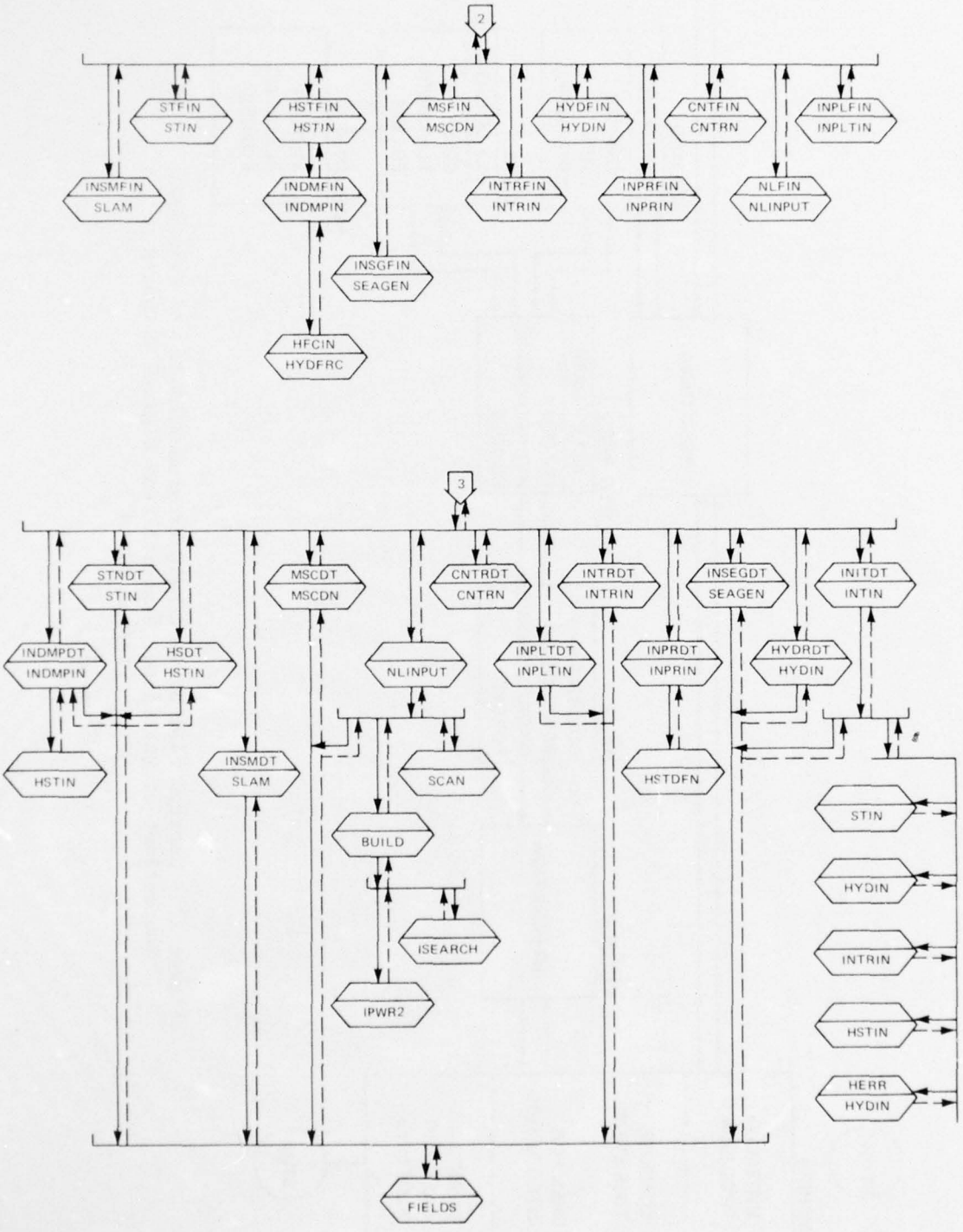


Figure 16 (Continued)



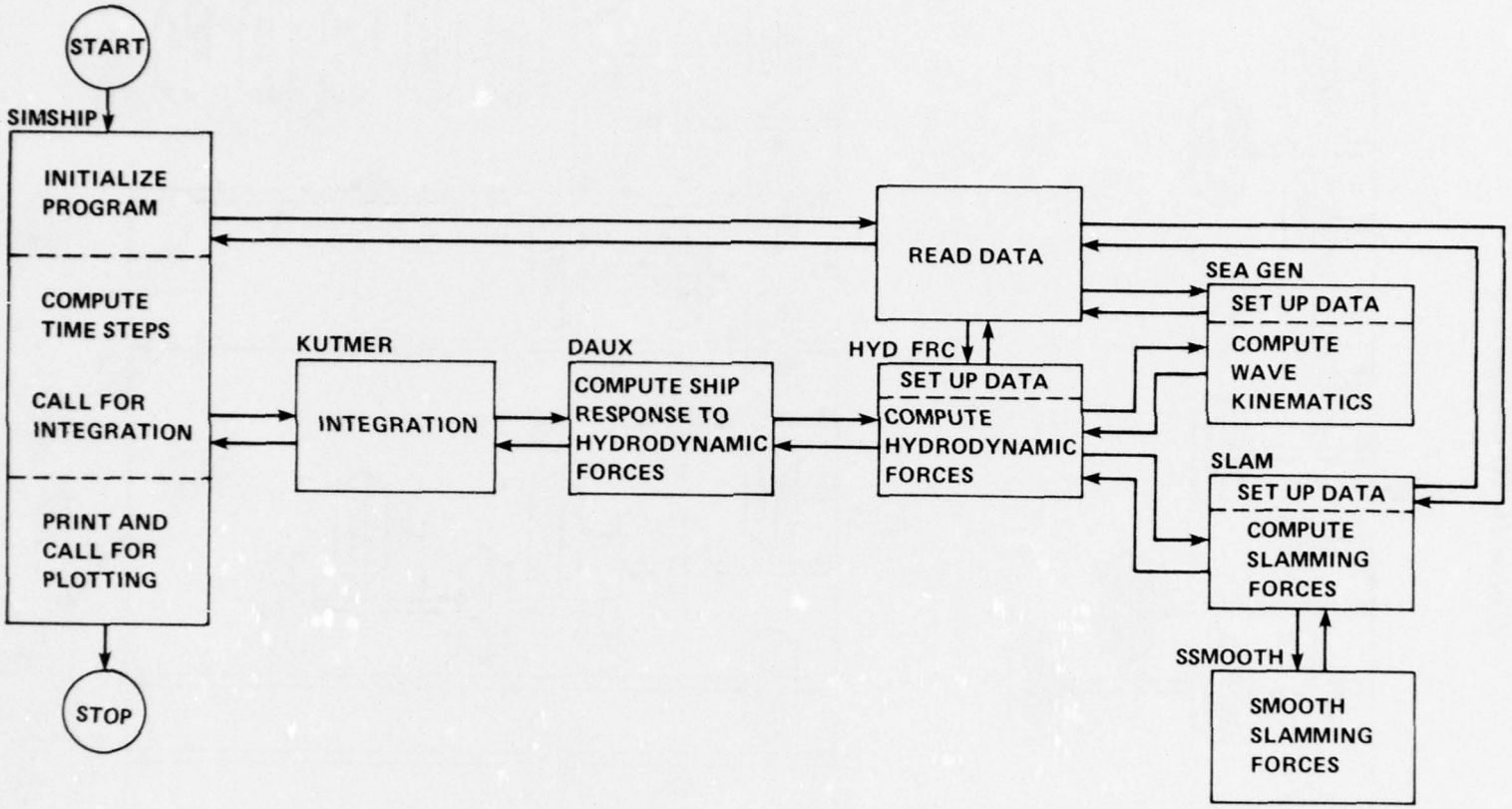


Figure 17 - Control Flow between Main Program SIMSHIP and Principal Subroutines of Structural Seaworthiness Digital Computer Program ROSAS

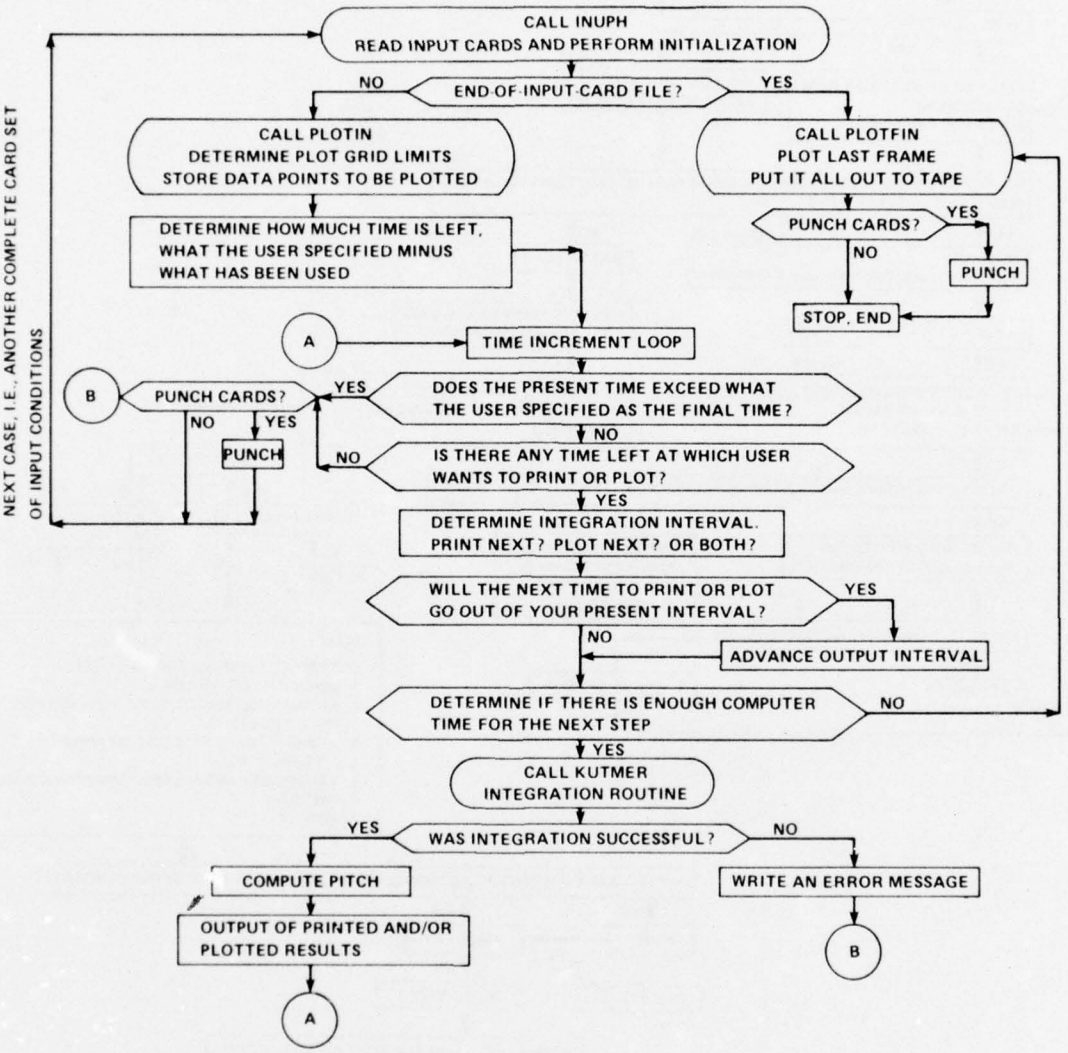


Figure 18 - Program SIMSHIP Flow Chart

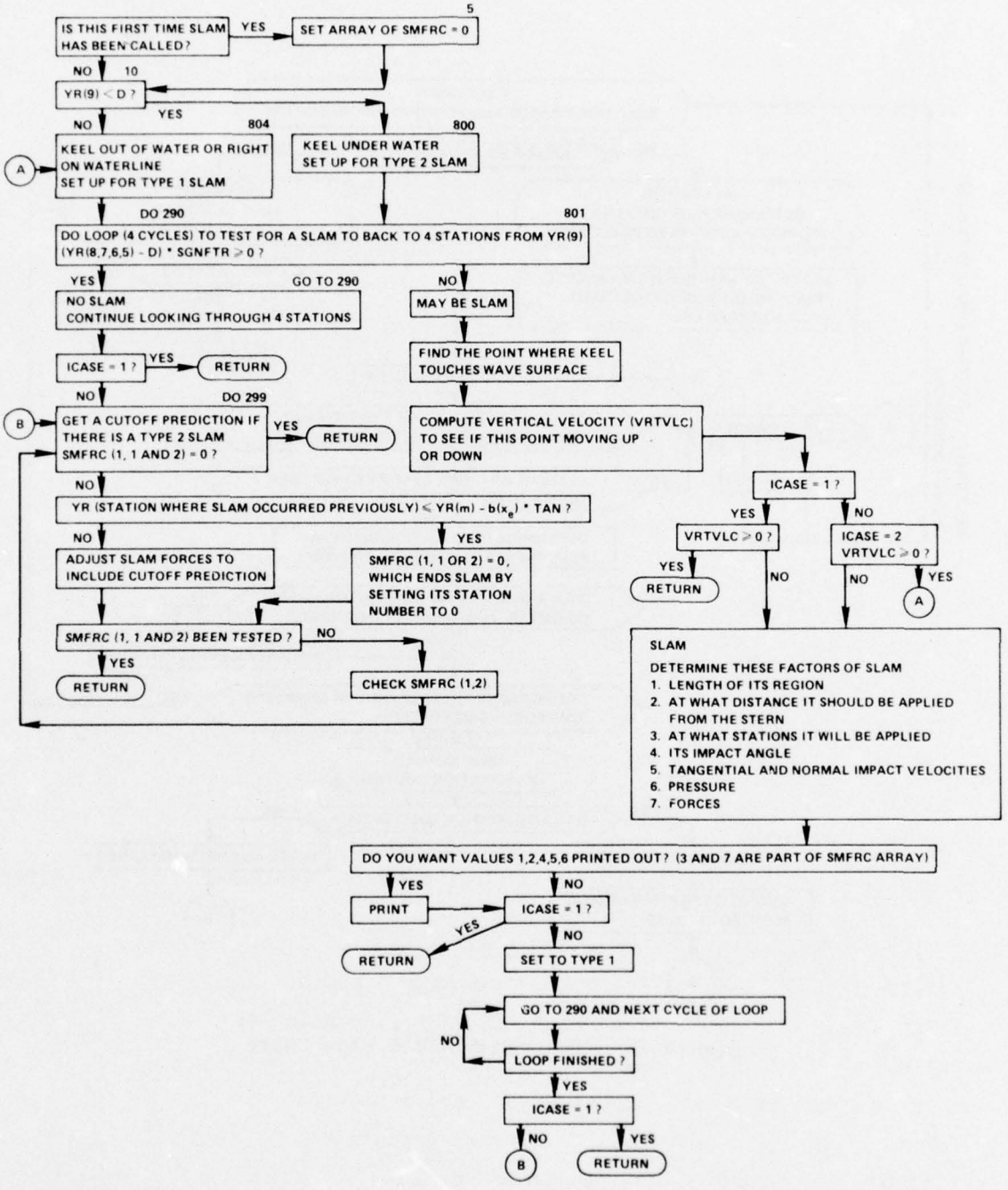


Figure 19 - Subroutine SLAM

TABLE 10 - GUIDE TO ROSAS SUBPROGRAMS

Name:	*Type**	*Description*
BUILD	S	Builds sorted list
CNTFIN	E(CNTRN)	Echoes CONTROL card data
CNTRDT	E(CNTRN)	Reads CONTROL cards
CNTRN	S	Sets CONTROL card defaults
DAUX	S	Computes derivative vector
FIELDS	S	Reads numeric data fields
HERR	E(HYDIN)	Prints diagnostic for illegally calculated hydroforce station
HFCIN	E(HYDFRC)	Initializes hydroforce calculations
HSDT	E(HSTIN)	Reads HALF-STTN cards
HSTDFN	S	Reads free-field bias-width definition
HSTFIN	E(HSTIN)	Echoes HALF-STTN card data
HSTIN	S	Sets HALF-STTN card defaults
HYDIN	S	Sets HYDRO card defaults
HYDFIN	E(HYDIN)	Echoes HYDRO card data
HYDFRC	S	Computes hydroforce
HYDRDT	E(HYDIN)	Reads HYDRO cards
INDMFIN	E(INDMPIN)	Echoes DAMPC card data
INDMPDT	E(INDMPIN)	Reads DAMPC cards
INDMPIN	S	Sets DAMPC defaults
INITDT	E(INTIN)	Reads INITIAL cards
IPWR2	F	Returns smallest power of two equally or exceeding argument
INPLFIN	E(INPLTIN)	Echoes PLOT and AXES card data
INPLTDT	E(INPLTIN)	Reads PLOT and AXES cards
INPLTIN	S	Sets PLOT and AXES card defaults
INPRDT	E(INPRIN)	Reads PRINT cards
INPRFIN	E(INPRIN)	Echoes PRINT card data
INPRIN	S	Sets PRINT card defaults
INSEGDT	E(SEAGEN)	Reads SEA GEN cards
INSEGIN	E(SEAGEN)	Sets SEA GEN card defaults

*E(nnnn) is alternate entry point of subprogram nnnn; F is function; S is subroutine.

TABLE 10 (Continued)

Name:	*Type**	*Description*
INSGFIN	E(SEAGEN)	Echoes SEA GEN card data; initializes sea generator calculations
INSMIN	E(SLAM)	Sets SLAM card defaults
INSMDT	E(SLAM)	Reads SLAM cards
INSMFIN	E(SLAM)	Echoes SLAM cards
INTIN	S	Initializes initial condition logic
INTRDT	E(INTRIN)	Reads INTGRTN cards
INTRIN	S	Sets INTGRTN card defaults
INTRFIN	E(INTRIN)	Echoes INTGRTN card data
INUPH	S	Executive routine for input data processing
ISEARCH	S	Performs binary search
KUTMER	S	Performs numerical integration
MSCDN	S	Sets SCTN and SPEED card defaults
MSCDT	E(MSCDN)	Reads SCTN and SPEED cards
MSFIN	E(MSCDN)	Echoes SCTN and SPEED card data
MYEND	S	Flushes buffers in the event of abnormal termination
NLIN	E(NLINPUT)	Sets NL_A/NL_B card defaults
NLINPUT	S	Reads NL_A/NL_B cards
NLFIN	E(NLINPUT)	Completes NL_A/NL_B card processing
NØLEX	F	Performs nonlinear expression evaluation
PLØT	E(PLØTIN)	Stores data to be plotted
PLØTFIN	E(PLOTIN)	Terminates plotting
PLØTIN	S	Initializes plotting
PUNCHIT	E(INTIN)	Punches final conditions
SCAN	S	Performs lexiographical scan
SEAGEN	S	Sea Generator
SLAM	S	Computes bottom slamming force
SSMOOTH	S	Smooths computed slamming force
STFIN	E(STIN)	Echoes STATION card data
STIN	S	Sets STATION card defaults
STNDT	E(STIN)	Reads STATION card data
TRYPIN1	S	Calculates vertical axes, plotting parameters
TRYPIN2	E(TRYPIN1)	Calculates horizontal axes, plotting parameters
TRYPLOT	E(TRYPIN1)	Plots one frame of data

*E(nnnn) is alternate entry point of subprogram nnnn; F is function; S is subroutine.

TABLE 11 - FUNCTIONING OF SUBROUTINES AND ALTERNATE ENTRIES

Name	Alternate Entries	Called By	Calls	Labeled Common Blocks
BUILD		NLINPUT	ISEARCH	
			IPWR2	
CNTRN	CNTFIN	INUPH		INPT2, INPT1
	CNTRDT			
DAUX		KUTMER	HYDFRC	HBLOK, TEST
FIELDS		INTIN		FBLOK
		HYDIN		
		SEAGEN		
		INTRIN		
		INPLTIN		
		NLINPUT		
		MSCDN		
		HSTIN		
		STIN		
		INDMPIN		
HSTDEN		INPRIN		
HSTIN	HSTFIN	INUPH	FIELDS	HBLOK, INPT2, FBLOK
	HSDT	INDMPIN		
		INTIN		

TABLE 11 (Continued)

Name	Alternate Entries	Called By	Calls	Labeled Common Blocks
HYDIN	HERR HYDRDT HYDFIN	INTIN INUPH	FIELDS	INPT2, HBLOK, FBLOK, HDBLOK
HYDFRC	HFCIN	DAUX INDMPIN	SEAGEN SLAM NOLEX	INPT1, NBLOCK, CBLOK, HBLOK, TEST, HDBLOK, PBLOK1
INDMPIN	INDMFIN INDMPDT	INUPH	HSTIN HYDFRC FIELDS	HBLOK, FBLOK, INPT2
INPLTIN	INPLFIN INPLTDT	INUPH	FIELDS	INPT2, FBLOK, PBLOK1, PBLOK4, INPT1
INPRIN	INPRDT INPRFIN	INUPH	HSTDFN	FBLOK, HBLOK, INPT1, INPT2
INTIN	INITDT PUNCHIT	INUPH	STIN INTRIN HYDIN HSTIN FIELDS	HBLOK, FBLOK, INPT1, INPT2
INTRIN	INTRDT INTRFIN	INUPH INTIN	FIELDS	FBLOK, INPT1, INPT2

TABLE 11 (Continued)

Name	Alternate Entries	Called By	Calls	Labeled Common Blocks
INUPH		SIMSHIP	CNTRN	INPT1, INPT2
			HYDIN	
			HSTIN	
			INDMPIN	
			INPLTIN	
			INPRIN	
			SEAGEN	
			INTRIN	
			MSCDN	
			NLINPUT	
			INTIN	
			STIN	
ISEARCH		BUILD		
KUTMER		SIMSHIP	DAUX	
MSCDN	MSCDT	INUPH	FIELDS	FBLOK, HBLOK, INPT1, INPT2
	MSFIN			
MYEND		ABNORMAI TERMINATION	PLOTIN	
NLINPUT	NLIN	INUPH	FIELDS	FBLOK, INPT2, CBLOK
	NLFIN		BUILD	
NOLEX		HYDFRC	SCAN	NBLOCK

TABLE 11 (Continued)

Name	Alternate Entries	Called By	Calls	Labeled Common Blocks
PLOTIN	PLOT	SIMSHIP	TRYPIN1	PBLOK1, CBLOK, PRNT1, PBLOK3
	PLOTFIN	MYEND	SEAGEN	
SCAN		NLINPUT		
SEAGEN	INSEGDT	INUPH	FIELDS	FBLOK, INPT2, HBLOK
	INSEGIN	PLOTIN		
	INSGFIN	HYDFRC		
SLAM	INSMIN	INUPH		
	INSMDT	HYDFRC	SSMOOTH	HBLOK, TEST
	INSMFIN			
SSMOOTH		SLAM		
STIN	STFIN	INUPH	FIELDS	HBLOK, INPT2, FBLOK
	STNDT	INTIN		
TRYPIN1	TRYPIN2	PLOTIN		PBLOK3, PBLOK4
	TRYPLOT			

TABLE 12 - STRUCTURAL SEAWORTHINESS DIGITAL COMPUTER PROGRAM ROSAS

```
      PROGRAM SIMSHIP(INPUT=128,OUTPUT=256,PUNCH=128,TAPE5=INPUT,TAPE8,   SMSHP     2
     1 TAPE6=OUTPUT,TAPE7=PUNCH,FILMPL=128,TAPE48=FILMPL,DEBUG=OUTPUT)    SMSHP     3
C$    DEBUG                                                               SMSHP     4
C$    ARRAYS                                                              SMSHP     5
      DIMENSION TP(2), BB(2),TPP(2),L(2)                                  SMSHP     6
      DIMENSION YH(9)                                                     SMSHP     7
      COMMON /HBLOK/ KAG(21),RINT(21),EI(20),SHPMS(20),DAMPC(20),         SMSHP     8
     1 DXI,NEVAL,SPEED,BUOY SPG(9),SCF(9),ADDMS(9),VRTMS(20),IBEG,IEND    SMSHP     9
      COMMON/INPT1/IEOF,B(11,2),TSTRT,HMNI,TF,W(10,2),NW(2),EPS(91) ,     CD5       2
     1ABSFAR(91),XLNGTH ,TCPU,CONTROL(4)                                  CD5       3
      COMMON /TEST/ IND                                                   SMSHP    11
      DIMENSION       YDOT(20),GDOT(21),BDM(20),SF(21),TEMP(21)           SMSHP    12
      EXTERNAL MYEND                                                      SMSHP    13
      EQUIVALENCE                                                         CD3       2
     1  (YS,YH),(YS(10),YDOT),(YS(30),GDOT),(YS(51),BDM),(YS(71),SF)      CD3       3
      COMMON /PRNT1/T,PITCH,TLEFT                                         SMSHP    15
      EQUIVALENCE (YS,YH)                                                 SMSHP    16
      COMMON YS(91)                                                       SMSHP    17
      DATA IPG,YES / 0,3HYES /                                            SMSHP    18
*                                                                         SMSHP    19
C**INITIALIZATION                                                         SMSHP    20
*                                                                         SMSHP    21
 5    CALL SECOND(TC1)                                                    SMSHP    22
      CALL INUPH                                                          SMSHP    23
      IF (IEOF .EQ. 1) GO TO 1050                                         SMSHP    24
      CALL PLOTIN                                                         SMSHP    25
      IND = 0                                                             SMSHP    26
      CALL SECOND(TC2)                                                    SMSHP    27
      TLEFT=.9*TCPU-TC2+TC1                                               SMSHP    28
      DO 10 K=1,2                                                         SMSHP    29
      TP(K)=B(1,K)                                                        SMSHP    30
      BB(K)=(1.+SIGN(.0001,B(2,K)))*B(2,K)                                SMSHP    31
      TPP(K)=(1.-SIGN(.0001,TP(K)))*TP(K)                                 SMSHP    32
 10   L(K)=1                                                              SMSHP    33
      T=TSTRT                                                             SMSHP    34
      NEVAL=0                                                             SMSHP    35
      ISTEPS=0                                                            SMSHP    36
*                                                                         SMSHP    37
C**TIME-INCREMENT LOOP                                                    SMSHP    38
*                                                                         SMSHP    39
 100  IF (T.GT.TF) GO TO 900                                              SMSHP    40
      IF (TP(1) .GE. 1.E20 .AND. TP(2) .GE. 1.E20) GO TO 900              SMSHP    41
C**DETERMINE INTEGRATION INTERVAL                                         SMSHP    42
*                                                                         SMSHP    43
*   WHICH IS NEXT?  PRINT(I=2), PLOT(I=1) OR BOTH(FALL THROUGH)           SMSHP    44
      DO 130 I=1,2                                                        SMSHP    45
      J=3-I                                                               SMSHP    46
      IF (TP(I) .LE. TPP(J)) GO TO 150                                    SMSHP    47
 130  CONTINUE                                                            SMSHP    48
      TT=(TP(1)+TP(2))*.5                                                 SMSHP    49
      K1=1                                                                SMSHP    50
      K2=2                                                                SMSHP    51
      GO TO 160                                                           SMSHP    52
 150  TT=TP(I)                                                            SMSHP    53
      K1=K2=I                                                             SMSHP    54
 160  H=TT-T                                                              SMSHP    55
C**DETERMINE NEXT PLOT/PRINT TIME                                         SMSHP    56
      JIND=0                                                              SMSHP    57
      DO 200 K=K1,K2                                                      SMSHP    58
      LL=L(K)                                                             SMSHP    59
      TTP=TP(K)+W(LL,K)                                                   SMSHP    60
*  WILL THE NEXT TIME GO OUT OF PRESENT INTERVAL, IF SO,                  SMSHP    61
*  ADVANCE INTERVAL                                                       SMSHP    62
      IF (TTP .GT. BB(K)) GO TO 170                                       SMSHP    63
      TP(K)=TTP                                                           SMSHP    64
      GO TO 195                                                           SMSHP    65
*  DOES THE NO. OF REQUESTED INTERVALS EXCEED THE NO. GIVEN, IF SO,       SMSHP    66
*  SET "INFINITE" STARTING TIME FOR THE NEXT OUTPUT.                      SMSHP    67
 170  IF (LL .LT. NW(K)) GO TO 180                                        SMSHP    68
```

Table 12 (Continued)

```
      TP(K)=1.E20                                                        SMSHP      69
      GO TO 195                                                          SMSHP      70
  180 LL=L(K)=L(K)+1                                                     SMSHP      71
      TP(K)=B(LL,K)+W(LL,K)                                              SMSHP      72
      BB(K)=(1.+SIGN(.0001,B(LL+1,K)))*B(LL+1,K)                         SMSHP      73
  195  TPP(K)=(1.-SIGN(.0001,TP(K)))*TP(K)                               SMSHP      74
      JIND=JIND+K                                                        SMSHP      75
  200 CONTINUE                                                           SMSHP      76
C**DETERMINE IF ENOUGH CPU TIME LEFT                                     SMSHP      77
      IF (ISTEPS .EQ. 3) GO TO 210                                       SMSHP      78
      ISTEPS=ISTEPS+1                                                    SMSHP      79
      GO TO (220,212,230) ISTEPS                                         SMSHP      80
  212 TCOMP2=0                                                           SMSHP      81
      GO TO 230                                                          SMSHP      82
  210 IF(H1 .NE. H2)          GO TO 215                                  SMSHP      83
      H1 = 0                                                             SMSHP      84
      TCOMP1 = 0.                                                        SMSHP      85
  215 FRACT = (H - H1)/(H2 - H1)                                         SMSHP      86
      TCOMP = TCOMP1 + FRACT*(TCOMP2 - TCOMP1)                           SMSHP      87
      IF (TLEFT-TCOMP.GT.0.) GO TO 230                                   SMSHP      88
      WRITE(6,850) H1,H2,H,TCOMP1,TCOMP2,TCOMP,TLEFT                     SMSHP      89
  850 FORMAT(25H COMPUTED TIME LIMIT            /                        SMSHP      90
     1       22H LAST TWO TIME STEPS    ,1P2E10.2,5X,15HNEXT TIME STEP   SMSHP      91
     2,1PE10.2 / 22H LAST TWO COMP TIMES     ,2F10.3,5X,                 SMSHP      92
     3 15HNEXT COMP TIME     ,F10.3 / 47X,15HCOMP TIME LEFT     ,F10.3)  SMSHP      93
      IF(CONTROL(2) .EQ. YES)          CALL PUNCHIT                      SMSHP      94
      GO TO 1050                                                         SMSHP      95
C**START SIMULATION WITH 2 BABY STEPS                                    SMSHP      96
  220 TIME=0                                                             SMSHP      97
      HH=1.E-05*HMNI                                                     SMSHP      98
      FIRST=0.                                                           SMSHP      99
      DO 225 KL=1,2                                                      SMSHP     100
      HMIN=HMNI                                                          SMSHP     101
      CALL SECOND (TIME1)                                                SMSHP     102
      CALL KUTMER(91,T,HH,YS,EPS,ABSERR,HMIN,FIRST)                      SMSHP     103
      CALL SECOND(TIME2)                                                 SMSHP     104
      TIME=TIME+TIME2-TIME1                                              SMSHP     105
      IF (FIRST .EQ. 2.) GO TO 875                                       SMSHP     106
  225 CONTINUE                                                           SMSHP     107
      H2=HH                                                              SMSHP     108
      H=H-HH-HH                                                          SMSHP     109
      IF (H .LE. 0) GO TO 300                                            SMSHP     110
C**INTEGRATE EQUATIONS                                                   SMSHP     111
  230 IF (H .EQ. H2) GO TO 240                                           SMSHP     112
      HMIN=H                                                             SMSHP     113
      FIRST=.5                                                           SMSHP     114
  235 IF(HMIN/HMNI .LT. 1.0001)          GO TO 240                       SMSHP     115
      HMIN=HMIN/2                                                        SMSHP     116
      GO TO 235                                                          SMSHP     117
  240 CONTINUE                                                           SMSHP     118
      IF (H .LE. 0) GO TO 100                                            SMSHP     119
      CALL SECOND(TIME1)                                                 SMSHP     120
      CALL KUTMER(91,T,H,YS,EPS,ABSERR,HMIN,FIRST)                       SMSHP     121
      CALL SECOND(TIME2)                                                 SMSHP     122
      IF(ABS(1.-H2/H) .LT. .01)        GO TO 250                         SMSHP     123
      H1 = H2                                                            SMSHP     124
      TCOMP1 = TCOMP2                                                    SMSHP     125
  250 TCOMP2 = TIME2 - TIME1                                             SMSHP     126
      TCOMP1=TCOMP2                                                      SMSHP     127
      TLEFT=TLEFT-TCOMP2                                                 SMSHP     128
      TIME=TIME+TCOMP2                                                   SMSHP     129
      H2=H                                                               SMSHP     130
      IF (FIRST .EQ. 2.) GO TO 875                                       SMSHP     131
*                                                                        SMSHP     132
C**COMPUTE THE PITCH                                                     SMSHP     133
*                                                                        SMSHP     134
  300 PITCH=57.296*ATAN((YH(9)-YH(1))/XLNGTH)                            SMSHP     135
*                                                                        SMSHP     136
C** OUTPUT RESULTS AND THEN JUMP BACK FOR ANOTHER TIME INCREMENT          SMSHP     137
*                                                                        SMSHP     138
```

Table 12 (Continued)

```
      IF (JIND .EQ. 2) GO TO 350                                        SMSHP   139
      CALL PLOT                                                         SMSHP   140
      IF (JIND .EQ. 1) GO TO 100                                        SMSHP   141
  350 INDSAV = IND                                                      SMSHP   142
      TX=(1.+1.E-10)*T                                                  SMSHP   143
      IND = 1                                                           SMSHP   144
      CALL HYDFRC (TX,YH,YDOT,TEMP)                                     SMSHP   145
      IND = INDSAV                                                      SMSHP   146
      IPG = IPG + 1                                                     SMSHP   147
      WRITE(6,49) T,TIME,HMIN,NEVAL                                     SMSHP   148
   48 FORMAT(1H1)                                                       SMSHP   149
   49 FORMAT(1H-/// 5H TIME,F14.5,5H SECS,F17.3,14H SECS CPU TIME /     SMSHP   150
     1 10H TIME STEP ,F9.5,5H SECS,I17,23H DERIVATIVE EVALUATIONS)      SMSHP   151
      WRITE (6,50)                                                      SMSHP   152
      WRITE (6,55) (YH(I),   I =  1, 9)                                 SMSHP   153
      WRITE (6,51)                                                      SMSHP   154
      WRITE (6,55) (YDOT(I), I =  1,20)                                 SMSHP   155
      WRITE (6,56) PITCH                                                SMSHP   156
      WRITE (6,52)                                                      SMSHP   157
      WRITE (6,55) (GDOT(I), I =  1,21)                                 SMSHP   158
      WRITE (6,53)                                                      SMSHP   159
      WRITE (6,55) (BDM(I) , I =  1,20)                                 SMSHP   160
      WRITE (6,54)                                                      SMSHP   161
      WRITE (6,55) (SF(I)  , I =  1,21)                                 SMSHP   162
   50 FORMAT (20H0VERTICAL POSITION   ,1P10E10.2)                       SMSHP   163
   51 FORMAT (20H0VERTICAL VELOCITY   ,1P10E10.2)                       SMSHP   164
   52 FORMAT (20H0ANGULAR VELOCITY    ,1P10E10.2)                       SMSHP   165
   53 FORMAT (20H0BENDING MOMENT      ,1P10E10.2)                       SMSHP   166
   54 FORMAT (20H0SHEAR FORCE         ,1P10E10.2)                       SMSHP   167
   55 FORMAT ( 1H                     ,1P10E12.4)                       SMSHP   168
   56 FORMAT (20H0PITCH               ,1P10E10.3)                       SMSHP   169
      GO TO 100                                                         SMSHP   170
*                                                                       SMSHP   171
C**COME HERE FOR CASE TERMINATION                                       SMSHP   172
*                                                                       SMSHP   173
  875 WRITE(6,880)                                                      SMSHP   174
  880 FORMAT(1X,*STOPPED WITH FIRST  = 2.*)                             SMSHP   175
  900 IF (CONTROL(2) .EQ. YES) CALL PUNCHIT                             SMSHP   176
      GO TO 5                                                           SMSHP   177
*                                                                       SMSHP   178
C** JOB TERMINATION                                                     SMSHP   179
*                                                                       SMSHP   180
 1050 CALL PLOTFIN                                                      SMSHP   181
      STOP                                                              SMSHP   182
      END                                                               SMSHP   183
```

Table 12 (Continued)

```
      SUBROUTINE HYDFRC (T,YH,YHDOT,HF)                                  HDFRC      2
C$    DEBUG                                                              HDFRC      3
C$    STORES(HFTEMP,VMTEMP,KCOND,IAD)                                    HDFRC      4
      REAL NOLEX                                                         HDFRC      5
      REAL KAG                                                           HDFRC      6
      DIMENSION KAG(21),RINT(21),EI(20),SHP MS(20),DAMPC(20)             HDFRC      7
     1          ,SCF(9),BUOY SPG(9),ADD MS(9),          VRT MS(20)       HDFRC      8
     2          ,YH(9),YHDOT(20),YR(9),YRDOT(9),YWDD(9),VR(9),HF(21)     HDFRC      9
     3          ,YW(9),YWDOT(9)                                          HDFRC     10
     4          ,AM(2),BS(2),SC(2),HD(2)                                 HDFRC     11
     5          ,SMFRC(6,2),SMPTS(12)                                    HDFRC     12
      DIMENSION CS1(10)                                                  HDFRC     13
C-----------------------------,----------------------------------------- HDFRC     14
      DIMENSION NLINV(9), ZBY4DXI(9),HFTEMP(9),IBNL(5,9),ITYPE(18)       HDFRC     15
      DIMENSION YCRIT(18), VCRIT(18)                                     HDFRC     16
      DIMENSION IBSTORE(350),BSTORE(350),VMTEMP(9), IMNL(5,9)            HDFRC     17
      DIMENSION IMSTORE(350,2),AMSTORE(350,2),NAPPLY(20),NL(9)           HDFRC     18
      EQUIVALENCE (IBSTORE,BSTORE),(IMSTORE,AMSTORE)                     HDFRC     19
      COMMON/NBLOCK/YDELT                                                HDFRC     20
      COMMON/CBLOK/NCALC,NHSTNS,NGROUP,NL,NLINV,IMNL,IBNL,ITYPE,YCRIT    HDFRC     21
     2,VCRIT,IBSTORE,IMSTORE,KWORDS(2),NP(2),IFIRST,NAPPLY               HDFRC     22
      COMMON /HBLOK/ KAG,RINT,EI,SHPMS,DAMPC,DXI,NEVAL,SPEED,            HDFRC     23
     1 BUOY SPG,SCF,ADDMS,VRTMS,IBEG,IEND                                HDFRC     24
      COMMON /HDBLOK/ HYDDAMP(9)                                         HDFRC     25
      COMMON/PBLOK1/NR,TTT(20),R(20),MSBSTN,NAME,CODE,PHONE              CD6       2
      COMMON/INPT1/IEOF,B(11,2),TSTRT,HMNI,TF,W(10,2),NW(2),EPS(91) ,    CD5       2
     1ABSERR(91),XLNGTH ,TCPU,CONTROL(4)                                 CD5       3
      COMMON /TEST/ IND                                                  HDFRC     28
      DATA TYR,TYRDOT,TYWDD,TWR,TAM,THF1,THF2,THF3,THF4,THF5,TWM,THF6    HDFRC     29
     1    ,THF7,TWH,TWV,SLM/                                            HDFRC     30
     1     9H REL DISP,8H REL VEL,9H WAVE ACC,9H RL WV VL,9H ADD MASS    HDFRC     31
     2    ,9H BUOY SPG,9H N-L B SP,9H N-L A MS,8H SPC DRV,4H SCF         HDFRC     32
     3,7H VRT MS,8H HYD FRC,8H HYD DMP,7H WV HGT,7H WV VEL,8H SLM FRC/   HDFRC     33
      DATA SMOTH/7H SMOOTH/                                              HDFRC     34
      DATA PI,LABEL / 3.14159265,0 /,AM,BS,SC,HD/1H ,10HADDED MASS,      HDFRC     35
     1     10H         BU , 10HOY. SPRING ,1H ,10HSMITH C.F.           HDFRC     36
     2    ,10H        HYD,10HRO DAMPING/,YES     /  3HYES    /,TWO PI   HDFRC     37
     3    / 6.283185307  /                                              HDFRC     38
      DATA CS1 / .000,.174,.342,.500,.643,.766,.866,.940,.985,1.00 /     HDFRC     39
      IF (NEVAL .EQ. IBEG)   IND = 1                                     HDFRC     40
      IF (NEVAL .EQ. IEND)   IND = 0                                     HDFRC     41
      IF(T .EQ. OLD T)                  RETURN                           HDFRC     42
      OLD T = T                                                          HDFRC     43
  100 CONTINUE                                                           HDFRC     44
      IM=0                                                               HDFRC     45
      IB=0                                                               HDFRC     46
      CALL SEAGEN(YW,YWDOT,YWDD,CEL,T)                                   HDFRC     47
   71 CONTINUE                                                           HDFRC     48
*                                                                        HDFRC     49
*  COMPUTE THE SHIPS VERTICAL DISPLACEMENT AND VELOCITY RELATIVE         HDFRC     50
*  TO THE SEA                                                            HDFRC     51
      DO 110 I=1,NCALC                                                   HDFRC     52
      YR(I) = YH(I) - YW(I)                                              HDFRC     53
      J=NLINV(I)+1                                                       HDFRC     54
      IF (J .NE. 1) GO TO 80                                             HDFRC     55
*  IF YRDOT(1) NEEDED, EXTRAPOLATE YHDOT FROM STATIONS .5 AND 1.5        HDFRC     56
      YRDOT(I) = 1.5 * YHDOT(1) - .5 * YHDOT(2) - YWDOT(I)               HDFRC     57
      GO TO 110                                                          HDFRC     58
 80   IF (J .NE.NHSP1) GO TO 90                                          HDFRC     59
      YRDOT(I) = -.5 * YHDOT(J-2) + 1.5 * YHDOT(J-1) - YWDOT(I)          HDFRC     60
      GO TO 110                                                          HDFRC     61
   90 YRDOT(I) = .5 * (YHDOT(J-1) + YWDOT(J)) - YWDOT(I)                 HDFRC     62
 110  CONTINUE                                                           HDFRC     63
*                                                                        HDFRC     64
*  COMPUTE THE VERTICAL VELOCITY OF THE WAVE RELATIVE TO THE SHIP        HDFRC     65
      VR(1)=YRDOT(1)-ZBY4DXI(1)*(YR (2)-YR(1))                           HDFRC     66
      VR(NCALC)=YRDOT(NCALC)-ZBY4DXI(NCALC)*(YR(NCALC)-YR(NCLCM1) )      HDFRC     67
      DO 135 I=2,NCLCM1                                                  HDFRC     68
 135  VR(I)=YRDOT(I)-ZBY4DXI(I)*(YR(I+1)-YR(I-1))                       HDFRC     69
*                                                                        HDFRC     70
```

Table 12 (Continued)

```
*  BEGIN ACCUMULATING THE HYDRODYNAMIC FORCE                                  HDFRC    71
*                                                                             HDFRC    72
*  LOOP TO 400 FOR BUOYANCY AND ADDED MASS AT EACH CALCULATED FORCE           HDFRC    73
*  HALF STATION                                                               HDFRC    74
*                                                                             HDFRC    75
      DO 400 I=1,NCALC                                                       HDFRC    76
      HFTEMP(I)=-BUOYSPG(I)*YR(I)                                            HDFRC    77
      JJGROUP=0                                                              HDFRC    78
      IF (CONTROL(3) .NE. YES .OR. NL(I) .EQ. 0 .OR. NL(I) .EQ. 2)           HDFRC    79
     1 GO TO 300                                                             HDFRC    80
*  COMPUTE NON-LINEAR BUOYANCY IF SELECTED.                                   HDFRC    81
      IB=IB+1                                                                HDFRC    82
      JGROUP=JJGROUP=IBNL(1,IB)                                              HDFRC    83
      JUMP=1                                                                 HDFRC    84
 200  KCOND=1                                                                HDFRC    85
      GO TO (280,220,240,220) ITYPE(JGROUP)                                  HDFRC    86
 220  YDELT=YR(I)-YCRIT(JGROUP)                                              HDFRC    87
      IF (YDELT .GT. 0) KCOND=2                                              HDFRC    88
      IF (ITYPE(JGROUP) .EQ. 2) GO TO 270                                    HDFRC    89
      IPLUS=2                                                                HDFRC    90
      GO TO 250                                                              HDFRC    91
 240  IPLUS=1                                                                HDFRC    92
 250  IF (VR(I) .GT. VCRIT(JGROUP  )) KCOND=KCOND+IPLUS                      HDFRC    93
 270  GO TO (280,380)JUMP                                                    HDFRC    94
 280  IAD=IBNL(KCOND+1,IB)                                                   HDFRC    95
      HFTEMP(I)=HFTEMP(I)+NOLEX(IBSTORE(IAD),BSTORE(IAD+1),YR(I))            HDFRC    96
 300  VMTEMP(I)=ADD MS(I)                                                    HDFRC    97
      IF (CONTROL(3).NE.YES.OR.NL(I).EQ. 0.OR.NL(I).EQ.1) GO TO 400          HDFRC    98
*  COMPUTE NON-LINEAR ADDED MASS IF SELECTED.                                 HDFRC    99
      IM=IM+1                                                                HDFRC   100
      JGROUP=IMNL(1,IM)                                                      HDFRC   101
      JUMP=2                                                                 HDFRC   102
      IF (JGROUP .NE. JJGROUP) GO TO 200                                     HDFRC   103
 380  IAD=IMNL(KCOND+1,IM)                                                   HDFRC   104
      VMTEMP(I)=VMTEMP(I)+NOLEX(IMSTORE(IAD,1),AMSTORE(IAD+1,1)              HDFRC   105
     1 YR(I))                                                                HDFRC   106
*  ADD THE NON-LINEAR ADDED MASS TERM TO HYDRO FORCE.                         HDFRC   107
      HFTEMP(I)=HFTEMP(I)-VR(I)*YRDOT(I)*NOLEX(IMSTORE(IAD,2),               HDFRC   108
     1 AMSTORE(IAD+1,2),YR(I))                                               HDFRC   109
 400  CONTINUE                                                               HDFRC   110
      IF (IND .NE. 1) GO TO 420                                              HDFRC   111
      WRITE(6,410) T,NEVAL                                                   HDFRC   112
  410 FORMAT(6H1TIME ,F11.6,I16,10H FCN EVALS  )                             HDFRC   113
   99 FORMAT(A9,(9E13.6))                                                    HDFRC   114
      WRITE (6,99) TWH,YW                                                    HDFRC   115
      WRITE (6,99) TWV,YWDOT                                                 HDFRC   116
      WRITE (6,99) TYWDD,YWDD                                                HDFRC   117
      WRITE (6,99) TYR,YR                                                    HDFRC   118
      WRITE (6,99) TYRDOT,YRDOT                                              HDFRC   119
      WRITE (6,99) TVR,VR                                                    HDFRC   120
      WRITE (6,99) TAM,(VMTEMP(I),I=1,NCALC)                                 HDFRC   121
      WRITE (6,99) THF3,(HFTEMP(I),I=1,NCALC)                                HDFRC   122
*                                                                            HDFRC   123
*  COMPUTE THE SPACE DERIVATIVE TERMS AND ADD TO THE HYDROFORCE               HDFRC   124
*                                                                            HDFRC   125
 420  DO 430 I=1,NCALC                                                       HDFRC   126
      I1=I-1                                                                 HDFRC   127
      IF (I.EQ. 1) I1=1                                                      HDFRC   128
      I2=I+1                                                                 HDFRC   129
      IF (I .EQ. NCALC) I2=NCALC                                             HDFRC   130
      HFTEMP(I)=HFTEMP(I)+ZBY4DXI(I)*(VMTEMP(I)*(YRDOT(I2)-YRDOT(I1))       HDFRC   131
     1+VMTEMP(I2)*VR(I2)-VMTEMP(I1)*VR(I1))                                  HDFRC   132
 430  CONTINUE                                                               HDFRC   133
      IF (IND .EQ. 1) WRITE(6,99) THF4,(HFTEMP(I),I=1,NCALC)                 HDFRC   134
*                                                                            HDFRC   135
*  COMPUTE THE SMITH CORRECTION AND ITS ASSOCIATED TERM AND ADD TO THE       HDFRC   136
*  HYDROFORCE                                                                HDFRC   137
*                                                                            HDFRC   138
      TEMP = ABS(CEL + SPEED)                                                HDFRC   139
      IF (TEMP .GT. 1.E-10)  GO TO 460                                       HDFRC   140
      TEMP = 0.                                                              HDFRC   141
      GO TO 461                                                              HDFRC   142
```

Table 12 (Continued)

```
  460 TEMP = (CEL / TEMP)**2                                          HDFRC    143
  461 CONTINUE                                                        HDFRC    144
      DO 470 I=1,NCALC                                                HDFRC    145
  470 HFTEMP(I)=HFTEMP(I)+(VMTEMP(I)+TEMP*SCF(I))*YWDD(I)             HDFRC    146
      IF (IND .EQ. 1) WRITE(6,99) THF5,(HFTEMP(I),I=1,NCALC)          HDFRC    147
*                                                                     HDFRC    148
*  COMPUTE THE HYDRODAMPIING AND ADD TO THE HYDROFORCE                HDFRC    149
*                                                                     HDFRC    150
      IF (CONTROL(4) .NE. YES) GO TO 640                              HDFRC    151
      DO 620 I=1,NCALC                                                HDFRC    152
  620 HFTEMP(I)=HFTEMP(I)-HYDDAMP(I)*VR(I)                            HDFRC    153
      IF (IND .EQ. 1) WRITE(6,99) THF7,(HFTEMP(I),I=1,NCALC)          HDFRC    154
  640 CONTINUE                                                        HDFRC    155
*                                                                     HDFRC    156
*  DISTRIBUTE THE HYDROFORCE AND VIRTUAL MASS TO ALL HALF STATIONS    HDFRC    157
*                                                                     HDFRC    158
      DO 690 J=1,NHSTNS                                               HDFRC    159
      K=NAPPLY(J)                                                     HDFRC    160
      IF (K .EQ. 0) GO TO 685                                         HDFRC    161
      HF(J)=HFTEMP(K)                                                 HDFRC    162
      VRTMS(J)=SHPMS(J)+VMTEMP(K)                                     HDFRC    163
      GO TO 690                                                       HDFRC    164
  685 HF(J)=0.                                                        HDFRC    165
      VRTMS(J)=SHPMS(J)                                               HDFRC    166
  690 CONTINUE                                                        HDFRC    167
      CALL SLAM (YRDOT,YR,YM,SMFRC,T,SMPTS)                           HDFRC    168
      DO 693 IIS=1,12                                                 HDFRC    169
      ISTN = IIS + 9                                                  HDFRC    170
  693 HF(ISTN) = HF(ISTN) + SMPTS(IIS)                                HDFRC    171
  696 IF (IND .NE. 1) RETURN                                          HDFRC    172
      WRITE (6,99) SLM,((SMFRC(I,J),J=1,2),I=1,6)                     HDFRC    173
      WRITE (6,99) SMOTH,(SMPTS(I),I=1,12)                            HDFRC    174
      WRITE(6,99) THF6,(HF(J),J=1,NHSTNS)                             HDFRC    175
      DO 700 I=1,NCALC                                                HDFRC    176
      K=NLINV(I)                                                      HDFRC    177
  700 VMTEMP(I)=VRTMS(K)                                              HDFRC    178
      WRITE(6,99) TVM,(VMTEMP(I),I=1,NCALC)                           HDFRC    179
      RETURN                                                          HDFRC    180
      ENTRY HFCIN                                                     HDFRC    181
      OLD T=-1.E20                                                    HDFRC    182
      NHSP1=NHSTNS+1                                                  HDFRC    183
      NCLCM1=NCALC-1                                                  HDFRC    184
      IND = 0                                                         HDFRC    185
C     U8Y4DXI = SPEED / (4*DXI)                                       HDFRC    186
      DO 25 I=1,NCALC                                                 HDFRC    187
      I1=I-1                                                          HDFRC    188
      IF (I.EQ. 1) I1=1                                               HDFRC    189
      I2=I+1                                                          HDFRC    190
      IF (I.EQ. NCALC) I2=NCALC                                       HDFRC    191
   25 Z8Y4DXI(I)=SPEED/((NLINV(I2)-NLINV(I1))*DXI)                    HDFRC    192
      DO 63 J=1,NHSTNS                                                HDFRC    193
      I=NAPPLY(J)                                                     HDFRC    194
      IF (NAPPLY(J) .EQ. 0) GO TO 62                                  HDFRC    195
      VRTMS(J)=SHPMS(J)+ADDMS(I)                                      HDFRC    196
      GO TO 63                                                        HDFRC    197
   62 VRTMS(J)=SHPMS(J)                                               HDFRC    198
   63 CONTINUE                                                        HDFRC    199
      RETURN                                                          HDFRC    200
      END                                                             HDFRC    201
```

Table 12 (Continued)

```
      SUBROUTINE SLAM (YRDOT,YR,YW,SMFRC,T,SMPTS)                          SLAM       2
C     THIS IS MULTI-SLAM SMOOTHING ROUTINE                                 SLAM       3
      DIMENSION YRDOT(9),YR(9),YW(9)                                       SLAM       4
      DIMENSION SMFRC(6,2),SMPTS(12)                                       SLAM       5
      DIMENSION COEF(8,3),WIDTH(10),DRHGT(10)                              SLAM       6
      COMMON/CELER/CELT                                                    SLAM       7
C     THIS COMMON IS SAME AS CD1, EXCEPT DX REPLACES DXI                   SLAM       8
      COMMON /HBLOK/ KAG(21),RINT(21),EI(20),SHPMS(20),DAMPC(20),          SLAM       9
     1DX,NEVAL,SPEED,BUOY SPG(9),SCF(9),ADDMS(9),VRTMS(20),IBEG,IEND       SLAM      10
      COMMON /PLTSLM/ RAWSLM,SSMPTS                                        SLAM      11
      COMMON /IHVSS/ IHVS                                                  SLAM      12
      DATA ISTSLM/0/                                                       SLAM      13
C                                                                          SLAM      14
C     VARIABLE  DEFINITION  (SYMBOL)                                       SLAM      15
C     --------  ----------  --------                                       SLAM      16
C     BTANG  BUTTOCK ANGLE  (ALPHA)                                        SLAM      17
C     DRANG  DEAD RISE ANGLE  (BETA)                                       SLAM      18
C     TRANG  TRIM ANGLE  (TAU)                                             SLAM      19
C     PHI  IMPACT ANGLE  (PHI)                                             SLAM      20
C     THETA  WAVE ANGLE  (THETA)                                           SLAM      21
C     SINPHI  SIN OF ANGLE PHI                                             SLAM      22
C     COSPHI  COSINE OF ANGLE PHI                                          SLAM      23
C     TANPHI  TANGENT OF ANGLE PHI                                         SLAM      24
C     SINTHE  SIN OF ANGLE THETA                                           SLAM      25
C     COSTHE  COSINE OF ANGLE THETA                                        SLAM      26
C     TANTHE  TANGENT OF ANGLE THETA                                       SLAM      27
C     SINAPH  SIN OF ANGLE (ALPHA + PHI)                                   SLAM      28
C     COSAPH  COSINE OF ANGLE (ALPHA + PHI)                                SLAM      29
C     TANAPH  TANGENT OF ANGLE (ALPHA + PHI)                               SLAM      30
C     SINTAL  SIN OF ANGLE (TAU + ALPHA)                                   SLAM      31
C     COSTAL  COSINE OF ANGLE (TAU + ALPHA)                                SLAM      32
C     TANTAL  TANGENT OF ANGLE (TAU + ALPHA)                               SLAM      33
C     COSBET  COSINE OF ANGLE BETA                                         SLAM      34
C     TANBET  TANGENT OF ANGLE BETA                                        SLAM      35
C     TANXI  TANGENT OF ANGLE XI                                           SLAM      36
C     SECBEH  SECANT OF ANGLE BETA(SUB EH)                                 SLAM      37
C     WIDTH(10)  WIDTH OF SLAM REGION  (SMALL B)                           SLAM      38
C     DRHGT(10)  HEIGHT OF DEAD RISE  (B)                                  SLAM      39
C     COEF(7,3)  POLYNOMIAL ORDERS + COEFFICIENTS                          SLAM      40
C     DRAFT  SHIPS DRAFT  (D)                                              SLAM      41
C     FACTK  SLAM PRESSURE FACTOR  (K(SUB XI))                             SLAM      42
C     SMLGT  LENGTH OF SLAM REGION  (DELTA SMALL L)                        SLAM      43
C     VLNIMP  IMPACT VELOCITY - NORMAL  (V(SUB N))                         SLAM      44
C     VLTIMP  IMPACT VELOCITY - TANGENTIAL  (V(SUB T))                     SLAM      45
C     VR(9)  RELATIVE VERTICAL VELOCITY  (V(SUB R))                        SLAM      46
C     YR(9)  RELATIVE VERTICAL DISPLACEMENT  (Y(SUB R))                    SLAM      47
C     YW(9)  WAVE HEIGHT  (Y(SUB W))                                       SLAM      48
C     SMSTRT  POINT WHERE SLAM STARTS  (X(SUB 0))                          SLAM      49
C     DX  STATION LENGTH  (DELTA X)                                        SLAM      50
C     DX2  2 TIMES STATION LENGTH  (2*DX)                                  SLAM      51
C     DX10  10 TIMES STATION LENGTH  (10*DX)                               SLAM      52
C     SMEND  POINT WHERE SLAM ENDS  (X(SUB E))                             SLAM      53
C     XS  DISTANCE FROM THE STERN AT WHICH THE SLAM FORCE SHOULD BE APPL   SLAM      54
C     SMWGT1  WEIGHTS USED TO DISTRIBUTE THE SLAM FORCE  (W(SUB S+))       SLAM      55
C     SMWGT2  WEIGHTS USED TO DISTRIBUTE THE SLAM FORCE  (W(SUB S-))       SLAM      56
C     SMFRC(6,2)  SLAM FORCES  (F(SUB S))                                  SLAM      57
C     DENS  MASS DENSITY OF WATER  (RHO)                                   SLAM      58
C     DNSBY2  MASS DENSITY OF WATER DIVIDED BY 2                           SLAM      59
C     EFIMPA  EFFECTIVE IMPACT ANGLE  (XI)                                 SLAM      60
C     KSLAM  NUMBER OF THE FIRST OF THE TWO STATIONS AT WHICH SLAM FORCE   SLAM      61
C            IS APPLIED  (K(SUB S))                                        SLAM      62
C     NEND  NUMBER OF STATION CLOSEST TO X(SUB E)  (N(SUB E))              SLAM      63
C                                                                          SLAM      64
C                                                                          SLAM      65
C*************************************************************************SLAM      66
C*    PROBABLY SHOULD BE MODIFIED SO THAT SLAM OCCURS IF               *   SLAM      67
C*    SHOULDER IS ABOVE WATER, RATHER THAN KEEL.                       *   SLAM      68
C*************************************************************************SLAM      69
C                                                                          SLAM      70
C                                                                          SLAM      71
```

Table 12 (Continued)

```
C                                                                       SLAM      72
C     TEST IF THIS IS FIRST CALL TO SLAM                                SLAM      73
C          IF SO, SET  SMFRC = 0.                                       SLAM      74
C                                                                       SLAM      75
      IF (ISTSLM .NE. 0)  GO TO 10                                      SLAM      76
      DO 5 I=1,2                                                        SLAM      77
      DO 5 J=1,6                                                        SLAM      78
    5 SMFRC(J,I) = 0.                                                   SLAM      79
      RAWSLM = 0.                                                       SLAM      80
      ISTSLM = 1                                                        SLAM      81
C                                                                       SLAM      82
C                                                                       SLAM      83
C     DETERMINE WHETHER TO START WITH TYPE1 OR TYPE2 SLAM               SLAM      84
C                                                                       SLAM      85
   10 CONTINUE                                                          SLAM      86
C                                                                       SLAM      87
C     CALL SLAM ON EVERY 5TH. (OR MULTIPLE OF) EVALUATION               SLAM      88
      RAWSLM = 0.                                                       SLAM      89
      IIHVS = IHVS - 4                                                  SLAM      90
      KNEVAL = 5                                                        SLAM      91
      IF (IHVS .GE. 5)  KNEVAL = KNEVAL * 2**IIHVS                      SLAM      92
      IF (MOD(NEVAL,KNEVAL) .NE. 0)  GO TO 301                          SLAM      93
C                                                                       SLAM      94
      IF (YR(9) .LT. DRAFT)  GO TO 800                                  SLAM      95
  804 NSLM = 1                                                          SLAM      96
      ICASE = 1                                                         SLAM      97
      SGNFTR = 1.                                                       SLAM      98
      ISGNF = IFIX(SGNFTR)                                              SLAM      99
  801 DO 290  KBWD=1,4                                                  SLAM     100
      KFWD = 9 - KBWD                                                   SLAM     101
      TEMP = YR(KFWD) - DRAFT                                           SLAM     102
      IF (TEMP*SGNFTR .GE. 0.)  GO TO 290                               SLAM     103
C                                                                       SLAM     104
C     FIND THE POINT WHERE THE KEEL TOUCHES THE WAVE SURFACE            SLAM     105
C                                                                       SLAM     106
      TANPHI = (YR(KFWD+1) - YR(KFWD)) / DX2                            SLAM     107
      TEMP = -(TEMP/TANPHI)                                             SLAM     108
      SMSTRT = (2*KFWD+1) * DX + TEMP                                   SLAM     109
      TEMP = TEMP / DX2                                                 SLAM     110
C                                                                       SLAM     111
C     COMPUTE THE VERTICAL VELOCITY AT THE POINT WHERE THE KEEL TOUCHES SLAM     112
C     THE WAVE SURFACE.  IF THIS POINT NOT MOVING DOWN, LOOK FOR NEXT   SLAM     113
C     SLAM TYPE.                                                        SLAM     114
C                                                                       SLAM     115
      VRTVLC = TEMP * YRDOT(KFWD+1) + (1.-TEMP) * YRDOT(KFWD)           SLAM     116
      IF (ICASE .EQ. 2) GO TO 803                                       SLAM     117
      IF (VRTVLC .GE. 0.)  GO TO 300                                    SLAM     118
      GO TO 900                                                         SLAM     119
  803 IF (VRTVLC .GE. 0.) GO TO 804                                     SLAM     120
      GO TO 900                                                         SLAM     121
C                                                                       SLAM     122
C     DETERMINE THE LENGTH OF SLAM REGION                               SLAM     123
C                                                                       SLAM     124
  900 SMLGT = 0.                                                        SLAM     125
      SMEND = SMSTRT                                                    SLAM     126
      NEND = 20 * ISGNF                                                 SLAM     127
      DO 25 J=1,10                                                      SLAM     128
      M = IFIX(SMEND/DX + .5) - 10                                      SLAM     129
      IF ( (M-NEND) * ISGNF .GE. 0) GO TO 30                            SLAM     130
      NEND = M                                                          SLAM     131
      IF (ISGNF .EQ. 1)  SMLGT = DRHGT(NEND) / TANPHI                   SLAM     132
      IF (ISGNF .EQ. -1)  SMLGT = SMLGT - ((DRHGT(NEND) + SMLGT *       SLAM     133
     1                                    TANPHI) / (BSLP - TANPHI))    SLAM     134
      SMEND = SMSTRT - SMLGT                                            SLAM     135
      IF (SMEND .GE. DX20) GO TO 29                                     SLAM     136
      IF (SMEND .LT. DX105) GO TO 28                                    SLAM     137
   25 CONTINUE                                                          SLAM     138
      GO TO 30                                                          SLAM     139
C                                                                       SLAM     140
C     THE SLAM FORCE SHOULD BE APPLIED                                  SLAM     141
C     AT THE DISTANCE X(S) FROM THE STERN                               SLAM     142
C                                                                       SLAM     143
```

Table 12 (Continued)

```
   28 SMEND = DX105                                                        SLAM      144
      GO TO 295                                                            SLAM      145
   29 SMEND = DX20                                                         SLAM      146
  295 SMLGT = SMSTRT - SMEND                                               SLAM      147
   30 XS = (5.*SMSTRT + 4.*SMEND) / 9.                                     SLAM      148
C                                                                          SLAM      149
C     THE SLAM FORCE WILL BE APPLIED AT STATIONS NUMBERED K(S) AND K(S)+   SLAM      150
C     USING WEIGHTS W(S+) AND W(S-)                                        SLAM      151
C                                                                          SLAM      152
      SMWGT1 = XS/DX + .5                                                  SLAM      153
      KSLAM = IFIX(SMWGT1)                                                 SLAM      154
      SMWGT1 = SMWGT1 - FLOAT(KSLAM)                                       SLAM      155
      SMWGT2 = 1. - SMWGT1                                                 SLAM      156
C                                                                          SLAM      157
C     DETERMINE THE EFFECTIVE IMPACT ANGLE XI                              SLAM      158
C                                                                          SLAM      159
      COSPHI = 1. / SQRT(1. + TANPHI**2)                                   SLAM      160
      SINPHI = TANPHI * COSPHI                                             SLAM      161
      SINAPH = SINAL * COSPHI + COSAL * SINPHI                             SLAM      162
      COSAPH = COSAL * COSPHI - SINAL * SINPHI                             SLAM      163
      TANAPH = SINAPH / COSAPH                                             SLAM      164
      TEMP1 = TANBET / (SINPHI + TANAL * COSPHI)                           SLAM      165
      SECBEH = SQRT(1. + TEMP1**2) * SGNFTR                                SLAM      166
      COSBEH = 1. / SECBEH                                                 SLAM      167
      TANXI = TANAPH * SECBEH                                              SLAM      168
      EFIMPA = ATAN(TANXI)                                                 SLAM      169
C                                                                          SLAM      170
C     CALCULATE THE TANGENTIAL AND NORMAL IMPACT VELOCITIES                SLAM      171
C                                                                          SLAM      172
      TANTHE = (YW(KFWD+1) - YW(KFWD)) / DX2                               SLAM      173
      COSTHE = 1. / SQRT(1. + TANTHE**2)                                   SLAM      174
      SINTHE = TANTHE * COSTHE                                             SLAM      175
      SINTAL = SINAPH * COSTHE   +  COSAPH * SINTHE                        SLAM      176
      COSTAL = COSAPH * COSTHE - SINAPH * SINTHE                           SLAM      177
      TEMP = VRTVLC * COSTAL   +  (SPEED+CELT) * SINTAL                    SLAM      178
      VLNIMP = TEMP * COSAPH / COSTAL**2                                   SLAM      179
      VLTIMP = TEMP  * SINAPH                                              SLAM      180
C                                                                          SLAM      181
C     CALCULATE THE FACTOR K(XI)  (FACTK)                                  SLAM      182
C                                                                          SLAM      183
      IF (EFIMPA .GE. .0383972) GO TO 85                                   SLAM      184
      FACTK = POLY(COEF(1,1),EFIMPA)                                       SLAM      185
      GO TO 110                                                            SLAM      186
   85 IF (EFIMPA .GE. .191986) GO TO 90                                    SLAM      187
      FACTK = POLY(COEF(1,2),EFIMPA)                                       SLAM      188
      GO TO 110                                                            SLAM      189
   90 IF (EFIMPA .GE. .349066) GO TO 100                                   SLAM      190
      FACTK = POLY(COEF(1,3),EFIMPA)                                       SLAM      191
      GO TO 110                                                            SLAM      192
C     .76856471/288. = .00266862747                                        SLAM      193
C     .00266862747 * 2.4674 = .00658457141                                 SLAM      194
  100 FACTK = .00266862747 + .00658457141 / TANXI**2                       SLAM      195
C                                                                          SLAM      196
C     CALCULATE THE SLAM PRESSURE                                          SLAM      197
C                                                                          SLAM      198
  110 PRESS = FACTK * VLNIMP**2 * 144.                                     SLAM      199
      PRESS = PRESS + (.5 * VLTIMP**2 * COSBEH)                            SLAM      200
      PRESS = PRESS * DENS / 2240.                                         SLAM      201
C                                                                          SLAM      202
C     CALCULATE AND DISTRIBUTE THE SLAM FORCE                              SLAM      203
C                                                                          SLAM      204
      TEMP = COSPHI * COSTHE - SINPHI * SINTHE                             SLAM      205
      TEMP = .5 * PRESS * DRHGT(NEND) * SGNFTR * SMLGT * TEMP              SLAM      206
C     CHANGE FORCE TO FORCE/FOOT                                           SLAM      207
      TEMP = TEMP/DX                                                       SLAM      208
C                                                                          SLAM      209
      TEMP=5.*TEMP                                                         SLAM      210
C     TEMPORARY INCREASE IN SLAM FOR DISPLAY PURPOSES                      SLAM      211
```

Table 12 (Continued)

```
C                                                                       SLAM     212
      RAWSLM = RAWSLM + TEMP                                            SLAM     213
      TIMCUT = YR(KFWD) - DRHGT(KSLAM-10)                               SLAM     214
      SMFRC(1,NSLM) = FLOAT(KSLAM)                                      SLAM     215
      SMFRC(2,NSLM) = SMWGT2 * TEMP                                     SLAM     216
      SMFRC(3,NSLM) = SMWGT1 * TEMP                                     SLAM     217
      SMFRC(4,NSLM) = YR(KFWD)                                          SLAM     218
      SMFRC(5,NSLM) = TIMCUT                                            SLAM     219
      SMFRC(6,NSLM) = KFWD                                              SLAM     220
      IF (ICASE .EQ. 1)  GO TO 300                                      SLAM     221
      NSLM = NSLM - 1                                                   SLAM     222
      ICASE = 1                                                         SLAM     223
      SGNFTR = 1.                                                       SLAM     224
      ISGNF = IFIX(SGNFTR)                                              SLAM     225
  290 CONTINUE                                                          SLAM     226
      IF (ICASE .EQ. 1)  GO TO 300                                      SLAM     227
      DO 299  NNSLM=1,2                                                 SLAM     228
      IF (SMFRC(1,NNSLM) .EQ. 0.)  GO TO 299                            SLAM     229
      OLDKFW = SMFRC(6,NNSLM)                                           SLAM     230
      IF (YR(OLDKFW) .LE. SMFRC(5,NNSLM) )  GO TO 802                   SLAM     231
      TIMFACT = (SMFRC(5,NNSLM) - YR(OLDKFW)) /                         SLAM     232
     1          (SMFRC(5,NNSLM) - SMFRC(4,NNSLM))                       SLAM     233
      SMFRC(2,NNSLM) = SMFRC(2,NNSLM) * TIMFACT                         SLAM     234
      SMFRC(3,NNSLM) = SMFRC(3,NNSLM) * TIMFACT                         SLAM     235
      SMFRC(4,NNSLM) = YR(OLDKFW)                                       SLAM     236
  299 CONTINUE                                                          SLAM     237
      GO TO 300                                                         SLAM     238
  802 SMFRC(1,NNSLM) = 0.                                               SLAM     239
      IF (NNSLM .EQ. 1)  GO TO 299                                      SLAM     240
      GO TO 300                                                         SLAM     241
  800 NSLM = 2                                                          SLAM     242
      ICASE = 2                                                         SLAM     243
      SGNFTR = -1.                                                      SLAM     244
      ISGNF = IFIX(SGNFTR)                                              SLAM     245
      GO TO 801                                                         SLAM     246
  300 CALL SSMOOTH (SM FRC,SMPTS,I1281)                                 SLAM     247
  301 RETURN                                                            SLAM     248
C                                                                       SLAM     249
C                                                                       SLAM     250
      ENTRY INSMIN                                                      SLAM     251
C     INSMIN SETS DEFAULTS                                              SLAM     252
      RETURN                                                            SLAM     253
C                                                                       SLAM     254
C                                                                       SLAM     255
      ENTRY INSMDT                                                      SLAM     256
C     INSMDT "READS" CARDS                                              SLAM     257
      READ (5,1000) DRAFT,DENS,BTANG,DRANG                              SLAM     258
 1000 FORMAT (4F10.0)                                                   SLAM     259
      READ (5,1010)((COEF(I,1),I=1,8),(COEF(I,2),I=1,8),                SLAM     260
     1(COEF(I,3),I=1,8),(WIDTH(I),I=1,10))                              SLAM     261
 1010 FORMAT (8F10.0)                                                   SLAM     262
      READ (5,1020) BSLP,I1281                                          SLAM     263
 1020 FORMAT (F10.0,I10)                                                SLAM     264
      RETURN                                                            SLAM     265
C                                                                       SLAM     266
C                                                                       SLAM     267
      ENTRY INSMFIN                                                     SLAM     268
C     INSMFIN PRE-COMPUTES AND WRITES DATA                              SLAM     269
C                                                                       SLAM     270
      WRITE (6,1990)                                                    SLAM     271
 1990 FORMAT (1H1,10HINPUT DATA)                                        SLAM     272
      WRITE (6,2000) DRAFT,DENS,BTANG,DRANG,DX                          SLAM     273
 2000 FORMAT (1H0,7X,22HDRAFT = SHIPS DRAFT = ,E13.6,1X,2HFT/           SLAM     274
     11H ,7X,31HDENS = MASS DENSITY OF WATER = ,E13.6,1X,12HSLUGS/CU.FT. SLAM     275
     2/1H ,7X,31HBTANG = ALPHA, BUTTOCK ANGLE = ,E13.6,1X,7HDEGREES/    SLAM     276
     31H ,7X,31HDRANG = BETA, DEADRISE ANGLE = ,E13.6,1X,7HDEGREES/     SLAM     277
     41H ,7X,22HDX = STATION LENGTH = ,E13.6,1X,2HFT)                   SLAM     278
      WRITE(6,2005) BSLP                                                SLAM     279
 2005 FORMAT (1H ,7X,38HBSLP = SLOPE RELATED TO WIDTH ARRAY = ,E13.6)   SLAM     280
```

Table 12 (Continued)

```
C     CHANGE ALPHA AND BETA TO RADIANS                                  SLAM    281
      BTANG = BTANG * .01745329252                                      SLAM    282
      DRANG = DRANG * .01745329252                                      SLAM    283
C                                                                       SLAM    284
      SINAL = SIN(BTANG)                                                SLAM    285
      COSAL = COS(BTANG)                                                SLAM    286
      TANAL = TAN(BTANG)                                                SLAM    287
      COSBET = COS(DRANG)                                               SLAM    288
      TANBET = TAN(DRANG)                                               SLAM    289
      DX2 = 2. * DX                                                     SLAM    290
      DX10 = 10. * DX                                                   SLAM    291
      DX105 = 10.5 * DX                                                 SLAM    292
      DX20 = 20. * DX                                                   SLAM    293
      DNSBY2 = DENS/2.                                                  SLAM    294
      DO 1 K=1,10                                                       SLAM    295
    1 DRHGT(K) = WIDTH(K) * TANBET                                      SLAM    296
      WRITE (6,2010)                                                    SLAM    297
 2010 FORMAT (1H-,17HPRE-COMPUTED DATA)                                 SLAM    298
      WRITE (6,2020) SINAL,COSAL,TANAL,COSBET,TANBET                    SLAM    299
 2020 FORMAT (1H0,7X,24HSINAL = SINE OF ALPHA = ,E13.6/                 SLAM    300
     11H ,7X,26HCOSAL = COSINE OF ALPHA = ,E13.6/                       SLAM    301
     21H ,7X,27HTANAL = TANGENT OF ALPHA = ,E13.6/                      SLAM    302
     31H ,7X,26HCOSBET = COSINE OF BETA = ,E13.6/                       SLAM    303
     41H ,7X,27HTANBET = TANGENT OF BETA = ,E13.6)                      SLAM    304
      WRITE (6,2030) DX2,DX10,DNSBY2                                    SLAM    305
 2030 FORMAT (1H ,7X,28HDX2 = 2. * STATION LENGTH = ,E13.6,1X,2HFT/     SLAM    306
     11H ,7X,30HDX10 = 10. * STATION LENGTH = ,E13.6,1X,2HFT/           SLAM    307
     21H ,7X,38HDNSBY2 = MASS DENSITY OF WATER / 2. = ,E13.6,1X,        SLAM    308
     312HSLUGS/CU.FT.)                                                  SLAM    309
      WRITE (6,2040)                                                    SLAM    310
 2040 FORMAT (1H0,14X,18HHEIGHT OF DEADRISE,14X,20HWIDTH OF SLAM REGION/ SLAM    311
     11H ,18X,9HDRHGT(10),24X,9HWIDTH(10))                              SLAM    312
      WRITE (6,2050) ((DRHGT(I),WIDTH(I)),I=1,10)                       SLAM    313
 2050 FORMAT (10(1H ,14X,E13.6,1X,2HFT,17X,E13.6,1X,2HFT/))             SLAM    314
      WRITE (6,2060)                                                    SLAM    315
 2060 FORMAT (1H-,14X,9HCOEF(8,1),14X,9HCOEF(8,2),14X,9HCOEF(8,3))      SLAM    316
      WRITE (6,2070) ((COEF(I,1),COEF(I,2),COEF(I,3)),I=1,8)            SLAM    317
 2070 FORMAT (8(1H ,11X,E13.6,10X,E13.6,10X,E13.6/))                    SLAM    318
      RETURN                                                            SLAM    319
      END                                                               SLAM    320
```

Table 12 (Continued)

```
      SUBROUTINE SSMOOTH (SMFRC,SMPTS,I1281)                          SSMOOTH     2
      DIMENSION SMFRC(6,2), SSMARY(12,1281), ACOEF(1281), SMPTS(12)   SSMOOTH     3
      DIMENSION KSTATN(12)                                            SSMOOTH     4
      COMMON /IHVSS/ IHVS                                             SSMOOTH     5
      COMMON /PLTSLM/ RAWSLM,SSMPTS                                   SSMOOTH     6
      DATA PI /3.14159265/                                            SSMOOTH     7
      DATA ISTSSM /0/                                                 SSMOOTH     8
C                                                                     SSMOOTH     9
C     IF THIS IS FIRST CALL TO SSMOOTH, DO SOME INITIALIZING          SSMOOTH    10
      IF (ISTSSM .NE. 0)  GO TO 1010                                  SSMOOTH    11
      SUMCF1 = 0.                                                     SSMOOTH    12
      SUMCF2 = 0.                                                     SSMOOTH    13
      SUMCF4 = 0.                                                     SSMOOTH    14
      SUMCF8 = 0.                                                     SSMOOTH    15
      DO 1001 I=1,I1281                                               SSMOOTH    16
      ACOEF(I) = SIN(PI * ((FLOAT(I)-1.) / (FLOAT(I1281)-1) ))        SSMOOTH    17
 1001 SUMCF1 = SUMCF1 + ACOEF(I)                                      SSMOOTH    18
      SUMCF1 = 1. / SUMCF1                                            SSMOOTH    19
      DO 1021 I=1,I1281,2                                             SSMOOTH    20
 1021 SUMCF2 = SUMCF2 + ACOEF(I)                                      SSMOOTH    21
      SUMCF2 = 1. / SUMCF2                                            SSMOOTH    22
      DO 1041 I=1,I1281,4                                             SSMOOTH    23
 1041 SUMCF4 = SUMCF4 + ACOEF(I)                                      SSMOOTH    24
      SUMCF4 = 1. / SUMCF4                                            SSMOOTH    25
      DO 1081 I=1,I1281,8                                             SSMOOTH    26
 1081 SUMCF8 = SUMCF8 + ACOEF(I)                                      SSMOOTH    27
      SUMCF8 = 1. / SUMCF8                                            SSMOOTH    28
      DO 1003 I=1,12                                                  SSMOOTH    29
 1003 SMPTS(I) = 0.                                                   SSMOOTH    30
      DO 1004 I=1,12                                                  SSMOOTH    31
 1004 KSTATN(I) = 0                                                   SSMOOTH    32
      SSMPTS = 0.                                                     SSMOOTH    33
      KB = 0                                                          SSMOOTH    34
      KE = 0                                                          SSMOOTH    35
      KBB = 1                                                         SSMOOTH    36
      IHVSLST = 1                                                     SSMOOTH    37
      ISTSSM = 1                                                      SSMOOTH    38
C                                                                     SSMOOTH    39
 1010 CONTINUE                                                        SSMOOTH    40
C                                                                     SSMOOTH    41
C     DO NOT SMOOTH IF THERE HAS NOT BEEN ANY SLAMMING YET            SSMOOTH    42
      IF (SMFRC(2,1) .EQ. 0.  .AND.  SMFRC(2,2) .EQ. 0.)  GO TO 5000  SSMOOTH    43
C                                                                     SSMOOTH    44
C                                                                     SSMOOTH    45
      IF (IHVS .GE. 1)  GO TO 1020                                    SSMOOTH    46
      WRITE(6,7) IHVS                                                 SSMOOTH    47
    7 FORMAT '1H1,E13.6,*.....IHVS IS BUM.....*)                      SSMOOTH    48
      GO TO 5000                                                      SSMOOTH    49
 1020 IF (SMFRC(1,1) .NE. 0.  .AND.                                   SSMOOTH    50
     1    (SMFRC(1,1) .LT. 10.  .OR.  SMFRC(1,1) .GT. 20.))  GO TO 1025  SSMOOTH  51
      IF (SMFRC(1,2) .EQ. 0.  .OR.                                    SSMOOTH    52
     1    (SMFRC(1,2) .GE. 10.  .AND.  SMFRC(1,2) .LE. 20.))  GO TO 1030  SSMOOTH 53
 1025 WRITE (6,8) SMFRC(1,1),SMFRC(1,2)                               SSMOOTH    54
    8 FORMAT (1H1,*SMFRC(1,1)=*,E13.6,5X,*SMFRC(1,2)=*,E13.6,*.....   SSMOOTH    55
     1.....BAD*)                                                      SSMOOTH    56
      GO TO 5000                                                      SSMOOTH    57
C                                                                     SSMOOTH    58
 1030 CONTINUE                                                        SSMOOTH    59
      IF (IHVS .GE. 4)  GO TO 4                                       SSMOOTH    60
      IF (IHVS-2)  1,2,3                                              SSMOOTH    61
    1 IMANY = 8                                                       SSMOOTH    62
C     WHOLE TIME INCREMENT, IHVS=1, USE EVERY 8TH. LOCATION           SSMOOTH    63
      TK = SUMCF8                                                     SSMOOTH    64
      GO TO 1040                                                      SSMOOTH    65
C                                                                     SSMOOTH    66
    2 IMANY = 4                                                       SSMOOTH    67
C     1/2 TIME INCREMENT, IHVS=2, USE EVERY 4TH. LOCATION             SSMOOTH    68
      TK = SUMCF4                                                     SSMOOTH    69
      GO TO 1040                                                      SSMOOTH    70
C                                                                     SSMOOTH    71
    3 IMANY = 2                                                       SSMOOTH    72
```

Table 12 (Continued)

```
C     1/4 TIME INCREMENT, IHVS=3, USE EVERY OTHER LOCATION                    SSMOOTH    73
      TK = SUMCF2                                                             SSMOOTH    74
      GO TO 1040                                                              SSMOOTH    75
C                                                                             SSMOOTH    76
    4 IMANY = 1                                                               SSMOOTH    77
C     1/8 OR SMALLER TIME INCREMENT, IHVS=4 OR MORE, USE EVERY LOCATION       SSMOOTH    78
      TK = SUMCF1                                                             SSMOOTH    79
C                                                                             SSMOOTH    80
 1040 CONTINUE                                                                SSMOOTH    81
C                                                                             SSMOOTH    82
C                                                                             SSMOOTH    83
C     PUT THE SLAM FORCE INTO THE ARRAY TO BE SMOOTHED.                       SSMOOTH    84
C     REPEAT VALUE AS MANY TIMES NECESSARY FOR THE TIME INCREMENT.            SSMOOTH    85
C                                                                             SSMOOTH    86
C     A SLAM CAN OCCUR ON STATION NUMBERS 10 THROUGH 20                       SSMOOTH    87
C     IF SLAMMING STATION WAS  10  THEN ISTATN AND K ARE  1                   SSMOOTH    88
C                              11 ..................  2                       SSMOOTH    89
C                              12 ..................  3                       SSMOOTH    90
C                              13 ..................  4                       SSMOOTH    91
C                              14 ..................  5                       SSMOOTH    92
C                              15 ..................  6                       SSMOOTH    93
C                              16 ..................  7                       SSMOOTH    94
C                              17 ..................  8                       SSMOOTH    95
C                              18 ..................  9                       SSMOOTH    96
C                              19 .................. 10                       SSMOOTH    97
C                              20 .................. 11                       SSMOOTH    98
C                                 .................. 12                       SSMOOTH    99
C                                                                             SSMOOTH   100
      DO 100 ITYPE=1,2                                                        SSMOOTH   101
      ISTATN = IFIX (SMFRC(1,ITYPE) - 9.)                                     SSMOOTH   102
      IF (SMFRC(1,1) .EQ. 0.)  ISTATN = -1                                    SSMOOTH   103
      IF (SMFRC(1,2) .EQ. 0.  .AND.  ITYPE .EQ. 2)  GO TO 150                 SSMOOTH   104
      IF (ITYPE .EQ. 2)  GO TO 96                                             SSMOOTH   105
      KB = KE + 1                                                             SSMOOTH   106
C     CHECK IHVS AS IT COMES IN.  IF IT IS GREATER THAN LAST TIME,            SSMOOTH   107
C     THEN THE INTERVAL HAS BEEN HALVED AND THE SSMARY MUST BE                SSMOOTH   108
C     REWRITTEN STARTING AT THE PREVIOUS PLACE (THAT IS KBB).                 SSMOOTH   109
      IF (IHVS .GT. IHVSLST)  KB = KBB                                        SSMOOTH   110
      KE = KB + IMANY - 1                                                     SSMOOTH   111
      DO 99 K=1,12                                                            SSMOOTH   112
      DO 99 III=KB,KE                                                         SSMOOTH   113
      IIIMOD = MOD(III,I1281)                                                 SSMOOTH   114
      IF (IIIMOD .EQ. 0)  IIIMOD = I1281                                      SSMOOTH   115
      IF (K .EQ. ISTATN  .OR.  K .EQ. ISTATN+1)  GO TO 90                     SSMOOTH   116
      SSMARY(K,IIIMOD) = 0.                                                   SSMOOTH   117
      GO TO 99                                                                SSMOOTH   118
   90 IF (K .EQ. ISTATN+1)  GO TO 95                                          SSMOOTH   119
      SSMARY(K,IIIMOD) = SMFRC(2,ITYPE)                                       SSMOOTH   120
      GO TO 99                                                                SSMOOTH   121
   95 SSMARY(K,IIIMOD) = SMFRC(3,ITYPE)                                       SSMOOTH   122
   99 CONTINUE                                                                SSMOOTH   123
      IF (ISTATN .LT. 1)  GO TO 100                                           SSMOOTH   124
      KSTATN (ISTATN  ) = I1281                                               SSMOOTH   125
      KSTATN (ISTATN+1) = I1281                                               SSMOOTH   126
  100 CONTINUE                                                                SSMOOTH   127
C                                                                             SSMOOTH   128
   96 ISTATN = IFIX (SMFRC(1,2) - 9.)                                         SSMOOTH   129
      DO 98 III=KB,KE                                                         SSMOOTH   130
      IIIMOD = MOD(III,I1281)                                                 SSMOOTH   131
      IF (IIIMOD .EQ. 0)  IIIMOD = I1281                                      SSMOOTH   132
      SSMARY(ISTATN  ,IIIMOD) = SSMARY(ISTATN  ,IIIMOD) + SMFRC(2,2)          SSMOOTH   133
   98 SSMARY(ISTATN+1,IIIMOD) = SSMARY(ISTATN+1,IIIMOD) + SMFRC(3,2)          SSMOOTH   134
      KSTATN (ISTATN  ) = I1281                                               SSMOOTH   135
      KSTATN (ISTATN+1) = I1281                                               SSMOOTH   136
  150 CONTINUE                                                                SSMOOTH   137
      KBB = KB                                                                SSMOOTH   138
C                                                                             SSMOOTH   139
C                                                                             SSMOOTH   140
C     INITIALIZE THE SMOOTHED POINTS.                                         SSMOOTH   141
      DO 310 I=1,12                                                           SSMOOTH   142
  310 SMPTS(I) = 0.                                                           SSMOOTH   143
```

Table 12 (Continued)

```
C                                                                       SSMOOTH   144
C     DETERMINE THE SMOOTHED POINTS BY 'FLIP-FLOP' MULTIPLICATION       SSMOOTH   145
      KBIC = MAX0(1,KE-(I1281-1))                                       SSMOOTH   146
      DO 301 K=1,12                                                     SSMOOTH   147
      IF (KSTATN(K) .LE. 0)  GO TO 301                                  SSMOOTH   148
      DO 300 IC=KBIC,KE,IMANY                                           SSMOOTH   149
      ICMOD = MOD(IC,I1281)                                             SSMOOTH   150
      IF (ICMOD .EQ. 0)  ICMOD = I1281                                  SSMOOTH   151
      IB = KE - IC + 1                                                  SSMOOTH   152
      SMPTS(K) = SMPTS(K) + SSMARY(K,IB) * ACOEF(ICMOD)                 SSMOOTH   153
  300 CONTINUE                                                          SSMOOTH   154
      SMPTS(K) = SMPTS(K) * TK                                          SSMOOTH   155
      KSTATN(K) = KSTATN(K) - IMANY                                     SSMOOTH   156
  301 CONTINUE                                                          SSMOOTH   157
C                                                                       SSMOOTH   158
C     SUM ALL THE SMOOTHED POINTS FOR PLOTTING ??????????????????????? SSMOOTH   159
      SSMPTS = 0.                                                       SSMOOTH   160
      DO 350 K=1,11,2                                                   SSMOOTH   161
  350 SSMPTS = SSMPTS + SMPTS(K) + SMPTS(K+1)                           SSMOOTH   162
C                                                                       SSMOOTH   163
      IMVSLST = IHVS                                                    SSMOOTH   164
 5000 RETURN                                                            SSMOOTH   165
      END                                                               SSMOOTH   166
```

Table 12 (Continued)

```
      SUBROUTINE SEA GEN(YW,YWDOT,YWDD,CEL,T)                                  DSTWT     2
      DIMENSION YW(9),YWDOT(9),YWDD(9),ACOSX(3,9),ASINX(3,9),                  DSTWT     3
     1          COSOMT(3),SINOMT(3),AMP(3),CK(3),TAU(3),XK(9),                 DSTWT     4
     2          OMEGA(3),OMG SQ(3)                                             DSTWT     5
      COMMON/FBLOK/FIELDA(6),IF(6)                                             DSTWT     6
      COMMON/INPT2/J,ISFIELD,BUFFER(60),ICFIELD                                DSTWT     7
      COMMON /HBLOK/ KAG(21),RINT(21),EI(20),SHPMS(20),DAMPC(20),              CD1       2
     1 DXI,NEVAL,SPEED,BUOY SPG(9),SCF(9),ADDMS(9),VRTMS(20),IBEG,IEND         CD1       3
      DATA T1,              T2,              T3              /                 DSTWT     9
     1     -28.409453%2,   -25.51319346,    17.02731343     /,                DSTWT    10
     2     DIFF,           AMP                              /                 DSTWT    11
     3     .2627819036,    3.270009343,     2*5.5261579     /                 DSTWT    12
      DO 130 I = 1,3                                                           DSTWT    13
      COSOMT(I) = COS(OMEGA(I)*T)                                              DSTWT    14
  130 SINOMT(I) = SIN(OMEGA(I)*T)                                              DSTWT    15
      DO 160 K = 1,9                                                           DSTWT    16
      TXK = T + XK(K)                                                          DSTWT    17
      IF (TXK.GT.T1) GO TO 120                                                 DSTWT    18
      YW(K) = 0.                                                               DSTWT    19
      YWDOT(K) = 0.                                                            DSTWT    20
      YWDD(K) = 0.                                                             DSTWT    21
      GO TO 160                                                                DSTWT    22
  120 IF (TXK.GT.T3) GO TO 140                                                 DSTWT    23
      YW(K) = ASINX(1,K) * COSOMT(1) + ACOSX(1,K) * SINOMT(1) + DIFF           DSTWT    24
      YWDOT(K)=OMEGA(1)*(ACOSX(1,K)*COSOMT(1)-ASINX(1,K)*SINOMT(1))            DSTWT    25
      YWDD(K)= -OMG SQ(1)*YW(K)                                                DSTWT    26
      IF (TXK.GT.T2) GO TO 160                                                 DSTWT    27
      YW(K) = C1 * (C2+YW(K))                                                  DSTWT    28
      YWDOT(K) = C1 * YWDOT(K)                                                 DSTWT    29
      YWDD(K) = C1 * YWDD(K)                                                   DSTWT    30
      GO TO 160                                                                DSTWT    31
  140 YW(K) = 0.                                                               DSTWT    32
      YWDOT(K) = 0.                                                            DSTWT    33
      YWDD(K) = 0.                                                             DSTWT    34
      DO 150 I = 2,3                                                           DSTWT    35
      TEMP=ACOSX(I,K)*COSOMT(I)-ASINX(I,K)*SINOMT(I)                           DSTWT    36
      YW(K) = YW(K) + TEMP                                                     DSTWT    37
      YWDOT(K) = YWDOT(K)-OMEGA(I)*(ASINX(I,K)*COSOMT(I)                       DSTWT    38
     1                             +ACOSX(I,K)*SINOMT(I))                    DSTWT    39
  150 YWDD(K) = YWDD(K) - OMG SQ(I)*TEMP                                       DSTWT    40
  160 CONTINUE                                                                 DSTWT    41
      CEL = CELT                                                               DSTWT    42
      RETURN                                                                   DSTWT    43
      ENTRY INSGFIN                                                            DSTWT    44
      DO 210 I = 1,3                                                           DSTWT    45
      OMG SQ(I) = OMEGA(I)**2.                                                 DSTWT    46
  210 CK(I) = .011*OMEGA(I)                                                    DSTWT    47
      CELT = (1 - .011*SPEED) / .011                                           DSTWT    48
      TEMP = 2.5                                                               DSTWT    49
      DO 250 K = 1,9                                                           DSTWT    50
      IF(K .EQ. 1)              GO TO 230                                      DSTWT    51
      TEMP = 1. + 2.*FLOAT(K)                                                  DSTWT    52
  230 XK(K) = DXI*TEMP                                                         DSTWT    53
      DO 240 I = 1,3                                                           DSTWT    54
      TEMP = CK(I)*XK(K) + TAU(I)                                              DSTWT    55
      ACOSX(I,K) = AMP(I)*COS(TEMP)                                            DSTWT    56
  240 ASINX(I,K) = AMP(I)*SIN(TEMP)                                            DSTWT    57
*                                                                              DSTWT    58
*           FOR LATER, XK(K) (=K*XI) IS USED TO SELECT W1 OR W2.               DSTWT    59
      XK(K) = .011*XK(K)                                                       DSTWT    60
  250 CONTINUE                                                                 DSTWT    61
      C1 = (AMP(1)+DIFF)/(2.*AMP(1))                                           DSTWT    62
      C2 = AMP(1) - DIFF                                                       DSTWT    63
      WRITE (6,10)                                                             DSTWT    64
   10 FORMAT (1H-,*SEAGEN IS USING DISCRETE WAVE TRAIN*)                       DSTWT    65
      RETURN                                                                   DSTWT    66
      ENTRY INSEGIN                                                            DSTWT    67
      RETURN                                                                   DSTWT    68
      ENTRY INSEGDT                                                            DSTWT    69
      CALL FIELDS(52528,6,BUFFER)                                              DSTWT    70
      DO 310 I = 1,3                                                           DSTWT    71
      TAU(I) = FIELDA(I + 3)                                                   DSTWT    72
  310 OMEGA(I) = FIELDA(I)                                                     DSTWT    73
      RETURN                                                                   DSTWT    74
      END                                                                      DSTWT    75
```

Table 12 (Continued)

```
      SUBROUTINE SEAGEN (YW,YWDOT,YWDD,CEL,T)                                SINWV     2
C                                                                            SINWV     3
C     CEL = CELERITY, WAVE SPEED                                             SINWV     4
C     DELTA = (GRAVITY * PI)/(CEL + SPEED) = TIME(SUB 1) - TIME(SUB 0)       SINWV     5
C     X(SUB 1,...,9) = ARRAY OF THE 9 STATIONS OF THE SHIP                   SINWV     6
C     XK(I) = K * X(I)                                                       SINWV     7
C     XX(1,...,9) IS TIME(SUB 0) FOR 9 STATIONS OF THE SHIP                  SINWV     8
C     OMEGAW = FREQUENCY OF THE WAVE                                         SINWV     9
C     OMGWSQ = OMEGAW**2                                                     SINWV    10
C     OMEGAE = FREQUENCY OF ENCOUNTER                                        SINWV    11
C     OMGESQ = OMEGAE**2                                                     SINWV    12
C     SINOMT = SIN(OMEGAE*TIME) = SIN OF THE ANGLE WITH TIME                 SINWV    13
C     COSOMT = COS(OMEGAE*TIME) = COSINE OF THE ANGLE WITH TIME              SINWV    14
C     ASINX = WAVE HEIGHT * SIN(K * X(SUB I))                                SINWV    15
C     ACOSX = WAVE HEIGHT * COS(K * X(SUB I))                                SINWV    16
C     SPEED = U                                                              SINWV    17
C     TIME = T                                                               SINWV    18
C     GRAV = GRAVITY = 32.23                                                 SINWV    19
C     YW = 1/OMEGAW                                                          SINWV    20
C     YWDOT = FIRST DERIVATIVE OF 1/OMEGA = VELOCITY OF WAVE HEIGHT          SINWV    21
C                                        AT SPECIFIC TIME + LOCATION         SINWV    22
C     YWDD = SECOND DERIVATIVE OF 1/OMEGA = ACCELERATION OF WAVE HEIGHT      SINWV    23
C                                        AT SPECIFIC TIME + LOCATION         SINWV    24
C     WVLGT = LAMBDA = WAVE LENGTH                                           SINWV    25
C     PI = 3.14159265                                                        SINWV    26
C     WVHGT = W = WAVE AMPLITUDE, WAVE HEIGHT                                SINWV    27
C                                                                            SINWV    28
      COMMON/CELER/CELT                                                      SINWV    29
      COMMON/FBLOK/FA(6),IF(6)                                               SINWV    30
      COMMON/INPT2/J,ISFIELD,BUFFER(60),ICFIELD                              SINWV    31
      COMMON /HBLOK/ KAG(21),RINT(21),EI(20),SHPMS(20),DAMPC(20),            CD1      2
     1 DXI,NEVAL,SPEED,BUOY SPG(9),SCF(9),ADDMS(9),VRTMS(20),IBEG,IEND       CD1      3
      DIMENSION X(9),XK(9),ASINX(9),ACOSX(9),YW(9),YWDOT(9),YWDD(9)          SINWV    33
      REAL K                                                                 SINWV    34
      DATA GRAV/32.23/,PI/3.14159265/                                        SINWV    35
      DATA X/2.5,5.,7.,9.,11.,13.,15.,17.,19./                               SINWV    36
C                                                                            SINWV    37
      CEL = CELT                                                             SINWV    38
      OMGTIM = OMEGAE * T                                                    SINWV    39
      SINOMT = SIN(OMGTIM)                                                   SINWV    40
      COSOMT = COS(OMGTIM)                                                   SINWV    41
      DO 400 I=1,9                                                           SINWV    42
      IF (WVHGT.EQ.0.) GO TO 210                                             SINWV    43
      IF (OMGTIM.GT.XK(I)) GO TO 230                                         SINWV    44
  210 YW(I) = 0.                                                             SINWV    45
      YWDOT(I) = 0.                                                          SINWV    46
      YWDD(I) = 0.                                                           SINWV    47
      GO TO 400                                                              SINWV    48
  230 YW(I) = ASINX(I) * COSOMT + ACOSX(I) * SINOMT                          SINWV    49
      YWDOT(I) = OMEGAE * (ACOSX(I) * COSOMT - ASINX(I) * SINOMT)            SINWV    50
      YWDD(I) = -(OMGESQ * YW(I))                                            SINWV    51
      IF (OMGTIM .GT. XK(I) + PI) GO TO 400                                  SINWV    52
      YW(I) = .5 * (YW(I) - WVHGT)                                           SINWV    53
      YWDOT(I) = .5 * YWDOT(I)                                               SINWV    54
      YWDD(I) = .5 * YWDD(I)                                                 SINWV    55
  400 CONTINUE                                                               SINWV    56
      RETURN                                                                 SINWV    57
C                                                                            SINWV    58
      ENTRY INSEGIN                                                          SINWV    59
C     INSEGIN SETS DEFAULTS                                                  SINWV    60
      OMEGAW = 0.                                                            SINWV    61
      WVHGT = 0.                                                             SINWV    62
      RETURN                                                                 SINWV    63
C                                                                            SINWV    64
      ENTRY INSEGOT                                                          SINWV    65
C     INSEGOT "READS" CARDS THAT ARE INPUT                                   SINWV    66
      CALL FIELDS (50008,6,BUFFER)                                           SINWV    67
      OMEGAW = FA(1)                                                         SINWV    68
      WVHGT = FA(2)                                                          SINWV    69
      RETURN                                                                 SINWV    70
C                                                                            SINWV    71
```

Table 12 (Continued)

```
      ENTRY INSGFIN                                                    SINWV     72
C     INSGFIN SETS UP DATA + WRITES OUT INPUT DATA                     SINWV     73
      OMGWSQ = OMEGAW ** 2                                             SINWV     74
      OMEGAE = OMEGAW + (SPEED * OMGWSQ / GRAV)                        SINWV     75
      OMGESQ = OMEGAE**2                                               SINWV     76
      WVLGT = 2. * PI * GRAV / OMGWSQ                                  SINWV     77
      K = OMGWSQ / GRAV                                                SINWV     78
      CELT = GRAV / OMEGAW                                             SINWV     79
      DELTA = GRAV * PI / (CELT + SPEED)                               SINWV     80
      PIBY2 = PI/2.                                                    SINWV     81
      DO 20 I=1,9                                                      SINWV     82
      XK(I) = K * DXI * FLOAT(I) * 2.                                  SINWV     83
      ASINX(I) = WVHGT * SIN(XK(I))                                    SINWV     84
      ACOSX(I) = WVHGT * COS(XK(I))                                    SINWV     85
   20 XK(I) = -(XK(I) - PIBY2)                                         SINWV     86
  250 WRITE (6,1000) OMEGAW,WVHGT,CELT,OMEGAE,WVLGT                    SINWV     87
 1000 FORMAT (1H1,17HWAVE FREQUENCY = ,F7.3/                           SINWV     88
     1            15H WAVE HEIGHT = ,F7.3/                            SINWV     89
     2            12H CELERITY = ,F7.3/                               SINWV     90
     3            26H FREQUENCY OF ENCOUNTER = ,F7.3/                 SINWV     91
     4            15H WAVE LENGTH = ,F7.3)                            SINWV     92
      RETURN                                                           SINWV     93
C                                                                      SINWV     94
      END                                                              SINWV     95
```

Table 12 (Continued)

```
      SUBROUTINE SEA GEN (HF,YWDOT,YWDD,CEL,T)                          STATWV     2
C$    DEBUG                                                             STATWV     3
C$    CALLS                                                             STATWV     4
C$    STORES(FREQ,WVHGH)                                                STATWV     5
C                                                                       STATWV     6
C**SEA GENERATOR                                                        STATWV     7
C                                                                       STATWV     8
      DIMENSION HF(1),YWDOT(1),YWDD(1),CS1(10)                          STATWV     9
      COMMON/FBLOK/FIELDA(6) ,IF(6)                                     STATWV    10
      COMMON /INPT2/J,ISFIELD,BUFFER(60),ICFIELD                        STATWV    11
      DATA CS1/.000,.174,.342,.5,.643,.766,.866,.94,.985,1./            STATWV    12
      CEL=CELT                                                          STATWV    13
      DO 5 I=1,20                                                       STATWV    14
 5    HF(I)=0.                                                          STATWV    15
      TEMP=WVHGH*SIN(OMEGA*T)                                           STATWV    16
      DO 10 I=1,5                                                       STATWV    17
 10   HF(I+5)=HF(16-I)=CS1(2*I)*TEMP                                    STATWV    18
      RETURN                                                            STATWV    19
C                                                                       STATWV    20
C**SET DEFAULTS                                                         STATWV    21
C                                                                       STATWV    22
      ENTRY INSEGIN                                                     STATWV    23
      FREQ=WVHGH=0.                                                     STATWV    24
      RETURN                                                            STATWV    25
C                                                                       STATWV    26
C** ACCEPT DATA                                                         STATWV    27
C                                                                       STATWV    28
      ENTRY INSEGOT                                                     STATWV    29
      CALL FIELDS(50003,6,BUFFER)                                       STATWV    30
      FREQ=FIELDA(1)                                                    STATWV    31
      WVHGH=FIELDA(2)                                                   STATWV    32
      RETURN                                                            STATWV    33
C                                                                       STATWV    34
C**DATA ECHO                                                            STATWV    35
C                                                                       STATWV    36
      ENTRY INSGFIN                                                     STATWV    37
      OMEGA=2.*3.14159265358 98*FREQ                                    STATWV    38
      CELT=32.23/OMEGA                                                  STATWV    39
      OMEGA2=OMEGA**2                                                   STATWV    40
      WRITE(6,90) FREQ,WV HGH                                           STATWV    41
 90   FORMAT(1H-,*SEA DATA (CASE 1)*//T5,*FREQ = *,E20.10,3X,*HZ*       STATWV    42
     1 /T5,*WAVE HEIGHT = *,E20.10,3X,*FT*)                             STATWV    43
      RETURN                                                            STATWV    44
      END                                                               STATWV    45
```

AD-A039 510 DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/G 13/10
STRUCTURAL SEAWORTHINESS DIGITAL COMPUTER PROGRAM ROSAS (A CONV--ETC(U)
MAY 77 S CHUANG, E A SCHROEDER, S WYBRANIEC

UNCLASSIFIED DTNSRDC-77-0001 NL

2 OF 2
AD A039510



END

DATE FILMED
6-77

Table 12 (Continued)

```
      SUBROUTINE DAUX (T, S,D S)                                          DAUX      2
      REAL KAG,MASS                                                       DAUX      3
      DIMENSION KAG(21),RINT(21),EI(20),SHP MS(20),DAMPC(20)              DAUX      4
     1 ,SCF(9),BUOY SPG(9),ADD MS(9),VRT MS(20)                           DAUX      5
     2          ,YS( 91),SF(21),GDOT(21),BDM(20),YDOT(20),YH( 9)         DAUX      6
     3          ,DYS( 91),DSF(21),DGDOT(21),DBDM(20),DYDOT(20),DYH( 9)   DAUX      7
     4          ,MF(20),S( 91),DS(91 )                                   DAUX      8
      COMMON /HBLOK/ KAG,RINT,EI,SHPMS,DAMPC,DXI,NEVAL,SPEED,             DAUX      9
     1 BUOY SPG,SCF,ADDMS,VRTMS,IBEG,IEND                                 DAUX     10
      COMMON / TEST / IND                                                 DAUX     11
      EQUIVALENCE                                                         DAUX     12
     1  (YS,YH),(YS(10),YDOT),(YS(30),GDOT),(YS(51),BDM),(YS(71),SF)      DAUX     13
     2 ,(DYS,DYH),(DYS(10),DYDOT),(DYS(30),DGDOT),(DYS(51),DBDM)          DAUX     14
     3 ,(DYS(71),DSF)                                                     DAUX     15
      DO  20 K = 1,91                                                     DAUX     16
   20 YS(K) = S(K)                                                        DAUX     17
      NEVAL = NEVAL + 1                                                   DAUX     18
      CALL HYD FRC (T,YH,YDOT,MF)                                         DAUX     19
      DO 100 I = 1,20                                                     DAUX     20
      DBDM(I) = EI(I)*(GDOT(I+1) - GDOT(I))/DXI                           DAUX     21
      DYDOT(I) = (MF(I)-DAMPC(I)*YDOT(I)-(SF(I+1)-SF(I))/DXI)/VRT MS(I)   DAUX     22
      IF (I .EQ. 1)  GO TO 100                                            DAUX     23
      DSF(I) = (GDOT(I) - (YDOT(I) - YDOT(I-1))/DXI)*KAG(I)               DAUX     24
      DGDOT(I) = ((BDM(I) - BDM(I-1))/DXI - SF(I))/RINT(I)                DAUX     25
  100 CONTINUE                                                            DAUX     26
      DSF(1)  = 0.                                                        DAUX     27
      DSF(21) = 0.                                                        DAUX     28
      DGDOT( 1) = DGDOT(2)                                                DAUX     29
      DGDOT(21) = DGDOT(20)                                               DAUX     30
      DO 110 I = 1,9                                                      DAUX     31
      K = 2*I+2                                                           DAUX     32
  110 DYH(I) = .5*(YDOT(K-1) + YDOT(K))                                   DAUX     33
      DO 120 K = 1,91                                                     DAUX     34
  120 DS(K) = DYS(K)                                                      DAUX     35
      IF (IND .EQ. 0)                    GO TO  6                         DAUX     36
      WRITE(6,4) NEVAL                                                    DAUX     37
    4 FORMAT(1X//1X,30(1H*)/1X,*NEVAL = *,I10//1X,*YH, YDOT, BDM, GDOT,   DAUX     38
     1 SF*)                                                               DAUX     39
      WRITE (6,5)  YH, YDOT, BDM, GDOT, SF                                DAUX     40
      WRITE (6,8)                                                         DAUX     41
    8 FORMAT(1X,*DYH, DYDOT, DBDM, DGDOT, DSF*)                           DAUX     42
      WRITE (6,5) DYH,DYDOT,DBDM,DGDOT,DSF                                DAUX     43
    5 FORMAT(/9E13.6//2(10E13.6/10E13.6//),2(10E13.6/10E13.6/E13.6//))    DAUX     44
    6 CONTINUE                                                            DAUX     45
      RETURN                                                              DAUX     46
      END                                                                 DAUX     47
```

Table 12 (Continued)

```
      SUBROUTINE KUTMER(ND,T,H,YO,EPSE,A,HCX,FIRST)                    KTMR      2
      DIMENSION YO(110),Y1(110),Y2(110),F0(110),F1(110),F2(110)        KTMR      3
      DIMENSION EPSE(110),A(110),NEQN(24)                              KTMR      4
C                                                                      KTMR      5
C     PASS TO SSMOOTH CONDITION OF TIME INTERVAL - WHOLE TO HALVES     KTMR      6
      COMMON /IHVSS/ IHVS                                              KTMR      7
C                                                                      KTMR      8
      DATA   HCX2 / 1.E200 /                                           KTMR      9
C                                                                      KTMR     10
C     ND= NUMBER OF EQUATIONS, NO. OF COMPONENTS OF YO                 KTMR     11
C     T= INDEPENDENT VARIABLE                                          KTMR     12
C     H=INCREMENT FOR WHICH SOLN IS TO BE RETURNED  + OR-              KTMR     13
C     YO IS THE VECTOR OF DEPENDENT VARIABLES.  ENTER WITH INITIAL     KTMR     14
C        VALUES AT T AND RETURNS WITH VALUES AT T+H                    KTMR     15
C     EPSE=RELATIVE ERROR CRITERION FOR COMPONENTS OF YO .GT. ABS(A)   KTMR     16
C     A=ABSOLUTE ERROR CRITERIA FOR COMPONENTS OF YO .LT. ABS(A)       KTMR     17
C NOTE-- EPSE AND A MUST BE SPECIFIED FOR EACH COMPONENT OF THE SYSTEM KTMR     18
C     HCX = THE SMALLEST STEP SIZE USED IN THE INTEGRATION             KTMR     19
C     DAUX(T,Y,F) RETURNS THE VECTOR F(T,Y)                            KTMR     20
C     HCX MAY SPECIFY A BEGINNING STEP SIZE = H / 2**K                 KTMR     21
C     ON RETURN HCX CONTAINS THE SMALLEST STEP SIZE USED               KTMR     22
C     SET FIRST = 0.  FOR THE FIRST STEP OR FOR A NEW VALUE OF H       KTMR     23
C                 0.5 FOR FIRST STEP OR NEW H, AND HCX SPECIFIED       KTMR     24
C                 1.0 TO CONTINUE INTEGRATION USING THE SAME H         KTMR     25
C                 1.5 TO CONTINUE WITH THE SAME H WHILE SPECIFYING HCX KTMR     26
C     FIRST = 2. ON RETURN, IF THE ERROR CRITERION COULD NOT           KTMR     27
C                BE MET BY REDUCING THE STEP SIZE BY 1/128             KTMR     28
C                                                                      KTMR     29
      IF (ABS(FIRST).LT.1.) IDBL=0                                     KTMR     30
      IFIRST = IFIX(2.*ABS(FIRST))+1                                   KTMR     31
                                        GO TO (15,10,20,5), IFIRST     KTMR     32
    5 IF (HC .EQ. 0.)                   GO TO 10                       KTMR     33
      HCX = AMIN1 (HC,HCX)                                             KTMR     34
   10 HC  = AMIN1 (H,HCX)                                              KTMR     35
      IHVS = 1                                                         KTMR     36
      MNSTPS = IFIX(H/HC + .00001)                                     KTMR     37
      IPLOC = MNSTPS                                                   KTMR     38
      FIRST = SIGN(1.5,FIRST)                                          KTMR     39
                                        GO TO 25                       KTMR     40
   15 HC = H                                                           KTMR     41
      IHVS = 1                                                         KTMR     42
      IPLOC=1                                                          KTMR     43
      FIRST = SIGN(1.,FIRST)                                           KTMR     44
   20 MNSTPS = 1                                                       KTMR     45
   25 LOC=0                                                            KTMR     46
      INEQN = 0                                                        KTMR     47
      IF (HC .EQ. 0.)                   GO TO 240                      KTMR     48
        HCX=HC                                                         KTMR     49
   26 IF (2.*HC .EQ. HCX2)              GO TO 30                       KTMR     50
   27 HC2 = HC/2.                                                      KTMR     51
      HC3 = HC/3.                                                      KTMR     52
      HC6 = HC/6.                                                      KTMR     53
      HC8 = HC/8.                                                      KTMR     54
      HC32 = 3.*HC2                                                    KTMR     55
      HC23 = 2.*HC3                                                    KTMR     56
      HC38 = 3.*HC8                                                    KTMR     57
      HCX2 = 2.*HC                                                     KTMR     58
   30 CALL DAUX(T,YO,F0)                                               KTMR     59
      DO 40 I=1,ND                                                     KTMR     60
   40 Y1(I)  =YO(I)  +    HC3*F0(I)                                    KTMR     61
      CALL DAUX(T+HC3  , Y1,F1)                                        KTMR     62
      DO 50 I=1,ND                                                     KTMR     63
   50 Y1(I)  =YO(I)  + HC6*(F0(I) + F1(I) )                            KTMR     64
      CALL DAUX(T+HC3  , Y1,F1)                                        KTMR     65
      DO 60 I=1,ND                                                     KTMR     66
   60 Y1(I)  =YO(I)  +  HC8*F0(I)+   HC38*F1(I)                        KTMR     67
      CALL DAUX(T+HC2  , Y1,F2)                                        KTMR     68
      DO 70 I=1,ND                                                     KTMR     69
   70 Y1(I)  =YO(I  )+  HC2*F0(I)-  HC32*F1(I)+ HCX2*F2(I)             KTMR     70
      CALL DAUX( T+HC,Y1,F1)                                           KTMR     71
      DO 80 I=1,ND                                                     KTMR     72
```

Table 12 (Continued)

```
   80 Y2(I)=Y0(I) + HC6*(F0(I) + F1(I) ) + HC23*F2(I)                    KTMR    73
      INC=0                                                             KTMR    74
      DO 110 I=1,ND                                                     KTMR    75
      ZZZ=ABS(Y2(I))-A(I)                                               KTMR    76
      IF (ZZZ .GE. 0.)                  GO TO 87                        KTMR    77
      ERROR=ABS(.2*(Y1(I)  -Y2(I)))                                     KTMR    78
      IF(ERROR - A(I)*EPSE(I))          100,100,90                      KTMR    79
   87 ERROR=ABS(.2-.2*Y1(I)/Y2(I)  )                                    KTMR    80
      IF (ERROR .LE. EPSE(I))           GO TO 100                       KTMR    81
   90 INEQN = INEQN + 1                                                 KTMR    82
      IF(INEQN .LE. 24)       NEQN(INEQN) = I                           KTMR    83
      IF (IPLOC .LT. 128*MNSTPS)         GO TO 95                       KTMR    84
      WRITE(6,92)I,T,ERROR                                              KTMR    85
   92 FORMAT( 18H FOR EQUATION NO. I2,27H, THE RELATIVE ERROR AT T = ,   KTMR    86
     1      E12.3, 3HIS  , F12.6 )                                      KTMR    87
        FIRST = 2.                                                      KTMR    88
                                     GO TO 327                          KTMR    89
   95 HC=HC/2.                                                          KTMR    90
      IPLOC=2 *IPLOC                                                    KTMR    91
      LOC=2 *LOC                                                        KTMR    92
      IHVS = IHVS + 1                                                   KTMR    93
      IF (IDBL .EQ. 1)        WRITE (6,99) I,T,ERROR                    KTMR    94
   99 FORMAT (70H KUTMER DOUBLED INTERVAL AND IMMEDIATELY HALVED IT DUE KTMR    95
     1TO EQUATION NO. ,I4,7H AT T = 1PE15.6 / 22H THE RELATIVE ERROR IS KTMR    96
     2 1PE15.6)                                                         KTMR    97
        HCX=HC                                                          KTMR    98
                                     GO TO 27                           KTMR    99
C     IF ERROR IS LESS THAN 1/100 EPSILON, TRY TO DOUBLE INTERVAL       KTMR   100
  100 IF (ERROR*100. .LE. EPSE(I))       GO TO 110                      KTMR   101
      INC=1                                                             KTMR   102
  110 CONTINUE                                                          KTMR   103
      IDBL = 0                                                          KTMR   104
      T=T+HC                                                            KTMR   105
      DO 112 I=1,ND                                                     KTMR   106
  112 Y0(I)  =Y2(I)                                                     KTMR   107
      LOC=LOC+1                                                         KTMR   108
C     IF ANY OF THESE TESTS IS PASSED, THE INTERVAL WILL NOT BE DOUBLED KTMR   109
      IF (LOC .GE. IPLOC)               GO TO 210                       KTMR   110
      IF(INC .NE. 0)                    GO TO 210                       KTMR   111
      IF(MOD(LOC,2) .NE. 0)             GO TO 210                       KTMR   112
      IF(IPLOC .LE. MNSTPS)             GO TO 210                       KTMR   113
      HC=2.*HC                                                          KTMR   114
      LOC=LOC/2                                                         KTMR   115
      IPLOC=IPLOC/2                                                     KTMR   116
      IHVS = IHVS - 1                                                   KTMR   117
      IDBL = 1                                                          KTMR   118
  210 IF(IPLOC-LOC)                     26,327,26                       KTMR   119
  240 IF( H.NE.0.)                      GO TO 15                        KTMR   120
      WRITE( 6,241)                                                     KTMR   121
  241 FORMAT(5X,47H KUTMER ENTERED WITH ZERO INTEGRATION INTERVAL )     KTMR   122
  327 IF (INEQN.EQ.0.OR.FIRST.GE.0.) RETURN                             KTMR   123
      IF(INEQN .LE. 24)                 GO TO 329                       KTMR   124
      WRITE (6,328) INEQN                                               KTMR   125
  328 FORMAT (90X,15HINTERVAL HALVED,I3,6H TIMES)                       KTMR   126
      INEQN = 24                                                        KTMR   127
  329 WRITE (6,330) (NEQN(I),I = 1,INEQN)                               KTMR   128
  330 FORMAT ( 90X,32HINTEGRATION INTERVAL WAS HALVED                   KTMR   129
     1        /90X,32HDUE TO ERROR IN VARIABLE NUMBER  / (89X,8I4))     KTMR   130
      RETURN                                                            KTMR   131
      END                                                               KTMR   132
```

Table 12 (Continued)

```
      SUBROUTINE PLOTIN                                                  PLTN      2
C$    DEBUG                                                              PLTN      3
C$    STORES(I,TR,TL,LOC)                                                PLTN      4
C$    CALLS(TRYPIN1,TRYPIN2,PLOT,PLOTFIN)                                PLTN      5
      DIMENSION BDM(20),NAPPLY(20)                                       PLTN      6
      COMMON YS(91)                                                      PLTN      7
      COMMON/PBLOK1/NR,TTT(20),R(20),MSBSTN,NAME,CODE,PHONE              CD6       2
      COMMON/PBLOK4/NREPLT,YLU(2,4)                                      CD7       2
      DIMENSION YR(9,3)                                                  PLTN     10
      COMMON/CBLOK/NCALC,NMSTNS,NGROUP,NL(9),NLINV(9),MNLBNL2(90),       PLTN     11
     1 ITYPE(18), YCRITA(18), VCRITA(18),IBSTORE(350),IMSTORE(700),      PLTN     12
     3 KWORDS(2),NP(2),IFIRST,NAPPLY                                     PLTN     13
      COMMON /PRNT1/T,PITCH,TLEFT                                        PLTN     14
      EQUIVALENCE(YS(51),BDM)                                            PLTN     15
C-------------------------------,-------------------------------------  PLTN     16
      COMMON /PBLOK3/ PSTORE(1024,6),LOC,INC,TL,TR,TRR                   PLTN     17
      COMMON /PLTSLM/ RAWSLM,SSMPTS                                      PLTN     18
      DATA MAXLOC /1024/                                                 PLTN     19
C-------------------------------,-------------------------------------  PLTN     20
C**INITIALIZATION                                                        PLTN     21
      DATA JUMP/0/                                                       PLTN     22
      IF (NR .LE. 0) RETURN                                              PLTN     23
      IF (JUMP .NE. 0) GO TO 1                                           PLTN     24
      CALL CAMRAV(935)                                                   PLTN     25
      CALL IDFRMV(NAME,CODE,PHONE)                                       PLTN     26
      JUMP=1                                                             PLTN     27
 1    CONTINUE                                                           PLTN     28
      I=0                                                                PLTN     29
      CALL TRYPIN1                                                       PLTN     30
C**DETERMINE LEFT-HAND ENDPOINT OF TIME AXIS                             PLTN     31
      LOC=0                                                              PLTN     32
 5    I=I+1                                                              PLTN     33
      IF (I.LE. NR) GO TO 10                                             PLTN     34
      TL=1.E20                                                           PLTN     35
      RETURN                                                             PLTN     36
 10   IF (R(I) .LE. 0.) GO TO 5                                          PLTN     37
      TR=R(I)                                                            PLTN     38
      CALL TRYPIN2                                                       PLTN     39
      TL=TTT(I)                                                          PLTN     40
C**RIGHT-HAND ENDPOINT OF TIME AXIS                                      PLTN     41
 15   TEMP=TL+R(I)                                                       PLTN     42
      TR=AMIN1(TTT(I+1),TEMP)                                            PLTN     43
      TRR=(TEMP-TR)/R(I)                                                 PLTN     44
      IF(JUMP .EQ. 1) RETURN                                             PLTN     45
*                                                                        PLTN     46
C**STORE FRAME'S WORTH OF DATA THEN PLOT                                 PLTN     47
*                                                                        PLTN     48
      ENTRY PLOT                                                         PLTN     49
      IF (NR .LE. 0) RETURN                                              PLTN     50
      IF ( T .LT. TL) RETURN                                             PLTN     51
      IF (T .GT. TR) GO TO 1000                                          PLTN     52
C**STORE ABSCISSA AND ORDINATES                                          PLTN     53
      LOC=LOC+1                                                          PLTN     54
      IF (LOC .LE. MAXLOC) GO TO 500                                     PLTN     55
      INC=1                                                              PLTN     56
      LOC=MAXLOC                                                         PLTN     57
      GO TO 1020                                                         PLTN     58
  500 PSTORE(LOC,6) = T                                                  PLTN     59
      PSTORE(LOC,1)=PITCH                                                PLTN     60
      PSTORE(LOC,2)=BDM(MSBSTN)                                          PLTN     61
      CALL SEAGEN(YR(1,1),YR(1,2),YR(1,3),CEL,T)                         PLTN     62
      PSTORE(LOC,3) = YR(4,1)                                            PLTN     63
C                                                                        PLTN     64
C     PLOT THE RAW SLAM FORCE ON TOP OF PITCH GRAPH                      PLTN     65
      PSTORE(LOC,4) = RAWSLM                                             PLTN     66
C                                                                        PLTN     67
C     PLOT THE SMOOTHED SLAM ON TOP OF PITCH GRAPH                       PLTN     68
      PSTORE(LOC,5) = SSMPTS                                             PLTN     69
      RETURN                                                             PLTN     70
```

Table 12 (Continued)

```
C**PREPARE PLOTS                                                    PLTN      71
 1000 INC=0                                                         PLTN      72
      IF (LOC .EQ. 0) GO TO 2000                                    PLTN      73
 1020 CALL TRYPLOT                                                  PLTN      74
 2000 IF (JUMP .NE. 3) GO TO 2010                                   PLTN      75
      CALL PLTND(0)                                                 PLTN      76
      IF(NREPLT .EQ. 2)       ENDFILE 8                             PLTN      77
      RETURN                                                        PLTN      78
 2010 CONTINUE                                                      PLTN      79
C**RE-INITIALIZE FOR NEXT FRAME                                     PLTN      80
      LOC=0                                                         PLTN      81
      JUMP=2                                                        PLTN      82
      TL=TL+R(I)                                                    PLTN      83
      IF (TL-.99999*TTT(I+1)) 15,5,5                                PLTN      84
*                                                                   PLTN      85
C*PLOT LAST FRAME                                                   PLTN      86
*                                                                   PLTN      87
      ENTRY PLOTFIN                                                 PLTN      88
      IF (NR .LE. 0) RETURN                                         PLTN      89
      JUMP=3                                                        PLTN      90
      IF(NREPLT .EQ. 3)       GO TO 1020                            PLTN      91
      GO TO 1000                                                    PLTN      92
      END                                                           PLTN      93
```

Table 12 (Continued)

```
      SUBROUTINE TRYPIN1                                                TRPN      2
C$    DEBUG                                                             TRPN      3
C$    STORES(IP1,IP2,DEL,DELA,YYLU,XXL,XXR,MTR)                         TRPN      4
      COMMON /PBLOK3/ PSTORE(1024,6),LOC,INC,TL,TR,TRR                  TRPN      5
      COMMON/PBLOK4/NREPLT,YLU(2,4)                                     GD7       2
      DIMENSION IYC(2,3),DELA(4),II(3),YYLU(2,3),NCH(3),LABEL(5,4)      TRPN      7
      DIMENSION XINC(6), FCTNORM(6), IEXP(4)                            TRPN      8
      DATA XINC/.3,.5,.75,1.,1.5,2./                                    TRPN      9
      DATA LABEL(1,1) / 10HPITCH (DEG   /                               TRPN     10
      DATA LABEL(1,2) / 31HMIDSHIP BENDING MOMENT (FT-TONS  /           TRPN     11
      DATA LABEL(1,3) / 26HWAVE HEIGHT AT MIDSHIP (FT  /                TRPN     12
      DATA NCH/10,31,26/                                                TRPN     13
      DATA IYC/60,345,375,660,690,975/                                  TRPN     14
*                                                                       TRPN     15
C*INITIALIZATION--SET PARAMETERS FOR PLOTTING ROUTINES                  TRPN     16
*                                                                       TRPN     17
C--VERTICAL AXES                                                        TRPN     18
      IP1=1                                                             TRPN     19
      IP2=3                                                             TRPN     20
      JUMP=1                                                            TRPN     21
      FCTNORM(4) = 30.                                                  TRPN     22
      FCTNORM(5) = 15.                                                  TRPN     23
      GO TO 15                                                          TRPN     24
C--HORIZONTAL (TIME) AXIS                                               TRPN     25
      ENTRY TRYPIN2                                                     TRPN     26
      IP1=IP2=4                                                         TRPN     27
      YLU(2,4)=TR                                                       TRPN     28
      YLU(1,4) = TL                                                     TRPN     29
      JUMP=2                                                            TRPN     30
C**CALCULATE GRID LINE SPACINGS                                         TRPN     31
 15   DO 110 IP=IP1,IP2                                                 TRPN     32
      DEL=YLU(2,IP)-YLU(1,IP)                                           TRPN     33
      IF (DEL .GT. 0.) GO TO 30                                         TRPN     34
      DELA(IP)=DEL                                                      TRPN     35
      GO TO 110                                                         TRPN     36
 30   IEXP(IP) = IFIX(ALOG10(DEL))                                      TRPN     37
      FNIP = 10. ** IEXP(IP)                                            TRPN     38
      DELNORM = DEL / FNIP                                              TRPN     39
      IF (DELNORM .LT. 10) GO TO 40                                     TRPN     40
      IF=6                                                              TRPN     41
      GO TO 90                                                          TRPN     42
 40   CONTINUE                                                          TRPN     43
      DO 75 IF=1,6                                                      TRPN     44
      IEMP=DELNORM/XINC(IF)                                             TRPN     45
      IF (IEMP .EQ. 3 .OR. IEMP .EQ. 4) GO TO 90                        TRPN     46
 75   CONTINUE                                                          TRPN     47
      STOP 11                                                           TRPN     48
 90   IF (IP .NE. 4) GO TO 95                                           TRPN     49
      IB=2+MOD(IF,2)                                                    TRPN     50
      IC=-IB                                                            TRPN     51
      AIB = 2. * FNIP / FLOAT(IB)                                       TRPN     52
      GO TO 100                                                         TRPN     53
 95   AIB = FNIP                                                        TRPN     54
100   DELA(IP) = XINC(IF)*AIB                                           TRPN     55
      IF (IP .EQ. 4)  FCTNORM(6) = FNIP                                 TRPN     56
      IF (IP .NE. 4)  FCTNORM(IP) = FNIP                                TRPN     57
110   CONTINUE                                                          TRPN     58
C  JUMP = 2 IF COMPUTING TIME AXIS DIVISIONS                            TRPN     59
      IF (JUMP .EQ. 2) GO TO 150                                        TRPN     60
C**DETERMINE WHICH GRID WILL HAVE TIME AXIS LABELING                    TRPN     61
      K=5                                                               TRPN     62
      DO 125 I=1,3                                                      TRPN     63
      IEXP(I) = IFIX(ALOG10(AMAX1(YLU(2,I),YLU(1,I))))                  TRPN     64
      FCTNORM(I) = 10.**IEXP(I)                                         TRPN     65
      IF(IEXP(I).LT.0 .OR. IEXP(I).GT.2)   GO TO 120                    TRPN     66
      FCTNORM(I) = 1.                                                   TRPN     67
      IEXP(I) = 0                                                       TRPN     68
120   DELA(I) = DELA(I)/FCTNORM(I)                                      TRPN     69
      II(I) = 0                                                         TRPN     70
      IF (DELA(I) .GT. 0.) K=MIN0(K,I)                                  TRPN     71
      JJ=1                                                              TRPN     72
C**MULTIPLY ENDPOINTS OF VERTICAL AXES BY FUDGE FACTORS                 TRPN     73
      DO 125 J=1,2                                                      TRPN     74
      JJ=-JJ                                                            TRPN     75
```

Table 12 (Continued)

```
  125 YYLU(J,I) = (1.+JJ*SIGN(.0001,YLU(J,I)))*YLU(J,I)/FCTNORM(I)          TRPN      76
      RETURN                                                               TRPN      77
C**SET RE-TRACING FREQUENCY FOR VERTICAL GRID LINES                        TRPN      78
  150 IF (K .NE. 5) II(K)=IC                                               TRPN      79
      RETURN                                                               TRPN      80
*                                                                          TRPN      81
C*  PLOT DATA, 3 PLOTS PER FRAME                                          TRPN      82
*                                                                          TRPN      83
      ENTRY TRYPLOT                                                        TRPN      84
      IF (K .EQ. 5) RETURN                                                 TRPN      85
      GO TO (201,202,203),NREPLT                                           TRPN      86
  202 WRITE(8) LOC,TL,TR,TRR                                               TRPN      87
      WRITE (8)  DELA(4),FCTNORM(6),IEXP(4)                                TRPN      88
      WRITE(8)  ((PSTORE(I,J),J=1,6),I=1,LOC)                              TRPN      89
      WRITE(6,210)                                                         TRPN      90
  210 FORMAT(1X,*PLOT DATA SAVED ON TAPE8*)                                TRPN      91
      GO TO 201                                                            TRPN      92
  203 READ (8) LOC,TL,TR,TRR                                               TRPN      93
      READ (8)  DELA(4),FCTNORM(6),IEXP(4)                                 TRPN      94
      READ(8) ((PSTORE(I,J),J=1,6),I=1,LOC)                                TRPN      95
      TRR = 1./12.                                                         TRPN      96
      DELA(4)=10.                                                          TRPN      97
      WRITE(6,1000) LOC,TL,TR,TRR                                          TRPN      98
 1000 FORMAT(I10,4E15.5)                                                   TRPN      99
  201 CONTINUE                                                             TRPN     100
C**MULTIPLY ENDPOINTS OF HORIZONTAL (TIME) AXIS BY FUDGE FACTOR            TRPN     101
      XXL=(1.-SIGN(.0001,TL))*TL                                           TRPN     102
      XXR=(1.+SIGN(.0001,TR))*TR                                           TRPN     103
C**RIGHT-HAND MARGIN                                                       TRPN     104
      MTR=TRR*1023                                                         TRPN     105
C**DRAW GRID,PLOT,AND ANNOTATE                                             TRPN     106
      L=0                                                                  TRPN     107
      KK=0                                                                 TRPN     108
      DO 400 I=1,5                                                         TRPN     109
      IF (I .EQ. 4  .OR.  I .EQ. 5)  GO TO 255                             TRPN     110
      IF (DELA(I) .LE. 0.) GO TO 400                                       TRPN     111
      KK=KK+1                                                              TRPN     112
      IF (KK .GT. 1) L=2                                                   TRPN     113
      CALL SETMIV(24,MTR,IYC(1,I),1023-IYC(2,I))                           TRPN     114
      CALL GRID1V(L,XXL,XXR,YYLU(1,I),YYLU(2,I),DELA(4),DELA(I),           TRPN     115
     10,0,-1,-1,4,4)                                                       TRPN     116
  255 DO 250 J=1,LOC                                                       TRPN     117
      IX1 = IXV(PSTORE(J,6))                                               TRPN     118
      IY1 = IYV(PSTORE(J,I) / FCTNORM(I))                                  TRPN     119
      CALL ROINTV(IX1,IY1,0)                                               TRPN     120
      IF (J.NE.1) CALL LINEV(IX1,IY1,IX2,IY2)                              TRPN     121
      IX2 = IX1                                                            TRPN     122
  250 IY2 = IY1                                                            TRPN     123
      IF (I .EQ. 4  .OR.  I .EQ. 5)  GO TO 400                             TRPN     124
      IX1 = 12                                                             TRPN     125
      IY1 = IYC(2,I)+4                                                     TRPN     126
      CALL PRINTV(NCH(I),LABEL(1,I),IX1,IY1)                               TRPN     127
      IX1 = 12+8*NCH(I)                                                    TRPN     128
      IF (IEXP(I).EQ.0) GO TO 270                                          TRPN     129
      CALL PRINTV(8,8H X 10  ),IX1,IY1)                                    TRPN     130
      AEXP = IEXP(I)                                                       TRPN     131
      CALL BNBCDV(AEXP,BCD,IX2)                                            TRPN     132
      IX1 = 8*(NCH(I)+7-IX2)                                               TRPN     133
      SGN = 1H                                                             TRPN     134
      IF(IEXP(I) .GT. 0)      SGN = 1H-                                    TRPN     135
      IY1 = IY1+8                                                          TRPN     136
      CALL PRINTV(1,SGN,IX1,IY1)                                           TRPN     137
      IX1 = IX1+8                                                          TRPN     138
      CALL PRINTV(-IX2,BCD,IX1,IY1)                                        TRPN     139
      GO TO 280                                                            TRPN     140
  270 CALL PRINTV(1,1H),IX1,IY1)                                           TRPN     141
  280 CONTINUE                                                             TRPN     142
      IF (II(I) .NE. 0) CALL PRINTV(-11,11HTIME (SECS),(959-MTR)/2,IYC(1 TRPN     143
     1,I)-8)                                                               TRPN     144
  400  CONTINUE                                                            TRPN     145
      IF (INC .EQ. 1) CALL PRINTV(-55,55HINCOMPLETE PLOT DUE TO PLOT FRE TRPN     146
     1QUENCY EXCEEDING STORAGE  ,30,IYC(1,1)-20)                           TRPN     147
      RETURN                                                               TRPN     148
      END                                                                  TRPN     149
```

Table 12 (Continued)

```
      SUBROUTINE MYEND(A,B,C)                                          MYEND     2
      DIMENSION A(17)                                                  MYEND     3
10    FORMAT(5(5X,O20))                                                MYEND     4
      ENDFILE 6                                                        MYEND     5
      CALL PLOTFIN                                                     MYEND     6
      WRITE(6,10) A,C                                                  MYEND     7
      STOP 6                                                           MYEND     8
      END                                                              MYEND     9
```

```
      SUBROUTINE BUILD(I,IA,N,NP)                                      BUILD     2
C                                                                      BUILD     3
C  *I* IS A POSSIBLE ADDITION TO ARRAY*IA* OF SIZE *N*.                BUILD     4
C  *IA* IS SEARCHED FOR *I*.  IF FOUND, NO ACTION IS TAKEN.            BUILD     5
C  IF NOT FOUND, *I* IS ADDED TO *IA* SO THAT *IA* IS IN INCREASING ORDE BUILD     6
C  *N* IS THEN INCREASED BU 1.  *NP* IS THE SMALLEST POWER OF 2 GREATER BUILD     7
C  THAN OR EQUAL TO N+1.                                               BUILD     8
      DIMENSION IA(1)                                                  BUILD     9
      IF (N .EQ. 0) NP=1                                               BUILD    10
      CALL ISEARCH(I,IA,N,NP,IL,ISTATE)                                BUILD    11
      IF  (ISTATE .EQ. 1) RETURN                                       BUILD    12
      IF  (IL .GT. N) GO TO 7                                          BUILD    13
      NN=N+1                                                           BUILD    14
      DO  5  J= IL,N                                                   BUILD    15
      NN=NN-1                                                          BUILD    16
5     IA(NN+1)=IA(NN)                                                  BUILD    17
7     IA(IL)=I                                                         BUILD    18
      N=N+1                                                            BUILD    19
      NP=IPWR2(N)                                                      BUILD    20
      RETURN                                                           BUILD    21
      END                                                              BUILD    22
```

Table 12 (Continued)

```
      SUBROUTINE FIELDS(IFORMAT,N,IBUFFER)                                FLDS      2
      COMMON/FBLOK/FIELD(6) ,IF(6)                                        FLDS      3
      DIMENSION IALPHA(15),LEN(6)                                         FLDS      4
      DIMENSION IBUFFER(1)                                                FLDS      5
      DATA LEN/60,30,20,15,12,10/                                         FLDS      6
      DATA IALPHA/1R0,1R1,1R2,1R3,1R4,1R5,1R6,1R7,1R8,1R9,1R.,1R-,1R+,    FLDS      7
     1 1RE,1R /                                                           FLDS      8
      ICODE=IFORMAT                                                       FLDS      9
      NN=N+1                                                              FLDS     10
      I1=61                                                               FLDS     11
      DO 100 J=1,N                                                        FLDS     12
C$    STORES(IF,FIELD,JJ)                                                 FLDS     13
      JJ=NN-J                                                             FLDS     14
      ITYPE=ICODE.AND.3B                                                  FLDS     15
      ICODE=SHIFT(ICODE,-2)                                               FLDS     16
      I2=I1-1                                                             FLDS     17
      I1=I1-LEN(N)                                                        FLDS     18
      IF (ITYPE .EQ. 0) GO TO 100                                         FLDS     19
      IBASE1=IBASE2=0                                                     FLDS     20
      IQUANT1=IQUANT2=0                                                   FLDS     21
      FACTOR=1.                                                           FLDS     22
      IDP=0                                                               FLDS     23
      IE=0                                                                FLDS     24
      SIGN1=1.                                                            FLDS     25
      ISIGN2=1                                                            FLDS     26
      DO 80 I=I1,I2                                                       FLDS     27
      DO 45 K=1,15                                                        FLDS     28
      IF (IBUFFER(I) .EQ. IALPHA(K)) GO TO(60,60,60,60,60,60,60,60,60,60  FLDS     29
     1,62,64,66,68,80) K                                                  FLDS     30
 45   CONTINUE                                                            FLDS     31
      GO TO 80                                                            FLDS     32
 60   IF (IE .EQ. 1) GO TO 61                                             FLDS     33
      IBASE1=10*IBASE1+K-1                                                FLDS     34
      IQUANT1=IQUANT1+1                                                   FLDS     35
      IF (IDP .EQ. 1) FACTOR=FACTOR/10.                                   FLDS     36
      GO TO 80                                                            FLDS     37
 61   IBASE2=10*IBASE2+K-1                                                FLDS     38
      IQUANT2=IQUANT2+1                                                   FLDS     39
      GO TO 80                                                            FLDS     40
 62   IDP=1                                                               FLDS     41
      GO TO 80                                                            FLDS     42
 64   IF (IE .EQ. 1 .OR. IQUANT1 .NE. 0) GO TO 65                         FLDS     43
      SIGN1=-SIGN1                                                        FLDS     44
      GO TO 80                                                            FLDS     45
 65   IF (IE .NE. 1) IE=1                                                 FLDS     46
      ISIGN2=-ISIGN2                                                      FLDS     47
      GO TO 80                                                            FLDS     48
 66   IF (IQUANT1 .EQ. 0) GO TO 80                                        FLDS     49
 68   IE=1                                                                FLDS     50
 80   CONTINUE                                                            FLDS     51
      IF (ITYPE .EQ. 2) GO TO 90                                          FLDS     52
      ISIGN1=SIGN1                                                        FLDS     53
      IFACTOR=FACTOR                                                      FLDS     54
      IF (JJ)=ISIGN1*IBASE1*IFACTOR                                       FLDS     55
      GO TO 100                                                           FLDS     56
 90   IF (IQUANT2 .EQ. 0) GO TO 94                                        FLDS     57
      FACTOR2=10.**(ISIGN2*IBASE2)                                        FLDS     58
      GO TO 96                                                            FLDS     59
 94   FACTOR2=1.                                                          FLDS     60
 96   FIELD(JJ)=SIGN1*FACTOR*FLOAT(IBASE1)*FACTOR2                        FLDS     61
C$    OFF(STORES)                                                         FLDS     62
100   CONTINUE                                                            FLDS     63
      RETURN                                                              FLDS     64
      END                                                                 FLDS     65
```

Table 12 (Continued)

```
      SUBROUTINE HSTDFN(IBUFFER,NTYPMX,NTYPES,BIAS,WIDTH,XMAX)                HST      2
      DIMENSION BIAS(NTYPMX),IBUFFER(60),WIDTH(NTYPMX),IALPHA(17)             HST      3
      DATA IALPHA /1R0,1R1,1R2,1R3,1R4,1R5,1R6,1R7,1R8,1R9,1R.,1R-,1R(,       HST      4
     1 1R),1R  ,1R,,1R+/                                                      HST      5
      NTYPES=0                                                                HST      6
      IEND=0                                                                  HST      7
      ISTART=1                                                                HST      8
      IOPT=0                                                                  HST      9
5     IBASE=0                                                                 HST     10
      IQUANT=0                                                                HST     11
      FACTOR=1.                                                               HST     12
      IDP=0                                                                   HST     13
      SIGN=1.                                                                 HST     14
      IOPT=IOPT+1                                                             HST     15
      IF (IOPT .EQ. 3) IOPT=1                                                 HST     16
      IF(ISTART .EQ. 0) GO TO 450                                             HST     17
      ISTART=0                                                                HST     18
      DO 450 L=1,60                                                           HST     19
      DO 400 M=1,17                                                           HST     20
      IF (IBUFFER(L) .EQ. IALPHA(M)) GO TO (410,410,410,410,410,410,          HST     21
     1 410,410,410,410,420,430,440,440,440,450,450     )M                    HST     22
400   CONTINUE                                                                HST     23
      GO TO 900                                                               HST     24
410   IQUANT=IQUANT+1                                                         HST     25
      IBASE=10*IBASE+M-1                                                      HST     26
      IF (IDP .EQ. 1) FACTOR=FACTOR/10.                                       HST     27
      GO TO 450                                                               HST     28
420   IDP=1                                                                   HST     29
      GO TO 450                                                               HST     30
430   SIGN=-SIGN                                                              HST     31
      GO TO 450                                                               HST     32
440   IF (IQUANT .NE. 0) GO TO 910                                            HST     33
450   CONTINUE                                                                HST     34
500   CONTINUE                                                                HST     35
900   IEND=1                                                                  HST     36
      IF (IQUANT .EQ. 0) GO TO 1900                                           HST     37
910   IF (IOPT .EQ. 2) GO TO 1000                                             HST     38
      IF (NTYPES .EQ. 0) GO TO 920                                            HST     39
      BIAS(NTYPES)=XMAX                                                       HST     40
920   XMAX=SIGN*FACTOR*FLOAT(IBASE)                                           HST     41
      IF (IEND .EQ. 1) GO TO 1900                                             HST     42
      GO TO 5                                                                 HST     43
1000  IF (IEND .EQ. 1) GO TO 1900                                             HST     44
      NTYPES=NTYPES+1                                                         HST     45
      IF (NTYPES .GT. NTYPMX) GO TO 1890                                      HST     46
      WIDTH(NTYPES)=SIGN*FACTOR*FLOAT(IBASE)                                  HST     47
      GO TO 5                                                                 HST     48
1890  NTYPES=NTYPES-1                                                         HST     49
1900  RETURN                                                                  HST     50
      END                                                                     HST     51
```

Table 12 (Continued)

```
      SUBROUTINE INUPH                                                    INPH      2
C$    DEBUG                                                               INPH      3
C$    ARRAYS                                                              INPH      4
C** INPUT DATA HANDLER                                                    INPH      5
*                                                                         INPH      6
C**READS INPUT CARDS AND PERFORMS LIMITED INITIALIZATION.  INPUT          INPH      7
C**FEATURES ARE THE FOLLOWING--                                           INPH      8
C    1.  COLS 1-10 ARE TH  MNEMONIC FIELD, COLS 71-80 THE CONTINUATION    INPH      9
C  FIELD, AND COLS 11-70 THE DATA FIELDS (SIX 10 COLUMN FIELDS OR THREE   INPH     10
C  20 COLUMN FIELDS)                                                      INPH     11
C    2. A CONTINUATION CARD HAS A + IN COL 1 FOLLOWED BY THE NON-INITIAL  INPH     12
C  CHARACTERS OF A CONTINUATION FIELD.                                    INPH     13
C    3.  PARAMETERS FOR WHICH NO INPUT CARDS APPEAR WILL MAINTAIN THE     INPH     14
C  VALUES THEY  HAD FOR THE PREVIOUS CASE.  FOR THE FIRST CASE, SUCH      INPH     15
C  PARAMETERS WILL BE SET TO DEFAULT VALUES.                              INPH     16
C    4.  INPUT CARDS MAY APPEAR IN ANY ORDER.                             INPH     17
C    5.  A SINGLE END OF FILE REPRESENTS END-OF-CASE.  A DOUBLE END OF FI INPH     18
C  END-OF JOB.                                                            INPH     19
C                                                                         INPH     20
      INTEGER CNTNATN                                                     INPH     21
      COMMON /INPT1/IEOF,B(11,2),TSTRT,NHALVES,TF,W(10,2),NW(2),EPS(91),  INPH     22
     1ABSERR(91),XLNGTH ,TCPU,CONTROL(4)                                  INPH     23
      COMMON/INPT2/J,ISFIELD,BUFFER(60),ICFIELD                           INPH     24
C-------------------------------------------------------------------      INPH     25
      DIMENSION ICNTAR(3,100),MNEREF(22)                                  INPH     26
      DATA ISTCNT,NMNES/100,22/                                           INPH     27
C-------------------------------------------------------------------      INPH     28
      DATA MNEREF/4LSTAT,4LHALF,4LINIT,4LHYDR,4LNLAD,4LNLBU,4LNL1A,       INPH     29
     1                                                        4LNL1B,      INPH     30
     14LNL2A,4LNL2B,4LINTG,4LDEBU,4LSEA-,4LSPEE,4LCONT,4LPRIN,4LCOMM,     INPH     31
     2 4LIDEN,  4LSCTN,4LPLOT,4LAXES,4LDAMP/                              INPH     32
      DATA IFIRST /0/                                                     INPH     33
      IF (IFIRST .EQ. 1) GO TO 1                                          INPH     34
      IFIRST=1                                                            INPH     35
C**JOB INITIALIZATION                                                     INPH     36
      IEOF=0                                                              INPH     37
C**SET DEFAULT VALUES                                                     INPH     38
      CALL NLIN                                                           INPH     39
      CALL STIN(0)                                                        INPH     40
      CALL HSTIN(0)                                                       INPH     41
      CALL MSCON(0)                                                       INPH     42
      CALL HYDIN(0)                                                       INPH     43
      CALL CNTRN                                                          INPH     44
      CALL INPLTIN                                                        INPH     45
      CALL INPRIN                                                         INPH     46
      CALL INSEGIN(NULL,NULL,NULL,NULL,NULL)                              INPH     47
      CALL INTRIN(0)                                                      INPH     48
C**CASE INITIALIZATION                                                    INPH     49
C     PATCH FOR SLAM ENTRIES                                              INPH     50
      CALL INSMIN(NULL)                                                   INPH     51
      CALL INSMDT(NULL)                                                   INPH     52
C                                                                         INPH     53
 1    ICARDS =0                                                           INPH     54
      CALL INTIN                                                          INPH     55
      CALL INDMPIN                                                        INPH     56
      ITIMES=0                                                            INPH     57
      ICFIELD=0                                                           INPH     58
      ISTORE=JSTORE=1                                                     INPH     59
      NSCNTN=0                                                            INPH     60
      IS=0                                                                INPH     61
      NTAPE=5                                                             INPH     62
      N1=7                                                                INPH     63
      WRITE(6,2)                                                          INPH     64
 2    FORMAT(1H1,*INPUT CARDS*)                                           INPH     65
C* READ A CARD                                                            INPH     66
 5    READ (NTAPE, 10) MFIELD, BUFFER, CNTNATN                            INPH     67
 10   FORMAT(A10,60R1,A10)                                                INPH     68
      IF (EOF(NTAPE)) 30,50                                               INPH     69
C**CHECK FOR END-OF-JOB                                                   INPH     70
 30   IF (ICARDS .GT. 0) GO TO 40                                         INPH     71
      IEOF=1                                                              INPH     72
      RETURN                                                              INPH     73
```

Table 12 (Continued)

```
C**CHECK SCRATCH DISK FOR CONTINUATION CARDS                                    INPH    74
 40   IF (IS .NE. 0) GO TO 45                                                   INPH    75
C**COMPLETE INPUT OPERATIONS AND RETURN                                         INPH    76
 42   CALL NLFIN                                                                INPH    77
      CALL STFIN(NULL)                                                          INPH    78
      CALL HSTFIN(NULL)                                                         INPH    79
      CALL INTRFIN(NULL)                                                        INPH    80
      CALL HSFIN(NULL)                                                          INPH    81
      CALL HYDFIN(NULL)                                                         INPH    82
      CALL CNTFIN                                                               INPH    83
      CALL INPLFIN                                                              INPH    84
      CALL INPRFIN                                                              INPH    85
      CALL INSGFIN(NULL,NULL,NULL,NULL,NULL)                                    INPH    86
      CALL INSMFIN(NULL)                                                        INPH    87
      RETURN                                                                    INPH    88
C**PROCESS CONTINUATION CARDS ON SCRATCH DISK                                   INPH    89
 45   ITIMES=ITIMES+1                                                           INPH    90
      IF (ITIMES .LT. 5) GO TO 47                                               INPH    91
      WRITE(6,46)                                                               INPH    92
 46   FORMAT(1X,*CONTINUATION FIELD ERROR(S)*)                                  INPH    93
      GO TO 42                                                                  INPH    94
 47   IF (NTAPE .NE. 5) REWIND NTAPE                                            INPH    95
      REWIND N1                                                                 INPH    96
      NTAPE=N1                                                                  INPH    97
      N1=15-NTAPE                                                               INPH    98
      IS=0                                                                      INPH    99
      GO TO 5                                                                   INPH   100
C**CHECK COLUMN 1 FOR +                                                         INPH   101
 50   ICARDS=ICARDS+1                                                           INPH   102
      ICOL1=MFIELD .AND. 77000000000000000008                                   INPH   103
      IF (ICOL1 .NE. 1L+) GO TO 500                                             INPH   104
C**CHECK IF CARD PRECEEDING CONTINUATION CARD HAS BEEN READ                     INPH   105
      IF (NSCNTN .EQ. 0) GO TO 400                                              INPH   106
C$    STORES(IS,ICOL2X)                                                         INPH   107
      ICOL2X=SHIFT(MFIELD-ICOL1,6)                                              INPH   108
      DO 100 I=1,NSCNTN                                                         INPH   109
C$    GOTOS                                                                     INPH   110
      IF (ICNTAR(1,I) .EQ. ICOL2X) GO TO 450                                    INPH   111
 100  CONTINUE                                                                  INPH   112
C**WRITE CONTINUATION CARD ON SCRATCH DISK                                      INPH   113
 400  IS=IS+1                                                                   INPH   114
C$    OFF(GOTOS)                                                                INPH   115
C$    OFF(STORES)                                                               INPH   116
      WRITE(N1,10) MFIELD, BUFFER , CNTNATN                                     INPH   117
      GO TO 5                                                                   INPH   118
C**RETRIEVE MNEMONIC AND SUPPORT FIELD FROM STORAGE ARRAY                       INPH   119
 450  MNEMNC=ICNTAR(2,I)                                                        INPH   120
      ISFIELD=ICNTAR(3,I)                                                       INPH   121
C**SET POINTERS INDICATING LOCATIONS AVAILABLE FOR STORAGE PURPOSES             INPH   122
      ICNTAR(2,I)=ISTORE                                                        INPH   123
      ICNTAR(1,I)=10H                                                           INPH   124
C$    STORES(ISTORE)                                                            INPH   125
      IF (I .EQ. NSCNTN) NSCNTN=NSCNTN-1                                        INPH   126
      ISTORE=I                                                                  INPH   127
C$    OFF(STORES)                                                               INPH   128
      GO TO 510                                                                 INPH   129
 500  MNEMNC=MFIELD .AND. 77777777000000000008                                  INPH   130
      ISFIELD=0                                                                 INPH   131
C**SEARCH VOCABULARY FOR MNEMONIC                                               INPH   132
 510  DO 530 K=1,NMNES                                                          INPH   133
      J=K                                                                       INPH   134
      IF (MNEMNC .EQ. MNEREF(K)) GO TO 550                                      INPH   135
 530  CONTINUE                                                                  INPH   136
      WRITE(6,540) MFIELD, BUFFER, CNTNATN                                      INPH   137
 540  FORMAT(1X/1X,*UNRECOGNIZABLE MNEMONIC*/1X,A10,60R1,A10/)                  INPH   138
      GO TO 5                                                                   INPH   139
 550  WRITE(6,560) MFIELD,BUFFER,CNTNATN                                        INPH   140
 560  FORMAT(1X,A10,60R1,A10)                                                   INPH   141
C**MOVE DATA FIELDS INTO APPROPRIATE PARAMETER VARIABLES                        INPH   142
      GO TO (600,610,660,670,640,640,640,640,640,640     ,720,700,710,        INPH   143
     1 650,680,700,5,5,650,690,690,730) J                                       INPH   144
 600  CALL STNDT(NULL) $ GO TO 5                                                INPH   145
```

Table 12 (Continued)

```
 610  CALL HSDT(NULL) $ GO TO 5                                                  INPH     146
 640  CALL NLINPUT $ GO TO 1990                                                  INPH     147
 650  CALL MSGDT(NULL) $ GO TO 5                                                 INPH     148
 660  CALL INITDT $ GO TO 5                                                      INPH     149
 670  CALL HYDRDT(NULL) $ GO TO 5                                                INPH     150
 680  CALL CNTRDT $ GO TO 5                                                      INPH     151
 690  CALL INPLTDT $ GO TO 5                                                     INPH     152
 700  CALL INPRDT $ GO TO 5                                                      INPH     153
 710  CALL INSEGDT(NULL,NULL,NULL,NULL,NULL) $ GO TO 5                           INPH     154
  720  CALL INTRDT(NULL) $GO TO 1990                                             INPH     155
 730  CALL INDMPDT $ GO TO 5                                                     INPH     156
C**CHECK IF CONTINATION FIELD EXPECTED                                          INPH     157
1990  IF (ICFIELD .EQ. 0) GO TO 5                                               INPH     158
C**CHECK FOR BLANK CONTINUATION FIELD                                           INPH     159
      IF (CNTNATN .EQ. 10H                          ) GO TO 5                   INPH     160
C**STORE CONTINUATION FIELD                                                     INPH     161
      ICOL2X=CNTNATN                                                            INPH     162
      DO 2000 I=1,10                                                            INPH     163
      IF((ICOL2X .AND. 77000000000000000000B ).NE. 1L    ) GO TO 2010           INPH     164
 2000 ICOL2X=SHIFT(ICOL2X,6)                                                    INPH     165
C$    STORES(ICNTAR)                                                            INPH     166
 2010 ICNTAR(1,ISTORE)=SHIFT(ICOL2X.AND. 00777777777777777777B,6)               INPH     167
C**STORE MNEMONIC FIELD AND A SUPPORT FIELD                                     INPH     168
      KSTORE=ICNTAR(2,ISTORE)                                                   INPH     169
      IF (ISTORE .LE. ISTCNT) GO TO 2020                                        INPH     170
      WRITE(6,2015)                                                             INPH     171
 2015 FORMAT(1X,*DECK SCRAMBLING EXCEEDS STORAGE CAPABILITIES.*/1X,*ORDE INPH     172
     1R DECK  BETTER OR INCREASE ISTCNT AND SECOND DIMENSION OF ICNTAR.* INPH     173
     2)                                                                         INPH     174
      STOP                                                                      INPH     175
 2020 CONTINUE                                                                  INPH     176
      ICNTAR(2,ISTORE)=MNEMNC                                                   INPH     177
      ICNTAR(3,ISTORE)=ISFIELD                                                  INPH     178
C$    OFF(STORES)                                                               INPH     179
C**SET POINTER INDICATING NEXT AVAILABLE STORAGE LOCATION                       INPH     180
      IF (ISTORE .NE. JSTORE) GO TO 2050                                        INPH     181
      KSTORE=JSTORE=JSTORE+1                                                    INPH     182
C$    STORES(ISTORE)                                                            INPH     183
 2050 IF (ISTORE .GT. NSCNTN) NSCNTN=ISTORE                                     INPH     184
      ISTORE=KSTORE                                                             INPH     185
C$    OFF(STORES)                                                               INPH     186
      ICFIELD=1                                                                 INPH     187
      GO TO 5                                                                   INPH     188
      END                                                                       INPH     189


      BLOCK DATA                                                                INPH     190
      DIMENSION NAPPLY(20)                                                      INPH     191
      COMMON/CBLOK/NCALC,NHSTNS,NGROUP,NL(9),NLINV(9),MNLBNL2(90),              INPH     192
     1 ITYPE(18), YCRITA(18), VCRITA(18),IBSTORE(350),IMSTORE(700),             INPH     193
     3 KWORDS(2),NP(2),IFIRST,NAPPLY                                            INPH     194
      DATA NCALC,NHSTNS/9,20/                                                   INPH     195
      DATA NLINV/3,5,7,9,11,13,15,17,19/                                        INPH     196
      DATA NAPPLY/0,1,1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,9,9/                      INPH     197
      END                                                                       INPH     198
```

Table 12 (Continued)

```
      SUBROUTINE NLINPUT                                                    NLPT      2
      COMPLEX GROUP,GROUPA                                                  NLPT      3
      COMMON/FBLOK/FIELDA(6) ,IF(6)                                         NLPT      4
      EQUIVALENCE(IF(1),ISTN),(IF(1),ICODE),(IF(2),NEXP),                   NLPT      5
     1 (FIELDA(2),YC),(FIELDA(3),C(1)),(FIELDA(3),VC),(LOCC(1) ,LOCA),      NLPT      6
     2(LOCC(2),LOCB),(AMSTORE(1),IMSTORE(1)),(BSTORE(1),IBSTORE(1))         NLPT      7
      COMMON /INPT2/J,ISFIELD,BUFFER(60),ICFIELD                            NLPT      8
      DIMENSION KWORDS(2) ,NP(2)                                            NLPT      9
      DIMENSION KEWDARY(4,2),NUKEWD(2),NWORDS(2)                            NLPT     10
      DIMENSION IFRMT(4),C(4),IWORD(5),LOCC(2),                             NLPT     11
     1 AMSTORE(350,2),BSTORE(350),MC(28),LCL(19),IBSTORE(350)               NLPT     12
     1,IMSTORE(350,2)                                                       NLPT     13
      COMMON/CBLOK /NCALC,NHSTNS,NGROUP,NL,NLINV,MNLBNL2,ITYPE,YCRITA       NLPT     14
     2, VCRITA,IBSTORE,IMSTORE,KWORDS,NP,IFIRST,NAPPLY                      NLPT     15
C-------------------------------------------------------------------------  NLPT     16
      DIMENSION NAPPLY(20)                                                  NLPT     17
      DIMENSION NCFS(18),NPEXP(18)                                          NLPT     18
      DIMENSION GROUPA(18),NL(9),NLINV(9),MNLBNL1(5,9,2),ITYPE(18),         NLPT     19
     1 YCRITA(18),VCRITA(18),MNLBNL2(5,9,2)                                 NLPT     20
      DATA NPNCMX/18/                                                       NLPT     21
      DATA NCALCMX /9/                                                      NLPT     22
C-------------------------------------------------------------------------  NLPT     23
      DATA IFRMT/26528,30008,32008,36528/                                   NLPT     24
      DATA NCFS/3*1,2,2*1,2,3,2*1,2,3,4,2*2,3,2,3/                          NLPT     25
      DATA KEWDARY/2RLL,2RGL,2RLG,2RGG,1RL,1RG/                             NLPT     26
      DATA NPEXP/5*1,3,2*4,5,6,7,2*8,2,2*14,5,7/                            NLPT     27
      DATA IFIRST/0/                                                        NLPT     28
      DATA MC/0,3*1,3*2,1,2,1,3*3,2,3,2,1,3,2,1,2,3,1,3,1,2*3,2/            NLPT     29
      DATA LCL/0,1,2,3,4,5,6,8,10,11,12,14,17,20,21,23,25,26,28/            NLPT     30
      DATA NUKEWD/2,4/                                                      NLPT     31
      IF (ISFIELD .EQ. 0) GO TO 20                                          NLPT     32
C**UNPACK THE SUPPORT FIELD                                                 NLPT     33
      ICARD=MOD(ISFIELD,10000)                                              NLPT     34
      MJSTN=ISFIELD-ICARD                                                   NLPT     35
      GO TO 25                                                              NLPT     36
C**SET FIRST-CARD INDICCATOR                                                NLPT     37
 20   ICARD=1                                                               NLPT     38
C**CATEGORY INDICATORS                                                      NLPT     39
C$    STORES(JJ)                                                            NLPT     40
 25   JJ=J-4                                                                NLPT     41
C$    OFF(STORES)                                                           NLPT     42
      MB=2-MOD(JJ,2)                                                        NLPT     43
      IT=(JJ+1)/2                                                           NLPT     44
      IF (ICARD .NE. 1) GO TO 5000                                          NLPT     45
C**FIRST CARD                                                               NLPT     46
      CALL FIELDS(  IFRMT(IT),6,BUFFER( 1))                                 NLPT     47
      VVC=VC                                                                NLPT     48
      YYC=YC                                                                NLPT     49
      IF (IT .NE. 3)VVC=1.E20                                               NLPT     50
      IF (IT .EQ. 1)YYC=1.E20                                               NLPT     51
C**CHECK IF NEW GROUP                                                       NLPT     52
      GROUP=CMPLX(YYC,VVC)                                                  NLPT     53
      IF (NGROUP .EQ. 0) GO TO 2320                                         NLPT     54
      DO 2310 I=1,NGROUP                                                    NLPT     55
      IF (GROUP .EQ. GROUPA(I)) GO TO 2325                                  NLPT     56
 2310  CONTINUE                                                             NLPT     57
 2320  NGROUP=NGROUP+1                                                      NLPT     58
      I=NGROUP                                                              NLPT     59
      GROUPA(NGROUP)=GROUP                                                  NLPT     60
      YCRITA(NGROUP)=YYC                                                    NLPT     61
      VCRITA(NGROUP)=VVC                                                    NLPT     62
      ITYPE(NGROUP)=IT+  ((IT-1)/2)                                         NLPT     63
C**PREPARE GROUP CODE WORD                                                  NLPT     64
 2325 MJSTN=10000*ISTN                                                      NLPT     65
      IWORD(1)=MJSTN+I                                                      NLPT     66
C**PREPARE FILLER CODE WORD(S)                                              NLPT     67
      II=4-IT                                                               NLPT     68
      INSERT=II+II/2                                                        NLPT     69
      IF (INSERT .EQ. 1) GO TO 2400                                         NLPT     70
      ITEMP=MJSTN+9900                                                      NLPT     71
      DO 2350 I=2,INSERT                                                    NLPT     72
 2350 IWORD(I)=ITEMP+I                                                      NLPT     73
```

Table 12 (Continued)

```
C**INSERT WORD(S)                                                    NLPT    74
      IKEY=0                                                         NLPT    75
 2400 DO 2420 I=1,INSERT                                             NLPT    76
 2420 CALL BUILD(IWORD(I),MNLBNL1(1,1,MB),NWORDS(MB),NP)             NLPT    77
      IF (IT .NE. 1 .OR. ICARD .NE. 1) GO TO 8000                    NLPT    78
      IF (IT .EQ. 1 .AND. IKEY .EQ. 1) GO TO 8000                    NLPT    79
      IKEY=1                                                         NLPT    80
      GO TO 5010                                                     NLPT    81
C**SUBSEQUENT CARDS                                                  NLPT    82
 5000 CALL FIELDS(IFRMT(4),6,BUFFER( 1))                             NLPT    83
 5010 LOC=LOCC(MB)=LOCC(MB)+1                                        NLPT    84
      IF (NEXP .GE. 1 .AND. NEXP .LE. NPNCMX) GO TO 5015             NLPT    85
      WRITE(6,5012)                                                  NLPT    86
 5012 FORMAT(1X,*ERROR IN COLS 21 - 30.  CARD IGNORED.*/)            NLPT    87
      GO TO 8000                                                     NLPT    88
 5015 CONTINUE                                                       NLPT    89
C$    STORES(NN)                                                     NLPT    90
      NN=NCFS(NEXP)                                                  NLPT    91
C$    OFF(STORES)                                                    NLPT    92
      IF (MB .EQ. 2) GO TO 6000                                      NLPT    93
C**ADDED MASS COEFICIENTS AND FIRST DERIVATIVE                       NLPT    94
      IMSTORE(LOC,1)=NEXP                                            NLPT    95
      IMSTORE(LOC,2)=NPEXP(NEXP)                                     NLPT    96
      DO 5345 I=1,NN                                                 NLPT    97
      LOCA=LOCA+1                                                    NLPT    98
      AMSTORE(LOCA,1)=C(I)                                           NLPT    99
 5345 AMSTORE(LOCA,2)=MC(LCL(NEXP)+I)*C(I)                           NLPT   100
      GO TO 7000                                                     NLPT   101
C**BUOYANCY COEFICIENTS                                              NLPT   102
 6000 IBSTORE(LOCB)=NEXP                                             NLPT   103
      DO 6350 I=1,NN                                                 NLPT   104
      LOCB=LOCB+1                                                    NLPT   105
 6350 BSTORE(LOCB)=C(I)                                              NLPT   106
C**DECIPHER CONDITION CODE                                           NLPT   107
 7000 IF (IT .EQ. 1) GO TO 7500                                      NLPT   108
      CALL SCAN(BUFFER, 1,10,KEWDARY(1,4-IT),NUKEWD(IT-1),IKEY,NULL) NLPT   109
      IF (IKEY .NE. 0) GO TO 7500                                    NLPT   110
      WRITE(6,7100)                                                  NLPT   111
 7100 FORMAT(1X,*ILLEGAL CONDITION CODE.  CARD IGNORED.*/)           NLPT   112
      GO TO 8000                                                     NLPT   113
C**PREPARE COEFICIENT/LOCATION CODE WORD                             NLPT   114
 7500 IWORD(1)=MJSTN+1000*IKEY+LOC                                   NLPT   115
      INSERT=1                                                       NLPT   116
      GO TO 2400                                                     NLPT   117
C**SET INDICATORS FOR SUBSEQUENT CARDS                               NLPT   118
 8000 IF (ICARD .EQ. 2*IT-1) GO TO 9000                              NLPT   119
      ICFIELD=1                                                      NLPT   120
      ISFIELD=MJSTN+ICARD +1                                         NLPT   121
      RETURN                                                         NLPT   122
 9000 ICFIELD=0                                                      NLPT   123
      RETURN                                                         NLPT   124
C**DECODE ARRAY OF CODE WORDS                                        NLPT   125
      ENTRY NLFIN                                                    NLPT   126
      IF (NWORDS(1) .EQ. KWORDS(1) .AND. NWORDS(2) .EQ. KWORDS(2))   NLPT   127
     1 RETURN                                                        NLPT   128
      IF (NWORDS(1)/5*5.NE. NWORDS(1) .OR. NWORDS(2)/5*5 .NE. NWORDS(2)) NLPT 129
     1 GO TO 10100                                                   NLPT   130
      KWORDS(1)=NWORDS(1)                                            NLPT   131
      KWORDS(2)=NWORDS(2)                                            NLPT   132
      NWORDS(1)=NWORDS(1)/5                                          NLPT   133
      NWORDS(2)=NWORDS(2)/5                                          NLPT   134
      DO 9180 IK=1,2                                                 NLPT   135
      IW=NWORDS(IK)                                                  NLPT   136
      DO 9180 IJ=1,IW                                                NLPT   137
      KSTN=MNLBNL1(1,IJ,IK)/10000                                    NLPT   138
      DO 9120 IJA=1,NCALC                                            NLPT   139
      IF (KSTN .EQ. NLINV(IJA)) GO TO 9130                           NLPT   140
 9120 CONTINUE                                                       NLPT   141
      GO TO 13000                                                    NLPT   142
 9130 NL(IJA)=NL(IJA)+3-IK                                           NLPT   143
      DO 9180 IJA=1,5                                                NLPT   144
```

Table 12 (Continued)

```
 9180 MNL8NL2(IJA,IJ,IK)=MOD(MNL8NL1(IJA,IJ,IK),1000)                    NLPT    145
      NWORDS(1)=KWORDS(1)                                                NLPT    146
      NWORDS(2)=KWORDS(2)                                                NLPT    147
      RETURN                                                             NLPT    148
C**ERROR MESSAGES                                                        NLPT    149
10100 WRITE (6,10200)                                                    NLPT    150
10200 FORMAT(1X,*COUNT ERROR IN NL-- CARDS*)                             NLPT    151
      STOP                                                               NLPT    152
13000 WRITE(6,13100)  KSTN                                               NLPT    153
13100 FORMAT(1X,*NL-- CARDS AND HYDRO-DATA DO NOT AGREE AT STATION*,      NLPT    154
     1 I5)                                                               NLPT    155
      STOP                                                               NLPT    156
C**INITIALIZATION                                                        NLPT    157
      ENTRY NLIN                                                         NLPT    158
      IF (IFIRST .EQ. 0) KWORDS(1)=KWORDS(2)=0                           NLPT    159
      NWORDS(1)=KWORDS(1)                                                NLPT    160
      NWORDS(2)=KWORDS(2)                                                NLPT    161
      IF (IFIRST .EQ. 1) RETURN                                          NLPT    162
      IFIRST=1                                                           NLPT    163
      NGROUP=0                                                           NLPT    164
      LOCA=LOCB=0                                                        NLPT    165
      NWORDS(1)=NWORDS(2)=0                                              NLPT    166
      DO 1400 I =1,NCALCMX                                               NLPT    167
 1400 NL(I)=0                                                            NLPT    168
      RETURN                                                             NLPT    169
      END                                                                NLPT    170
```

Table 12 (Continued)

```
      SUBROUTINE CNTRN                                                  INCN     2
C                                                                       INCN     3
C** CONTROL PARAMETERS                                                  INCN     4
C                                                                       INCN     5
      DIMENSION KEWDARY(1)                                              INCN     6
      COMMON/INPT2/J,ISFIELD,IBUFFER(60),ICFIELD                        INCN     7
      COMMON/INPT1/IEOF,B(11,2),TSTRT,HMNI,TF,W(10,2),NW(2),EPS(91) ,   CDS      2
     1ABSERR(91),XLNGTH ,TCPU,CONTROL(4)                                CDS      3
C**SET DEFAULTS                                                         INCN     9
      CONTROL(2)=CONTROL(3)=CONTROL(4)=2HNO                             INCN    10
      RETURN                                                            INCN    11
C**ACCEPT DATA                                                          INCN    12
      ENTRY CNTRDT                                                      INCN    13
      DO 10 K=1,3                                                       INCN    14
      LS1=20*K-9                                                        INCN    15
      LS2=LS1+9                                                         INCN    16
      DO 5 KK =LS1,LS2                                                  INCN    17
      IF (IBUFFER(KK) .EQ. 1RY) GO TO 7                                 INCN    18
 5    CONTINUE                                                          INCN    19
      GO TO 10                                                          INCN    20
 7    CONTROL(K+1)=3HYES                                                INCN    21
10    CONTINUE                                                          INCN    22
      RETURN                                                            INCN    23
C**DATA ECHO                                                            INCN    24
      ENTRY CNTFIN                                                      INCN    25
      WRITE (6,50) (CONTROL(L),L=2,4)                                   INCN    26
 50   FORMAT(1H-,*CONTROL OPTIONS*/T10,*PUNCH FINAL CONDITIONS*,T40,A3/ INCN    27
     1 T10,*NON-LINEAR HYDRO FORCE*,T40,A3/T10,*HYDRO DAMPING*,T40,A3)  INCN    28
      RETURN                                                            INCN    29
      END                                                               INCN    30
```

Table 12 (Continued)

```
      SUBROUTINE HSTIN (ITS)                                                    INHY      2
C                                                                               INHY      3
C**HALF STATION DATA                                                            INHY      4
C                                                                               INHY      5
      COMMON /HBLOK/ KAG(21),RINT(21),EI(20),SHPMS(20),DAMPC(20),               CD1       2
     1 DXI,NEVAL,SPEED,BUOY SPG(9),SCF(9),ADDMS(9),VRTMS(20),IBEG,IEND          CD1       3
      COMMON /INPT2/J,ISFIELD,BUFFER(60),ICFIELD                                INHY      7
      DIMENSION YS(91),SF(21),GDOT(21),BDM(20),YDOT(20),YH(9)                   INHY      8
      COMMON/FBLOK/FIELDA(6) ,IF(6)                                             CD2       2
      COMMON //YS                                                               INHY     10
      EQUIVALENCE                                                               CD3       2
     1  (YS,YH),(YS(10),YDOT),(YS(30),GDOT),(YS(51),BDM),(YS(71),SF)            CD3       3
      DATA KX/0/                                                                INHY     12
      IF (ITS .EQ. 0) GO TO 4                                                   INHY     13
      GO TO (2,3) ITS                                                           INHY     14
 2    ITC =1                                                                    INHY     15
      RETURN                                                                    INHY     16
 3    IDC=1                                                                     INHY     17
      RETURN                                                                    INHY     18
C**SET DEFAULT VALUES                                                           INHY     19
  4   ITC=IDC=0                                                                 INHY     20
      IF (KX .EQ. 1) RETURN                                                     INHY     21
      KX=1                                                                      INHY     22
      DO 5 I=1,20                                                               INHY     23
 5    SHPMS(I)=EI(I)=DAMPC(I)=YDOT(I)=BDM(I)=0.                                 INHY     24
      RETURN                                                                    INHY     25
C**ACCEPT DATA                                                                  INHY     26
      ENTRY HSDT                                                                INHY     27
      CALL FIELDS(52528,6,BUFFER)                                               INHY     28
      I=FIELDA(1)+.500001                                                       INHY     29
      SHPMS(I)=FIELDA(2)                                                        INHY     30
      EI(I)=FIELDA(3)                                                           INHY     31
      IF (IDC .EQ. 1) GO TO 10                                                  INHY     32
      DAMPC(I)=FIELDA(4)                                                        INHY     33
 10   IF (ITC .EQ. 1) RETURN                                                    INHY     34
      YDOT(I)=FIELDA(5)                                                         INHY     35
      BDM(I)=FIELDA(6)                                                          INHY     36
      RETURN                                                                    INHY     37
C**DATA ECHO                                                                    INHY     38
      ENTRY HSTFIN                                                              INHY     39
      CALL INDMFIN                                                              INHY     40
      WRITE(6,20)                                                               INHY     41
 20   FORMAT(1H-,*HALF STATION DATA*///T10,*SHPMS = MASS PER UNIT LENGTH        INHY     42
     1*/T10,                                *EI = BENDING RIGIDITY             INHY     43
     1*/T10,*DAMPC = STRUCTURAL DAMPING COEFICIENT*/T10,*YDOT = INITIAL         INHY     44
     2 VERTICAL VELOCITY*/T10,*BDM = INITIAL BENDING MOMENT*///                 INHY     45
     3 1X,*HALF-STATION*,T19,* SHPMS*,T34,*EI*,T49,*DAMPC*,T64,*YDOT*,          INHY     46
     4 T79,*BDM*)                                                               INHY     47
      DO 40 L=1,20                                                              INHY     48
      M=L-.5                                                                    INHY     49
 40   WRITE(6,50) M,SHPMS(L),EI(L),DAMPC(L),YDOT(L),BDM(L)                      INHY     50
 50   FORMAT(1X,F5.1,T16,5E15.7)                                                INHY     51
      END                                                                       INHY     52
```

Table 12 (Continued)

```
      SUBROUTINE HYDIN(ITS)                                              INHYD     2
C                                                                        INHYD     3
C** HYDRO DATA                                                           INHYD     4
C                                                                        INHYD     5
      COMMON /INPT2/J,ISFIELD,BUFFER(60),ICFIELD                         INHYD     6
      COMMON /HBLOK/ KAG(21),RINT(21),EI(20),SHPMS(20),DAMPC(20),        COM1      2
     1 DXI,NEVAL,SPEED,BUOY SPG(9),SCF(9),ADDMS(9),VRTMS(20),IBEG,IEND   COM1      3
      COMMON/FBLOK/FIELDA(6) ,IF(6)                                      COM2      2
      DIMENSION YS(91),SF(21),GDOT(21),BDM(20),YDOT(20),YH(9)            INHYD     9
      COMMON //YS                                                        INHYD    10
      EQUIVALENCE                                                        COM3      2
     1   (YS,YH),(YS(10),YDOT),(YS(30),GDOT),(YS(51),BDM),(YS(71),SF)    COM3      3
      COMMON /HDBLOK/HYDDAMP(9)                                          INHYD    12
      DATA KX /0/                                                        INHYD    13
      IF (ITS .EQ. 0) GO TO 1                                            INHYD    14
      ITC = 1                                                            INHYD    15
      RETURN                                                             INHYD    16
C**SET DEFAULT VALUES                                                    INHYD    17
 1    ITC=1                                                              INHYD    18
      IF (KX .EQ. 1) RETURN                                              INHYD    19
      KX=1                                                               INHYD    20
      DO 5 I=1,9                                                         INHYD    21
 5    ADDMS(I)=BUOY SPG(I)=SCF(I)=HYDDAMP(I)=YH(I)=0.                    INHYD    22
      RETURN                                                             INHYD    23
C**ACCEPT DATA                                                           INHYD    24
      ENTRY HYDRDT                                                       INHYD    25
      CALL FIELDS(32528,6,BUFFER)                                        INHYD    26
      IF (IF(1) .LT. 3 .OR. IF(1) .GT. 19 .OR. (IF(1)/2)*2 .EQ. IF(1))   INHYD    27
     1 GO TO 90                                                          INHYD    28
      I= (IF(1)-1)/2                                                     INHYD    29
      ADDMS(I)=FIELDA(2)                                                 INHYD    30
      BUOY SPG(I)=FIELDA(3)                                              INHYD    31
      SCF(I)=FIELDA(4)                                                   INHYD    32
      HYDDAMP(I)=FIELDA(5)                                               INHYD    33
      IF (ITC .EQ. 1) RETURN                                             INHYD    34
      YH(I)=FIELDA(6)                                                    INHYD    35
      RETURN                                                             INHYD    36
      ENTRY HERR                                                         INHYD    37
 90   WRITE (6,100) IF (1)                                               INHYD    38
 100  FORMAT(1X,*HYDRO-FORCE CANNOT BE CALCULATED AT STATION*,I10)       INHYD    39
      RETURN                                                             INHYD    40
C**DATA ECHO                                                             INHYD    41
      ENTRY HYDFIN                                                       INHYD    42
      WRITE(6,200)                                                       INHYD    43
 200  FORMAT(1H-,*HYDRO-FORCE DATA*//T10,*ADDMS = LINEAR ADDED MASS*/    INHYD    44
     1 T10,*BUOY = LINEAR BYOYANCY*/T10,*SCF = SMITH CORRECTION*/,T10,*H INHYD    45
     2DMP = HYDRO-DAMPING*/T10,*YH = INITIAL VERTICAL DISPLACEMENT*///   INHYD    46
     3 1X,*STATION*,T20,*ADDMS*,T35,*BUOY*,T50,*SCF*,T65,*HDMP*,T80,     INHYD    47
     4*YH*)                                                              INHYD    48
      DO 250 L=1,9                                                       INHYD    49
      I=2*L+1                                                            INHYD    50
250   WRITE(6,300) I,ADDMS(L),BUOYSPG(L),SCF(L),HYDDAMP(L),YH(L)         INHYD    51
 300  FORMAT(1X,I5,10X,5E15.7)                                           INHYD    52
      END                                                                INHYD    53
```

Table 12 (Continued)

```
      SUBROUTINE INDMPIN                                                DMP        2
C                                                                       DMP        3
C** PARAMETERS FOR CALCULATING STRUCTURAL DAMPINF                       DMP        4
C                                                                       DMP        5
      COMMON /HBLOK/ KAG(21),RINT(21),EI(20),SHPMS(20),DAMPC(20),       CO1        2
     1 DXI,NEVAL,SPEED,BUOY SPG(9),SCF(9),ADDMS(9),VRTMS(20),IBEG,IEND  CO1        3
      COMMON/FBLOK/FIELDA(6) ,IF(6)                                     CO2        2
      COMMON/INPT2/J,ISFIELD,BUFFER(60),ICFIELD                         DMP        8
      DATA KX,PI/0, 3.1415926535898/                                    DMP        9
C**ALLOW COEFICIENTS TO BE ON HALF-STATION CARDS                        DMP       10
      CALL HSTIN(0)                                                     DMP       11
      FPI = 2.                                                          DMP       12
      RETURN                                                            DMP       13
C**DAMPC CARD DETECTED                                                  DMP       14
      ENTRY INDMPDT                                                     DMP       15
      KX=1                                                              DMP       16
      CALL HSTIN(2)                                                     DMP       17
      CALL FIELDS(5000B,6,BUFFER)                                       DMP       18
      CPSNAT=FIELDA(1)                                                  DMP       19
      PCCD=FIELDA(2)                                                    DMP       20
      RETURN                                                            DMP       21
C**CALCULATE DAMPING COEFICIENT                                         DMP       22
      ENTRY INDMFIN                                                     DMP       23
      CALL HFGIN(NULL,NULL,NULL,NULL)                                   DMP       24
      IF (KX .EQ. 0) RETURN                                             DMP       25
      PW=FPI*CPSNAT*PCCD                                                DMP       26
      WRITE (6,1) CPSNAT,PCCD                                           DMP       27
 1    FORMAT(1H-,*STRUCTURAL DAMPING PARAMETERS*/T10,*NATURAL FREQUENCY* DMP       28
     1 ,T30,E20.10,3X,*HZ*/T10,*PERCENT DAMPING*,T30,E20.10)            DMP       29
      DO 5 IH=1,20                                                      DMP       30
 5    DAMPC(IH)=VRTMS(IH)*PW                                            DMP       31
      RETURN                                                            DMP       32
      END                                                               DMP       33
```

Table 12 (Continued)

```
      SUBROUTINE INPLTIN                                                    INPL     2
C                                                                           INPL     3
C** PLOTING PARAMETERS                                                      INPL     4
C                                                                           INPL     5
      DIMENSION DFYLU(3)                                                    INPL     6
      COMMON /INPT2/J,ISFIELD,BUFFER(60),ICFIELD                            INPL     7
      COMMON/FBLOK/FIELDA(6) ,IF(6)                                         CD2      2
      COMMON/PBLOK1/NR,TTT(20),R(20),MSBSTN,NAME,CODE,PHONE                 CD6      2
      COMMON/PBLOK4/NREPLT,YLU(2,4)                                         CD7      2
      COMMON/INPT1/IEOF,B(11,2),TSTRT,HMNI,TF,W(10,2),NW(2),EPS(91) ,       CD5      2
     1ABSERR(91),XLNGTH ,TCPU,CONTROL(4)                                    CD5      3
C**SET DEFAULTS                                                             INPL    12
      DATA DFYLU,IDF / 4.,1.E6,15.,0  /                                     INPL    13
      IF(IDF .EQ. 1)          GO TO 6                                       INPL    14
      MSBSTN=11                                                             INPL    15
      NR=0                                                                  INPL    16
      B(1,1)=W(1,1)=1.E20                                                   INPL    17
      NW(1)=1                                                               INPL    18
      DO 5 L = 1,3                                                          INPL    19
      YLU(1,L) = -DFYLU(L)                                                  INPL    20
    5 YLU(2,L) =  DFYLU(L)                                                  INPL    21
      IDF = 1                                                               INPL    22
    6 CONTINUE                                                              INPL    23
      RETURN                                                                INPL    24
C**ACCEPT DATA                                                              INPL    25
      ENTRY INPLTDT                                                         INPL    26
      IF (J .EQ. 21) GO TO 15                                               INPL    27
C**PLOT CARD                                                                INPL    28
      ENCODE(30,10,NAME)(BUFFER(L),L=1,30)                                  INPL    29
 10   FORMAT(30R1)                                                          INPL    30
      CALL FIELDS(518,6,BUFFER)                                             INPL    31
      NR=1                                                                  INPL    32
      W(1,1)=FIELDA(4)                                                      INPL    33
      R(1)=FIELDA(5)                                                        INPL    34
      NREPLT = IF(6)                                                        INPL    35
      RETURN                                                                INPL    36
C**AXES CARD                                                                INPL    37
 15   CALL FIELDS(5252B,6,BUFFER)                                           INPL    38
      DO 20 L=1,6                                                           INPL    39
 20   YLU(L)=FIELDA(L)                                                      INPL    40
      RETURN                                                                INPL    41
C**DERIVED PLOTTING PARAMETERS                                              INPL    42
      ENTRY INPLFIN                                                         INPL    43
      IF(NR .EQ. 0)           GO TO 900                                     INPL    44
      IF(NREPLT .EQ. 3)       GO TO 800                                     INPL    45
      B(2,1) = TF                                                           INPL    46
      TTT(1)=B(1,1)=5.                                                      INPL    47
      TTT(2)=TF                                                             INPL    48
C**DATA ECHO                                                                INPL    49
  800 WRITE(6,25) NAME,CODE,PHONE,W(1,1),R(1),YLU                           INPL    50
 25   FORMAT(1H-,*PLOTTING PARAMETERS*,20X,3A10/E20.8,5X,*SECS BETWEEN P    INPL    51
     1LOTTED VALUES*/E20.8,5X,*SECS PER FRAME*///5X,*AXES LIMITS*//         INPL    52
     2 T10,*PITCH*T20,2E20.8/T10,*MOSHP BNDG MMNT*,T20,2E20.8/              INPL    53
     2T10,*WAVE HEIGHT*,T20,2E20.8)                                         INPL    54
      RETURN                                                                INPL    55
 900  WRITE(6,910)                                                          INPL    56
 910  FORMAT(1H-,*NO PLOTTING*)                                             INPL    57
      RETURN                                                                INPL    58
      END                                                                   INPL    59
```

Table 12 (Continued)

```
      SUBROUTINE INPRIN                                                  INPR    2
C                                                                        INPR    3
C**PRINTING TIMES                                                        INPR    4
C                                                                        INPR    5
      COMMON/FBLOK/FIELDA(6) ,IF(6)                                      CD2     2
      COMMON /HBLOK/ KAG(21),RINT(21),EI(20),SHPMS(20),DAMPC(20),        CD1     2
     1 DKI,NEVAL,SPEED,BUOY SPG(9),SCF(9),ADDMS(9),VRTMS(20),IBEG,IEND   CD1     3
      COMMON/INPT1/IEOF,B(11,2),TSTRT,HMNI,TF,W(10,2),NW(2),EPS(91) ,    CD5     2
     1ABSERR(91),XLNGTH ,TCPU,CONTROL(4)                                 CD5     3
      COMMON /INPT2/J,ISFIELD,BUFFER(60),ICFIELD                         INPR    9
C**SET DEFAULTS (PART 1)                                                 INPR    10
      NW(2)=0                                                            INPR    11
      B(1,2) = W(1,2) = 1.E20                                            INPR    12
      IBEG=IEND=-1                                                       INPR    13
      RETURN                                                             INPR    14
C**ACCEPT DATA                                                           INPR    15
      ENTRY INPROT                                                       INPR    16
      IF (J .EQ. 16) GO TO 5                                             INPR    17
C**DEBUG                                                                 INPR    18
      CALL FIELDS(2400B,6,BUFFER)                                        INPR    19
      IBEG=IF(1)                                                         INPR    20
      IEND=IF(2)                                                         INPR    21
      RETURN                                                             INPR    22
C**PRINT                                                                 INPR    23
 5    CALL HSTOFN(BUFFER,10,NW(2),B(1,2),W(1,2),XMAX)                    INPR    24
      IN=NW(2)+1                                                         INPR    25
      B(IN,2)=XMAX                                                       INPR    26
      RETURN                                                             INPR    27
      ENTRY INPRFIN                                                      INPR    28
C**DATA ECHO                                                             INPR    29
 10   I1=NW(2)                                                           INPR    30
      WRITE(6,20) IBEG,IEND,(B(I,2),W(I,2),I=1,I1),B(I1+1,2)             INPR    31
 20   FORMAT(1H-,*DEBUG*/T10,*IBEG*,I10,5X,*IEND*I10/1H-,*PRINTING TIMES INPR    32
     1*,//T5,E15.5,3(*(*,E15.5,*)*,E15.5))                               INPR    33
      RETURN                                                             INPR    34
      END                                                                INPR    35
```

Table 12 (Continued)

```
      SUBROUTINE INTIN                                                    ININ      2
C                                                                         ININ      3
C**INITIAL CONDITIONS                                                     ININ      4
C                                                                         ININ      5
      COMMON /HBLOK/ KAG(21),RINT(21),EI(20),SHPNS(20),DAMPC(20),         CO1       2
     1 DXI,NEVAL,SPEED,BUOY SPG(9),SCF(9),ADDMS(9),VRTMS(20),IBEG,IEND    CO1       3
      COMMON/FBLOK/FIELDA(6) ,IF(6)                                       CO2       2
      COMMON/INPT1/IEOF,B(11,2),TSTRT,HMNI,TF,W(10,2),NW(2),EPS(91) ,     CO5       2
     1ABSERR(31),XLNGTH ,TCPU,CONTROL(4)                                  CO5       3
      COMMON /INPT2/J,ISFIELD,BUFFER(60),ICFIELD                          ININ      9
      DIMENSION YS(91),SF(21),GDOT(21),BDM(20),YDOT(20),YH(9)             ININ     10
      COMMON //YS                                                         ININ     11
      EQUIVALENCE                                                         CO3       2
     1  (YS,YH),(YS(10),YDOT),(YS(30),GDOT),(YS(51),BDM),(YS(71),SF)      CO3       3
      DATA I2,I3,I4/2,3,4/                                                ININ     13
C**ALLOW I.C."S TO BE ON STATION, HALF-STATION, HYDROFORC CARDS            ININ     14
      CALL STIN (0)                                                       ININ     15
      CALL  HSTIN(0)                                                      ININ     16
      CALL HYDIN(0)                                                       ININ     17
      CALL INTRIN(0)                                                      ININ     18
      IST=IHS=IHY=0                                                       ININ     19
      RETURN                                                              ININ     20
C**TRANSGENERATED I.C."S DETECTED                                         ININ     21
      ENTRY INITDT                                                        ININ     22
      CALL FIELDS(30008,6,BUFFER)                                         ININ     23
      KTYPE=IF(1)                                                         ININ     24
      FIELDA(1)=FIELDA(2)                                                 ININ     25
      CALL FIELDS(128,3,BUFFER)                                           ININ     26
      GO TO (10,20,30,40) KTYPE                                           ININ     27
 10   TSTRT=FIELDA(2)                                                     ININ     28
      CALL INTRIN(1)                                                      ININ     29
      RETURN                                                              ININ     30
 20   IF (IHS .EQ. 1) GO TO 25                                            ININ     31
      IHS=1                                                               ININ     32
      CALL HSTIN(1)                                                       ININ     33
 25   I=FIELDA(1)+.50001                                                  ININ     34
      YDOT(I)=FIELDA(2)                                                   ININ     35
      BDM(I)=FIELDA(3)                                                    ININ     36
      RETURN                                                              ININ     37
 30   IF (IST .EQ. 1) GO TO 35                                            ININ     38
      IST=1                                                               ININ     39
      CALL STIN(1)                                                        ININ     40
 35   I=FIELDA(1)+.0001                                                   ININ     41
      GDOT(I)=FIELDA(2)                                                   ININ     42
      SF(I)=FIELDA(3)                                                     ININ     43
      RETURN                                                              ININ     44
 40   IF (IHY .EQ. 1) GO TO 45                                            ININ     45
      IHY=1                                                               ININ     46
      CALL HYDIN(1)                                                       ININ     47
 45   IF(1)=FIELDA(1)+.0001                                               ININ     48
      IF (IF(1) .LT. 3 .OR. IF(1) .GT. 19  .OR. (IF(1)/2)*2 .EQ. IF(1))   ININ     49
     1 CALL HERR                                                          ININ     50
      I=(IF(1)-1)/2                                                       ININ     51
      YH(I)=FIELDA(2)                                                     ININ     52
      RETURN                                                              ININ     53
C                                                                         ININ     54
C  PUNCH FINAL CONDITIONS                                                 ININ     55
C                                                                         ININ     56
      ENTRY PUNCHIT                                                       ININ     57
      WRITE (7,50)  TSTRT                                                 ININ     58
 50   FORMAT('INITIAL',T20,'1',T31,E20.10)                                ININ     59
      DO 60 I=1,20                                                        ININ     60
      HS=I-.5                                                             ININ     61
 60   WRITE(7,70) I2,HS,YDOT(I),BDM(I)                                    ININ     62
 70   FORMAT('INITIAL',T11,I10,F10.1,2E20.10)                             ININ     63
      DO 80 I=1,21                                                        ININ     64
      IF (I .GT. 9) GO TO 75                                              ININ     65
      II=2*I+1                                                            ININ     66
      WRITE(7,90) I4,II,YH(I)                                             ININ     67
 75   II=I-1                                                              ININ     68
 80   WRITE(7,90) I3,II,GDOT(I),SF(I)                                     ININ     69
 90   FORMAT('INITIAL',T11,2I10,2E20.10)                                  ININ     70
      END                                                                 ININ     71
```

Table 12 (Continued)

```
      SUBROUTINE INTRIN(ITS)                                                INTR      2
C                                                                           INTR      3
C**INTEGRATION DATA                                                         INTR      4
C                                                                           INTR      5
      COMMON/FBLOK/FIELDA(6) ,IF(6)                                         CD2       2
      COMMON/INPT1/IEOF,B(11,2),TSTRT,HMNI,TF,W(10,2),NW(2),EPS(91) ,       CD5       2
     1ABSERR(91),XLNGTH ,TCPU,CONTROL(4)                                    CD5       3
      COMMON /INPT2/J,ISFIELD,BUFFER(60),ICFIELD                            INTR      6
      DATA KX/0/                                                            INTR      9
      IF (ITS .EQ. 0) GO TO 1                                               INTR     10
      ITC=1                                                                 INTR     11
      RETURN                                                                INTR     12
C**SET DEFAULT VALUES                                                       INTR     13
 1    ITC=0                                                                 INTR     14
      IF (KX .EQ. 1) RETURN                                                 INTR     15
      KX=1                                                                  INTR     16
      HMNI=TSTRT=TF=TCPU=0.                                                 INTR     17
      RETURN                                                                INTR     18
C**ACCEPT DATA                                                              INTR     19
      ENTRY INTRDT                                                          INTR     20
      ISFIELD=ISFIELD+1                                                     INTR     21
      IF (ISFIELD .NE. 1) GO TO 20                                          INTR     22
C**FIRST CARD                                                               INTR     23
 10   CALL FIELDS(52400B,6,BUFFER)                                          INTR     24
      HMNI=FIELDA(1)                                                        INTR     25
      TF=FIELDA(3)                                                          INTR     26
      TCPU=FIELDA(4)                                                        INTR     27
      IF (ITC .EQ. 1) GO TO 29                                              INTR     28
      TSTRT=FIELDA(2)                                                       INTR     29
      GO TO 29                                                              INTR     30
C**SECOND CARD                                                              INTR     31
 20   CALL FIELDS(52500B,6,BUFFER)                                          INTR     32
      IF (ISFIELD .EQ. 3)    GO TO 21                                       INTR     33
      DO 35 I = 1,20                                                        INTR     34
      IF (I .GT. 9) GO TO 33                                                INTR     35
      EPS(I)=FIELDA(1)                                                      INTR     36
 33   EPS(I+9)=FIELDA(2)                                                    INTR     37
      EPS(I+29)=FIELDA(3)                                                   INTR     38
      EPS(I+50)=FIELDA(4)                                                   INTR     39
 35   EPS(I+70)=FIELDA(5)                                                   INTR     40
      EPS(50)=FIELDA(3)                                                     INTR     41
      EPS(91)=FIELDA(5)                                                     INTR     42
 29   ICFIELD=1                                                             INTR     43
      RETURN                                                                INTR     44
C**THIRD CARD                                                               INTR     45
 21   DO 25 I = 1,20                                                        INTR     46
      IF (I .GT. 9) GO TO 23                                                INTR     47
      ABSERR(I)=FIELDA(1)                                                   INTR     48
 23   ABSERR(I+9)=FIELDA(2)                                                 INTR     49
      ABSERR(I+29)=FIELDA(3)                                                INTR     50
      ABSERR(I+50)=FIELDA(4)                                                INTR     51
 25   ABSERR(I+70)=FIELDA(5)                                                INTR     52
      ABSERR(50)=FIELDA(3)                                                  INTR     53
      ABSERR(91)=FIELDA(5)                                                  INTR     54
      ICFIELD=0                                                             INTR     55
      RETURN                                                                INTR     56
C**DATA ECHO                                                                INTR     57
      ENTRY INTRFIN                                                         INTR     58
      WRITE (6,200) HMNI,TSTRT,TF,TCPU,                                     INTR     59
     1            EPS(1),   EPS(10),   EPS(30),   EPS(51),   EPS(71),      INTR     60
     2            ABSERR(1),ABSERR(10),ABSERR(30),ABSERR(51),ABSERR(71)    INTR     61
 200  FORMAT(1H-,*INTEGRATION DATA*//T10,*MAXIMUM INTERNAL STEP SIZE (HM    INTR     62
     1NI)*,T50,E20.10/T10,*SIMULATION START (TSTRT)*,T50,E20.10/T10,*SIM    INTR     63
     3ULATION END (TF)*,T50,E20.10/T10,*CPU SECS*,T50,E20.10///T5,*MAXIM    INTR     64
     1UM ERRORS*/T25,*VERTICAL*,                                            INTR     65
     4 T40,*VERTICAL*,T60,*ANGULAR*,T80,*BENDING*,T100,*SHEAR*/T20,*DISP    INTR     66
     5LACEMENT*,T40,*VELOCITY*,T60,*VELOCITY*,T80,*MOMENT*,T100,*FORCE*/    INTR     67
     6T5,*RELATIVE*,T15,5E20.10/T5,*ABSOLUTE*,T15,5E20.10)                  INTR     68
      RETURN                                                                INTR     69
      END                                                                   INTR     70
```

Table 12 (Continued)

```
      SUBROUTINE MSCON                                                  INMS      2
C                                                                       INMS      3
C  MISCELLANEOUS DATA--                                                 INMS      4
C       SPEED                                                           INMS      5
C       SECTION LENGTH                                                  INMS      6
C                                                                       INMS      7
      COMMON/FBLOK/FIELDA(6) ,IF(6)                                     CD2       2
      COMMON /HBLOK/ KAG(21),RINT(21),EI(20),SHPMS(20),DAMPC(20),       CD1       2
     1 DXI,NEVAL,SPEED,BUOY SPG(9),SCF(9),ADDMS(9),VRTMS(20),IBEG,IEND  CD1       3
      COMMON/INPT1/IEOF,B(11,2),TSTRT,HMNI,TF,W(10,2),NW(2),EPS(91) ,   CD5       2
     1ABSERR(91),XLNGTH ,TCPU,CONTROL(4)                                CD5       3
      COMMON /INPT2/J,ISFIELD,BUFFER(60),ICFIELD                        INMS     11
C**SET DEFAULT VZLUES                                                   INMS     12
      SPEED=0.                                                          INMS     13
      DXI=1.                                                            INMS     14
      RETURN                                                            INMS     15
C**ACCEPT DATA                                                          INMS     16
      ENTRY MSCDT                                                       INMS     17
      CALL FIELDS(40008,6,BUFFER)                                       INMS     18
      IF (J .EQ. 19) GO TO 10                                           INMS     19
C**SPEED                                                                INMS     20
      SPEED=FIELDA(1)                                                   INMS     21
      RETURN                                                            INMS     22
C**SECTION LENGTH                                                       INMS     23
 10   DXI=FIELDA(1)                                                     INMS     24
      XLNGTH = 16.*DXI                                                  INMS     25
      RETURN                                                            INMS     26
C**DATA ECHO                                                            INMS     27
      ENTRY MSFIN                                                       INMS     28
      WRITE(6,20) SPEED,DXI                                             INMS     29
 20   FORMAT(1H-,*MISCELLANEOUS DATA*//T10,*SPEED = *,T20,E15.7/T10,    INMS     30
     1                                                *SECTION LENG    INMS     31
     1TH = *,T30,E15.7)                                                 INMS     32
      RETURN                                                            INMS     33
      END                                                               INMS     34
```

Table 12 (Continued)

```
      SUBROUTINE STIN(ITS)                                                    INST      2
C                                                                             INST      3
C**STATION CARDS                                                              INST      4
C                                                                             INST      5
      REAL KAG                                                                INST      6
      COMMON /HBLOK/ KAG(21),RINT(21),EI(20),SHPMS(20),DAMPC(20),             CO1       2
     1 DXI,NEVAL,SPEED,BUOY SPG(9),SCF(9),ADDMS(9),VRTMS(20),IBEG,IEND        CO1       3
      COMMON /INPT2/J,ISFIELD,BUFFER(60),ICFIELD                              INST      8
      DIMENSION YS(91),SF(21),GDOT(21),BDM(20),YDOT(20),YH(9)                 INST      9
      COMMON //YS                                                             INST     10
      COMMON/FBLOK/FIELDA(6) ,IF(6)                                           CO2       2
      EQUIVALENCE                                                             CO3       2
     1  (YS,YH),(YS(10),YDOT),(YS(30),GDOT),(YS(51),BDM),(YS(71),SF)          CO3       3
      DATA KX /0/                                                             INST     13
      IF (ITS .EQ. 0) GO TO 1                                                 INST     14
      ITC=1                                                                   INST     15
      RETURN                                                                  INST     16
C**SET DEFAULT VALUES                                                         INST     17
 1    ITC=0                                                                   INST     18
      IF (KX .EQ. 1) RETURN                                                   INST     19
      KX=1                                                                    INST     20
      DO 5 I=1,21                                                             INST     21
 5    KAG(I)=RINT(I)=SF(I)=GDOT(I)=.0                                         INST     22
      RETURN                                                                  INST     23
C**ACCEPT DATA                                                                INST     24
      ENTRY STNDT                                                             INST     25
      CALL FIELDS(32508,6,BUFFER)                                             INST     26
      I=IF(1)+1                                                               INST     27
      KAG(I)=FIELDA(2)                                                        INST     28
      RINT(I)=FIELDA(3)                                                       INST     29
      IF (ITC .EQ. 1) RETURN                                                  INST     30
      GDOT(I)=FIELDA(4)                                                       INST     31
      SF(I)=FIELDA(5)                                                         INST     32
      RETURN                                                                  INST     33
C**DATA ECHO                                                                  INST     34
      ENTRY STFIN                                                             INST     35
      WRITE(6,10)                                                             INST     36
 10   FORMAT(1H1,*STATION DATA*///T10,*KAG = SHEAR STIFFNESS*/                INST     37
     1 T10,*RINT = MOMENT OF INERTIA*/                                        INST     38
     1 T10,*GDOT = INITIAL ANGULAR VELOCITY*/T10,*SF = INITIAL SHEAR FOR      INST     39
     2CE*                                                                     INST     40
     2////1X,*STATION*,T19,*KAG*,T34,*RINT*,T49,*GDOT*,T64,                   INST     41
     3 *SF*)                                                                  INST     42
      DO 20 L=1,21                                                            INST     43
      I=L-1                                                                   INST     44
 20   WRITE(6,30) I,KAG(L),RINT(L),GDOT(L),SF(L)                              INST     45
 30   FORMAT(1X,I5,10X,4E15.7)                                                INST     46
      RETURN                                                                  INST     47
      END                                                                     INST     48
```

Table 12 (Continued)

```
      FUNCTION IPWR2(J)                                                  IPWR     2
C  VALUE RETURNED IS SMALLEST POWER OF 2 GREATER THAN OR EQUAL TO *J*    IPWR     3
      JJ=J+1                                                             IPWR     4
      IPWR2=1                                                            IPWR     5
 1    CONTINUE                                                           IPWR     6
      IF(IPWR2 .GE. JJ) RETURN                                           IPWR     7
      IPWR2=IPWR2*2                                                      IPWR     8
      GO TO 1                                                            IPWR     9
      END                                                                IPWR    10
```

```
      SUBROUTINE ISEARCH(X,CLSMRK,NUCMS,IPP,IL,ISTATE)                   SRCH     2
C  ARRAY *CLSMRK* OF SIZE *NUCMS* IS SEARCHED FOR *X*.  IF FOUND,        SRCH     3
C  *ISTATE* IS SET TO 1, AND *IL* IS SUCH THAT  *CLSMRK(IL)* IS EQUAL    SRCH     4
C  TO *X*.  IF NOT FOUND, *ISTATE* IS ZEROED AND *IL* IS SUCH THAT       SRCH     5
C  *CLSMRK(IL)* .LT. *X*, PLUS *CLSMRK(IL+1)*  .GT. *X* FOR  *IL* .NE.   SRCH     6
C  *NUCMS*.                                                             SRCH     7
      INTEGER CLSMRK,X                                                   SRCH     8
      DIMENSION CLSMRK(50)                                               SRCH     9
      KKK=0                                                              SRCH    10
      IF (NUCMS.EQ. 0) GO TO 700                                         SRCH    11
      J=IPP/2                                                            SRCH    12
      K=IPP/4                                                            SRCH    13
      IF(J .GT. NUCMS) GO TO 20                                          SRCH    14
 9    IF (X .GE. CLSMRK(J)) GO TO 10                                     SRCH    15
 20   IF(K.EQ. 0) GO TO 41                                               SRCH    16
      J=J-K                                                              SRCH    17
      K=K/2                                                              SRCH    18
      IF(J.GT.NUCMS) GO TO 20                                            SRCH    19
      GO TO 9                                                            SRCH    20
 10   IF (K .EQ. 0) GO TO 31                                             SRCH    21
      J=J+K                                                              SRCH    22
      K=K/2                                                              SRCH    23
      IF ( J .GT. NUCMS) GO TO 20                                        SRCH    24
      GO TO 9                                                            SRCH    25
 31   KKK=J                                                              SRCH    26
      GO TO 51                                                           SRCH    27
 41   KKK=J-1                                                            SRCH    28
 51   IF(KKK .EQ. 0) GO TO 700                                           SRCH    29
      IF(X .NE. CLSMRK(KKK)) GO TO 700                                   SRCH    30
      ISTATE=1                                                           SRCH    31
      IL=KKK                                                             SRCH    32
      RETURN                                                             SRCH    33
 700  ISTATE=0                                                           SRCH    34
      IL=KKK+1                                                           SRCH    35
      RETURN                                                             SRCH    36
      END                                                                SRCH    37
```

Table 12 (Continued)

```
      REAL FUNCTION NOLEX(NTYPE,C,X)                                    NLX      2
C$    DEBUG                                                             NLX      3
C$    STORES(NOLEX)                                                     NLX      4
      DIMENSION C(1)                                                    NLX      5
      COMMON/NBLOCK/DX                                                  NLX      6
      GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18) NTYPE        NLX      7
 1    NOLEX =C(1) $ RETURN                                              NLX      8
 2    NOLEX=C(1)*X      $ RETURN                                        NLX      9
 3    NOLEX=C(1)*DX $ RETURN                                            NLX     10
 4    NOLEX=C(1)*X+C(2) $ RETURN                                        NLX     11
 5    NOLEX=C(1)*X**2 $ RETURN                                          NLX     12
 6    NOLEX=C(1)*DX**2 $ RETURN                                         NLX     13
 7    NOLEX=X*(C(1)*X+C(2)) $ RETURN                                    NLX     14
 8    NOLEX=X*(C(1)*X+C(2))+C(3)            $ RETURN                    NLX     15
 9    NOLEX=C(1)*X**3 $ RETURN                                          NLX     16
 10   NOLEX=C(1)*DX**3 $ RETURN                                         NLX     17
 11   NOLEX=X**2*(C(1)*X+C(2)) $ RETURN                                 NLX     18
 12   NOLEX=X*(X*(C(1)*X+C(2))+C(3)  )                                  NLX     19
 13   NOLEX=X*(X*(C(1)*X+C(2))+C(3))+C(4) $ RETURN                      NLX     20
 14   NOLEX=C(1)*X**2+C(2) $ RETURN                                     NLX     21
 15   NOLEX=X*(C(1)*X**2+C(2)) $ RETURN                                 NLX     22
 16   NOLEX=X*(C(1)*X**2+C(2))+C(3) $ RETURN                            NLX     23
 17   NOLEX=C(1)*X**3+C(2) $ RETURN                                     NLX     24
 18   NOLEX=X**2*(C(1)*X+C(2))+C(3) $ RETURN                            NLX     25
      END                                                               NLX     26
```

Table 12 (Continued)

```
      SUBROUTINE SCAN (ISTRING,ISTART,LENGTH,KEWDARY,NUKEWD,KEYCODE,ILAS SCAN        2
     1T)                                                                 SCAN        3
C--  ISTRING AND KEWDARY MUST BE RIGHT JUSTIFIED WITH ZERO FILL          SCAN        4
      DIMENSION ISTRING(1),IALA(10)                                      SCAN        5
      DIMENSION KEWDARY(1)                                               SCAN        6
      IF(ISTART .GT. LENGTH .OR. NUKEWD .LE. 0) GO TO 150                SCAN        7
      KK=0                                                               SCAN        8
      J=0                                                                SCAN        9
      DO 100 I=ISTART,LENGTH                                             SCAN       10
      IF (ISTRING(I) .NE. 1RG .AND. ISTRING(I) .NE. 1RL) GO TO 100       SCAN       11
      J=J+1                                                              SCAN       12
      KK=ISTRING(I)+KK*64                                                SCAN       13
  100 CONTINUE                                                           SCAN       14
      DO 65 L=1,NUKEWD                                                   SCAN       15
      IF(KK .EQ. KEWDARY(L)) GO TO 200                                   SCAN       16
   65 CONTINUE                                                           SCAN       17
  150 ILAST=LENGTH    +ISTART                                            SCAN       18
      KEYCODE=0                                                          SCAN       19
      RETURN                                                             SCAN       20
  200 ILAST=I                                                            SCAN       21
      KEYCODE=L                                                          SCAN       22
      RETURN                                                             SCAN       23
      END                                                                SCAN       24
```

```
      FUNCTION POLY(A,X)                                                 POLY        2
      DIMENSION A(1)                                                     POLY        3
      N = IFIX(A(1)) + 1                                                 POLY        4
      J = N + 2                                                          POLY        5
      POLY = 0.                                                          POLY        6
      DO 400 I=1,N                                                       POLY        7
      J = J - 1                                                          POLY        8
  400 POLY = POLY * X + A(J)                                             POLY        9
      RETURN                                                             POLY       10
      END                                                                POLY       11
```

Table 12 (Continued)

```
INPUT CARDS
COMMENT    ESSEX   DISCRETE WAVE TRAIN
SEA-DATA  1.08470672 .598444   .804804    4.11238748 -21.97272 -29.54952
CONTROL   PUNCH =     NO       NL HYDRO =    NO      HYD DAMP=    YES
PRINT     0.          (1.)     50.        $
INTGRTN   .015625     0.       50.        1500.                          +INT1
+INT1     .1          .01      .01        .2          .05                +INT2
+INT2     1.          1.       .2         2000.       300.
PLOT      SWYBRANIEC1844       71940      0.1         50.        2       0.
AXES      -3.         3.       -1.+6      1.+6        -12.       12.     0.
DAMPC     .750        .023
SPEED     65.
SCTN LNGTH 41.
STATION   0           1339870.  26.86
STATION   1           1636730.  254.08
STATION   2           2484850.  419.05
STATION   3           3531440.  615.63
STATION   4           4770220.  830.89
STATION   5           6056130.  1015.57
STATION   6           7142850.  1157.16
STATION   7           7765150.  1267.88
STATION   8           7831900.  1337.42
STATION   9           7509160.  1341.24
STATION   10          7180390.  1290.96
STATION   11          6366860.  1234.65
STATION   12          7020550.  1169.07
STATION   13          7529840.  1061.59
STATION   14          8151090.  910.29
STATION   15          7321430.  736.49
STATION   16          4000000.  546.97
STATION   17          2455090.  354.50
STATION   18          1855200.  203.79
STATION   19          1431060.  130.25
STATION   20          1269350.  84.31
HALF-STTN .5          .45467    5.05075+9  .018137
HALF-STTN 1.5         .74401    7.76725+9  .080892
HALF-STTN 2.5         .84735    12.31675+9 .085016
HALF-STTN 3.5         1.13281   18.30495+9 .096434
HALF-STTN 4.5         1.70245   27.046 +9  .17205
HALF-STTN 5.5         2.10287   36.95395+9 .18807
HALF-STTN 6.5         2.20362   44.24075+9 .22888
HALF-STTN 7.5         2.14550   49.67695+9 .22655
HALF-STTN 8.5         2.09512   55.16235+9 .24224
HALF-STTN 9.5         2.10158   58.75385+9 .2425
HALF-STTN 10.5        2.04374   59.00065+9 .2317
HALF-STTN 11.5        1.99308   50.67175+9 .22967
HALF-STTN 12.5        1.94270   37.47675+9 .19103
HALF-STTN 13.5        1.85357   33.20715+9 .16746
HALF-STTN 14.5        1.76574   34.4828 +9 .13112
HALF-STTN 15.5        1.65465   28.64295+9 .12667
HALF-STTN 16.5        1.35756   19.27405+9 .074156
HALF-STTN 17.5        .8706     13.65365+9 .054678
```

Table 12 (Continued)

```
HALF-STTN 18.5        .54122     9.82855 +9 .025649
HALF-STTN 19.5        .36426     7.4482  +9 .018570
HYDRO     3           1.27805    2.605691   .9077      .9
HYDRO     5           2.59878    3.184146   2.006      1.5
HYDRO     7           3.518293   3.295122   2.6419     1.5
HYDRO     9           3.960976   3.301220   2.8519     1.5
HYDRO     11          3.74878    3.273171   2.7917     1.5
HYDRO     13          2.832927   3.032927   2.3768     1.7
HYDRO     15          1.512195   2.321951   1.592      1.5
HYDRO     17          .496341    1.235366   .8216      .53
HYDRO     19          .1         .340244    .3184      .05
NL1B          3          0.                                          GG1
+G1             G        5        .05179                              GG2
+G2             L        1          0.
NL1BUOYANCY       15       0.                                         BB1
+B1           YGYC     5             .02512                          BB2
+B2             L          1       0.
NLBUOYANCY  17             5        .01345
NL1BUOYANCY       19      -10                                         AA1
+A1           YLYC         6        .03024                          AA2
+A2           YGYC         1          0
NL2ADDMASS     3           0.         0.                             FF1
+F1            LG          2        -.03732                          FF2
+F2           GG         2          -.06829                          FF3
+F3          L L         4            -.01474   .5602                FF4
+F4           G L           4       -.09154     .5602
NL1ADDMASS      15         0.                                         EE1
+E1             G        7          -.7805E-03-26.13-3               EE2
+E2           L          2          -26.13E-3
NL2ADDMASS  17           0.           0.                             DD1
+D1           LEG          7        1.568-03  -13.17E-3               DD2
+D2           G G             2      -13.17E-3                        DD3
+D3            LL           7        2.72E-3    -13.17E-3            DD4
+D4               GL           2     -13.17E-3
NL2ADDMASS          19     -10.        -0                            CC1
+C1             LG         10        -.0001476                        CC2
+C2            GG          1           -0.                           CC3
+C3           L L             10      -.2658E-3                        CC4
+C4            GAL             1        0
```

Table 12 (Continued)

STATION DATA

KAG = SHEAR STIFFNESS
RINT = MOMENT OF INERTIA
GDOT = INITIAL ANGULAR VELOCITY
SF = INITIAL SHEAR FORCE

STATION	KAG	RINT	GDOT	SF
0	.1339870E+07	.2686000E+02	0.	0.
1	.1636730E+07	.2540800E+03	0.	0.
2	.2484850E+07	.4190500E+03	0.	0.
3	.3531440E+07	.6156300E+03	0.	0.
4	.4770220E+07	.8308900E+03	0.	0.
5	.6056130E+07	.1015570E+04	0.	0.
6	.7142860E+07	.1157160E+04	0.	0.
7	.7765150E+07	.1267860E+04	0.	0.
8	.7831900E+07	.1337420E+04	0.	0.
9	.7509160E+07	.1341240E+04	0.	0.
10	.7180390E+07	.1290960E+04	0.	0.
11	.6966860E+07	.1234650E+04	0.	0.
12	.7020550E+07	.1169070E+04	0.	0.
13	.7529840E+07	.1061590E+04	0.	0.
14	.8151090E+07	.9102900E+03	0.	0.
15	.7321430E+07	.7364900E+03	0.	0.
16	.4000000E+07	.5469700E+03	0.	0.
17	.2455090E+07	.3545000E+03	0.	0.
18	.1855200E+07	.2037900E+03	0.	0.
19	.1431060E+07	.1302500E+03	0.	0.
20	.1269350E+07	.8431000E+02	0.	0.

STRUCTURAL DAMPING PARAMETERS

NATURAL FREQUENCY	.7500000000E+00	HZ
PERCENT DAMPING	.2300000000E-01	

HALF STATION DATA

SHPMS = MASS PER UNIT LENGTH
EI = BENDING RIGIDITY
DAMPC = STRUCTURAL DAMPING COEFICIENT
YDOT = INITIAL VERTICAL VELOCITY
BDM = INITIAL BENDING MOMENT

Table 12 (Continued)

HALF-STATION	SHPMS	EI	DAMPC	YDOT	BDM
.5	.4546700E+00	.5050750E+10	.1568612E-01	0.	0.
1.5	.7440100E+00	.7767250E+10	.6976107E-01	0.	0.
2.5	.8473500E+00	.1231675E+11	.7332630E-01	0.	0.
3.5	.1132810E+01	.1830495E+11	.8317467E-01	0.	0.
4.5	.1702450E+01	.2704600E+11	.1483924E+00	0.	0.
5.5	.2102870E+01	.3695395E+11	.1622069E+00	0.	0.
6.5	.2203620E+01	.4424075E+11	.1974060E+00	0.	0.
7.5	.2145500E+01	.4967695E+11	.1954009E+00	0.	0.
8.5	.2095120E+01	.5516235E+11	.2089353E+00	0.	0.
9.5	.2101580E+01	.5875385E+11	.2091582E+00	0.	0.
10.5	.2043740E+01	.5900065E+11	.1998419E+00	0.	0.
11.5	.1993080E+01	.5067175E+11	.1980942E+00	0.	0.
12.5	.1942700E+01	.3747675E+11	.1647591E+00	0.	0.
13.5	.1853570E+01	.3320715E+11	.1616841E+00	0.	0.
14.5	.1765740E+01	.3448280E+11	.1130888E+00	0.	0.
15.5	.1654650E+01	.2864295E+11	.1092562E+00	0.	0.
16.5	.1357560E+01	.1927405E+11	.6395958E-01	0.	0.
17.5	.8706000E+00	.1365365E+11	.4715946E-01	0.	0.
18.5	.5412200E+00	.9828550E+10	.2212209E-01	0.	0.
19.5	.3642600E+00	.7448200E+10	.1601697E-01	0.	0.

INTEGRATION DATA

MAXIMUM INTERNAL STEP SIZE (HMNI)	.1562500000E-01
SIMULATION START (TSTRT)	0.
SIMULATION END (TF)	.5000000000E+02
CPU SECS	.1500000000E+04

MAXIMUM ERRORS

	VERTICAL DISPLACEMENT	VERTICAL VELOCITY	ANGULAR VELOCITY	BENDING MOMENT	SHEAR FORCE
RELATIVE	.1000000000E+00	.1000000000E-01	.1000000000E-01	.2000000000E+00	.5000000000E-01
ABSOLUTE	.1000000000E+01	.1000000000E+01	.2000000000E+00	.2000000000E+04	.3000000000E+03

MISCELLANEOUS DATA

SPEED = .6500000E+02
SECTION LENGTH = .4100000E+02

HYDRO-FORCE DATA

ADDMS = LINEAR ADDED MASS
BUOY = LINEAR BYOYANCY
SCF = SMITH CORRECTION
HDMP = HYDRO-DAMPING
YH = INITIAL VERTICAL DISPLACEMENT

Table 12 (Continued)

STATION	ADDMS	BUOY	SCF	HDMP	YH
3	.1278050E+01	.2605691E+01	.9077000E+00	.9000000E+00	0.
5	.2598780E+01	.3184148E+01	.2006000E+01	.1500000E+01	0.
7	.3518293E+01	.3295122E+01	.2641900E+01	.1500000E+01	0.
9	.3960976E+01	.3301220E+01	.2651900E+01	.1500000E+01	0.
11	.3748780E+01	.3273171E+01	.2791700E+01	.1500000E+01	0.
13	.2832927E+01	.3032927E+01	.2376800E+01	.1700000E+01	0.
15	.1512195E+01	.2321951E+01	.1592000E+01	.1500000E+01	0.
17	.4963410E+00	.1235366E+01	.8216000E+00	.5300000E+00	0.
19	.1000000E+00	.3402442E+00	.3184000E+00	.5000000E-01	0.

CONTROL OPTIONS

PUNCH FINAL CONDITIONS	NO
NON-LINEAR HYDRO FORCE	NO
HYDRO DAMPING	YES

PLOTTING PARAMETERS SWYBRANIEC1844 71940

.10000000E+00 SECS BETWEEN PLOTTED VALUES
.50000000E+02 SECS PER FRAME

AXES LIMITS

PITCH	-.30000000E+01	.30000000E+01
MDSHP BNDG	-.10000000E+07	.10000000E+07
WAVE HEIGH	-.12000000E+02	.12000000E+02

PLOTTING PARAMETERS

DEBUG
IBEG -1 IEND -1

PRINTING TIMES

0. (.10000E+01) .50000E+02

SEAGEN IS USING DISCRETE WAVE TRAIN

Table 12 (Continued)

INPUT DATA

DRAFT = SHIPS DRAFT = .100000E+02 FT
DENS = MASS DENSITY OF WATER = .200000E+01 SLUGS/CU.FT.
BTANG = ALPHA, BUTTOCK ANGLE = 0. DEGREES
DRANG = BETA, DEADRISE ANGLE = .300000E+01 DEGREES
DX = STATION LENGTH = .410000E+02 FT
BSLP = SLOPE RELATED TO WIDTH ARRAY = .100000E+00

PRE-COMPUTED DATA

SINAL = SINE OF ALPHA = 0.
COSAL = COSINE OF ALPHA = .100000E+01
TANAL = TANGENT OF ALPHA = 0.
COSBET = COSINE OF BETA = .998630E+00
TANBET = TANGENT OF BETA = .524078E-01
DX2 = 2. * STATION LENGTH = .820000E+02 FT
DX10 = 10. * STATION LENGTH = .410000E+03 FT
DNSBY2 = MASS DENSITY OF WATER / 2. = .100000E+01 SLUGS/CU.FT.

HEIGHT OF DEADRISE DRHGT(10)	WIDTH OF SLAM REGION WIDTH(10)
.241076E+01 FT	.460000E+02 FT
.241076E+01 FT	.460000E+02 FT
.241076E+01 FT	.460000E+02 FT
.241076E+01 FT	.460000E+02 FT
.241076E+01 FT	.460000E+02 FT
.241076E+01 FT	.460000E+02 FT
.241076E+01 FT	.460000E+02 FT
.241076E+01 FT	.460000E+02 FT
.241076E+01 FT	.460000E+02 FT
.241076E+01 FT	.460000E+02 FT

COEF(8,1)	COEF(8,2)	COEF(8,3)
.200000E+01	.500000E+01	.500000E+01
.320000E+00	.218209E+01	.474674E+01
.854664E+01	-.541549E+02	-.770645E+02
.150461E+03	.668885E+03	.517540E+03
0.	-.439937E+04	-.174879E+04
0.	.146328E+05	.294746E+04
0.	-.193390E+05	-.197505E+04
0.	0.	0.

Table 12 (Continued)

```
TIME   25.000000          20235 FCN EVALS
WV HGT     .209510E+01  .580456E+01  .674629E+01  .477054E+01  .231953E+00 -.517776E+01 -.911319E+01 -.964912E+01 -.623443E+01
WV VEL     .366895E+01  .239626E+01 -.501813E+00 -.381372E+01 -.592360E+01 -.561665E+01 -.273463E+01  .165536E+01  .568915E+01
REL VEL   -.850287E+00 -.109518E+01  .314247E+00  .217737E+01  .288405E+01  .123813E+01 -.290789E+01 -.848482E+01 -.136738E+02
REL DISP -.807934E-01 -.397971E+01 -.513114E+01 -.338568E+01  .907192E+00  .606785E+01  .976407E+01  .100770E+02  .645522E+01
WAVE ACC   .240909E+00 -.245826E+01 -.373185E+01 -.329713E+01 -.114352E+01  .185680E+01  .432788E+01  .504491E+01  .354574E+01
RL WV VL   .224032E+01  .906477E+00  .738079E-01 -.215867E+00 -.862701E+00 -.227222E+01 -.449690E+01 -.717339E+01 -.108028E+02
ADD MASS   .127805E+01  .259878E+01  .351829E+01  .396098E+01  .374878E+01  .283293E+01  .151220E+01  .496341E+00  .100000E+00
N-L A MS   .210523E+00  .126720E+02  .169077E+02  .111769E+02 -.296939E+01 -.184033E+02 -.226717E+02 -.124486E+02 -.219635E+01
SPC DRV   -.439870E+00  .128465E+02  .201986E+02  .138194E+02 -.657728E+01 -.263198E+02 -.273590E+02 -.122997E+02 -.641690E+00
SCF       -.114215E+00  .605752E+01  .626799E+01 -.423831E-02 -.111234E+02 -.207612E+02 -.202548E+02 -.945901E+01 -.195416E+00
HYD DMP   -.213850E+01  .469781E+01  .614978E+01  .319563E+00 -.982922E+01 -.168384E+02 -.135094E+02 -.565712E+01  .344726E+00
SLM FRC    .140000E+02  .178000E+02  .106390E+03  .327671E+01  .542238E+02  .858718E+01  .976407E+01  .100770E+02  .735331E+01
 .766628E+01  .700000E+01  .800000E+01
SMOOTH    0.           0.           0.           .197822E+00  .589673E+00  .154132E+00 0.           .177019E-02  .969519E+00
 .339672E+01  .134619E+01 0.
HYD FRC   0.          -.213850E+01 -.213850E+01 -.213850E+01  .469781E+01  .469781E+01  .614978E+01  .614978E+01  .319563E+00
 .319563E+00 -.982922E+01 -.982922E+01 -.166406E+02 -.162488E+02 -.133553E+02 -.135094E+02 -.565535E+01 -.468760E+01
 .374144E+01  .169091E+01
VRT MS     .212540E+01  .438123E+01  .572191E+01  .605610E+01  .579252E+01  .477563E+01  .327794E+01  .185390E+01  .641220E+00
```

```
TIME       25.00000 SECS          183.822 SECS CPU TIME
TIME STEP   .00625 SECS            20235 DERIVATIVE EVALUATIONS

VERTICAL POSITION
  2.8143E+00  1.8248E+00  1.6152E+00  1.3849E+00  1.1391E+00  8.9009E-01  6.5087E-01  4.2792E-01  2.2079E-01

VERTICAL VELOCITY
  4.7206E+00  3.9647E+00  3.1971E+00  2.4403E+00  1.6744E+00  9.2778E-01  1.7934E-01 -5.5446E-01 -1.2803E+00 -1.9919E+00
 -2.6947E+00 -3.3844E+00 -4.0509E+00 -4.7062E+00 -5.3381E+00 -5.9469E+00 -6.5480E+00 -7.1109E+00 -7.6915E+00 -8.2778E+00

PITCH              -1.566E-01

ANGULAR VELOCITY
 -1.8396E-02 -1.8396E-02 -1.8403E-02 -1.8321E-02 -1.8252E-02 -1.8089E-02 -1.7964E-02 -1.7752E-02 -1.7581E-02 -1.7293E-02
 -1.7110E-02 -1.6800E-02 -1.6455E-02 -1.6109E-02 -1.5460E-02 -1.5441E-02 -1.4746E-02 -1.4684E-02 -1.4711E-02 -1.4242E-02
 -1.4242E-02

BENDING MOMENT
  0.          -2.1804E+03 -1.3059E+04 -3.2264E+04 -5.8964E+04 -8.2101E+04 -9.7899E+04 -1.0300E+05 -9.3476E+04 -7.5509E+04
 -4.4135E+04 -1.4764E+04  1.5018E+04  3.4695E+04  4.1025E+04  4.2824E+04  3.0499E+04  1.8687E+04  5.3852E+03  0.

SHEAR FORCE
  0.          -5.1617E+01 -2.6395E+02 -4.6475E+02 -6.4469E+02 -5.5276E+02 -3.7698E+02 -1.0479E+02  2.2813E+02  4.6577E+02
  7.5320E+02  7.2708E+02  7.3892E+02  4.5566E+02  2.1740E+02 -2.9350E+01 -2.4464E+02 -2.9267E+02 -3.4257E+02 -1.1681E+02
  0.
```

TABLE 13 - PRINCIPAL FORTRAN VARIABLES FOR SUBROUTINE HYD FRC

Call Letters:	*Definition*	*Symbol*
ADDMS(I)	Added mass at still waterline	m_0
BUOYSPG(I)	Buoyancy spring	$K_b = \rho g b_1 \Delta X$
CEL	Celerity of wave propagation	c
DXI	Length of one station	ΔX
EI(I)	Bending rigidity	EI
HF(I)	Total hydrodynamic force	P
HFTEMP(I)	Intermediate sums of hydrodynamic force terms	
HYDDAMP(I)	Real part of hydrodynamic damping coefficient	$C(\omega)$
NAPPLY(J)	Array giving station number at which Jth hydrodynamic force is applied	
NCALC	Number of stations at which hydrodynamic force is calculated	
NLINV(J)	Station number at which Jth nonlinear hydrodynamic force is computed	
SCF(I)	Smith correction factor	$\rho A_0 \Delta X / m_0$
SHPMS(I)	Ship mass	m_s, m
SPEED	Forward ship speed	U
T	Time coordinate	t
VCRIT(J)	Relative velocity V_r at which nonlinear force changes expression for Jth nonlinear group	
VMTEMP(I)	Intermediate sums of mass terms (ship- plus added-mass)	
VR(I)	Vertical velocity of ship relative to wave	V_r
VRTMS	Total mass (ship- plus added-mass)	$m_s + m_v$
YCRIT(J)	Relative displacement Y_r at which nonlinear force changes expression for Jth nonlinear group	
YH	Vertical displacement of ship	Y
YHDOT	Vertical velocity of ship	$\dot{Y}$
YR	Vertical displacement of ship relative to wave	Y_r
YRDOT	Partial time derivative of Y_r	$\partial Y_r / \partial t$

TABLE 13 (Continued)

Call Letters:	*Definition*	*Symbol*
YW	Height of wave surface	Y_w
YWDOT	Vertical velocity of wave surface	$\dot{Y}_w$
YWDD	Vertical acceleration of wave surface	$\ddot{Y}_w$
ZBY4DXI		$U/4\Delta X$

TABLE 14 - INPUT CARD TYPES

Type:	*Logical Card Count Should Equal*	*Action if One or More Omitted*
HYDRØ	Number of calculated hydroforce stations (9)	At omitted stations ADDMS=BUOY=SCF=HDMP =YH=0
NL_A	Number of nonlinear added mass stations (≤9)	At omitted stations added mass is linear
NL_B	Number of nonlinear buoyancy stations (≤9)	At omitted stations buoyancy is linear
CØMMENT	0 or more	None
SCTN	0 or 1	DXI=1
SPEED	0 or 1	SPEED=0
CØNTRØL	0 or 1	$CNTRL_2$=$CNTRL_3$=$CNTRL_4$=2HNØ
DEBUG	0 or 1	IBEG=IEND=-1
INTGRTN	0 or 1	HMNI=TSTRT=TF=TCPU=0
AXES	0 or 1	P_ℓ=-5, P_u=5, W_ℓ=-10, W_u=10, B_ℓ=-5.E6 B_u=5.E6
PLØT	0 or 1	No plots
PRINT	0 or 1	No printed output
SEA-GEN (Case 1)	0 or 1	FREQ=WVHGH=0
DAMPC	0 or 1	Structural damping coefficient on HALF-STTN cards are used
HALF-STTN	Number of half-stations (20)	At omitted half-stations SHPMS=EI=DAMPC =YDØT=BDM=0

TABLE 14 (Continued)

Type:	*Logical Card Count Should Equal*	*Action if One or More Omitted*
INITIAL	0 or more	Initial conditions on corresponding INTGRTN, HALF-STTN, STATIØN, or HYDRØ card are used if columns 11 through 20 of omitted card contain 1, 2, 3, or 4, respectively
STATIØN	Number of stations (21)	At omitted stations KAG=RINT=SF=GDØT=0

Input cards comprise one or more physical records. In a case input deck, physical records may appear in any order. Columns 1 through 10 of the initial record of a card contain the card-type name. Input cards consisting of more than one physical record require a unique continuation field in columns 72 through 80 of all its nonterminal records. On noninitial records, column 1 contains a +, and columns 2 through 10 match the continuation field of the logically preceding record.

Numeric input requirements are less strict than I and E, i.e., integer and exponent, formats in that

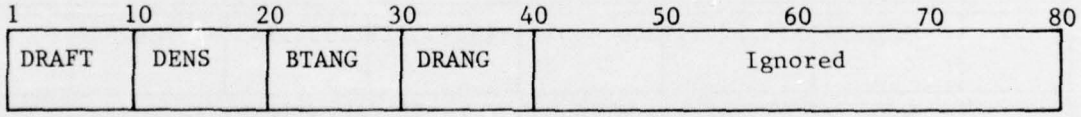
1. Integer fields need not be right justified
2. Nonnumeric characters are ignored, except for the decimal point, e.g., blank and comma
3. Real fields lacking a decimal point are treated as fractionless real numbers
4. Integer fields containing a decimal point are truncated.

In using the program ROSAS, the following input cards are needed and are explained in detail.

ΔΔΔ1. Slamming Data

Seven data cards are needed for SLAM. They must come first in the data deck and be in the following order.

Card 1:



Field:	*Format*	*Contents*
DRAFT	F10.0	Draft of ship in feet
DENS	F10.0	Mass density of water in slugs per cubic foot
BTANG	F10.0	Buttock angle in degrees
DRANG	F10.0	Deadrise angle in degrees

Example: DRAFT = 10. ft

DENS = 2. slugs/ft^3

BTANG = 0. deg

DRANG = 3. deg

1	10	20	30	40	50	60	70	80
10.	2.	0.	3.					

Cards 2, 3, and 4:

1	10	20	30	40	50	60	70	80
CØEF(1,1) (1,2) (1,3)	(2,1) (2,2) (2,3)	(3,1) (3,2) (3,3)	(4,1) (4,2) (4,3)	e	t	c.	(8,1) (8,2) (8,3)

Field:	*Format*	*Contents*
CØEF(8,3)	3(8F10.0)	Polynomial Orders and Coefficients

Example:

$K_1 = 0.32 + 8.5466395\ \xi + 150.46086\ \xi^2$ for $0 \leq \xi < 0.0383972$

$K_2 = 2.1820894 - \cdots - 19339.04\ \xi^5$ for $0.0383972 \leq \xi < 0.191986$

$K_3 = 4.748742 - \cdots - 1975.052\ \xi^5$ for $0.191986 \leq \xi < 0.349066$

1	10	20	30	40	50	60	70	80
2.	.32	8.5466395	150.46086	0.	0.	0.	0.	
5.	2.1820894	-54.154911	668.88525	-4399.3721	14632.763	-19339.040	0.	
5.	4.748742	-77.064451	517.53967	-1748.7944	2947.4596	-1975.0520	0.	

Cards 5 and 6:

1	10	20	30	40	50	60	70	80
WIDTH(1) (9)	(2) (10)	(3)	(4)	(5)	(6)	(7)	(8)	

Field:	*Format*	
WIDTH(10)	2(8F10.0)	Width in feet of the slam region at Stations 11 through 20

Example: b=WIDTH=46 ft

1	10	20	30	40	50	60	70	80
46.	46.	46.	46.	46.	46.	46.	46.	

1	10	20	30	40	50	60	70	80
46.	46.							

Card 7:

1	10	20	30	40	50	60	70	80
BSLP	I1281							

Field:	*Format*	*Contents*
BSLP		BSLP is a parameter used by an iteration that determines the length of the slamming region. BSLP should be chosen so that $$BSLP \geq \frac{1.1}{\Delta X} \max \{b_i - b_{i+1}\} \tan \beta + 0.1$$ where β is the deadrise angle, b_i, b_{i+1} are the half widths at adjacent stations, and ΔX is distance between stations
I1281	I10	Number of coefficients used to smooth the slam forces

Example: BSLP=0.1

1	10	20	30	40	50	60	70	80
.1	1281							

ΔΔΔ2. Hydrodynamic Force Data

HYDRØ - Hydrodynamic Force Calculation Card

1	10	20	30	40	50	60	70	80
HYDRØ	ISTN	ADDMS	BUØY	SCF	HDMP	YH		

Field:	*Contents*
ISTN	Station number at which HYDRØ forces are to be calculated. At present, ISTN can only assume values 3, 5, 7, 9, 11, 13, 15, 17, and 19.
ADDMS	Linear added mass in ton-seconds squared per square foot (ton-s^2/ft^2) at station ISTN

Field:	*Contents*
BUØY	Linear buoyancy in tons per square foot (ton/ft^2) at station ISTN
SCF	Smith correction factor in square feet per ton-seconds squared (ft^2/ton-s^2) at station ISTN
HDMP	Hydrodamping in ton-seconds per foot (ton-s/ft) at station ISTN
YH	Initial vertical displacement in feet at station ISTN. (This field is ignored if appropriate INITIAL cards are present in the input deck.)

Example: Station=ISTN=3

m_0=ADDMS=1.27805 ton-s^2/ft^2

k_b=BUØY=2.605691 ton/ft^2

ρA_0=SCF=0.9077 ft^2/ton-s^2

C(ω)=HDMP=0.9 ton-s/ft^2

Y=YH=0

1	10	20	30	40	50	60	70	80
HYDRØ	3	1.27805	2.605691	.9077	.9			

NL_A/NL_B - Nonlinear Added Mass/Buoyancy Card

(a) Unconditional (No Restriction on Y_r and V_r):

1	10	20	30	40	50	60	70	80
NLA NLB	ISTN	NEXP	C1	C2	C3	C4		

(b) Y_r Conditional:

1	10	20	30	40	50	60	70	80
NL1A NL1B	ISTN	YC	ignored				+abc	

1	10	20	30	40	50	60	70	80
+abc	CØDEY$_1$	NEXP$_1$	C1$_1$	C2$_1$	C3$_1$	C4$_1$	+def	
+def	CØDEY$_2$	NEXP$_2$	C1$_2$	C2$_2$	C3$_2$	C4$_2$		

(c) Y_r and V_r Conditional:

1	10	20	30	40	50	60	70	80
NL2A NL2B	ISTN	YC	VC	ignored			+ghi	
+ghi	CØDE2$_1$	NEXP$_1$	C1$_1$	C2$_1$	C3$_1$	C4$_1$	+jkl	
+jkl	CØDE2$_2$	NEXP$_2$	C1$_2$	C2$_2$	C3$_2$	C4$_2$	+mno	
+mno	CØDE2$_3$	NEXP$_3$	C1$_3$	C2$_3$	C3$_3$	C4$_3$	+pqr	
+pqr	CØDE2$_4$	NEXP$_4$	C1$_4$	C2$_4$	C3$_4$	C4$_4$		

Field:	*Contents*
ISTN	Station number as before
YC	Y_c (Y_r - critical) in feet
VC	V_c (V_r - critical) in feet
CØDEY	Alphanumeric condition code (Y_r - conditional): YLYC for $Y_r < Y_c$ YGYC for $Y_r > Y_c$
CØDE2	Alphanumeric condition code (Y_r, V_r - conditional): YLYC/VLVC for $Y_r < Y_c$, $V_r < V_c$ YGYC/VLVC for $Y_r > Y_c$, $V_r < V_c$ YLYC/VGVC for $Y_r < Y_c$, $V_r > V_c$ YGYC/VGVC for $Y_r > Y_c$, $V_r > V_c$
C_s	Coefficients
NEXP	Expression number

NEXP	Expression	Number of Coefficients
1	C_1	1
2	$C_1 Y_r$	1
3	$C_1(Y_r - Y_c)$	1
4	$C_1 Y_r + C_2$	2
5	$C_1 Y_r^2$	1
6	$C_1 (Y_r - Y_c)^2$	1
7	$C_1 Y_r^2 + C_2 Y_r$	2
8	$C_1 Y_r^2 + C_2 Y_r + C_3$	3
9	$C_1 Y_r^3$	1
10	$C_1 (Y_r - Y_c)^3$	1
11	$C_1 Y_r^3 + C_2 Y_r^2$	2
12	$C_1 Y_r^3 + C_2 Y_r^2 + C_3 Y_r$	3
13	$C_1 Y_r^3 + C_2 Y_r^2 + C_3 Y_r + C_4$	4
14	$C_1 Y_r^2 + C_2$	2
15	$C_1 Y_r^3 + C_2 Y_r$	2
16	$C_1 Y_r^3 + C_2 Y_r + C_3$	3
17	$C_1 Y_r^3 + C_2$	2
18	$C_1 Y_r^3 + C_2 Y_r^2 + C_3$	3

Example: Station 17, Added mass 10^3 ton-s^2/ft^2

$$\bar{m} = -13.17\, Y_r + 1.568\, Y_r^2 \quad Y_r < 0,\ V_r > 0$$
$$= -13.17\, Y_r \quad > 0, \quad > 0$$
$$= -13.17\, Y_r + 2.720\, Y_r^2 \quad < 0, \quad < 0$$
$$= -13.17\, Y_r \quad > 0, \quad < 0$$

NL2ADDMASS	17	0.	0.				DD1
+D1	LG	7	1.568E-3	-13.17E-3			DD2
+D2	GG	2	-13.17E-3				DD3
+D3	LL	7	2.72E-3	-13.17E-3			DD4
+D4	GL	2	-13.17E-3				

ΔΔΔ3. Ship Response Data

DAMPC - Calculated Structural Damping Card

1	10	20	30	40	50	60	70	80
DAMPC	CPSNAT	PCCP	Ignored					

Field: *Contents*

CPSNAT — Ship natural frequency in hertz

PCCP — Percent of critical damping, e.g., 10 percent is entered as .10

Example: Ship natural frequency=CPSNAT=1 Hz
Percent critical damping=2 percent

1	10	20	30	40	50	60	70	80
DAMP	1.	.02						

HALF-STTN - Half-Station Card

1	10	20	30	40	50	60	70	80
HALF-STTN	HSTN	SHPMS	EI	DAMPC	YDØT	BDM		

Field:	*Contents*
HSTN	Half-station number, e.g., 4.5
SHPMS	Ship mass per unit length in ton-s^2/ft^2 at half-station HSTN
EI	Bending rigidity in ton-feet squared (ton-ft^2) at half-station HSTN
DAMPC	Structural damping coefficient in ton-seconds per square foot (ton-s/ft^2) at half-station HSTN (This field is ignored if a DAMPC card is present in the input deck.)
YDØT*	Initial vertical velocity of ship in feet per second at half station HSTN.
BDM*	Initial bending moment in feet-tons at half station HSTN.

Example: At Station 1.5

HSTN = 1.5

m = SHPMS= 0.74401 ton-s^2/ft^2

EI = 7.76725×10^9 ton-ft^2

C = DAMPC= 0.080882

$\dot{Y}_0$ = YDØT = 0 (initial condition)

M_0 = BDM = 0 (initial condition)

1	10	20	30	40	50	60	70	80
HALF-STTN	1.5	.74401	7.76725+9	.080882				

STATIØN - Station Card

1	10	20	30	40	50	60	70	80
STATIØN	ISTN	KAG	RINT	GDØT	SF	ignored		

Field:	*Contents*
ISTN	Station number
KAG	Shear stiffness in tons at station ISTN

*Ignored if appropriate INITIAL cards are present in input deck.

Field:	*Contents*
RINT	Moment of inertia in ton-seconds squared at station ISTN
GDØT*	Initial angular velocity in radians per second at station ISTN
SF*	Initial shear force in tons at station ISTN

Example: ISTN = Station 0

KAG = 13.3987×10^5 tons

I_{mz} = RINT = 0.02686×10^3 ton-s^2

$\dot{\gamma}$ = GDØT = 0 (Initial condition)

V = SF = 0 (Initial condition)

1	10	20	30	40	50	60	70	80
STATIØN	0	1339870.	26.86					

INITIAL - Initial Conditions Card

(Initial conditions cards are the punched final conditions of a previous run.)

1	10	20	30	50	70	80	
INITIAL	1	ignored	TSTRT	ignored			see INTGRTN card
INITIAL	2	HSTN	YDØT	BDM			see HALF-STTN card
INITIAL	3	ISTN	GDØT	SF			see STATIØN card
INITIAL	4	ISTN	YH	ignored			see HYDRØ card

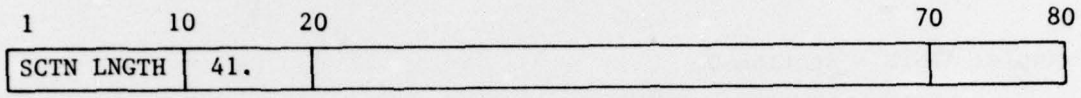
SCTN LNGTH - Section Length Card

1	10	20	30	40	50	60	70	80
SCTN LNGTH	DXI	ignored						

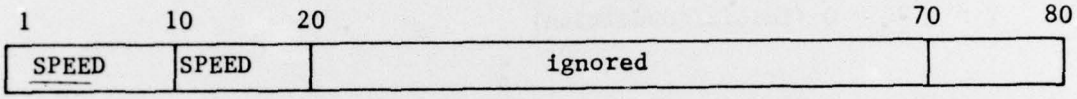
*Ignored if appropriate INITIAL cards are present in input deck.

Field:	*Contents*
DXI	Distance between adjacent stations in feet

Example: ΔX = DXI = 41 ft

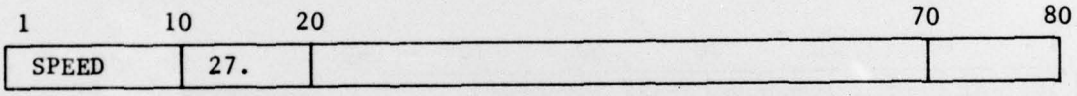


SPEED - Speed Card

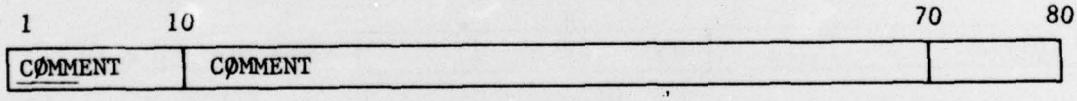


Field:	*Contents*
SPEED	Ship speed in feet per second

Example: U = SPEED = 27 fps



COMMENT - Comment Card



Field:	*Contents*
CØMMENT	Commentary material inserted into echo listing of input deck. The CØMMENT cards are otherwise ignored.

Example:

1	10	70	80
CØMMENT	ESSEX DISCRETE WAVE TRAIN		

ΔΔΔ4. Option Selection Data

CØNTRØL - Control Card

1	10	20	30	40	50	60	70	80
CØNTRØL	ignored	$CNTRL_2$	ignored	$CNTRL_3$	ignored	$CNTRL_4$		

Field:	*Contents*
$CNTRL_2$	Punch YES if having final conditions punched
$CNTRL_3$	Punch YES if including nonlinear hydrodynamic force
$CNTRL_4$	Punch YES if including hydrodynamic damping

Example:

1	10	20	30	40	50	60	70	80
CØNTRØL	PUNCH=	YES	NL HYDRØ=	YES	HYD DAMP=	YES		

DEBUG - Debug Card

1	10	20	30	70	80
DEBUG	IBEG	IEND	ignored		

Field:	*Contents*
IBEG	Intermediate calculations will be put out between derivative evaluations number IBEG
IEND	IEND

Example: In the subroutine KUTMER, HCX is the smallest step size used in the integration. For debugging, it has been decided to print,

first, two steps for examination; then, IBEG=1 and IEND=2 and debug cards will be as follows

1	10	20	30	70	80
DEBUG	1	2			

INTGRTN - Integration Card

1	10	20	30	40	50	60	70	80
INTGRTN	HMNI	TSTRT	TF	TCPU	ignored		+aaa	
+aaa	YHE	YDØTE	GDØTE	BDME	SFE	ignored	+bbb	
+bbb	YHAB	YDØTAB	GDØTAB	BDMAB	SFAB	ignored		

Field: *Contents*

HMNI — Maximum internal step size

TSTRT — Start of simulation in simulation seconds. This is ignored if an appropriate INITIAL card is present in input deck.

TF — End of simulation in simulation seconds

TCPU — Job time limit in CPU seconds

Max Relative Error in Units Per Unit	Max Absolute Error in Unit	Unit	Dependent Variable
YHE	YHAB	feet	Vertical displacement
YDØTE	YDØTAB	feet per second	Vertical velocity
GDØTE	GDØTAB	radians per second	Angular velocity
BDME	BDMAB	foot-tons	Bending moment
SFE	SFAB	tons	Shear force

Example:

HMNI = 0.0125 s

TSTRT = 0 s

TF = 2 s

TCPU = 1500 s

YHE = 0.1 ft

YHAB = 1 ft

YDØTE = 0.01 ft/s

YDØTAB = 1 ft/s

GDØTE = 0.01 rad/s

GDØTAB = 0.2 rad/s

BDME = 0.2 ft-ton

BDMAB = 2000 ft-tons

SFE = 0.05 ton

SFAB = 300 tons

1	10	20	30	40	50	60	70	80
INTGRTN	.0125	0.	2.	1500.			+INT1	
+INT1	.1	.01	.01	.2	.05		+INT2	
+INT2	1.	1.	.2	2000.	300.			

ΔΔΔ5. Output Description Data

AXES - Axes Card for Plotting

1	10	20	30	40	50	60	70	80
AXES	P_ℓ	P_u	B_ℓ	B_u	W_ℓ	W_u		

Axis Lower Limit	Axis Upper Limit	Unit	Plotted Variable
P_ℓ	P_u	degree	Pitch
B_ℓ	B_u	foot-ton	Midship bending moment
W_ℓ	W_u	feet	Wave height at bow

Example: $P_\ell = -3$ deg $\quad P_u = 3$ deg
$B_\ell = -6(10)^5$ ft-ton $\quad B_u = 6(10)^5$ ft-ton
$W_\ell = -12$ ft $\quad W_u = 12$ ft

PLØT - Plot Card

1	10	20	30	40	50	60	70	80
PLØT	NAME	CØDE	PHØNE	W_1	R	ignored		

Field: *Contents*

NAME, CØDE, PHONE: PLOT IDENTIFICATION

Field: *Contents*

W_1: Time in simulation seconds between plotted points
R: Number of simulation seconds of data per frame

Example: Name = Schroeder
Code = 1844
Phone = 71426
W_1 = 0.1 s between plotted points
R = 60 s per frame

1	10	20	30	40	50	60	70	80
PLØT	SCHROEDER	1844	71426	0.1	60.			

PRINT - Print Card

1	10		70	80
PRINT	TIMES			

Field: *Contents*

TIMES — Print times in simulation seconds expressed in free-field "bias width" notation. For example, 0(.1) 1(.5) 5(.1) 6$ represents 0, 0.1, 0.2, . . . 0.9, 1, 1.5, 2, 2.5, . . ., 4.5, 5, 5.1, 5.2, . . ., 5.9, 6, and 6$ means that 6 is last simulated second.

Example: Print results every second from 0 to 90 simulation seconds.

1	10		70	80
PRINT	0. (1.) 90. $			

ΔΔΔ6. Sea Dynamics Data

SEA-GEN - Sea Generator Card (Case 1 - Point application of sinusoidal standing wave to determine ship vibratory frequency.)

1	10	20	30	70	80
SEA-GEN	FREQ	WVHGH	ignored		

Field: *Contents*

FREQ — Wave frequency in hertz

WVHGH — Wave height in feet

SEA-GEN - Sea Generator Card (Case 2 - Sinusoidal sea wave application with wave velocity C = g/ω to determine ship RAO's.)

1	10	20	30	70	80
SEA-GEN	FREQ	WVHGH			

SEA-DATA - Sea Generator Card (Case 3 - Discrete wave train; at present, program is written for ESSEX only; it will be necessary to revise the program for sea trial data of other ships.)

1	10	20	30	40	50	60	70	80
SEA-DATA	OMEGA(1)	OMEGA(2)	OMEGA(3)	TAU(1)	TAU(2)	TAU(3)		

Example: ω_1 = OMEGA (1) = 1.08470672 rad/s
ω_2 = OMEGA (2) = 0.598444 rad/s
ω_3 = OMEGA (3) = 0.804804 rad/s
τ_1 = TAU (1) = 4.11238748 rad
τ_2 = TAU (2) = -21.97272 rad
τ_3 = TAU (3) = -29.54952 rad

1	10	20	30	40	50	60	70	80
SEA-DATA	1.08470672	.598444	.804804	4.11238748	-21.97272	-29.54952		

REFERENCES

1. Andrews, J.N. and S.-L. Chuang, "Seaworthiness Analog Computer," David Taylor Model Basin Report 1829 (Aug 1965).

2. Jasper, N.H. and J.T. Birmingham, "Strains and Motions of USS ESSEX (CVA-9) during Storms Near Cape Horn," David Taylor Model Basin Report 1216 (Aug 1958).

3. Andrews, J.N., "A Method for Computing the Response of a Ship to a Transient Force," David Taylor Model Basin Report 1544 (Nov 1963).

4. McGoldrick, R.T., "Ship Vibration," David Taylor Model Basin Report 1451; Figure 3-8, p. 3-18; Figures A-2 and A-3, p. A-3; Table 8-2, p. 8-10 (Dec 1960).

5. Chuang, S.-L. et al., "Experimental Investigation of Catamaran Cross-Structure Slamming," NSRDC Report 4653 (Sep 1975).

6. Henderson, F., "Transient Response Calculation in the Frequency Domain with General Bending Response Program (GBRP)," NSRDC Report 3613 (Feb 1971).

7. Kaplan, P. et al., "An Investigation of the Utility of Computer Simulation to Predict Ship Structural Response in Waves," Ship Structure Committee Report SSC-197 (Jun 1969).

8. Bruck, H.A., "Procedure for Calculating Vibration Parameters of Surface Ships," NSRDC Report 2875 (Oct 1968).

9. Carnahan, B. et al., "Applied Numerical Methods," Chapter 1, John Wiley & Son Inc., New York (1969).

10. Grim, O., "Berechnung der durch Schwingungen eines Schiffskörpers erzeugten hydrodynamischen Kräfte," Jahrbuch der Schiffbautechnischen Gesellschaft, Vol. 47 (1953).

11. Golovato, P., "A Study of the Forces and Moments on a Heaving Surface Ship," David Taylor Model Basin Report 1074 (Sep 1957).

INITIAL DISTRIBUTION

Copies		
7	CHONR	
	1	ONR 400R
	1	ONR 400R1
	1	ONR 430B
	1	ONR 432
	1	ONR 436
	1	ONR 474
	1	ONR 438
1	CHONR, London	
2	NAVMAT	
	1	MAT 03T2
	1	MAT 034
3	NRL	
	1	2620
	1	4000
	1	8400
1	USNA	
1	USNAVPGSCHOL	
1	NWC, China Lake	
10	NAVSEA	
	1	SEA 00G
	1	SEA 012
	1	SEA 03
	1	SEA 031
	1	SEA 032
	1	SEA 0322
	1	SEA 035 G. Sorkin
	1	SEA 0351 C. Pohler
	2	SEA 09G3
3	NAVAIR	
	1	5301
	1	5302
	1	520
2	NAVFAC	
	1	FAC 03
	1	FAC 04

Copies		
18	NAVSEC	
	1	SEC 6105F D. Lewis
	1	SEC 6110
	1	SEC 6113
	1	SEC 6113B5 J. Berner
	1	SEC 6114
	2	SEC 6114D E. Aronne
		T. Wickert
	1	SEC 6120
	1	SEC 6120D
	1	SEC 6122
	4	SEC 6128 S. Arntson
		J. O'Brien
	4	SEC 6136 R. Johnson
		E. Comstock
		D. Byers
12	DDC	
2	USCG, Washington	
	1	SSC W. Benkert
1	Lib of Congress	
1	MARAD	
2	COMSC	
	1	M4E
	1	M4E4
1	Nat'l Sci Foundation	
2	Univ. of Calif, Dept of NA	
	1	Library
	1	Mr. O. J. Sibul
1	Catholic Univ.	
	1	Prof. M. Soteriades
1	MIT, Dr. A.H. Keil	
1	MIT, Dept. Ocean Engr./ Dr. N. Jones	
2	Univ. of Michigan, Dept of NAME	
	1	Prof. H. Benford
	1	Prof. F. Ogilvie

PRECEDING PAGE BLANK-NOT FILMED

Copies	
2	SIT, Davidson Lab 1 Mr. John Dalzell 1 Library
1	SWRI
1	Web Institute, E.V. Lewis
3	SNAME 1 Slamming Panel 1 Hull Loading Panel
3	ABS 1 Dr. A. Elbatouti 1 Mr. Stiansen 1 Library
1	Design Assoc., Inc./ Mr. M. Kawasaki
1	GIANNOTTI & BUCK ASSOC., INC./ Dr. J. G. Giannotti
1	J. G. ENGR. Res Assoc./ Dr. J. E. Greenspan
1	Dr. Norman H. Jasper
1	Nashville Bridge Co./ Mr. K. E. Hoffmeister
1	National Maritime Res Cen/ Dr. W. M. Maclean
1	Oceanics, Inc./ Mr. P. Kaplan
1	Wheeler Industries, Inc./ Mr. Edgar D. Hoyt

CENTER DISTRIBUTION

Copies	Code
1	15
1	1506
1	152
1	154

Copies	Code	
1	156	
1	1568	
1	1572	
1	17	
1	172	
1	173	
1	1730.3	
1	1730.3	J. Andrews
30	1730.3	S. Chuang
1	1730.5	
1	174	
1	177	
1	178	
1	18	
1	1809	
1	184	
1	1805	E. Cuthill
1	1844	S. Dhir
5	1844	E. Schroeder
5	1844	S. Wybraniec
4	1962	G. Antonides N. Huzil A. Kukk A. Hagen
30	5214.1	Reports Distribution
1	522.1	Library C
1	522.2	Library A