

(1)

Carderock Division Naval Surface Warfare Center

Bethesda, MD 20084-5000

AD-A282 341

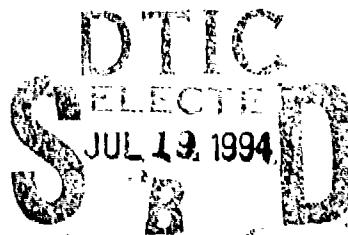


CRDKNSWC/HD-0936-07 June 1994

Hydromechanics Directorate
Directorate Report

SMP93-PC: STANDARD SHIP MOTION PROGRAM FOR PERSONAL COMPUTER WITH SMALL BOAT CAPABILITY

by
T. C. Smith
W. G. Meyers



201P
94-22485



CRDKNSWC/HD-0936-07 SMP93-PC: Standard Ship Motion Program for Personal Computer with Small Boat Capability



Approved for public release, distribution unlimited.

94 7 18 023

MAJOR DTRC TECHNICAL COMPONENTS

- CODE 011 DIRECTOR OF TECHNOLOGY, PLANS AND ASSESSMENT
12 SHIP SYSTEMS INTEGRATION DEPARTMENT
14 SHIP ELECTROMAGNETIC SIGNATURES DEPARTMENT
15 SHIP HYDROMECHANICS DEPARTMENT
16 AVIATION DEPARTMENT
17 SHIP STRUCTURES AND PROTECTION DEPARTMENT
18 COMPUTATION, MATHEMATICS & LOGISTICS DEPARTMENT
19 SHIP ACOUSTICS DEPARTMENT
27 PROPULSION AND AUXILIARY SYSTEMS DEPARTMENT
28 SHIP MATERIALS ENGINEERING DEPARTMENT

DTRC ISSUES THREE TYPES OF REPORTS:

1. **DTRC reports, a formal series**, contain information of permanent technical value. They carry a consecutive numerical identification regardless of their classification or the originating department.
2. **Departmental reports, a semiformal series**, contain information of a preliminary, temporary, or proprietary nature or of limited interest or significance. They carry a departmental alphanumerical identification.
3. **Technical memoranda, an informal series**, contain technical documentation of limited use and interest. They are primarily working papers intended for internal use. They carry an identifying number which indicates their type and the numerical code of the originating department. Any distribution outside DTRC must be approved by the head of the originating department on a case-by-case basis.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

CONTENTS

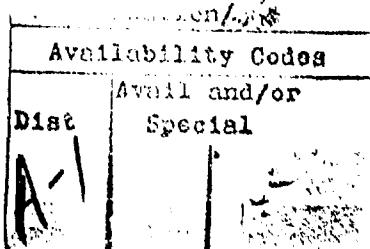
	Page
ABSTRACT	1
ADMINISTRATIVE INFORMATION	1
INTRODUCTION	1
PC ASPECT OF SMP93	3
RUNNING SMP93-PC	4
Running SMP93-PC stand alone	5
Running SMP93-PC using PREDICT	7
UPDATES TO THEORY	7
SMALL BOAT FREQUENCY RANGES	7
HORIZONTAL FORCE ESTIMATOR	10
SMP PROGRAM CHANGES	11
INPUT	11
ORG File Start and Stop Option	12
Horizontal Force Estimator	12
OUTPUT	12
ACKNOWLEDGEMENTS	13
APPENDIX A: LINKING SMP93-PC USING OVERLAYS	19
APPENDIX B: SMP93-PC SOURCE CODE LISTING	21
REFERENCES	201

FIGURES

	Page
1. Directory structure for SMP93-PC.	6
2. Example SMPSYS.TEX file.	6
3. PREDICT applications manager organizational structure.	8
4. Wave frequency distribution for SMP-93 frequency sets.	10

TABLES

	Page
1. Subroutine new to SMP93-PC and their function.	4
2. PREDICT sub-program descriptions.	9
3. SMP93-PC wave frequency set summary.	11
4. Example SMP93-PC input deck.	14
5. Example of new parts of SMP93-PC output file.	16
6. Description of possible SMP output files.	18
7. PCLINK overlay instructions for SMP93-PC.	19



ABSTRACT

The Standard Ship Motion Program, SMP, was developed at the Carderock Division, Naval Surface Warfare Center, and documented in 1981 as a prediction tool for use in the Navy's ship design process. SMP provides predictions of the response of a ship advancing at constant forward speed with arbitrary heading in both regular and irregular seas. In 1984, a number of corrections and enhancements were finished and documented.

In 1987, the Naval Sea System Command installed an undocumented FORTRAN 77 version on their Digital VAX hardware. From there SMP migrated to the personal computer. Additional features and improvements have been made. Predictions of horizontal force estimator were incorporated into SMP. An option was added to reduce run time and minimize unwanted output. Extra wave frequency sets designed for small boats were also added.

ADMINISTRATIVE INFORMATION

The SMP93-PC updates were performed at the Carderock Division, Naval Surface Warfare Center, (CARDEROCKDIV) over the years 1991 to 1993. This documentation was funded by the United States Coast Guard (USCG) through CARDEROCKDIV work unit numbers 1-1561-059-01 and 1-5610-353-01. The fund code for this task is 28693 and the USCG authorization reference is DTCG23-93-F-AWP003.

INTRODUCTION

The Standard Ship Motion Program, SMP¹, was developed over a number of years at the Carderock Division, Naval Surface Warfare Center, (CARDEROCKDIV)[†] and finally documented in 1981. This program provides a standard ship motion prediction tool for use in the Navy's ship design process. It was enhanced and updated in 1984². These two early versions, designated SMP81 and SMP84, ran on Carderock's mainframe computer, a CDC CYBER 6600 (FORTRAN IV) and later on a VAX (FORTRAN 77).

The early PC version of SMP, developed in early 1988, was based on the VAX (FORTRAN 77) version of SMP84, maintained by Naval Sea System Command (NAVSEA) Code 55W3 and designated unofficially as SMP87. It should be noted that this VAX

¹Then named David Taylor Naval Ship Research and Development Center (DTNSRDC)

version of SMP84 was altered by NAVSEA staff to "improve" the wave frequency ranges selected automatically for ships with shorter roll periods than those normally associated with carriers or destroyers and frigates and thus became SMP87. SMP87 was used as normal practice by NAVSEA; CARDEROCKDIV; and other US Government agencies in ship design and related studies.

The deployment of an intial version of SMP87-PC on the USS Constellation in the Indian Ocean clearly brought out the need to simplify the SMP input interface for the user. As a result, a PC user interface, PREDICT³, was developed to run SMP.

It is necessary when using SMP to define both the base range and distribution of wave frequencies for the computed transfer functions. All other required transfer function values at frequencies of the encountered waves are obtained from this basic set by interpolation. SMP relieves the user of the chore of providing this wave frequency information by automatically selecting a suitable range of these wave frequencies. This choice is made on the basis of the natural roll frequency from a set of frequency ranges "built into" the program.

The early versions of SMP, SMP81 and SMP84, contained just two built in wave frequency ranges. The transfer functions at the base frequencies must be calculated with a fine enough resolution to permit a good definition of the narrow banded roll response.

In 1991, two studies of a USCG Buoy Tender (WPB), identified the need for extra frequency sets that provide adequate resolution for the transfer functions, especially for roll. The third wave frequency set, added by NAVSEA, resulted in numerical instabilities of the responses, particularly those related strongly to roll, for these smaller ships. These instabilities were illustrated by erratic variations of the root mean square (RMS) responses with consecutive heading values or ship speed. The modification of the third wave frequency range to account for the responses of the much smaller Buoy Tender (WPB) forced a revision of the PC based SMP87 and was designated SMP91-PC (dated 4/21/91).

Later work with the 1993 USCG Seakeeping Criterion Definition Program resulted in the addition of a fourth wave frequency range. The resulting program designated

SMP93-PC now contains the four ranges of wave frequencies where the most recent addition was the one required for small boats (47 - 110 feet).

The collective set of updates to SMP84, that result in SMP93-PC, are designated as the SMP93-PC updates. They include:

1. Changes in FORTRAN source code due to change from CDC CYBER to VAX.
2. Changes in FORTRAN source code due to change from VAX to PC compiler.
3. Addition of Origin transfer function (ORG) file only run option.
4. Addition of horizontal force estimator response⁴
5. Two extra wave frequency sets with frequency resolution appropriate for ships and boats with short roll periods.

This report deals just with the updates to the SMP84 version² that result in SMP93-PC. The theory for the predictions are documented in References 1 and 2, and will not be repeated here. The changes to the input description are minimal and only the changes since SMP84 will be dealt with in this report. References 1 and 2 fully document the input "deck" description for SMP84.

PC ASPECT OF SMP93

The PC version of SMP is coded in Lahey FORTRAN 77 and requires a math co-processor to run. Appendix A describes linking SMP93-PC using overlays. Though the code has not been converted to a 32 bit compiler, using one would probably avoid overlay linking.

The differences between SMP versions due to FORTRAN compilers deal mainly with the opening and closing of files. Random access files needed restructuring. Also the subroutine CPINTG was made double precision to avoid numerical difficulties due to loss of accuracy from the 60-bit word (CDC CYBER) to the 32-bit word (VAX and PC).

Additionally, four new utility subroutines were added: ELTIME, EXP, RDSMPSYS, and SLENGTH. Table 1 lists the new subroutines and their function. Subroutine SECT has been renamed SECT1, but still performs the same functions.

Table 1. Subroutine new to SMP93-PC and their function.

ELTIME	Calculates elapsed time using DOS functions.
EXP	Avoids underflow with Lahey FORTRAN 77 EXP function.
RDSMPSYS	Read SMPSYS.TEX and sets file names and paths.
SLENGTH	Returns the length of a character string.

Also the common block IO was changed. The new common block is:

```
COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL
      INTEGER      SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL
```

And the common block SMPSYS was added to subroutines: AINPUT, EQMOTN, HSTAT, HYDCAL, INPUT, LRAOOUT, OUTPUT, ROAOUT, RDBASE, READ, REGWAV, RMSOUT, RMSTOE, SEVMOT, SPLNFT, and WAVMAK.

```
COMMON /SMPSYS/ FIS,AS,SIS,SOS,SDS,HALOS,DEV,PRN,SMPPS,SMPIS,
2 SMPDS,SHPTYPS,SHIPS,VARS,CYCLS,TITLES,OPTION,LSIS,LSOS,
2 LSDS,LHALOS,LDEV,LPRN,LSMPPS,LSMPIS,LSMPOS,LSMPDS,LSHPTYPS,
2 LSHIPS,LTITLES
CHARACTER*160 AS
CHARACTER*80 FIS,SIS,SOS,SDS,TITLES
CHARACTER*20 HALOS,DEV,PRN,SMPPS,SMPIS,SMPOS,SMPDS,SHPTYPS
CHARACTER SHIPS*6,VARS*2,CYCLS*2
INTEGER*2 OPTION
```

Rather than listing all the source code differences individually, Appendix B has a complete listing of the SMP93-PC version source code. Though the PREDICT user's manual³ has a listing of an early version of SMP93-PC source code and a brief description of the changes, this report is more complete, and should be used as the definitive reference for SMP93-PC.

RUNNING SMP93-PC

SMP93-PC can be run as either a stand-alone program or using a user interface, such as PREDICT³.

Running SMP93-PC stand alone

The steps for running SMP93-PC as a stand alone program, assuming a directory structure is in place, are:

1. Change directory to SMP input directory and make changes to SMP input file if needed.
2. Change directory to SMP executable directory.
3. Update SMPSYS.TEX if needed.
4. Run SMP93-PC executable.

To run SMP93-PC as a stand alone program requires a specific directory structure, file location, and naming convention. When SMP93-PC runs, it reads a control file, SMPSYS.TEX, for the names and directory paths of the input and output. SMPSYS.TEX must be named SMPSYS.TEX and must be in the same directory as the SMP93-PC executable. See Figure 1 for an example directory structure. Figure 2 shows an example SMPSYS.TEX file that corresponds to the directory tree in Figure 1.

SMP93-PC generates file names and paths using the information in SMPSYS.TEX. To change file names and path, edit SMPSYS.TEX and change only the data after the equals (=) sign.

The directory tree and naming convention seen in Figure 1 are briefly described next.

Directory structure The directory/sub-directory structure is mandatory, though their names can be any valid DOS name. The directory names must match the paths given in SMPSYS.TEX. SHIP TYPE, e.g. DESTROYR, is a sub-directory of the directories named in the SMP INPUT PATH, e.g. SMPINPUT, and SMP OUTPUT PATH, e.g. SMPOUTPT. CURRENT SHIP, e.g. DD965, is used as the basis for the file names and as a subdirectory of SMP DATA PATH, e.g. SMPDATA.

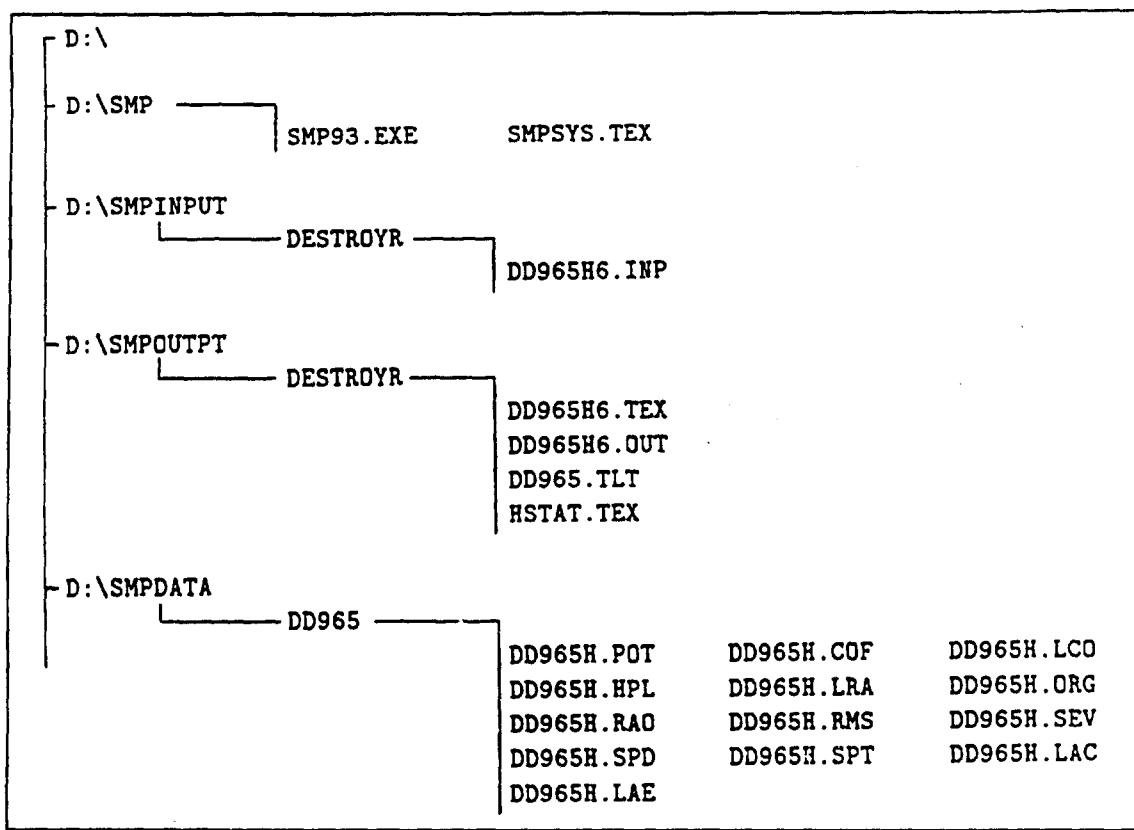


Fig. 1. Directory structure for SMP93-PC.

```

HALO PROGRAM PATH=C:\HPRO_FOR
HALO GRAPHICS SCREEN DRIVER=IBME
HALO PRINTER DRIVER=EPSN
SMP PROGRAM PATH=D:\SMP
SMP INPUT PATH=D:\SMPINPUT
SMP OUTPUT PATH=D:\SMPOUTPUT
SMP DATA PATH=D:\SMPDATA
SHIP TYPE=DESTROYR
CURRENT SHIP=DD965
VARIANT=H
CYCLE=6
TITLE= from HFDS T=39.7 92-12-11
OPTION=2
  
```

Fig. 2. Example SMPSYS.TEX file.

Naming convention: The SMP input file is the concatenation of the CURRENT SHIP, VARIANT, and CYCLE variables with INP for an extension, e.g. DD965H6.INP. The CURRENT SHIP variable, e.g. DD965, has a maximum length of five characters. The variant typically indicates major changes to the input file and the cycle keeps track of the number of times the file has been changed.

The SMP output files are the concatenation of CURRENT SHIP and VARIANT, e.g. DD965H with the appropriate extension. The files and their extension are discussed in the **OUTPUT** section.

Running SMP93-PC using PREDICT

PREDICT³ is a menu driven shell that allows the user to choose, view, and edit input SMP files, run SMP93, select output files to save, and make polar plots of the responses, or plots of the hull form. PREDICT uses the same directory structure and naming convention used when running SMP93-PC as a stand-alone program, Figure 1. The PREDICT user's manual³ describes the file structure and naming convention fully. PREDICT also provides the option to continue and generate time history data using Simulation Time History and Access Time History programs⁵. Figure 3 shows the overview of PREDICT and where SMP93-PC fits into it. Table 2 gives a brief description of the different parts of the PREDICT package.

UPDATES TO THEORY

The ship motion theory used for the predictions is the same for the PC version as for the VAX and CDC versions. It is assumed the user is already familiar with the ship motion theory, variables, coordinate system, files, and input/output schemes that are described in the SMP User's Manual.^{1, 2} These details will not be repeated here.

SMALL BOAT FREQUENCY RANGES

The original two wave frequency sets have frequency resolution in the range most applicable to carriers, destroyers, and frigates. The original third frequency set had numerical instabilities with ships smaller than frigate size, with natural roll periods less than nine seconds. This was especially noted in responses that had a roll component.

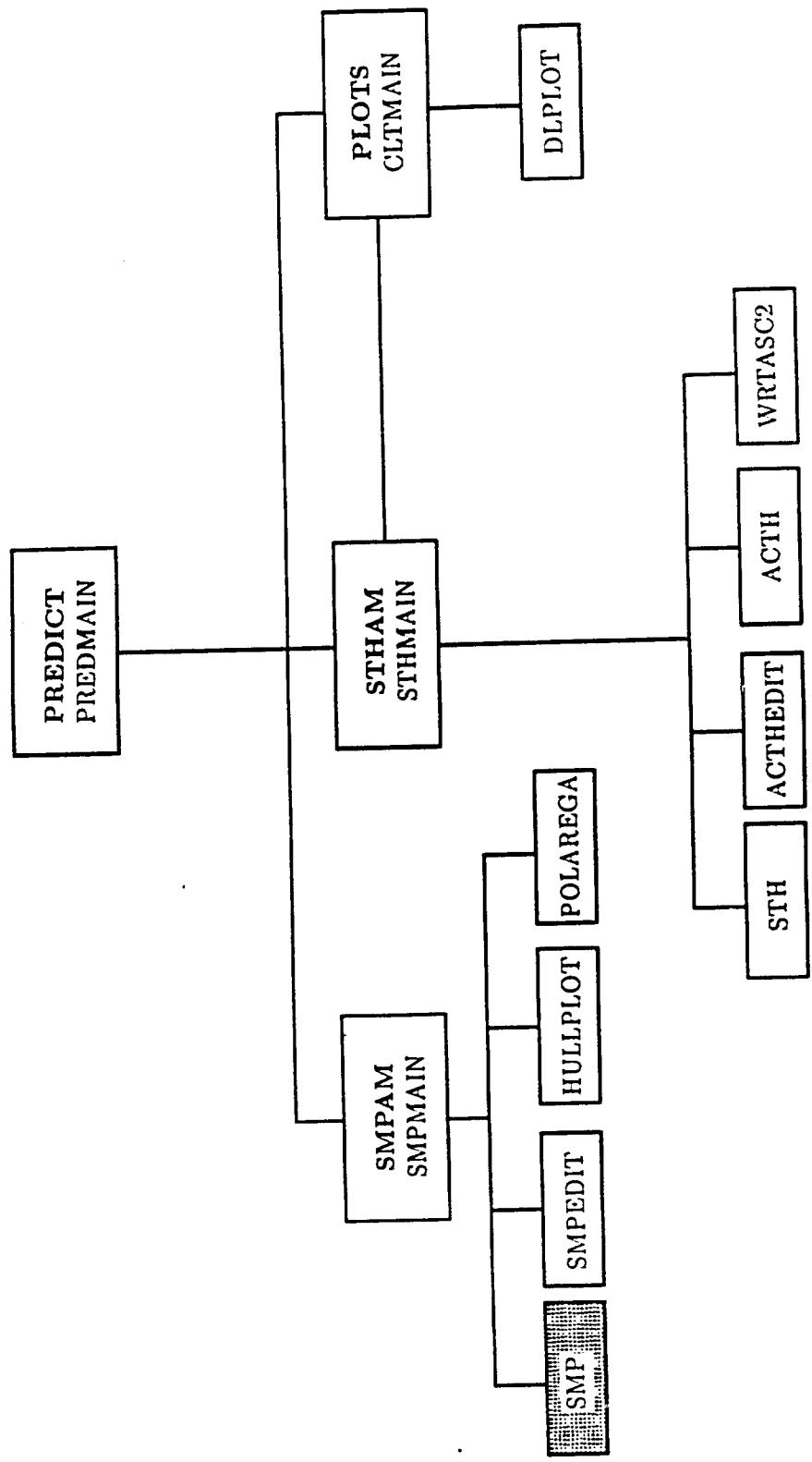


Fig. 3. PREDICT applications manager organizational structure.

Table 2. PREDICT sub-program descriptions.

Program	Description
PREDMAIN	The top level menu and link between the frequency, time domain, and plotting branches of PREDICT.
SMPMAIN	The main menu for the frequency domain branch of PREDICT.
SMP	Frequency domain predictions of ship motion. Uses most recent version.
SMPEDIT	Editor for SMP input files that keeps track of file naming convention and correct format fields for data.
HULLPLOT	Generates plots of the hull form.
POLAREGA	Generates speed polar plots of the ship motion response.
STHMAIN	The main menu for the time domain branch of PREDICT
STH	Time domain predictions of six degree of freedom response at ship center of gravity.
ACTH	Time domain predictions of absolute and/or relative point motion.
ACTHEDIT	Editor for ACTH input files.
WRTASC2	Converts ACTH binary format output to ASCII format.
CLTMAIN	The main menu for the plotting branch of PREDICT.
DLPLOT	Plots time histories of ACTH predictions.

numerical instabilities with ships smaller than frigate size, with natural roll periods less than nine seconds. This was especially noted in responses that had a roll component. The original two wave frequency sets would have had similar numerical instabilities had they been used for smaller ships.

The cause of the numerical instabilities in the root mean square (RMS) responses as a function of ship speed and heading were traced to an inadequacy of the defined roll transfer function frequency range. With only 30 wave frequencies per set, it is important to have more wave frequencies grouped near the natural roll frequency. With small boats this also means increasing the range of modal wave periods to span the shorter response periods.

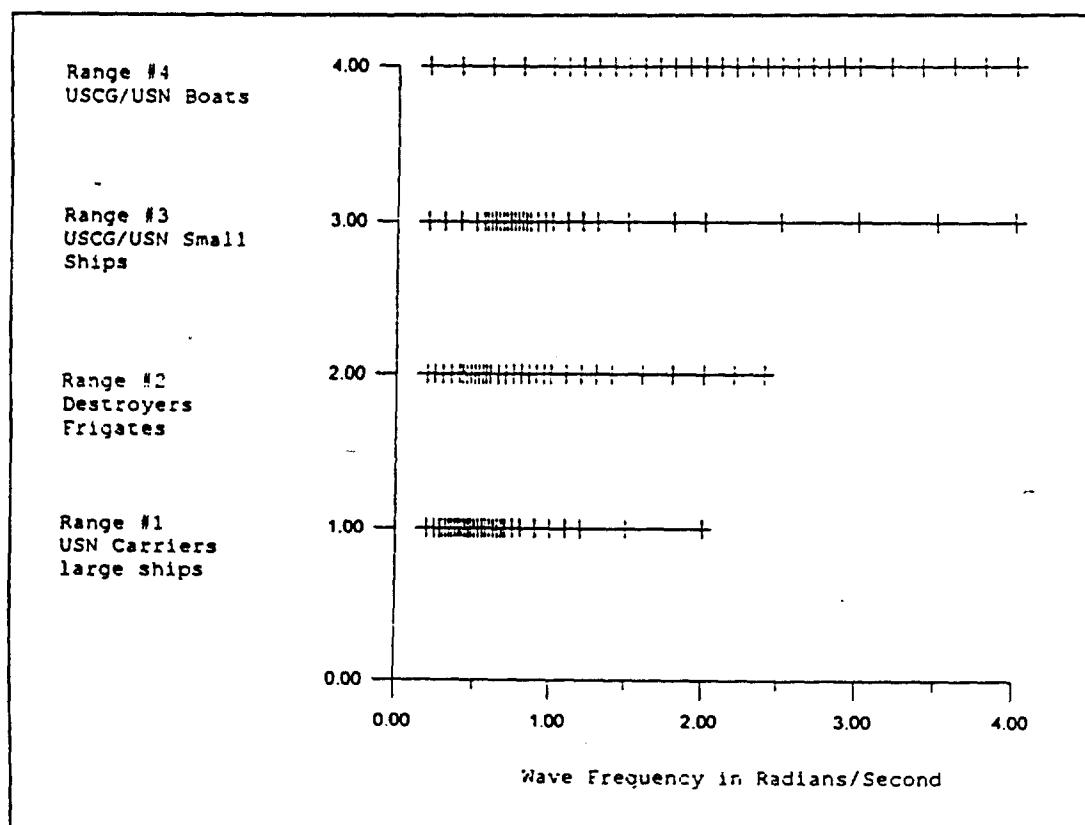


Fig. 4. Wave frequency distribution for SMP-93 frequency sets.

As a result SMP93-PC uses four wave frequency sets. The last two have increased resolution in the frequency range most applicable to small boats. The new wave frequency sets, ranges # 3 and # 4, are used when the roll period is less than or equal

Table 3. SMP93-PC wave frequency set summary.

Range	Type of Ship/Boat	Roll Period Seconds	Wave Periods Seconds	Maximum Resolution Range Seconds
#1	Carriers/large ships	$T_\phi > 15$	3.14 – 31.4	12.56 – 22.43
#2	Frigates/Destroyers	$9 < T_\phi < 15$	2.62 – 31.4	10.47 – 15.70
#3	USCG/USN small ships	$5 < T_\phi < 9$	1.57 – 31.4	6.28 – 12.56
#4	USCG boats	$T_\phi < 5$	1.57 – 31.4	2.09 – 6.28

The two new wave frequency sets, FREQ3 and FREQ4, were added in subroutine READ. FREQ3 is: 0.2, 0.3, 0.4, 0.5, 0.55, 0.575, 0.6, 0.625, 0.65, 0.675, 0.7, 0.725, 0.75, 0.775, 0.8, 0.825, 0.85, 0.875, 0.9, 0.95, 1.0, 1.1, 1.2, 1.3, 1.5, 1.8, 2.0, 2.5, 3.0, 3.5, and 4.0. FREQ4 is: 0.2, 0.4, 0.6, 0.8, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 3.0, 3.2, 3.4, 3.6, 3.8, and 4.0.

HORIZONTAL FORCE ESTIMATOR

SMP93-PC has the capability to estimate the horizontal force at the points defined in the Motions at a Point Data Card Set. The horizontal force estimator (HFE) is the estimated ship-referenced acceleration in the horizontal plane⁴. It is a combination of the earth referenced lateral acceleration and the horizontal component of gravitational acceleration due to roll (heel). Thus, the horizontal forces applied to people and equipment on the ship by the motions are now also predicted. Horizontal Force Estimator is defined by:

$$HFE = \frac{-\omega_e}{g}(\zeta_2 - Z\zeta_4 + X\zeta_6) + \zeta_4 \quad (1)$$

Where ζ is a transfer function with sub-scripts 2, 4, and 6 referring to sway, roll, and yaw respectively. X and Z are the x and z coordinates of the point locations, see Figure C-1 of Reference 1 for a coordinate system diagram; ω_e is the wave encounter frequency; and g is a gravitational constant.

SMP PROGRAM CHANGES

INPUT

The input for SMP consists of underwater hull form shape, ship weight distribution, appendage details, specific shipboard locations, and wave data. This input is broken down into 15 Data Card Sets which are described in Appendix C of Reference 1. The modification to start or stop after generating an origin transfer function (ORG) file required changes to Data Card Set 2 (Program Options). The addition of horizontal force estimator (HFE) required changes to Data Card Set 12, card 1. See Table 4 for a sample input file. The changes to the two data sets for these new options are described below:

ORG File Start and Stop Option

With this option activated, SMP93-PC can either start using an existing ORG file or stop after generating the ORG file. This provides a time saving if only the ORG file is wanted. The major run option, OPTN, must be either 4 or 5 to start using an existing ORG file. The ORG file contains the transfer functions of the ship's six degrees-of-freedom about the center of gravity.

The flag, ORGOPTN, was added to Data Card Set 2, Program Options, as the seventh variable (integer, column 40). Possible values are:

- 0 or blank = Normal run.
- 1 = Stop execution after generating ORG file. Do not perform statistical calculations.
- 2 = Start execution using an existing ORG file (OPTN=4 or 5). Read ORG file and perform statistical calculations.

Horizontal Force Estimator

SMP93-PC has the capability to estimate the horizontal force at the points defined in the Motions at a Point Data Card Set 12. To turn this feature on, set column 10 of Data Card Set 12, card 1, to 1.

- 0 or blank = No horizontal force estimate calculations or output.
- 1 = Horizontal force estimates for motions at point locations.

OUTPUT

The changes to the output file (OUT) are minimal. Horizontal force estimates have the same format as other response output. The change in wave frequency ranges also changes the range of modal periods used for irregular seas calculations. The modal periods for the new wave frequency sets, FREQ3 and FREQ4, are: 3, 5, 7, 9, 11, 13, 15, and 17. See Table 5 for an example of the output affected by the input changes. Note, the example assumes that the user already has an ORG file, otherwise SMP93-PC will stop before performing irregular seas calculations and there would be no HFE response tables.

The main difference between the VAX and PC versions is the treatment of the output files. Table 6 gives a list of the extensions used by the SMP93-PC and how they compare with the SMP87 version. The SPL file in the SMP87 version either has spline fits of the body plan or response data for speed polar plots depending on main run option. SMP93-PC writes body plan spline fits to the HPL file and splits the response data found in the SPL file into two files, SPD and SPT, to avoid memory problems. The SPD file has RMS response and modal period data in a binary format. The SPT file has speed, heading, and response name information in an ASCII format.

The PC output files are written in sub-directories in the SMP DATA PATH and SMP OUTPUT PATH directories from the SMPSYS.TEX file. Only the output file (OUT), run log (TEX), and hydrostatic output (HSTAT.TEX) are in the SMP OUTPUT PATH sub-directories. All the rest are in the SHIP TYPE sub-directory of SMP DATA PATH directory, e.g. SMPDATA\DESTROYR.

ACKNOWLEDGEMENTS

The authors would like to thank Valerie Scott, Dana Gentile, and Claude Williams of ORI for their early work on this subject. Ms. Valerie Scott did the initial conversion of SMP87 to the PC requiring tedious labor with the FORTRAN overlays and waiting for test calculations to finish on the early PCs. This group also produced SMPEDIT, a precursor to, and now part of, PREDICT.

Table 4. Example SMP93-PC input deck.

WPB 8236 82' PATROL BOAT C CLASS FL 5/18/93 PROP STRUTS INPUT AS PASSIVE RUDDER										
2	0	0	1	0	0	1				
FEET		1.9905	32.17250	0.0001279						
78.0000	15.7800	5.9500	72.30	8.0000	8.0000	0.0000				
3.2400	0.0000	7.2100	0.2500	0.3700	0.2500					
21	0									
0.0000	1	0								
0.0000	0.00									
0.0000	5.95									
1.0000	5	0								
1.0000	0.00	0.35	0.71	0.98	1.21					
1.0000	2.23	3.00	4.00	5.00	5.95					
2.0000	5	0								
2.0000	0.00	0.97	1.54	1.96	2.33					
2.0000	2.00	3.00	4.00	5.00	5.95					
3.0000	7	0								
3.0000	0.00	1.00	1.57	2.00	2.39	2.86	3.27			
3.0000	1.83	2.48	3.00	3.45	4.00	5.00	5.95			
4.0000	6	0								
4.0000	0.00	1.00	2.00	3.00	3.80	4.21				
4.0000	1.72	2.24	2.89	3.75	5.00	5.95				
5.0000	7	0								
5.0000	0.00	1.00	2.00	3.00	4.00	4.65	5.07			
5.0000	1.66	2.11	2.59	3.16	4.00	5.00	5.95			
6.0000	8	0								
6.0000	0.00	1.00	2.00	3.00	4.00	4.72	5.44	5.89		
6.0000	1.63	2.00	2.46	2.86	3.39	4.00	5.00	5.95		
7.0000	9	0								
7.0000	0.00	1.00	2.00	3.00	4.00	5.00	5.38	6.15	6.55	
7.0000	1.66	1.96	2.32	2.65	3.10	3.69	4.00	5.00	5.95	
8.0000	9	0								
8.0000	0.00	1.00	2.00	3.00	4.00	5.00	5.96	6.77	7.11	
8.0000	1.68	1.95	2.28	2.61	2.95	3.38	4.00	5.00	5.95	
9.0000	9	0								
9.0000	0.00	1.00	2.00	3.00	4.00	5.00	6.00	7.00	7.53	
9.0000	1.77	1.99	2.28	2.57	2.91	3.19	3.66	4.59	5.95	
10.0000	10	0								
10.0000	0.00	0.17	0.17	0.24	0.52	1.00	3.00	5.00	6.86	7.89
10.0000	1.03	1.03	1.20	1.85	1.93	2.07	2.60	3.15	4.00	5.95
11.0000	10	0								
11.0000	0.00	0.17	0.26	0.30	0.95	2.00	4.00	6.00	7.17	8.05
11.0000	0.92	0.92	1.18	2.08	2.25	2.45	2.92	3.45	4.00	5.95
12.0000	10	0								
12.0000	0.00	0.17	0.28	0.29	1.00	2.00	4.00	6.00	7.28	8.07
12.0000	0.84	0.84	1.47	2.19	2.33	2.59	3.01	3.47	4.00	5.95
13.0000	10	0								
13.0000	0.00	0.17	0.28	0.31	0.75	3.00	5.00	7.20	7.95	8.06
13.0000	0.71	0.71	1.47	2.44	2.54	2.96	3.33	4.00	5.00	5.95

Table 4. Continued.

14.0000	10	0									
14.0000	0.00	0.17	0.27	0.29	0.31	0.75	2.17	5.00	7.42	8.05	
14.0000	0.63	0.63	1.20	2.19	2.65	2.75	3.00	3.47	4.36	5.95	
15.0000	10	0									
15.0000	0.00	0.17	0.26	0.31	0.36	0.70	3.00	5.00	7.39	7.97	
15.0000	0.49	0.49	1.00	2.44	2.88	3.00	3.34	3.64	4.44	5.95	
16.0000	10	0									
16.0000	0.00	0.17	0.26	0.30	0.39	1.00	3.00	5.00	7.32	7.87	
16.0000	0.39	0.39	0.80	2.20	2.98	3.29	3.56	3.82	4.50	5.95	
17.0000	10	0									
17.0000	0.00	0.17	0.26	0.31	0.40	1.00	3.00	5.00	7.26	7.74	
17.0000	0.25	0.25	0.92	2.65	3.40	3.56	3.80	4.03	4.57	5.95	
18.0000	7	0									
18.0000	0.00	1.00	2.00	4.00	6.00	7.20	7.58				
18.0000	3.74	3.83	3.95	4.15	4.34	4.66	5.95				
19.0000	9	0									
19.0000	0.00	1.00	2.00	4.00	5.53	6.72	7.14	7.25	7.43		
19.0000	4.00	4.11	4.21	4.36	4.43	4.56	4.72	5.00	5.95		
20.0000	9	0									
20.0000	0.00	1.00	2.00	4.00	5.58	6.76	7.07	7.11	7.22		
20.0000	4.29	4.41	4.47	4.59	4.62	4.71	4.79	5.00	5.95		
0											
1											
9.2308	17.0000	18.0000	0.0000	1.7824	0.2500	3.7413					
2											
19.2100	19.6600	3.7500	4.3750	4.3750							
19.2100	19.6600	3.7500	1.8750	1.8750							
18.1649	18.4923	3.9565	3.8020	3.9063							
18.3521	18.5724	4.2215	2.0313	1.8906							
0											
0	0	0									
3	1										
1	PILOT HOUSE AT HELMSMAN CHAIR				8.2051	0.0000	20.0000				
2	FWD BERTHING, PORT/TOP BUNK				3.3333	5.0000	12.0000				
3	MAIN DECK, BOAT DECK, STBD RAIL				15.3846	-7.5000	10.5000				
3	0										
1	PORT PROPELLER TIP	2	18.9017	4.1600	3.7500	0.8538					
2	STATION 2, SLAMMING	1	2.0000	0.0000	2.0000	0.8538					
3	MAIN DECK, STBD RAIL	3	15.3846	-7.5000	10.5000	0.8538					
1	2.0000 SIGNIFICANT										
2.6200											

STOP

Table 5. Example of new parts of SMP93-PC output file.

DATA CARD SET 2 - PROGRAM OPTIONS						
OPTION	VLAOPR	RAOPR	RUDMPR	LRAOPR	ADRPR	ORCOPTN
2	0	0	1	0	0	1

DATA CARD SET 12 - MOTIONS AT A POINT						
NPTLOC	HPE					
3		1				

NUMBER	NAME					
1	PILOT HOUSE AT HELMSMAN CHAIR					
2	FWD BERTHING, PORT/TOP BUNK					
3	MAIN DECK, BOAT DECK, STARBOARD RAIL					

Table 5. Continued.

WPB 8236 82' PATROL BOAT C CLASS FL 5/18/93 PROP STRUTS INPUT AS PASSIVE RUDDER

LONGCRESTED																	
SIGNIFICANT WAVE HEIGHT = 2.62 FEET																	
FWD BERTHING, PORT/TOP BUNK			XFP = 3.33			YCL = 5.00			ZBL = 12.00								
HORIZONTAL FORCE ESTIMATOR						(G)											
(ACC. X 100)						RMS VALUE / ENCOUNTERED MODAL PERIOD (TOE)											
SHIP HEADING ANGLE IN DEGREES																	
V		HEAD		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0		15		30		45		60							
0		0															

Table 6. Description of possible SMP output files.

Menu choice	Description	SMP93 Extension	Data Type	FORTRAN Unit	SMP87 VAX Extension
Potential file	Potential flow velocity potential	POT	Binary	2	
Coefficient file	Added mass and damping, excitation	COF	Binary	3	BSC
Load coefficient file	Loads	LCO	Binary	4	
Hull plot file	Spline fit of offsets for HULLPLOT	HPL	ASCII	9	SPL
Input file	Ship input data	INP	ASCII	5	SMP
Load response operator file	Response operators for loads	LRA	Binary	10	
Output file	Hydrostatics, roll damping, and response output	OUT	ASCII	7	OUT
Origin file	Ship origin transfer functions	ORG	Binary	11	ORG
Response operator file	Response amplitude operators	RAO	Binary	12	
RMS file	Response RMS for unit wave height	RMS	Binary	13	RMS
Severe motion file	Worst case response and sea conditions	SEV	Binary	14	
Speed polar data file	RMS response data for speed polar plots	SPD	Binary	15	SPL
Speed polar text file	Labels and titles for speed polar plots	SPT	ASCII	16	SPL
Lateral coefficient file	Frequency domain coefficients for rudder roll stabilization	LAC	Binary	17	
Lateral excitation file	Frequency domain coefficients for rudder roll stabilization	LAE	Binary	18	

APPENDIX A: LINKING SMP93-PC USING OVERLAYS

Due to memory constraints, it is necessary to link SMP93-PC using an overlay loader. The authors used an overlay loader designed for the PC called PCLINK, a third party program. Table 7 has the PCLINK overlay instructions assuming the object files reside in the directory C:\SMPPC\SMP. The current version of SMP93-PC has not been converted to the newer 32 bit compilers. Such a conversion should be straight forward and would probably result in further restructuring of the random access files and not require an overlay loader to link the program.

Table 7. PCLINK overlay instructions for SMP93-PC.

```
OUTPUT C:\SMPPC\SMP
FILE SMP93, ALGRNG, ATAN2D, BMAX, CPFIT, CPINTG, CPLVAL, LRAO, RAOPHA,
      SPINTG, SPFIT, SPLVAL, EXP, ELTIME, SLENTH, RDSMPSYS, PRELIM
OVERLAY F77LCODE, F77LDATA, F77LCOMN
BEG SECT FILE INPUT, CUBCO2, SPINT2, SPLNT2
  BEG SECT FILE READ, AINPUT, GENOFS, BRWVSP, SPLNAR
    SECT FILE HSTAT, NORMAL, VUNIT2, CONIWT, PDER, PADD, RSOLVE, SPPLV2,
          NORMT5, RDCOMP, PMPY, PVAL, PINT, TRIM, SPLNFT
    SECT FILE HSTOUT
  END
  SECT FILE REGWAV, CDCOMP, CSOLVE, EDMKSP, FINTSP, REVAL, SKFRSP
  BEG SECT FILE HYDCAL
    BEG SECT FILE HYD2D, TWODPT, GRNLOG, GRNFRO, ALAG, EXPINT, WTPELM, ATAN3
      SECT FILE T3DAMD, RPHI2D, T2DAMD, AMDPRN
      SECT FILE COFOU, AMD, RDPELM, EXFOR
    END
    SECT FILE RDBASE, RDPRIN, WAVMAK, HLLIFT, RDLIFT, SKLIFT, BKLIIFT, FNLIIFT,
          SKNFRC, RDEDDY, HLEDDY, BKEDDY, FNEDDY, CEVAL, SECT1, TANAKA,
          VISC, SERAB, SERD, SERE, FTWO, FIG56, FIG7, FIG8, FIG10, FIG11,
          CALRGM, BILGEK, CMINR, SBEDDY, SBLIFT
    SECT FILE EQMOTN, LIMIT, SOLVE, CLIP, TRNLAT, RDEVAL, RVSLAT, LSCOF,
          INERST, ACTFIN
  END
  SECT FILE IRGSEA, RMSTOE, WEDEFN, RAOPHS, PRAO, ADRES, ORAO, VELACC, RELMOT,
        RMS, TOE, PSPSC, SCB2, XMSSC, PSPLC, INTRPL, TEPEAK, FNRAO
  SECT FILE OUTPUT, RSTITL, RLITR
  BEG SECT FILE RAOOUT, ORGRAO, TFNFFIT
    SECT FILE LRAOUT
    SECT FILE RMSOUT, RLITER, FETCH, DKWSLM, SETSEV, SEVMOT
  END
END
```


APPENDIX B: SMP93-PC SOURCE CODE LISTING

This appendix is a listing of the source code similar, but not identical, to Appendix 1 of Reference 1.

C SMP93 PROGRAM LIBRARY -

* PROGRAM SMP93

* Standard Ship Motion Program (SMP93)
* for Personal Computers

* Operating system MS-DOS Version 4.01
* FORTRAN 77 using Lahey Fortran
* Overlay linking using FLINK86

* Hull plot and Speed Polar plots
* done in separate programs
* using HALO graphics language

C SUBROUTINE LIST

C DECK ACTFIN - active fins
SUBROUTINE ACTFIN (IV,ZERO,V,OMGE,OMGE2,TAF)

COMMON /APPEND/ NBKSET,NBKSTN(2),BKIMAG(2),BKFS(2),BKAS(2),
2 BKWD(2),BKSTM(10,2),BKHB(10,2),BKLNTB,BKWDTH,
2 BKWL(10,2),BKAN(10,2),NSKSET,SKIMAG(2),SKFLS(2),SKALS(2),
2 SKAUS(2),SKHB(2),SKFLWL(2),SKALWL(2),SKAUWL(2),NRDSET,RDIMAG(2),
2 RDRFS(2),RDRDAS(2),RDRHB(2),RDRFWL(2),RDRDRAWL(2),RDTFS(2),RDTAS(2),
2 RDTHB(2),RDTFWL(2),RDTAWL(2),NSBSET,SBIMAG(2),SOBRFS(2),SOBRAS(2),
2 SOBRHB(2),SOBRFW(2),SOBRAW(2),SIBRFS(2),SIBRAS(2),SIBRHB(2),
2 SIBRFW(2),SIBRAW(2),SBTFS(2),SBTAS(2),SBTB(2),SBTFWL(2),
2 SBTAWL(2),NFNSET,FNIMAG(2),FNRFS(2),FNRAS(2),
2 FNRHB(2),FNRFWL(2),FNRAWL(2),FNTFS(2),FNTAS(2),FNTHB(2),
2 FNTFWL(2),FNTAWL(2),NEXPRD,ENRDO(8),ENRDS(8)

COMMON /FINCON/ IACTFW,IFLCLS,FGAIN(8),FK(3),FA(3),FB(3),
2 FCLCS(8,2)

COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

COMMON /RLDBK/ FSUR(25),BMK(25),DK(25),CAK(25),HQ,HSPAN,HMNCHD,
2 HAREA,HXCP,HYCP,HZCP,HGAMMA,HYHAT,HEAR,HLCS,RQ(2),RSPAN(2),
2 RMNCHD(2),RAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),
2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
2 SYCP(2),SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2),BQ(2),
2 BSPAN(2),BMNCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),
2 PQ(2,2),PSPAN(2,2),PMNCHD(2,2),PAREA(2,2),PXCP(2,2),PYCP(2,2),
2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),PLCS(2,2),
2 STADMP(10),SHPDMP(10,8),ENCON,VPHI,TPHI,WMELM(4,9),SFELM(4,9,8),
2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
2 ENWM,ENSF(8,8),ENRE(8),ENPE(6),ENFE(8),ENHE(8),ENBE(8),
2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(6),ENSL(6),ENEL(6),
2 ENSHP(8,8),RELM(4,9),ITS(25),RD(25),EDDY(8,25),RGB(25)

```

REAL RDBLK(2692)
EQUIVALENCE (PSUR(1),RDBLK(1))

COMPLEX TAF(3),FGC,CTERM,ZERO

DO 10 I=1,3
TAF(I) = ZERO
CONTINUE
10 FGC = ((FK(1)-OMGE2*FK(3))+II*OMGE*FK(2))/(((FA(1)-OMGE2*FA(3))+
2 II*OMGE*FA(2))*((FB(1)-OMGE2*FB(3))+II*OMGE*FB(2)))
DO 30 K=1,NFNSET
XCP = FXCP(K)
ARM = - FMNCHD(K)/6
YHAT = FYHAT(K)
AP = PI*RHO*FSPAN(K)*(FMNCHD(K)/2)**2
TEMP = FLCS(K)
IF (IFCLCS .EQ. 1) TEMP = FCLCS(IV,K)
FZ = (RHO/2)*FAREA(K)*TEMP
SINGAM = SIN(FGAMMA(K)*DEGRAD)
CTERM = FGC*(ARM*AP*OMGE2-II*OMGE*(ARM*FZ-3*AP)*V+FZ*V*V)
M1 = 1
IF (FNIMAG(K) .EQ. 2) M1 = 2
* SIN(180-GAMMA)=SIN(GAMMA) FOR FIN ON STBD SIDE
DO 20 M=1,M1
TAF(1) = TAF(1) - SINGAM*CTERM
TAF(2) = TAF(2) + YHAT*CTERM
TAF(3) = TAF(3) - SINGAM*XCP*CTERM
20 CONTINUE
30 CONTINUE

RETURN
END

C DECK ADRES
SUBROUTINE ADRES (NL,NU,MOTV,MOTL,HJV,HJL,H7,RAO1,PHS1,RAO2,PHS2,
2 OMEGA,NMOT,NPLANE,NOMEGA,RADDEG,COSMU,RHO,IPHS)
COMPLEX MOTV(NMOT,NOMEGA),MOTL(NMOT,NOMEGA),HJV(NMOT,NOMEGA),
2 HJL(NMOT,NOMEGA),H7(NOMEGA),ARES,TEMPL
DIMENSION RAO1(NOMEGA),PHS1(NOMEGA),RAO2(NOMEGA),PHS2(NOMEGA),
2 OMEGA(NOMEGA)

DO 30 I=NL,NU
DO 20 J=1,NPLANE
ARES = H7(I)
DO 10 N=1,NMOT
TEMPL = MOTL(N,I)
IF (J .EQ. 2) MOTL(N,I) = - MOTL(N,I)
ARES = ARES + MOTV(N,I)*HJV(N,I) + MOTL(N,I)*HJL(N,I)
MOTL(N,I) = TEMPL
10 CONTINUE
TEMP = - 0.5*RHO*OMEGA(I)*COSMU*AIMAG(ARES)
IF (J .EQ. 1) RAO1(I) = TEMP
IF (J .EQ. 2) RAO2(I) = TEMP
IF (IPHS.EQ.1 .AND. J.EQ.1) PHS1(I) = 0.
IF (IPHS.EQ.1 .AND. J.EQ.2) PHS2(I) = 0.
20 CONTINUE
30 CONTINUE

RETURN
END

C DECK AINPUT
SUBROUTINE AINPUT

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL
INTEGER   SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL

```

```

COMMON /SMPSSYS/ FIS,AS,SIS,SOS,SDS,HALOS,DEV,PRN,SMPPS,SMPIS,
2 SMPDS,SHPTYPS,SHIPS,VARS,CYCLS,TITLES,OPTION,LSIS,LSOS,
2 LSDS,LHALOS,LDEV,LPRN,LSMPPS,LSMPIS,LSMPOS,LSMPDS,LSHPTYPS,
2 LSHIPS,LTITLES
CHARACTER*160 AS
CHARACTER*80 FIS,SIS,SOS,SDS,TITLES
CHARACTER*20 HALOS,DEV,PRN,SMPPS,SMPIS,SMPOS,SMPDS,SHPTYPS
CHARACTER SHIPS*6,VARS*2,CYCLS*2
INTEGER*2 OPTION

CHARACTER*4 ALINE(20)

FIS = SIS(1:LSIS)//'.INP'
OPEN (UNIT=ICARD,FILE=FIS,STATUS='OLD')

10 L = 0
L = L + 1
IF (MOD(L,50) .EQ. 1) WRITE (IPRIN,1000) (I,I=1,8)
1000 FORMAT (1H1,42X,21HI N P U T   C A R D S//50X,6HCOLUMN/8X,
2 8(9X,I1)/8H CARD ,8(10H1234567890)//)
READ (ICARD,1010) ALINE
WRITE (IPRIN,1020) L,ALINE
1010 FORMAT (20A4)
1020 FORMAT (1X,I4,3X,20A4)
IF (ALINE(1) .NE. 'STOP') GO TO 10

CLOSE (UNIT=ICARD)

RETURN
END

C DECK ALAG
FUNCTION ALAG(X)

* this function sets ALOG(X)=0 when x=0
IF (X .LE. 1. E-08) GO TO 7
ALAG=ALOG(X)
GO TO 8
7 ALAG=0.

8 RETURN
END

C DECK ALGRNG
SUBROUTINE ALGRNG (N,W,S,AREA)

* This subroutine computes the area under the curve for a particular
* spectrum. An odd number of points (frequencies) should be used.

DIMENSION W(N),S(N)

MN=N-2
AREA=0.
TEMP = 0.
DO 20 M=1,MN,2
A=W(M+2)-W(M)
B=W(M+1)-W(M)
C=W(M)-W(M)
PAREA = A*A/6.*(S(M)*(3.*C-A)/(A*C)+S(M+1)*A/(B*C)+
2 S(M+2)*(2.*A-3.*C)/(A*B))
TEMP = PAREA
IF (PAREA .LT. 0.) TEMP = 0.
AREA = AREA + TEMP
20 CONTINUE
IF (MOD(N,2) .EQ. 1) GO TO 30
DELW = W(N) - W(N-1)
DELS = S(N) - S(N-1)
AREA = AREA + S(N-1)*DELW + .5*DELS*DELW
30 CONTINUE

```

```

      AREA = ABS(AREA)
      RETURN
      END

C DECK AMD
      SUBROUTINE AMD (OMEGAE,TELEM,TV,TL)

*   UNPACKS ZERO-SPEED ADDED MASS AND DAMPING AND ADDS FORWARD SPEED
*   TERMS

      COMMON /CH3D/ ISIGMA,SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI,
      2 IMMIN,IMMAX,IMDEL,LMIN,LMAX
      REAL SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI(4)
      INTEGER ISIGMA,IMMIN,IMMAX,IMDEL,LMIN,LMAX

      COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
      1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
      INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
      REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
      2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

      COMMON /PHYSCL/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
      2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
      COMPLEX II
      CHARACTER*4 PUNITS(2)
      REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
      1 RHOF,GNUS,GNUF,FTMETR

      COMMON /STATE/ LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL
      LOGICAL LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL

      COMPLEX TELEM(4,9,10)
      COMPLEX T3D(10),TV(3,3),TL(3,3)
      DIMENSION LDX(6,6)

      DATA ((LDX(I,J),J=1,6),I=1,6)
      +/ 1, 0, 0, 0, 0, 0,
      + 0, 5, 0, 8, 0, 9,
      + 0, 0, 2, 0, 4, 0,
      + 0, -8, 0, 6, 0, 10,
      + 0, 0, -4, 0, 3, 0,
      + 0, -9, 0,-10, 0, 7/

      DO 20 L=1,10
      T3D(L) = (0.0,0.0)
20    CONTINUE
      DO 40 L=LMIN,LMAX
      DO 30 K=1,4
      T3D(L) = T3D(L) + WTSI(K)*TELEM(K,ISIGMA,L)
30    CONTINUE
40    CONTINUE
      IF(.NOT.VRT) GO TO 3
      DO 1 I=1,3
      IDX=2*I-1
      DO 2 J=1,3
      JDX=2*J-1
      L=LDX(IDX,JDY)
      IF(L.EQ.0) TV(I,J)=(0.0,0.0)
      IF(L.GT.0) TV(I,J)=T3D(L)
      IF(L.LT.0) TV(I,J)=TV(J,I)
2     CONTINUE
1     CONTINUE
      TV(2,3)=TV(2,3)+V*TV(2,2)/(II*OMEGAE)
      TV(3,2)=TV(3,2)-V*TV(2,2)/(II*OMEGAE)
      TV(3,3)=TV(3,3)+V*V*TV(2,2)/OMEGAE**2
      IF(.NOT.LAT) GO TO 6
3     CONTINUE
      DO 4 I=1,3
      IDX=2*I
      DO 5 J=1,3
      JDX=2*J

```

```

L=LDX(IDX, JDX)
IF(L.EQ.0) TL(I,J)=(0.0,0.0)
IF(L.GT.0) TL(I,J)=T3D(L)
IF(L.LT.0) TL(I,J)=TL(J,I)
5 CONTINUE
4 CONTINUE
TL(1,3)=TL(1,3)-V*TL(1,1)/(II*OMEGAE)
TL(2,3)=TL(2,3)-V*TL(2,1)/(II*OMEGAE)
TL(3,1)=TL(3,1)+V*TL(1,1)/(II*OMEGAE)
TL(3,2)=TL(3,2)+V*TL(1,2)/(II*OMEGAE)
TL(3,3) = TL(3,3) + V*V*TL(1,1)/OMEGAE**2
6 CONTINUE

RETURN
END

C DECK AMDPRN
SUBROUTINE AMDPRN (PROMG,NPROMG)

* nondimensionalizes and prints zero-speed added mass and damping

COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8),
REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL
INTEGER SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL

COMMON /PHYSICO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

COMMON /STATE/ LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL
LOGICAL LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL

COMMON/TELEM/TELEM
COMPLEX TELEM(4,9,10)

DIMENSION LPWR(10),LDX(10)
DIMENSION A(10),B(10,30)
COMPLEX T,CDUM
DIMENSION PROMG(30)

DATA LPWR /0,0,2,1,0,2,2,1,1,2/
DATA LDX /1,3,5,9,2,4,6,7,8,10/

SRGDL=SQRT(GRAV/LPP)
LMIN=-1
IF(.NOT.VRT) LMIN=5
LMAX=10

```

```

IF(.NOT.LAT) LMAX=4
DO 1 I=1,10
A(I)=0.
DO 2 J=1,NPROMG
B(I,J)=0.
2 CONTINUE
1 CONTINUE
WRITE (IPRIN,601) TITLE
WRITE (IPRIN,602)
DO 3 IOMEGA=1,NPROMG
DO 4 L=LMIN,LMAX
LL=LDX(L)
ASCALE=RHO*NEBLA*LPP**LPWR(L)
BSCALE=ASCALE*SRGDL
CALL CPLVAL (SIGMA,NSIGMA,TELEM(1,1,L),PROMG(IOMEGA),T,
1 CDUM, IDUM)
A(LL)=REAL(T)/(-PROMG(IOMEGA)**2)/ASCALE
B(LL,IOMEGA)=AIMAG(T)/PROMG(IOMEGA)/BSCALE
4 CONTINUE
OMGND=PROMG(IOMEGA)/SRGDL
WRITE (IPRIN,604) OMGND,(A(L),L=1,10)
3 CONTINUE
WRITE (IPRIN,603)
DO 5 IOMEGA=1,NPROMG
OMGND=PROMG(IOMEGA)/SRGDL
WRITE (IPRIN,604) OMGND,(B(L,IOMEGA),L=1,10)
5 CONTINUE
WRITE (IPRIN,605)
601 FORMAT (1H1,23X,20A4//42X,
2 46HZERO-SPEED ADDED-MASS AND DAMPING COEFFICIENTS//)
602 FORMAT (' NON-DIMENSIONAL ADDED-MASS'//
1 ' SIGMA',3X,'A(1,1)',6X,'A(2,2)',6X,'A(3,3)',6X,'A(4,4)',6X,
1 'A(5,5)',6X,'A(6,6)',6X,'A(2,4)',6X,'A(2,6)',6X,'A(3,5)',6X,
1 'A(4,6)',/)
603 FORMAT (' NON-DIMENSIONAL DAMPING'//
1 ' SIGMA',3X,'B(1,1)',6X,'B(2,2)',6X,'B(3,3)',6X,'B(4,4)',6X,
1 'B(5,5)',6X,'B(6,6)',6X,'B(2,4)',6X,'B(2,6)',6X,'B(3,5)',6X,
2 'B(4,6)',/)
604 FORMAT (1X,F6.3,1P10E12.4)
605 FORMAT (///' (SIGMA IS NON-DIMENSIONAL FREQUENCY)')

RETURN
END

C DECK ATAN2D
FUNCTION ATAN2D (B,A,RADDEG)

* arctangent function in degrees for any quadrant
DATA EPS /1.E-10/

IF (B .EQ. 0.) ATAN2D = 0.
IF (B .GT. 0.) ATAN2D = 90.
IF (B .LT. 0.) ATAN2D = -90.
IF (ABS(A) .GT. EPS) ATAN2D = ATAN2(B,A)*RADDEG

RETURN
END

C DECK ATAN3
FUNCTION ATAN3(X,Y)

* this function is to take care of the case of ATAN2(0,0)

AX=ABS(X)
AY=ABS(Y)
IF(AX .LE. 1.E-08 .AND.AY .LE. 1. E-08) GO TO 5
ATAN3=ATAN2(X,Y)
GO TO 10
5 ATAN3=0.

10 RETURN

```

END

```
C DECK BILGEK
      SUBROUTINE BILGEK (IBLGK)

*   calculates bilge keel damping using method of KATO
*   W. R. MCCREIGHT, DTNSRDC

COMMON /APPEND/ NBKSET,NBKSTN(2),BKIMAG(2),BKFS(2),BKAS(2),
2 BKWD(2),BKSTN(10,2),BKHB(10,2),BKLNTH,BKWDTH,
2 BKWL(10,2),BKAN(10,2),NSKSET,SKIMAG(2),SKFLS(2),SKALS(2),
2 SKAUS(2),SKHB(2),SKFLWL(2),SKALWL(2),SKAUWL(2),NRDSET,RDIMAG(2),
2 RDRFS(2),RDRAS(2),RDRHB(2),RDRFWL(2),RDRAWL(2),RDTFS(2),RDTAS(2),
2 RDTHB(2),RDTFWL(2),RDTAWL(2),NSBSET,SBIMAG(2),SOBRFS(2),SOBRAS(2)
2 SOBRHB(2),SOBRFW(2),SOBRAW(2),SIBRFS(2),SIBRAS(2),SIBRHB(2),
2 SIBRFW(2),SIBRAW(2),SBTFS(2),SBTAS(2),SBTHB(2),SBTFWL(2),
2 SBTAWL(2),NFNSET,FNIMAG(2),FNRFS(2),FNRAS(2),
2 FNRHB(2),FNRFWL(2),FNRAWL(2),FNTFS(2),FNTAS(2),FNTHB(2),
2 FNTFWL(2),FNTAWL(2),NEXPRD,ENRDO(8),ENRDS(8)

COMMON /CH3D/ ISIGMA,SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI,
2 IMMIN,IMMAX,IMDEL,LMIN,LMAX
REAL SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI(4)
INTEGER ISIGMA,IMMIN,IMMAX,IMDEL,LMIN,LMAX

COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /PHYSICO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

COMMON /RDGEO/ BKLEN,WBKMAX,DBLKEL(25),SRBS(25),PHIS(25),CPS(25),
2 BKT(25),RKS(25),SSTR(25)

COMMON /RLDBK/ PSUR(25),BMK(25),DK(25),CAK(25),HQ,HSPAN,HMNCHD,
2 HAREA,HXCP,HYCP,HZCP,HGAMMA,HYHAT,HEAR,HLCS,RQ(2),RSPAN(2),
2 RMNCHD(2),RAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),
2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
2 SYCP(2),SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2),BQ(2),
2 BSPAN(2),BMNCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),
2 PQ(2,2),PSPAN(2,2),PMNCHD(2,2),PAREA(2,2),PXCP(2,2),PYCP(2,2),
2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),PLCS(2,2),
2 STADMP(10),SHPDMP(10,8),ENCON,WPHI,TPHI,WHELM(4,9),SFELM(4,9,8),
2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
2 ENWM,ENSF(8,8),ENRE(8),ENPE(8),ENFE(8),ENHE(8),ENBE(8),
2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(8),ENSL(8),ENBL(8),
2 ENSHP(8,8),RELM(4,9),ITS(25),RD(25),EDDY(8,25),RGB(25)
REAL RDBLK(2692)
EQUIVALENCE (PSUR(1),KDBLK(1))
```

REAL KAPPA,KG,LAMBDA,LBKEEL
CHARACTER*4 METER

EXTERNAL EXP

DATA METER /'METE'/

```
LBKEEL=BKLEN
NSM = NSTATN - 1
DO 40 K=2,NSM
IF (NOFSET(K) .LT. 2) GO TO 40
IF (DLBKEL(K) .EQ. 0.) GO TO 40
NNODES = NOFSET(K)
R=RD(K)
BLOCAL = 2*BMK(K)
TLOCAL = ABS(BKT(K))
KG = VCG + TLOCAL
BBKEEL = BKWD(IBLGK)
PHI=PHIS(K)
COSPHI=CPS(K)
RK=RKS(K)
SS = SSTR(K)
SRB=SRBS(K)
RF=SRB*Y(NNODES,K)
EPS=ATAN(SRB)
CO=1000.*(1.44+3.8*PHI**3)
KAPPA = R*(1.0 + RF/BLOCAL)**2 / SQRT(BLOCAL*KG/2.)
XI=BBKEEL/(RK*PHI**0.75)
AN=1.40+2.03*EXP(-25.*XI)
ALPHA=2.0-AN
CK=1.0+3.5*EXP(-9.0*KAPPA)
SGM=2.0*BBKEEL/LBKEEL
CN=1.98*EXP(-5.5*SGM)
Q = (0.5*BLOCAL*TAN(PI/4. - EPS/2.) + RF - KG) * SIN(PI/4. +
2 EPS/2.)
PO = KG - TLOCAL/3. - 2.*RF/3.
P1 = 0.88*(KG - TLOCAL - 0.54*(BLOCAL/2. - (TLOCAL - RF)*TAN(
2 PI/4. + EPS/2.)))
LAMBDA = R/(TLOCAL - RF*(BLOCAL - 2.*R)/BLOCAL)
FLAMB=1.34*SIN(PI*LAMBDA/3.6)/
1 (1.0+0.162*SIN(PI*(LAMBDA-0.9)/1.8))
BCIRC = COSPHI + SS*(Q+PO-(PO-P1)*FLAMB)/(2.*BBKEEL*RK)
DAKEEL=2.0*DLBKEL(K)*BBKEEL
CON = 4.0*RHO/(3.0*PI)*CK*CN*BCIRC*DAKEEL*RK**3
DO 30 IA=1,NRANG
DO 20 IS=1,NSIGMA
PERE = TPI/SIGMA(IS)
F = RK*RANG(IA)*PHI**1.7/(PERE*SQRT(BBKEEL))
```

* F must be in meters

```
IF (PUNITS(1) .NE. METER) F = F*SQRT(FTMETR)
CS = CO*F**(-ALPHA)/(2.68*1000.0)
CA = 1.
RN = (8.*BBKEEL*RK*RANG(IA) / (PERE*GNU)) * REYSCL
IF (RN .GE. 1000.) GO TO 17
AL10RN = ALOG(RN)/ALOG(10.)
CA = 1.95 - 0.25*AL10RN + 0.20*SIN(PI*(AL10RN-2.19)/0.54)
17 CONTINUE
STADMP(IS) = CON*CS*CA*SIGMA(IS)*RANG(IA)
STADMP(IS) = SIGMA(IS)*STADMP(IS)
SHPDMP(IS,IA) = SHPDMP(IS,IA) + STADMP(IS)
20 CONTINUE
30 CONTINUE
40 CONTINUE
```

RETURN
END

C DECK BKEDDY
SUBROUTINE BKEDDY

```

COMMON /APPEND/ NBKSET,NBKSTN(2),BKIMAG(2),BKFS(2),BKAS(2),
2 BKWD(2),BKSTN(10,2),BKHB(10,2),BKLNTH,BKWDTH,
2 BKWL(10,2),BKAN(10,2),NSKSET,SKIMAG(2),SKFLS(2),SKALS(2),
2 SKAUS(2),SKHB(2),SKFLWL(2),SKALWL(2),NRDSET,RDIMAG(2),
2 RDRFS(2),RDRAS(2),RDRHB(2),RDRFWL(2),RDRRAWL(2),RDTFS(2),RDTAS(2),
2 RDTHB(2),RDTFWL(2),RDTAWL(2),NSBSET,SBIMAG(2),SOBRFS(2),SOBRAS(2),
2 SOBRHB(2),SOBRFW(2),SOBRAW(2),SIBRF(2),SIBRAS(2),SIBRHB(2),
2 SIBRFW(2),SIBRAW(2),SBTFS(2),SETAS(2),SBTHB(2),SBTFWL(2),
2 SBTAWL(2),NFNSET,FNIMAG(2),FNRFS(2),FNRAS(2),
2 FNRRHB(2),FNRFWL(2),FNRAWL(2),FNTFS(2),FNTAS(2),FNTHB(2),
2 FNTFWL(2),FNTAWL(2),NEXPRD,ENRDO(8),ENRDS(8)

COMMON /CH3D/ ISIGMA,SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI,
2 IMMIN,IMMAX,IMDEL,LMIN,LMAX
REAL SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI(4)
INTEGER ISIGMA,IMMIN,IMMAX,IMDEL,LMIN,LMAX

COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELM,NEBLA,KPITCH,KROLL,KYAWL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELM,NEBLA,KPITCH,KROLL,KYAWL,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /RLDBK/ PSUR(25),BMK(25),DK(25),HQ,HSPAN,HMNCHD,
2 HAREA,HXCP,HZCP,HGAMMA,HYHAT,HEAR,HLCS,RQ(2),RSPAN(2),
2 RMNCHD(2),RAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),
2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
2 SYCP(2),SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2),BQ(2),
2 BSPAN(2),BMNCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),
2 PQ(2,2),PSPAN(2,2),PMNCHD(2,2),PAREA(2,2),PXCP(2,2),PYCP(2,2),
2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),PLCS(2,2),
2 STADMP(10),SHPDMP(10,8),ENCON,WPHI,TPHI,WMELM(4,9),SFELM(4,9,8),
2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
2 ENWM,ENSF(8,8),ENRE(8),ENPE(8),ENFE(8),ENHE(8),ENBE(8),
2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(8),ENSL(8),ENBL(8),
2 ENSHP(8,8),RELM(4,9),ITS(25),RD(25),EDDY(8,25),RGB(25)
REAL RDBLK(2692)
EQUIVALENCE (PSUR(1),RDBLK(1))

DO 20 IA=1,NRANG
ENBE(IA) = 0
DO 10 IS=1,NSIGMA
SHPDMP(IS,IA) = 0
10 CONTINUE
20 CONTINUE
IF (NBKSET .EQ. 0) GO TO 100
DO 30 I=1,NBKSET
CALL CALRGM(I)
CALL BILGEK(I)
30 CONTINUE
DO 40 IA=1,NRANG
CALL SPFIT (SIGMA,SHPDMP(1,IA),BEELM(1,1,IA),NSIGMA)
ENBE(IA) = ENCON*REVAL(BEELM(1,ISIGMA,IA),WTSI)
40 CONTINUE
100 CONTINUE

RETURN

```

```

END

C DECK BKLIIFT
SUBROUTINE BKLIIFT

COMMON /APPEND/ NBKSET,NBKSTM(2),BKIMAG(2),BKFS(2),BKAS(2),
2 BKWD(2),BKSTN(10,2),BKHB(10,2),BKLNTB,BKWDTH,
2 BKWL(10,2),BKAN(10,2),NSKSET,SKIMAG(2),SKFLS(2),SKALS(2),
2 SKAUS(2),SKHB(2),SKFLWL(2),SKALWL(2),SKAUWL(2),NRDSET,RDIMAG(2),
2 RDRFS(2),RDRAS(2),RDRHB(2),RDRFWL(2),RDRAWL(2),RDTFS(2),RDTAS(2),
2 RDTHB(2),RDTFWL(2),RDTAWL(2),NSBSET,SBIMAG(2),SOBRFS(2),SOBRAS(2),
2 SOBRHB(2),SOBRFW(2),SOBRAW(2),SIBRFS(2),SIBRAS(2),SIBRHB(2),
2 SIBRFW(2),SIBRAW(2),SIBTFS(2),SBTAS(2),SBTHB(2),SBTFWL(2),
2 SBTAWL(2),NFNSET,FNIMAG(2),FNRFS(2),FNRAS(2),
2 FNRHB(2),FNRFWL(2),FNRAWL(2),FNTFS(2),FNTAS(2),FNTHB(2),
2 FNTFWL(2),FNTAWL(2),NEXPRD,ENRDO(8),ENRDS(8)

COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS.
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FEDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),TITLE(20),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

COMMON /RLDBK/ PSUR(25),BMK(25),DK(25),CAK(25),HQ,HSPAN,HMNCHD,
2 HAREA,HXCP,HYCP,HZCP,HGAMMA,HYHAT,HEAR,HLCS,RQ(2),RSPAN(2),
2 RMNCHD(2),RAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),
2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
2 SYCP(2),SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2),BQ(2),
2 BSPAN(2),BMNCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),
2 PQ(2,2),PSPAN(2,2),PMNCHD(2,2),PAREA(2,2),PXCP(2,2),PYCP(2,2),
2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),PLCS(2,2),
2 STADMP(10),SHPDMP(10,8),ENCON,WPHI,TPHI,WELLM(4,9),SFELM(4,9,8),
2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
2 ENWM,ENSF(8,8),ENRE(8),ENPE(8),ENFE(8),ENHE(8),ENBE(8),
2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(8),ENSL(8),ENBL(8),
2 ENSHP(8,8),RELM(4,9),ITS(25),RD(25),EDDY(8,25),RGB(25)
REAL RDBLK(2692)
EQUIVALENCE (PSUR(1),RDBLK(1))

REAL LCS,MCHORD

IF (NBKSET .EQ. 0) GO TO 30
EN = 0
STASPC = LPP/20
DO 20 K=1,NBKSET
NBKS = NBKSTM(K)
XBKF = LCB - BKFS(K)*STASPC
XBKA = LCB - BKAS(K)*STASPC
M = NBKS/2
IF (M .EQ. 0) M = 1
YBK = BKHB(M,K)

```

```

ZBKF = BKWL(1,K) - (DBLWL+VCG)
ZBKA = BKWL(NBKS,K) - (DBLWL+VCG)
Q = 2
SUM = 0
DO 10 I=1,NBKS
    SUM = SUM + BKAN(I,K)
10 CONTINUE
    GAMMA = - SUM/NBKS
    SPAN = BKWD(K)
    MCHORD = XBKF - XBKA

* area
    AREA = SPAN*MCHORD

* center of pressure
    XCP = XBKF - 0.5*MCHORD
    YCP = YBK + 0.5*SPAN
    ZCP = (ZBKF + ZBKA)/2

* moment arm
    GAM = GAMMA*DEGRAD
    YHAT = YCP*COS(GAM) + ZCP*SIN(GAM)

* effective aspect ratio
    EAR = 2*SPAN/MCHORD

* lift curve slope
    LCS = (PI/2)*EAR
    BQ(K) = Q
    BSPAN(K) = SPAN
    BMNCHD(K) = MCHORD
    BAREA(K) = AREA
    BXCP(K) = XCP
    BYCP(K) = YCP
    BZCP(K) = ZCP
    BGAMMA(K) = GAMMA
    BYHAT(K) = YHAT
    BEAR(K) = EAR
    BLCS(K) = LCS
    EN = EN + Q*(RHO/2)*AREA*LCS*YHAT*YHAT*WPHI*ENCON
20 CONTINUE
30 CONTINUE
    DO 40 IV=1,NVK
        ENBL(IV) = 0
        IF (NBKSET .GT. 0) ENBL(IV) = EN*VFS(IV)
40 CONTINUE

    RETURN
END

C DECK BMAX
FUNCTION BMAX(N,X)

DIMENSION X(30)
A=X(1)
IF(N.LE.1) GO TO 2
DO 1 I=2,N
    IF(X(I).GT.A) A=X(I)
1 CONTINUE
2 CONTINUE
BMAX=A

RETURN
END

C DECK BRWVSP
SUBROUTINE BRWVSP (NOK,SIGWH,TO,W,S)

```

```
* this routine calculates a BRETSCHNEIDER 2-parameter wave spectrum
* (significant wave height, modal wave period)
* W.G.MEYERS, DTNSRDC, 072977
```

```
DIMENSION W(NOK),S(NOK)

EXTERNAL EXP

DATA A,B /487.0626,1948.2444/
T04 = T0**4

* for Pierson-Moskowitz wave spectrum
* T04 = 68.0936*SIGHWH**2

CON1 = A*SIGHWH**2/T04
CON2 = B/T04
DO 10 I=1,NOK
W4 = W(I)**4
W5 = W(I)*W4
ARG = CON2/W4
IF (ARG.GT.50.) S(I)=0.
IF (ARG.GT.50.) GO TO 10
S(I) = CON1/W5*EXP(-ARG)
10 CONTINUE

RETURN
END
```

```
C DECK CALRGM
SUBROUTINE CALRGM (IBLGK)
```

```
COMMON /APPEND/ NBKSET,NBKSTN(2),BKIMAG(2),BKFS(2),BKAS(2),
2 BKWD(2),BKSTN(10,2),BKHB(10,2),BKLNTH,BKWDTH,
2 BKWL(10,2),BKAN(10,2),NSKSET,SKIMAG(2),SKFLS(2),SKALS(2),
2 SKAUS(2),SKHB(2),SKFLWL(2),SKALWL(2),SKAUWL(2),NRDSET,RDIMAG(2),
2 RDRFS(2),RDRAS(2),RDRHB(2),RDRFWL(2),RDRAWL(2),RDTFS(2),RDTAS(2),
2 RDTHB(2),RDTFWL(2),RDTAWL(2),NSBSET,SBIMAG(2),SOBRFS(2),SOBRAS(2),
2 SOBRHB(2),SOBRFW(2),SOBRAW(2),SIBRFS(2),SIBRAS(2),SIBRHB(2),
2 SIBRFW(2),SIBRAW(2),SBTFS(2),SBTAS(2),SETHB(2),SBTFWL(2),
2 SBTAWL(2),NFNSET,FNIMAG(2),FNRFS(2),FNRA(2),
2 FNRHB(2),FNRFWL(2),FNRAWL(2),FNTFS(2),FNTAS(2),FNTHB(2),
2 FNTFWL(2),FNTAWL(2),NEXPRD,ENRDO(8),ENRDS(8)

COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
2 LRAOPR,ADRPR,ORGOPTN,GMMOM,KG,STATN(25),NSOFST(25),
2 NLEWF(25),HLFBTH(10,25),WTRLNE(10,25),BLEWF(25),TLEWF(25),
2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
2 ZPTFBD(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
2 STATNM,STATIS
CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
2 FBNUMB,PTNUMB,ORGOPTN
REAL KG

COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPICTH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(26),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
```

```

2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),ESTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /RDGEO/ BKLEN,WBKMAX,DLBKEL(25),SRBS(25),PHIS(25),CPS(25),
2 BKT(25),RKS(25),SSTR(25)

REAL LBKEEL
LBKEEL=0.
NBKS = NBKSTN(IBLGK)
STASPC = LPP/20
M = NSTATN
NSM = NSTATN - 1
DO 1 K=2,NSM
M = M - 1
IF (NOFSET(K) .LT. 2) GO TO 1
DELTAL=0.
SRB=0.
PHI=0.
COSPHI=1.
RK=1.
S=0.
IF (STATN(M) .GT. BKAS(IBLGK) .OR. STATN(M) .LT. BKFS(IBLGK))
2 GO TO 6
IF (STATN(M+1) .GT. BKAS(IBLGK))
2 DELTAL = (BKAS(IBLGK) - STATN(M))*STASPC
IF (STATN(M-1) .LT. BKFS(IBLGK))
2 DELTAL = (STATN(M) - BKFS(IBLGK))*STASPC
IF (STATN(M+1) .LE. BKAS(IBLGK))
2 DELTAL = DELTAL + (STATN(M+1) - STATN(M))*STASPC/2
IF (STATN(M-1) .GE. BKFS(IBLGK))
2 DELTAL = DELTAL + (STATN(M) - STATN(M-1))*STASPC/2
NNODES=NOFSET(K)
DO 10 L=1,NBKS
IF (STATN(M) .NE. BKSTN(L,IBLGK)) GO TO 10
R0 = SQRT((BKHB(L,IBLGK)**2 + (BKWL(L,IBLGK) - (DBLWL+VCG))**2)
ARG = BKAN(L,IBLGK)*DEGRAD
YBKC = BKHB(L,IBLGK) + 0.5*BKWD(IBLGK)*COS(ARG)
ZBKC = (BKWL(L,IBLGK) - DBLWL) - 0.5*BKWD(IBLGK)*SIN(ARG)
RK = SQRT((YBKC**2 + (ZBKC-VCG)**2)
P1 = ASIN(-VCG/R0)
P2 = ATAN2(VCG + DBLWL - BKWL(L,IBLGK),BKHB(L,IBLGK))
PHI = P1 + P2
COSPHI = COS(ARG - P2)
S=0.
NNM=NNODES-1
DO 3 J=1,NNM
JS=NNODES-J+1
IF (BKHB(L,IBLGK) .GE. Y(JS-1,K)) GO TO 4
S=S+SQRT((Y(JS,K)-Y(JS-1,K))**2+(Z(JS,K)-Z(JS-1,K))**2)
3 CONTINUE
4 CONTINUE
S = S + SQRT((Y(JS,K) - BKHB(L,IBLGK))**2 + (Z(JS,K) -
2 (BKWL(L,IBLGK) - DBLWL))**2)

* find minimum slope for deadrise calculation in "BILGEK"
M2 = JS - 1
LS = M2 - 1
SRB = (Z(M2,K) - Z(LS,K)) / (Y(M2,K) - Y(LS,K))
J = JS
DO 130 I=2,M2
J = J - 1
JS1 = J - 1
SLOPE = (Z(J,K) - Z(JS1,K)) / (Y(J,K) - Y(JS1,K))
IF (SLOPE .EQ. 0.) GO TO 140
IF (SLOPE .GT. SRB) GO TO 140
LS= JS1
SRB = SLOPE
130 CONTINUE

* extrapolate slope to centerline to get local draft

```

```

*      (excluding skeg offsets)

140 BKT(K) = Z(LS,K) - SRB*Y(LS,K)
IF (BKT(K) .LT. Z(1,K)) BKT(K) = Z(1,K)
LBKEEL=LBKEEL+DELTAL
10  CONTINUE
6   CONTINUE
   DLBKEL(K)=DELTAL
   SRBS(K)=SRB
   PHIS(K)=PHI
   CPS(K)=COSPHI
   RKS(K)=RK
   SSTR(K)=S
1   CONTINUE
BKLEN=LBKEEL

RETURN
END

C DECK CDCOMP
SUBROUTINE CDCOMP ( N, NDIM, A, UL, IP )
*
*      COMPLEX MATRIX TRIANGULARIZATION BY GAUSSIAN ELIMINATION.

*      INPUT...
*          N = ORDER OF MATRIX.
*          NDIM = DECLARED DIMENSION OF ARRAY A .
*          A = COMPLEX MATRIX TO BE TRIANGULARIZED.

*      OUTPUT...
*          UL(I,J), I .LE. J = UPPER TRIANGULAR FACTOR, U .
*          UL(I,J), I .GT. J = MULTIPLIERS = LOWER TRIANGULAR
*                                FACTOR, I - L .
*          IP(K), K .LT. N = INDEX OF K-TH PIVOT ROW.
*          IP(N) = (-1)**(NUMBER OF INTERCHANGES) OR 0 .

*      USE "SOLVE" TO OBTAIN SOLUTION OF LINEAR SYSTEM.
*      DETERM( A ) = IP(N)*UL(1,1)*UL(2,2)*...*UL(N,N).
*      IF IP(N) = 0, A IS SINGULAR, SOLVE WILL DIVIDE BY ZERO.

*      INTERCHANGES FINISHED IN U, ONLY PARTIALY IN L .

REAL CABS
COMPLEX A, UL, T
INTEGER N, NDIM, IP, K, KP1, M, I, J
DIMENSION A(NDIM,NDIM), UL(NDIM,NDIM)
DIMENSION IP(NDIM)

DO 1050 I = 1, NDIM
DO 1000 J = 1, NDIM
UL(J,I) = A(J,I)
1000 CONTINUE
1050 CONTINUE

IP(N) = 1
DO 1700 K = 1, N
IF ( K.EQ. N ) GO TO 1600
KP1 = K + 1
M = K
DO 1100 I = KP1, N
IF ( CABS( UL(I,K) ) .GT. CABS( UL(M,K) ) ) M = I
1100 CONTINUE
IP(K) = M
IF ( M.NE. K ) IP(N) = -IP(N)
T = UL(M,K)
UL(M,K) = UL(K,K)
UL(K,K) = T
IF ( CABS(T) .EQ. 0.0 ) GO TO 1600
DO 1200 I = KP1, N
UL(I,K) = -UL(I,K)/T
1200 CONTINUE
DO 1500 J = KP1, N

```

```

T = UL(M,J)
UL(M,J) = UL(K,J)
UL(K,J) = T
IF ( CABS(T) .EQ. 0.0 ) GO TO 1400
DO 1300 I = KP1, N
UL(I,J) = UL(I,J) + UL(I,K)*T
1300 CONTINUE
1400 CONTINUE
1500 CONTINUE
1600 CONTINUE
IF ( CABS( UL(K,K) ) .EQ. 0.0 ) IP(N) = 0
1700 CONTINUE
99999 CONTINUE

RETURN
END

C DECK CEVAL
COMPLEX FUNCTION CEVAL (CSPLNE,WEIGHT)

COMPLEX CSPLNE(4)
DIMENSION WEIGHT(4)

CEVAL = (0.,0.)
DO 10 I=1,4
CEVAL = CEVAL + WEIGHT(I)*CSPLNE(I)
10 CONTINUE

RETURN
END

C DECK CLIP
SUBROUTINE CLIP (LIMIT,TFN,TFNMOD)

*   this routine imposes a limit on the magnitude of a dimensional
*   transfer function (surge, sway or yaw in quartering seas)
*   W.G.MEYERS, DTNSRDC, 072977

REAL LIMIT,MAGN
COMPLEX TFN,TFNMOD
MAGN = CABS(TFN)
IF (LIMIT.LE.0. .OR. MAGN.LE.LIMIT) GO TO 10

*   transfer function clipped

    RATIO = LIMIT/MAGN
    TFNMOD = RATIO*TFN
    GO TO 20
10 CONTINUE

*   transfer function not clipped

    TFNMOD = TFN
20 CONTINUE

RETURN
END

C DECK CMINR
FUNCTION CMINR (ISKIP,AA)

DIMENSION AA(3,4)
SUM=0.0
DO 1 I1=1,4
IF(I1.EQ.ISKIP) GO TO 2
I2=I1+1
IF(I2.GT.4) I2=1
IF(I2.EQ.ISKIP) I2=I2+1
IF(I2.GT.4) I2=1
I3=I2+1
IF(I3.GT.4) I3=1
IF(I3.EQ.ISKIP) I3=I3+1

```

```

IF(I3.GT.4) I3=1
SUM=SUM+AA(1,I1)*(AA(2,I2)*AA(3,I3)-AA(2,I3)*AA(3,I2))
2 CONTINUE
1 CONTINUE
CMINR=SUM

RETURN
END

C DECK COFOU
SUBROUTINE COFOU

* generate coefficient file containing speed-dependant added-mass
* and damping, exciting forces and KOCHIN functions

COMMON /CH3D/ ISIGMA,SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI,
2 IMMIN,IMMAX,IMDEL,LMIN,LMAX
REAL SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI(4)
INTEGER ISIGMA,IMMIN,IMMAX,IMDEL,LMIN,LMAX

COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
2 LRAOPR,ADRPR,ORGOPTN,GMNOM,KG,STATN(25),NSOFST(25),
2 NLEWF(25),HLFBTH(10,25),WTRLNE(10,25),BLEWF(25),TLEWF(25),
2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
2 ZPTFBD(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
2 STATNM,STATIS
CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
2 FBNUMB,PTNUMB,ORGOPTN
REAL KG

COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
INTEGER NVK,NMU,OMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL
INTEGER SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL

COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADdeg,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADdeg,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

COMMON /STATE/ LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL
LOGICAL LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL

COMMON /STELEM/ STELEM
COMPLEX STELEM(4,9,250)

COMMON/TELEM/TELEM

```

```

COMPLEX TELEM(4,9,10)

COMMON /WGHTS/ WTDL,NORM
REAL WTDL(10,25),NORM(4,10,25)

COMPLEX TV(3,3),TL(3,3),EXCV(3),EXCL(3),HJV(3),HJL(3),H7
COMPLEX STV(3,3),CDUM(3,3),SF3(25),SH3(25)
DIMENSION SA33(25),SB33(25)

DATA ISIGO /0/

READ (SCRFIL) WTDL,NORM
REWIND SCRFILE
REWIND COFFIL
READ (COFFIL) TELEM
DO 300 IV=1,NVK
V = VFS(IV)
NMU = NMNU(IV)
DO 200 IH=1,NMU
HDNG = MU(IH,IV)
SINMU = SIN(HDNG)
COSMU = COS(HDNG)
CON = V*COSMU/GRAV
DO 100 IW=1,NOMEGA
ALPHA = GMEGA(IW)*CON
OMEGAE = ABS(OMEGA(IW)*(1.0-ALPHA))
IF (OMEGAE .LT. SIGMA(1)) OMEGAE = SIGMA(1)
WE = OMEGAE
WE2 = WE*WE
CALL FINTSP (OMEGAE)
DO 50 K=1,NSTATN
SA33(K) = 0.
SB33(K) = 0.
NPT = NOFSET(K)
IF (NPT .LT. 2) GO TO 50
M = (K-1)*10 + 1
CALL AMD (OMEGAE,STELEM(1,1,M),STV,CDUM)
SA33(K) = REAL(STV(2,2))/(-WE2)
SB33(K) = AIMAG(STV(2,2))/WE
50 CONTINUE
CALL AMD (OMEGAE,TELEM,TV,TL)
IF (ISIGMA .NE. ISIGO) CALL RDPELM
ISIGO = ISIGMA
CALL EXFOR (OMEGA(IW),OMEGAE,EXCV,EXCL,HJV,HJL,H7,SF3,SH3)
WRITE (COFFIL) OMEGAE,TV,TL,EXCV,EXCL,HJV,HJL,H7
IF (LOADS) WRITE (LCOFIL) (SF3(I),SH3(I),SA33(I),SB33(I),I=1,
2 NSTATN)
100 CONTINUE
200 CONTINUE
300 CONTINUE
REWIND COFFIL

RETURN
END

```

```

C DECK CONIWT
SUBROUTINE CONIWT (W,CELEM,NNODE)

* SUBROUTINE TO GENERATE WEIGHTS FOR INTEGRAL ALONG CONTOUR
* DEFINED BY PARAMETRIC SPLINE CURVE

* INPUT
* CELEM(8,J),J=1,(NNODE-1) PARAMETRIC SPLINE FIT TO HULL
* CONTOUR IN ENDPOINT-TANGENT FORMAT-
* X(0),Y(0),DX(0),DY(0),X(1),Y(1),DX(1),DY(1)

* OUTPUT
* W(J), J=1,NNODE WEIGHTS SUCH THAT INTEGRAL OF F.DS =
* SUM OF F(J).W(J)

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,JCARD,TEXFIL,IPRIN,
2 SCRFILE,NPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,

```

```

2 SPTFIL,LACFIL,LAEFIL
2 INTEGER      SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL

DIMENSION CELEM(8,9),FELM(4,9)
DIMENSION TG(10),DLDT(6)
DIMENSION A(5,5),          IP(5),W(10)
DIMENSION F(10),CF(4),CD(5),CG(8),CGI(9)
DIMENSION X(4),Y(4),STORCD(9,5)
DIMENSION SCR(3),XDS(5),YDS(5),SUM(5)

IF (NNODE.GT.10)  WRITE (IPRIN,602) NNODE
IF (NNODE.GT.10)  STOP
IF (NNODE.LT.2)   WRITE (IPRIN,602) NNODE
IF (NNODE.LT.2)   STOP
602 FORMAT (' ERROR - CONIWT - NNODE = ',I5)
NELEM=NNODE-1
DO 1 I=1,NNODE
TG(I)=I-1
1 CONTINUE
DO 2 I=1,NNODE
W(I)=0.0
2 CONTINUE

* fit polynomial to dl/dt
* set up matrices

DO 3 I=1,5
T=0.25*(I-1)
A(I,1)=1.0
DO 4 J=2,5
A(I,J)=T*A(I,J-1)
4 CONTINUE
3 CONTINUE
CALL RDCOMP (5,5,A,IP)
IF (IP(5).EQ.0) GO TO 101
DO 5 K=1,NELEM
X(1)=CELEM(1,K)
X(2)=CELEM(3,K)
X(3)=3.0*(CELEM(5,K)-CELEM(1,K))-2.0*CELEM(3,K)-CELEM(7,K)
X(4)=CELEM(7,K)+CELEM(3,K)+2.0*(CELEM(1,K)-CELEM(5,K))
Y(1)=CELEM(2,K)
Y(2)=CELEM(4,K)
Y(3)=3.0*(CELEM(6,K)-CELEM(2,K))-2.0*CELEM(4,K)-CELEM(8,K)
Y(4)=CELEM(8,K)+CELEM(4,K)+2.0*(CELEM(2,K)-CELEM(6,K))

* evaluate dl/dt at five points over (0,1)

CALL PDER (SCR,IDXD,X,4)
CALL PMPY (XDS,IDXDS,SCR,IDXD,SCR,IDXD)
CALL PDER (SCR,IDXD,Y,4)
CALL PMPY (YDS,IDXDS,SCR,IDXD,SCR,IDXD)
CALL PADD (SUM,IDSUM,XDS,IDXDS,YDS,IDXDS)
DO 6 I=1,5
T=0.25*(I-1)
CALL PVAL (TEMP,T,SUM,IDSUM)
DLDT(I)=SQRT(TEMP)
6 CONTINUE

* fit polynomial to dl/dt
* evaluate matrix solution

CALL RSOLVE (5,5,A,DLDT,IP)
DO 7 I=1,5
STORCD(K,I)=DLDT(I)
7 CONTINUE
5 CONTINUE

* calculate weights

DO 8 I=1,NNODE

```

```

      DO 9 J=1,NNODE
      F(J)=0.0
  9  CONTINUE
      F(I)=1.0
      CALL SPFIT (TG,F,FELM,NNODE)
      DO 10 J=1,NELEM
      CF(1)=FELM(1,J)
      CF(2)=(FELM(3,J)-FELM(1,J)-FELM(2,J)/3.-FELM(4,J)/6.)
      CF(3)=FELM(2,J)/2.
      CF(4)=(FELM(4,J)-FELM(2,J))/6.
      DO 11 K=1,6
      CD(K)=STORCD(J,K)
  11 CONTINUE
      CALL PMPY (CG, IDG, CD, 5, CF, 4)
      CALL PINT (CGI, IDGI, CG, IDG)
      CALL PVAL (VAL0, 0.0, CGI, IDGI)
      CALL PVAL (VAL1, 1.0, CGI, IDGI)
      W(I)=W(I)+VAL1-VAL0
  10 CONTINUE
  8  CONTINUE

      RETURN
 101 CONTINUE
      WRITE (IPRIN,601) IP(5)

      STOP
 601 FORMAT (' ERROR - CONIWT - IP(5) = ',I5)
      END

C DECK CPFIT
      SUBROUTINE CPFIT (X, Z, CELEMS, NPTS)

*      CPFIT CREATED FROM SPFIT          E N HUBBLE    JUNE 1977
*      FITS CUBIC NON-PARAMETRIC SPLINE SEGMENTS
*          TO SET OF COMPLEX DATA POINTS

*      INPUTS
*          X      = ARRAY OF REAL INDEPENDENT VARIABLES
*          Z      = ARRAY OF COMPLEX DEPENDENT VARIABLES
*          NPTS  = NUMBER OF (X,Z) DATA POINTS

*      RETURN
*          CELEMS = ARRAY OF (NPTS-1) SEGMENTS IN FOLLOWING FORM
*                  ( (Z(I), D(I), Z(I+1), D(I+1)) ) WHERE
*          D      = ARRAY OF SECOND DERIVATIVES AT DATA POINTS

*      ARRAYS A,B,C ARE MAINLY SUB DIAG., DIAGONAL, AND SUPER DIAG.
*      D ARRAY IS THE RIGHT HAND SIDE OF MATRIX EQUATION
*      SECOND DERIVATIVES AT NODES ARE PLACED IN D ARRAY AFTER SOLUTION
*      SOLUTION TECHNIQUE IS GAUSSIAN ELIMINATION
*      BOUNDARY CONDITIONS SET BY EXTRAPOLATION OF SECOND DERIVATIVES

      COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
      2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
      2 SPTFIL,LACFIL,LAEFIL
      INTEGER   SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
      2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
      2 SPTFIL,LACFIL,LAEFIL

      COMPLEX Z, ZDD, STORE, D, CELEMS
      DIMENSION X(NPTS),Z(NPTS),CELEMS(4,NPTS)
      DIMENSION A(100), B(100), C(100), D(100)

      N = NPTS
      NL1 = N - 1
      NL2 = N - 2
      DO 50 I=2,N
      IF (X(I) .GT. X(I-1)) GO TO 50
      WRITE (IPRIN,888) X(I-1),X(I)
      GO TO 8888
  50  CONTINUE

```

```

100      IF (N .LE. 100) GO TO 100
      WRITE (IPRIN,999)
      N = 100
      CONTINUE
      IF (N .GT. 2) GO TO 125
      D(1) = (0.0, 0.0)
      D(2) = (0.0, 0.0)
      GO TO 375
125      CONTINUE
      IF (N .GT. 3) GO TO 150
      ZDD = 2.*((X(3)-X(2))*Z(1)+(X(2)-X(1))*Z(3)-(X(3)-X(1))*Z(2))
      . /((X(3)-X(2))*(X(2)-X(1))*(X(3)-X(1)))
      D(1) = ZDD
      D(2) = ZDD
      D(3) = ZDD
      GO TO 375
150      CONTINUE
      DO 200 I=1,N
      A(I) = 0.0
      B(I) = 0.0
      C(I) = 0.0
      D(I) = (0.0, 0.0)
200      CONTINUE
      *
      set up matrices (a tridiagonal structure)
      A(1) = (X(3)-X(2))/(X(3)-X(1))
      C(1) = 2.0
      B(1) = 1.0 - A(1)
      D(1) = 6.0*((Z(3)-Z(2))/(X(3)-X(2))-(Z(2)-Z(1))/.
      1 (X(2)-X(1)))/(X(3)-X(1))
      H = X(3) - X(2)
      DO 250 I=3,NL1
      HP = X(I+1) - X(I)
      C(I) = HP / (H+HP)
      B(I) = 2.0
      A(I) = 1.0 - C(I)
      D(I) = 6.0*((Z(I+1)-Z(I))/HP-(Z(I)-Z(I-1))/H)/(HP+H)
      H = HP
250      CONTINUE
      *
      set boundary conditions
      C(2) = (X(2)-X(1))/(X(3)-X(2))
      A(2) = 1.0
      B(2) = -1.0-C(2)
      C(2) = -A(2)*A(1)/B(1) + C(2)
      D(2) = (0.0, 0.0)
      C(N) = (X(N)-X(N-1))/(X(N-1)-X(N-2))
      A(N) = -1.0 - C(N)
      B(N) = 1.0
      D(N) = (0.0, 0.0)
      *
      solve equations
      II = 1
      DO 300 I=1,NL2
      I1 = I + 1
      I2 = I + 2
      AUGH = ABS (B(I))
      IF (AUGH .LT. 1.OE-06) GO TO 275
      CONST = A(I1) / B(I)
      B(I1) = B(I1) - CONST*C(I)
      D(I1) = D(I1) - CONST*D(I)
      IF (I .NE. NL2) GO TO 300
      A(N) = A(N) - C(N)*C(I) / B(I)
      D(N) = D(N) - C(N)*D(I) / B(I)
      GO TO 300
      CONTINUE
      II = I + 1
      D(I) = D(I) / C(I)
      D(I1) = D(I1) - B(I1)*D(I)

```

```

B(I1) = A(I1)
A(I1) = 0.0
D(I2) = D(I2) - A(I2)*D(I)
A(I2) = 0.0
IF (I .NE. NL2) GO TO 300
A(N) = C(N)
300  CONTINUE
DET = B(NL1)*B(N) - C(NL1)*A(N)
STORE = D(N)
D(N) = (B(NL1)*D(N) - D(NL1)*A(N)) / DET
D(NL1) = (D(NL1)*B(N) - C(NL1)*STORE) / DET
IP = 0
DO 350 I=2,NL2
JI = N - I
IF (JI .EQ. IP) GO TO 350
IF (JI .EQ. II) GO TO 325
D(JI) = (D(JI)-C(JI)*D(JI+1))/B(JI)
GO TO 350
325  CONTINUE
IP = JI-1
STORE = D(JI)
D(JI) = D(IP)
D(IP) = (STORE - C(IP)*D(JI+1))/B(IP)
350  CONTINUE
D(1) = (D(1) - A(1)*D(3) - C(1)*D(2)) / B(1)

*      set up spline segments

375  CONTINUE
DO 400 I=1,NL1
I1 = I + 1
CELEMS(1,I) = Z(I)
CELEMS(2,I) = D(I)
CELEMS(3,I) = Z(I1)
CELEMS(4,I) = D(I1)
400  CONTINUE
99999  CONTINUE

88888  RETURN
88888  CONTINUE

STOP
888  FORMAT ('0 CPFIT-- X VALUES NOT ASCENDING', 2E16.8)
999  FORMAT ('0 CPFIT-- NPTS EXCEEDS 100. ONLY 99 SEGMENTS RETURNED')

END

C DECK CPINTG
SUBROUTINE CPINTG (SXA,SXB,SX,NPTS,SELEMS,AS,INTGS)

*      CPINTG CREATED FROM SPINTG
*      EVALUATES THE INTEGRAL OF A COMPLEX FUNCTION DEFINED BY
*      COMPLEX NON-PARAMETRIC SPLINE SEGMENTS

*      INPUTS
*      XA      = LOWER LIMIT OF INTEGRATION
*      XB      = UPPER LIMIT OF INTEGRATION
*      X       = ARRAY OF REAL INDEPENDENT VARIABLES
*      NPTS    = NUMBER OF VALUES IN X-ARRAY
*      CELEMS  = NON-PARAMETRIC SPLINE SEGMENTS GENERATED BY CPFIT
*      A       = CONSTANT FOR SPECIFIC INTEGRAL TO BE EVALUATED

*      RETURNS
*      INTG   = INTEGRAL OF F(X) * EXP(II*A*X)
*      IF A = 0.0 , THEN INTG = INTEGRAL OF F(X)

IMPLICIT REAL*8(A-H,O-Z)
COMPLEX SELEMS,SZA,SZB,SSA,SSB,INTGS,CTEMP
REAL SXA,SXB,SX,AS
COMPLEX*16 CELEMS, INTG, II, CINTG, SINTG, CISEG, SISEG, SEGINT,
ZA, ZB, SA, SB, Z1, Z2, S1, S2, ZAA, ZBB, ZCC, PPP, QQQ
DIMENSION X(25),CELEMS(4,25),SX(NPTS),SELEMS(4,NPTS)

```

```

EXTERNAL EXP

DATA II / (0.D+0, 1.D+0) /

CINTG = (0.D+0, 0.D+0)
SINTG = (0.D+0, 0.D+0)
CALL CPLVAL (SX, NPTS, SELEMS, SXA, SZA, SSA, IA)
CALL CPLVAL (SX, NPTS, SELEMS, SXB, SZB, SSB, IB)
XA=DBLE(SXA)
XB=DBLE(SXB)
DO 5 JJ=1,NPTS
X(JJ)=DBLE(SX(JJ))
DO 6 JI=1,4
CTEMP=SELEMS(JI,JJ)
6 CELEMS(JI,JJ)=DCMPLX(DBLE(REAL(CTEMP)),DBLE(AIMAG(CTEMP)))
5 CONTINUE
ZA=DCMPLX(DBLE(REAL(SZA)),DBLE(AIMAG(SZA)))
ZB=DCMPLX(DBLE(REAL(SZB)),DBLE(AIMAG(SZB)))
SA=DCMPLX(DBLE(REAL(SSA)),DBLE(AIMAG(SSA)))
SB=DCMPLX(DBLE(REAL(SSB)),DBLE(AIMAG(SSB)))
A =DBLE(AS)
A2 = A * A
A3 = A * A2
A4 = A * A3
DO 500 I=IA,IB
IF (I .GT. IA) GO TO 100
X1 = XA
X2 = X(I+1)
Z1 = ZA
Z2 = CELEMS(3,I)
S1 = SA
S2 = CELEMS(4,I)
GO TO 300
100 CONTINUE
IF (I .LT. IB) GO TO 200
X1 = X(I)
X2 = XB
Z1 = CELEMS(1,I)
Z2 = ZB
S1 = CELEMS(2,I)
S2 = SB
GO TO 300
200 CONTINUE
X1 = X(I)
X2 = X(I+1)
Z1 = CELEMS(1,I)
Z2 = CELEMS(3,I)
S1 = CELEMS(2,I)
S2 = CELEMS(4,I)
300 CONTINUE
XX = X2 - X1
IF (A .NE. 0.0) GO TO 400
SEGINT = (Z2+Z1) * XX / 2. - (S2+S1) * XX**3 / 24.
CINTG = CINTG + SEGINT
GO TO 500
400 CONTINUE
ZAA = (S2-S1) / (XX * 6.)
ZBB = S1 / 2.
ZCC = (Z2-Z1) / XX - (S2 + 2.*S1) * XX / 6.
AXX = A * XX
E = DSIN (AXX)
F = DCOS (AXX)
XX2 = XX * XX
XX3 = XX * XX2
P = (3.*A2*XX2 - 6.) / A4
Q = (A2*XX3 - 6.*XX) / A3
AA1 = F*P + E*Q + 6./A4
AA2 = E*P - F*Q
PP = (2.*XX) / A2
QQ = (A2*XX2 - 2.) / A3
BB1 = F*PP + E*QQ

```

```

BB2 = E*PP - F*QQ - 2./A3
XXA = XX / A
CC1 = (F-1.)/A2 + E*XXA
CC2 = E/A2 - F*XXA
DD1 = E/A
DD2 = (1.-F)/A
AX1 = A * X1
VV = DCOS (AX1)
UU = DSIN (AX1)
PPP = (AA1*ZAA + BB1*ZBB + CC1*ZCC + DD1*Z1)
QQQ = (AA2*ZAA + BB2*ZBB + CC2*ZCC + DD2*Z1)
SISEG = UU*PPP + VV*QQQ
CISEG = VV*PPP - UU*QQQ
CINTG = CINTG + CISEG
SINTG = SINTG + SISEG
500 CONTINUE
INTG = CINTG + II*SINTG
INTGS=CMPLX(REAL(INTG),REAL(AIMAG(INTG)))

      RETURN
      END

C DECK CPLVAL
SUBROUTINE CPLVAL (X, NPTS, CELEMS, X0, Z0, S0, IELM)

* CPLVAL CREATED FROM SPLVAL
* EVALUATES A COMPLEX NON-PARAMETRIC SPLINE

* INPUTS
*   X      = ARRAY OF REAL INDEPENDENT VARIABLES
*   NPTS  = NUMBER OF VALUES IN X-ARRAY
*   CELEMS= COMPLEX SPLINE SEGMENTS GENERATED BY CPPFIT
*   X0    = X-VALUE AT WHICH SPLINE IS TO BE EVALUATED

* RETURNS
*   Z0    = F(X0) = Z-VALUE EVALUATED AT X0
*   S0    = SECOND DERIVATIVE EVALUATED AT X0
*   IELM  = INDEX OF SPLINE SEGMENT CONTAINING X0

COMPLEX CELEMS, Z0, Z1, Z2, S0, S1, S2
DIMENSION X(NPTS),CELEMS(4,NPTS)

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL
INTEGER   SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL

N = NPTS
IF (X0.GE.X(1) .AND. X0.LE.X(N)) GO TO 100
100  WRITE (IPRIN,999) X0
      GO TO 99999
CONTINUE
DO 200 I=2,N
IF (X0 .GT. X(I)) GO TO 200
GO TO 300
200  CONTINUE
300  CONTINUE
      I = I - 1
      XX = X(I+1) - X(I)
      X1 = X0 - X(I)
      X2 = X(I+1) - X0
      XX6 = XX * XX / 6.0
      Z1 = CELEMS(1,I)
      Z2 = CELEMS(3,I)
      S1 = CELEMS(2,I)
      S2 = CELEMS(4,I)
      Z0 = (S1 * X2**3 + S2 * X1**3) / (6.0 * XX) +
. ((Z1 - S1*XX6) * X2 + (Z2 - S2*XX6) * X1) / XX
      S0 = (S1 + X2 + S2 + X1) / XX
      IELM = I

```

```

      RETURN
99999  CONTINUE

      STOP
999  FORMAT ('0 EXTRAPOLATION NOT ALLOWED.  X0 = ', E16.8)
      END

C DECK CSOLVE
SUBROUTINE CSOLVE ( N, NDIM, UL, B, X, IP )

*      SOLUTION OF COMPLEX LINEAR SYSTEM, A*X = B .

*      INPUT...
*      N = ORDER OF MATRIX.
*      NDIM = DECLARED DIMENSION OF ARRAY UL.
*      UL = TRIANGULARIZED MATRIX OBTAINED FROM "DECOMP".
*      B = COMPLEX RIGHT HAND VECTOR.
*      IP = PIVOT VECTOR OBTAINED FROM "DECOMP".
*      DO NOT USE SOLVE IF DECOMP HAS SET IP(N) = 0 .
*
*      OUTPUT...
*      X = COMPLEX SOLUTION VECTOR.

COMPLEX UL, B, X, T
INTEGER   NDIM, IP, I, K, KB, KM1, KP1, M, N, NM1
DIMENSION UL(NDIM,NDIM), B(NDIM), X(NDIM)
DIMENSION IP(NDIM)

DO 1000 K = 1, NDIM
X(K) = B(K)
1000  CONTINUE

IF ( N .EQ. 1 ) GO TO 1500
NM1 = N - 1
DO 1200 K = 1, NM1
KP1 = K + 1
M = IP(K)
T = X(M)
X(M) = X(K)
X(K) = T
DO 1100 I = KP1, N
X(I) = X(I) + UL(I,K)*T
1100  CONTINUE
1200  CONTINUE
DO 1400 KB = 1, NM1
KM1 = N - KB
K = KM1 + 1
X(K) = X(K)/UL(K,K)
T = -X(K)
DO 1300 I = 1, KM1
X(I) = X(I) + UL(I,K)*T
1300  CONTINUE
1400  CONTINUE
1500  CONTINUE
X(1) = X(1)/UL(1,1)
99999  CONTINUE

      RETURN
      END

C DECK CUBCO2
SUBROUTINE CUBCO2 ( SEG, CC )

*      CUBCO2 CREATED FROM CUBCO ( NAVSEC-N072 ) - A M REED JULY 1976
*
*      CONVERT CUBIC CURVE SEGMENT REPRESENTATION FROM ENDPOINT-TANGENT
*      FORM AS GIVEN IN THE ARRAY SEG TO CUBIC POLYNOMIAL COEFFICIENTS
*      IN THE ARRAY CC.  SET POLYNOMIAL COEFFICIENTS FOR THE EVALUATION
*      OF TANGENT VECTORS AND THE DX AND DY VALUES IN ARRAY CC.

```

```

*      INPUT
*      SEG = ( X(1), Y(1), DX(1), DY(1), X(2), Y(2), DX(2), DY(2) )
*
*      RETURN
*      CC = (AX,AY,BX,BY,CX,CY,DY,3AX,3AY,2BX,2BY,X2-X1,Y2-Y1)
*
*      DIMENSION SEG(8), CC(14)
*
DO 1000 I = 1, 2
D = SEG(I)
C = SEG(I+2)
DELTA = SEG(I+4) - D
A = SEG(I+6) + C - 2.0*DELTA
B = DELTA - A - C
CC(I) = A
CC(I+2) = B
CC(I+4) = C
CC(I+6) = D
CC(I+8) = 3.0*A
CC(I+10) = 2.0*B
CC(I+12) = DELTA
1000 CONTINUE
*
      RETURN
      END
*
C DECK DKWSLM
SUBROUTINE DKWSLM (KR,IC,IM,NPREDH,N,NDATA,DATA,INDXRL,INDXHD,
2 HEADNG,HDNG,LINEAR,SYMMET,SPINDX,TOINDX,IP,PROB,NUMH,
2 RM,ROLL)
*
COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
2 LRAOPR,ADRPR,ORGOPTN,GMNOM,KG,STATN(25),NSOFST(25),
2 NLEWF(25),HLFBTH(10,25),WTRLNE(10,25),BLEWF(25),TLEWF(25),
2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFB(10),YPTFB(10),
2 ZPTFB(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
2 STATNM,STATIS
CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
2 FBNUMB,PTNUMB,ORGOPTN
REAL KG
*
COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S.NNMU,FRNUM,VFS
INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8),
REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)
*
COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELG,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELG,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)
*
COMMON /INDEX/ PFIDX,LPFIDX,RMIDX,LRMIDX,SVIDX,LSVIDX
INTEGER LPFIDX,LRMIDX,LSVIDX
REAL PFIDX(235),RMIDX(183),SVIDX(3)
*
COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFILE,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL
INTEGER SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFILE,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL

```

```

2 COMMON /PHYSICO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

INTEGER HEADNG
REAL NUMH,NHMAX,NHMIN
LOGICAL LINEAR,SYMMET
CHARACTER*1 BLANK
CHARACTER*2 AC(2),AT,AVK
CHARACTER*3 FBT(3),PT(10)
CHARACTER*4 ACOND(3,2),METER,MET,FT,BS,SUNIT
CHARACTER*110 PARS,SEA
DIMENSION RM(8,24),RV(8,24),ROLL(13,64,4),DATA(432),INDXRL(25),
2 INDXHD(25),HEADNG(26),HDNG(24),SPINDX(9),
2 TOINDX(9),PROB(25,8,8),NUMH(25,8,8),ELM(4,8)
DIMENSION PRB(13),NHR(13)

EXTERNAL EXP

DATA FBT/'SLM','EMG','SBM'/
DATA PT /'P1','P2','P3','P4','P5','P6','P7','P8','P9','P10'/
DATA METER,MET,FT /'METE',' M. ',' FT.'/
DATA BLANK '/'/
DATA AC /'LC','SC'/
DATA ACOND /'LONG','CRES','TED ','SHOR','TCRE','STED'/

IF (FBCODE(IP) .EQ. 1) WRITE (PARS,3030) FBT(1),PT(IP),
2 XPTFB(1),YPTFB(1),ZPTFB(1),RDOT(IP)
3030 FORMAT (2A3,12X,30HSLAMMING IN NUMBER PER HOUR AT,4X,5HXFP =,
2 F6.2,3X,5HYCL =,F7.2,3X,5HZBL =,F7.2,5X,6HRDOT =,F6.2)
IF (FBCODE(IP) .EQ. 2) WRITE (PARS,3040) FBT(2),PT(IP),
2 XPTFB(1),YPTFB(1),ZPTFB(1)
3040 FORMAT (2A3,12X,31HEMERGENCE IN NUMBER PER HOUR AT,4X,5HXFP =,
2 F6.2,3X,5HYCL =,F7.2,3X,5HZBL =,F7.2,16X)
IF (FBCODE(IP) .EQ. 3) WRITE (PARS,3050) FBT(3),PT(IP),
2 XPTFB(1),YPTFB(1),ZPTFB(1)
3050 FORMAT (2A3,12X,33HSUBMERGENCE IN NUMBER PER HOUR AT,4X,5HXFP =,
2 F6.2,3X,5HYCL =,F7.2,3X,5HZBL =,F7.2,14X)
NSPIND = NVK + 1
NTOIND = NTMOD + 1
PRIDIR = 90.
SECDIR = 0.
JR = KR - 1
DO 300 IS=1,NSIGWH
CON = SIGWH(IS)*STATIS
K = 0
DO 200 ITO=1,NTMOD
DO 100 IV=1,NVK
K = K + 1
SWHMAX = .202*TMODAL(ITO)**2
IF (PUNITS(1) .EQ. METER) SWHMAX = SWHMAX*FTMETR
IF (SIGWH(IS) .GT. SWHMAX) GO TO 100

* relative motion

CALL FETCH (JR,IV,IT0,DATA,RMIDX,SPINDX,TOINDX,NDATA,LRMIDX,
2 NVK,NTMOD,RMSFIL)
L = 2*NPREDH
DO 10 IA=1,N
DO 10 IH=1,NPREDH
IF (IC .EQ. 1) TEMP = DATA(L+1)
IF (IC .EQ. 2) TEMP = DATA(L+2)
L = L + 2
RM(IA,IH) = TEMP*CON
10 CONTINUE

* relative velocity

```

```

2 CALL FETCH (KR,IV,ITO,DATA,RMIDX,SPINDX,TOINDX,N DATA,LRMIDX,
2 NVK,NTMOD,RMSFIL)
L = 2*NPREDH
DO 15 IA=1,N
DO 15 IH=1,NPREDH
IF (IC .EQ. 1) TEMP = DATA(L+1)
IF (IC .EQ. 2) TEMP = DATA(L+2)
L = L + 2
RV(IA,IH) = TEMP*CON
15 CONTINUE
NHEAD = 24
N1 = NHEAD + 1
DO 80 IH=1,N1
IF (IH .GT. NPREDH) GO TO 70
LH = INDXHD(IH)
IF (.NOT. LINEAR) GO TO 20
RELMOT = RM(1,IH)
RELVEL = RV(1,IH)
GO TO 50
20 KH = INDXRL(IH)
RLCALC = ROLL(KH,K,IS)
IF (RLCALC .GE. RLANG(1)) GO TO 30
RELMOT = RM(1,IH)
RELVEL = RV(1,IH)
GO TO 50
30 IF (RLCALC .LE. RLANG(NRANG)) GO TO 40
RELMOT = RM(NRANG,IH)
RELVEL = RV(NRANG,IH)
GO TO 50
40 CALL SPFIT (RLANG,RM(1,IH),ELM,NRANG)
CALL SPLVAL (RLANG,NRANG,ELM,RLCALC,RELMOT,DUM,IELM)
CALL SPFIT (RLANG,RV(1,IH),ELM,NRANG)
CALL SPLVAL (RLANG,NRANG,ELM,RLCALC,RELVEL,DUM,IELM)
50 IF (FBCODE(IP) .GT. 1) GO TO 60

* slamming
      T = FBDZV(IV,IP)
      ARG = (STATIS*T/RELMOT)**2 / 2
      PROBI = 0.
      IF (ARG .LE. 50.) PROBI = EXP(-ARG)
      ARG = (STATIS*RDOT(IP)/RELVEL)**2 / 2
      PROBV = 0.
      IF (ARG .LE. 50.) PROBV = EXP(-ARG)
      PROBS = PROBI*PROBV
      NUMSLM = 3600/(2*PI) * (RELVEL/RELMOT) * PROBS
      PROB(LH,ITO,IV) = PROBS
      NUMH(LH,ITO,IV) = NUMSLM
      GO TO 80

* emergence and submergence
60 F = FBDZV(IV,IP)
ARG = (STATIS*F/RELMOT)**2 / 2
PROBE = 0.
IF (ARG .LE. 50.) PROBE = EXP(-ARG)
NUMEMG = 3600/(2*PI) * (RELVEL/RELMOT) * PROBE
PROB(LH,ITO,IV) = PROBE
NUMH(LH,ITO,IV) = NUMEMG
GO TO 80
70 JH = INDXRL(IH)
PROB(IH,ITO,IV) = PROB(JH,ITO,IV)
NUMH(IH,ITO,IV) = NUMH(JH,ITO,IV)
80 CONTINUE
100 CONTINUE
IF (SIGWH(IS) .GT. SWEMAX) GO TO 200
NHMAX = NUMH(1,ITO,1)
NHMIN = NBMAX
DO 110 IV=1,NVK
DO 110 IH=1,NHEAD
TEMP = NUMH(IH,ITO,IV)
IF (TEMP .LT. NHMIN) NHMIN = TEMP

```

```

110 IF (TEMP .GT. NHMAX) NHMAX = TEMP
CONTINUE
ISIGWH = SIGWH(IS)*100
IF (ISIGWH .GE. 1000) WRITE (BS,3001) ISIGWH
IF (ISIGWH .LT. 1000) WRITE (BS,3002) ISIGWH
IF (ISIGWH .LT. 100) WRITE (BS,3003) ISIGWH
IF (ISIGWH .LT. 10) WRITE (BS,3004) ISIGWH
3001 FORMAT (I4)
3002 FORMAT (1H0,I3)
3003 FORMAT (2H00,I2)
3004 FORMAT (3H000,I1)
3000 FORMAT (1H0,I1)
3010 FORMAT (I2)
ITMODL = TMODAL(ITO) + .5
IF (ITMODL .LT. 10) WRITE (AT,3000) ITMODL
IF (ITMODL .GE. 10) WRITE (AT,3010) ITMODL
SUNIT = MET
IF (PUNITS(1) .NE. METER) SUNIT = FT
WRITE (SEA,3020) BS,AT,ACC(IC),SIGWH(IS),SUNIT,TMODAL(ITO),
2 (ACOND(I,IC),I=1,3),(STATNM(I),I=1,3)
3020 FORMAT (2HBR,A4,2A2,32H BRETSCHNEIDER SEAWAY - SIGWH =,F6.2,A4,
2 1OH TMODAL =,F6.2,7H SEC ,3A4,4X,3A4,7X)
* WRITE (SPTFIL,5022) PARS,SEA
* WRITE (SPTFIL,5026) NHMIN,NHMAX
5022 FORMAT (A110)
5026 FORMAT (1P2E15.4)
* WRITE (SPDFIL) ((NUMH(IH,ITO,IV),IV=1,NVK),IH=1,NHEAD)
200 CONTINUE

*      print number of occurrences in 1 hour for slamming, emergence,
*      and submergence

DO 250 IPAGE=1,2
IF (IPAGE.EQ.2 .AND. SYMMET) GO TO 250
WRITE (IPRIN,1000) TITLE
1000 FORMAT (1H1,22X,20A4)
IF (IC .EQ. 1) WRITE (IPRIN,1010)
IF (IC .EQ. 2) WRITE (IPRIN,1020)
1010 FORMAT (/58X,11HLONGCRESTED)
1020 FORMAT (/58X,12HSHORTCRESTED)
IF (PUNITS(1) .NE. METER) WRITE (IPRIN,1030) SIGWH(IS)
1030 FORMAT (45X,25HSIGNIFICANT WAVE HEIGHT =,F6.2,5H FEET)
IF (PUNITS(1) .EQ. METER) WRITE (IPRIN,1031) SIGWH(IS)
1031 FORMAT (45X,25HSIGNIFICANT WAVE HEIGHT =,F6.2,7H METERS)
WRITE (IPRIN,1032) (FBNAME(I,IP),I=1,5),XPTFBD(IP),YPTFBD(IP),
2 ZPTFBD(IP)
1032 FORMAT (/33X,5A4,3X,5HFFF =,F6.2,2X,5HYCL =,F7.2,2X,5HZBL =,F7.2)
IF (FBCODE(IP) .GT. 1) GO TO 120
WRITE (IPRIN,1033)
1033 FORMAT (/58X,8HSLAMMING)
IF (PUNITS(1) .EQ. METER) WRITE (IPRIN,1034) RDOT(IP)
1034 FORMAT (43X,20HTHRESHOLD VELOCITY =,F6.2,11H METERS/SEC)
IF (PUNITS(1) .NE. METER) WRITE (IPRIN,1035) RDOT(IP)
1035 FORMAT (43X,20HTHRESHOLD VELOCITY =,F6.2,9H FEET/SEC)
120 IF (FBCODE(IP) .EQ. 2) WRITE (IPRIN,1036)
1036 FORMAT (/58X,9HEMERGENCE)
IF (FBCODE(IP) .EQ. 3) WRITE (IPRIN,1037)
1037 FORMAT (/58X,11HSUBMERGENCE)
WRITE (IPRIN,1040)
1040 FORMAT (/43X,45HPROBABILITYX100 / NO. OF OCCURRENCES PER HOUR)
IF (IPAGE .EQ. 2) GO TO 225

*      starboard headings

1042 WRITE (IPRIN,1042) (HEADNG(IH),IH=1,13)
FORMAT (/58X,29HSHIP HEADING ANGLE IN DEGREES/4X,1HV,2X,2HT0,7X,
2 4HHEAD,47X,9HSTBD BEAM,46X,6HFOLLOW/10X,13(6X,I3))
DO 220 IV=1,NVK
IVK = VK(IV) + .5001
WRITE (AVK,1045) IVK
104b FORMAT (I2)
WRITE (IPRIN,1050)

```

```

1050 FORMAT (1H )
DO 220 ITO=1,NTMOD
      SWHMAX = .202*TMODAL(ITO)**2
      IF (PUNITS(1) .EQ. METER) SWHMAX = SWHMAX*FTMETR
      IF (SIGWH(IS) .GT. SWHMAX) GO TO 220
      IMP = TMODAL(ITO) + .5001
      DO 210 IH=1,13
      PRF(IH) = PROB(IH,ITC,IV)*100
      NHR(IH) = NUMH(IH,ITO,IV)
210  CONTINUE
      WRITE (IPRIN,1052) AVK,IMP,(PRB(IH),NHR(IH),IH=1,13)
1052 FORMAT (3X,A2,2X,I2,3X,13(F5.1,1H/,13))
      AVK = BLANK
220  CONTINUE
      GO TO 250

*   port headings

225  WRITE (IPRIN,1043) (HEADNG(IH),IH=14,26)
1043 FORMAT (/5X,29HSHIP HEADING ANGLE IN DEGREES/4X,1HV,2X,2HT0,7X,
2 4HHEAD,47X,9HPORT BEAM,46X,6HFOLLOW/10X,13(6X,13))
      DO 240 IV=1,NVK
      IVK = VK(IV) + .5001
      WRITE (AVK,1045) IVK
      WRITE (IPRIN,1050)
      DO 240 ITO=1,NTMOD
      SWHMAX = .202*TMODAL(ITO)**2
      IF (PUNITS(1) .EQ. METER) SWHMAX = SWHMAX*FTMETR
      IF (SIGWH(IS) .GT. SWHMAX) GO TO 240
      IMP = TMODAL(ITO) + .5001
      LH = 26
      DO 230 IH=1,13
      LH = LH - 1
      PRB(IH) = PROB(LH,ITO,IV)*100
      NHR(IH) = NUMH(LH,ITO,IV)
230  CONTINUE
      WRITE (IPRIN,1052) AVK,IMP,(PRB(IH),NHR(IH),IH=1,13)
      AVK = BLANK
240  CONTINUE
250  CONTINUE
300  CONTINUE

      RETURN
      END

C DECK EDMKSP
      FUNCTION EDMKSP (WE,LPP,V,EMD)

      REAL LPP

      EDMKSP = EMD/(1.+625.* (V/(WE*LPP))**2)

      RETURN
      END

C DECK ELTIME - computes elapsed time
      SUBROUTINE ELTIME (TS,ES)

      COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL
      INTEGER    SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL

      COMMON /SMPSYS/ FIS,AS,SIS,SOS,SDS,HALOS,DEV,PRN,SMPPS,SMPIS,
2 SMPPOS,SMPDS,SHPTYPS,SHIPS,VARS,CYCLS,TITLES,OPTION,LSIS,LSOS,
2 LSDS,LHALOS,LDEV,LPRN,LSMPPS,LSMPIS,LSMPOS,LSMPDS,LSHPTYPS,
2 LSHIPS,LTITLES
      CHARACTER*160 AS
      CHARACTER*80 FIS,SIS,SOS,SDS,TITLES
      CHARACTER*20 HALOS,DEV,PRN,SMPPS,SMPIS,SMPOS,SMPPDS,SHPTYPS

```

```

CHARACTER SHIPS*6,VARS*2,CYCLS*2
INTEGER*2 OPTION

CHARACTER*20 TS,ES

AS = '(I2,1X,I2,1X,F5.2)'
READ (TS,AS) IH,IM,BSEC
READ (ES,AS) JH,JM,ESEC

IF (ESEC .GE. BSEC) GO TO 10
ESEC = ESEC + 60.
JM = JM - 1
10 IF (JM .GE. IM) GO TO 20
JM = JM + 60.
JH = JH - 1
20 IF (JH.LT.IH) JH=JH+24

KH=JH-IH
KM=JM-IM
DELSEC=ESEC-BSEC
KS=DELSEC+.5

AS = '(//29X,"ELAPSED TIME"/16X,39("-")//'
2 '17X,I3," Hours",2X,I3, " Minutes",2X,I3," Seconds")'
WRITE (*,AS) KH,KM,KS
WRITE (TEXFIL,AS) KH,KM,KS

RETURN
END

C DECK EQMOTN
SUBROUTINE EQMOTN

COMMON /APPEND/ NBKSET,NBKSTN(2),BKIMAG(2),BKFS(2),BKAS(2),
2 BKWD(2),BKSTN(10,2),BKHB(10,2),BKLNTB,BKWDTH,
2 BKWL(10,2),BKAN(10,2),NSKSET,SKIMAG(2),SKFLS(2),SKALS(2),
2 SKAUS(2),SKHB(2),SKFLWL(2),SKALWL(2),SKAUWL(2),NRDSET,RDIMAG(2),
2 RDRFS(2),RDRAS(2),RDRHB(2),RDRFWL(2),RDRAWL(2),RDTFS(2),RDTAS(2),
2 RDTHB(2),RDTFWL(2),RDTAWL(2),NSBSET,SBIMAG(2),SOBRFS(2),SOBRAS(2),
2 SOBRHB(2),SOBRFW(2),SOBRAW(2),SIBRFS(2),SIBRAS(2),SIBRHB(2),
2 SIBRFW(2),SIBRAW(2),SBTFS(2),SBTAS(2),SETHB(2),SBTFWL(2),
2 SBTAWL(2),NFNSET,FNIMAG(2),FNRF(2),FNRAS(2),
2 FNRHB(2),FNRFWL(2),FNRAWL(2),FNTFS(2),FNTAS(2),FNTHB(2),
2 FNTFWL(2),FNTAWL(2),NEXPRD,ENRDO(8),ENRDS(8)

COMMON /CH3D/ ISIGMA,SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI,
2 IMMIN,IMMAX,IMDEL,LMIN,LMAX
REAL SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI(4)
INTEGER ISIGMA,IMMIN,IMMAX,IMDEL,LMIN,LMAX

COMMON /DATIMP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
2 LRAOPR,ADRPR,ORGOPTN,GMNOM,KG,STATN(25),NSOFST(25),
2 NLEWF(25),HLFBTH(10,25),WTRLNE(10,25),BLEWF(25),TLEWF(25),
2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
2 ZPTLUC(10),NBB,FBNUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
2 ZPTFBD(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
2 STATNM,STATIS
CHARACT'R#4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
2 FBNUMB,PTNUMB,ORGOPTN
REAL KG

COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

COMMON /FINCON/ IACTFN,IFCLCS,FGAIN(8),FK(3),FA(3),FB(3),
2 FCLCS(8,2)

COMMON /GEOM/ X,NSTATN,Y,Z,NUFSET,LPP,BEAM,DRAFT,LCF,

```

```

1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,PSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCB,GM,DELGM,NERLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GM..
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /HULL/ A26

COMMON /IO/ SYSFIL,POTFIL,CCFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEOFIL
INTEGER SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEOFIL

COMMON /PHYSICO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

COMMON /RLDBK/ PSUR(25),BMK(25),DK(25),CAK(25),HQ,HSPAN,HMNCHD,
2 HAREA,HXCP,HYCP,HZCP,HGAMMA,HYHAT,HEAR,HLC,S,RQ(2),RSPAN(2),
2 RMNCHD(2),RAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),
2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
2 SYCP(2),SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2),BQ(2),
2 BSPAN(2),BMNCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),
2 PQ(2,2),PSPAN(2,2),PMNCHD(2,2),PAREA(2,2),PXCP(2,2),PYCP(2,2),
2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),PLCS(2,2),
2 STADMP(10),SHPDMP(10,8),ENCON,WPHI,TPHI,WMEIM(4,9),SFELM(4,9,8),
2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
2 ENWM,ENSF(8,8),ENRE(8),ENPE(8),ENFE(8),ENHE(8),ENBE(8),
2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(8),ENSL(8),ENBL(8),
2 ENSHP(8,8),RELM(4,9),ITS(25),RD(25),EDDY(8,25),RGB(25)
REAL RDBLK(2692)
EQUIVALENCE (PSUR(1),RDBLK(1))

COMMON /SMPSYS/ FIS,AS,SIS,SL,SDS,HALOS,DEV,PRN,SMPPS,SMPIS,
2 SMPOS,SMPDS,SHPTYPS,SHIPS,VARS,CYCLS,TITLES,OPTION,LSIS,LSOS,
2 LSDS,LHALOS,LDEV,LPRN,LSMPPS,LSMPIS,LSMPOS,LSMPDS,LSHPTYPS,
2 LSHIPS,LTITLES
CHARACTER*160 AS
CHARACTER*80 FIS,SIS,SOS,SDS,TITLES
CHARACTER*20 HALOS,DEV,PRN,SMPPS,SMPIS,SMPUS,SMPDS,SHPTYPS
CHARACTER SHIPS*6,VARS*2,CYCLS*2
INTEGER*2 OPTION

COMMON /STATE/ LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL
LOGICAL LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL

COMMON /TELEM/ TELEM
COMPLEX TELEM(4,9,10)

COMPLEX TV(3,3),TL(3,3),EXCV(3),EXCL(3),HJV(3,30),HJL(3,30),
2 H7(30),TLG(3,3),EXCLG(3),MOTV(3,30),MOTLG(3),TLGC(3,3),
2 EXCLGC(3),MOTL(3,30,8),UL(3,3)
COMPLEX ZERO,TAF(3),CTEMP
DIMENSION T44T(8)
CHARACTER*4 METER
REAL OMEGAE(30)
INTEGER IP(3)

DATA METER,EPS /'Mete',0.001/

```

```

* solve equations of motion

ZERO = (0.,0.)

FIS = SDS(1:LSDS) //' .SCR'
OPEN (UNIT=SCRFILE,FILE=FIS,FORM='UNFORMATTED',STATUS='UNKNOWN')
READ (SCRFILE) RDBLK
CLOSE (UNIT=SCRFILE)

FIS = SDS(1:LSDS) //' .COF'
OPEN (UNIT=CUFFILE,FILE=FIS,FORM='UNFORMATTED',STATUS='UNKNOWN')
READ (CUFFILE) TELEM

FIS = SDS(1:LSDS) //' .ORG'
OPEN (UNIT=ORGFILE,FILE=FIS,FORM='UNFORMATTED',STATUS='UNKNOWN')

FIS = SDS(1:LSDS) //' .LAC'
OPEN (UNIT=LACFILE,FILE=FIS,FORM='UNFORMATTED',STATUS='UNKNOWN')

FIS = SDS(1:LSDS) //' .LAE'
OPEN (UNIT=LAEFILE,FILE=FIS,FORM='UNFORMATTED',STATUS='UNKNOWN')

FACT = FTMETR
IF (PUNITS(1) .NE. METER) FACT = 1
NMU = 13
VKINC = VK(2) - VK(1)

WRITE (ORGFILE) TITLE,NVK,NMU,NOMEGA,OMEGA,NRANG,RLANG,VRT,LAT,
2 ADDRES,LPP,BEAM,DRAFT,DISPLM,GM,DELGM,KG,KROLL,LCB,GRAV,RHO,
2 VKDES,VKINC,DBLWL

DO 300 IV=1,NVK
V = VFS(IV)
FRNO = FRNUM(IV)
NMU = NNMU(IV)
DO 200 IH=1,NMU
HDNG = MU(IH,IV)
SINMU = SIN(HDNG)
COSMU = COS(HDNG)
HDNG = HDNG*RADDEG
ICLIP = 0
IF (ABS(HDNG-EPS) .LE. 90.) ICLIP = 1
IF (ICLIP .EQ. 1) CALL LIMIT (XLIM,YLIM,PSILIM,HDNG,FRNO,DEGRAD,
2 FACT)
DO 50 IW = 1,NOMEGA
READ (CUFFILE) OMEGAE(IW),TV,TL,EXCV,EXCL,(HJV(I,IW),I=1,3),
2 (HJL(I,IW),I=1,3),H7(IW)
WE = OMEGAE(IW)
WE2 = WE*WE
A22 = REAL(TL(1,1))/(-WE2)
B22 = AIMAG(TL(1,1))/WE
A26V = REAL(TL(1,3))/(-WE2)
A26 = A26V - (V/WE2)*B22
CALL FINTSP (OMEGAE(IW))
CALL INERST (OMEGAE(IW),TV,TL)
IF (.NOT. VRT) GO TO 10
CALL SOLVE (3,TV,EXCV,MOTV(1,IW),UL,IP,IPRIN)
IF (ICLIP.EQ.1 .AND. OMEGAE(IW).LT.0.20)
2 CALL CLIP (XLIM,MOTV(1,IW),MOTV(1,IW))
10 IF (.NOT. LAT) GO TO 50
IF (HDNG.GT.EPS .AND. ABS(HDNG-180.).GT.EPS) GO TO 30
DO 20 J=1,3
DO 20 IA=1,NRANG
MOTL(J,IW,IA) = (0.0,0.0)
20 CONTINUE
GO TO 60
30 CALL TRNLAT (VCG,TL,EXCL,TLG,EXCLG)
CALL RDEVAL (IV,OMEGA(IW),OMEGAE(IW),NRANG,TLG,EXCLG,TLGC,EXCLGC,
2 T44T)
IF (IACTFN .EQ. 0) GO TO 34

```

```

*      add active fin coefficients

    OMGE = OMEGAE(IW)
    OMGE2 = OMGE*OMGE
    CALL ACTFIN (IV,ZERO,V,OMGE,OMGE2,TAF)
    DO 32 I=1,3
    TLGC(I,2) = TLGC(I,2) + FGAIN(IV)*TAF(I)
32  CONTINUE
34  CTEMP = TLGC(2,2)

    IF (IW.EQ.1) WRITE (LAEFIL) VK(IV),HDNG
    WRITE (LAEFIL) EXCLGC

*      IF (IW.EQ.1) WRITE (LAEFIL,1170) VK(IV),HDNG
*      WRITE (LAEFIL,1180) EXCLGC
* 1170   FORMAT(8F8.3)
* 1180   FORMAT(1P6E13.4)

    IF (IH.NE.7) GO TO 43

    IF (IW.EQ.1) WRITE (LACFIL) VK(IV),HDNG,RLANG
    WRITE (LACFIL) OMEGA(IW),OMEGAE(IW)
    WRITE (LACFIL) T44T
    DO 37 J=1,3
37  WRITE (LACFIL) (TLGC(I,J),I=1,3)
    WRITE (LACFIL) EXCLGC
43  CONTINUE

*      IF (IW.EQ.1) WRITE (LACFIL,1170) VK(IV),HDNG,RLANG
*      WRITE (LACFIL,1170) OMEGA(IW),OMEGAE(IW)
*      WRITE (LACFIL,1180) T44T
*      DO 37 J=1,3
* 37  WRITE (LACFIL,1180) (TLGC(I,J),I=1,3)
*      WRITE (LACFIL,1180) EXCLGC
* 43  CONTINUE

*      add viscous/bilgekeel eddy damping

    DO 40 IA=1,NRANG
    TLGC(2,2) = CTEMP + II*T44T(IA)
    CALL SOLVE (3,TLGC,EXCLGC,MOTLG,UL,IP,IPRIN)
    IF (IH.EQ.7) WRITE (LACFIL) MOTLG
*      IF (IH.EQ.7) WRITE (LACFIL,1180) MOTLG

    CALL RVSLAT (VCG,MOTLG,MOTL(1,IW,IA))
    IF (ICLIP .EQ. 0) GO TO 40
    IF (OMEGAE(IW) .GE. 0.20) GO TO 40
    CALL CLIP (YLIM,MOTL(1,IW,IA),MOTL(1,IW,IA))
    CALL CLIP (PSILIM,MOTL(3,IW,IA),MOTL(3,IW,IA))
40  CONTINUE
50  CONTINUE
    WRITE (ORGFIL) VK(IV),HDNG,OMEGAE
    IF (VRT) WRITE (ORGFIL) MOTV
    IF (LAT) WRITE (ORGFIL) MOTL
    IF (ADDRES) WRITE (ORGFIL) HJV,HJL,H7
200 CONTINUE
300 CONTINUE

    CLOSE (UNIT=COFFIL)
    CLOSE (UNIT=ORGFIL)
    CLOSE (UNIT=LACFIL)
    CLOSE (UNIT=LAEFIL)

    RETURN
    END

C DECK EXFOR
    SUBROUTINE EXFOR (OMEGA,OMEGAE,FXV,FYL,HJV,HJL,H7,F3,H3)

*      calculates exciting forces and corresponding loads data

```

```

COMMON /CH3D/ ISIGMA,SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI,
2 IMMIN,IMMAX,IMDEL,LMIN,LMAX
REAL SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI(4)
INTEGER ISIGMA,IMMIN,IMMAX,IMDEL,LMIN,LMAX

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /PELEM/ PELEM
COMPLEX PELEM(4,1000)

COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

COMMON /STATE/ LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL
LOGICAL LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL

COMMON /WGHTS/ WTDL,NORM
REAL WTDL(10,25),NORM(4,10,25)

REAL NORM2,NORM3
COMPLEX TEMP
INTEGER PSTORE
COMPLEX CIV(25,3),CIL(25,3),ELEMS(4,25)
COMPLEX CT(3)
COMPLEX HTV(25,3),HTL(25,3),HT7(25),HT(3),HT3,HT2
COMPLEX PHI2D(4),FXV(3),FXL(3),HJV(3),HJL(3),H7
COMPLEX CEP,F3(25),H3(25),TF3,TH3

EXTERNAL EXP

TEST=0.005*TPI/LPP
W = OMEGA
WN = W*W/GRAV
ARGLI = - WN*COSMU
IF (ABS(ARGLI) .LE. TEST) ARGLI = 0.
PSSTORE=1
DO 1 K=1,NSTATN
NNODE=NOFSET(K)
DO 10 I=1,3
CIL(K,I)=(0.0,0.0)
CIV(K,I)=(0.0,0.0)
IF (.NOT. ADDRES) GO TO 100
HTV(K,I)=(0.,0.)
HTL(K,I)=(0.,0.)
100 CONTINUE
10 CONTINUE
IF (.NOT. LOADS) GO TO 210
F3(K)=(0.,0.)
H3(K)=(0.,0.)
210 CONTINUE
IF (ADDRES) HT7(K)=(0.,0.)
IF (NNODE.LT.2) GO TO 1
DO 2 J=1,NNODE
EK2=EXP(WN*Z(J,K))
DO 9 IM=IMMIN,IMMAX,IMDEL
TEMP=(0.0,0.0)

```

```

DO 19 I=1,4
TEMP=TEMP+WTSI(I)*PELEM(I,PSTORE)
19 CONTINUE
PSTORE=PSTORE+1
PHI2D(IM)=TEMP
9 CONTINUE
ARG=WN*Y(J,K)*SINMU
CARG=COS(ARG)
SARG=SIN(ARG)
NORM2=NORM(2,J,K)
NORM3=NORM(3,J,K)
IF (.NOT. VRT) GO TO 3
TOD=NORM3*CARG-NORM2*SINMU*SARG
CT(1)=EKZ*(GRAV*NORM(1,J,K)*CARG+II*W*TOD*PHI2D(1))
CT(2)=EKZ*(GRAV*NORM3*CARG+II*W*TOD*PHI2D(3))
IF (.NOT. LOADS) GO TO 220
TF3=EKZ*NORM3*CARG
TH3=EKZ*II*TOD*PHI2D(3)
F3(K)=F3(K)+WTDL(J,K)*TF3
H3(K)=H3(K)+WTDL(J,K)*TH3
220 CONTINUE
IF (.NOT. ADDRES) GO TO 90
HT3=-WN*TOD*PHI2D(3)
HT(1)=(II*W*NORM(1,J,K)*CARG-WN*TOD*PHI2D(1))*EKZ
HT(2)=(II*W*NORM3*CARG+HT3)*EKZ
HT(3)=HT(2)*X(K)-II*V/OMEGAE*HT3*EKZ
90 CONTINUE
CT(3)=-X(K)*CT(2)-W*V/OMEGAE*EKZ*TOD*PHI2D(3)
DO 4 I=1,3
CIV(K,I)=CIV(K,I)+WTDL(J,K)*CT(I)
IF (ADDRES) HTV(K,I)=HTV(K,I)+WTDL(J,K)*HT(I)
4 CONTINUE
IF (ADDRES) HT7(K)=HT7(K)+II*WN*WTDL(J,K)*PHI2D(3)*TOD*W/OMEGAE*
1 EXP(WN*Z(1,K)/2.)
3 CONTINUE
IF (.NOT. LAT) GO TO 6
TEV=NORM3*SARG+NORM2*SINMU*CARG
CT(1)=EKZ*(-II*GRAV*NORM2*SARG+W*TEV*PHI2D(2))
CT(2)=EKZ*(-II*GRAV*NORM(4,J,K)*SARG+W*TEV*PHI2D(4))
IF (.NOT. ADDRES) GO TO 30
HT2=-II*WN*TEV*PHI2D(2)
HT(1)=(-W*NORM2*SARG+HT2)*EKZ
HT(2)=(-W*NORM(4,J,K)*SARG-II*WN*TEV*PHI2D(4))*EKZ
HT(3)=HT(1)*X(K)+II*V/OMEGAE*HT2*EKZ
30 CONTINUE
CT(3)=X(K)*CT(1)-II*W*V/OMEGAE*TEV*PHI2D(2)
DO 7 I=1,3
CIL(K,I)=CIL(K,I)+WTDL(J,K)*CT(I)
IF (ADDRES) HTL(K,I)=HTL(K,I)+WTDL(J,K)*HT(I)
7 CONTINUE
IF (ADDRES) HT7(K)=HT7(K)+II*WN*SINMU*WTDL(J,K)*PHI2D(2)*TEV*W*
2 OMEGAE*EXP(WN*Z(1,K)/2.)
6 CONTINUE
2 CONTINUE
2 CONTINUE
IF (.NOT. LOADS) GO TO 230
* sectional froude-kriloff "force", f3 w/o cexp(-ii*k*x*cos(mu))
F3(K)=2*GRAV*F3(K)
* sectional diffraction "force" , h3 w/o cexp(-ii*k*x*cos(mu))
H3(K)=2*W*H3(K)
230 CONTINUE
1 CONTINUE
IF (.NOT. VRT) GO TO 11
DO 13 I=1,3
CALL CPFIT(X,CIV(1,I),ELEMS,NSTATN)
CALL CPINTG(X(1),X(NSTATN),X,NSTATN,ELEMS,ARGLI,FXV(I))
FXV(I)=2.0*RHO*FXV(I)
IF (.NOT. ADDRES) GO TO 130

```

```

    CALL CPFIT (X,HTV(1,I),ELEMS,NSTATN)
    CALL CPINTG (X(1),X(NSTATN),X,NSTATN,ELEMS,ARGLI,HJV(I))
130  CONTINUE
13  CONTINUE
11  CONTINUE
    IF (.NOT. LAT) GO TO 12
    DO 14 I=1,3
    CALL CPFIT (X,CIL(1,I),ELEMS,NSTATN)
    CALL CPINTG (X(1),X(NSTATN),X,NSTATN,ELEMS,ARGLI,FXL(I))
    FXL(I)=2.0*RHO*FXL(I)
    IF (.NOT. ADDRES) GO TO 140
    CALL CPFIT (X,HTL(1,I),ELEMS,NSTATN)
    CALL CPINTG (X(1),X(NSTATN),X,NSTATN,ELEMS,ARGLI,HJL(I))
140  CONTINUE
14  CONTINUE
12  CONTINUE
    IF (ADDRES) CALL CPFIT (X,HT7,ELEMS,NSTATN)
    IF (ADDRES) CALL CPINTG (X(1),X(NSTATN),X,NSTATN,ELEMS,0.,H7)

    RETURN
    END

C DECK EXP
    FUNCTION EXP(X)

*   avoid underflow with F77L EXP routine

    IF(X.LT.(-50))THEN
        EXP=0.
    ELSE
        EXP=DEXP(X)
    ENDIF

    RETURN
    END

C DECK EXPINT
    SUBROUTINE EXPINT (X,Y,E,C,S,RA,RB,CIN,SON)

*   this subroutine computes the frequency-dependent part(principal-
*   value integral) of a pulsating source in or below free surface
*   which can be expressed in terms of exponential integral.

    DIMENSION F(5),D(5)

    EXTERNAL EXP

    DATA (F(I),I=1,5)/0.52175561,0.39866681,0.07594245,
1  0.003611758,0.000023369972/
    DATA (D(I),I=1,5)/0.26356032,1.4134031,3.5964258,
1  7.08581,12.640801/
    DATA Q, GAMMA / 3.1415926535897, 0.57721566490153 /
    DATA TEST5,TEST6,TEST7,TEST8 /1.E-05,1.E-06,1.E-07,1.E-08/

    AT=ATAN2(X,Y)
    ARG=AT-0.5*Q
    IF(Y.GT.50) THEN
        E=0.
    ELSE
        E=EXP(-Y)
    ENDIF
    C=COS(X)
    S=SIN(X)
    R=X**2+Y**2
    AL=0.5* ALOG(R)
    A=-Y
    B=-X
    IF (A .GE. 0.0) GO TO 78
    IF (B .EQ. 0.0) GO TO 79
78    IF (R .GE. 100.) GO TO 10
79    TEST = TEST5
    IF (R .LT. 4.0) TEST = TEST7

```

```

IF (R .LT. 2.0) TEST = TEST6
IF (R .LT. 1.0) TEST = TEST5
SUMC=GAMMA+AL+Y
SUMS=AT+X
TC=Y
TS=X
DO 1 K=1,500
TO=TC
COX=K
CAY=K+1
FACT=COX/CAY**2
TC=FACT*(Y*TC-X*TS)
TS=FACT*(Y*TS+X*TO)
SUMC=SUMC+TC
SUMS=SUMS+TS
IF (K .GE. 500) GO TO 3
IF ((ABS(TC)+ABS(TS)) .GT. TEST) GO TO 1
3 CIN=E*(C*SUMC+S*SUMS)
SON=E*(S*SUMC-C*SUMS)
GO TO 4
1 CONTINUE
10 G1=0.
G2=0.
DO 20 I=1,5
DEN=(-Y+D(I))**2+X**2
GA=F(I)*(-Y+D(I))/DEN
GB=F(I)*(-X)/DEN
G1=G1+GA
G2=G2+GB
CIN=E*Q*S-G1
SON=-(E*Q*S+G2)
4 RA=AL-CIN
RB=ARG+SON

RETURN
END

C DECK FETCH
SUBROUTINE FETCH (IRESP,IVK,ITO,DATA,RMIDX,SPINDX,TOINDX,NDATA,
2 LRMIDX,NVK,NTMOD,RMSFIL)

INTEGER RMSFIL
DIMENSION DATA(NDATA),RMIDX(LRMIDX),SPINDX(NVK+1),TOINDX(NTMOD+1)

* change for VAX/VMS version
* CDC CALL READMS (RMSFIL,SPINDX,NSPIND,IRESP)
* CDC CALL STINDX (RMSFIL,SPINDX,NSPIND)
* CDC CALL READMS (RMSFIL,TOINDX,NTOIND,IVK)
* CDC CALL STINDX (RMSFIL,TOINDX,NTOIND)
* CDC CALL READMS (RMSFIL,DATA,NDATA,ITO)
* CDC CALL STINDX (RMSFIL,SPINDX,NSPIND)
* CDC CALL STINDX (RMSFIL,RMIDX,LRMIDX)

INDEX = NTMOD * NVK * (IRESP - 2) + NTMOD * (IVK - 1) + ITO + 3
READ (RMSFIL,REC=INDEX) DATA

RETURN
END

C DECK FIG10
FUNCTION FIG10 (GDB)

* generates function of figure 10 of TANAKA,
* J. ZOSEW KIOKAI, V. 109, 1961

DIMENSION GKDB(6),RFORE(6)

GKDB(1)=1.2
GKDB(2)=1.4
GKDB(3)=1.6
GKDB(4)=1.8

```

```

GKDB(5)=2.0
GKDB(6)=2.05
RFORE(1)=1.0
RFORE(2)=0.6
RFORE(3)=0.34
RFORE(4)=0.15
RFORE(5)=0.04
RFORE(6)=0.0
IF (GDB-2.05) 22,23,23
23 CONTINUE
RBIL=0.0
GO TO 24
22 CONTINUE
DO 25 J=2,6
ITEMP = J
IF (GDB-GKDB(J)) 26,26,25
25 CONTINUE
26 CONTINUE
J = ITEMP
RBIL=(RFORE(J)-RFORE(J-1))/(GKDB(J)-GKDB(J-1))*(GDB-GKDB(J-1))
2 +RFORE(J-1)
24 CONTINUE
FIG10=RBIL

RETURN
END

```

```

C DECK FIG11
FUNCTION FIG11 (BDG)
DIMENSION BAFT(5),CAFT(5)

```

* generates function of figure 11 of TANAKA,
* J. ZOSEN KIOKAI, V. 109, U961

```

BAFT(1)=1.0
BAFT(2)=1.25
BAFT(3)=1.5
BAFT(4)=2.0
BAFT(5)=2.25
CAFT(1)=0.22
CAFT(2)=0.24
CAFT(3)=0.3
CAFT(4)=0.5
CAFT(5)=0.63
DO 33 J=2,5
ITEMP = J
IF (BDG-BAFT(J)) 34,34,33
33 CONTINUE
34 CONTINUE
J = ITEMP
FIG11=(CAFT(J)-CAFT(J-1))/(BAFT(J)-BAFT(J-1))*  

+ (BDG-BAFT(J-1))+CAFT(J-1)

RETURN
END

```

```

C DECK FIG56
FUNCTION FIG56 (THM,BDG)

```

* generates function of figures 5 and 6 of TANAKA,
* J. ZOSEN KIOKAI, VOL. 109, 1961

```

DIMENSION F1(15),BDKG(15)

IF (THM-0.1745) 3,3,4
3 CONTINUE
F1(1)=0.455
F1(2)=0.52
F1(3)=0.42
F1(4)=0.35
F1(5)=0.52

```

```

4 GO TO 5
CONTINUE
IF (THM-0.2618) 6,6,7
6 CONTINUE
FAC=(THM-0.1745)/(0.2618-0.1745)
F1(1)=(0.32-0.455)*FAC+0.455
F1(2)=(0.34-0.52)*FAC+0.52
F1(3)=(0.29-0.42)*FAC+0.42
F1(4)=(0.31-0.35)*FAC+0.35
F1(5)=(0.48-0.52)*FAC+0.52
GO TO 5
7 CONTINUE
IF (THM-0.3491) 8,9,9
8 CONTINUE
FAC=(THM-0.2618)/(0.3491-0.2618)
F1(1)=(0.25-0.32)*FAC+0.32
F1(2)=(0.25-0.34)*FAC+0.34
F1(3)=(0.22-0.29)*FAC+0.29
F1(4)=(0.28-0.31)*FAC+0.31
F1(5)=(0.45-0.48)*FAC+0.48
GO TO 5
9 CONTINUE
F1(1)=0.25
F1(2)=0.25
F1(3)=0.22
F1(4)=0.28
F1(5)=0.45
5 CONTINUE
F1(6)=0.63
F1(7)=0.63
F1(8)=0.59
F1(9)=0.53
F1(10)=0.4
F1(11)=0.35
F1(12)=0.32
F1(13)=0.3
DO 1 I=1,5
BDKG(I)=1./(60.-I*10.)
1 CONTINUE
BDKG(6)=1./5.
DO 2 I=7,13
BDKG(I)=0.5+0.5*(I-7)
2 CONTINUE
DO 27 J=2,13
ITEMP = J
IF (BDG-BDKG(J)) 28,28,27
27 CONTINUE
28 CONTINUE
J = ITEMP
FONE=(F1(J)-F1(J-1))/(BDKG(J)-BDKG(J-1))*(BDG-BDKG(J-1))+F1(J-1)
FIG56=FONE
RETURN
END

```

C DECK FIG7
FUNCTION FIG7 (THM)

* generates function of figure 7 of TANAKA,
* J. ZOSEN KIOKAI, VOL. 109, 1961

```

10 IF (THM-0.0873) 10,10,11
CONTINUE
AEX=10.6
GO TO 12
11 CONTINUE
IF (THM-0.1745) 13,13,14
13 CONTINUE
AEX=(7.66-10.6)/(0.1745-0.0873)*(THM-0.0873)+10.6
GO TO 12
14 CONTINUE
IF (THM-0.2618) 15,15,16

```

```

15  CONTINUE
    AEX=(6.34-7.66)/(0.2618-0.1745)*(THM-0.1745)+7.66
    GO TO 12
16  CONTINUE
    AEX=(6.28-6.34)/(0.3491-0.2618)*(THM-0.2618)+6.34
12  CONTINUE
    FIG7=AEX

    RETURN
    END

C DECK FIG8
FUNCTION FIG8(RDD,ALF)

*   generates function of figure 8 of tanaka,
*   J. ZOSED KIOKAI, V. 109, 1961

    DIMENSION ALF2(5),F2(5)

    ALF2(1)=0.0
    ALF2(2)=0.0873
    ALF2(3)=0.1745
    ALF2(4)=0.3491
    ALF2(5)=0.5235
    F2(1)=1.
    IF (RDD) 44,44,45
44  CONTINUE
    F2(2)=0.855
    F2(3)=0.765
    F2(4)=0.682
    F2(5)=0.646
    GO TO 46
45  CONTINUE
    IF (RDD-0.0571) 47,47,48
47  CONTINUE
    F2(2)=(0.745-0.855)/0.0571*RDD+0.855
    F2(3)=(0.670-0.765)/0.0571*RDD+0.765
    F2(4)=(0.745-0.682)/0.0571*RDD+0.682
    F2(5)=(0.915-0.646)/0.0571*RDD+0.646
    GO TO 46
48  CONTINUE
    IF (RDD-0.1142) 49,49,50
49  CONTINUE
    F2(2)=0.74
    F2(3)=(0.72-0.670)/(0.1142-0.0571)*(RDD-0.0571)+0.67
    F2(4)=(0.89-0.745)/(0.1142-0.0571)*(RDD-0.0571)+0.745
    F2(5)=(1.34-0.915)/(0.1142-0.0571)*(RDD-0.0571)+0.915
    GO TO 46
50  CONTINUE
    IF (RDD-0.1713) 51,51,52
51  CONTINUE
    F2(2)=(0.70-0.74)/(0.1713-0.1142)*(RDD-0.1142)+0.74
    F2(3)=0.72
    F2(4)=(1.20-0.89)/(0.1713-0.1142)*(RDD-0.1142)+0.89
    F2(5)=(1.94-1.34)/(0.1713-0.1142)*(RDD-0.1142)+1.34
    GO TO 46
52  CONTINUE
    F2(2)=0.7
    F2(3)=0.72
    F2(4)=1.2
    F2(5)=1.94
46  CONTINUE
    DO 53 J=2,5
    ITEMP = J
    IF (ALF-ALF2(J)) 54,54,53
53  CONTINUE
54  CONTINUE
    J = ITEMP
    F2ALF=(F2(J)-F2(J-1))/(ALF2(J)-ALF2(J-1))*(ALF-ALF2(J-1))+F2(J-1)
    FIG8=F2ALF

    RETURN

```

```

END

C DECK FINTSP
SUBROUTINE FINTSP (OMEGA)

COMMON /CH3D/ ISIGMA,SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI,
2 IMMIN,IMMAX,IMDEL,LMIN,LMAX
REAL SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI(4)
INTEGER ISIGMA,IMMIN,IMMAX,IMDEL,LMIN,LMAX

COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

ISIGMA = 1
NSIGMX = NSIGMA - 1
DO 10 IS=1,NSIGMX
IF (OMEGA .LT. SIGMA(IS)) GO TO 20
ISIGMA = IS
10 CONTINUE
20 CONTINUE
SIGMIN = SIGMA(ISIGMA)
SIGMAX = SIGMA(ISIGMA+1)
X1 = OMEGAE - SIGMIN
X2 = SIGMAX - OMEGAE
XX = SIGMAX - SIGMIN
WTSI(1) = X2/XX
WTSI(2) = (X2*X2/XX - XX)*X2/6.0
WTSI(3) = X1/XX
WTSI(4) = (X1*X1/XX - XX)*X1/6.0

RETURN
END

C DECK FNEDDY
SUBROUTINE FNEDDY

COMMON /APPEND/ NBKSET,NBKSTN(2),BKIMAG(2),BKFS(2),BKAS(2),
2 BKWD(2),BKSTN(10,2),BKHB(10,2),BKLNTB,BKWDTH,
2 BKWL(10,2),BKNL(10,2),NSKSET,SKIMAG(2),SKFLS(2),SKALS(2),
2 SKAUS(2),SKHB(2),SKFLWL(2),SKALWL(2),SKAUWL(2),NRDSET,RDIMAG(2),
2 RDRFS(2),RDRAS(2),RDRHB(2),RDRFWL(2),RDRAWL(2),RDTFS(2),RDTAS(2),
2 RDTHB(2),RDTFWL(2),RDTAWL(2),NSBSET,SBIMAG(2),SOBRFS(2),SOBRAS(2),
2 SOBRHB(2),SOBRFW(2),SOBRAW(2),SIBRFS(2),SIBRAS(2),SIBRH(2),
2 SIBRFW(2),SIBRAW(2),SBTFS(2),SBTAS(2),SBTHB(2),SBTFWL(2),
2 SETAWL(2),NFNSET,FNIMAG(2),FNRFS(2),FNRAS(2),
2 FNRHB(2),FNRFWL(2),FNRawl(2),FNTFS(2),FNTAS(2),FNTHB(2),
2 FNTFWL(2),FNTAWL(2),NEXPRD,ENRDO(8),ENRDS(8)

COMMON /CH3D/ ISIGMA,SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI,
2 IMMIN,IMMAX,IMDEL,LMIN,LMAX
REAL SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI(4)
INTEGER ISIGMA,IMMIN,IMMAX,IMDEL,LMIN,LMAX

COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMTR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSC,
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMTR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

COMMON /RLDBK/ PSUR(25),BMK(25),DK(25),CAK(25),HQ,HSPAN,HMNCHD,
2 HAREA,HXCP,HYCP,HZCP,HGAMMA,HYHAT,HEAR,HLCS,RQ(2),RSPAN(2),
2 RMNCHD(2),RAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),

```

```

2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
2 SYCP(2),SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2),BQ(2),
2 BSPAN(2),BMNCHD(2),BAREA(2),EXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),
2 PQ(2,2),PSPAN(2,2),PMNCHD(2,2),PAREA(2,2),PXCP(2,2),PYCP(2,2),
2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),PLCS(2,2),
2 SHPDMP(10,8),ENCON,WPHI,WMELM(4,9),SFELM(4,9,8),
2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
2 ENWM,ENSF(8,8),ENRE(8),ENPE(8),ENFE(8),ENHE(8),ENBE(8),
2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(8),ENSL(8),ENBL(8),
2 ENSHP(8,8),RELM(4,9),ITS(25),RD(25),EDDY(8,25),RGB(25)
      REAL RDBLK(2692)
      EQUIVALENCE (PSUR(1),RDBLK(1))

      DO 20 IA=1,NRANG
      ENFE(IA) = 0
      DO 10 IS=1,NSIGMA
      SHPDMP(IS,IA) = 0
10    CONTINUE
20    CONTINUE
      IF (NFNSET .EQ. 0) GO TO 100
      DO 50 K=1,NFNSET
      YHAT = SQRT(FYCP(K)*FYCP(K) + FZCP(K)*FZCP(K))
      GAMMAE = FGAMMA(K) + 1.
      ALF = ATAN( ABS( ((FYCP(K)/FZCP(K)) + TAN(GAMMAE*DEGRAD))/(1. -
2      (FYCP(K)/FZCP(K))*TAN(GAMMAE*DEGRAD)) ) )
      C = 0.0065 + (FLCS(K)+FLCS(K))/(0.9*PI*FEAR(K))
      CON = FQ(K)*4./(3.*PI)*RHO*YHAT**3*FAREA(K)*C*SIN(ALF)
      DO 40 IA=1,NRANG
      DO 30 IS=1,NSIGMA
      SHPDMP(IS,IA) = SHPDMP(IS,IA) + (CON*SIGMA(IS)*RANG(IA)) *
2      SIGMA(IS)
30    CONTINUE
40    CONTINUE
50    CONTINUE
      DO 60 IA=1,NRANG
      CALL SPFIT (SIGMA,SHPDMP(1,IA),FEELM(1,1,IA),NSIGMA)
      ENFE(IA) = ENCON*REVAL(FEELM(1,ISIGMA,IA),WTSI)
60    CONTINUE
100   CONTINUE

      RETURN
      END

```

```

C DECK FNLIIFT
SUBROUTINE FNLIIFT

COMMON /APPEND/ NBKSET,NBKSTN(2),BKIMAG(2),BKFS(2),BKAS(2),
2 BKWD(2),BKSTN(10,2),BKHB(10,2),BKLNTB,BKWDTH,
2 BKWL(10,2),BKAN(10,2),NSKSET,SKIMAG(2),SKFLS(2),SKALS(2),
2 SKAHL(2),SKFLWL(2),SKALWL(2),SKAUWL(2),NRDSET,RDIMAG(2),
2 RDRFS(2),RDRAS(2),RDRHB(2),RDRFWL(2),RDRAWL(2),RDTFS(2),RDTAS(2),
2 RDTHB(2),RDTFWL(2),RDTAWL(2),NSBSET,SBIMAG(2),SOBRFS(2),SOBRAS(2),
2 SOBRHB(2),SOBRFW(2),SOBRAW(2),SIBRFS(2),SIBRAS(2),SIBRHB(2),
2 SIBRFW(2),SIBRAW(2),SBTFS(2),SBTAS(2),SBTHB(2),SBTFWL(2),
2 SBTAWL(2),NFNSET,FNIMAG(2),FNRFS(2),FNRAS(2),
2 FNRHB(2),FNRFWL(2),FNRawl(2),FNTFS(2),FNTAS(2),FNTHB(2),
2 FNTFWL(2),FNTAWL(2),NEXPRD,ENRDO(8),ENRDS(8)

COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
1 INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8),
1 REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FEDZV,DELWL,TLCB
2 INTEGER NSTATN,NOFSET(25),NFREBD,NPTS

```

```

CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZFT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPICTH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /PHYSCO/ II,TPI,PI,P1OT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,P1OT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

COMMON /RLDBK/ PSUR(25),BMK(25),DK(25),CAK(25),HQ,HSPAN,HMNCHD,
2 HAREA,HXCP,HYCP,HZCP,HGAMMA,HYHAT,HEAR,HLCS,RQ(2),RSPAN(2),
2 RMNCHD(2),RAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),
2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
2 SYCP(2),SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2),BQ(2),
2 BSPAN(2),BMNCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),
2 PQ(2,2),PSPAN(2,2),PMNCHD(2,2),PAREA(2,2),PXCP(2,2),PYCP(2,2),
2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),PLCS(2,2),
2 STADMP(10),SHPDMP(10,8),ENCON,WPHI,TPHI,WMELM(4,9),SFELM(4,9,8),
2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
2 ENWM,ENSF(8,8),ENRE(8),ENPE(8),ENFE(8),ENHE(8),ENBE(8),
2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(8),ENSL(8),ENBL(8),
2 ENSHP(8,8),RELM(4,9),ITS(25),RD(25),EDDY(8,25),RGB(25)
REAL RDBLK(2692)
EQUIVALENCE (PSUR(1),RDBLK(1))

REAL LCS,MCHORD

IF (NFNSET .EQ. 0) GO TO 20
EN = 0
STASPC = LPP/20
DO 10 K=1,NFNSET
XRTF = LCB - FNRF(S(K))*STASPC
XRTA = LCB - FNRA(S(K))*STASPC
XTPF = LCB - FNTF(S(K))*STASPC
XTPA = LCB - FNTA(S(K))*STASPC
YRT = FNRRH(S(K))
YTP = FNTHB(S(K))
ZRT = {FNRFWL(S(K)) + FNRAWL(S(K))}/2 - (DBLWL+VCG)
ZTP = {FNTFWL(S(K)) + FNTAWL(S(K))}/2 - (DBLWL+VCG)
SPAN = SQRT((ZRT-ZTP)**2 + (YTP-YRT)**2)
Q = FNIMAG(S(K))
MCHORD = 0.5*((XRTF-XRTA) + (XTPF-XTPA))
CR = XRTF - XRTA
CT = XTPF - XTPA
XRQC = XRTF - 0.25*CR
XTQC = XTPF - 0.25*CT
DX = XRQC - XTQC
H = SQRT(DX*DX + SPAN*SPAN)
COSLAM = SPAN/H
SECLAM2 = 1.0/(COSLAM*COSLAM)

* LAM = ACOS(SPAN/H)
*      = quarter chord sweep angle in radians

* area
AREA = SPAN*MCHORD

* center of pressure
ZP = 0.5*(ZRT + ZTP)
YP = 0.5*(YRT + YTP)
XO = 0.5*(XRTF + XTPF)
XCP = XO - 0.25*MCHORD
YCP = YP

```

```

ZCP = ZF

* moment arm
    ARG = (ZRT-ZTP) / SPAN
    GAMMA = - 90
    IF (ARG .LT. 1) GAMMA = - ASIN(ARG)*RADDEG
    GAM = GAMMA*DEGRAD
    YHAT = YCP*COS(GAM) + ZCP*SIN(GAM)

* effective aspect ratio
    EAR = 2*SPAN/MCHORD

* lift curve slope
    LCS = 1.8*PI*EAR/(COSLAM*SQRT((EAR*SECLAM2)**2 + 4) + 1.8)
    FQ(K) = Q
    FSPAN(K) = SPAN
    FMNCHD(K) = MCHORD
    FAREA(K) = AREA
    FXCP(K) = XCP
    FYCP(K) = YCP
    FZCP(K) = ZCP
    FGAMMA(K) = GAMMA
    FYHAT(K) = YHAT
    FEAR(K) = EAR
    FLCS(K) = LCS
    EN = EN + Q*(RHO/2)*AREA*LCS*YHAT*YHAT*WPHI*ENCON
10   CONTINUE
20   CONTINUE
    DO 30 IV=1,NVK
    ENFL(IV) = 0
    IF (NFNSET .GT. 0) ENFL(IV) = EN*VFS(IV)
30   CONTINUE

    RETURN
    END

C DECK FNRAO
SUBROUTINE FNRAO (IV,NL,NU,MOTL,RAO,PHS,NMOT,NOMEGA,OMEGAE,IPHS)
COMMON /FINCON/ IACTFN,IFCLCS,FGAIN(8),FK(3),FA(3),FB(3),
2 FCLCS(8,2)
COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

COMPLEX FGC,MOTL(NMOT,NOMEGA),BETA,ROLL
DIMENSION OMEGAE(NOMEGA),RAO(NOMEGA),PHS(NOMEGA)

DO 10 I=NL,NU
ROLL = MOTL(2,I)*RADDEG
OMGE = OMEGAE(I)
OMGE2 = OMGE*OMGE
FGC = ((FK(1)-OMGE2*FK(3))+II*OMGE*FK(2))/(((FA(1)-OMGE2*FA(3))+
2 II*OMGE*FA(2))*((FB(1)-OMGE2*FB(3))+II*OMGE*FB(2)))
BETA = FGAIN(IV)*FGC*ROLL
CALL RAOpha (BETA,RAO(I),PHS(I),RADDEG,IPHS)
10 CONTINUE

    RETURN
    END

C DECK FTWO
FUNCTION FTWO (K,TLOCAL,RD)
COMMON /GEOM/ X,NSTATN,Y,Z,WOFSET,LPP,BEAM,DRAFT,LCF,

```

```

1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

```

```

NNODES=NOFSET(K)
FTWO=1.0
IF ((Y(NNODES-1,K)-Y(NNODES,K)) .GE. 0.) RETURN
BR=(Y(NNODES,K)-Y(NNODES-1,K))/(-Z(NNODES-1,K))
ALF=ATAN(BR)
RDD = RD/ABS(TLOCAL)
FTWO=FIG8(RDD,ALF)

RETURN
END

```

```

C DECK GENOFS
SUBROUTINE GENOFS (BEAM,DRAFT,SECARE,NOFSET,HLFBTH,WTRLNE,
2 PI,DBLWL)

```

```

* this routine generates a set of offsets for evenly spaced angles
* from the beam, draft and sectional area coefficients of a station
* using the LEWIS form mapping.
* W.G.MEYERS, DTSNRDC, 080977

* LEWIS FORM representation:
*  $Z = A1 * ZETA + A2 * ZETA^{**-1} + A3 * ZETA^{**-3}$ 
* where ZETA is a complex mapping variable and A1, A2 and A3
* are coefficients.

* phi is an angle measured from the waterline down and is negative.

```

```

DIMENSION HLFBTH(NOFSET),WTRLNE(NOFSET)

```

```

HBEAM = BEAM/2
AREA = SECARE*BEAM*DRAFT
A3 = -.25*(HBEAM+DRAFT) + .25*SQRT((HBEAM+DRAFT)**2 +
2 8*(HBEAM*DRAFT-2*AREA/PI))
A2 = .50*(HBEAM-DRAFT)
A1 = .50*(HBEAM+DRAFT) - A3
DELPHI = (PI/2)/(NOFSET-1)
KOFSET = NOFSET + 1
DO 10 IOFSET=1,NOFSET
PHI = -(IOFSET-1)*DELPHI
KOFSET = KOFSET - 1
HLFBTH(KOFSET) = (A1+A2)*COS(PHI) + A3*COS(3*PHI)
WTRLNE(KOFSET) = (A1-A2)*SIN(PHI) - A3*SIN(3*PHI)
WTRLNE(KOFSET) = WTRLNE(KOFSET) + DBLWL
10 CONTINUE

```

```

RETURN
END

```

```

C DECK GRNFRQ

```

```

SUBROUTINE GRNFRQ (YS, ZS, NPT, SIGMA2, POTLOG, PTNLOG, CN, SN,
2 CTV, CTL, GREENV, GREENL)

```

```

* this subroutine provides the necessary input for the subroutine
* EXPINT, and provides the entire expression of the pulsating source
* which are stored in GREENV(I,J) for the symmetric flow(surge and
* heave) and in GREENL(I,J) for the anti-symmetric flow(sway and
* roll) where
* I = location of source

```

```

*      J = location of the field point on the cross section
*      boundary
*      the normal derivatives of the foregoing green functions are stored
*      in CTV(I,J) and CTL(I,J) respectively.

2 COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
  RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMTR,PUNITS,REYSCL
  COMPLEX II
  CHARACTER*4 PUNITS(2)
  REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
  RHOF,GNUS,GNUF,FTMTR

COMMON /TWOD/ YY, ZZ, ENN, ISTA
INTEGER ISTA
REAL YY(10,25),ZZ(10,25),ENN(4,10,25)

COMPLEX CTV(10,10), CTL(10,10), GREENV(10,10), GREENL(10,10)
DIMENSION YS(11), ZS(11)
DIMENSION POTLOG(2,10,10), PTNLOG(2,10,10), CN(10), SN(10)

DO 1 I=1,NPT
YR1 = SIGMA2*(YY(I,ISTA)-YS(1))
ZR1 = -SIGMA2*(ZZ(I,ISTA)+ZS(1))
YL1 = SIGMA2*(YY(I,ISTA)+YS(1))
ZL1 = ZR1
CALL EXPINT(YR1,ZR1,EJ1,CXR1,SXR1,RAR1,RBR1,CR1,SR1)
CALL EXPINT(YL1,ZL1,EJ1,CXL1,SXL1,RAL1,RBL1,CL1,SL1)
DO 1 J=1,NPT
YR2 = SIGMA2*(YY(I,ISTA)-YS(J+1))
ZR2 = -SIGMA2*(ZZ(I,ISTA)+ZS(J+1))
YL2 = SIGMA2*(YY(I,ISTA)+YS(J+1))
ZL2 = ZR2
CALL EXPINT(YR2,ZR2,EJ2,CXR2,SXR2,RAR2,RBR2,CR2,SR2)
CALL EXPINT(YL2,ZL2,EJ2,CXL2,SXL2,RAL2,RBL2,CL2,SL2)
SIPJ = SN(I)*CN(J)+SN(J)*CN(I)
CIPJ = CN(I)*CN(J)-SN(I)*SN(J)
SIMJ = SN(I)*CN(J)-SN(J)*CN(I)
CIMJ = CN(I)*CN(J)+SN(I)*SN(J)
DPR = 2.* (SIPJ*(CR1-CR2)-CIPJ*(SR1-SR2))
DPL = 2.* (CIMJ*(SL1-SL2)-SIMJ*(CL1-CL2))
PPR = 2./SIGMA2*(SN(J)*(RAR1-RAR2)+CN(J)*(RBR1-RBR2))
PPL = 2./SIGMA2*(SN(J)*(RAL1-RAL2)+CN(J)*(RBL2-RBL1))
DWR = TPI*(EJ2*(SXR2*CIPJ-CXR2*SIPJ)-EJ1*(SXR1*CIPJ-CXR1*SIPJ))
DWL = TPI*(EJ1*(SXL1*CIMJ-CXL1*SIMJ)-EJ2*(SXL2*CIMJ-CXL2*SIMJ))
PWR = TPI/SIGMA2*(EJ1*(SXR1*CN(J)-CXR1*SN(J))-EJ2*(SXR2*CN(J)-
2 CXR2*SN(J)))
PWL = TPI/SIGMA2*(EJ2*(SXL2*CN(J)+CXL2*SN(J))-EJ1*(SXL1*CN(J)+CXL1*SN(J)))
CTV(I,J) = PTNLOG(1,I,J)+DPR+DPL-II*(DWR+DWL)
CTL(I,J) = PTNLOG(2,I,J)+DPR-DPL-II*(DWR-DWL)
GREENV(I,J) = POTLOG(1,I,J)+PPR+PPL-II*(PWR+PWL)
GREENL(I,J) = POTLOG(2,I,J)+PPR-PPL-II*(PWR-PWL)
IF (J-NPT) 2,1,1
2 YR1 = YR2
ZR1 = ZR2
CXR1 = CXR2
SXR1 = SXR2
RAR1 = RAR2
RBR1 = RBR2
CR1 = CR2
SR1 = SR2
YL1 = YL2
ZL1 = ZL2
EJ1 = EJ2
CXL1 = CXL2
SXL1 = SXL2
RAL1 = RAL2
RBL1 = RBL2
CL1 = CL2
SL1 = SL2
1 CONTINUE

```

```

      RETURN
      END

C DECK GRNLOG
      SUBROUTINE GRNLOG (YS, ZS, NPT, POTLOG, PTNLOG, CN, SN)

*   this subroutine computes the logarithm part of the pulsating
*   source and its normal derivative, and are stored in, respectively,
*   in POTLOG(M,I,J) and POTNLOG(M,I,J) where
*       m=1 symmetric flow about the z-axis(surge and heave)
*           2 anti-symmetric flow(sway and roll)
*       i=location of source
*       j=field-point location on cross-section boundary

      COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
      COMPLEX II
      CHARACTER*4 PUNITS(2)
      REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

      COMMON /TWOD/ YY, ZZ, ENN, ISTA
      INTEGER ISTA
      REAL YY(10,25),ZZ(10,25),ENN(4,10,25)

      DIMENSION YS(11), ZS(11)
      DIMENSION POTLOG(2,10,10), PTNLOG(2,10,10), CN(10), SN(10)

      DO 15 I=1,NPT
      SN(I) = -ENN(2,I,ISTA)
      CN(I) = ENN(3,I,ISTA)
15     DO 10 I=1,NPT
      YM1 = YY(I,ISTA) - YS(1)
      ZM1 = ZZ(I,ISTA) - ZS(1)
      YP1 = YY(I,ISTA) + YS(1)
      ZP1 = ZZ(I,ISTA) + ZS(1)
      FPR1 = 0.5*ALAG(YM1**2+ZM1**2)
      FPL1 = 0.5*ALAG(YP1**2+ZP1**2)
      FCR1 = 0.5*ALAG(YM1**2+ZP1**2)
      FCL1 = 0.5*ALAG(YP1**2+ZP1**2)
      APR1 = ATAN3(ZM1,YM1)
      APL1 = ATAN3(ZM1,YP1)
      ACR1 = ATAN3(ZP1,YM1)
      ACL1 = ATAN3(ZP1,YP1)
      DO 10 J=1,NPT
      YM2 = YY(I,ISTA) - YS(J+1)
      ZM2 = ZZ(I,ISTA) - ZS(J+1)
      YP2 = YY(I,ISTA) + YS(J+1)
      ZP2 = ZZ(I,ISTA) + ZS(J+1)
      APR2 = ATAN3(ZM2,YM2)
      FPR2 = 0.5*ALAG(YM2**2+ZM2**2)
      FCR2 = 0.5*ALAG(YM2**2+ZP2**2)
      FPL2 = 0.5*ALAG(YP2**2+ZM2**2)
      FCL2 = 0.5*ALAG(YP2**2+ZP2**2)
      J1 = J + 1
      IF (YM2 .GE. 0.) GO TO 4
      IF (J1 .GT. I) GO TO 6

*   below takes care of a concave top or a flat top
      IF (ZM2 .LT. 0.) APR2 = APR2+TPI
      GO TO 5

*   below takes care of a convex bottom or a flat bottom
      6 IF (ZM2 .GE. 0.) APR2 = APR2-TPI
      5 IF (ZP2 .LT. 0.) GO TO 4
          ACR2 = - 0.5*TPI
          GO TO 3
      4 ACR2 = ATAN3(ZP2,YM2)
      3 ACL2 = ATAN3(ZP2,YP2)
          APL2 = ATAN3(ZM2,YP2)

```

```

SIMJ = SN(I)*CN(J)-SN(J)*CN(I)
CIMJ = CN(I)*CN(J)+SN(I)*SN(J)
SIPJ = SN(I)*CN(J)+SN(J)*CN(I)
CIPJ = CN(I)*CN(J)-SN(I)*SN(J)
DPNR = SIMJ*(FPR1-FPR2)+CIMJ*(APR1-APR2)
DPNL = SIPJ*(FPL2-FPL1)+CIPJ*(APL2-APL1)
DCNR = SIPJ*(FCR1-FCR2)+CIPJ*(ACR1-ACR2)
DCNL = SIMJ*(FCL2-FCL1)+CIMJ*(ACL2-ACL1)
PTNLOG(1,I,J) = DPNR+DPNL-DCNR-DCNL
PTNLOG(2,I,J) = DPNR-DPNL-DCNR+DCNL
PPR = CN(J)*(YM1*FPR1-ZM1*APR1-YM1-YM2*FPR2+ZM2*APR2+YM2) +
2 SN(J)*(ZM1*FPR1+YM1*APR1-ZM1-ZM2*FPR2-YM2*APR2+ZM2)
PPL = CN(J)*(YP2*FPL2-ZM2*APL2-YP2-YP1+FPL1+ZM1*APL1+YP1) +
2 SN(J)*(ZM1*FPL1+YP1*APL1+ZM2-ZM2*FPL2-YP2*APL2-ZM1)
PCR = CN(J)*(YM1*FCR1-ZP1*ACR1-YM1-YM2*FCR2+ZP2*ACR2+YM2) +
2 SN(J)*(ZP2*FCR2+YM2*ACR2+ZP1*FCR1-YM1*ACR1-ZP2)
PCL = CN(J)*(YP2*FCL2-ZP2*ACL2-YP2-YP1*FCL1+ZP1*ACL1+YP1) +
2 SN(J)*(ZP2*FCL2+YP2*ACL2-ZP2-ZP1*FCL1-YP1*ACL1+ZP1)
POTLOG(1,I,J) = PPR+PPL-PCR-PCL
PCTLOG(2,I,J) = PPR-PPL-PCR+PCL
IF (J-NPT) 475,10,10
475 YM1 = YM2
ZM1 = ZM2
YP1 = YP2
ZP1 = ZP2
FPR1 = FPR2
FPL1 = FPL2
FCR1 = FCR2
FCL1 = FCL2
APR1 = APR2
APL1 = APL2
ACR1 = ACR2
ACL1 = ACL2
10 CONTINUE
      RETURN
      END

C DECK HLEDDY
SUBROUTINE HLEDDY

CALL SECT1
CALL TANAKA
CALL VISC

RETURN
END

C DECK HLLIFT
SUBROUTINE HLLIFT

COMMON /DATIMP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
2 LRAOPR,ADRPR,ORGOPTN,GMNOM,KG,STATN(25),NSOFST(25),
2 NLEWF(25),HLFBTH(10,25),WTRLNE(10,25),BLEWF(25),TLEWF(25),
2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
2 ZPTFBD(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
2 STATNM,STATIS
CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
2 FBNUMB,PTNUMB,ORGOPTN
REAL KG

COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)
COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPI,BEAM,DRAFT,LCF,
1 VCG,GM,DELM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,

```

```

2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

COMMON /RLDBK/ PSUR(25),BMK(25),DK(25),CAK(25),HQ,HSPAN,HMNCHD,
2 HAREA,HXCP,HZCP,HGAMMA,HYHAT,HEAR,HLCS,RQ(2),RSPAN(2),
2 RMNCHD(2),RAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),
2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
2 SYCP(2),SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2),BQ(2),
2 BSPAN(2),BMMCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),
2 PQ(2,2),PSPAN(2,2),PMNCHD(2,2),PAREA(2,2),PXCP(2,2),PYCP(2,2),
2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),PLCS(2,2),
2 STADM(10),SHPDMP(10,8),ENCON,WPHI,TPHI,WMELM(4,9),SFELM(4,9,8),
2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
2 ENWM,ENSF(8,8),ENRE(8),ENPE(8),ENFE(8),ENHE(8),ENBE(8),
2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(8),ENSL(8),ENBL(8),
2 ENSHP(8,8),RELM(4,9),ITS(25),RD(25),EDDY(8,25),RGB(25)
REAL RDBLK(2692)
EQUIVALENCE (PSUR(1),RDBLK(1))

REAL LCS,MCHORD

Q = 1
GAMMA = - 90
ORG = VCG + DRAFT
SPAN = DRAFT
MCHORD = LPP

* area
    AREA = SPAN*MCHORD

* center of pressure
    SS = 0
    SP = 0
    DO 5 L=1,NSTATN
    IF (L .EQ. 1) DX = (X(2) - X(1))/2
    IF (L .EQ. NSTATN) DX = (X(NSTATN) - X(NSTATN-1))/2
    IF (L.GT.1 .AND. L.LT.NSTATN) DX = (X(L+1) - X(L-1))/2
    DX = ABS(DX)
    NPT = NOFSET(L)
    IF (NPT .LT. 2) GO TO 5
    T = ABS(Z(1,L))
    A = T*DX
    SP = SP + A
    SS = SS + X(L)*A
5  CONTINUE
    XCP = SS/SP
    YCP = 0.0
    ZCP = 0.0

* moment arm
    GAM = GAMMA*DEGRAD
    YHAT = YCP*COS(GAM) + ZCP*SIN(GAM)

* effective aspect ratio

```

```

EAR = 2*SPAN/MCHORD

* lift curve slope

LCS = (PI/2)*EAR
HQ = Q
HSPAN = SPAN
HMNCHD = MCHORD
HAREA = AREA
HXCP = XCP
HYCP = YCP
HZCP = ZCP
HGAMMA = GAMMA
HYHAT = YHAT
HEAR = EAR
HLCS = LCS
EN = Q*(RHO/2)*AREA*LCS*YHAT*YHAT*WPHI*ENCON
DO 10 IV=1,NVK
ENHL(IV) = EN*VFS(IV)
10 CONTINUE

RETURN
END

C DECK HSTAT
SUBROUTINE HSTAT

* HYDROSTATIC CALCULATIONS

* INPUTS
* X(K)      = location of station k (distance fwd of ap) in meters
* NSTATN    = number of stations in x-array
* Y(J,K)    = y-coordinate of offset j at station k
*             (half-breadth) in meters
* Z(J,K)    = z-coordinate of offset j at station k
*             (distance from waterline, negative down) in metres
* NOFSET(K)= number of offsets given for station k

* NOTE
* first station must be at stern, last station at bow
* first offset must be at keel, last offset at waterline

COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
2 LRAOPR,ADRPR,ORGOPTN,GMMOM,KG,STATN(25),NSOFST(25),
2 NLEWF(25),HLFBTH(10,25),WTRLNE(10,25),BLEWF(25),TLEWF(25),
2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
2 ZPTFBD(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
2 STATNM,STATIS
CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
2 FBNUMB,PTNUMB,ORGOPTN
REAL KG

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,

```

```

2 SPTFIL,LACFIL,LAEFIL
1 INTEGER     SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL

COMMON /LOADS/ NLOADS,SWGHT(25),SMASS(25),XLDSTN(10),XLDXPT(25),
2 LSTATN(25)

COMMON /PHYSICO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

COMMON /SMPSYS/ FIS,AS,SIS,SOS,SDS,HALOS,DEV,PRN,SMPBS,SMPIS,
2 SMPDS,SHPTYPS,SHIPS,VARS,CYCLS,TITLES,OPTION,LSIS,LSOS,
2 LSDS,LHALOS,LDEV,LPRN,LSMPPS,LSMPIS,LSMPOS,LSMPDS,LSHPTYPS,
2 LSHIPS,LTITLES
CHARACTER*160 AS
CHARACTER*80 FIS,SIS,SOS,SDS,TITLES
CHARACTER*20 HALOS,DEV,PRN,SMPBS,SMPIS,SMPDS,SHPTYPS
CHARACTER SHIPS*6,VARS*2,CYCLS*2
INTEGER*2 OPTION

COMMON /TWOD/ YY, ZZ, ENN, ISTA
INTEGER ISTA
REAL YY(10,25),ZZ(10,25),ENN(4,10,25)

COMMON /WGHTS/ WTDL,NORM
REAL WTDL(10,25),NORM(4,10,25)

REAL MX,IXX,IYY
DIMENSION P(2,10),NDI(2),ENDI(2,2),CC(14),PSPL(2,70),SEGS(8,69),
2 ELEMS(4,24),
XMT(25), ZMT(25), BBB(25), XB(25), XXB(25),
PSEGS(8,9,25)
CHARACTER*4 METER

DATA METER /'Mete'/
DATA NDI, ENDI / 2 * 1, 4 * 0.0 /
DATA ZERO, ONE, TWO / 0.0, 1.0, 2.0 /

AREAMX = ZERO
DO 50 K=1,NSTATN
NP = NOFSET(K)
IF (NP .GT. 1) GO TO 10
ASTAT(K) = ZERO
ZMT(K) = ZERO
GO TO 40
CONTINUE
10  NS = NP - 1
DO 20 J=1,MP
P(1,J) = Y(J,K)
P(2,J) = Z(J,K)
CONTINUE
20  CALL SPLNT2 (PSEGS(1,1,K), P, NP, NDI, ENDI)
CALL SPLNT2 (PSEGS(1,1,K), NS, AREA, 1, ZERO, NS, ONE, 0)
ASTAT(K) = TWO * AREA
IF (ASTAT(K) .GT. AREAMX) AREAMX = ASTAT(K)
L = 0
DO 30 J=1,NS
CALL CUBCO2 (PSEGS(1,J,K),CC)
NT = 7
DT = 1./(NT-1)
DO 25 I=1,NT
L = L + 1
T = (I-1)*DT
T2 = T*T
T3 = T*T2
YSPL = CC(1)*T3 + CC(3)*T2 + CC(5)*T + CC(7)
ZSPL = CC(2)*T3 + CC(4)*T2 + CC(6)*T + CC(8)

```

```

      PSPL(1,L) = YSPL*ZSPL
      PSPL(2,L) = ZSPL
25    CONTINUE
30    CONTINUE
      CALL SPLNT2 (SEGS,PSPL,L,NDI,ENDI)
      CALL SPINT2 (SEGS,L-1,AREA,1,ZERO,L-1,ONE,0)
      ZMT(K) = TWO * AREA
40    CONTINUE
      BSTAT(K) = TWO * Y(NP,K)
      BBB(K) = Y(NP,K)**3
      XMT(K) = ASTAT(K) * X(K)
50    CONTINUE

*      PSEGS(1,J,K) = parametric spline segment representing station k
*                                ( I = 1, 8 )          ( J = 1, NDFSET(K)-1 )
*      BSTAT(K) = full beam of station k at waterplane in m
*      ASTAT(K) = area of station k in m**2
*      ZMT(K) = moment of sta.k area about wp in m**3
*      XMT(K) = moment of sta.k area about fp in m**3

      NS = NSTATN
      CALL SPFIT (X, ASTAT, ELEMS, NS)
      CALL SPINTG (X(1), X(NS), X, NS, ELEMS, ZERO, NEBLA, DUMMY)
      CALL SPFIT (X, XMT, ELEMS, NS)
      CALL SPINTG (X(1), X(NS), X, NS, ELEMS, ZERO, XXMT, DUMMY)
      CALL SPFIT (X, ZMT, ELEMS, NS)
      CALL SPINTG (X(1), X(NS), X, NS, ELEMS, ZERO, ZZMT, DUMMY)

*      NEBLA = displaced volume in m**3
*      XXMT = moment of displ. vol. about ap in m**4
*      ZZMT = moment of displ. vol. about wp in m**4

      LCB = LPP - XXMT/NEBLA
      VCB = ZZMT / NEBLA

*      LCB = longitudinal center of buoyancy in m
*            (distance from fp, positive aft)
*      VCB = vertical center of buoyancy in m
*            (distance from wp, negative down)
*      find local draft at lcb (necessary for trimmed ship)

      STASPC = LPP/20
      SLCB = LCB/STASPC
      DO 240 I=1,NSTATN
      IF (STATN(I) .LT. SLCB) GO TO 240
      SDIS = SLCB - STATN(I-1)
      SLOPE = (WTRLNE(1,I) - WTRLNE(1,I-1)) / (STATN(I) - STATN(I-1))
      TLCB = DBLWL - (WTRLNE(1,I-1) + SDIS*SLOPE)
      GO TO 250
240  CONTINUE
250  IF (NPTLOC .EQ. 0) GO TO 270
      DO 260 I=1,NPTS
      XPT(I) = XPT(I) - (LPP-LCB)
260  CONTINUE
270  IF (NFREBD .EQ. 0) GO TO 290
      DO 280 I=1,NFREBD
      FBDX(I) = FBDX(I) - (LPP-LCB)
280  CONTINUE
      CALL TRIM
290  CONTINUE

*      transform origin of x-axis to LCB

      DO 150 K=1,NSTATN
      X(K) = X(K) - (LPP-LCB)
      XB(K) = X(K) * BSTAT(K)
      XXB(K) = X(K) * XB(K)
150  CONTINUE

*      X(K) = distance of station k from LCB (negative aft) in meters

```

```

CALL SPFIT (X, BSTAT, ELEMS, NS)
CALL SPINTG (X(1), X(NS), X, NS, ELEMS, ZERO, AWP, DUMMY)
CALL SPFIT (X, BBB, ELEMS, NS)
CALL SPINTG (X(1), X(NS), X, NS, ELEMS, ZERO, IYY, DUMMY)
IYY = TWO * IYY / 3.
BM = IYY / NEBLA

* KG is the distance from the keel to the center of gravity
* at the LCE

KG = KG + DELGM
VCG = KG - TLCB
BG = VCG - VCB
GM = BM - BG

* AWP = area of waterplane in m**2
* IYY = transverse moment of inertia of wp in m**4
* BM = center of buoyancy to transverse metacenter in meters
* GM = transverse metacentric height in meters

CALL SPFIT (X, XB, ELEMS, NS)
CALL SPINTG (X(1), X(NS), X, NS, ELEMS, ZERO, MX, DUMMY)
CALL SPFIT (X, XBX, ELEMS, NS)
CALL SPINTG (X(1), X(NS), X, NS, ELEMS, ZERO, IXX, DUMMY)
BML = IXX / NEBLA
GML = BML - BG
LCF = LCB - MX/AWP

* MX = longitudinal moment of wp about lcb in m**3
* IXX = longitudinal moment of inertia of wp in m**4
* BML = center of buoyancy to longitudinal metacenter in m
* GML = longitudinal metacentric height in m
* LCF = longitudinal center of flotation in m
* (distance from fp, positive aft)
* mass, displacement and moment of inertia definitions
* roll moment of inertia is about the LCG in the waterplane
* roll radius of gyration is about the VCG

MASS = RHO*NEBLA
DISPLM = MASS*GRAV
IPITCH = MASS*(KPITCH*LPP)**2
IROLL = MASS*((KROLL*BEAM)**2 + VCG**2)
IYAW = MASS*(KYAW*LPP)**2
IYAWRL = MASS*(KYAWRL*LPP**2)

* restoring definitions

CHEAVE = RHO*GRAV*AWP
CPITCH = DISPLM*GML
CHEAPI = - RHO*GRAV*AWP*(LCB-LCF)

* note that LCB and LCF are measured from the fp, pos aft

CROLL = DISPLM*GM
CALL NORMAL (PSEGS)
CALL NORMT5 (PSEGS)
DO 60 K=1,NSTATN
NP = NOFSET(K)
IF (NP .GT. 1) CALL CONIWT (WTDL(1,K),PSEGS(1,1,K),NP)
60 CONTINUE
IF (NLOADS .EQ. 0) GO TO 69

* obtain locations for load calculations

DO 65 IP=1,NLOADS
XLS = XLDSTN(IP)
N1 = NSTATN - 1
DO 63 K=1,N1
IF (.NOT. (XLS.GE.STATN(K) .AND. XLS.LT.STATN(K+1))) GO TO 63
XLDSTN(IP) = 0.5*(STATN(K) + STATN(K+1))
GO TO 64
63 CONTINUE

```

```

64 XLDXPT(IP) = LCB - XLDSTN(IP)*LPP/20
65 LSTATN(IP) = NSTATN + 1 - K
65 CONTINUE

* compute section mass

L = NSTATN + 1
DO 68 K=1,NSTATN
L = L - 1
IF (PUNITS(1) .EQ. METER) SMASS(L) = SWGHT(K)*1000
IF (PUNITS(1) .NE. METER) SMASS(L) = SWGHT(K)*2240/GRAV
68 CONTINUE
69 CONTINUE

* calculation of wetted surface

NS = NSTATN
DO 80 K=1,NSTATN
NP = NOFSET(K)
GIRTH(K) = ZERO
IF (NP .LT. 2) GO TO 80
DO 70 J=1, NP
GIRTH(K) = GIRTH(K) + WTDL(J,K)
70 CONTINUE
GIRTH(K) = TWO * GIRTH(K)
80 CONTINUE
CALL SPFIT (X, GIRTH, ELEMS, NS)
CALL SPINTG (X(1), X(NS), X, NS, ELEMS, ZERO, WSURF, DUMMY)

* write offsets to HPLFIL for graphics

CALL SPLNFT

* write scratch file

FIS = SDS(1:LSDS)//'.SCR'
OPEN (UNIT=SCRFIL,FILE=FIS,FORM='UNFORMATTED',STATUS='UNKNOWN')

WRITE (SCRFIL) YY,ZZ,ENN,ISTA
WRITE (SCRFIL) WTDL,NORM

CLOSE (UNIT=SCRFIL)

RETURN
END

C DECK HSTOUT
SUBROUTINE HSTOUT

COMMON /APPEND/ NBKSET,NBKSTN(2),BKIMAG(2),BKFS(2),BKAS(2),
2 BKWD(2),BKSTN(10,2),BKHB(10,2),BKLNTH,BKWDTH,
2 BKWL(10,2),BKA(10,2),NSKSET,SKIMAG(2),SKFLS(2),SKALS(2),
2 SKAUS(2),SKHB(2),SKFLWL(2),SKALWL(2),SKAUWL(2),NRDSET,RDIMAG(2),
2 RDRFS(2),RDRAS(2),RDRHB(2),RDRFWL(2),RDRAWL(2),RDTFS(2),RDTAS(2),
2 RDTHB(2),RDTFWL(2),RDTAWL(2),NSBSET,SBIMAG(2),SOBRFS(2),SOBRAS(2),
2 SOBRHB(2),SORRFW(2),SOBRRAW(2),SIBRFS(2),SIBRAS(2),SIBRHB(2),
2 SIBRFW(2),SIBRAW(2),SBTAS(2),SBTHB(2),SBTFWL(2),
2 SBTAWL(2),WFNSET,WNIMAG(2),FNRFS(2),FNRAS(2),
2 FNRHB(2),FNRFWL(2),FNRAWL(2),FNTFS(2),FNTAS(2),FNTHB(2),
2 FNTFWL(2),FNTAWL(2),NEXPRD,ENRDO(8),ENRDS(8)

COMMON /DATINP/ OPTN,MGTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
2 LRAOPR,ADRPR,ORGOPTN,GMNOM,KG,STATN(25),NSOFST(25),
2 NLEWF(25),HLFBTH(10,25),WTRLNE(10,25),BLEWF(25),TLEWF(25),
2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
2 ZPTFBD(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
2 STATNM,STATIS
CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
INTEGER OPTN,MGTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
2 FBNUMB,PTNUMB,ORGOPTN
REAL KG

```

```

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL
INTEGER SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL

COMMON /PHYSICO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

COMMON /STATE/ LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL
LOGICAL LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL

CHARACTER*4 UMETR(2),UFEET(2),UNITS(2)
DIMENSION BKL(2),IBKWS(2)
DIMENSION SKL(2),SKH(2),ISKWS(2)
DIMENSION RRT(2),RTP(2),RDMS(2),RDMCH(2),IRDWS(2)
DIMENSION SBORT(2),SBIRT(2),SBTIP(2),SBOMC(2),SBIMC(2),
2 SBOMS(2),SBIMS(2),ISBWBS(2)
DIMENSION FRT(2),FTP(2),FNMS(2),FNMCH(2),IFNWS(2)
REAL LPPND,LCBND,LCFND,KGND,GMND,KM,KMND,KB,KBND
CHARACTER*4 METER,TONSM,TONSE,TON

DATA METER //'METER'
DATA UMETR //' MET', 'ERS '/
DATA UFEET //' FEE', 'T '/
DATA TONSM //' M. '/
DATA TONSE //' L. '/

UNITS(1) = UMETR(1)
UNITS(2) = UMETR(2)
TON = TONSM
IF (PUNITS(1) .NE. METER) UNITS(1) = UFEET(1)
IF (PUNITS(1) .NE. METER) UNITS(2) = UFEET(2)
IF (PUNITS(1) .NE. METER) TON = TONSE

LPPND = LPP/BEAM
BEAMND = BEAM/DRAFT
DRFTND = DRAFT/BEAM

* convert displacement from mass to tons

DISPLT = MASS*.001
IF (PUNITS(1) .NE. METER) DISPLT = MASS * GRAV / 2240.
VOL = MASS/RHO
DISPND = VOL/(0.1*LPP)**3
IF (PUNITS(1) .NE. METER) DISPND = DISPLT/(0.01*LPP)**3
LCBND = LCB/LPP
LCFND = LCF/LPP
VCGND = VCG/BEAM
KGND = KG/BEAM
GMND = GM/BEAM
KM = KG+GM
KMND = KM/BEAM

```

```

KB = VCB + TLCB
KBND = KB/BEAM

*      waterplane and wetted surface
AWPND = AWP/(LPP*BEAM)
WSRFND = WSURF/(2.*LPP*DRAFT + 2.*BEAM*DRAFT + LPP*BEAM)

*      gyroradii
RGYRAD = KROLL*BEAM
PGYRAD = KPITCH*LPP
YGYRAD = KYAW*LPP

*      hydrostatic coefficients
CB = NEBLA/(LPP*BEAM*DRAFT)
CX = AREAMX/(BEAM*DRAFT)
CP = CB/CX

*      convert design speed in knots to froude number
*      estimated roll period based on equation -
ROLPER = (TPI/SQRT(GRAV)) * SQRT( (RGYRAD**2+A44)/GM )
where A44 = 0.25*RGYRAD**2

ROLPER = (TPI/SQRT(GRAV)) * SQRT(1.25*RGYRAD**2/GM)
ROLFRQ = TPI/ROLPER
SS = LPP/20.

*      bilge keel
IF (NBKSET .EQ. 0) GO TO 15
DO 10 IBK=1,NBKSET
BKL(IBK) = (BKAS(IBK) - BKFS(IBK)) * SS
IBKWS(IBK) = 4. * BKL(IBK) * BKWD(IBK)
10 CONTINUE

*      skeg
15 IF (NSKSET .EQ. 0) GO TO 25
DO 20 ISK=1,NSKSET
SKL(ISK) = (SKALS(ISK) - SKFLS(ISK)) * SS
SKH(ISK) = (SKAUWL(ISK) - SKALWL(ISK))
FACTOR = 1.0
IF (SKHB(ISK) .GT. 0.) FACTOR = 2.0
ISKWS(ISK) = FACTOR * SKL(ISK) * SKH(ISK)
20 CONTINUE

*      rudder
25 IF(NRDSET .EQ. 0) GO TO 35
DO 30 IRD=1,NRDSET
RRT(IRD) = (RDRAS(IRD) - RDRFS(IRD)) * SS
RTP(IRD) = (RDTAS(IRD) - RDTFS(IRD)) * SS
A = RDTHB(IRD) - RDRHB(IRD)
B = ((RDRFWL(IRD)+RDRAWL(IRD)) - (RDTFWL(IRD)+RDTAWL(IRD))) / 2
RDMSP(IRD) = SQRT(A*A + B*B)
RDMCH(IRD) = (((RDRAS(IRD)+RDTAS(IRD)) - (RDRFS(IRD)+RDTFS(IRD)))
2 /2) * SS
FACTOR = 2.0
IF (RDRHB(IRD) .GT. 0.) FACTOR = 4.0
IRDWS(IRD) = FACTOR * RDMSP(IRD) * RDMCH(IRD)
30 CONTINUE

*      propeller shaft brackets
35 IF (NSBSET .EQ. 0) GO TO 45
DO 40 ISB=1,NSBSET
SBORT(ISB) = (SOBRAS(ISB) - SOBRFS(ISB)) * SS
SBTIP(ISB) = (SBTAS(ISB) - SBTFS(ISB)) * SS

```

```

1
2 SBOMC(ISB) = (((SOBRAS(ISB)+SBTAS(ISB)) - (SOBRFS(ISB) +
3 SBTFS(ISB))) / 2) * SS
4 A = SOBRHB(ISB) - SBTHB(ISB)
5 B = ((SOBRFW(ISB)+SOBRAW(ISB)) - (SBTFWL(ISB)+SBTAWL(ISB))) / 2
6 SBOMS(ISB) = SQRT(A*A + B*B)
7 FACTOR = 4.0
8 ISBWS(ISB) = FACTOR * (SBOMS(ISB)*SBOMC(ISB))
9 IF (SBTHB(ISB) .EQ. 0.) GO TO 40
10 SBIRT(ISB) = (SIBRAS(ISB) - SIBRFS(ISB)) * SS
11 SBIMC(ISB) = (((SIBRAS(ISB)+SBTAS(ISB)) - (SIBRFS(ISB) +
12 SBTFS(ISB))) / 2) * SS
13 A = SBTHB(ISB) - SIBRHB(ISB)
14 B = ((SIBRFW(ISB)+SIBRAW(ISB)) - (SBTFWL(ISB)+SBTAWL(ISB))) / 2
15 SBIMS(ISB) = SQRT(A*A + B*B)
16 ISBWS(ISB) = ISBWS(ISB) + (FACTOR * (SBIMS(ISB)*SBIMC(ISB)))
17
18 40 CONTINUE
19
20 * fin
21
22 45 IF (NFNSET .EQ. 0) GO TO 55
23 DO 50 IFN=1,NFNSET
24 FRT(IFN) = (FNRAS(IFN) - FNRFs(IFN)) * SS
25 FTP(IFN) = (FNTAS(IFN) - FNTFs(IFN)) * SS
26 A = FNTHB(IFN) - FNRHB(IFN)
27 B = ((FNRFWL(IFN)+FNRAWL(IFN)) - (FNTFWL(IFN)+FNTAWL(IFN))) / 2
28 FNMSP(IFN) = SQRT(A*A + B*B)
29 FNMCH(IFN) = (((FNRAS(IFN)+FNTAS(IFN)) - (FNRFs(IFN)+FNTFs(IFN)))
30 2 /2) * SS
31 FACTOR = 2.0
32 IF (FNRHB(IFN) .GT. 0) FACTOR = 4.0
33 IFNWS(IFN) = FACTOR * FNMSP(IFN) * FNMCH(IFN)
34 50 CONTINUE
35 55 CONTINUE
36
37 * transform real variables to integers
38
39 IDISPL = DISPLT + .5001
40 IAWP = AWP + .5001
41 IWSURF = WSURF + .5001
42
43 * **** ship particulars table ****
44
45 WRITE (IPRIN,1000)
46 WRITE (IPRIN,1005) (TITLE(I),I=1,20)
47 WRITE (IPRIN,1010)
48 WRITE (IPRIN,1015)
49 WRITE (IPRIN,1020) LPP,(UNITS(I),I=1,2),LPPND
50 WRITE (IPRIN,1025) BEAM,(UNITS(I),I=1,2),BEAMND
51 WRITE (IPRIN,1030) DRAFT,(UNITS(I),I=1,2),DRFTND
52 IF (PUNITS(1) .NE. METER) WRITE (IPRIN,1035) DISPLT,TON,DISPND
53 IF (PUNITS(1) .EQ. METER) WRITE (IPRIN,1036) DISPLT,TON,DISPND
54 WRITE (IPRIN,1040) VKDES,FNDES
55 WRITE (IPRIN,1045)
56 WRITE (IPRIN,1050) VCG,(UNITS(I),I=1,2),VCGND
57 WRITE (IPRIN,1055) KG,(UNITS(I),I=1,2),KGND
58 WRITE (IPRIN,1060) GM,(UNITS(I),I=1,2),GMND
59 WRITE (IPRIN,1065) KM,(UNITS(I),I=1,2),KMND
60 WRITE (IPRIN,1070) KB,(UNITS(I),I=1,2),KBND
61 WRITE (IPRIN,1075)
62 WRITE (IPRIN,1080) LCB,(UNITS(I),I=1,2),LCBND
63 WRITE (IPRIN,1085) LCB,(UNITS(I),I=1,2),LCBND
64 WRITE (IPRIN,1090) LCF,(UNITS(I),I=1,2),LCFND
65 WRITE (IPRIN,1095)
66 WRITE (IPRIN,2000) RGYRAD,(UNITS(I),I=1,2),KROLL
67 WRITE (IPRIN,2005) PGYRAD,(UNITS(I),I=1,2),KPITCH
68 WRITE (IPRIN,2010) YGYRAD,(UNITS(I),I=1,2),KYAW
69 WRITE (IPRIN,2015) ROLPER,ROLFRQ
70 WRITE (IPRIN,2020)

```

```

        WRITE (IPRIN,2025) AWP,(UNITS(I),I=1,2),AWPND
        WRITE (IPRIN,2030) WSURF,(UNITS(I),I=1,2),WSRFND
        WRITE (IPRIN,1010)
        WRITE (IPRIN,2040)
        WRITE (IPRIN,2045) CB,CX,CP
        WRITE (IPRIN,2050)

***** appendage particulars table ****

        WRITE (IPRIN,1000)
        WRITE (IPRIN,2999) (TITLE(I),I=1,20)
        WRITE (IPRIN,1010)
        IF (NBKSET .EQ. 0) GO TO 100
        WRITE (IPRIN,3000)
        WRITE (IPRIN,3005) BKL(1),(UNITS(I),I=1,2)
        IF (NBKSET .EQ. 2) WRITE(IPRIN,3010) BKL(2),(UNITS(I),I=1,2)
        WRITE (IPRIN,3015) BKWD(1),(UNITS(I),I=1,2)
        IF (NBKSET .EQ. 2) WRITE(IPRIN,3010) BKWD(2),(UNITS(I),I=1,2)
        WRITE (IPRIN,3020) IBKWS(1),(UNITS(I),I=1,2)
        IF (NBKSET .EQ. 2) WRITE(IPRIN,3025) IBKWS(2),(UNITS(I),I=1,2)
        WRITE (IPRIN,1010)
100   IF (NSKSET .EQ. 0) GO TO 110
        WRITE (IPRIN,3030)
        WRITE (IPRIN,3035) SKL(1),(UNITS(I),I=1,2)
        IF (NSKSET .EQ. 2) WRITE(IPRIN,3040) SKL(2),(UNITS(I),I=1,2)
        WRITE (IPRIN,3045) SKH(1),(UNITS(I),I=1,2)
        IF (NSKSET .EQ. 2) WRITE(IPRIN,3050) SKH(2),(UNITS(I),I=1,2)
        WRITE (IPRIN,3055) ISKWS(1),(UNITS(I),I=1,2)
        IF (NSKSET .EQ. 2) WRITE(IPRIN,3060) ISKWS(2),(UNITS(I),I=1,2)
        WRITE (IPRIN,1010)
110   IF (NRDSET .EQ. 0) GO TO 120
        WRITE (IPRIN,3065)
        WRITE (IPRIN,3070) RRT(1),(UNITS(I),I=1,2)
        IF (NRDSET .EQ. 2) WRITE(IPRIN,3075) RRT(2),(UNITS(I),I=1,2)
        WRITE (IPRIN,3080) RTP(1),(UNITS(I),I=1,2)
        IF (NRDSET .EQ. 2) WRITE(IPRIN,3075) RTP(2),(UNITS(I),I=1,2)
        WRITE (IPRIN,3090) RDMS(1),(UNITS(I),I=1,2)
        IF (NRDSET .EQ. 2) WRITE(IPRIN,3095) RDMS(2),(UNITS(I),I=1,2)
        WRITE (IPRIN,3100) IRDWS(1),(UNITS(I),I=1,2)
        IF (NRDSET .EQ. 2) WRITE(IPRIN,3105) IRDWS(2),(UNITS(I),I=1,2)
        WRITE (IPRIN,1010)
120   IF (NSBSET .EQ. 0) GO TO 130
        WRITE (IPRIN,3110)
        WRITE (IPRIN,3115) SBORT(1),(UNITS(I),I=1,2)
        IF (NSBSET .EQ. 2) WRITE(IPRIN,3120) SBORT(2),(UNITS(I),I=1,2)
        WRITE (IPRIN,3145) SBOMS(1),(UNITS(I),I=1,2)
        IF (NSBSET .EQ. 2) WRITE(IPRIN,3150) SBOMS(2),(UNITS(I),I=1,2)
        WRITE (IPRIN,3135) SBTIP(1),(UNITS(I),I=1,2)
        IF (NSBSET .EQ. 2) WRITE(IPRIN,3140) SBTIP(2),(UNITS(I),I=1,2)
        IF (SBTHR(1).EQ.0 .AND. NSBSET.EQ.1) GO TO 125
        IF (SBTHR(1).EQ.0 .AND. NSBSET.EQ.2) GO TO 122
        WRITE (IPRIN,3125) SBIRT(1),(UNITS(I),I=1,2)
        IF (NSBSET .EQ. 2) WRITE(IPRIN,3120) SBIRT(2),(UNITS(I),I=1,2)
        WRITE (IPRIN,3155) SBIMS(1),(UNITS(I),I=1,2)
        IF (NSBSET .EQ. 2) WRITE(IPRIN,3150) SBIMS(2),(UNITS(I),I=1,2)
        GO TO 125
122   WRITE (IPRIN,3126) SBIRT(2),(UNITS(I),I=1,2)
        WRITE (IPRIN,3156) SBIMS(2),(UNITS(I),I=1,2)
125   WRITE (IPRIN,3165) ISBWS(1),(UNITS(I),I=1,2)
        IF (NSBSET .EQ. 2) WRITE(IPRIN,3170) ISBWS(2),(UNITS(I),I=1,2)
        WRITE (IPRIN,1010)
130   IF (NFNSET .EQ. 0) GO TO 140
        WRITE (IPRIN,3175)
        WRITE (IPRIN,3180) FRT(1),(UNITS(I),I=1,2)
        IF (NFNSET .EQ. 2) WRITE(IPRIN,3185) FRT(2),(UNITS(I),I=1,2)
        WRITE (IPRIN,3190) FTP(1),(UNITS(I),I=1,2)
        IF (NFNSET .EQ. 2) WRITE(IPRIN,3185) FTP(2),(UNITS(I),I=1,2)
        WRITE (IPRIN,3200) FNMS(1),(UNITS(I),I=1,2)
        IF (NFNSET .EQ. 2) WRITE(IPRIN,3205) FNMS(2),(UNITS(I),I=1,2)
        WRITE (IPRIN,3210) IFNWS(1),(UNITS(I),I=1,2)
        IF (NFNSET .EQ. 2) WRITE(IPRIN,3215) IFNWS(2),(UNITS(I),I=1,2)
140   CONTINUE

```

```

        WRITE (IPRIN,3300)
1000  FORMAT (1H1)
1005  FORMAT (1HO,1X,20A4//28X,25HTABLE OF SHIP PARTICULARS)
1010  FORMAT (/)
1015  FORMAT (10X,22HSHIP CHARACTERISTICS -,/)
1020  FORMAT (10X,24H  SHIP LENGTH (LPP)      , F7.2,2A4,
2 4X,12HLENGTH/BEAM ,7X,F7.3)
1025  FORMAT (10X,25H  BEAM AT MIDSHIPS      ,F6.2,2A4,
2 4X,12HBEAM/DRAFT ,7X,F7.3)
1030  FORMAT (10X,25H  DRAFT AT MIDSHIPS     ,F6.2,2A4,
2 4X,12HDRAFT/BEAM ,7X,F7.3)
1035  FORMAT (10X,23H  DISPLACEMENT (S.W.)   ,F8.1,A4,4HTONS,
2 4X,17HDISPL/(.01LPP)**3,1X,F8.3)
1036  FORMAT (10X,23H  DISPLACEMENT (S.W.)   ,F8.1,A4,4HTONS,4X,
2 17HVOLUME/(.1LPP)**3,1X,F8.3)
1040  FORMAT (10X,25H  DESIGN SHIP SPEED    ,F6.2,8H KNOTS ,
2 4X,13HFROUDE NUMBER,6X,F7.3)
1045  FORMAT (10X,20HVERTICAL LOCATIONS -,/)
1050  FCRMAT (10X,25H  C. OF GRAVITY (VCG)* ,F6.2,2A4,
2 4X,12HVCG/BEAM ,7X,F7.3)
1055  FORMAT (10X,25H  C. OF GRAVITY (KG)** ,F6.2,2A4,
2 4X,12HKG/BEAM ,7X,F7.3)
1060  FORMAT (10X,25H  METACENTRIC HT. (GM)  ,F6.2,2A4,
2 4X,12HGM/BEAM ,7X,F7.3)
1065  FORMAT (10X,25H  METACENTER (KM)**   ,F6.2,2A4,
2 4X,12HKM/BEAM ,7X,F7.3)
1070  FORMAT (10X,25H  C. OF BUOYANCY (KB)** ,F6.2,2A4,
2 4X,12HKB/BEAM ,7X,F7.3)
1075  FORMAT (10X,27HLONGITUDINAL LOCATIONS*** -,/)
1080  FORMAT (10X,25H  C. OF GRAVITY (LCG)   ,F6.2,2A4,
2 4X,12HLCG/LENGTH ,7X,F7.3)
1085  FORMAT (10X,25H  C. OF BUOYANCY (LCB)   ,F6.2,2A4,
2 4X,12HLCB/LENGTH ,7X,F7.3)
1090  FORMAT (10X,25H  C. OF FLOTATION (LCF)  ,F6.2,2A4,
2 4X,12HLCF/LENGTH ,7X,F7.3)
1095  FORMAT (10X,24HMOTION CHARACTERISTICS -,/)
2000  FORMAT (10X,25H  ROLL GYRADIUS       ,F6.2,2A4,
2 4X,12HRG/BEAM ,7X,F7.3)
2005  FORMAT (10X,25H  PITCH GYRADIUS     ,F6.2,2A4,
2 4X,12HPG/LPP ,7X,F7.3)
2010  FORMAT (10X,25H  YAW GYRADIUS      ,F6.2,2A4,
2 4X,12HYG/LPP ,7X,F7.3)
2015  FORMAT (10X,25H  ESTIMATED ROLL PERIOD ,F6.2,8H SECONDS,
2 4X,19HROLL FREQ (RADIAN),F7.3)
2020  FORMAT (10X,16HCOMPUTED AREAS -,/)
2025  FORMAT (10X,23H  WATERPLANE        ,F8.1,4H SQ.,2A4,
2 14HAWP/(LPP*BEAM),5X,F7.3)
2030  FORMAT (10X,23H  WETTED SURFACE, HULL ,F8.1,4H SQ.,2A4,
2 15HWS/(2LD+2RD+LB),4X,F7.3)
2040  FORMAT (10X,19HHULL COEFFICIENTS -,/)
2045  FORMAT (10X,25H  BLOCK (CB)         ,F6.3,/
2 10X,25H  SECTION (CX)           ,F6.3,/
3 10X,25H  PRISMATIC (CP)         ,F6.3)
2050  FORMAT (1HO,/10X,22H* WATERLINE REFERENCE,
2 /10X,17H** KEEL REFERENCE
2 /10X,17H***F.P. REFERENCE)
2999  FORMAT (1HO,1X,20A4//23X,35HTABLE OF SHIP APPENDAGE PARTICULARS)
3000  FORMAT (10X,28HBILGE KEEL CHARACTERISTICS -,/)
3005  FORMAT (12X,29HBILGE KEEL LENGTH (SET NO. 1),20X,F7.2,2A4)
3010  FORMAT (30X,11H(SET NO. 2),20X,F7.2,2A4)
3015  FORMAT (12X,29HBILGE KEEL WIDTH (SET NO. 1),20X,F7.2,2A4)
3020  FORMAT (12X,42HTOTAL WETTED SURFACE AREA (B.K. SET NO. 1),7X,I7,
2 4H SQ.,2A4)
3025  FORMAT (38X,16H(B.K. SET NO. 2),7X,I7,4H SQ.,2A4)
3030  FORMAT (10X,22HSKEG CHARACTERISTICS -,/)
3035  FORMAT (12X,34HSKEG LENGTH ALONG KEEL (SET NO. 1),15X,F7.2,2A4)
3040  FORMAT (36X,11H(SET NO. 2),15X,F7.2,2A4)
3045  FORMAT (12X,23HSKEG HEIGHT (SET NO. 1),26X,F7.2,2A4)
3050  FORMAT (24X,11H(SET NO. 2),26X,F7.2,2A4)
3055  FORMAT (12X,42HTOTAL WETTED SURFACE AREA (SKEG SET NO. 1),7X,I7,
2 4H SQ.,2A4)
3060  FORMAT (38X,16H(SKEG SET NO. 2),7X,I7,4H SQ.,2A4)

```

```

3065 FORMAT (10X,24HRUDDER CHARACTERISTICS -,/),
3070 FORMAT (12X,36HRUDDER ROOT CHORD LENGTH (SET NO. 1),13X,F7.2,2A4)
3075 FORMAT (37X,11H(SET NO. 2),13X,F7.2,2A4)
3080 FORMAT (12X,36HRUDDER TIP CHORD LENGTH (SET NO. 1),13X,F7.2,2A4)
3090 FORMAT (12X,28HRUDDER MEAN SPAN (SET NO. 1),21X,F7.2,2A4)
3095 FORMAT (29X,11H(SET NO. 2), 21X,F7.2,2A4)
3100 FORMAT (12X,44HTOTAL WETTED SURFACE AREA (RUDDER SET NO. 1),5X,
2 I7,4H SQ.,2A4)
3105 FORMAT (38X,18H(RUDDER SET NO. 2),5X,I7,4H SQ.,2A4)
3110 FORMAT (10X,42HPROPELLER SHAFT BRACKETS CHARACTERISTICS -,/),
3115 FORMAT (12X,45HOUTSIDE BRACKET ROOT CHORD LENGTH (SET NO. 1),4X,
2 F7.2,2A4)
3120 FORMAT (46X,11H(SET NO. 2),4X,F7.2,2A4)
3125 FORMAT (12X,45HINSIDE BRACKET ROOT CHORD LENGTH (SET NO. 1),4X,
2 F7.2,2A4)
3126 FORMAT (12X,45HINSIDE BRACKET ROOT CHORD LENGTH (SET NO. 2),4X,
2 F7.2,2A4)
3135 FORMAT (12X,37HBRACKET TIP CHORD LENGTH (SET NO. 1),12X,F7.2,
2 2A4)
3140 FORMAT (38X,11H(SET NO. 2),12X,F7.2,2A4)
3145 FORMAT (12X,37HOUTSIDE BRACKET MEAN SPAN (SET NO. 1),12X,F7.2,
2 2A4)
3150 FORMAT (38X,11H(SET NO. 2),12X,F7.2,2A4)
3155 FORMAT (12X,37HINSIDE BRACKET MEAN SPAN (SET NO. 1),12X,F7.2,
2 2A4)
3156 FORMAT (12X,37HINSIDE BRACKET MEAN SPAN (SET NO. 2),12X,F7.2,
2 2A4)
3165 FORMAT (12X,45HTOTAL WETTED SURFACE AREA (BRACKET SET NO. 1),4X,
2 I7,4H SQ.,2A4)
3170 FORMAT (38X,19H(BRACKET SET NO. 2),4X,I7,4H SQ.,2A4)
3175 FORMAT (10X,21HFIN CHARACTERISTICS -,/),
3180 FORMAT (12X,33HFIN ROOT CHORD LENGTH (SET NO. 1),16X,F7.2,2A4)
3185 FORMAT (34X,11H(SET NO. 2),16X,F7.2,2A4)
3190 FORMAT (12X,33HFIN TIP CHORD LENGTH (SET NO. 1),16X,F7.2,2A4)
3200 FORMAT (12X,25HFIN MEAN SPAN (SET NO. 1),24X,F7.2,2A4)
3205 FORMAT (26X,11H(SET NO. 2),24X,F7.2,2A4)
3210 FORMAT (12X,41HTOTAL WETTED SURFACE AREA (FIN SET NO. 1),8X,I7,
2 4H SQ.,2A4)
3215 FORMAT (38X,15H(FIN SET NO. 2),8X,I7,4H SQ.,2A4)
3200 FORMAT (1H0,9X,39HNOTE: IF A "SET" REPRESENTS A PAIR OF
2 31HAPPENDAGES (E.G., BILGE KEELS),/,17X,16HTHEN THE WETTED
2 46HSURFACE IS COMPUTED FOR THE TOTAL AREA OF BOTH,/,17X,
2 11HAPPENDAGES.)

```

```

RETURN
END

```

```

C DECK HYD2D
SUBROUTINE HYD2D

```

```

* fits spline to 2-d potentials and forces as function of frequency
* SIGMA is array of frequencies
* PHI2D is array of 2-d velocity potentials
* (frequency, node, mode)
* PHELM is array of spline segments for PHI2D versus SIGMA
*  

COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,MRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,MRANG,NNMU(8),
REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,

```

```

4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPIFCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /INDEX/ PFIDX,LPFIDX,RMIDX,LRMIDX,SVIDX,LSVIDX
INTEGER LPFIDX,LRMIDX,LSVIDX
REAL PFIDX(235),RMIDX(183),SVIDX(3)

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL
INTEGER SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL

COMMON /PELEM/ PELEM
COMPLEX PELEM(4,1000)

COMMON /TWOD/ YY, ZZ, ENN, ISTA
INTEGER ISTA
REAL YY(10,25),ZZ(10,25),ENN(4,10,25)

COMPLEX PHI2D(10,10,4),PHELM(4,9,40)
EQUIVALENCE (PELEM(1,1),PHI2D(1,1,1)),(PELEM(1,101),PHELM(1,1,1))

READ (SCRFILE) YY,ZZ,ENN,ISTA

DO 30 K=1,NSTATN
NPT = NOFSET(K)
IF (NPT .LT. 2) GO TO 30

* compute 2d potentials
CALL TWODPT (K,Y(1,K),Z(1,K),NPT,PHI2D)
* compute spline coefficients for PHI2D
DO 20 L=1,4
LM = (L-1)*10
DO 10 J=1,NPT
M = LM + J
CALL CPFIT (SIGMA,PHI2D(1,J,L),PHELM(1,1,M),NSIGMA)
10 CONTINUE
20 CONTINUE
* write spline coefficients to potential file
CALL WTPELM (K,PHELM)
30 CONTINUE
RETURN
END

C DECK HYDCAL
SUBROUTINE HYDCAL

COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
2 LRAOPR,ADRPR,ORGOPTN,GMNOM,KG,STATN(25),NSOFST(25),
2 NLEWF(25),HLFBTH(10,25),WTRLNE(10,25),BLEWF(25),TLEWF(25),
2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
2 ZPTFED(10),FBCODE(10),FLTYPE,RDOT(10),VKDES,FNDES,
2 STATNM,STATIS
CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
2 FBNUMB,PTNUMB,ORGOPTN
REAL KG

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL
INTEGER SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL

```

```

2 SPTFIL,LACFIL,LAEFIL

COMMON /SMPSYS/ FIS,AS,SIS,SOS,SDS,HALOS,DEV,PRN,SMPPS,SMPIS,
2 SMPOS,SMPDS,SHPTYPS,SHIPS,VARS,CYCLS,TITLES,OPTION,LSIS,LSOS,
2 LSDS,LHALOS,LDEV,LPRN,LSMPPS,LSMPIS,LSMPOS,LSMPDS,LSHPTYPS,
2 LSHIPS,LTITLES
CHARACTER*160 AS
CHARACTER*80 FIS,SIS,SOS,SDS,TITLES
CHARACTER*20 HALOS,DEV,PRN,SMPPS,SMPIS,SMPOS,SMPDS,SHPTYPS
CHARACTER SHIPS*6,VARS*2,CYCLS*2
INTEGER*2 OPTION

COMMON/STATE/LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL
LOGICAL LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL

IF (OPTN .GT. 3) GO TO 10

FIS = SDS(1:LSDS)//'.SCR'
OPEN (UNIT=SCRFIL,FILE=FIS,FORM='UNFORMATTED',STATUS='UNKNOWN')

FIS = SDS(1:LSDS)//'.POT'
OPEN (UNIT=POTFIL,FILE=FIS,STATUS='UNKNOWN',
2 ACCESS='DIRECT',RECL=1750)

FIS = SDS(1:LSDS)//'.COF'
OPEN (UNIT=COFFIL,FILE=FIS,FORM='UNFORMATTED',STATUS='UNKNOWN')

FIS = SDS(1:LSDS)//'.LCO'
IF (LOADS)
2 OPEN (UNIT=LCOFIL,FILE=FIS,FORM='UNFORMATTED',STATUS='UNKNOWN')

AS = '(/4X,"CALLING HYD2D")'
WRITE (*,AS)
WRITE (TEXFIL,AS)

CALL HYD2D

AS = '(4X,"CALLING T3DAMD")'
WRITE (*,AS)
WRITE (TEXFIL,AS)

CALL T3DAMD

AS = '(4X,"CALLING COFOUT")'
WRITE (*,AS)
WRITE (TEXFIL,AS)

CALL COFOUT

CLOSE (UNIT=SCRFIL)
CLOSE (UNIT=POTFIL)
CLOSE (UNIT=COFFIL)
IF (LOADS) CLOSE (UNIT=LCOFIL)

10 CONTINUE

RETURN
END

C DECK INERST
SUBROUTINE INERST (OMEGA,E,TV,TL)

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML ASTAT,BSTAT,TITLE,MASS,
2 D(SPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSF*(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL A(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,

```

```

4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /STATE/ LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL
LOGICAL LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL

COMPLEX TV(3,3),TL(3,3)

OMGE2 = OMEGAE*OMEGAE
IF (.NOT. VRT) GO TO 10

* vertical mode

* add mass, mass*vcg and inertia terms

    TV(1,1) = TV(1,1) - OMGE2*MASS
    TV(1,3) = TV(1,3) - OMGE2*MASS*VCG
    TV(2,2) = TV(2,2) - OMGE2*MASS
    TV(3,1) = TV(3,1) - OMGE2*MASS*VCG
    TV(3,3) = TV(3,3) - OMGE2*IPITCH

* add restoring terms

    TV(2,2) = TV(2,2) + CHEAVE
    TV(2,3) = TV(2,3) + CHEAPI
    TV(3,2) = TV(3,2) + CHEAPI
    TV(3,3) = TV(3,3) + CPITCH
10 IF (.NOT. LAT) GO TO 20

* lateral mode

* add mass, mass*vcg and inertia terms

    TL(1,1) = TL(1,1) - OMGE2*MASS
    TL(1,2) = TL(1,2) + OMGE2*MASS*VCG
    TL(2,1) = TL(2,1) + OMGE2*MASS*VCG
    TL(2,2) = TL(2,2) - OMGE2*IROLL
    TL(2,3) = TL(2,3) + OMGE2*IYAWRL
    TL(3,2) = TL(3,2) + OMGE2*IYAWRL
    TL(3,3) = TL(3,3) - OMGE2*IYAW

* add restoring term to roll

    TL(2,2) = TL(2,2) + CROLL
20 CONTINUE

RETURN
END

C DECK INPUT
SUBROUTINE INPUT

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL
INTEGER   SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL

COMMON /SMPSSYS/ FIS,AS,SIS,SOS,SDS,HALOS,DEV,PRN,SMPPS,SMPIS,
2 SMPDS,SHPTYPS,SHIPS,VARS,CYCLS,TITLES,OPTION,LSIS,LSOS,
2 LSDS,LHALOS,LDEV,LPRN,LSMPPS,LSMPIS,LSMPOS,LSMPDS,LSHPTYPS,
2 LSHIPS,LTITLES
CHARACTER*160 AS
CHARACTER*80 FIS,SIS,SOS,SDS,TITLES
CHARACTER*20 HALOS,DEV,PRN,SMPPS,SMPIS,SMPDS,SHPTYPS
CHARACTER SHIPS*6,VARS*2,CYCLS*2
INTEGER*2 OPTION

AS = '(/X,"CALLING PRELIM")'
WRITE (*,AS)
WRITE (TEXFIL,AS)

```

```

CALL PRELIM

FIS = SOS(1:LSOS)//'.OUT'
OPEN (UNIT=IPRIN,FILE=FIS,STATUS='UNKNOWN')

AS = '(4X,"CALLING AINPUT")'
WRITE (*,AS)
WRITE (TEXFIL,AS)

CALL AINPUT

AS = '(4X,"CALLING READ")'
WRITE (*,AS)
WRITE (TEXFIL,AS)

CALL READ

AS = '(4X,"CALLING HSTAT")'
WRITE (*,AS)
WRITE (TEXFIL,AS)

CALL HSTAT

AS = '(4X,"CALLING HSTOUT")'
WRITE (*,AS)
WRITE (TEXFIL,AS)

CALL HSTOUT

RETURN
END

C DECK INTRPL
SUBROUTINE INTRPL (N,XN,YN,M,XM,YM)

*   This routine obtains a finer resolution of a function using
*   linear interpolation. The function is assumed to be zero
*   outside of the frequency range of definition.
*   W.G.MEYERS, DTNSRDC, 072877

DIMENSION XN(N),YN(N),XM(M),YM(M)

KL = 1
KU = 2
DENOM = XN(KU) - XN(KL)
SLOPE = 0.
IF (DENOM .GT. 0.) SLOPE = (YN(KU) - YN(KL)) / DENOM
DO 20 I=1,M
IF {XM(I)} .LT. XN(KL)) GO TO 20
IF {XM(I)} .LE. XN(KU)) GO TO 10
5   KL = KL + 1
IF (KL .EQ. N) GO TO 30
KU = KU + 1
IF (XM(I) .GT. XN(KU)) GO TO 5
DENOM = XN(KU) - XN(KL)
SLOPE = 0.
IF (DENOM .GT. 0.) SLOPE = (YN(KU) - YN(KL)) / DENOM
10  YM(I) = YN(KL) + SLOPE * (XM(I) - XN(KL))
20  CONTINUE
30  CONTINUE

RETURN
END

C DECK IRGSEA
SUBROUTINE IRGSEA

COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLLDMPR,DISPLMT,
2 LRAOPR,ADRPR,ORGOPTN,GMNOM,KG,STATN(25),NSOFST(25),
2 NLEWF(25),HLEFBTH(10,25),WTRLNE(10,25),BLEWF(25),TLEWF(25),
2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),

```

```

2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
2 ZPTFBD(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
2 STATNM,STATIS
CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
2 FBNUMB,PTNUMB,ORGOPTN
REAL KG

IF (OPTN .EQ. 6 .OR. ORGOPTN .EQ. 1) GO TO 10
CALL RMSTOE
10 CONTINUE

RETURN
END

C DECK LIMIT
SUBROUTINE LIMIT (XLIM,YLIM,PSILIM,HEAD,FRNO,DEGRAD,FTMETR)

* This routine determines the limiting values of surge, sway or yaw
* regular wave dimensional transfer functions in quartering seas.
* This is to prevent blow-up for encounter frequencies near zero.
* surge and sway limits are nondimensional. The yaw limit is
* converted to deg/m from deg/ft as it was in the original
* source.
* W.G.MEYERS, DTNSRDC, 072977

* VERSION 1 - CDC 6700 - L I M I T - NOVEMBER, 1973
* S. BALES, A. E. BAITIS, W. MCCREIGHT

* Subroutine to impose limits on surge, sway, and yaw in quartering
* and following seas to prevent blow-up for near zero encounter
* frequencies. Limits are selected from experimental data :
* BAITIS - DE-1006 destroyer,
* WACHNIK & ZARNICK - A/C carrier,
* TASAI - single screw tanker.

* The limits should always be positive (surge) and may not be valid
* for froude numbers > 0.4

* Surge limit is a function of heading angle (degrees) and ship
* speed (froude number) and is in units of feet/feet.

XLIM = .8174 + 5.946 * FRNO - 0.020614 * HEAD

* Sway limit is a function of heading angle (degrees) and is in
* units of feet/feet.

YLIM = 0.0255 * HEAD + 0.3

* Yaw limit is a function of heading angle (degrees) and is in
* units of degrees/feet. A constant wave slope of one degree
* is assumed.

IF ((HEAD -40.) .LE. 0.005) PSILIM = 0.0206 * HEAD + 0.275
IF ((HEAD -40.) .GT. 0.005) PSILIM = 1.875 - 0.0193 * HEAD

* Yaw limit is converted to units of radians / meter

PSILIM = PSILIM * DEGRAD / FTMETR

RETURN
END

C DECK LRAO
SUBROUTINE LRAO (IM,NL,NU,MOTV,SF3,SH3,SA33,SB33,V,COSMU,
2 OMEGA,OMEGAE,IP,RAO,PHS,NMOT,NOMEGA,IPHS)

COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
2 LRAOPR,ADRPR,ORGOPTN,GMNOM,KG,STATN(25),NSDFST(25),
2 BLEWF(25),HLFETX(10,25),WTBLNE(10,25),BLEWF(25),TLEWF(25),
2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10);

```

```

2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
2 ZPTFBD(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
2 STATNM,STATIS
CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
2 FBNUMB,PTNUMB,ORGOPTN
REAL KG

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FEDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /LOADS/ NLOADS,SWGHT(25),SMASS(25),XLDSRN(10),XLDXPT(25),
2 LSTATN(25)

COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

COMPLEX MOTV(NMOT,NOMEGA),IWE,VIWE,HEAVE,HEAVEL,HEAACC,PITCH,
2 PITVEL,PITACC,VERVEL,VERACC,ZERO,INERT,RESTOR,EXCIT,CEP,
2 HYDRO,LOAD,SP3(25,NOMEGA),SH3(25,NOMEGA)
COMPLEX STEMP(25),ELEMS(4,25),EXF,SAB33(25),CDUM,HYD,CSUM
DIMENSION RAO(NOMEGA),PHS(NOMEGA),OMEGA(NOMEGA),OMEGAE(NOMEGA)
DIMENSION SA33(25,NOMEGA),SB33(25,NOMEGA)
CHARACTER*4 METER

DATA METER /'METER'/

ZERO = (0.,0.)
XP = XLDXPT(IP)
KSTATN = LSTATN(IP) - 1
NPS = NSTATN - KSTATN + 1
V2 = V*V
CON = 1000
IF (PUNITS(1) .NE. METER) CON = 2240
RHOG = RHO*GRAV
DO 100 I=NL,NU
W = OMEGA(I)
WN = W*W/GRAV
TEST = .005*TPI/LPP
ARGLI = - WN*COSH
IF (ABS(ARGLI) .LE. TEST) ARGLI = 0.
WE = OMEGAE(I)
WE2 = WE*WE
IWE = II*WE
VIWE = V/IWE
VWE2 = V/WE2
V2WE2 = V2/WE2
HEAVE = MOTV(2,I)
HEAVEL = IWE*HEAVE
HEAACC = IWE*HEAVEL
PITCH = MOTV(3,I)
PITVEL = IWE*PITCH
PITACC = IWE*PITVEL
VERVEL = HEAVEL - XP*PITVEL
VERACC = HEAACC - XP*PITACC

```

* inertia term

```

INERT = ZERO
M1 = KSTATN + 1
DO 10 K=M1,NSTATN
STEMP(K) = SMASS(K)*(HEAAC - X(K)*PITACC)
IF (IM .EQ. 13) STEMP(K) = -(X(K)-XP)*STEMP(K)
INERT = INERT + STEMP(K)
10 CONTINUE

* restoring term

DO 20 K=KS1ATN,NSTATN
NPT = NOFSET(K)
SBEAM = 2*Y(NPT,K)
STEMP(K) = SBEAM*(HEAVE - X(K)*PITCH)
IF (IM .EQ. 11) STEMP(K) = - STEMP(K)
IF (IM .EQ. 13) STEMP(K) = (X(K)-XP)*STEMP(K)
20 CONTINUE
CALL CPFIT (X(KSTATN),STEMP(KSTATN),ELEMS,NPS)
CALL CPINTG (XP,X(NSTATN),X(KSTATN),NPS,ELEMS,0.,RESTOR)
RESTOR = RHOG*RESTOR

* exciting term

DO 30 K=KSTATN,NSTATN
STEMP(K) = SF3(K,I) + SH3(K,I)
IF (IM .EQ. 13) STEMP(K) = - ((X(K)-XP)*STEMP(K) +
2 VIWE*SH3(K,I))
30 CONTINUE
CALL CPFIT (X(KSTATN),STEMP(KSTATN),ELEMS,NPS)
CALL CPINTG (XP,X(NSTATN),X(KSTATN),NPS,ELEMS,ARGLI,EXCIT)
IF (.NOT. IM.EQ.11) GO TO 35
CALL CPFIT (X(KSTATN),SH3(KSTATN,I),ELEMS,NPS)
CALL CPLVAL (X(KSTATN),NPS,ELEMS,XP,EXF,CDUM,IELM)
CEP = CEXP(II*XP*ARGLI)
EXCIT = EXCIT + VIWE*CEP*EXF
35 EXCIT = RHO*EXCIT

* hydrodynamic term

DO 40 K=KSTATN,NSTATN
A33 = SA33(K,I)
B33 = SB33(K,I)
SAB33(K) = A33 + II*B33
IF (IM .EQ. 11) STEMP(K) = - (A33*(HEAAC-X(K)*PITACC) +
2 B33*(HEAVEL-X(K)*PITVEL) - VWE2*B33*PITACC + V*A33*PITVEL)
IF (IM .EQ. 13) STEMP(K) = (X(K)-XP)*(A33*(HEAAC-X(K)*PITACC) +
2 + B33*(HEAVEL-X(K)*PITVEL)) + (V*A33*VERVEL - VWE2*B33*VERACC -
2 - V2WE2*(A33*PITACC + B33*PITVEL))
40 CONTINUE
CALL CPFIT (X(KSTATN),STEMP(KSTATN),ELEMS,NPS)
CALL CPINTG (XP,X(NSTATN),X(KSTATN),NPS,ELEMS,0.,HYDRO)
IF (.NOT. IM.EQ.11) GO TO 45
CALL CPFIT (X(KSTATN),SAB33(KSTATN),ELEMS,NPS)
CALL CPLVAL (X(KSTATN),NPS,ELEMS,XP,HYD,CDUM,IELM)
A33 = REAL(HYD)
B33 = AIMAG(HYD)
HYDRO = HYDRO - (V*A33*VERVEL - VWE2*B33*VERACC -
2 V2WE2*(A33*PITACC + B33*PITVEL))
45 CONTINUE
CSUM = RESTOR + EXCIT + HYDRO
LOAD = INERT - CSUM
LOAD = LOAD/CON
CALL RAOPHA (LOAD,RAO(I),PHS(I),RADDEG,IPHS)
100 CONTINUE

RETURN
END

C DECK LRAOUT
SUBROUTINE LRAOUT

COMMON /DATINP/ OPTN,MOTN,BSCFIL,VACPR,RAOPR,RLDMPR,DISPLMT,

```

```

2 LRAOPR,ADRPR,ORGOPTR,GMNOM,KG,STATN(25),NSOFST(25),
2 NLEWF(25),HLFBTH(10,25),WTRLNE(10,25),BLEWF(25),TLEWF(25),
2 AREALF(25),NPTLLOC,PTNUMB(10),PTNAME,XPTLLOC(10),YPTLLOC(10),
2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
2 ZPTFBD(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
2 STATNM,STATIS
CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
INTEGER OPTN,MOTN,BSCFIL,VLACPR,RADPR,RLDMPR,FBCODE,
2 FBNUMB,PTNUMB,ORGOPTR
REAL KG

COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
INTEGER NVK,NMU,OMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLAY,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLAY,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEOFIL
INTEGER SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEOFIL

COMMON /LOADS/ MLOADS,SWGHT(25),SMASS(25),XLDSRN(10),XLDXPT(25),
2 LSTATN(25)

COMMON /PHYSICO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

COMMON /SMPSSYS/ FIS,AS,SIS,SOS,SDS,HALOS,DEV,PRN,SMPPS,SMPIS,
2 SMPPOS,SMPDPS,SHPTYPS,SHIPS,VARS,CYCLS,TITLES,OPTION,LSIS,LSOS,
2 LSDS,LHALOS,LDEV,LPRN,LSMPPS,LSMPIS,LSMPOS,LSMPDPS,LSHPTYPS,
2 LSHPSS,TITLES
CHARACTER*160 AS
CHARACTER*80 FIS,SIS,SOS,SDS,TITLES
CHARACTER*20 HALOS,DEV,PRN,SMPPS,SMPIS,SMPPOS,SMPDPS,SHPTYPS
CHARACTER SHIPS*6,VARS*2,CYCLS*2
INTEGER*2 OPTION

COMMON /STATE/ LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL
LOGICAL LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL

COMPLEX MOTV(3,30),MOTL(3,30,8),HJV(3,30),HJL(3,30),H7(30),
2 SF3(25,30),SH3(25,30)
DIMENSION SA33(25,30),SE33(25,30),OMEGAE(30),VSFRAO(30),
2 VSFPHS(30),VBMRAO(30),VBMPPHS(30)
CHARACTER*4 METER
CHARACTER*2 UNITS

DATA METER /'METER'/
IF (PUNITS(1) .EQ. METER) UNITS = 'M'
IF (PUNITS(1) .NE. METER) UNITS = 'FT'

```

```

FIS = SDS(1:LSDS)///'.ORG'
OPEN (UNIT=ORGFIL,FILE=FIS,FORM='UNFORMATTED',STATUS='UNKNOWN')

FIS = SDS(1:LSDS)///'.ICO'
OPEN (UNIT=LCOFIL,FILE=FIS,FORM='UNFORMATTED',STATUS='UNKNOWN')

FIS = SDS(1:LSDS)///'.LRA'
OPEN (UNIT=LRAFIL,FILE=FIS,FORM='UNFORMATTED',STATUS='UNKNOWN')

READ (ORGFIL) TITLE,NVK,NMU,NOMEGA,OMEGA,NRANG,RLANG,VRT,LAT,
2 ADDRES,LPP,BEAM,DRAFT,DISPLM,GM,DELGM,KG,KROLL,LCB,GRAV,RHO,
2 VKDES,VKINC,DBLWL

WRITE (LRAFIL) TITLE,NOMEGA,OMEGA,NVK,NMU,LPP,BEAM,DRAFT,DISPLM,
2 GM,DELGM,KG,KROLL,LCB,DBLWL,GRAV,NSTATN,STATN,NLOADS,SWGHT,SMASS,
2 XLDSTN,XLDXPT,X

DO 300 IV=1,NVK
DO 200 IH=1,NMU
READ (ORGFIL) VKNOTS,HEADNG,OMEGAE
IF (VRT) READ (ORGFIL) MOTV
IF (LAT) READ (ORGFIL) MOTL
IF (ADDRES) READ (ORGFIL) HJV,HJL,H7
HDNG = 180. - HEADNG
COSMU = COS(MU(IH,IV))
DO 10 IW=1,NOMEGA
READ (LCOFIL) (SF3(I,IW),SH3(I,IW),SA33(I,IW),SB33(I,IW),
2 I=1,NSTATN)
10 CONTINUE
DO 100 IP=1,NLOADS
IM = 11
CALL LRAO (IM,1,NOMEGA,MOTV,SF3,SH3,SA33,SB33,VFS(IV),COSMU,
2 OMEGA,OMEGAE,IP,VSFRAO,VSFPHS,3,NOMEGA,1)
IM = 13
CALL LRAO (IM,1,NOMEGA,MOTV,SF3,SH3,SA33,SB33,VFS(IV),COSMU,
2 OMEGA,OMEGAE,IP,VBMRAO,VBMPHS,3,NOMEGA,1)
WRITE (IPRIN,1000) TITLE,XLDSTN(IP),VKNOTS,HDNG
1000 FORMAT (1H1,/,,28X,20A4,///,43X,
2 33HLOAD RESPONSE AMPLITUDE OPERATORS
2 18H (RAOS) AND PHASES,///,60X,7HSTATION,F5.1,///,55X,
2 12HSHIP SPEED =,F5.0,6H KNOTS,/,53X,14HSHIP HEADING =,
2 F5.0,8H DEGREES)
C WRITE (IPRIN,1030) TMODAL(KS),STATIS,(STATNM(I),I=1,3)
C1030 FORMAT (/54X'MODAL PERIOD ='F4.0' SECONDS'/54X'STATISTIC ='F5.2,
C 2 ' ('3A4')//')
C 2 WRITE (IPRIN,1010) UNITS
1010 FORMAT (/,20X,11HV.SHEAR(V3),9X,10HV.MOM.(V5),/,2X,
2 12HOMEGA OMEGAE,4X,2(15HAMPL. PHASE,4X),/,4X,3HRPS,4X,3HRPS,
2 4X,6H TONS,6X,3HDEG,4X,A2,5H-TONS,5X,3HDEG,/)
DO 20 IW=1,NOMEGA
WRITE (IPRIN,1020) OMEGA(IW),OMEGAE(IW),VSFRAO(IW),VSFPHS(IW),
2 VBMRAO(IW),VBMPHS(IW)
1020 FORMAT (2F7.3,2(1PE12.4,0PF7.1))
20 CONTINUE
WRITE (IPRIN,2100)
2100 FORMAT (//2X'NOTES: 1) VERTICAL RAOS ARE '
* 'LINEAR AND INDEPENDANT OF SEA STATE.'/ 9X'2) LATERAL RAOS '
* 'ARE NONLINEAR AND CHANGE WITH SEA STATE AND '
2 'STATISTIC.')
IF (PUNITS(1) .EQ. METER) WRITE (IPRIN,2110)
IF (PUNITS(1) .NE. METER) WRITE (IPRIN,2120)
2110 FORMAT (9X'3) AMPL. IS IN (PHYS.UNITS/METER)',2H**,,'2 AND PHASE '
2 'IS IN DEGREES.')
2120 FORMAT (9X'3) AMPL. IS IN (PHYS.UNITS/FOOT)',2H**,,'2 AND PHASE '
2 'IS IN DEGREES.')
WRITE (IPRIN,2130)
2130 FORMAT (9X'4) HEADING CONVENTION: 0 DEG=HEAD, 90 DEG=STBD BEAM,
2 '180 DEG=FOLLOWING SEAS.')
C WRITE (IPRIN,1030)
C1030 FORMAT (/,2X,45HNOTE: HEADING CONVENTION: 0 DEG=HEAD, 90 DEG=,
C 2 36H STBD BEAM, 180 DEG= FOLLOWING SEAS.)
WRITE (LRAFIL) XLDSTN(IP),VKNOTS,HDNG,OMEGAE,VSFRAO,VSFPHS,

```

```

2 VBMRAO,VBMPHS
100 CONTINUE
200 CONTINUE
300 CONTINUE

CLOSE (UNIT=ORGFIL)
CLOSE (UNIT=LCOFIL)
CLOSE (UNIT=LRAFIL)

RETURN
END

C DECK LSCOF
SUBROUTINE LSCOF (OMEGA,OMEGAE,IAPAM,SPAN,MCHORD,AREA,LCS,
2 GAMMA,XCP,YCP,ZCP,TLG,EXCLG,TLGC,EXLGC)

COMMON /CH3D/ ISIGMA,SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI,
2 IMMIN,IMMAX,IMDEL,LMIN,LMAX
REAL SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI(4)
INTEGER ISIGMA,IMMIN,IMMAX,IMDEL,LMIN,LMAX

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /HULL/ A26

COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

COMPLEX TLG(3,3),EXCLG(3),TLGC(3,3),EXLGC(3),F2,DF2,DF4,DF6
COMPLEX TF(3,3)
COMPLEX VIW,ZERO,CTEMP
REAL MCHORD,LCS
REAL I44G
LOGICAL HULL

EXTERNAL EXP

OMGE2 = OMEGAE*OMEGAE
ZERO = (0.,0.)
VIW = V/(II+OMEGAE)
V2W2 = (V/OMEGAE)**2
CK = OMEGA*OMEGA/GRAV
SINGAM = SIN(GAMMA*DEGRAD)
COSGAM = COS(GAMMA*DEGRAD)
SIN2GM = SINGAM*SINGAM
ARG = - CK*(XCP*COSMU + YCP*SINMU)
FZ = (RHO/2)*AREA*V*LCS
YHAT = YCP*COSGAM + ZCP*SINGAM
AP = 0
IF (IAPAM .NE. 1) GO TO 5

* added-mass due to bilgekeel

I44G = MASS*(KROLL*BEAM)**2
CB = NEBLA/(LPP*BEAM*DRAFT)
DA44BK = (.184 - .365*CB + .299*CB*CB)*I44G/2
AP = DA44BK/(YHAT*YHAT)

```

```

      GO TO 8
5 IF (IAPAM .EQ. 0) GO TO 8
* rudder or fin added-mass
   AP = PI*RHO*SPAN*(MCHORD/2)**2
8 CONTINUE
HULL = .FALSE.
IF (MCHORD .EQ. LPP) HULL = .TRUE.
IF (.NOT. HULL) GO TO 52

* hull
DF2 = ZERO
DF4 = ZERO
DF6 = ZERO
SP = 0
DO 42 L=1,NSTATN
IF (L .EQ. 1) DX = (X(2) - X(1))/2
IF (L .EQ. NSTATN) DX = (X(NSTATN) - X(NSTATN-1))/2
IF (L.GT.1 .AND. L.LT.NSTATN) DX = (X(L+1) - X(L-1))/2
DX = ABS(DX)
NPT = NOFSET(L)
IF (NPT .LT. 2) GO TO 42
T = ABS(Z(1,L))
Z2 = Z(1,L)/2
A = T*DX
SP = SP + A
F2 = FZ*OMEGA*(SINGAM*SINMU - II*COSGAM)*
2 CEXP(CK*(Z2 - II*X(L)*COSMU))
CTEMP = F2*SINGAM*A
DF2 = DF2 + CTEMP
DF6 = DF6 + X(L)*CTEMP + VIW*CTEMP
42 CONTINUE
DF2 = DF2/SP
DF6 = DF6/SP
CB = NEBLA/(LPP*BEAM*DRAFT)
CX = AREAMX/(BEAM*DRAFT)
CP = CB/CX
GO TO 62
52 CONTINUE
F2 = (FZ + II*OMEGA*AP)*OMEGA*(SINGAM*SINMU - II*COSGAM)
2 *EXP(CK*ZCP)*(COS(ARG) + II*SIN(ARG))
DF2 = F2*SINGAM
DF4 = - F2*YHAT
DF6 = XCP*DF2
62 CONTINUE
DA22 = AP*SIN2GM
DB22 = FZ*SIN2GM
DA24 = - AP*YHAT*SINGAM
DB24 = - FZ*YHAT*SINGAM
DA26 = XCP*DA22
DB26 = XCP*DB22 - V*DA22
DC26 = - V*DB22
DA42 = DA24
DB42 = DB24
DA44 = AP*YHAT*YHAT
DB44 = FZ*YHAT*YHAT
DA46 = XCP*DA24
DB46 = XCP*DB24 - V*DA24
DC46 = - V*DB24
DA62 = DA26
DB62 = XCP*DB22 + V*DA22
DC62 = V*DB22
DA64 = DA46
DB64 = XCP*DB24 + V*DA24
DC64 = V*DB24
DA66 = XCP*XCP*DA22
IF (.NOT. HULL) DB66 = XCP*ICP*DB22 + V2W2*DB22
IF (HULL) DB66 = (CP*LPP/2)**2 * DB22 + V*A26 + V2W2*DB22
DC66 = - V*V*DA22
TF(1,1) = - OMGE2*DA22 + II*OMEGA*DB22

```

```

TF(1,2) = - OMGE2*DA24 + II*OMEGAE*DB24
TF(1,3) = - OMGE2*DA26 + DC26 + II*OMEGAE*DB26
TF(2,1) = TF(1,2)
TF(2,2) = - OMGE2*DA44 + II*OMEGAE*DB44
TF(2,3) = - OMGE2*DA46 + DC46 + II*OMEGAE*DB46
TF(3,1) = - OMGE2*DA26 + DC62 + II*OMEGAE*DB62
TF(3,2) = - OMGE2*DA46 + DC64 + II*OMEGAE*DB64
TF(3,3) = - OMGE2*DA66 + DC66 + II*OMEGAE*DB66
DO 20 I=1,3
DO 10 J=1,3
TLGC(I,J) = TLG(I,J) + TF(I,J)
10 CONTINUE
20 CONTINUE
EXCLGC(1) = EXCLG(1) + DF2
EXCLGC(2) = EXCLG(2) + DF4
IF (.NOT. HULL) EXCLGC(3) = EXCLG(3) + DF6 + VIW*DF2
IF (HULL) EXCLGC(3) = EXCLG(3) + DF6

```

```

RETURN
END

```

```

C DECK NORMAL
SUBROUTINE NORMAL (PSEGS)

```

```

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

```

```

COMMON /WGHTS/ WTDL,NORM
REAL WTDL(10,25),NORM(4,10,25)

```

```

DIMENSION PSEGS(8,9,25)
DIMENSION V(2), P(2), PT(2), PM(2,5), SEGS(8,4), NDI(2),
2 ENDI(2,2)

DATA NDI, ENDI / 2 * 1, 4 * 0.0 /

* calculate 2d normals (n2, n3, n4) at nodes

DO 100 K=1,NSTATN
NP = NOFSET(K)
IF (NP .LT. 2) GO TO 100
DO 50 J=1, NP
IF (J .EQ. NP) GO TO 20
CY = PSEGS(3,J,K)
CZ = PSEGS(4,J,K)
GO TO 30
20 CONTINUE
JJ = J - 1
CY = PSEGS(7,JJ,K)
CZ = PSEGS(8,JJ,K)
30 CONTINUE
DEN = SQRT (CY*CY + CZ*CZ)
IF (DEN .GT. 0.0) GO TO 40
NORM(2,J,K) = 0.0
NORM(3,J,K) = 0.0
GO TO 45
40 CONTINUE
NORM(2,J,K) = -CZ / DEN
NORM(3,J,K) = CY / DEN
45 CONTINUE
NORM(4,J,K) = Y(J,K)*NORM(3,J,K) - Z(J,K)*NORM(2,J,K)
50 CONTINUE

```

```

100    CONTINUE
*
*      calculate longitudinal normals (n1) at nodes
DO 500 K=1,NSTATN
NP = NOFSET(K)
IF (NP .LT. 2) GO TO 500
NSEGS = NP - 1
IB = K - 2
IF (IB .GT. 1) GO TO 150
IB = 1
IF (NOFSET(1) .LT. 2) IB = 2
150    CONTINUE
IE = IB + 4
IF (IE .LT. NSTATN) GO TO 200
IE = NSTATN
IF (NOFSET(NSTATN) .LT. 2) IE = NSTATN - 1
IB = IE - 4
200    CONTINUE
DO 400 J=1,NP
II = 0
P(1) = Y(J,K)
P(2) = Z(J,K)
V(1) = NORM(3,J,K)
V(2) = -NORM(2,J,K)
DO 300 I=IB,IE
IF (I .NE. K) GO TO 220
II = II + 1
IK = II
PT(1) = P(1)
PT(2) = P(2)
PM(2,II) = 0.0
GO TO 290
220    CONTINUE
NSEGS = NOFSET(I) - 1
CALL SPPLV2 (V, P, PSEGS(1,1,I), NSEGS, PT, NI, TI, INT)
IF (INT .NE. 1) GO TO 300
II = II + 1
DY = PT(1) - P(1)
DZ = PT(2) - P(2)
AA = SQRT (DY*DY + DZ*DZ)
BB = DY*NORM(2,J,K) + DZ*NORM(3,J,K)
IF (BB .NE. 0.0) GO TO 280
PM(2,II) = 0.0
GO TO 290
280    CONTINUE
PM(2,II) = AA * BB / ABS(BB)
290    CONTINUE
PM(1,II) = X(I)
300    CONTINUE
IF (II .GT. 1) GO TO 320
NORM(1,J,K) = -0.0
GO TO 400
320    CONTINUE
CALL SPLNT2 (SEGS, PM, II, NDI, ENDI)
IF (IK .EQ. II) GO TO 340
CX = SEGS(3,IK)
CM = SEGS(4,IK)
GO TO 360
340    CONTINUE
IKK = IK - 1
CX = SEGS(7,IKK)
CM = SEGS(8,IKK)
360    CONTINUE
NORM(1,J,K) = -CM / SQRT (CX*CX + CM*CM)
400    CONTINUE
500    CONTINUE
RETURN
END

```

C DECK NORMTS

SUBROUTINE NORMT5 (PSEGS)

```

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,2PT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /TWOD/ YY, ZZ, ENN, ISTA
INTEGER ISTA
REAL YY(10,25),ZZ(10,25),ENN(4,10,25)

DIMENSION PSEGS(8,9,25), CC(14)
DIMENSION V(2), P(2), PT(2), PM(2,5), SEGS(8,4), NDI(2),
2 ENDI(2,2)

DATA NDI, ENDI / 2 * 1, 4 * 0.0 /

* calculate 2d normals (n2, n3, n4) at midpoints

T = 0.5
T2 = 0.25
T3 = 0.125
DO 100 K=1,NSTATN
NP = NOFSET(K) - 1
IF (NP .LT. 1) GO TO 100
DO 50 J=1,NP
CALL CUBCO2 (PSEGS(1,J,K), CC)
YY(J,K) = CC(1)*T3 + CC(3)*T2 + CC(5)*T + CC(7)
ZZ(J,K) = CC(2)*T3 + CC(4)*T2 + CC(6)*T + CC(8)
CY = CC(9)*T2 + CC(11)*T + CC(5)
CZ = CC(10)*T2 + CC(12)*T + CC(6)
DEN = SQRT (CY*CY + CZ*CZ)
IF (DEN .GT. 0.0) GO TO 40
ENN(2,J,K) = 0.0
ENN(3,J,K) = 0.0
GO TO 45
40 CONTINUE
ENN(2,J,K) = -CZ / DEN
ENN(3,J,K) = CY / DEN
45 CONTINUE
ENN(4,J,K) = YY(J,K)*ENN(3,J,K) - ZZ(J,K)*ENN(2,J,K)
50 CONTINUE
J = NP + 1
YY(J,K) = 0.5 * YY(NP,K)
ZZ(J,K) = 0.0
ENN(1,J,K) = 0.0
ENN(2,J,K) = 0.0
ENN(3,J,K) = -1.0
ENN(4,J,K) = -YY(J,K)
100 CONTINUE

* calculate longitudinal normals (n1) at midpoints

DO 500 K=1,NSTATN
NP = NOFSET(K) - 1
IF (NP .LT. 1) GO TO 500
NSEGS = NP
IB = K - 2
IF (IB .GT. 1) GO TO 150
IB = 1
IF (NOFSET(1) .LT. 2) IB = 2
CONTINUE
IE = IB + 4
IF (IE .LT. NSTATN) GO TO 200

```

```

IE = NSTATN
IF (NOFSET(NSTATN) .LT. 2) IE = NSTATN - 1
IB = IE - 4
200 CONTINUE
DO 400 J=1,NP
II = 0
P(1) = YY(J,K)
P(2) = ZZ(J,K)
V(1) = ENN(3,J,K)
V(2) = -ENN(2,J,K)
DO 300 I=IB,IE
IF (I .NE. K) GO TO 220
II = II + 1
IK = II
PT(1) = P(1)
PT(2) = P(2)
PM(2,II) = 0.0
GO TO 290
220 CONTINUE
NSEGS = NOFSET(I) - 1
CALL SPPLV2 (V, P, PSEGS(1,1,I), NSEGS, PT, NI, T1, INT)
IF (INT .NE. 1) GO TO 300
II = II + 1
DY = PT(1) - P(1)
DZ = PT(2) - P(2)
AA = SQRT (DY*DY + DZ*DZ)
BB = DY*ENN(2,J,K) + DZ*ENN(3,J,K)
IF (BB .NE. 0.0) GO TO 280
PM(2,II) = 0.0
GO TO 290
280 CONTINUE
PM(2,II) = AA * BB / ABS(BB)
290 CONTINUE
PM(1,II) = X(I)
300 CONTINUE
IF (II .GT. 1) GO TO 320
ENN(1,J,K) = - 0.0
GO TO 400
320 CONTINUE
CALL SPLNT2 (SEGS, PM, II, NDI, ENDI)
IF (IK .EQ. II) GO TO 340
CX = SEGS(3,IK)
CM = SEGS(4,IK)
GO TO 360
340 CONTINUE
IKK = IK - 1
CX = SEGS(7,IKK)
CM = SEGS(8,IKK)
360 CONTINUE
ENN(1,J,K) = -CM / SQRT (CX*CX + CM*CM)
400 CONTINUE
500 CONTINUE

RETURN
END

C DECK ORAO
SUBROUTINE ORAO (IM,NL,NU,MOTV,MOTL,RAO,PHS,NMOT,NOMEGA,
2 RADDEG,IPHS)

* This routine obtains the six degree of freedom response amplitude
* operators and phase angles (degrees) for waves from both port
* and starboard headings.
* W.G.MEYERS, DTNSRDC, 100577

COMPLEX MOTV(NMOT,NOMEGA),MOTL(NMOT,NOMEGA),TFN
DIMENSION RAO(NOMEGA),PHS(NOMEGA)

DO 10 I=NL,NU
IF (IM .EQ. 1) TFM = MOTV(1,I)
IF (IM .EQ. 2) TFM = MOTL(1,I)
IF (IM .EQ. 3) TFM = MOTV(2,I)

```

```

IF (IM .EQ. 4) TFN = MOTL(2,I)
IF (IM .EQ. 5) TFN = MOTV(3,I)
IF (IM .EQ. 6) TFN = MOTL(3,I)
IF (IM .GT. 3) TFN = TFN * RADDEG
CALL RAOPHA (TFN,RAO(I),PHS(I),RADDEG,IPHS)
10 CONTINUE

      RETURN
      END

C DECK ORGRAO
      SUBROUTINE ORGRAO (RLANG,NRANG,RLANS,MOTV,MOTL,NOMEGA,IM,RAO,PHS,
2 RADDEG)

      DIMENSION RLANG(8),RAO(30,6),PHS(30,6)
      COMPLEX MOTV(3,30),MOTL(3,30,8),CTFN

      DO 70 IW=1,NOMEGA
      GO TO (10,20,30,40,50,60),IM

*     surge
10   CALL RAOPHA (MOTV(1,IW),RAO(IW,IM),PHS(IW,IM),RADDEG,1)
      GO TO 70

*     sway
20   CALL TFNFFIT (RLANG,NRANG,RLANS,MOTL,1,IW,CTFN)
      CALL RAOPHA (CTFN,RAO(IW,IM),PHS(IW,IM),RADDEG,1)
      GO TO 70

*     heave
30   CALL RAOPHA (MOTV(2,IW),RAO(IW,IM),PHS(IW,IM),RADDEG,1)
      GO TO 70

*     roll
40   CALL TFNFFIT (RLANG,NRANG,RLANS,MOTL,2,IW,CTFN)
      CTFN = RADDEG*CTFN
      CALL RAOPHA (CTFN,RAO(IW,IM),PHS(IW,IM),RADDEG,1)
      GO TO 70

*     pitch
50   CTFN = RADDEG*MOTV(3,IW)
      CALL RAOPHA (CTFN,RAO(IW,IM),PHS(IW,IM),RADDEG,1)
      GO TO 70

*     yaw
60   CALL TFNFFIT (RLANG,NRANG,RLANS,MOTL,3,IW,CTFN)
      CTFN = RADDEG*CTFN
      CALL RAOPHA (CTFN,RAO(IW,IM),PHS(IW,IM),RADDEG,1)
70   CONTINUE

      RETURN
      END

C DECK OUTPUT
      SUBROUTINE OUTPUT(TIMADR,TRAQ,TRMS)

      COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
2 LRAOPR,ADRPR,ORGOPTN,GMNOM,KG,STATN(25),NSOFST(25),
2 NLEWF(25),HLFBTH(10,25),WTRLNE(10,25),BLEWF(25),TLEWF(25),
2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
2 ZPTFBD(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
2 STATNM,STATIS
      CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
      INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
2 FBNUMB,PTNUMB,ORGOPTN

```

```

REAL KG

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL
    INTEGER    SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL

COMMON /LOADS/ NLOADS,SWGHT(25),SMASS(25),XLDSTN(10),XLDXPT(25),
2 LSTATN(25)

COMMON /SMPSYS/ FIS,AS,SIS,SOS,SDS,HALOS,DEV,PRN,SMPPS,SMPIS,
2 SMPOS,SMPDS,SHPTYPS,SHIPS,VARS,CYCLS,TITLES,OPTION,LSIS,LSOS,
2 LSDS,LHALOS,LDEV,LPRN,LSMPPS,LSMPIS,LSMPOS,LSMPDS,LSHPTYPS,
2 LSHIPS,LTITLES
    CHARACTER*160 AS
    CHARACTER*80 FIS,SIS,SOS,SDS,TITLES
    CHARACTER*20 HALOS,DEV,PRN,SMPPS,SMPIS,SMPOS,SMPDS,SHPTYPS
    CHARACTER SHIPS*6,VARS*2,CYCLS*2
    INTEGER*2 OPTION

IF (ORGOPRN .EQ. 1) GO TO 10

IF (RAOPR .GT. 0) THEN
    AS = '(/4X,"CALLING RAOOUT")'
    WRITE (*,AS)
    WRITE (TEXFIL,AS)
    CALL RAOOUT
ENDIF

IF (NLOADS.GT.0 .AND. LRAOPR.GT.0) THEN
    AS = '(/4X,"CALLING LRAOUT")'
    WRITE (*,AS)
    WRITE (TEXFIL,AS)
    CALL LRAOUT
ENDIF

IF (OPTN.LT.6 .OR. RAOPR.NE.2) THEN
    AS = '(/4X,"CALLING RMSOUT")'
    WRITE (*,AS)
    WRITE (TEXFIL,AS)
    CALL RMSOUT
ENDIF

10 CONTINUE

RETURN
END

C DECK PADD
SUBROUTINE PADD(Z,IDZ,X,IDX,Y,IDX)

* subroutine to add two polynomials
* based on SSP, P.171

* INPUT
*   X(J),J=1,IDX - coefficients of polynomial with terms
*   X(J)*T**(J-1)

*   Y(J),J=1,IDX - coefficients of polynomial with terms
*   Y(J)*T**(J-1)

* OUTPUT
*   Z(J),J=1,IDX - coefficients of polynomial with terms
*   Z(J)*T**(J-1)

DIMENSION X(5),Y(5),Z(5)

NDIM=MAX0(IDX,IDX)
IF (NDIM .LE. 0) GO TO 90
30 CONTINUE

```

```

      DO 80 I=1,NDIM
      IF (I .GT. IDX) GO TO 60
      IF (I .GT. IDY) GO TO 70
      Z(I)=X(I)+Y(I)
      GO TO 80
  60  CONTINUE
      Z(I)=Y(I)
      GO TO 80
  70  CONTINUE
      Z(I)=X(I)
  80  CONTINUE
  90  IDZ=NDIM

      RETURN
      END

C DECK PDER
      SUBROUTINE PDER(Y,IDX,X,IDX)

* subroutine to calculate derivative of polynomial in polynomial form
* based on SSP, P.175

* INPUT
*   X(J),J=1,IDX - coefficients of polynomial with terms
*   X(J)*T**(J-1)

* OUTPUT
*   Y(J),J=1,IDX - coefficients of polynomial with terms
*   Y(J)*T**(J-1)

      DIMENSION X(4),Y(3)

      IDY=0
      IF (IDX .LE. 1) RETURN
      IDY=IDX-1
      EXPT=0.0
      DO 2 I=1,IDX
      EXPT=EXPT+1.0
      Y(I)=EXPT*X(I+1)
  2  CONTINUE

      RETURN
      END

C DECK PINT
      SUBROUTINE PINT(Y,IDX,X,IDX)

* subroutine to integrate polynomial
* based on SSP, P.176

* INPUT
*   X(J),J=1,IDX - coefficients of polynomial with terms
*   X(J)*T**(J-1)

* OUTPUT
*   Y(J),J=1,IDX - coefficients of polynomial with terms
*   Y(J)*T**(J-1)

      DIMENSION X(8),Y(9)

      IDY=IDX+1
      Y(1)=0.0
      IF (IDX .LE. 0) RETURN
      EXPT=1.0
      DO 3 I=2,IDX
      Y(I)=X(I-1)/EXPT
      EXPT=EXPT+1.0
  3  CONTINUE

      RETURN
      END

```

```

C DECK PMPY
SUBROUTINE PMPY(Z,IDX,X,IDX,Y,IDX)
* subroutine to multiply two polynomials
* based on SSP, P.172
* INPUT
*   X(J),J=1,IDX - coefficients of polynomial with terms
*   X(J)*T**(J-1)
*   Y(J),J=1,IDX - coefficients of polynomial with terms
*   Y(J)*T**(J-1)
* OUTPUT
*   Z(J),J=1,IDX - coefficients of polynomial with terms
*   Z(J)*T**(J-1)
DIMENSION X(5),Y(4),Z(8)
IF (IDX*IDX .LE. 0) GO TO 10
IDZ=IDX+IDX-1
DO 1 I=1,IDX
Z(I)=0.0
1 CONTINUE
DO 2 I=1,IDX
DO 3 J=1,IDX
K=I+J-1
Z(K)=Z(K)+X(I)*Y(J)
3 CONTINUE
2 CONTINUE
RETURN
10 CONTINUE
IDZ=0
RETURN
END

C DECK PRAO
SUBROUTINE PRAO (IM,NL,NU,MOTV,MOTL,XPT,YPT,ZPT,RAO1,PHS1,RAO2,
2 PHS2,NMOT,NPLANE,NOMEGA,RADDEG,IPHS,OMEGAE,GRAV)
* This routine obtains the longitudinal, lateral and vertical
* "MOTIONS AT A POINT" rao and phase angles for waves from both
* port and starboard headings.
* W.G.MEYERS, DTNSRDC, 100577
COMPLEX MOTV(NMOT,NOMEGA),MOTL(NMOT,NOMEGA),
2 SURGE,SWAY,HEAVE,ROLL,PITCH,YAW,TFN,LATACC
DIMENSION RAO1(NOMEGA),PHS1(NOMEGA),RAO2(NOMEGA),PHS2(NOMEGA),
2 OMEGAE(NOMEGA)
DO 30 I=NL,NU
SURGE = MOTV(1,I)
SWAY = MOTL(1,I)
HEAVE = MOTV(2,I)
ROLL = MOTL(2,I)
PITCH = MOTV(3,I)
YAW = MOTL(3,I)
DO 30 J=1,NPLANE
IF (J .EQ. 1) GO TO 10
SWAY = - SWAY
ROLL = - ROLL
YAW = - YAW
10 CONTINUE
IF (IM .EQ. 1) TFN = SURGE - YPT*YAW + ZPT*PITCH
IF (IM .EQ. 2) TFN = SWAY - ZPT*ROLL + XPT*YAW
IF (IM .EQ. 3) TFN = HEAVE - XPT*PITCH + YPT*ROLL
IF (IM .EQ. 15) THEN
IF (J .EQ. 1) CON = - OMEGAE(I) * OMEGAE(I) / GRAV
LATACC = CON * (SWAY - ZPT * ROLL + XPT * YAW)
TFN = LATACC + ROLL

```

```

ENDIF
IF (J .EQ. 1) CALL RAOPHA (TFN,RAO1(I),PHS1(I),RADDEG,IPHS)
IF (J .EQ. 2) CALL RAOPHA (TFN,RAO2(I),PHS2(I),RADDEG,IPHS)
30 CONTINUE

RETURN
END

C DECK PRELIM
SUBROUTINE PRELIM

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL
INTFGER   SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL

COMMON /PHYSICO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMTR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMTR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

* PHYSICO definitions
II = (0.0, 1.0)
PI = 3.1415927
TPI = 2*PI
PIOT = PI/2
DEGRAD = PI/180
RADDEG = 1./DEGRAD
FTMETR = .3048
VKMTR = 1.689*FTMETR

* IO definitions
SYSFIL = 1
POTFIL = 2
COFFIL = 3
LCOFIL = 4
ICARD = 5
TEXFIL = 6
IPRIN = 7
SCRFIL = 8
HPLFIL = 9
LRAFIL = 10
ORGFIL = 11
RAOFIL = 12
RMSFIL = 13
SEVFIL = 14
SPDFIL = 15
SPTFIL = 16
LACFIL = 17
LAEFIL = 18

RETURN
END

C DECK PSPLC
SUBROUTINE PSPLC (NOMEGA,OMEGA,OMEGAE,VK,HDNG,DEGRAD,GRAV,VKMTR,
2 DUM1,DUM2,RAO,S,R,NWEVN,WEVN,ARLC1,ARLC2,ARLC3,RLC)

DIMENSION OMEGA(30),OMEGAE(30),DUM1(30),DUM2(30),RAO(30),S(30),
2 R(30),WEVN(100),ARLC1(100),ARLC2(100),ARLC3(100),RLC(100)

CON = VKMTR*VK*COS(HDNG*DEGRAD)/GRAV
W1 = OMEGA(NOMEGA) + 1
IF(ABS(CON) .GT. 0.000001) W1=1./(2*CON)
IF(ABS(CON) .GT. 0.000001) W2=1./CON
NR1=0

```

```

NR2=0
NR3=0
DO 40 I=1,NOMEGA
XJACOB = 18
IF (OMEGA(I) .NE. W1) XJACOB = ABS(1./(1.-2*OMEGA(I)*CON))
IF (XJACOB .GT. 18.) XJACOB = 18.
R(I) = XJACOB*RAO(I)*S(I)
IF (ABS(CON) .GT. 0.000001) GO TO 10

* region 1
NR1 = NR1+1
GO TO 40
10 IF (OMEGA(I) .GT. W1) GO TO 20

* region 1
NR1 = NR1+1
GO TO 40
20 IF (OMEGA(I) .GT. W2) GO TO 30

* region 2
NR2 = NR2+1
GO TO 40

* region 3
30 NR3 = NR3+1
40 CONTINUE
DO 50 I=1,NWEVN
ARLC1(I) = 0.
ARLC2(I) = 0.
ARLC3(I) = 0.
50 CONTINUE

* interpolate longcrested response spectrum
IF (NR1 .LT. 2) GO TO 60
CALL INTRPL (NR1,OMEGAE,R,NWEVN,WEVN,ARLC1)
60 IF (NR2 .LT. 2) GO TO 80
M = NR1+1
N = NR1+NR2
L = N + 1
DO 70 I=M,N
L = L - 1
DUM1(I) = OMEGAE(L)
DUM2(I) = R(L)
70 CONTINUE
CALL INTRPL (NR2,DUM1(M),DUM2(M),NWEVN,WEVN,ARLC2)
80 IF (NR3 .LT. 2) GO TO 90
M = NR1+NR2+1
CALL INTRPL (NR3,OMEGAE(M),R(M),NWEVN,WEVN,ARLC3)
90 CONTINUE

* sum longcrested spectra in regions 1, 2 and 3
DO 100 I=1,NWEVN
RLC(I) = ARLC1(I) + ARLC2(I) + ARLC3(I)
100 CONTINUE

RETURN
END

C DECK PSPSC
SUBROUTINE PSPSC (NWEVN,WEVN,RLC,NBETA,B2,NLCH,IPH,ERLC,ERSC,
2 TOELC,TOESC,TPI)
DIMENSION WEVN(NWEVN),RLC(NWEVN,NBETA),B2(NLCH),ERLC(NWEVN),
2 ERSC(NWEVN)
KH = IPH + 5

```

```

IF (KH .GT. 24) KH = KH - 24
DO 20 I=1,NWEVN
K = IPH - 1
ERSC(1) = 0.
DO 10 L=1,NLCH
K = K + 1
IF (K .GT. 24) K = K - 24
10 ERSC(I) = ERSC(I) + B2(L)*RLC(I,K)
CONTINUE
ERLC(I) = RLC(I,KH)
20 CONTINUE
CALL TEPEAK {NWEVN,WEVN,ERLC,TOELC,TPI}
CALL TEPEAK {NWEVN,WEVN,ERSC,TOESC,TPI}

RETURN
END

C DECK PVAL
SUBROUTINE PVAL (VAL,ARG,POLY,IPOLY)

* subroutine to evaluate polynomial
* based on SSP, P.174

* INPUT
* POLY(J),J=1,IPOLY coefficients of polynomial with terms
* POLT(J)*T**(J-1)
* ARG - point at which polynomial is to be evaluated

* OUTPUT
* VAL - value of polynomial at t=arg

DIMENSION POLY(9)

VAL=0.0
J=IPOLY
1 CONTINUE
IF (J .LE. 0) GO TO 2
VAL=VAL*ARG+POLY(J)
J=J-1
GO TO 1
2 CONTINUE

RETURN
END

C DECK RAQOUT
SUBROUTINE RAQOUT

COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
2 LRAOPR,ADRPR,ORGOPTN,GMMOM,KG,STATN(25),NSOFST(25),
2 NLEWF(25),NLFBTH(10,25),WIELNE(10,25),BLEWF(25),TLEWF(25),
2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
2 ZPTFBD(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
2 STATNM,STATIS
CHARACTER*4 PTNAME(8,10),FRNAME(8,10),STATNM(5),FBTYPE(3,10)
INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
2 FBNUMB,PTNUMB,ORGOPTN
REAL KG

COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,WOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNU,FRNUM,VFS
INTEGER NVK,NMU,WOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMC(8),
REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(6)

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,WFRBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,ESTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB

```

```

INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BFAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWF,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL
    INTEGER     SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL

COMMON /PHYSICO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

COMMON /SMPSSYS/ FIS,AS,SIS,SOS,SDS,HALOS,DEV,PRN,SMPPS,SMPIS,
2 SMPDS,SHPTYPS,SHIPS,VARS,CYCLS,TITLES,OPTION,LSIS,LSOS,
2 LSDS,LHALOS,LDEV,LPRN,LSMPPS,LSMPIS,LSMPOS,LSMPDS,LSHPTYPS,
2 LSHIPS,LTITLES
CHARACTER*160 AS
CHARACTER*80 FIS,SIS,SOS,SDS,TITLES
CHARACTER*20 HALOS,DEV,PRN,SMPPS,SMPIS,SMPDS,SHPTYPS
CHARACTER SHIPS*6,VARS*2,CYCLS*2
INTEGER*2 OPTION

COMMON /STATE/ LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL
LOGICAL LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL

COMMON /WAVE/ MOTV(3,30),MOTL(3,30,8),HJV(3,30),HJL(3,30),H7(30)
DIMENSION OMEGAE(30),RAO(30,6),PHS(30,6),R(30),RLCALC(8),
2 IMODL(4)
CHARACTER*4 METER

DATA METER /'NETE'/
DATA EPS /0.0001/

DO 5 IS=1,NSIGWH
  SWH = SIGWH(IS)
  IF (PUNITS(1) .NE. METER) SWH = SWH*FTMETR
*   significant wave height ranges below are in meters
*   sea state 1
  IF (SWH .LE. 0.59) PER = 5.0
*   sea state 2
  IF (SWH.GT.0.59 .AND. SWH.LE.1.26) PER = 6.0
*   sea state 3
  IF (SWH.GT.1.26 .AND. SWH.LE.1.73) PER = 7.0
*   sea state 4
  IF (SWH.GT.1.73 .AND. SWH.LE.2.24) PER = 7.0
*   sea state 5
  IF (SWH.GT.2.24 .AND. SWH.LE.3.97) PER = 9.0
*   sea state 6
  IF (SWH.GT.3.97 .AND. SWH.LE.6.34) PER = 11.0

```

```

* sea state 7
  IF (SWH.GT.6.34 .AND. SWH.LE.12.29) PER = 15.0

* sea state 8
  IF (SWH.GT.12.29 .AND. SWH.LE.18.77) PER = 19.0

* greater than sea state 8
  IF (SWH .GT. 18.77) PER = 19.0
  IF (PER .LT. TMODAL(1)) PER = TMODAL(1)
  IF (PER .GT. TMODAL(NTMOD)) PER = TMODAL(NTMOD)
  IMODL(IS) = 1
  DO 3 IT=1,NTMOD
  IF (ABS(PER-TMODAL(IT)) .LT. EPS) IMODL(IS) = IT
3  CONTINUE
5  CONTINUE

FIS = SDS(1:LSDS)//'.ORG'
OPEN (UNIT=ORGFIL,FILE=FIS,FORM='UNFORMATTED',STATUS='UNKNOWN')

FIS = SDS(1:LSDS)//'.RAO'
OPEN (UNIT=RAOFIL,FILE=FIS,FORM='UNFORMATTED',STATUS='UNKNOWN')

READ (ORGFIL) TITLE,NVK,NMU,NOMEGA,OMEGA,NRANG,RLANG,VRT,LAT,
2 ADDRES,LPP,BEAM,DRAFT,DISPLM,GM,DELGM,KG,KROLL,LCB,GRAV,RHO,
2 VKDES,VKINC,DBLWL

WRITE (RAOFIL) TITLE,NOMEGA,OMEGA,NVK,NMU,NSIGWH,STATIS,
2 (STATNM(I),I=1,3),LPP,BEAM,DRAFT,DISPLM,GM,DELGM,KG,KROLL,
2 LCB,DBLWL,GRAV

DO 300 IV=1,NVK
DO 200 IH=1,NMU
READ (ORGFIL) VKNOTS,HEADNG,OMEGAE
IF (VRT) READ (ORGFIL) MOTV
IF (LAT) READ (ORGFIL) MOTL
IF (ADDRES) READ (ORGFIL) HJV,HJL,H7
HDNG = 180. - HEADNG
IF (IH.GT.1 .AND. IH.LT.NMU) GO TO 20

* following or head waves - lateral mode (sway,roll,yaw) RAOS
* are zero

DO 10 IM=2,6,2
DO 10 IW=1,NOMEGA
RAO(IW,IM) = 0.
PHS(IW,IM) = 0.
10  CONTINUE

* vertical mode (surge,heave,pitch) RAOS

20  DO 30 IM=1,5,2
CALL ORGRAO (RLANG,NRANG,O.,MOTV,MOTL,NOMEGA,IM,RAO,PHS,RADDEG)
30  CONTINUE
40  DO 100 IS=1,NSIGWH
KS = IMODL(IS)
IF (IH.EQ.1 .OR. IH.EQ.NMU) GO TO 80

* perform roll iteration for each sea state and for the specified
* statistic

DO 60 IA=1,NRANG
DO 50 IW=1,NOMEGA
R(IW) = CABS(MOTL(2,IW,IA))**2 * S(IW,KS)
50  CONTINUE
CALL ALGRNG (NOMEGA,OMEGA,R,AREA)
RLCALC(IA) = STATIS*SIGWH(IS)*SQRT(ABS(AREA))*RADDEG
60  CONTINUE
CALL RLITR (RLANG,NRANG.RLCALC,RLANS)

```

```

* lateral mode (sway,roll,yaw) raos

      DO 70 IM=2,6,2
      CALL ORGRAO (RLANG,NRANG,RLANS,MOTV,MOTL,NOMEGA,IM,RAO,PHS,
2 RADDEG)
70   CONTINUE
80   CONTINUE
      WRITE (IPRIN,1000) TITLE,VKNOTS,HDNG
1000 FORMAT (1H1,/,.28X,20A4,///,
2 45X,'RESPONSE AMPLITUDE OPERATORS (RAOS) ',
2 'AND PHASES',///,55X,'SHIP SPEED =',F5.0,', KNOTS',
2 /53X,'SHIP HEADING =',
2 F5.0,', DEGREES')
      IF (PUNITS(1) .EQ. METER) WRITE (IPRIN,1010) SIGWH(IS)
      IF (PUNITS(1) .NE. METER) WRITE (IPRIN,1020) SIGWH(IS)
1010 FORMAT (/42X,'SEA STATE: SIGNIFICANT WAVE HEIGHT =',
2 F5.2,', METERS')
1020 FORMAT (/42X,'SEA STATE: SIGNIFICANT WAVE HEIGHT =',F5.2,
2 ' FEET')
      WRITE (IPRIN,1030) TMODAL(KS),STATIS,(STATNM(I),I=1,3)
1030 FORMAT (54X,'MODAL PERIOD =',F4.0,' SECONDS',/,54X,
2 'STATISTIC =',F5.2,
2 (' ,3A4,'),///,2X,'OMEGA OMEGAE',9X,'SURGE',14X,'SWAY',15X,
2 'HEAVE',14X,
2 'ROLL',15X,'PITCH',14X,'YAW',/,18X,6('AMPL.     PHASE',4X)/)
      DO 90 IW=1,NOMEGA
      WRITE (IPRIN,2000) OMEGA(IW),OMEGAE(IW),(RAO(IW,IM),
2 PHS(IW,IM),IM=1,6)
2000 FORMAT (2F7.3,6(1PE12.4,OPF7.1))
90   CONTINUE
      WRITE (IPRIN,2100)
2100 FORMAT (//2X'NOTES: 1) VERTICAL RAO'S (SURGE,HEAVE,PITCH) ARE '
* 'LINEAR AND INDEPENDANT OF SEA STATE.'// 9X'2) LATERAL RAO'S '
* '(SWAY,ROLL,YAW) ARE NONLINEAR AND CHANGE WITH SEA STATE AND '
2 'STATISTIC.')
      IF (PUNITS(1) .EQ. METER) WRITE (IPRIN,2110)
      IF (PUNITS(1) .NE. METER) WRITE (IPRIN,2120)
2110 FORMAT (9X'3) AMPL. IS IN (PHYS.UNITS/METER)',2H**,,'2 AND PHASE '
2 'IS IN DEGREES.')
2120 FORMAT (9X'3) AMPL. IS IN (PHYS.UNITS/FOOT)',2H**,,'2 AND PHASE '
2 'IS IN DEGREES.')
      WRITE (IPRIN,2130)
2130 FORMAT (9X'4) HEADING CONVENTION: 0 DEG=HEAD, 90 DEG=STBD BEAM, '
2 '180 DEG=FOLLOWING SEAS.')
      WRITE (RAOFIL) VKNOTS,HDNG,SIGWH(IS),TMODAL(KS),OMEGAE,RAO,PHS
      IF (IH.EQ.1 .OR. IH.EQ.NMU) GO TO 200
100  CONTINUE
200  CONTINUE
300  CONTINUE

      CLOSE (UNIT=ORGFIL)
      CLOSE (UNIT=RAOFIL)

      RETURN
END

C DECK RAOPHA
      SUBROUTINE RAOPHA (TFN,RAO,PHS,RADDEG,IPHS)

* This routine obtains a response amplitude operator, RAO, and a
* phase angle, PHS. The response as a function of time can be
* written as-           RESP = SQRT(RAO) * COS(WE*T+PHS*DEGRAD)
*                               W.G.MEYERS, DTNSRDC, 100777

      COMPLEX TFM

      ARL = REAL (TFN)
      AIM = AIMAG (TFN)
      RAO = ARL*ARL + AIM*AIM
      IF (IPHS .EQ. i) PHS = ATAN2D (AIM,ARL,RADDEG)

```

```

      RETURN
      END

C DECK RAOPHS
SUBROUTINE RAOPHS (OMEGAE,RAO1,PHS1,RAO2,PHS2,IREC,IR,IV,IH,IPHS)

COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
2 LRAOPR,ADRPR,ORGOPTN,GNMOM,KG,STATN(25),NSOFST(25),
2 NLEWF(25),HLFBTH(10,25),WTRLNE(10,25),BLEWF(25),TLEWF(25),
2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
2 ZPTFBD(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
2 STATNM,STATIS
CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
2 FBNUMB,PTNUMB,ORGOPTN
REAL KG

COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEOFIL
INTEGER SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEOFIL

COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

COMMON /RESPN/ NRESP,IPOINT(182),IMOTN(182),ITYPE(182),
2 ILIN(182),ISYM(182)
LOGICAL ILIN,ISYM

COMMON /STATE/ LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL
LOGICAL LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL

COMPLEX MOTV(3,30),MOTL(3,30,8),HJV(3,30),HJL(3,30),H7(30)
COMPLEX SF3(25,30),SH3(25,30)
DIMENSION SA33(25,30),SB33(25,30)
DIMENSION OMEGAE(30),RAO1(30,8),PHS1(30,8),RAO2(30,8),PHS2(30,8)
LOGICAL LINEAR,SYMMET

DATA EPS /.001/

NMOT = 3
IP = IPOINT(IR)
IM = IMOTN(IR)
IT = ITYPE(IR)
LINEAR = ILIN(IR)
SYMMET = ISYM(IR)

```

```

N = 1
IF (.NOT. LINEAR) N = NRANG
HDNG = MU(IH,IV)*RADDEG
ARG = HDNG*DEGRAD
COSMU = COS(ARG)
SINMU = SIN(ARG)
READ (ORGFIL) VKNOTS,HEADNG,OMEGAE
IF (VRT) READ (ORGFIL) MOTV
IF (LAT) READ (ORGFIL) MOTL
IF (ADDRES) READ (ORGFIL) HJV,HJL,H7
IREC = 1
IF ((ABS(HEADNG).LT.EPS .OR. ABS(HEADNG-180.).LT.EPS) .AND.
2 (.NOT.LINEAR) .AND. SYMMET) IREC = 0
IF ((ABS(HEADNG).GT.EPS .AND. ABS(HEADNG-180.).GT.EPS) .AND.
2 (.NOT.SYMMET)) IREC = 2
IF (IREC.EQ.0) GO TO 100
M1 = 1
M2 = NOMEWA
DO 30 IA=1,N
  IF (IP.EQ.0 .AND. IM.LT.7) CALL ORAO (IM,M1,M2,MOTV,MOTL(1,1,IA),
2 RAO1(1,IA),PHS1(1,IA),NMOT,NOMEWA,RADDEG,IPHS)
  IF (IP.GT.0 AND. IM.LT.4) CALL PRAO (IM,M1,M2,MOTV,MOTL(1,1,IA),
2 XPT(IP),YPT(IP),ZPT(IP),RAO1(1,IA),PHS1(1,IA),RAO2(1,IA),
2 PHS2(1,IA),NMOT,IREC,NOMEWA,RADDEG,IPHS,OMEGAE,GRAV)
  IF (IP.GT.0 .AND. IM.EQ.15) CALL PRAO(IM,M1,M2,MOTV,MOTL(1,1,IA),
2 XPT(IP),YPT(IP),ZPT(IP),RAO1(1,IA),PHS1(1,IA),RAO2(1,IA),
2 PHS2(1,IA),NMOT,IREC,NOMEWA,RADDEG,IPHS,OMEGAE,GRAV)
  IF (IM.EQ. 8) CALL RELMOT (IM,M1,M2,MOTV,
2 MOTL(1,1,IA),FBDX(IP),FBDY(IP),RAO1(1,IA),PHS1(1,IA),RAO2(1,IA),
2 PHS2(1,IA),NMOT,IREC,NOMEWA,OMEGA,COSMU,SINMU,GRAV,RADDEG,IPHS)
  IF (IP.EQ.0 .AND. IM.EQ.9) CALL FNRDAO (IV,M1,M2,MOTL(1,1,IA),
2 RAO1(1,IA),PHS1(1,IA),NMOT,NOMEWA,OMEGAE,IPHS)
  IF (IT.GT.1) CALL VELACC (IM,IT,GRAV,M1,M2,OMEGAE,
2 RAO1(1,IA),PHS1(1,IA),RAO2(1,IA),PHS2(1,IA),NOMEWA,IREC,IPHS)
  IF (IM.EQ. 7) CALL ADRES (M1,M2,MOTV,MOTL(1,1,IA),HJV,HJL,H7,
2 RAO1(1,IA),PHS1(1,IA),RAO2(1,IA),PHS2(1,IA),OMEGA,NMOT,IREC,
2 NOMEWA,RADDEG,COSMU,RHO,IPHS)
30 CONTINUE
  IF (.NOT. (IP.GT.0 .AND. (IM.GE.10.AND.IM.LE.14))) GO TO 100
  DO 40 IW=1,NOMEWA
    READ (LCOFIL) (SF3(I,IW),SH3(I,IW),SA33(I,IW),SB33(I,IW),
2 I=1,NSTATN)
40 CONTINUE
  CALL LRAO (IM,M1,M2,MOTV,SF3,SH3,SA33,SB33,VFS(IV),COSMU,
2 OMEGA,OMEGAE,IP,RAO1,PHS1,NMOT,NOMEWA,IPHS)
100 CONTINUE

  RETURN
END

```

C DECK RDBASE

SUBROUTINE RDBASE

```

COMMON /CH3D/ ISIGMA,SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI,
2 IMMIN,IMMAX,IMDEL,LMIN,LMAX
REAL SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI(4)
INTEGER ISIGMA,IMMIN,IMMAX,IMDEL,LMIN,LMAX

COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
2 LRAOPR,ADRPR,ORGOPTN,GMNOM,KG,STATN(25),NSOFST(25),
2 NLEWF(25),HLFBTH(10,25),WTRLNE(10,26),BLEWF(25),TLEWF(25),
2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
2 ZPTFBD(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
2 STATNM,STATIS
CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
2 FBNUMB,PTNUMB,ORGOPTN
REAL KG

COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEWA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMDAL,NTHOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS

```

```

INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL
INTEGER SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL

COMMON /PHYSICO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

COMMON /RLDBK/ PSUR(25),BMK(25),DK(25),CAK(25),HQ,HSPAN,HMNCHD,
2 HAREA,HXCP,HYCP,HZCP,HGAMMA,HYHAT,HEAR,HLCS,RQ(2),RSPAN(2),
2 RMNCHD(2),RAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),
2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
2 SYCP(2),SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2),BQ(2),
2 BSPAN(2),BMNCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),
2 PQ(2,2),PSPAN(2,2),PMNCHD(2,2),PAREA(2,2),PXCP(2,2),PYCP(2,2),
2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),PLCS(2,2),
2 STADMP(10),SHPDMP(10,8),ENCON,WPHI,TPHI,WELM(4,9),SFELM(4,9,8),
2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
2 ENWM,ENSF(8,8),ENRE(8),ENPE(8),ENFE(8),ENHE(8),ENBE(8),
2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(8),ENSL(8),ENBL(8),
2 ENSHP(8,8),RELM(4,9),ITS(25),RD(25),EDDY(8,25),RGB(25)
REAL RDBLK(2692)
EQUIVALENCE (PSUR(1),RDBLK(1))

COMMON /SMPSYS/ FIS,AS,SIS,SOS,SDS,HALOS,DEV,PRN,SMPPS,SMPIS,
2 SMPDS,SHPTYPS,SHIPS,VARS,CYCLS,TITLES,OPTION,LSIS,LSOS,
2 LSDS,LHALOS,LDEV,LPRN,LSMPPS,LSMPIS,LSMPOS,LSMPDS,LSHPTYPS,
2 LSHIPS,LTITLES
CHARACTER*160 AS
CHARACTER*80 FIS,SIS,SOS,SDS,TITLES
CHARACTER*20 HALOS,DEV,PRN,SMPPS,SMPIS,SMPDS,SMPDS,SHPTYPS
CHARACTER SHIPS*6,VARS*2,CYCLS*2
INTEGER*2 OPTION

DO 30 K=1,NSTATN
IF (NOFSET(K) .LT. 2) GO TO 30
NNODES = NOFSET(K)
NSEGS = NNODES - 1
SGIRTH = 0.
DO 10 J=1,NSEGS
SGIRTH = SGIRTH + SQRT((Y(J+1,K)-Y(J,K))**2 + (Z(J+1,K)
2 -Z(J,K))**2)
10 CONTINUE
IF (K .EQ. 1) DS = 0.5*(X(2)-X(1))
IF (K .GT. 1 .AND. K .LT. NSTATN) DS = 0.5*(X(K+1)-X(K-1))
IF (K .EQ. NSTATN) DS = 0.5*(X(NSTATN)-X(NSTATN-1))
PSUR(K) = 2.*DS*SGIRTH

```

```

      BMK(K) = BMAX(NNODES,Y(1,K))
      DK(K) = 0.
      DO 20 J=1,NNODES
      IF (Z(J,K) .LT. DK(K)) DK(K) = Z(J,K)
20   CONTINUE
      CAK(K) = ASTAT(K)/(2*BMK(K)*ABS(DK(K)))
30   CONTINUE
      CALL WAVMAK
      CALL HLLIFT
      CALL RDLIFT
      CALL SBLIFT
      CALL SKLIFT
      CALL BKLIIFT
      CALL FNLLIFT
      CALL SKNFRC
      CALL RDEDDY
      CALL SBEDDY
      CALL FNEDDY
      CALL HLEDDY
      DO 50 IA=1,NRANG
      ENEMO = ENHE(IA) + ENRE(IA) + ENPE(IA) + ENFE(IA)
      DO 40 IV=1,NVK
      V = VFS(IV)
      ENEMV(IV,IA) = EDMKSP(WPHI,LPP,V,ENEMO)
40   CONTINUE
50   CONTINUE
      CALL BKEDDY
      DO 60 IA=1,NRANG
      DO 60 IV=1,NVK
      ENSHP(IV,IA) = ENWM + ENHL(IV) + ENRL(IV) + ENPL(IV) + ENSL(IV) +
2     ENBL(IV) + ENFL(IV) + ENSF(IV,IA) + ENEMV(IV,IA) + ENBE(IA)
60   CONTINUE

      IF (RLDMPR .GT. 0) CALL RDPRIN

      F1S = SDS(1:LSDS)//'.SCR'
      OPEN (UNIT=SCRFILE,FILE=F1S,FORM='UNFORMATTED',STATUS='UNKNOWN')
      WRITE (SCRFILE) RDBLK
      CLOSE (UNIT=SCRFILE)

      RETURN
      END

C DECK RDCOMP
SUBROUTINE RDCOMP (N,NDIM,A,IP)

*      matrix triangularization by gaussian elimination.

*      INPUT...
*          N = order of matrix.
*          NDIM = declared dimension of array A .
*          A = matrix to be triangulized.

*      OUTPUT...
*          A(I,J), I .LE. J = upper triangular factor, U .
*          A(I,J), I .GT. J = multipliers = lower triangular
*                           factor, I - L .
*          IP(K), K .LT. N = index of k-th pivot row.
*          IP(N) = (-1)**(number of interchanges) or 0 .

*      use "solve" to obtain solution of linear system.
*      DETERM( A ) = IP(N)*A(1,1)*A(2,2)*...*A(N,N).
*      IF IP(N) = 0, A is singular, SOLVE will divide by zero.

*      interchanges finished in u, only partialy in l .

      REAL A, T, ABS
      INTEGER N, NDIM, IP, K, KP1, M, I, J
      DIMENSION A(NDIM,NDIM)
      DIMENSION IP(NDIM)

      IP(N) = 1

```

```

DO 1700 K = 1, N
IF (K .EQ. N) GO TO 1600
KP1 = K + 1
M = K
DO 1100 I = KP1, N
IF (ABS(A(I,K)) .GT. ABS(A(M,K))) M = I
CONTINUE
IP(K) = M
IF (M .NE. K) IP(N) = -IP(N)
T = A(M,K)
A(M,K) = A(K,K)
A(K,K) = T
IF (T .EQ. 0.0) GO TO 1600
DO 1200 I = KP1, N
A(I,K) = -A(I,K)/T
CONTINUE
DO 1500 J = KP1, N
T = A(M,J)
A(M,J) = A(K,J)
A(K,J) = T
IF (T .EQ. 0.0) GO TO 1400
DO 1300 I = KP1, N
A(I,J) = A(I,J) + A(I,K)*T
CONTINUE
1400 CONTINUE
1500 CONTINUE
1600 CONTINUE
IF (A(K,K) .EQ. 0.0) IP(N) = 0
1700 CONTINUE
99999 CONTINUE

      RETURN
      END

C DECK RDEDDY
SUBROUTINE RDEDDY

COMMON /APPEND/ NBKSET,NBKSTN(2),BKIMAG(2),BKFS(2),BKAS(2),
2 BKWD(2),BKSTN(10,2),BKHB(10,2),BKLNTH,BKWDTH,
2 BKWL(10,2),BKA(10,2),NSKSET,SKIMAG(2),SKFLS(2),SKALS(2),
2 SKAUS(2),SKHB(2),SKFLWL(2),SKALWL(2),SKAUWL(2),NRDSET,RDIMAG(2),
2 RDRAS(2),RDRAS(2),RDRHB(2),RDRFWL(2),RDRAWL(2),RDTFS(2),RDTAS(2),
2 RDTHB(2),RDIFWL(2),RDTAWL(2),NSBSET,SBIMAG(2),SOBRFS(2),SOBRAS(2),
2 SOBRHB(2),SOBRFW(2),SOBRAW(2),SIBRFS(2),SIBRAS(2),SIBRHB(2),
2 SIBRFW(2),SIBRAW(2),SIBTFS(2),SIBTAS(2),SIBTB(2),SRTFWL(2),
2 SBTAWL(2),NFMSET,FNIMAG(2),FNRFS(2),FNRAS(2),
2 FNRHB(2),FNRFWL(2),FNRAWL(2),FNTFS(2),FNTAS(2),FNTHB(2),
2 FNCFWL(2),FNTAWL(2),NEXPRD,ENRDO(8),ENRDS(8)

COMMON /CH34/ ISIGMA,SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI,
2 IMMIN,IMMAX,IMDEL,LMIN,LMAX
REAL SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI(4)
INTEGER ISIGMA,IMMIN,IMMAX,IMDEL,LMIN,LMAX

COMMON /ENVIOR/ VK, NVK, MU, NMU, OMEGA, NOMEWA, SIGMA, NSIGMA, SIGWH,
1 NSIGWH,TMODAL,NTMOD, NRANG, RANG, RLANG, S, NNMMU, FRNUM, VFS
INTEGER NVK, NMU, NOMEWA, NSIGMA, NSIGWH, NTMOD, NRANG, NNMMU(8)
REAL VK(8), MU(37,8), OMEGA(30), SIGMA(10), SIGWH(4), TMODAL(8),
2 RANG(8), RLANG(8), S(30,8), FRNUM(8), VFS(8)

COMMON /PHYSCO/ II, TPI, PI, PIOT, DEGRAD, RADDEG, VKMETR, METRVK, GRAV,
2 RHO, GNU, RHOS, RHOF, GNUS, GNUF, FTMETR, PUNITS, REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI, PI, PIOT, DEGRAD, RADDEG, VKMETR, METRVK, GRAV, RHO, GNU, RHOS,
1 RHOF, GNUS, GNUF, FTMETR

COMMON /RLDBK/ PSUR(25), BMK(25), DK(25), CAK(25), HQ, HSPAN, HMNCHD,
2 HAREA, HXCP, HYCP, HZCP, HGAMMA, HYHAT, HEAR, HLCS, RQ(2), RSPAN(2),
2 RMNCHD(2), RAREA(2), RXCP(2), RYCP(2), RZCP(2), RGAMMA(2), RYHAT(2),
2 REAR(2), RLCS(2), SQ(2), SSFAN(2), SHNCHD(2), SAREA(2), SYCP(2),
2 SYCP(2), SZCP(2), SGAMMA(2), SYHAT(2), SEAR(2), SLCS(2), BQ(2),

```

```

2 BSPAN(2),BMINCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),
2 PQ(2,2),PSPAN(2,2),PMNCHD(2,2),PAREA(2,2),PXCP(2,2),PYCP(2,2),
2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),PLCS(2,2),
2 STADMP(10),SHPDMP(10,8),ENCON,WPHI,TPHI,WMELM(4,9),SFELM(4,9,8),
2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
2 ENWM,ENSF(8,8),ENRE(8),ENPE(8),ENFE(8),ENHE(8),ENBE(8),
2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(8),ENSL(8),ENBL(8),
2 ENSHP(8,8),RELM(4,9),ITS(25),RD(25),EDDY(8,25),RGB(25)
      REAL RDBLK(2692)
      EQUIVALENCE (PSUR(1),RDBLK(1))

      DO 20 IA=1,NRANG
      ENRE(IA) = 0
      DO 10 IS=1,NSIGMA
      SHPDMP(IS,IA) = 0
10    CONTINUE
20    CONTINUE
      IF (NRDSET .EQ. 0) GO TO 100
      DO 50 K=1,NRDSET
      YHAT = SQRT(RYCP(K)*RYCP(K) + RZCP(K)*RZCP(K))
      GAMMAE = RGAMMA(K) + 1.
      ALF = ATAN( ABS(((RYCP(K)/RZCP(K)) + TAN(GAMMAE*DEGRAD))/(1. -
2 (RYCP(K)/RZCP(K))*TAN(GAMMAE*DEGRAD))) )
      C = 0.0065 + (RLCS(K)*RLCS(K))/(0.9*PI*REAR(K))
      CON = RQ(K)*4./(3.*PI)*RHO*YHAT**3*RAREA(K)*C*SIN(ALF)
      DO 40 IA=1,NRANG
      DO 30 IS=1,NSIGMA
      SHPDMP(IS,IA) = SHPDMP(IS,IA) + (CON*SIGMA(IS)*RANG(IA)) *
2 SIGMA(IS)
30    CONTINUE
40    CONTINUE
50    CONTINUE
      DO 60 IA=1,NRANG
      CALL SPFIT (SIGMA,SHPDMP(1,IA),REELM(1,1,IA),NSIGMA)
      ENRE(IA) = ENCON*REVAL(REELM(1,ISIGMA,IA),WTS1)
60    CONTINUE
100   CONTINUE

      RETURN
      END

C DECK RDEVAL
SUBROUTINE RDEVAL (IV,OMEGA,OMEGAE,NRANG,TLG,EXCLG,TLGC,EXLGC,
2 T44T)

COMMON /APPEND/ NBKSET,NBKSTN(2),BKIMAG(2),BKFS(2),BKAS(2),
2 BKWD(2),BKSTN(10,2),BKHB(10,2),BKLNTH,BKWDTH,
2 BKWL(10,2),BKAN(10,2),NSKSET,SKIMAG(2),SKFLS(2),SKALS(2),
2 SKAUS(2),SKHB(2),SKFLWL(2),SKALWL(2),SKAUWL(2),NRDSET,RDIMAG(2),
2 RDRFS(2),RDRAS(2),RDRHB(2),RDRFWL(2),RDRFLW(2),RDRAWL(2),RDTFS(2),RDTAS(2),
2 RDTHB(2),RDTFWL(2),RDTAWL(2),NSBSET,SBIMAG(2),SOBRFS(2),SOBRAS(2),
2 SOBRHB(2),SOBRFW(2),SOBRAW(2),SIBRFS(2),SIBRAS(2),SIBRH(2),
2 SIBRFW(2),SIBRAW(2),SBTFS(2),SBTAS(2),SBTHB(2),SBTFWL(2),
2 SBTAWL(2),NFNSET,FNIMAG(2),FNRFS(2),FNRAS(2),
2 FNRRHB(2),FNRFWL(2),FNRAWL(2),FNTFS(2),FNTAS(2),FNTHB(2),
2 FNTFWL(2),FNTAWL(2),NEXPRD,ENRE(8),EERDS(8)

COMMON /CH3D/ ISIGMA,SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI,
2 IMMIN,IMMAX,IMDEL,LMIN,LMAX
REAL SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI(4)
INTEGER ISIGMA,IMMIN,IMMAX,IMDEL,LMIN,LMAX

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*40 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,

```

```

2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAWL,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPICTH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

COMMON /RLDBK/ PSUR(25),BMK(25),DK(25),CAK(25),HQ,HSPAN,HMNCHD,
2 HAREA,HXCP,HYCP,HZCP,HGAMMA,HYHAT,HEAR,HLCS,RQ(2),RSPAN(2),
2 RMNCHD(2),RAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),
2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
2 SYCP(2),SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2),BQ(2),
2 BSPAN(2),BMNCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),
2 PQ(2,2),PSPAN(2,2),PMNCHD(2,2),PAREA(2,2),PXCP(2,2),PYCP(2,2),
2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),PLCS(2,2),
2 STADMP(10),SHFDMP(10,8),ENCON,WPHI,TPHI,WMELM(4,9),SFELM(4,9,8),
2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
2 ENWM,ENSP(8,8),ENRE(8),ENPE(8),ENFE(8),ENHE(8),ENBE(8),
2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(8),ENSL(8),ENBL(8),
2 ENSHP(8,8),RELM(4,9),ITS(25),RD(25),EDDY(8,25),RGB(25)
REAL RDBLK(2692)
EQUIVALENCE (PSUR(1),RDBLK(1))

COMMON /FINCON/ IACTFN,IFCLCS,FGAIN(8),FK(3),FA(3),FB(3),
2 FCLCS(8,2)

COMPLEX TLG(3,3),EXCLG(3),TLGC(3,3),EXCLGC(3)
DIMENSION T44T(NRANG)

* TLG = LHS containing wavemaking damping only
* EXCLG = RHS
* TLGC and EXCLGC are LHS and RHS corrected by appendage and hull
damping
* T44 = WE*B44 (imaginary part of TLG array)
* B44 = roll damping moment
* N (roll decay coefficient) = B44*WE/(2*C44)
* C44 = DISPLACEMENT*GM
* B44 (TOTAL) = B44WM +
* B44HL + B44RL + B44PL + B44SL + B44BL + B44FL +
* B44SFV + B44EMV + B44BE
* Wavemaking damping - B44WM = AIMAG(TLG(2,2))/WE
* hull lift damping

CALL LSCOF (OMEGA,OMEGAE,0,HSPAN,HMNCHD,HAREA,HLCS,HGAMMA,
2 HXCP,HYCP,HZCP,TLG,EXCLG,TLGC,EXCLGC)

* rudder lift damping

IF (NRDSET .EQ. 0) GO TO 15
DO 10 K=1,NRDSET
CALL LSCOF (OMEGA,OMEGAE,2,RSPAN(K),RMNCHD(K),RAREA(K),RLCS(K),
2 RGAMMA(K),RXCP(K),RYCP(K),RZCP(K),TLGC,EXCLGC,TLGC,EXCLGC)
ANGLE = 180. - RGAMMA(K)
IF (RDIMAG(K) .GT. 1.)
2 CALL LSCOF (OMEGA,OMEGAE,2,RSPAN(K),RMNCHD(K),RAREA(K),RLCS(K),
2 ANGLE,RXCP(K),-RYCP(K),RZCP(K),TLGC,EXCLGC,TLGC,EXCLGC)
10 CONTINUE
15 CONTINUE

* propeller shaft bracket lift damping

IF (NSBSET .EQ. 0) GO TO 19
DO 16 K=1,NSBSET
DO 14 L=1,2
IF (L.EQ.2 .AND. SBTHB(K).EQ.0.) GO TO 14

```

```

ANGLE = PGAMMA(K,L)
IF (ANGLE .GT. 0.) ANGLE = ANGLE + 180.
CALL LSCOF (OMEGA,OMEGAE,2,PSPAN(K,L),PMNCHD(K,L),PAREA(K,L),
2 PLCS(K,L),ANGLE,PXCP(K,L),PYCP(K,L),PZCP(K,L),TLGC,EXCLGC,
2 TLGC,EXCLGC)
ANGLE = - PGAMMA(K,L)
IF (SBIMAG(K) .GT. 1. .AND. ANGLE .GT. 0.) ANGLE = ANGLE + 180.
IF (SBIMAG(K) .GT. 1.)
2 CALL LSCOF (OMEGA,OMEGAE,2,PSPAN(K,L),PMNCHD(K,L),PAREA(K,L),
2 PLCS(K,L),ANGLE,PXCP(K,L),-PYCP(K,L),PZCP(K,L),TLGC,
2 EXCLGC,TLGC,EXCLGC)
14 CONTINUE
16 CONTINUE
19 CONTINUE

* skeg lift damping
    IF (NSKSET .EQ. 0) GO TO 25
DO 20 K=1,NSKSET
    CALL LSCOF (OMEGA,OMEGAE,0,SSPAN(K),SMNCHD(K),SAREA(K),SLCS(K),
2 SGAMMA(K),SXCP(K),SYCP(K),SZCP(K),TLGC,EXCLGC,TLGC,EXCLGC)
    ANGLE = 180. - SGAMMA(K)
    IF (SKIMAG(K) .GT. 1.)
2 CALL LSCOF (OMEGA,OMEGAE,0,SSPAN(K),SMNCHD(K),SAREA(K),SLCS(K),
2 ANGLE,SXCP(K),-SYCP(K),SZCP(K),TLGC,EXCLGC,TLGC,EXCLGC)
20 CONTINUE
25 CONTINUE

* bilgekeel lift damping
    IF (NBKSET .EQ. 0) GO TO 35
DO 30 K=1,NBKSET
    CALL LSCOF (OMEGA,OMEGAE,1,BSPAN(K),BMNCHD(K),BAREA(K),BLCS(K),
2 BGAMMA(K),BXCP(K),BYCP(K),BZCP(K),TLGC,EXCLGC,TLGC,EXCLGC)
    ANGLE = 180. - BGAMMA(K)
    IF (BKIMAG(K) .GT. 1.)
2 CALL LSCOF (OMEGA,OMEGAE,1,BSPAN(K),BMNCHD(K),BAREA(K),BLCS(K),
2 ANGLE,BXCP(K),-BYCP(K),BZCP(K),TLGC,EXCLGC,TLGC,EXCLGC)
30 CONTINUE
35 CONTINUE

* fin lift damping
    JF (NFNSET .EQ. 0) GO TO 45
DO 40 K=1,NFNSET
    TEMP = FLCS(K)
    IF (IFCLCS .EQ. 1) TEMP = FCLCS(IV,K)
    CALL LSCOF (OMEGA,OMEGAE,2,FSPAN(K),FMNCHD(K),FAREA(K),TEMP,
2 FGAMMA(K),FXCP(K),FYCP(K),FZCP(K),TLGC,EXCLGC,TLGC,EXCLGC)
    ANGLE = 180. - FGAMMA(K)
    IF (FNIMAG(K) .GT. 1.)
2 CALL LSCOF (OMEGA,OMEGAE,2,FSPAN(K),FMNCHD(K),FAREA(K),TEMP,
2 ANGLE,FXCP(K),-FYCP(K),FZCP(K),TLGC,EXCLGC,TLGC,EXCLGC)
40 CONTINUE
45 CONTINUE
    DO 100 IA=1,NRANG

* skin friction damping at speed
    T44SF = REVAL(SFELM(1,ISIGMA,IA),WTSI)
    T44SFV = SKFRSP (OMEGAE,LPP,V,T44SF)

* rudder eddy damping
    T44RE = 0
    IF (NRDSET.GT.0) T44RE = REVAL(REELM(1,ISIGMA,IA),WTSI)

* propeller shaft bracket eddy damping
    T44PE = 0
    IF (NSBSET.GT.0) T44PE = REVAL(PEELM(1,ISIGMA,IA),WTSI)

```

```

*   fin eddy damping
    T44FE = 0
    IF (NFnSET .GT. 0) T44FE = REVAL(FEELM(1,ISIGMA,IA),WTSI)

*   hull eddy damping
    T44HE = REVAL(HEELM(1,ISIGMA,IA),WTSI)

*   eddymaking at speed
    T44EM = T44HE + T44RE + T44PE + T44FE
    T44EMV = EDMKSP (OMEGA,E,LPP,V,T44EM)

*   bilgekeel eddy damping
    T44BE = 0.
    IF (NBKSET .EQ. 0) GO TO 70
    T44BE = REVAL(BEELM(1,ISIGMA,IA),WTSI)
70  CONTINUE
    T44T(IA) = T44SFV + T44EMV + T44BE
100 CONTINUE

    RETURN
    END

C DECK RDLIFT
SUBROUTINE RDLIFT

COMMON /APPEND/ NBKSET,NBKSTN(2),BKIMAG(2),BKFS(2),BKAS(2),
2 BKWD(2),BKSTN(10,2),BKHB(10,2),BKLNTB,BKWDTH,
2 BKWL(10,2),BKAN(10,2),NSKSET,SKIMAG(2),SKFLS(2),SKALS(2),
2 SKAUS(2),SKHB(2),SKFLWL(2),SKALWL(2),SKAUWL(2),NRDSET,RDIMAG(2),
2 RDRFS(2),RDRAS(2),RDRHB(2),RDRFWL(2),RDRAWL(2),RDTFS(2),RDTAS(2),
2 RDTHB(2),RDTFWL(2),RDTAWL(2),NSESET,SBIMAG(2),SOBRFS(2),SOBRAS(2),
2 SOBRHB(2),SOBRFW(2),SOBRAW(2),SIBRFS(2),SIBRAS(2),SIBRHB(2),
2 SIBRFW(2),SIBRAW(2),SETFS(2),SBTAS(2),SBTHB(2),SBTFWL(2),
2 SBTAWL(2),NFnSET,FNIMAG(2),FNRFS(2),FNKAS(2),
2 FNRHB(2),FNRFWL(2),FNRRAWL(2),FNTFS(2),FNTAS(2),FNTHB(2),
2 FNTFWL(2),FNTAWL(2),NEXPRD,ENRDO(8),ENRDS(8)

COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8),
REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),TITLE(20),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /PHYSICO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

COMMON /RLDBK/ PSUR(25),BMK(25),DK(25),CAK(25),HQ,HSPAN,HMNCHD,
2 HAREA,HXCP,HYCP,HZCP,HGAMMA,HYHAT,HEAR,HLCS,RQ(2),RSPAN(2),
2 RMNCHD(2),RAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),
2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
2 SYCP(2),SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2),BQ(2),
2 ESPAN(2),EMNCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),

```

```

2 BYHAT(2), BEAR(2), BLCS(2), FQ(2), FSPAN(2), FMNCHD(2), FAREA(2),
2 FXCP(2), FYCP(2), FZCP(2), FGAMMA(2), FYHAT(2), FFEAR(2), FLCS(2),
2 PQ(2,2), PSPAN(2,2), PMNCHD(2,2), PAREA(2,2), PXCP(2,2), PYCP(2,2),
2 PZCP(2,2), PGAMMA(2,2), PYHAT(2,2), PEAR(2,2), PLCS(2,2),
2 STADMP(10), SHPDMP(10,8), ENCON WPH1, TPH1, WMELM(4,9), SFELM(4,9,6),
2 REELM(4,9,8), PEELM(4,9,8), FEELM(4,9,8), HEELM(4,9,8), BEELM(4,9,8),
2 ENWM, ENSF(8,8), ENRE(8), ENFE(8), ENHE(8), ENBE(8),
2 ENEMV(8,8), ENRL(8), ENPL(8), ENFL(8), ENHL(8), ENSL(8), ENBL(8),
2 ENSHP(8,8), RELM(4,9), ITS(25), RD(25), EDDY(8,25), RGB(25)
REAL RDBLK(2692)
EQUIVALENCE (PSUR(1), RDBLK(1))

```

REAL LCS, MCHORD

```

IF (NRDSET .EQ. 0) GO TO 20
EN = 0
STASPC = LPP/20
DO 10 K=1, NRDSET
XRTF = LCB - RDRFS(K)*STASPC
XRTA = LCB - RDRAS(K)*STASPC
XTPF = LCB - RDTFS(K)*STASPC
XTPA = LCB - RDTAS(K)*STASPC
YRT = RDRHB(K)
YTP = RDTHB(K)
ZRT = (RDRFWL(K) + RDRAWL(K))/2 - (DBLWL+VCG)
ZTP = (RDTFWL(K) + RDTAWL(K))/2 - (DBLWL+VCG)
SPAN = SQRT((ZRT-ZTP)**2 + (YTP-YRT)**2)
Q = RDIMAG(K)
MCHORD = 0.5*((XTPF-XTPA) + (XRTF-XRTA))
CR = XRTF - XRTA
CT = XTPF - XTPA
XRQC = XRTF - 0.25*CR
XTQC = XTPF - 0.25*CT
DX = XRQC - XTQC
H = SQRT(DX*DX + SPAN*SPAN)
COSLAM = SPAN/H
SECLAM2 = 1./(COSLAM*COSLAM)

```

* LAM = ACOS(SPAN/H) = quarter chord sweep angle in radians

* area

AREA = SPAN*MCHORD

* center of pressure

```

ZP = 0.5*(ZRT + ZTP)
YP = 0.5*(YRT + YTP)
XO = 0.5*(XRTF + XTPF)
XCP = XO - 0.25*MCHORD
YCP = YP
ZCP = ZP

```

* moment arm

```

ARG = (ZRT-ZTP) / SPAN
GAMMA = - 90
IF (ARG .LT. 1) GAMMA = - ASIN(ARG)*RADDEG
GAM = GAMMA*DEGRAD
YHAT = YCP*COS(GAM) + ZCP*SIN(GAM)

```

* effective aspect ratio

EAR = 2*SPAN/MCHORD

* lift curve slope

```

LCS = 1.8*PI*EAR/(COSLAM*SQRT((EAR*SECLAM2)**2 + 4) + 1.8)
RQ(K) = Q
RSPAN(K) = SPAN
RMNCHD(K) = MCHORD
RAREA(K) = AREA

```

```

RXCP(K) = XCF
RYCP(K) = YCP
RZCP(K) = ZCP
RGAMMA(K) = GAMMA
RYHAT(K) = YHAT
REAR(K) = EAR
RLCS(K) = LCS
EN = EN + Q*(RHO/2)*AREA*LCS*YHAT*YHAT*WPHI*ENCON
10 CONTINUE
20 CONTINUE
DO 30 IV=1,NVK
ENRL(IV) = 0.
IF (NRDSET .GT. 0) ENRL(IV) = EN*VFS(IV)
30 CONTINUE

RETURN
END

C DECK RDPELM
SUBROUTINE RDPELM

* reads spline element data for 2-d potentials and forces
* W.R.MCCREIGHT DTNSRDC JULY,1977

COMMON /CH3D/ ISIGMA,SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI,
2 IMMIN,IMMAX,IMDEL,LMIN,LMAX
REAL SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI(4)
INTEGER ISIGMA,IMMIN,IMMAX,IMDEL,LMIN,LMAX

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CWIDTH,CWIDTHAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CWIDTH,CWIDTHAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /INDEX/ PFIDX,LPFIDX,RMIDX,LRMIDX,SVIDX,LSVIDX
INTEGER LPFIDX,LRMIDX,LSVIDX
REAL PFIDX(235),RMIDX(183),SVIDX(3)

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL
INTEGER SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL

COMMON /PELEM/ PELEM
COMPLEX PELEM(4,1000)

COMMON /STATE/ LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL
LOGICAL LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL

DIMENSION DATA(320)
INTEGER PSTORE

MODSK=0
PSTORE=1
DO 1 ISTATN=1,NSTATN
INDEX=(ISIGMA-1)*NSTATN+ISTATN
NNODE=NOFSET(ISTATN)
IF(NNODE .LT. 2) GO TO 1
NDATP=0
IF (VRT) NDATP=16*NNODE
IF (LAT) NDATP=NDATP+10*NNODE

```

```

*      change for VAX/VMS version
*  CDC    CALL READMS(POTFIL,DATA,NDATP,INDEX)

      READ (POTFIL,REC=INDEX) DATA

      NEXT=1
      DO 2 J=1,NNODE
      DO 3 LMODE=IMMIN,IMMAX,IMDEL
      DO 4 I=1,4
      PELEM(I,PSTORE)=CMPLX(DATA(NEXT),DATA(NEXT+1))
      NEXT=NEXT+2
  4  CONTINUE
      PSTORE=PSTORE+1
  3  CONTINUE
  2  CONTINUE
      NODSK=NODSK+NNODE
      IF ((NEXT-1).NE.NDATP) WRITE (IPRIN,601) ISIGMA,ISTATN
  601 FORMAT ('/ WARNING - IN RDPELM FOR ISIGMA = ',IS,
      + ' AND ISTATN = ',IS,
      + ' NO. OF DATA ELEMENTS READ IS NOT EQUAL TO NO. OF DATA',
      + ' ELEMENTS UNPACKED'//)
  1  CONTINUE

      RETURN
      END

C DECK RDPRIN
      SUBROUTINE RDPRIN

      COMMON /APPEND/ NBKSET,NBKSTN(2),BKIMAG(2),BKFS(2),BKAS(2),
      2 BKWD(2),BKSTN(10,2),BKHB(10,2),BKLNTB,BKWDTH
      2 BKWL(10,2),BKAN(10,2),NSKSET,SKIMAG(2),SKFLS(2),SKALS(2),
      2 SKAUS(2),SKHB(2),SKFLWL(2),SKALWL(2),SKAUWL(2),NRDSET,RDIMAG(2),
      2 RDRFS(2),RDRAS(2),RDRHB(2),RDRFWL(2),RDRAWL(2),RDTFS(2),RDTAS(2),
      2 RDTHB(2),RDTFWL(2),RDTAWL(2),NSBSET,SBIMAG(2),SOBRFS(2),SOBRAS(2),
      2 SOBRHB(2),SOBRFW(2),SOBRAV(2),SIPRFS(2),SIBRAS(2),SIFRHP(2),
      2 SIBRFW(2),SIBRAW(2),SBTFS(2),SBTAS(2),SBTHB(2),SBTFWL(2),
      2 SBTAWL(2),NFNSET,FNIMAG(2),FNRF(2),FNRS(2),
      2 FNHB(2),FNRFWL(2),FNRAWL(2),FNTFS(2),FNTAS(2),FNTHB(2),
      2 FNTFWL(2),FNTAWL(2),NEXPRD,ENRDO(8),ENRDS(6)

      COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
      1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
      INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8),
      REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
      2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(6)

      COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF
      1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
      2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
      2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
      2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
      INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
      CHARACTER*4 TITLE(20)
      REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
      2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
      2 FBDZ(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
      4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
      5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

      COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
      2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
      2 SPTFIL,LACFIL,LAEFIL
      INTEGER     SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
      2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
      2 SPTFIL,LACFIL,LAEFIL

      COMMON /PHYSICO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
      2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
      COMPLEX II
      CHARACTER*4 PUNITS(2)
      REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,

```

```

1 RHOF,GNUS,GNUF,FTMETR

COMMON /RLDBK/ PSUR(25),BMK(25),DK(25),CAK(25),HQ,HSPAN,HMNCHD,
2 HAREA,HXCP,HYCP,HZCP,HGAMMA,HYHAT,HEAR,HLCS,RQ(2),RSPAN(2),
2 RMNCHD(2),RAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),
2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
2 SYCP(2),SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2),BQ(2),
2 BSPAN(2),BMNCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),
2 PQ(2,2),PSPAN(2,2),PMNCHD(2,2),PAREA(2,2),PXCP(2,2),PYCP(2,2),
2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),PLCS(2,2),
2 STADMP(10),SHPDMP(10,8),ENCON,WPHI,TPHI,WMELM(4,9),SFELM(4,9,8),
2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
2 ENWM,ENSF(8,8),ENRE(8),ENPE(8),ENFE(8),ENHE(8),ENBE(8),
2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(8),ENSL(8),ENBL(8),
2 ENSHP(8,8),RELM(4,9),ITS(25),RD(25),EDDY(8,25),RGB(25)
REAL RDBLK(2692)
EQUIVALENCE (PSUR(1),RDBLK(1))

DIMENSION TEMP(8)
CHARACTER*4 METER

DATA METER//'Mete'/
WRITE (IPRIN,1000) TITLE

* ship particulars
DISPLT = MASS*.001
IF (PUNITS(1) .NE. METER) DISPLT = MASS*GRAV/2240.
WRITE (IPRIN,1010) LPP,GM,BEAM,KROLL,DRAFT,WPHI,DISPLT,TPHI

* hull and appendage particulars
WRITE (IPRIN,1020)
WRITE (IPRIN,1030) HQ,HGAMMA,HMNCHD,HSPAN,HAREA,HXCP,HYCP,HZCP,
2 HYHAT,HEAR,HLCS
IF (NSKSET .GT. 0) WRITE (IPRIN,1040) (SQ(I),SGAMMA(I),SMNCHD(I),
2 SSPAN(I),SAREA(I),SXCP(I),SYCP(I),SZCP(I),SYHAT(I),SEAR(I),
2 SLCS(I),I=1,NSKSET)
IF (NRDSET .GT. 0) WRITE (IPRIN,1050) (RQ(I),RGAMMA(I),RMNCHD(I),
2 RSPAN(I),RAREA(I),RXCP(I),RYCP(I),RZCP(I),RYHAT(I),REAR(I),
2 RLCS(I),I=1,NRDSET)
IF (NSBSET .EQ. 0) GO TO 4
DO 2 K=1,NSBSET
M = 2
IF (SBTHB(K) .EQ. 0.) M = 1
WRITE (IPRIN,1055) (PQ(K,L),PGAMMA(K,L),PMNCHD(K,L),PSPAN(K,L),
2 PAREA(K,L),PXCP(K,L),PYCP(K,L),PZCP(K,L),PYHAT(K,L),PEAR(K,L),
2 PLCS(K,L),L=1,M)
2 CONTINUE
4 CONTINUE
IF (NBKSET .GT. 0) WRITE (IPRIN,1060) (BQ(I),BGAMMA(I),BMNCHD(I),
2 BSPAN(I),BAREA(I),BXCP(I),BYCP(I),BZCP(I),BYHAT(I),BEAR(I),
2 BLCS(I),I=1,NBKSET)
IF (NFSSET .GT. 0) WRITE (IPRIN,1070) (FQ(I),FGAMMA(I),FMNCHD(I),
2 FSPAN(I),FAREA(I),FXCP(I),FYCP(I),FZCP(I),FYHAT(I),FEAR(I),
2 FLCS(I),I=1,NFSSET)

* total roll decay coefficient, N
WRITE (IPRIN,1075)
WRITE (IPRIN,1080) (RLANG(IA),IA=1,NRANG)
DO 10 IV=1,NVK
WRITE (IPRIN,1090) VK(IV),(ENSHP(IV,IA),IA=1,NRANG)
10 CONTINUE
WRITE (IPRIN,1000) TITLE

* roll decay coefficients grouped by hull and appendages
WRITE (IPRIN,1100)

```

```

* hull and skeg
      WRITE (IPRIN,1110)
      WRITE (IPRIN,1080) (RLANG(IA),IA=1,NRANG)
      DO 30 IV=1,NVK
      DO 20 IA=1,NRANG
      ENHEV = EDMKSP(WPHI,LPP,VFS(IV),ENHE(IA))
      TEMP(IA) = ENWM + ENSF(IV,IA) + ENHEV + ENHL(IV) + ENSL(IV)
20   CONTINUE
      WRITE (IPRIN,1090) VK(IV),(TEMP(IA),IA=1,NRANG)
30   CONTINUE

* rudder
      IF (NRDSET .EQ. 0) GO TO 60
      WRITE (IPRIN,1120)
      WRITE (IPRIN,1080) (RLANG(IA),IA=1,NRANG)
      DO 50 IV=1,NVK
      DO 40 IA=1,NRANG
      ENREV = EDMKSP(WPHI,LPP,VFS(IV),ENRE(IA))
      TEMP(IA) = ENREV + ENRL(IV)
40   CONTINUE
      WRITE (IPRIN,1090) VK(IV),(TEMP(IA),IA=1,NRANG)
50   CONTINUE
60   CONTINUE

* propeller shaft brackets
      IF (NSBSET .EQ. 0) GO TO 66
      WRITE (IPRIN,1125)
      WRITE (IPRIN,1080) (RLANG(IA),IA=1,NRANG)
      DO 64 IV=1,NVK
      DO 62 IA=1,NRANG
      ENPEV = EDMKSP(WPHI,LPP,VFS(IV),ENPE(IA))
      TEMP(IA) = ENPEV + ENPL(IV)
62   CONTINUE
      WRITE (IPRIN,1090) VK(IV),(TEMP(IA),IA=1,NRANG)
64   CONTINUE
66   CONTINUE

* bilgekeel
      IF (NBKSET .EQ. 0) GO TO 90
      WRITE (IPRIN,1130)
      WRITE (IPRIN,1080) (RLANG(IA),IA=1,NRANG)
      DO 80 IV=1,NVK
      DO 70 IA=1,NRANG
      TEMP(IA) = ENBE(IA) + ENBL(IV)
70   CONTINUE
      WRITE (IPRIN,1090) VK(IV),(TEMP(IA),IA=1,NRANG)
80   CONTINUE
90   CONTINUE

* fin
      IF (NFMSET .EQ. 0) GO TO 120
      WRITE (IPRIN,1140)
      WRITE (IPRIN,1080) (RLANG(IA),IA=1,NRANG)
      DO 110 IV=1,NVK
      DO 100 IA=1,NRANG
      ENFEV = EDMKSP(WPHI,LPP,VFS(IV),ENFE(IA))
      TEMP(IA) = ENFEV + ENFL(IV)
100  CONTINUE
      WRITE (IPRIN,1090) VK(IV),(TEMP(IA),IA=1,NRANG)
110  CON 110E
120  CONTINUE
      WRITE (IPRIN,1000) TITLE

* roll decay coefficients grouped by damping mechanism
      WRITE (IPRIN,1150)

```

```

*      wavemaking
      WRITE (IPRIN,1160) ENWM

*      skin friction
      WRITE (IPRIN,1170)
      WRITE (IPRIN,1080) (RLANG(IA),IA=1,NRANG)
      DO 130 IV=1,NVK
      WRITE (IPRIN,1090) VK(IV),(ENSF(IV,IA),IA=1,NRANG)
130    CONTINUE

*      eddymaking (excluding bilgekeel)
      WRITE (IPRIN,1180)
      WRITE (IPRIN,1080) (RLANG(IA),IA=1,NRANG)
      DO 140 IV=1,NVK
      WRITE (IPRIN,1090) VK(IV),(ENEMV(IV,IA),IA=1,NRANG)
140    CONTINUE
      IF (NBKSET .EQ. 0) GO TO 145

*      bilgekeel eddymaking
      WRITE (IPRIN,1190)
      WRITE (IPRIN,1200) (RLANG(IA),IA=1,NRANG)
      WRITE (IPRIN,1205) (ENBE(IA),IA=1,NRANG)
145    CONTINUE

*      lift N values
      WRITE (IPRIN,1210)
      DO 150 IV=1,NVK
      ENLFT = ENRL(IV) + ENSL(IV) + ENPL(IV) + ENBL(IV)
      2 + ENFL(IV)
      WRITE (IPRIN,1090) VK(IV),ENHL(IV),ENSL(IV),ENRL(IV),ENPL(IV),
      2 ENBL(IV),ENFL(IV),ENLFT
150    CONTINUE
1000   FORMAT (1H1,9X,20A4//)
1010   FORMAT (40X,16HSHIP PARTICULARS//32X,8HLPP  =,F8.2,8X,
      2 7HGM  =,F6.2/32X,8HBEAM  =,F8.2,8X,7HKROLL =,F6.2,1HB/32X,
      2 8HDRAFT =,F8.2,8X,7HWPHI =,F6.3/32X,8HDISPLM =,F8.0,8X,
      2 7HTPHI =,F6.2)
1020   FORMAT (///35X,30HHULL AND APPENDAGE PARTICULARS//17X,
      2 34HQ  GAMMA MCHORD MSPAN AREA,5X,3HXCP,5X,3HYCP,5X,3HZCP,
      2 4X,4HYHAT,5X,3HEAR,5X,3HLC5/)
1030   FORMAT (' HULL ',F8.0,F8.1,7F8.2,2F8.3)
1040   FORMAT (' SKEG  ',F8.0,F8.1,7F8.2,2F8.3)
1050   FORMAT (' RUDDER ',F8.0,F8.1,7F8.2,2F8.3)
1055   FORMAT (' BRACKET ',F8.0,F8.1,7F8.2,2F8.3)
1060   FORMAT (' BILGEKEEL',F8.0,F8.1,7F8.2,2F8.3)
1070   FORMAT (' FIN   ',F8.0,F8.1,7F8.2,2F8.3)
1075   FORMAT (///35X,30HSHIP ROLL DECAY COEFFICIENT, N)
1080   FORMAT (//13X,10HSHIP SPEED,17X,20HMEAN ROLL ANGLE (SA)/15X,
      2 7H(KNOTS),23X,9H(DEGREES)/23X,8F7.1/)
1090   FORMAT (16X,F4.0,3X,8F7.3)
1100   FORMAT (33X,34HROLL DECAY COEFFICIENTS GROUPED BY/
      2 40X,19HHULL AND APPENDAGES)
1110   FORMAT (//40X,19HBARE HULL PLUS SKEG/25X,
      2 49H(WAVEMAKING, SKIN FRICTION, EDDYMAKING, AND LIFT))
1120   FORMAT (//45X,6HRUDDER/37X,22H(EDDYMAKING PLUS LIFT))
1125   FORMAT (//36X,24HPROPELLER SHAFT BRACKETS/37X,
      2 22H(EDDYMAKING PLUS LIFT))
1130   FORMAT (//43X,9HBILGEKEEL/37X,22H(EDDYMAKING PLUS LIFT))
1140   FORMAT (//47X,3HFIN/37X,22H(EDDYMAKING PLUS LIFT))
1150   FORMAT (33X,34HROLL DECAY COEFFICIENTS GROUPED BY/43X,
      2 13HDAMPING TYPES)
1160   FORMAT (//40X,12HWAVEMAKING =,F7.3)
1170   FORMAT (//42X,13HSKIN FRICTION)
1180   FORMAT (//32X,33HEDDYMAKING (EXCLUDING BILGEKEELS)/30X,
      2 38H(HULL, SKEG, RUDDER, BRACKET, AND FIN))
1190   FORMAT (//38X,20HBILGEKEEL EDDYMAKING)

```

```

1200 FORMAT (/40X,20HMEAN ROLL ANGLE (SA)/45X,9H(DEGREES)/23X,8F7.1/)
1205 FORMAT (23X,8F7.3)
1210 FORMAT (//47X,4HLIFT//13X,10HSHIP SPEED,
2 49H HULL SKEG RUDDER BRACKT BILGKL FIN TOTAL/15X,
2 7H(KNOTS)/)

      RETURN
      END

C DECK RDSMPSYS - Read SMPSYS.TEX FILE
SUBROUTINE RDSMPSYS

      COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEOFIL
      INTEGER   SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEOFIL

      COMMON /SMPSYS/ FIS,AS,SIS,SOS,SDS,HALOS,DEV,PRN,SMPPS,SMPIS,
2 SMPOS,SMPDS,SHPTYPS,SHIPS,VARS,CYCLS,TITLES,OPTION,LSIS,LSOS,
2 LSDS,LHALOS,LDEV,LPRN,LSMPPS,LSMPIS,LSMPOS,LSMPDS,LSHPTYPS,
2 LSHIPS,LTITLES
      CHARACTER*160 AS
      CHARACTER*80 FIS,SIS,SOS,SDS,TITLES
      CHARACTER*20 HALOS,DEV,PRN,SMPPS,SMPIS,SMPOS,SMPDS,SHPTYPS
      CHARACTER SHIPS*6,VARS*2,CYCLS*2
      INTEGER*2 OPTION

10   FORMAT (A)

      FIS = 'SMPSYS.TEX'
      OPEN (SYSFIL,FILE=FIS,STATUS='OLD')

      READ (SYSFIL,10) AS           ! HALO program path
      CALL SLENTH (AS,LHALOS)
      HALOS = AS(19:LHALOS)
      LHALOS = LHALOS - 18

      READ (SYSFIL,10) AS           ! HALO graphics screen driver
      CALL SLENTH (AS,LDEV)
      DEV = AS(29:LDEV)
      LDEV = LDEV - 28

      READ (SYSFIL,10) AS           ! HALO printer driver
      CALL SLENTH (AS,LPRN)
      PRN = AS(21:LPRN)
      LPRN = LPRN - 20

      READ (SYSFIL,10) AS           ! SMP program path
      CALL SLENTH (AS,LSMPPS)
      SMPPS = AS(18:LSMPPS)
      LSMPPS = LSMPPS - 17

      READ (SYSFIL,10) AS           ! SMP input path
      CALL SLENTH (AS,LSMPIS)
      SMPIS = AS(16:LSMPIS)
      LSMPIS = LSMPIS - 16

      READ (SYSFIL,10) AS           ! SMP output path
      CALL SLENTH (AS,LSMPOS)
      SMPOS = AS(17:LSMPOS)
      LSMPOS = LSMPOS - 16

      READ (SYSFIL,10) AS           ! SMP data path
      CALL SLENTH (AS,LSMPDS)
      SMPDS = AS(15:LSMPDS)
      LSMPDS = LSMPDS - 14

      READ (SYSFIL,10) AS           ! Ship type
      CALL SLENTH (AS,LSHPTYPS)

```

```

SHPTYPS = AS(11:LSHPTYPS)
LSHPTYPS = LSHPTYPS - 10

READ (SYSFIL,10) AS          ! Current ship
CALL SLENTH (AS,LSHIPS)
SHIPS = AS(14:LSHIPS)
LSHIPS = LSHIPS - 13
IF (LSHIPS .GT. 5) LSHIPS = 5

READ (SYSFIL,10) AS          ! Hull variant letter
VARS = AS(9:9)

READ (SYSFIL,10) AS          ! Cycle number
CALL SLENTH (AS,LAS)
CYCLS = AS(7:LAS)
CALL SLENTH (CYCLS,LCYCLS)

READ (SYSFIL,10) AS          ! Title
CALL SLENTH (AS,LTITLES)
TITLES = AS(9:LTITLES)
LTITLES = LTITLES - 8

READ (SYSFIL,10) AS          ! Option number
READ (AS,'(7X,I2)') OPTION

CLOSE (SYSFIL)

SIS = SMPIS(1:LSMPIS)//'\'//SHPTYPS(1:LSHPTYPS)//'\'//
2 SHIPS(1:LSHIPS)//VARS(1:1)//CYCLS(1:LCYCLS)
CALL SLENTH (SIS,LSIS)

SOS = SMPOS(1:LSMPOS)//'\'//SHPTYPS(1:LSHPTYPS)//'\'//
2 SHIPS(1:LSHIPS)//VARS(1:1)//CYCLS(1:LCYCLS)
CALL SLENTH (SOS,LSOS)

SDS = SMPDS(1:LSMPDS)//'\'//SHIPS(1:LSHIPS)//'\'//
2 SHIPS(1:LSHIPS)//VARS(1:1)
CALL SLENTH (SDS,LSDS)

RETURN
END

C DECK READ
SUBROUTINE READ

COMMON /APPEND/ NBKSET,NBKSTN(2),BKIMAG(2),BKFS(2),BKAS(2),
2 BKWD(2),BKSTN(10,2),BKHB(10,2),BKLNTH,BKWDTH,
2 BKWL(10,2),BKAN(10,2),NSKSET,SKIMAG(2),SKFLS(2),SKALS(2),
2 SKAUS(2),SKHB(2),SKFLWL(2),SKAUWL(2),NRDSET,RDIMAG(2),
2 RDRFS(2),RDRAS(2),RDRHB(2),RDRFWL(2),RDRAWL(2),RDTFS(2),RDTAS(2),
2 RDTHB(2),RDTFWL(2),RDTAWL(2),NSESET,SBIMAG(2),SOBRFS(2),SOBRAS(2),
2 SOBRHB(2),SOBRFW(2),SOBRWF(2),SIBRFS(2),SIBRAS(2),SIBRHB(2),
2 SIBRFW(2),SIBRAW(2),SBTFS(2),SBTAS(2),SBTHB(2),SBTFWL(2),
2 SBTAWL(2),NFNSET,FNIMAG(2),FNRFS(2),FNRAS(2),
2 FNRHB(2),FNRFWL(2),FNRRAWL(2),FNTFS(2),FNTAS(2),FNTHB(2),
2 FNTFWL(2),FNTAWL(2),NEXPRD,ENRDO(8),ENRDS(8)

COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
2 LRAOPR,ADRPR,ORGOPTN,GMNOM,KG,STATN(25),NSOFST(25),
2 NLEWF(25),HLFBTH(10,25),WTRLNE(10,25),BLEWF(25),TLEWF(25),
2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFB(10),YPTFB(10),
2 ZPTFB(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
2 STATNM,STATIS
CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
2 FBNUMB,PTNUMB,ORGOPTN
REAL KG

COMMON /ENVIOR/ VK,NVK,NMU,NMUM,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)

```

```

REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

COMMON /FINCON/ IACTFN,IFCLCS,FGAIN(8),FK(3),FA(3),FB(3),
2 FCLCS(8,2)

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDZ(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /INDEX/ PFIDX,LPFIDX,RMIDX,LRMIDX,SVIDX,LSVIDX
INTEGER LPFIDX,LRMIDX,LSVIDX
REAL PFIDX(235),RMIDX(183),SVIDX(3)

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL
INTEGER SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL

COMMON /LOADS/ NLOADS,SWGHT(25),SMASS(25),XLDSRN(10),XLDXPT(25),
2 LSTATN(25)

COMMON /PHYSICO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMTR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSC
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMTR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

COMMON /RESPN/ NRSPN,IPOINT(182),IMOTN(182),ITYPE(182),
2 ILIN(182),ISYM(182)
LOGICAL ILIN,ISYM

COMMON /SMPSYS/ FIS,AS,SIS,SOS,SDS,HALOS,DEV,PRN,SMPPS,SMPIS,
2 SMPOS,SMPDS,SHPTYP,SHIPS,VARS,CYCLS,TITLES,OPTION,LSIS,LSOS,
2 LSOS,LHALOS,LDEV,LPRN,LSMPPS,LSMPIS,LSMPOS,LSMPDS,LSHPTYP,
2 LSHIPS,LTITLE
CHARACTER*160 AS
CHARACTER*80 FIS,SIS,SOS,SDS,TITLES
CHARACTER*20 HALOS,DEV,PRN,SMPPS,SMPIS,SMPOS,SMPDS,SHPTYP
CHARACTER SHIPS*6,VARS*2,CYCLS*2
INTEGER*2 OPTION

COMMON /STATE/ LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL
LOGICAL LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL

CHARACTER*4 ALL,ROLL,EQLIN
CHARACTER*4 ROOT,TIP,ROOTOB,ROOTIB,LINES,LEWIS,TLEWIS
INTEGER HFE
DIMENSION FREQ1(30),FREQ2(30),FREQ3(30),FREQ4(30),
2 P(2,10),PSEG(8,9)
CHARACTER*4 SLAM(3),EMEG(3),SUBM(3),STNM1(3)
CHARACTER*4 METER

DATA METER // 'METE'/
DATA SLAM // 'SLA','MMIN','G' //
DATA EMEG // 'EME','RGEN','CE' //
DATA SUBM // 'SUB','MERG','ENCE'//
DATA FREQ1 /.2,.25,.28,.3,.32,.34,.36,.38,.4,.42,.44,.46,.48,.5,
2 .525,.55,.575,.6,.625,.65,.675,.7,.75,.8,.9,.1,.1.1,.1.2,.1.3,.2./
DATA FREQ2 /.2,.25,.3,.35,.4,.425,.45,.475,.5,.525,.55,.575,.6,
```

```

*   2 .65,.7,.75,.8,.85,.9,.95,1.,1.1,1.2,1.3,1.4,1.6,1.8,2.,2.2,2.4/
*   DATA FREQ3 / .2,.3,.4,.5,.6,.7,.8,.85,.9,.95,1.0,1.05,1.1,1.15,
*   2 1.2,1.3,1.4,1.5,1.6,1.7,1.8,1.9,2.,2.2,2.4,2.6,2.8,3.0,3.5,4.0/
DATA FREQ3 / .2,.3,.4,.5,.55,.575,.6,.625,.65,.675,.7,.725,.75,
*   2 .775,.8,.825,.85,.9,.95,1.,1.1,1.2,1.3,1.5,1.8,2.,2.5,3.,3.5,4.0/
DATA FREQ4 / .2,.4,.6,.8,1.,1.1,1.2,1.3,1.4,1.5,1.6,1.7,1.8,1.9,
*   2 2.,2.1,2.2,2.3,2.4,2.5,2.6,2.7,2.8,2.9,3.,3.2,3.4,3.6,3.8,4.0/
DATA ALL,ROLL / 'ALL','ROLL'/
DATA STNM1 / 'SIGN','IF.A','MPL'/
DATA EQVLIN / 'EQVL'/
DATA ROOT,TIP / 'ROOT','TIP'/
DATA ROOTOB,ROOTIB / 'RTOB','RTIB'/
DATA LINES,LEWIS / 'LINE','LEWI'/

      WRITE (IPRIN,990)
990  FORMAT (1H1,49X,22HINPUT CARD DESCRIPTION)

      FIS = SIS(1:LSIS) //'INP'
      OPEN (UNIT=ICARD,FILE=FIS,STATUS='OLD')

*     DATA CARD SET 1 - TITLE
      READ (ICARD,1000) TITLE
      WRITE (IPRIN,1010) TITLE
1000  FORMAT (20A4)
1010  FORMAT (///25H DATA CARD SET 1 - TITLE/5X,20A4)

*     DATA CARD SET 2 - PROGRAM OPTIONS
      READ (ICARD,1025) OPTN,VLACPR,RAOPR,RLDMPR,LRAOPR,ADRPR,ORGOPTN
      WRITE (IPRIN,1030) OPTN,VLACPR,RAOPR,RLDMPR,LRAOPR,ADRPR,ORGOPTN
1025  FORMAT (16I5)
1030  FORMAT (///35H DATA CARD SET 2 - PROGRAM OPTIONS//4X,6HOPTION,
      2 4X,6HVLCPR,5X,5HRAOPR,4X,6HRLDMPR,4X,6HLRAOPR,5X,5HADRPR,3X,
      2 7HORGOPTN/7I10)

*     DATA CARD SET 3 - PHYSICAL UNITS
      READ (ICARD,1040) PUNITS,RHO,GRAV,GNU
      WRITE (IPRIN,1050) PUNITS,RHO,GRAV,GNU
1040  FORMAT (2A4,2X,2F10.4,F10.8)
1050  FORMAT (///34H DATA CARD SET 3 - PHYSICAL UNITS//5X,5HUNITS,6X,
      2 3HRHO,6X,4HGRAV,5X,3HGNU/2X,2A4,2F10.4,F10.8)

*     DATA CARD SET 4 - HULL PARTICULARS
      READ (ICARD,1060) LPP,BEAM,DRAFT,DSPLMT,VKDES,VKINC,AMODL
      WRITE (IPRIN,1070) LPP,BEAM,DRAFT,DSPLMT,VKDES,VKINC,AMODL
1060  FORMAT (3F10.4,F10.2,3F10.4)
1070  FORMAT (///36H DATA CARD SET 4 - HULL PARTICULARS//6X,3HLPP,
      2 7X,4HBEAM,5X,5HDRAFT,4X,6HDSPLMT,5X,5HVKDES,5X,5HVKINC,5X,
      2 5HAMODL/3F10.4,F10.2,3F10.4)

*     speed definition
      IF (PUNITS(1) .NE. METER) VKMETR = VKMETR/FTMETR
      METRVK = 1./VKMETR
      CON = VKMETR/SQRT(GRAV*LPP)
      IF (VKINC .EQ. 0.) VKINC = 5.
      IV = 0
5     IV = IV + 1
      VK(IV) = (IV-1)*VKINC
      VFS(IV) = VKMETR*VK(IV)
      FRNUM(IV) = CON*VK(IV)
      IF (VK(IV) .LT. VKDES .AND. IV .LT. 8) GO TO 5
      NVK = IV
      FNDES = CON*VKDES

```

```

* DATA CARD SET 5 - LOAD PARTICULARS
    READ (ICARD,1080) GMNOM,DELGM,KG,KPITCH,KROLL,KYAW
    WRITE (IPRIN,1090) GMNOM,DELGM,KG,KPITCH,KROLL,KYAW
1080 FORMAT (8F10.4)
1090 FORMAT (//36H DATA CARD SET 5 - LOAD PARTICULARS//5X,5HGMNOM,
2 5X,5HDELGM,6X,2HKG,6X,6HKPITCH,5X,5HKROLL,6X,4HKYAW/
2 7F10.4)

* DATA CARD SET 6 - HULL LINES - LEWIS FORM OR OFFSETS
    READ (ICARD,1020) NSTATN,NLOADS
    WRITE (IPRIN,1100) NSTATN,NLOADS
1020 FORMAT (3I5)
1100 FORMAT (///47H DATA CARD SET 6 - HULL LINES - LEWIS FORM OR ,
2 7HOFFSETS//19H NO. OF STATIONS =,I3,4X,8HNLOADS =,I3//,
2 3X,7HSTATION,5X,5HNLEWF,6X,4HBEAM,5X,5HDRAFT,4X,6HSECARE,
2 5X,5HDBLWL/)
    DO 70 K=1,NSTATN
    READ (ICARD,1110) STATN(K),NSOFST(K),NLEWF(K)
1110 FORMAT (F10.4,2I5)
    IF (NSCFST(K) .EQ. 0) NSOFST(K) = 1
    NP = NSOFST(K)
    IF (NLEWF(K) .EQ. 0) GO TO 20
    TLEWIS = LEWIS
    READ (ICARD,1120) STATN(K),BLEWF(K),TLEWF(K),AREALF(K),DBLWL
1120 FORMAT (F10.4,10F7.2)
    IF (NP .GT. 1) GO TO 10
    HLFBTH(1,K) = 0.
    WTRLNE(1,K) = 0.
    GO TO 60
10  CALL GENOFS (BLEWF(K),TLEWF(K),AREALF(K),NP,HLFBTH(1,K),
2 WTRLNE(1,K),PI,DBLWL)
    GO TO 60
20 TLEWIS = LINES
    READ (ICARD,1120) STATN(K),(HLFBTH(J,K),J=1,NP)
    READ (ICARD,1120) STATN(K),(WTRLNE(J,K),J=1,NP)
    BLEWF(K) = 2*HLFBTH(1,K)
    IF (NP .GT. 1) GO TO 30
    TLEWF(K) = 0.
    AREALF(K) = 0.
    GO TO 60
30 DO 40 J=1,NP
    P(1,J) = HLFBTH(J,K)
    P(2,J) = WTRLNE(J,K)
40 CONTINUE
    NP1 = NP - 1
    CALL SPLNAR (P,NP,AREALF(K),PSEGS,NP1)
    AREALF(K) = 2*AREALF(K)
    BLEWF(K) = 2*BMAX(NP,HLFBTH(1,K))
    TLEWF(K) = WTRLNE(1,K)
    DO 50 J=1,NP
    IF (WTRLNE(J,K) .LT. TLEWF(K)) TLEWF(K) = WTRLNE(J,K)
50 CONTINUE
    TLEWF(K) = ABS(TLEWF(K) - WTRLNE(NP,K))
    AREALF(K) = AREALF(K)/(BLEWF(K)*TLEWF(K))
60 IF (NLEWF(K) .EQ. 0) WRITE (IPRIN,1130) STATN(K),TLEWIS,
2 BLEWF(K),TLEWF(K),AREALF(K)
    IF (NLEWF(K) .GT. 0) WRITE (IPRIN,1130) STATN(K),TLEWIS,
2 BLEWF(K),TLEWF(K),AREALF(K),DBLWL
1130 FORMAT (F10.4,5X,A4,1HS,4F10.4)
70 CONTINUE
    WRITE (IPRIN,1150)
1150 FORMAT (//3X,7HSTATION,3X,7HNOFFSET,5X,5HNLEWF,10X,
2 48HOFFSETS- Y=HALF BREADTH, Z=WATERLINE (FROM KEEL))
    DO 80 K=1,NSTATN
    NP = NSOFST(K)
    TLEWIS = LINES
    IF (NLEWF(K) .GT. 0) TLEWIS = LEWIS
    WRITE (IPRIN,1155)

```

```

1155 FORMAT (1X)
      WRITE (IPRIN,1160) STATN(K),NP,TLEWIS,(HLFBTH(J,K),J=1,NP)
      WRITE (IPRIN,1165) STATN(K),NP,WTRLNE(J,K),J=1,NP)
1160 FORMAT (F10.4,I10,5X,A4,1HS,4X,2HY=,10F7.2)
1165 FORMAT (F10.4,I10,5X,A4,1HS,4X,2HZ=,10F7.2)
80 CONTINUE

      IF (NLOADS .EQ. 0) GO TO 85

*     read weight curve

      READ (ICARD,1080) (SWGHT(K),K=1,NSTATN)
      WRITE (IPRIN,1333)
1333 FORMAT (//7X,12HWEIGHT CURVE//3X,7HSTATION,4X,6HWEIGHT/)
      SUMLD = 0.0
      SUMOM = 0.0
      DO 157 K=1,NSTATN
      WRITE (IPRIN,1080) STATN(K),SWGHT(K)
      SUMLD = SUMLD + SWGHT(K)
      SUMOM = SUMOM + SWGHT(K) * STATN(K)
157 CONTINUE
      XLDCG = (SUMOM / SUMLD) * (LPP / 20.0)
      XLDCGN = XLDCG / LPP
      WRITE (IPRIN,1334) SUMLD,XLDCG,PUNITS,XLDCGN
1334 FORMAT (10H TOTAL = ,F10.4,//,10X,25H C. OF GRAVITY (LCG)
2 F6.2,2A4,4X,12HLCG/LENGTH ,7X,F7.3)

*     read locations (stations) where loads are to be calculated

      READ (ICARD,1080) (XLDSTN(K),K=1,NLOADS)
      WRITE (IPRIN,1336)
1336 FORMAT (//4X,14HLOAD STATIONS-/)
      WRITE (IPRIN,1080) (XLDSTN(K),K=1,NLOADS)

85 CONTINUE

*     DATA CARD SET 7 - BILGEKEEL PARTICULARS

      READ (ICARD,1020) NBKSET
      WRITE (IPRIN,1170) NBKSET
1170 FORMAT (///4H DATA CARD SET 7 - BILGEKEEL PARTICULARS//4X,
2 6HNBKSET/I10)
      BKLNTH = 0.
      BKWDTH = 0.
      IF (NBKSET .EQ. 0) GO TO 105
      DO 100 K=1,NBKSET
      READ (ICARD,1180) NBKSTN(K),BKFS(K),BKAS(K),BKWD(K)
      WRITE (IPRIN,1190) K,NBKSTN(K),BKFS(K),BKAS(K),BKWD(K)
1180 FORMAT (15,5X,3F10.4)
1190 FORMAT (//5X,5HBKSET,4X,6HNBKSTN,6X,4HBKFS,6X,4HBKAS,6X,4HBKWD/
2 I10,3F10.4//5X,5HBKSTN,6X,4HBKFB,6X,4HBKWL,6X,4HBKAN/)
      BKIMAG(K) = 2
      NBKS = NBKSTN(K)
      DO 90 I=1,NBKS
      READ (ICARD,1080) BKSTN(I,K),BKHB(I,K),BKWL(I,K),BKAN(I,K)
      WRITE (IPRIN,1080) BKSTN(I,K),BKHB(I,K),BKWL(I,K),BKAN(I,K)
90 CONTINUE
      BKLNTH = BKLNTH + (BKAS(K) - BKFS(K))*LPP/20
      BKWDTH = BKWDTH + BKWD(K)
100 CONTINUE
      BKWDTH = BKWDTH/NBKSET
      WRITE (IPRIN,1200) BKLNTH,BKWDTH
1200 FORMAT (//10X,24HTOTAL BILGEKEEL LENGTH =,F10.4,4X,
2 14HAVERAGE SPAN =,F10.4)
105 CONTINUE

*     DATA CARD SET 8 - SKEG PARTICULARS

      READ (ICARD,1020) NSKSET
      WRITE (IPRIN,1210) NSKSET

```

```

1210 FORMAT ( //36H DATA CARD SET 8 - SKEG PARTICULARS//4X,6HNSKSET/
2 I10)
IF (NSKSET .EQ. 0) GO TO 115
WRITE (IPRIN,1211)
1211 FORMAT (//5X,5HSKSET,5X,5HSKFLS,5X,5HSKALS,5X,
2 5HSKAUS,6X,4HSKHB,4X,6HSKFLWL,4X,6HSKALWL,4X,6HSKAUWL)
DO 110 K=1,NSKSET
WRITE (IPRIN,i177)
1177 FORMAT (1X)
READ (ICARD,1080) SKFLS(K),SKALS(K),SKAUS(K),
2 SKHB(K),SKFLWL(K),SKALWL(K),SKAUWL(K)
WRITE (IPRIN,1185) K,SKFLS(K),SKALS(K),SKAUS(K),
2 SKHB(K),SKFLWL(K),SKALWL(K),SKAUWL(K)
1185 FORMAT (I10,7F10.4)
SKIMAG(K) = 1
IF (SKHB(K) .NE. 0.) SKIMAG(K) = 2
110 CONTINUE
115 CONTINUE

```

* DATA CARD SET 9 - RUDDER PARTICULARS

```

READ (ICARD,1020) NRDSET
WRITE (IPRIN,1220) NRDSET
1220 FORMAT ( //38H DATA CARD SET 9 - RUDDER PARTICULARS//4X,
2 6HNRDSET/I10)
IF (NRDSET .EQ. 0) GO TO 125
WRITE (IPRIN,1221)
1221 FORMAT (//5X,5HRDSET,2X,8HLOCATION,4X,6HFWDSTN,
2 4X,6HAFTSTN,5X,5HHLFBM,5X,5HFWDWL,5X,5HAFTWL)
DO 120 K=1,NRDSET
WRITE (IPRIN,1177)
READ (ICARD,1080) RDRFS(K),RDRAS(K),RDRHB(K),
2 RDRFWL(K),RDRRAWL(K)
WRITE (IPRIN,1195) K,ROOT,RDRFS(K),RDRAS(K),
2 RDRHB(K),RDRFWL(K),RDRRAWL(K)
1195 FORMAT (I10,4X,A4 2X,7F10.4)
READ (ICARD,1080) RDTFS(K),RDTAS(K),RDTHB(K),
2 RDTFWL(K),RDTAWL(K)
WRITE (IPRIN,1195) K,TIP,RDTFS(K),RDTAS(K),
2 RDTHB(K),RDTFWL(K),RDTAWL(K)
RDIMAG(K) = 1
IF (RDRHB(K) .NE. 0.) RDIMAG(K) = 2
120 CONTINUE
125 CONTINUE

```

* DATA CARD SET 10 - PROPELLER SHAFT BRACKETS

```

READ (ICARD,1020) NSBSET
WRITE (IPRIN,1510) NSBSET
1510 FORMAT ( //45H DATA CARD SET 10 - PROPELLER SHAFT BRACKETS//
2 4X,6HNSBSET/I10)
IF (NSBSET .EQ. 0) GO TO 129
WRITE (IPRIN,1520)
1520 FORMAT (//5X,5HSBSET,2X,8HLOCATION,4X,6HFWDSTN,4X,6HAFTSTN,5X,
2 5HHLFBM,5X,5HFWDWL,5X,5HAFTWL)
DO 128 K=1,NSBSET
SBIMAG(K) = 2
READ (ICARD,1080) SOBRFS(K),SOBRAS(K),SOBRHB(K),SOBRFW(K),
2 SOBRAW(K)
WRITE (IPRIN,1195) K,ROOTOB,SOBRFS(K),SOBRAS(K),SOBREB(K),
2 SOBRFW(K),SOBRAW(K)
READ (ICARD,1080) SBTFS(K),SBTAS(K),SBTHB(K),SBTFWL(K),SBTAWL(K)
WRITE (IPRIN,1195) K,TIP,SBTFS(K),SBTAS(K),SBTHB(K),SBTFWL(K),
2 SBTAWL(K)
IF (SBTHB(K) .EQ. 0. .AND. SOBRHB(K) .EQ. 0.) SBIMAG(K) = 1
IF (SBTHB(K) .EQ. 0. .OR. SBTHB(K) .EQ. SOBRHB(K)) GO TO 128
READ (ICARD,1080) SIBRFS(K),SIBRAS(K),SIBRHB(K),SIBRFW(K),
2 SIBRAW(K)
WRITE (IPRIN,1195) K,ROOTIB,SIBRFS(K),SIBRAS(K),SIBRHB(K),
2 SIBRFW(K),SIBRAW(K)

```

```
128 CONTINUE  
129 CONTINUE
```

```
* DATA CARD SET 11 - FIN PARTICULARS
```

```
READ (ICARD,1020) NFNSET,IACTFN,IFCLCS  
WRITE (IPRIN,1230) NFNSET,IACTFN,IFCLCS  
1230 FORMAT ( //36H DATA CARD SET 11 - FIN PARTICULARS//  
2 4X,6HNFSNSET,4X,6HIACTFN,4X,6HIFCLCS/3I10)  
IF (NFNSET .EQ. 0) GO TO 135  
IF (IACTFN .EQ. 0) GO TO 132  
READ (ICARD,1080) {FGAIN(IV),IV=1,NVK}  
WRITE {IPRIN,2010} {VK(IV) IV=1,NVK}  
2010 FORMAT ( /22H SHIP SPEED (KNOTS) =,8F10.3)  
WRITE (IPRIN,2020) (FGAIN(IV),IV=1,NVK)  
2020 FORMAT ( /22H FIN GAIN FACTORS =,8F10.3)  
READ (ICARD,1080) FK  
WRITE {IPRIN,2030} FK  
2030 FORMAT ( /22H CONTROLLER COEFF. =,3F10.3)  
READ (ICARD,1080) FA  
WRITE (IPRIN,2040) FA  
2040 FORMAT ( /22H SERVO COEFFICIENTS =,3F10.3)  
READ (ICARD,1080) FB  
WRITE (IPRIN,2050) FB  
2050 FORMAT ( /22H COMPENSATION COEFF.=,3F10.3)  
132 IF (IFCLCS .EQ. 0) GO TO 136  
WRITE (IPRIN,2060)  
2060 FORMAT ( /39X,30H CORRECTED FIN LIFT CURVE SLOPE)  
WRITE (IPRIN,2010) (VK(IV),IV=1,NVK)  
WRITE (IPRIN,1177)  
DO 134 K=1,NFNSET  
READ (ICARD,1080) (FCLCS(IV,K),IV=1,NVK)  
WRITE (IPRIN,2070) K,(FCLCS(IV,K),IV=1,NVK)  
2070 FORMAT (7H FNSET,I2,13H - FCLCS =,8F10.3)  
134 CONTINUE  
136 CONTINUE  
WRITE (IPRIN,1231)  
1231 FORMAT ( /6X,6HNFSNSET,2X,8H LOCATION,4X,6HFWDSTN,4X,  
2 6HAFTSTN,5X,5HHFBM,5X,5HFWDWL,5X,SHAFTWL)  
DO 130 K=1,NFNSET  
WRITE (IPRIN,1177)  
READ (ICARD,1080) FNRFS(K),FNRAS(K),FNRHB(K),  
2 FNRFWL(K),FNRRAWL(K)  
WRITE (IPRIN,1195) K,ROOT,FNRFS(K),FNRAS(K),FNRHB(K),  
2 FNRFWL(K),FNRRAWL(K)  
READ (ICARD,1080) FNTFS(K),FNTAS(K),FNTHB(K),  
2 FNTFWL(K),FNTAWL(K)  
WRITE (IPRIN,1195) K,TIP,FNTFS(K),FNTAS(K),FNTHB(K),  
2 FNTFWL(K),FNTAWL(K)  
FNIMAG(K) = 1.  
IF (FNRHB(K) .NE. 0.) FNIMAG(K) = 2.  
130 CONTINUE  
135 CONTINUE
```

```
* DATA CARD SET 12 - MOTIONS AT A POINT
```

```
READ (ICARD,1020) NPTLOC,HFE  
WRITE (IPRIN,1240) NPTLOC,HFE  
1240 FORMAT ( //39H DATA CARD SET 12 - MOTIONS AT A POINT//4X,  
2 6HNPTLOC,19X,3HHFE/I10,20X,I2/)  
IF (NPTLOC .EQ. 0) GO TO 145  
WRITE (IPRIN,1241)  
1241 FORMAT ( /4X,6HNUMBER,8X,4HNAME,39X,6HXPTLOC,4X,6HYPTLOC,4X,  
2 6HZPTLOC)  
DO 140 K=1,NPTLOC  
WRITE (IPRIN,1177)  
READ (ICARD,1250) PTNUMB(K),(PTNAME(I,K),I=1,8),XPTLOC(K),  
2 YPTLOC(K),ZPTLOC(K)  
WRITE (IPRIN,1260) PTNUMB(K),(PTNAME(I,K),I=1,6),XPTLOC(K),  
2 YPTLOC(K),ZPTLOC(K)
```

```

1250 FORMAT (15.5X,8A4,8X,3F10.4)
1260 FORMAT (110,4X,8A4,11X,3F10.4)
140 CONTINUE
145 CONTINUE

* DATA CARD SET 13 - RELATIVE MOTION

    READ (ICARD,1020) NFREBD,NBB
    WRITE (IPRIN,1270) NFREBD,NBB
1270 FORMAT ( //36H DATA CARD SET 13 - RELATIVE MOTION//
2 4X,6HNFREBD,4X,3HNBB/I10,17)
    IF (NFREBD .EQ. 0) GO TO 155
    WRITE (IPRIN,1271)
1271 FORMAT ( //4X,6HNUMBER,8X,4HNAME,20X,6HFBCODE,13X,6HXPTFBD,4X,
2 6HYPTFBD,4X,6HZPTFBD,10X,4HRDOT)
    DO 150 K=1,NFREBD
    WRITE (IPRIN,1177)
    READ (ICARD,1272) FBNUMB(K),(FBNAME(I,K),I=1,5),FBCODE(K),
2 XPTFBD(K),YPTFBD(K),ZPTFBD(K),RDOT(K)
1272 FORMAT (15.5X,5A4,I5,5X,4F10.4)
* RDOT = 12*SQRT(LPP/520) IN ENGLISH UNITS
* RDOT = 3.66*SQRT(LPP/158.5) IN METRIC UNITS
    IF (FBCODE(K) .LE. 0) FBCODE(K) = 1
    J = FBCODE(K)
    DO 148 I=1,3
    IF (J .EQ. 1) FBTYPE(I,K) = SLAM(I)
    IF (J .EQ. 2) FBTYPE(I,K) = EMEG(I)
    IF (J .EQ. 3) FBTYPE(I,K) = SUBM(I)
148 CONTINUE
    WRITE (IPRIN,1273) FBNUMB(K),(FBNAME(I,K),I=1,5),FBCODE(K),
2 (FBTYPE(I,K),I=1,3),XPTFBD(K),YPTFBD(K),ZPTFBD(K),RDOT(K)
1273 FORMAT (I10,4X, 5A4,I5,2H =,3A4,4X,3F10.4,4X,F10.4)
150 CONTINUE
155 CONTINUE

* DATA CARD SET 14 - SEASTATE AND ROLL ITERATION

    READ (ICARD,1280) NSIGWH,STATIS,(STATNM(I),I=1,3)
    WRITE (IPRIN,1290) NSIGWH,STATIS,(STATNM(I),I=1,3)
    IF (NSIGWH .EQ. 0) GO TO 165
1280 FORMAT (15.5X,F10.4,5A4)
1290 FORMAT ( //48H DATA CARD SET 14 - SEASTATE AND ROLL ITERATION//
2 4X,6HNSIGWH,4X,14HSTATISTIC (SA),4X,14HSTATISTIC NAME/
2 I10,4X,F10.4,10X,3A4//5X,5HSIGWH/)
    DO 160 K=1,NSIGWH
    READ (ICARD,1080) SIGWH(K)
    WRITE (IPRIN,1080) SIGWH(K)
160 CONTINUE
165 CONTINUE

* DATA CARD SET 15 - STOP

    READ (ICARD,1000) STOP
    WRITE (IPRIN,1310) STOP
1310 FORMAT (///26H DATA CARD SET 15 - STOP//4X,4HSTOP/4X,A4)

* inactive data card set used for inputting particular responses

    NRESP = 0
    IF (NRESP .EQ. 0) GO TO 200
    WRITE (IPRIN,1305)
1305 FORMAT (///40H DATA CARD SET 16 - RESPONSE DEFINITION//4X,
2 45HNRESP POINT MOTN TYPE LIN SYM/)
    DO 190 IR=1,NRESP
    READ (ICARD,1315) IP,IM,IT
1315 FORMAT (3I6)
    IPOINT(IR) = IP
    IMOTN(IR) = IM
    ITYPE(IR) = IT

```

```

ILIN(IR) = .TRUE.
ISYM(IR) = .TRUE.
IF (IP .GT. 0) GO TO 180
IF (IM.EQ.2 .OR. IM.EQ.4 .OR. IM.EQ.6 .OR. IM.EQ.7)
2 ILIN(IR) = .FALSE.
GO TO 185
180 IF ((IM.EQ.1 .OR. IM.EQ.3) .AND. YPTLOC(IP).NE.0.)
2 ILIN(IR) = .FALSE.
IF (IM.EQ.2) ILIN(IR) = .FALSE.
IF ((IM.EQ.1 .OR. IM.EQ.3) .AND. YPTLOC(IP).NE.0.)
2 ISYM(IR) = .FALSE.
IF (IM.EQ.8 .AND. YPTFBD(IP).NE.0.)
2 ILIN(IR) = .FALSE.
IF ((IM.EQ.8 .OR. IM.EQ.9) .AND. YPTFBD(IP).NE.0.)
2 ISYM(IR) = .FALSE.
185 WRITE (IPRIN,1320) IR,IPOINT(IR),IMOTN(IR),ITYPE(IR),ILIN(IR),
2 ISYM(IR)
1320 FORMAT (4I8,2L8)
190 CONTINUE
200 CONTINUE
CLOSE (UNIT=ICARD)
WRITE (IPRIN,1330)
1330 FORMAT (///20H END DATA CARD SETS)

* note on REYNOLDS no. scaling for frictional roll damping
* REYN = V*LPP/VNY . Since viscosity, VNY, is assumed constant,
* REYN scales as LPP**2/T. The period scales as SQRT(LPP). Thus
* REYN scales as LPP**1.5

REYSCL = 1.
IF (AMODL .GT. 0.) REYSCL = (AMODL/LPP)**1.5

* hull form transformations to internal coordinate system

* find distance from baseline to waterline, dblwl

DBLWL = 0.
DO 205 I=1,NSTATN
NP = NSOFST(I)
WL = WTRLNE(NP,I)
IF (WL .GT. DBLWL) DBLWL = WL
205 CONTINUE
K = NSTATN + 1
DO 220 I=1,NSTATN
K = K - 1
X(K) = LPP - STATN(I)*LPP/20
NOFSET(K) = NSOFST(I)
NP = NOFSET(K)
DO 210 J=1,NP
Y(J,K) = HLFBTH(J,I)
Z(J,K) = WTRLNE(J,I) - DBLWL
IF (NP .GT. 1) GO TO 210
Z(J,K) = 0.
210 CONTINUE
220 CONTINUE

* MOTIONS-AT-A-POINT transformation

IF (NPTLOC .EQ. 0) GO TO 240
NPTS = NPTLOC
DO 230 IP=1,NPTS
XPT(IP) = LPP - XPTLOC(IP)*LPP/20
YPT(IP) = YPTLOC(IP)
ZPT(IP) = ZPTLOC(IP) - DBLWL
230 CONTINUE
240 CONTINUE

* relative motion location transformation

IF (NFREBD .EQ. 0) GO TO 260
DO 250 IP=1,NFREBD
FBDX(IP) = LPP - XPTFBD(IP)*LPP/20

```

```

FBDY(IP) = YPTFBDS(IP)
FBDZ(IP) = ZPTFBDS(IP) - DBLWL
250 CONTINUE
260 CONTINUE

*    ENVIOR default values

*    sigma defaults

    NSIGMA = 10
    SIGMA(1) = .05
    SIGMA(2) = .10
    SIGMA(3) = .25
    SIGMA(4) = .50
    SIGMA(5) = .75
    SIGMA(6) = 1.00
    SIGMA(7) = 1.50
    SIGMA(8) = 2.00
    SIGMA(9) = 5.00
    SIGMA(10) = 10.00

*    heading definition

    DO 420 IV=1,NVK
    NNMU(IV) = 13
    NH = NNMU(IV)
    DO 410 IH=1,NH
    MU(IH,IV) = (IH-1)*15*DEGRAD
410 CONTINUE
420 CONTINUE

*    wave frequency definition

    RGYRAD = KROLL*BEAM
    ROLPER = 10.
    IF (GMNOM .GT. 0.)
2     ROLPER = (TPI/SQRT(GRAV)) + SQRT(1.25+RGYRAD**2/(GMNOM DELGM))
    ICASE = 0
    IF (ROLPER .LE. 15.) ICASE = 1
    IF (ROLPER .LE. 9.) ICASE = 2
    IF (ROLPER .LE. 5.) ICASE = 3
    NOMEWA = 30
    DO 430 IW=1,NOMEWA
    OMEGA(IW) = FREQ1(IW)
    IF (ICASE .EQ. 1) OMEGA(IW) = FREQ2(IW)
    IF (ICASE .EQ. 2) OMEGA(IW) = FREQ3(IW)
    IF (ICASE .EQ. 3) OMEGA(IW) = FREQ4(IW)
430 CONTINUE
    IF (NSIGWH .GT. 0) GO TO 450

*    seastate default values

*    SS4(2M), SS5(3M), SS6(5M), AND SS7(7.5M)

    NSIGWH = 4
    IF (ICASE .GT. 0) GO TO 433
    SIGWH(1) = 2.0
    SIGWH(2) = 3.0
    SIGWH(3) = 5.0
    SIGWH(4) = 7.5
    GO TO 437

*    SS3(1.5M), SS4(2M), SS5(3M), AND SS6(5M)

433 IF (ICASE .GT. 1) GO TO 435
    SIGWH(1) = 1.5
    SIGWH(2) = 2.0
    SIGWH(3) = 3.0
    SIGWH(4) = 5.0
    GO TO 437

*    SS2(1M), SS3(1.5M), SS4(2M), AND SS5(3M)

```

```

435  SIGWH(1) = 1.0
      SIGWH(2) = 1.5
      SIGWH(3) = 2.0
      SIGWH(4) = 3.0
437  CONTINUE
      IF (PUNITS(1) .EQ. METER) GO TO 450
      DO 440 I=1,NSIGWH
440  SIGWH(I) = SIGWH(I)/FTMETR
450  CONTINUE

*      statistic default value
      IF (STATIS .GT. 0.) GO TO 470
      STATIS = 2.00
      DO 460 I=1,3
      STATNM(I) = STNM1(I)
460  CONTINUE
470  CONTINUE

*      modal wave period definition
      NTMOD = 8
      PERINT = 7.0
      IF (ICASE .EQ. 1) PERINT = 5.0
      IF (ICASE .EQ. 2) PERINT = 3.0
      IF (ICASE .EQ. 3) PERINT = 3.0
      DO 500 IT=1,NTMOD
      TMODAL(IT) = PERINT + (IT-1) * 2.

*      define 2-parameter (significant wave height, modal wave period)
*      Bretschneider sea spectra, for unit significant wave height
      CALL BRWVSP (NOMEGA,1.,TMODAL(IT),OMEGA,S(1,IT))

500  CONTINUE

*      mean roll angle definition
      NRANG = 8
      RLANG(1) = .50
      RLANG(2) = 1.00
      RLANG(3) = 2.50
      RLANG(4) = 5.00
      RLANG(5) = 10.00
      RLANG(6) = 15.00
      RLANG(7) = 25.00
      RLANG(8) = 40.00
      DO 35 IA=1,NRANG
      RANG(IA) = RLANG(IA)*DEGRAD
35   CONTINUE

*      response definitions (max of 182)
*      6*3 = 18 origin
*      1*3 = 3 fins
*      10*3*3 = 90 point
*      1 = 1 added resistance
*      10*(2+3) = 50 rel+abs
*      10*2 = 20 loads
      IF (NRESP .GT. 0) GO TO 580
      IF (OPTN.NE.3 .AND. OPTN.NE.5) GO TO 510

*      roll
      L = 1
      IPOINT(1) = 0
      IMOTN(1) = 4
      ITYPE(1) = 1
      ILIN(1) = .FALSE.
      ISYM(1) = .TRUE.

```

```

        IF (.N.GT.(VLACPR.GT.0.OR.STATNM(1).EQ.EQVLIN)) GO TO 502
* roll velocity
    L = L + 1
    IPOINT(2) = 0
    IMOTN(2) = 4
    ITYPE(2) = 2
    ILIN(2) = .FALSE.
    ISYM(2) = .TRUE.

502 IF (IACTFN .EQ. 0) GO TO 508
* fin & fin velocity
    M1 = 1
    IF (VLACPR .GT. 0) M1 = 2
    DO 505 IT=1,M1
    L = L + 1
    IPOINT(L) = 0
    IMOTN(L) = 9
    ITYPE(L) = IT
    ILIN(L) = .FALSE.
    ISYM(L) = .TRUE.
505 CONTINUE
508 NRESP=L
    GO TO 580
510 CONTINUE
* 6 DOF responses at the origin
    L = 0
    DO 520 J=1,3
    DO 520 I=1,6
    L = L + 1
    IPOINT(L) = 0
    IMOTN(L) = I
    ITYPE(L) = J
    ILIN(L) = .TRUE.
    IF (I.EQ.2 .OR. I.EQ.4 .OR. I.EQ.6) ILIN(L) = .FALSE.
    ISYM(L) = .TRUE.
520 CONTINUE
    IF (IACTFN .EQ. 0) GO TO 525
* fin, fin velocity, and fin acceleration
    DO 522 IT=1,3
    L = L + 1
    IPOINT(L) = 0
    IMOTN(L) = 9
    ITYPE(L) = IT
    ILIN(L) = .FALSE.
    ISYM(L) = .TRUE.
522 CONTINUE
525 CONTINUE
    IF (NPTLOC .EQ. 0) GO TO 535
    IF (HFE.EQ.0) GO TO 528
* horizontal force estimator
    DO 527 K= 1,NPTLOC
    L=L+1
    IPOINT(L) = K
    IMOTN(L) = 15
    ITYPE(L) = 1

```

```

      ILIN(L) = .FALSE.
      ISYM(L) = .TRUE.
527  CONTINUE

*      RESPONSES AT SELECTED POINTS

528  DO 530 K=1,NPTLOC
      DO 530 J=1,3
      DO 530 I=1,3
      L = L + 1
      IPOINT(L) = K
      IMOTN(L) = I
      ITYPE(L) = J
      ILIN(L) = .TRUE.
      IF ((I.EQ.1 .OR. I.EQ.3) .AND. YPTLOC(K).NE.0.) ILIN(L) = .FALSE.
      IF (I.EQ. 2) ILIN(L) = .FALSE.
      ISYM(L) = .TRUE.
      IF ((I.EQ.1 .OR. I.EQ.3) .AND. YPTLOC(K).NE.0.) ISYM(L) = .FALSE.
530  CONTINUE

535  CONTINUE
      IF (ADRPR .EQ. 0) GO TO 537

*      added resistance

      L = L + 1
      IPOINT(L) = 0
      IMOTN(L) = 7
      ITYPE(L) = 1
      ILIN(L) = .FALSE.
      ISYM(L) = .FALSE.

537  CONTINUE
      IF (NFREBD .EQ. 0) GO TO 570

*      relative motions and velocities at points

      DO 560 K=1,NFREBD
      DO 540 J=1,2
      L = L + 1
      IPOINT(L) = K
      IMOTN(L) = 8
      ITYPE(L) = J
      ILIN(L) = .TRUE.
      IF (YPTFBD(K) .NE. 0.) ILIN(L) = .FALSE.
      ISYM(L) = .TRUE.
      IF (YPTFBD(K) .NE. 0.) ISYM(L) = .FALSE.
540  CONTINUE
560  CONTINUE

570  CONTINUE
      IF (NLOADS .EQ. 0) GO TO 700

*      loads at specified stations

      DO 620 K=1,NLOADS
*      I=10 (H.SHEAR) (NOT CALCULATED)
*      I=11 (V.SHEAR)
*      I=12 (T.MOM.) (NOT CALCULATED)
*      I=13 (V.MOM.)
*      I=14 (H.MOM.) (NOT CALCULATED)
          DO 610 I=10,14
          IF (.NOT. (I.EQ.11.OR.I.EQ.13)) GO TO 610
          L = L + 1
          IPOINT(L) = K
          IMOTN(L) = I
          ITYPE(L) = 1
          ILIN(L) = .TRUE.
          ISYM(L) = .TRUE.

```

```

610  CONTINUE
620  CONTINUE
700  CONTINUE
      NRESP = L
580  CONTINUE
*   state definitions
      VRT = .TRUE.
      LAT = .TRUE.
      LOADS = .FALSE.
      IF (NLOADS .GT. 0) LOADS = .TRUE.

*   CDC    ADDRES = .TRUE.
*           IF (ISKIP .EQ. 1) ADDRES = .FALSE.
*           IF (ADRPR .EQ. 0) ADDRES = .FALSE.

      BKEEL = .FALSE.
      IF (NBKSET .GT. 0) BKEEL = .TRUE.
      EXROLL = .FALSE.
      KYAWRL = 0.
      NEXPRD = 0
      IF (NEXPRD .GT. 0) EXROLL = .TRUE.

*   modified to run on VAX/VMS
*   IF (OPTN .EQ. 6 .AND. RAOPR .EQ. 2 .OR. LRAOPR .EQ. 2) GO TO 590
*   open random access files
      LPFIDX = 235
*   CDC    CALL OPENMS (POTFIL,PFIDX,LPFIDX,0)
      LRMIDX = 183
*   CDC    CALL OPENMS (RMSFIL,RMIDX,LRMIDX,0)
      LSVIDX = 3
*   CDC    CALL OPENMS (SEVFIL,SVIDX,LSVIDX,0)
590  CONTINUE

      WRITE (IPRIN,'(SF8.3)') OMEGA
      WRITE (IPRIN,'(SF8.3)') TMODAL
      RETURN
      END

C DECK REGWAV
SUBROUTINE REGWAV

COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
2 LRAOPR,ADRPR,ORGOPTN,GMNOM,KG,STATN(25),NSOFST(25),
2 MLEWF(25),HLFBTH(10,25),WTRLNE(10,25),BLEWF(25),TLEWF(25),
2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
2 ZPTLOC(10),NBB,FENUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
2 ZPTFBD(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
2 STATNM,STATIS
CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
2 FENUMB,PTNUMB,ORGOPTN
REAL KG

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFILE,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL
INTEGER SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFILE,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL

COMMON /SMPSYS/ FIS,AS,SIS,SOS,SDS,HALOS,DEV,PRN,SMPPS,SMPIS,
2 SMPDS,SHPTYP,SHIPS,VARS,CYCLS,TITLES,OPTION,LSIS,LSOS,
2 LSDS,LHALUS,LDEV,LPRN,LSHPPS,LSMPIS,LSMPDS,LSHPTYP,
2 LSHIPS,LTITLES
CHARACTER*160 AS

```

```

CHARACTER*80 FIS,SIS,SOS,SDS,TITLES
CHARACTER*20 HALOS,DEV,PRN,SMPPS,SMPIS,SMPOS,SMPDS,SHPTYPS
CHARACTER SHIPS*6,VARS*2,CYCLS*2
INTEGER*2 OPTION

IF (OPTN .EQ. 6 ) GO TO 10

IF (OPTN.NE.2 .AND. OPTN.NE.3 .AND. ORGOPTN.EQ.2) GO TO 10

AS = '(/4X,"CALLING HYDCAL")'
WRITE (*,AS)
WRITE (TEXFIL,AS)

CALL HYDCAL

AS = '(4X,"CALLING RDBASE")'
WRITE (*,AS)
WRITE (TEXFIL,AS)

CALL RDBASE

AS = '(4X,"CALLING EQMOTN")'
WRITE (*,AS)
WRITE (TEXFIL,AS)

CALL EQMOTN

10 CONTINUE

RETURN
END

C DECK RELMOT
SUBROUTINE RELMOT (IM,NL,NU,MOTV,MOTL,XPT,YPT,RAO1,PHS1,RAO2,
2 PHS2,NMOT,NPLANE,NOMEGA,OMEGA,COSMU,SINMU,GRAV,RADDEG,IPHS)

* This routine computes relative and absolute motion
* WAVE = EXP(-I(K*(XP*COSMU + YP*SINMU)) + K*ZP + I(WE*T))
* VERT = HEAVE - XP*PITCH + YP*ROLL
* RELMOT = VERT - WAVE
* W.G.MEYERS, DTNSRDC, 021679

COMPLEX MOTV(NMOT,NOMEGA),MOTL(NMOT,NOMEGA),CVER,CWAVE,HEAVE,
2 PITCH,ROLL,TFN
DIMENSION OMEGA(NOMEGA),RAO1(NOMEGA),PHS1(NOMEGA),RAO2(NOMEGA),
2 PHS2(NOMEGA)

DO 30 I=NL,NU
HEAVE = MOTV(2,I)
PITCH = MOTV(3,I)
ROLL = MOTL(2,I)
DO 20 J=1,NPLANE
IF (J .EQ. 2) ROLL = - ROLL
TFN = HEAVE - XPT*PITCH + YPT*ROLL
CVER = TFN
IF (J .EQ. 1) WAVNUM = OMEGA(I)*OMEGA(I)/GRAV
IF (J .EQ. 2) SINMU = - SINMU
ARG = - WAVNUM*(XPT*COSMU + YPT*SINMU)
AR = COS(ARG)
AI = SIN(ARG)
CWAVE = CMPLX(AR,AI)
TFN = CVER - CWAVE
IF (J .EQ. 2) SINMU = - SINMU
IF (J .EQ. 1) CALL RAOPHA (TFN,RAO1(I),PHS1(I),RADDEG,IPHS)
IF (J .EQ. 2) CALL RAOPHA (TFN,RAO2(I),PHS2(I),RADDEG,IPHS)
20 CONTINUE
30 CONTINUE

RETURN
END

C DECK REVAL

```

```

FUNCTION REVAL (RSPLNE,WEIGHT)
DIMENSION RSPLNE(4),WEIGHT(4)

REVAL = 0
DO 10 I=1,4
  REVAL = REVAL + WEIGHT(I)*RSPLNE(I)
10 CONTINUE

RETURN
END

C DECK RLITER
SUBROUTINE RLITER (SPINDX,TOINDX,NSPIND,NTOIND,DATA,IC,RLCALC,
2 ROLL)

* roll iteration

COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
2 LRAOPR,ADRPR,ORGOPTN,GMNOM,KG,STATN(25),NSOFST(25),
2 NLEWF(25),HLFBTH(10,25),WTRLNE(10,25),BLEWF(25),TLEWF(25),
2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
2 ZPTFBD(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
2 STATNM,STATIS
CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
2 FBNUMB,PTNUMB,ORGOPTN
REAL KG

COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
INTEGER NVK,NMU,OMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8),
REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

COMMON /INDEX/ PFIDX,LPFIDX,RMIDX,LRMIDX,SVIDX,LSVIDX
INTEGER LPFIDX,LRMIDX,LSVIDX
REAL PFIDX(235),RMIDX(183),SVIDX(3)

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL
INTEGER SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL

COMMON /RESPN/ NRESP,IPOINT(182),IMOTN(182),ITYPE(182),
2 ILIN(182),ISYM(182)
LOGICAL ILIN,ISYM

LOGICAL LINEAR,SYMMET
DIMENSION DATA(432),SPINDX(9),TOINDX(9),RLCALC(8,24),
2 ROLL(13,64,4)

IR = 1
DO 5 N=1,NRESP
  IF (IMOTN(N).EQ.4 .AND. ITYPE(N).EQ.1) IR = N
5 CONTINUE
KR = IR + 1
LINEAR = ILIN(IR)
SYMMET = ISYM(IR)
NPREDH = 13
NDATA = (2 + 2*NRANG)*NPREDH
DO 300 IS=1,NSIGWH
  K = 0
  CON = SIGWH(IS)*STATIS
  DO 200 ITO=1,NTMOD
    DO 100 IV=1,NVK
      K = K + 1
      CALL FETCH (KR,IV,ITO,DATA,RMIDX,SPINDX,TOINDX,NDATA,LRMIDX,
2 NVK,NTMOD,RMSFIL)

```

```

L = 2*NPREDH
DO 10 IA=1,NRANG
DO 10 IH=1,NPREDH
IF (IC .EQ. 1) TEMP = DATA(L+1)
IF (IC .EQ. 2) TEMP = DATA(L+2)
L = L + 2
RLCALC(IA,IH) = TEMP*CON
10 CONTINUE
DO 50 IH=1,NPREDH
CALL RLITR (RLANG,NRANG,RLCALC(1,IH),ROLL(IH,K,IS))
50 CONTINUE
100 CONTINUE
200 CONTINUE
300 CONTINUE

RETURN
END

C DECK RLITR
SUBROUTINE RLITR (RLANG,NRANG,RLCALC,RLANS)
DIMENSION RLANG(8),RLCALC(8),DIFF(8),ELM(4,8)

DO 10 IA=1,NRANG
DIFF(IA) = RLANG(IA) - RLCALC(IA)
10 CONTINUE
XO = 0.
IF (XO .GE. DIFF(1)) GO TO 20
RLANS = RLCALC(1)
GO TO 40
20 IF (XO .LE. DIFF(NRANG)) GO TO 30
RLANS = RLCALC(NRANG)
GO TO 40
30 CALL SPFIT (DIFF,RLANG,ELM,NRANG)
CALL SPLVAL (DIFF,NRANG,ELM,O.,RLANS,DUM,IELM)
40 CONTINUE

RETURN
END

C DECK RMS
SUBROUTINE RMS (KREC,RAO1,RAO2,IT,N,R,B2,NPREDH,NLCH,N1,N2,DATA,
2 IRESP,NBETA)
COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8),
REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

DIMENSION DATA(432),KREC(13),RAO1(30,8,13),RAO2(30,8,11),R(30),
2 B2(35)
REAL LMS(24)

5 CONTINUE
L = 2*NPREDH
DO 60 IA=1,N
DO 40 IH=1,NMU
I1 = N1 + IH
I2 = N2 - IH
IF (I2 .LE. 0) I2 = I2 + NBETA
IF (KREC(IH) .GT. 0) GO TO 10
LMS(I1) = 0.
GO TO 40
10 DO 20 I=1,NOMEGA
20 R(I) = RAO1(I,IA,IH)*S(I,IT)
CALL ALGRNG (NOMEGA,OMEGA,R,LMS(I1))
IF (KREC(IH) .EQ. 1) LMS(I2) = LMS(I1)
IF (KREC(IH) .EQ. 1) GO TO 40
KH = IH - 1
DO 30 I=1,NOMEGA
30 R(I) = RAO2(I,IA,KH)*S(I,IT)

```

```

40 CALL ALGRNG (NOMEGA,OMEGA,R,LMS(I2))
40 CONTINUE

DO 50 IPH=1,NPREDH
CALL XMSSC (IPH,B2,LMS,NLCH,RMSLC,RMSSC)
IF (IRESP .EQ. 7) GO TO 45
RMSLC = SQRT(RMSLC)
RMSSC = SQRT(RMSSC)
45 L = L + 1
DATA(L) = RMSLC
L = L + 1
DATA(L) = RMSSC
50 CONTINUE
60 CONTINUE

RETURN
END

C DECK RMSOUT
SUBROUTINE RMSOUT

COMMON /DATINP/ OPTN,MOTN,BSCFIL,VACPR,RAOPR,RLDMPR,DISPLMT,
2 LRAOPR,ADRPR,ORGOPTN,GMNOM,KG,STATN(25),NSOFST(25),
2 NLEWF(25),HLFBTH(10,25),WTRLNE(10,25),BLEWF(25),TLEWF(25),
2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
2 ZPTFBD(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
2 STATNM,STATIS
CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
INTEGER OPTN,MOTN,BSCFIL,VACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
2 FBNUMB,PTNUMB,ORGOPTN
REAL KG

COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELM,NEBLA,KPITCH,KROLL,KYAWL,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELM,NEBLA,KPITCH,KROLL,KYAWL,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /INDEX/ LPFIDX,LPMIDX,LRMIDX,LSDIDX,SVIDX,LSVIDX
INTEGER LPFIDX,LRMIDX,LSVIDX
REAL PFIDX(235),RMIDX(183),SVIDX(3)

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL
INTEGER SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL

COMMON /LOADS/ NLOADS,SWGHT(25),SMASS(25),XLDSRN(10),XLDSPT(25),
2 LSTATN(25)

COMMON /PHYSICO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

```

```

COMMON /RESPN/ NRESP,IPOINT(182),IMOTN(182),ITYPE(182),
2 ILIN(182),ISYM(182)
LOGICAL ILIN,ISYM

COMMON /SEVERE/ NRSIND,RSINDEX,NSWIND,SWINDEX,RSVTOE,RV,RH
REAL RSINDEX(14),SWINDEX(5),RSVTOE(402)
INTEGER RV(13),RH(13)

COMMON /SMPSYS/ FIS,AS,SIS,SOS,SDS,HALOS,DEV,PRN,SMPPS,SMPJS,
2 SMPOS,SMPDS,SHPTYPS,SHIPS,VARS,CYCLS,TITLES,OPTION,LSIS,LSOS,
2 LSDS,LHALOS,LDEV,LPRN,LSMPPS,LSMPIS,LSMPOS,LSMPDS,LSHPTYPS,
2 LSHIPS,LTITLES
CHARACTER*160 AS
CHARACTER*80 FIS,SIS,SOS,SDS,TITLES
CHARACTER*20 HALOS,DEV,PRN,SMPPS,SMPJS,SMPOS,SMPDS,SHPTYPS
CHARACTER SHIPS*6,VARS*2,CYCLS*2
INTEGER*2 OPTION

DIMENSION XID(911),YID(182,5)
DIMENSION IMODL(4),LSVRSP(13)
CHARACTER*4 RTITLE(2),RTYPE(3),RUNIT(3),RSPNME(2,13)
CHARACTER*1 BLANK,BT(80)
CHARACTER*2 AC(2),AT,AVK
CHARACTER*10 PARS1
CHARACTER*110 PARS,SEA
CHARACTER*100 PARS2
DIMENSION HDNG(24)
DIMENSION DATA(432),SPINDEX(9),TOINDEX(9),RMS(8,24),ROLL(13,64,4),
2 ELM(4,8),RMSTBL(25,8,8),TOETBL(25,8,8),TEMRMS(13),TEMTOE(13)
DIMENSION INDXRL(25),INDXHD(25),HEADNG(26)
LOGICAL LINEAR,SYMMET
INTEGER TOETBL,HEADNG,TEMTOE
EQUIVALENCE (IPOINT,YID),(XID,MRESP)
CHARACTER*4 METER,MET,FT,ACOND(3,2),BS,SUNIT

DATA METER,MET,FT /'METE','M','FT,'/
DATA LSVRSP /3,5,2,4,6,9,8,9,8,9,8,9,8/
DATA RSPNME /'HEAV','E','PITC','H','SWA','Y','ROL','L',
2 'YA','W','P1VA','C','P1LA','C','P2VA','C','P2LA','C',
2 'P3VA','C','P3LA','C','P4VA','C','P4LA','C'/
DATA INDXRL /1,2,3,4,5,6,7,8,9,10,11,12,13,12,11,10,9,8,7,6,5,4,
2 3,2,1/
DATA INDXHD /13,12,11,10,9,8,7,6,5,4,3,2,1,24,23,22,21,20,19,18,
2 17,16,15,14,13/
DATA HEADNG /0,15,30,45,60,75,90,105,120,135,150,165,180,
2 360,345,330,315,300,285,270,255,240,225,210,195,180/
DATA HDNG /0.,15.,30.,45.,60.,75.,90.,105.,120.,135.,150.,165.,
2 180.,195.,210.,225.,240.,255.,270.,285.,300.,315.,330.,345./
DATA AC /'LC','SC'/
DATA BLANK /'/'/
DATA ACOND /'LONG','CRES','TED ','SHOR','TCRE','STED'/

NSVRSP = 13
NHEAD = 24
NSPIND = NVK + 1
NTOIND = NTMOD + 1
NID = 911

FIS = SDS(1:LSDS) //'RMS'
OPEN (UNIT=RMSFIL,FILE=FIS,STATUS='UNKNOWN',
2 ACCESS='DIRECT',RECL=1750)

*      modified to run on VAX/VMS
*  CDC    CALL READMS (RMSFIL,XID,NID,1)

READ (RMSFIL,REC=1) (XID(I),I=1,432)
READ (RMSFIL,REC=2) (XID(I),I=433,796),L
IF (L .EQ. NID) READ (RMSFIL,REC=3) (XID(I),I=797,911)

*  CDC    L = LENGTH(RMSFIL)

```

```

M = (L-1)/5
IF(L .NE. NID) M = 159

*   M = 159 means RMSFIL was generated by SMP81
*   M = 182 means RMSFIL was generated by SMP84

      K = 1
      DO 750 J=1,M
      DO 750 I=1,M
      K = K + 1
      YID(I,J) = XID(K)
750   CONTINUE
      NRESP = MRESP
      DO 770 IS=1,NSIGWH

*   find most probable period
      SWH = SIGWH(IS)
      IF (PUNITS(1) .NE. METER) SWH = SWH*FTMETR

*   significant wave height ranges below are in meters

*   sea state 1
      IF (SWH .LE. 0.59) PER = 5.0

*   sea state 2
      IF (SWH.GT.0.59 .AND. SWH.LE.1.26) PER = 5.0

*   sea state 3
      IF (SWH.GT.1.26 .AND. SWH.LE.1.73) PER = 7.0

*   sea state 4
      IF (SWH.GT.1.73 .AND. SWH.LE.2.24) PER = 7.0

*   sea state 5
      IF (SWH.GT.2.24 .AND. SWH.LE.3.97) PER = 9.0

*   sea state 6
      IF (SWH.GT.3.97 .AND. SWH.LE.6.34) PER = 11.0

*   sea state 7
      IF (SWH.GT.6.34 .AND. SWH.LE.12.29) PER = 15.0

*   sea state 8
      IF (SWH.GT.12.29 .AND. SWH.LE.18.77) PER = 19.0

*   greater than sea state 8
      IF (SWH .GT. 18.77) PER = 19.0
      IF (PER .LT. TMODAL(1)) PER = TMODAL(1)
      IF (PER .GT. TMODAL(NTMOD)) PER = TMODAL(NTMOD)
      IMODL(IS) = 1
      DO 760 LT=1,NTMOD
      IF (ABS(PER-TMODAL(LT)) .LT. 0.0001) IMODL(IS) = LT
760   CONTINUE
770   CONTINUE
      ISKPSV = 0
      IF (IMOTN(1) .NE. 1) ISKPSV = 1

*   ISKPSV = 0 all motions - output severe motion tables
*   ISKPSV = 1 roll motion only - skip severe motion tables

      IF (ISKPSV .EQ. 1) GO TO 820

```

```

FIS = SDS(1:LSDS) //' .SEV'
OPEN (UNIT=SEVFIL,FILE=FIS,STATUS='UNKNOWN',
2 ACCESS='DIRECT',RECL=1620)

NSVRSP = 5 + 2*NPTLOC
IF (NSVRSP .GT. 13) NSVRSP = 13
CALL SETSEV (NSVRSP,LSVRSP)
NRSIND = NSVRSP + 1
NSWIND = NSIGWH + 1
NRECD = 0
820 CONTINUE

L = - 3
DO 2 I=1,20
L = L + 4
K = L + 3
READ (TITLE(I),5000) (BT(J),J=L,K)
5000 FORMAT (4A1)
2 CONTINUE
L = 0
DO 4 I=1,80
L = L + 1
IF (BT(I) .NE. BLANK) GO TO 6
4 CONTINUE
6 CONTINUE
IF (L.EQ.80 .AND. BT(80).EQ.BLANK) L = 1
M = L + 9
IF (M .GT. 80) M = 80
WRITE (PARS1,5010) (BT(I),I=L,M)
5010 FORMAT (10A1)
WRITE (PARS2,5020) TITLE
5020 FORMAT (20A4,20X)

* write to speed polar data and text files

FIS = SDS(1:LSDS) //' .SPD'
OPEN(SPDFIL,FILE=FIS,ACCESS='DIRECT',STATUS='UNKNOWN',
2 FORM='UNFORMATTED',RECL=768)

FIS = SDS(1:LSDS) //' .SPT'
OPEN(SPTFIL,FILE=FIS,STATUS='UNKNOWN')

WRITE (SPTFIL,5022) PARS1,PARS2
5022 FORMAT(A10/A100)

PRIDIR = 90.
SECDIR = 0.

WRITE (SPTFIL,5023) NVK,NHEAD
5023 FORMAT(2I5)

WRITE (SPTFIL,5024) {VK(IV),IV=1,NVK}
WRITE (SPTFIL,5024) {HDNG(IH),IH=1,NHEAD)
5024 FORMAT(8F10.4)

* loop over longcrested, shortcrested waves

DO 500 IC=1,2
CALL RLITER (SPINDX,TOINDX,NSPIND,NTOIND,DATA,IC,RMS,ROLL)

* change for VAX/VMS version
* CDC CALL STINDX (SEVFIL,RSINDX,NRSIND)
* CDC DO 7 I=1,NRSIND
* CDC RSINDX(I) = 0.
* CDC 7 CONTINUE

* loop over response

DO 400 IR=1,NRESP
JR = 0

```

```

IF (ISKPSV .EQ. 1) GO TO 19
DC 18 LR=1,NSVRSP
IF (IR .NE. LSVRSP(LR)) GO TO 18
JR = LR
GO TO 19
18 CONTINUE
19 CONTINUE
KR = IR + 1
IP = IPOINT(IR)
IM = IMOTN(IR)
IT = ITYPE(IR)
CALL RSTITL (IP,IM,IT,RTITL,RTYPE,RUNIT,PARS)
LINEAR = ILIN(IR)
SYMMET = ISYM(IR)
NPREDH = 13
IF (.NOT. SYMMET) NPREDH = 24
N = 1
IF (.NOT. LINEAR) N = NRANG
NDATA = (2 + 2*N)*NPREDH

* change for VAX/VMS version
* CDC IF (JR.EQ. 0) GO TO 21
* CDC CALL STINDX (SEVFIL,SWINDX,NSWIND)
* CDC DO 8 I=1 NSWIND
* CDC SWINDX(I) = 0.
* CDC 8 CONTINUE
* CDC21 CONTINUE

* loop over significant wave height
DO 300 IS=1,NSIGWH
CON = SIGWH(IS)*STATIS
IF (IM.EQ.15) CON = SIGWH(IS)

* loop over modal wave period
K = 0
DO 200 ITO=1,NTMOD
SWHMAX = .202*TMODAL(ITO)**2
IF (PUNITS(1) .EQ. METER) SWHMAX = SWHMAX*FTMETR

* loop over speed
DO 100 IV=1,NVK
K = K + 1
IF (SIGWH(IS) .GT. SWHMAX) GO TO 100
CALL FETCH (KR,IV,ITO,DATA,RMIDX,SPINDX,TOINDX,NDATA,LRMIDX,
2 NVK,NTMOD,RMSFIL)

* loop over heading
L = 2*NPREDH
DO 10 IA=1,N
DO 10 IH=1,NPREDH
IF (IC .EQ. 1) TEMP = DATA(L+1)
IF (IC .EQ. 2) TEMP = DATA(L+2)
L = L + 2
RMS(IA,IH) = TEMP*CON
10 CONTINUE
N1 = NHEAD + 1
DO 60 IH=1,N1
IF (IH .GT. NPREDH) GO TO 50
LH = INDXHD(IH)
JC = (IH-1)*2 + IC
IF (.NOT. LINEAR) GO TO 20
RMSTBL(LH,ITO,IV) = RMS(1,IH)
GO TO 40
20 KH = INDXRL(IH)
RLCALC = ROLL(KH,K,IS)
IF (RLCALC .GE. RLANG(1)) GO TO 30
RMSTBL(LH,ITO,IV) = RMS(1,IH)
GO TO 40

```

```

30 IF (RLCALC .LE. RLANG(NRANG)) GO TO 35
RMSTBL(LH,ITO,IV) = RMS(NRANG,IH)
GO TO 40
35 CALL SPFIT (RLANG,RMS(1,IH),ELM,NRANG)
CALL SPLVAL (RLANG,NRANG,ELM,RLCALC,RMSTBL(LH,ITO,IV),DUM,IELM)
40 TOETBL(LH,ITO,IV) = DATA(JC) + .5001
GO TO 60
50 JH = INDXRL(IH)
RMSTBL(IH,ITO,IV) = RMSTBL(JH,ITO,IV)
TOETBL(IH,ITO,IV) = TOETBL(JH,ITO,IV)
60 CONTINUE
100 CONTINUE
IF (SIGWH(IS) .GT. SWHMAX) GO TO 200
TOEMIN = 99.0
TOEMAX = 0.0
RMSMIN = RMSTBL(1,ITO,1)
RMSMAX = RMSMIN
DO 120 IV=1,NVK
DO 110 IH=1,NHEAD
TEMP = RMSTBL(IH,ITO,IV)
VTMP = TOETBL(IH,ITO,IV)
IF (VTMP .GT. 99.) VTMP = 99.
IF (TEMP .LT. RMSMIN) RMSMIN = TEMP
IF (VTMP .LT. TOEMIN) TOEMIN = VTMP
IF (VTMP .GT. TOEMAX) TOEMAX = VTMP
IF (TEMP .LT. RMSMAX) GO TO 110
RMSMAX = TEMP
IF (JR .EQ. 0) GO TO 110
IF (ITO .NE. IMODL(IS)) GO TO 110
IF (SYMMET .AND. IH.GT.13) GO TO 110
MXV = IV
MXH = IH
110 CONTINUE
120 CONTINUE
IF (JR .EQ. 0) GO TO 150
IF (ITO .NE. IMODL(IS)) GO TO 150
RSVTOE(1) = MXV
RSVTOE(2) = MXH
IE = 2
DO 130 IV=1,NVK
DO 130 IH=1,NHEAD
IE = IE + 1
RSVTOE(IE) = RMSTBL(IH,ITO,IV)
IE = IE + 1
RSVTOE(IE) = TOETBL(IH,ITO,IV)
130 CONTINUE

*      write to severe motion file

*      change for VAX/VMS version
*      CDC      CALL WRITMS (SEVFIL,RSVTOE,IE,IS)

NRECD = NRECD + 1
WRITE (SEVFIL,REC=NRECD) RSVTOE

150 CONTINUE

*      write to speed polar file

ISIGWH = SIGWH(IS)*100.
IF (ISIGWH .GE. 1000) WRITE (BS,3001) ISIGWH
IF (ISIGWH .LT. 1000) WRITE (BS,3002) ISIGWH
IF (ISIGWH .LT. 100) WRITE (BS,3003) ISIGWH
IF (ISIGWH .LT. 10) WRITE (BS,3004) ISIGWH
3001 FORMAT (I4)
3002 FORMAT (1H0,I3)
3003 FORMAT (2H00,I2)
3004 FORMAT (3H000,I1)
3000 FORMAT (1H0,I1)
3010 FORMAT (I2)
ITMODL = TMODAL(ITO) + .6
IF (ITMODL .LT. 10) WRITE (AT,3000) ITMODL

```

```

IF (ITMODL .GE. 10) WRITE (AT,3010) ITMODL
SUNIT = MET
IF (PUNITS(1) .NE. METER) SUNIT = FT
WRITE (SEA,3020) BS,AT,AC(IC),SIGWH(IS),SUNIT,TMODAL(IT0),
2 (ACOND(I,IC),I=1,3),(STATNM(I),I=1,3)
3020 FORMAT (2HBR,A4,2A2,32H BRETSCHNEIDER SEAWAY - SIGWH =,F6.2,A4,
2 10H TMODAL =,F6.2,7H SEC, ,3A4,4X,3A4,7X)
WRITE (SPTFIL,5025) PARS,SEA
5025 FORMAT(A110)
WRITE (SPTFIL,5026) RMSMIN,RMSMAX,TOEMIN,TOEMAX
5026 FORMAT(4F10.5)
WRITE (SPDFIL) ((RMSTBL(IH,IT0,IV),IV=1,NVK),IH=1,NHEAD)
WRITE (SPDFIL) ((TOETBL(IH,IT0,IV),IV=1,NVK),IH=1,NHEAD)
200 CONTINUE
IF (IT .GT. 1 .AND. VLACPR.EQ.0) GO TO 300

* print RMS/TOE tables

DO 250 IPAGE=1,2
IF (IPAGE.EQ.2 .AND. SYMMET) GO TO 250
WRITE (IPRIN,1000) TITLE
1000 FORMAT (1H1,22X,20A4)
IF (IC .EQ. 1) WRITE (IPRIN,1010)
IF (IC .EQ. 2) WRITE (IPRIN,1020)
1010 FORMAT (/58X,11HLONGCRESTED)
1020 FORMAT (/58X,12HSHORTCRESTED)
IF (PUNITS(1) .NE. METER) WRITE (IPRIN,1030) SIGWH(IS)
1030 FORMAT (45X,25HSIGNIFICANT WAVE HEIGHT =,F6.2,5H FEET)
IF (PUNITS(1) .EQ. METER) WRITE (IPRIN,1031) SIGWH(IS)
1031 FORMAT (45X,25HSIGNIFICANT WAVE HEIGHT =,F6.2,7H METERS)
IF (IP.GT.0 .AND. IM.LE.3) WRITE (IPRIN,1032) (PTNAME(I,IP),
2 I=1,8),XPTLOC(IP),YPTLOC(IP),ZPTLOC(IP)
IF (IP.GT.0 .AND. IM.EQ.15) WRITE (IPRIN,1032) (PTNAME(I,IP),
2 I=1,8),XPTLOC(IP),YPTLOC(IP),ZPTLOC(IP)
1032 FORMAT (/27X,8A4,2X,5HXFP =,F7.2,2X,5HYCL =,F7.2,2X,5HZBL =,F7.2)
IF (IP.GT.0 .AND. IM.EQ.8) WRITE (JPRIN,1033) (FBNAME(I,IP),
2 I=1,5),XPTFBD(IP),YPTFBD(IP),ZPTFBD(IP)
1033 FORMAT (/33X,5A4,2X,5HXFP =,F7.2,2X,5HYCL =,F7.2,2X,5HZBL =,F7.2)
IF (IP.GT.0 .AND. (IM.GE.10 .AND. IM.LE.14)) WRITE (IPRIN,1073)
2 XLDSTN(IP)
1073 FORMAT (/58X,7HSTATION,F5.1)
IF (IM.NE.15) WRITE (IPRIN,1034) RTITL,RTYPE,RUNIT
1034 FORMAT (/54X,2A4,1X,3A4/58X,3A4)
IF (IM.EQ.15) WRITE (IPRIN,1035)
1035 FORMAT(/50X,26HHORIZONTAL FORCE ESTIMATOR/58X,4H (G))
IF (IM.LT.4 .AND. IT.EQ.3) WRITE (IPRIN,1036)
IF (IM.EQ.15) WRITE (IPRIN,1036)
1036 FORMAT (68X,12H(ACC, X 100))
IF (IP.GT.0 .AND. (IM.GE.10 .AND. IM.LE.11)) WRITE (IPRIN,1063)
1063 FORMAT (/67X,14H(FORCE / 100 ))
IF (IP.GT.0 .AND. (IM.GE.12 .AND. IM.LE.14)) WRITE (IPRIN,1065)
1065 FORMAT (/54X,16H(MOMENT / 10000))
IF (IM.EQ.7) WRITE (IPRIN,1038)
1038 FORMAT (57X,14H(FORCE / 1000))
IF (IM.NE.15) WRITE (IPRIN,1040)(STATNM(I),I=1,3)
IF (IM.EQ.15) WRITE (IPRIN,1041)
1040 FORMAT (/40X,3A4,39H VALUE / ENCOUNTERED MODAL PERIOD (TOE))
1041 FORMAT(51X,42HRMS VALUE / ENCOUNTERED MODAL PERIOD (TOE))
IF (IPAGE .EQ. 2) GO TO 225

* starboard headings

1042 WRITE (IPRIN,1042) (HEADNG(IH),IH=1,13)
FORMAT (/58X,29HSHIP HEADING ANGLE IN DEGREES/4X,1HV,2X,2HTO,7X,
2 4HHEAD,47X,9HSTBD BEAM,46X,6HFOLLOW/10X,13(6X,I3))
DO 220 IV=1,NVK
IVK = VK(IV) + .5001
WRITE (AVK,1045) IVK
1045 FORMAT (I2)
WRITE (IPRIN,1050)
1050 FORMAT (iH )
DO 220 IT0=1,NTMOD

```

```

SWHMAX = .202*TMODAL(ITO)**2
IF (PUNITS(1) .EQ. METER) SWHMAX = SWHMAX*FTMETR
IF (SIGWH(IS) .GT. SWHMAX) GO TO 220
IMP = TMODAL(ITO) + .5001
DO 210 IH=1,13
TEMRMS(IH) = RMSTBL(IH,IT0,IV)
IF (IM.EQ.15) TEMRMS(IH) = TEMRMS(IH) * 100
IF (IM.LT.4 .AND. IT.EQ.3)
2 TEMRMS(IH) = TEMRMS(IH) * 100
IF (IM.EQ.7) TEMRMS(IH) = TEMRMS(IH) / 1000.
IF (IP.GT.0 .AND. (IM.GE.10 .AND. IM.LE.11)) TEMRMS(IH) =
2 TEMRMS(IH)/100
IF (IP.GT.0 .AND. (IM.GE.12 .AND. IM.LE.14)) TEMRMS(IH) =
2 TEMRMS(IH)/10000
TEMTOE(IH) = TOETBL(IH,IT0,IV)
IF(TEMTOE(IH) .GT. 99) TEMTOE(IH)=99
210 CONTINUE
WRITE (IPRIN,1052) AVK,IMP,(TEMRMS(IH),TEMTOE(IH),IH=1,13)
1052 FORMAT (3X,A2,2X,I2,3X,13(IX,F5.2,1H/,I2))
AVK = BLANK
220 CONTINUE
GO TO 250

* port headings

226 WRITE (IPRIN,1043) (HEADNG(IH),IH=14,26)
1043 FORMAT (/58X,29HSHIP HEADING ANGLE IN DEGREES/4X,1HV,2X,2HT0,7X,
2 4HHEAD,47X,9HPORT BEAM,46X,6HFOLLOW/10X,13(6X,I3))
DO 240 IV=1,NVK
IVK = VK(IV) + .5001
WRITE (AVK,1045) IVK
WRITE (IPRIN,1060)
DO 240 IT0=1,NTMOD
SWHMAX = .202*TMODAL(IT0)**2
IF (PUNITS(1) .EQ. METER) SWEMAX = SWHMAX*FTMETR
IF (SIGWH(IS) .GT. SWHMAX) GO TO 240
IMP = TMODAL(IT0) + .50C1
LH = 26
DC 230 IH=1,13
LH = LH - 1
TEMRMS(IH) = RMSTBL(LH,IT0,IV)
IF (IM.EQ.15) TEMRMS(IH) = TEMRMS(IH) * 100
IF ((IM.LT.4 .OR. IM.EQ.9) .AND. IT.EQ.3)
2 TEMRMS(IH) = TEMRMS(IH) * 100
IF (IM.EQ.7) TEMRMS(IH) = TEMRMS(IH) / 1000.
IF (IP.GT.0 .AND. (IM.GE.10 .AND. IM.LE.11)) TEMRMS(IH) =
2 TEMRMS(IH)/100
IF (IP.GT.0 .AND. (IM.GE.12 .AND. IM.LE.14)) TEMRMS(IH) =
2 TEMRMS(IH)/10000
TEMTOE(IH) = TOETBL(LH,IT0,IV)
IF(TEMTOE(IH) .GT. 99) TEMTOE(IH)=99
230 CONTINUE
WRITE (IPRIN,1052) AVK,IMP,(TEMRMS(IH),TEMTOE(IH),IH=1,13)
AVK = BLANK
240 CONTINUE
250 CONTINUE
300 CONTINUE

* change for VAX/VMS version
* CDC IF (JR .EQ. 0) GO TO 310
* CDC CALL STINDX (SEVFIL,RSINDEX,NRSIND)
* CDC CALL WRITMS (SEVFIL,SWINDEX,NSWIND,JR)
* CDC310 CONTINUE

IF (IM.EQ.8 .AND. IT.EQ.2) CALL DKWSLM (KR,IC,IM,NPREDH,N,NDATA,
2 DATA,INDXRL,INDXHL,HEADNG,HDNG,LINEAR,SYMMET,SPINDEX,TOINDEX,IP,
2 RMSTBL,TOETBL,RMS,ROLL)

400 CONTINUE

* change for VAX/VMS version
* CDC IF (ISKPSV .EQ. 1) GO TO 410

```

```

* CDC      CALL STINDX (SEVFIL,SVIDX,LVIDX)
* CDC      CALL WRITMS (SEVFIL,RSINDEX,NRSIND,IC)
* CDC410 CONTINUE

500  CONTINUE

CLOSE (UNIT=RMSFIL)
IF (ISKPSV .EQ. 0) CLOSE (UNIT=SEVFIL)
CLOSE (UNIT=SPDFIL)
CLOSE (UNIT=SPTFIL)

IF (ISKPSV .EQ. 0) CALL SEVMOT (NSVRSP,RESPNME,HDNG,IMODL)

RETURN
END

C DECK RMSTOE
SUBROUTINE RMSTOE

* The purpose of the rmstoe segment is to compute the rms, second and
* fourth moments, encounter spectra and associated periods of maximum
* spectral energy for any ship response. The calculations are done
* for unit significant wave height in long and shortcrested seas for
* a series of modal wave periods. The shortcrested calculations are
* performed using a cosine-squared weighting function.
* W.G.MEYERS, DTNSRDC, 100777

COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
2 LRAOPR,ADRPR,ORGOPTN,GMNOM,KG,STATN(25),NSQFST(25),
2 NLEWF(25),HLFBTH(10,25),WTRLNE(10,25),BLEWF(25),TLEWF(25),
2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
2 ZPTFBD(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
2 STATNM,STATIS
CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,FBCODE,
2 FBNUMB,PTNUMB,ORGOPTN
REAL KG

COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

COMMON /GEOM/ X,NFTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAWL,CHEAVE,CPICTH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DRLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DELWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPICTH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /INDEX/ PFIDX,LPFIDX,RMIDX,LRMIDX,SVIDX,LVIDX
INTEGER LPFIDX,LRMIDX,LVIDX
REAL PFIDX(235),RMIDX(183),SVIDX(3)

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL
INTEGER SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL

COMMON /PHYSICO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II

```

```

REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOFGNUS,GNUF,FTMETR,PUNITS(2)

COMMON /RESPN/ NRESP,IPOINT(182),IMOTN(182),ITYPE(182),
2 ILIN(182),ISYM(182)
LOGICAL ILIN,ISYM

COMMON /SMPSYS/ FIS,AS,SIS,SOS,SDS,HALOS,DEV,PRN,SMPPS,SMPIS,
2 SMPOS,SMPDS,SHPTYPS,SHIPS,VARS,CYCLS,TITLES,OPTION,LSIS,LSOS,
2 LSDS,LHALOS,LDEV,LPRN,LSMPPS,LSMPIS,LSMPOS,LSMPDS,LSHPTYPS,
2 LSHIPS,LTITLES
CHARACTER*160 AS
CHARACTER*80 FIS,SIS,SOS,SDS,TITLES
CHARACTER*20 HALOS,DEV,PRN,SMPPS,SMPIS,SMPOS,SMPDS,SHPTYPS
CHARACTER SHIPS*6,VARS*2,CYCLS*2
INTEGER*2 OPTION

COMMON /STATE/ LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL
LOGICAL LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL

DIMENSION WEVN(100),SPINDEX(9),TOINDX(9),DATA(432),AOMGE(30,13),
2 R(30),RAO1(30,8,13),RAO2(30,8,11),KREC(13),B2(35)
INTEGER DELBET
LOGICAL LINEAR,SYMMET
DIMENSION XID(911)
EQUIVALENCE (NRESP,XID)

NID = 911
NWEVN = 100
CALL WEDEFN (NWEVN,WEVN)
DELBET = 15
NLCH = 11
CALL SCB2 (DELBET,B2,PI,NLCH)
NPLANE = 2
NSPIND = NVK + 1
NTOIND = NTMOD + 1

FIS = SDS(1:LSDS) //' .RMS'
OPEN (UNIT=RMSFIL,FILE=FIS,STATUS='UNKNOWN',
2 ACCESS='DIRECT',RECL=1750)

FIS = SDS(1:LSDS) //' .ORG'
OPEN (UNIT=ORGFIL,FILE=FIS,FORM='UNFORMATTED',STATUS='UNKNOWN')

FIS = SDS(1:LSDS) //' .LCO'
IF (LOADS)
2 OPEN (UNIT=LCOFIL,FILE=FIS,FORM='UNFORMATTED',STATUS='UNKNOWN')

* modified to run on VAX/VMS
* CDC CALL WRITMS (RMSFIL,XID,NID,1)

WRITE (RMSFIL,REC=1) (XID(I),I=1,432)
WRITE (RMSFIL,REC=2) (XID(I),I=433,796),NID
WRITE (RMSFIL,REC=3) (XID(I),I=797,911)
NRECD = 3

DO 60 IR=1,NRESP
LINEAR = ILIN(IR)
SYMMET = ISYM(IR)
NPREDH = 13
IF (.NOT. SYMMET) NPREDH = 24
JA = 1
IF (.NOT. LINEAR) JA = 5
N = 1
IF (.NOT. LINEAR) N = NRANG
IF (LOADS) REWIND LCOFIL
REWIND ORGFIL
READ (ORGFIL) TITLE,NVK,NMU,NOMEGA,OMEGA,NRANG,RLANG,VRT,LAT,
2 ADDRES,LPP,BEAM,DRAFT,DISPLM,GM,DELGM,KG,KROLL,LCB,GRAV,RHO,
2 VKDES,VKINC,DBLWL

* define 2-parameter (significant wave height, modal wave period)

```

```

*     Bretschneider sea spectra, for unit significant wave height

      DO 500 IT=1,NTMOD
      CALL BRWVSP (NOMEGA,1.,TMODAL(IT),OMEGA,S(1,IT))
 500  CONTINUE

      IPHS = 0
      NDATA = (2 + N*2)*NPREDH

*     modified for VAX/VMS
*     CDC      CALL STINDX (RMSFIL,SPINDX,NSPIND)
*     CDC      DO 10 I=1,NSPIND
*     CDC      SPINDX(I) = 0.
*     CDC 10  CONTINUE

      DO 50 IV=1,NVK
      NMU = NMU(IV)
      NLCH = NMU - 2
      N1 = NMU/2 - 1
      N2 = NMU/2 + 1
      NBETA = 2*(NMU-1)

      DO 15 IH=1,NMU
      KH = IH - 1
      IF (IH .EQ. 1) KH = 1
      IF (IH .EQ. 13) KH = 11
      CALL RAOPHS (AOMGE(1,IH),RAO1(1,1,IH),DUM,RAO2(1,1,KH),DUM,
 2 KREC(IH),IR,IV,IH,IPHS)
 15  CONTINUE

*     modified for VAX/VMS
*     CDC      CALL STINDX (RMSFIL,TOINDX,NTOIND)
*     CDC      DO 20 I=1,NTOIND
*     CDC      TOINDX(I) = 0.
*     CDC 20  CONTINUE

      DO 40 IT=1,NTMOD

      CALL RMS (KREC,RAO1,RAO2,IT,N,R,B2,NPREDH,NLCH,N1,N2,DATA,
 2 IMOTN(IR),NBETA)

      CALL TOE (KREC,AOMGE,RAO1,RAO2,JA,IT,R,B2,NPREDH,
 2 NLCH,N1,N2,NBETA,DELBET,NWEVN,WEVN,IV,DATA)

*     modified for VAX/VMS
*     CDC      CALL WRITMS (RMSFIL,DATA,NDATA,IT)

      NRECD = NRECD + 1
      WRITE (RMSFIL,REC=NRECD) DATA
 40  CONTINUE

*     CDC      CALL STINDX (RMSFIL,SPINDX,NSPIND)
*     CDC      CALL WRITMS (RMSFIL,TOINDX,NTOIND,IV)

      50  CONTINUE

*     CDC      CALL STINDX (RMSFIL,RMIDX,LRMIDX)
      KR = IR + 1

*     CDC      CALL WRITMS (RMSFIL,SPINDX,NSPIND,KR)
 60  CONTINUE

      CLOSE (UNIT=RMSFIL)
      CLOSE (UNIT=ORGFIL)
      IF (LOADS) CLOSE (UNIT=LCOFIL)

      RETURN
      END

C DECK RPHI2D

```

```

SUBROUTINE RPHI2D (K,PHI2D)

COMMON /CH3D/ ISIGMA,SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI,
2 IMMIN,IMMAX,IMDEL,LMIN,LMAX
REAL SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI(4)
INTEGER ISIGMA,IMMIN,IMMAX,IMDEL,LMIN,LMAX

COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /INDEX/ PFIDX,LPFIDX,RMIDX,LRMIDX,SVIDX,LSVIDX
INTEGER LPFIDX,LRMIDX,LSVIDX
REAL PFIDX(235),RMIDX(183),SVIDX(3)

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL
INTEGER SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL

COMMON /STATE/ LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL
LOGICAL LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL

COMPLEX PHI2D(10,10,4)
REAL DATA(320)

NNODE = NOFSET(K)
NDATP = 0
IF (VRT) NDATP = 16*NNODE
IF (LAT) NDATP = NDATP + 16*NNODE
ISIGMX = NSIGMA - 1
DO 30 ISIGMA=1,ISIGMX
INDEX = (ISIGMA-1)*NSTATN + K

* modified for VAX/VMS
* CDC CALL READMS (POTFIL,DATA,NDATP,INDEX)
READ (POTFIL,REC=INDEX) DATA

NEXT = 1
DO 20 J=1,NNODE
DO 10 I=IMMIN,IMMAX,IMDEL
PHI2D(ISIGMA,J,I) = CMPLX(DATA(NEXT),DATA(NEXT+1))
IF (ISIGMA .EQ. ISIGMX) PHI2D(NSIGMA,J,I) =
2 CMPLX(DATA(NEXT+4),DATA(NEXT+5))
NEXT = NEXT + 8
10 CONTINUE
20 CONTINUE
30 CONTINUE

RETURN
END

C DECK RSOLVE
SUBROUTINE RSOLVE( N, NDIM, A, B, IP )

```

```

*      solution of linear system, A*X = B .

*      INPUT...
*          N = order of matrix.
*          NDIM = declared dimension of array A .
*          A = triangularized matrix obtained from "DECOMP".
*          B = right hand vector.
*          IP = PIVOT vector obtained from "DECOMP".
*          do not use solve if DECOMP has set IP(N) = 0 .

*      OUTPUT...
*          B = solution vector, X .

REAL A, B, T
INTEGER N, NDIM, IP, I, K, KB, KM1, KP1, M, NM1
DIMENSION A(NDIM,NDIM), B(NDIM)
DIMENSION IP(NDIM)

IF (N .EQ. 1) GO TO 1500
NM1 = N - 1
DO 1200 K = 1, NM1
KP1 = K + 1
M = IP(K)
T = B(M)
B(M) = B(K)
B(K) = T
DO 1100 I = KP1, N
B(I) = B(I) + A(I,K)*T
CONTINUE
1100
1200
CONTINUE
DO 1400 KB = 1, NM1
KM1 = N - KB
K = KM1 + 1
B(K) = B(K)/A(K,K)
T = -B(K)
DO 1300 I = 1, KM1
B(I) = B(I) + A(I,K)*T
1300
1400
1500
CONTINUE
CONTINUE
B(1) = B(1)/A(1,1)
99999
CONTINUE

RETURN
END

C DECK RSTITL
SUBROUTINE RSTITL (IP,IM,IT,RTITL,RTYPE,RUNIT,PARS)

COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
2 LRAOPR,ADRPR,ORGOPTN,GMNOM,KG,STATNC(25),NSOFST(25),
2 NLEWF(25),HLEFBTH(10,25),WTRLNE(10,25),BLEWF(25),TLEWF(25),
2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
2 ZPTFBD(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
2 STATNM,STATIS
CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
2 FBNUMB,PTNUMB,ORGOPTN
REAL KG

COMMON /LOADS/ NLOADS,SWGHT(25),SMASS(25),XLDSRN(10),XLDXPT(25),
2 LSTATN(25)

COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMTR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMTR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

CHARACTER*3 PT(10),TT(3),PPM(3),RLT(2)

```

```

CHARACTER*4 METER,MUNIT(3,7),PNTMOT(2,3),LOAD(2,5)
CHARACTER*4 LTYPE(3,2),LUNIT(3,3),TYPE(3,5),RELMOT(2)
CHARACTER*4 ADRES(2),ADRTYP(3),UNIT(3,7),RUNIT(3)
CHARACTER*4 RTITLE(2),RTYPE(3),HFEMOT(2)
CHARACTER*8 PPLM(3),TTLM(3),HFEM,HHFEM
CHARACTER*6 ORGMOT(2,6)
CHARACTER*10 OMOT(3,6),FMOT(3)
CHARACTER*110 PARS

DATA METER /'METER'/
DATA MUNIT /'(M','ETER','S) ','(MET','ERS','SEC)',',(G','DEG)',',(DE','G/SE','C)',',(DEG',
2 ') /'SEC',2) ','(LBS)',/ /
DATA PPLM /'LONG.','LAT.','VERT.'/
DATA HFEM /'HOR'/
DATA HHFEM /'HORZ'/
DATA TTLM /'DISP','VEL','ACC'/
DATA PT /'P1','P2','P3','P4','P5','P6','P7','P8','P9','P10'/
DATA TT /'DSP','VEL','ACC'/
DATA PPM /'LON','LAT','VER'/
DATA OMOT /'SURGE','SURVEL','SURACC','SWAY','SWAVER','SWAACC',
2 ') /'HEAVE','HEAVEL','HEAACC','ROLL','ROLVEL','ROLACC','PITCH',
2 ') /'PITVEL','PITACC','YAW','YAWVEL','YAWACC'/
DATA ORGMOT /'(S','URGE','SWAY','H','EAVE',',
2 ') /'ROLL','P','ITCH','YAW'/
DATA PNTMOT /'LON','GIT','LAT','ERAL','VERT','ICAL'/
DATA HFEMOT /'HOR','IZ'/
DATA FMOT /'FINANG','FINVEL','FINACC'/
DATA LOAD /'H.S','HEAR','V.S','HEAR','T','ORS','V.B',
2 ') /'END','H.B','END'/
DATA LTYPE /'FORC','E','MOME','NT',/ /
DATA LUNIT /'(T','ONS)',',(M','TONS',') ',,(FT',
2 ') /'TON','S') /
DATA TYPE /'DISP','LACE','MENT','VELO','CITY',',',ACCE',
2 ') /'LERA','TION','ANGL','E',',',MOTI,'ON',/ /
DATA RLT /'RLM','RLV'/
DATA RELMOT /'RELA','TIVE'/
DATA ADRES /'A','DDED'/
DATA ADRTYP /'RESI','STAN','CE',/ /
DATA UNIT /'(F','EET'),',(FEET','T/SE','C)',',(G','DEG)',',(DE','G/SE','C)',',(DEG',
2 ') /'SEC',2) ','(LBS)',/ /
RUNIT(1) = UNIT(1,IT)
RUNIT(2) = UNIT(2,IT)
RUNIT(3) = UNIT(3,IT)
IF (PUNITS(1) .EQ. METER) RUNIT(1) = MUNIT(1,IT)
IF (PUNITS(1) .EQ. METER) RUNIT(2) = MUNIT(2,IT)
IF (PUNITS(1) .EQ. METER) RUNIT(3) = MUNIT(3,IT)
JT = IT + 3
IF (IP .GT. 0) GO TO 20
IF (IM .GT. 6) GO TO 10
* origin motions
RTITLE(1) = ORGMOT(1,IM)
RTITLE(2) = ORGMOT(2,IM)
RTYPE(1) = TYPE(1,IT)
RTYPE(2) = TYPE(2,IT)
RTYPE(3) = TYPE(3,IT)
IF (IM.GT.3 .AND. IT.EQ.1) RTYPE(1) = TYPE(1,4)
IF (IM.GT.3 .AND. IT.EQ.1) RTYPE(2) = TYPE(2,4)
IF (IM.GT.3 .AND. IT.EQ.1) RTYPE(3) = TYPE(3,4)
IF (IM .GT. 3) RUNIT(1) = UNIT(1,JT)
IF (IM .GT. 3) RUNIT(2) = UNIT(2,JT)
IF (IM .GT. 3) RUNIT(3) = UNIT(3,JT)
IF (PUNITS(1) .EQ. METER .AND. IM .GT. 3) RUNIT(1) = MUNIT(1,JT)
IF (PUNITS(1) .EQ. METER .AND. IM .GT. 3) RUNIT(2) = MUNIT(2,JT)
IF (PUNITS(1) .EQ. METER .AND. IM .GT. 3) RUNIT(3) = MUNIT(3,JT)
3000 FORMAT (A10,10X,A10,80X)

```

```

        GO TO 50

10 IF (IM .NE. 7) GO TO 30

*     added resistance

    RTITL(1) = ADRES(1)
    RTITL(2) = ADRES(2)
    RTYPE(1) = ADRTYP(1)
    RTYPE(2) = ADRTYP(2)
    RTYPE(3) = ADRTYP(3)
    RUNIT(1) = UNIT(1,7)
    RUNIT(2) = UNIT(2,7)
    RUNIT(3) = UNIT(3,7)
    GO TO 50

20 IF (IM .GT. 3) GO TO 30

*     motions at a point

    RTITL(1) = PNTMOT(1,IM)
    RTITL(2) = PNTMOT(2,IM)
    RTYPE(1) = TYPE(1,IT)
    RTYPE(2) = TYPE(2,IT)
    RTYPE(3) = TYPE(3,IT)
    WRITE (PARS,3010) PPM(IM),TT(IT),PT(IP),PPLM(IM),TTLM(IT),
2 XPTLOC(IP),YPTLOC(IP),ZPTLOC(IP)
3010 FORMAT (3A3,11X,A5,1X,A5,4X,2HAT,4X,5HXFP =,F6.2,3X,5HYCL =,
2 F7.2,3X,5HZBL =,F7.2,28X)
    GO TO 50

30 IF (IM .NE. 8) GO TO 50

*     relative motion

    RTITL(1) = RELMOT(1)
    RTITL(2) = RELMOT(2)
    RTYPE(1) = TYPE(1,IT)
    RTYPE(2) = TYPE(2,IT)
    RTYPE(3) = TYPE(3,IT)
    IF (IT .EQ. 1) RTYPE(1) = TYPE(1,5)
    IF (IT .EQ. 1) RTYPE(2) = TYPE(2,5)
    IF (IT .EQ. 1) RTYPE(3) = TYPE(3,5)
    WRITE (PARS,3020) RLT(IT),PT(IP),RTITL,RTYPE,XPTFBD(IP),
2 YPTFBD(IP),ZPTFBD(IP)
3020 FORMAT (2A3,14X,2A4,1X,3A4,2HAT,4X,5HXFP =,F6.2,3X,5HYCL =,
2 F7.2,3X,5HZBL =,F7.2,22X)

50 IF (IM .NE. 9) GO TO 72

*     anti-roll fins

    RTITL(1) = ''
    RTITL(2) = ' FIN'
    IF (IT .EQ. 1) JT = 4
    IF (IT .GT. 1) JT = IT
    DO 60 I=1,3
60   RTYPE(I) = TYPE(I,JT)
    JT = IT + 3
    DO 70 I=1,3
70   RUNIT(I) = UNIT(I,JT)
    WRITE (PARS,3000) FMOT(IT),FMOT(IT)

72 IF (IM .NE. 15) GO TO 80
    RTITL(1) = HFEMOT(1)
    RTITL(2) = HFEMOT(2)
    RTYPE(1) = TYPE(1,3)
    RTYPE(2) = TYPE(2,3)
    RTYPE(3) = TYPE(3,3)
    RUNIT(1) = UNIT(1,3)
    RUNIT(2) = UNIT(2,3)
    RUNIT(3) = UNIT(3,3)

```

```

      WRITE (PARS,3010) HFEM,TT(3),PT(IP),HHFEM,TTLM(3),
2 XPTLOC(IP),YPTLOC(IP),ZPTLOC(IP)

80 IF (.NOT. (IP.GT.0.AND.(IM.GE.10.AND.IM.LE.14))) GO TO 100

*   loads

      JM = IM - 9
      RTITLE(1) = LOAD(1,JM)
      RTITLE(2) = LOAD(2,JM)
      LT = 1
      IF (IM .GT. 11) LT = 2
      MT = LT
      IF (LT.EQ.2.AND.(PUNITS(1).NE.METER)) MT = 3
      DO 82 I=1,3
      RTYPE(I) = LTYPE(I,LT)
      RUNIT(I) = LUNIT(I,MT)
82  CONTINUE
      IF (JM .EQ. 1) WRITE (PARS,3031) PT(IP),XLDSTN(IP)
      IF (JM .EQ. 2) WRITE (PARS,3032) PT(IP),XLDSTN(IP)
      IF (JM .EQ. 3) WRITE (PARS,3033) PT(IP),XLDSTN(IP)
      IF (JM .EQ. 4) WRITE (PARS,3034) PT(IP),XLDSTN(IP)
      IF (JM .EQ. 5) WRITE (PARS,3035) PT(IP),XLDSTN(IP)
3031 FORMAT(6HHSHEAR,A3,11X,29HHORIZ. SHEAR FORCE AT STATION,F6.2,55X)
3032 FORMAT(6HVSHEAR,A3,11X,29HVERT. SHEAR FORCE AT STATION,F6.2,55X)
3033 FORMAT(4HTMOM, A3,13X,29HTORSIONAL MOMENT AT STATION,F6.2,55X)
3034 FORMAT(4HVMOM, A3,13X,29HVERT. BEND. MOM. AT STATION,F6.2,55X)
3035 FORMAT(4HMMOM, A3,13X,29HHORIZ. BEND. MOM. AT STATION,F6.2,55X)
100 CONTINUE

      RETURN
      END

C DECK RVSLAT
      SUBROUTINE RVSLAT (VCG,MOTLG,MOTL)

      COMPLEX MOTLG(3),MOTL(3)

      MOTL(1) = MOTLG(1) + VCG*MOTLG(2)
      MOTL(2) = MOTLG(2)
      MOTL(3) = MOTLG(3)

      RETURN
      END

C DECK SBEDDY
      SUBROUTINE SBEDDY

      COMMON /APPEND/ NBKSET,NBKSTN(2),BKIMAG(2),BKFS(2),BKAS(2),
2 BKWD(2),BKSTM(10,2),BKH(10,2),BKLNTH,BKWDTH,
2 BKWL(10,2),BKAN(10,2),NSKSET,SKIMAG(2),SKFLS(2),SKALS(2),
2 SKAUS(2),SKHB(2),SKFLWL(2),SKALWL(2),SKAUWL(2),NPDSET,RD1MAG(2),
2 RDRFS(2),RDRAS(2),RDRHB(2),RDRFWL(2),RDRAWL(2),RUTFS(2),RDTAS(2),
2 RDTHB(2),RDTFWL(2),RDTAWL(2),NSBSET,SBIMAG(2),SOBRFS(2),SOBRAS(2),
2 SOBRHB(2),SOBRFW(2),SOBRRAW(2),SIBRFS(2),SIBRAS(2),SIBRHB(2),
2 SIBRFW(2),SIBRAW(2),SBTFS(2),SBTAS(2),SBTHB(2),SBTFWL(2),
2 SBTAWL(2),NFNSET,FNIMAG(2),FNRFS(2),FNRAS(2),
2 FNRHB(2),FNRFWL(2),FNRRAWL(2),FNTFS(2),FNTAS(2),FNTHB(2),
2 FNTFWL(2),FNTAWL(2),NEXPRD,ENRDO(8),ENRDS(8)

      COMMON /CH3D/ ISIGMA,SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI,
2 IMMIN,IMMAX,IMDEL,LMIN,LMAX
      REAL SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI(4)
      INTEGER ISIGMA,IMMIN,IMMAX,IMDEL,LMIN,LMAX

      COMMON /ENVIOR/ VK, NVK, MU, NMU, OMEGA, NOMEWA, SIGMA, NSIGMA, SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
      INTEGER NVK,NMU,NOMEWA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8),
      REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

```

```

COMMON /PHYSICO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

COMMON /RLDBK/ PSUR(25),BMK(25),DK(25),CAK(25),HQ,HSPAN,HMNCHD,
2 HAREA,HXCP,HYCP,HZCP,HGAMMA,HYHAT,HEAR,HLCS,RQ(2),RSPAN(2),
2 RMNCHD(2),RAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),
2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
2 SYCP(2),SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2),BQ(2),
2 BSPAN(2),BMNCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),
2 PQ(2,2),PSPAN(2,2),PMNCHD(2,2),PAREA(2,2),PXCP(2,2),PYCP(2,2),
2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),PLCS(2,2),
2 STADMP(10),SHPDMP(10,8),ENCON,WPHI,TPHI,WMELM(4,9),SFELM(4,9,8),
2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
2 ENWM,ENSF(8,8),ENRE(8),ENPE(8),ENFE(8),ENHE(8),ENBE(8),
2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(8),ENSL(8),ENBL(8),
2 ENSHP(8,8),RELM(4,9),ITS(25),RD(25),EDDY(8,25),RGB(25)
REAL RDBLK(2692)
EQUIVALENCE (PSUR(1),RDBLK(1))

DO 20 IA=1,NRANG
ENPE(IA) = 0
DO 10 IS=1,NSIGMA
SHPDMP(IS,IA) = 0
10 CONTINUE
20 CONTINUE
IF (NSBSET .EQ. 0) GO TO 100
DO 60 K=1,NSBSET
DO 50 L=1,2
IF (L.EQ.2 .AND. SBTHB(K).EQ.0.) GO TO 50
YHAT = SQRT(PYCP(K,L)**2 + PZCP(K,L)**2)
GAMMAE = PGAMMA(K,L) + 1.
ALF = ATAN( ABS( ((PYCP(K,L)/PZCP(K,L)) + TAN(GAMMAE*DEGRAD)) /
2 (1. - (PYCP(K,L)/PZCP(K,L))*TAN(GAMMAE*DEGRAD)) ) )
C = 0.0065 + (PLCS(K,L)**2)/(0.9*PI*PEAR(K,L))
CON = PQ(K,L)*4. / (3.*PI)*RHO*YHAT**3 * PAREA(K,L)*C*SIN(ALF)
DO 40 IA=1,NRANG
DO 30 IS=1,NSIGMA
SHPDMP(IS,IA) = SHPDMP(IS,IA) + (CON*SIGMA(IS)*RANG(IA)) *
2 SIGMA(IS)
30 CONTINUE
40 CONTINUE
50 CONTINUE
60 CONTINUE
DO 70 IA=1,NRANG
CALL SPFIT (SIGMA,SHPDMP(1,IA),PEELM(1,1,IA),NSIGMA)
ENPE(IA) = ENCON*REVAL(PEELM(1,ISIGMA,IA),WTSI)
70 CONTINUE
100 CONTINUE

RETURN
END

```

C DECK SBLIFT
SUBROUTINE SBLIFT

```

COMMON /APPEND/ NBKSET,NBKSTN(2),BKIMAG(2),BKFS(2),BKAS(2),
2 BKWD(2),BKSTN(10,2),BKHB(10,2),BKLNTH,BKWDTH,
2 BKWL(10,2),BKAN(10,2),NSKSET,SKIMAG(2),SKFLS(2),SKALS(2),
2 SKAUS(2),SKHB(2),SKFLWL(2),SKALWL(2),SKAUWL(2),NRDSET,RDIMAG(2),
2 RDRFS(2),RDRAS(2),RDRHB(2),RDRFWL(2),RDRAWL(2),RDTFS(2),RDTAS(2),
2 RDTHB(2),RDTFWL(2),RDTAWL(2),NSBSET,SBIIMAG(2),SOBRFS(2),SOBRAS(2),
2 SOBRHB(2),SOBRFW(2),SOBRAW(2),SIBRFS(2),SIBRAS(2),SIBRH(2),
2 SIBRFW(2),SIBRAW(2),SBTFWS(2),SETAS(2),SBTHB(2),SBTFWL(2),
2 SBTAWL(2),FNIMAG(2),FNRFS(2),FNRAS(2),
2 FNRHB(2),FNRFWL(2),FNRAWL(2),FNTFS(2),FNTAS(2),FNTHB(2),
2 FNTFWL(2),FNTAWL(2),NEXPRD,ENRDO(8),ENRDS(8)

```

```

COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
2 INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8),
REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
3 INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /PHYSICO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

COMMON /RLDBK/ PSUR(25),BMK(25),DK(25),CAK(25),HQ,HSPAN,HMNCHD,
2 HAREA,HXCP,HYCP,HZCP,HGAMMA,HYHAT,HEAR,HLCS,RQ(2),RSPAN(2),
2 RMNCHD(2),RAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),
2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
2 SYCP(2),SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2),BQ(2),
2 BSPAN(2),BMNCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),
2 PQ(2,2),PSPAN(2,2),PMNCHD(2,2),PAREA(2,2),PXCP(2,2),PYCP(2,2),
2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),PLCS(2,2),
2 STADMP(10),SHPDMP(10,8),ENCON,WPHI,TPHI,WMLM(4,9),SFELM(4,9,8),
2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
2 ENWM,ENSF(8,8),ENRE(8),ENPE(8),ENFE(8),ENHE(8),ENBE(8),
2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(8),ENSL(8),ENBL(8),
2 ENSHP(8,8),RELM(4,9),ITS(25),RD(25),EDDY(8,25),RGB(25)
REAL RDBLK(2692)
EQUIVALENCE (PSUR(1),RDBLK(1))

REAL LCS,MCHORD

IF (NSBSET .EQ. 0) GO TO 60
EN = 0
STASPC = LPP/20
DO 50 K=1,NSBSET
DO 40 L=1,2
IF (L.EQ.2 .AND. SBTHB(K).EQ.0.) GO TO 40
IF (L.EQ. 2) GO TO 20

* outer brackets

XRTF = LCB - SOBRFS(K)*STASPC
XRTA = LCB - SOBRAS(K)*STASPC
XTPF = LCB - SBTFS(K)*STASPC
XTPA = LCB - SBTAS(K)*STASPC
YRT = SOBRHB(K)
YTP = SBTHB(K)
ZRT = (SOBRFW(K) + SOBRAW(K))/2 - (DBLWL+VCG)
ZTP = (SBTFWL(K) + SBTAWL(K))/2 - (DBLWL+VCG)
GO TO 30

* inner bracket

20 XRTF = LCB - SIBRFS(K)*STASPC
XRTA = LCB - SIBRAS(K)*STASPC
YRT = SIBRHB(K)

```

```

30  ZRT = (SIBRFW(K) + SIBRAW(K))/2 - (DBLWL+VCG)
    CONTINUE
    RCHORD = XRTF - XRTA
    TCHORD = XTPF - XTPA
    SPAN = SQRT((ZRT-ZTP)**2 + (YTP-YRT)**2)
    Q = 2
    MCHORD = 0.5*((XRTF-XRTA) + (XTPF-XTPA))

    * area
    AREA = SPAN*MCHORD

    * center of pressure
    ZP = 0.5*(ZRT+ZTP)
    YP = 0.5*(YRT + YTP)
    X0 = 0.5*(XRTF + XTPF)
    XCP = X0 - 0.25*MCHORD
    YCP = YP
    ZCP = ZP

    * moment arm
    ARG = (ZRT-ZTP) / SPAN
    GAMMA = - 90
    IF (ARG .LT. 1) GAMMA = - ASIN(ARG)*RADDEG
    IF (L .EQ. 1) GAMMA = - GAMMA
    GAM = GAMMA*DEGRAD
    YHAT = YCP*COS(GAM) + ZCP*SIN(GAM)

    * effective aspect ratio
    EAR = 2*SPAN/MCHORD

    * lift curve slope
    LCS = 2*PI
    PQ(K,L) = Q
    PSPAN(K,L) = SPAN
    PMNCHD(K,L) = MCHORD
    PAREA(K,L) = AREA
    PXCF(K,L) = XCP
    PYCP(K,L) = YCP
    PZCP(K,L) = ZCP
    PGAMMA(K,L) = GAMMA
    PYHAT(K,L) = YHAT
    PEAR(K,L) = EAR
    PLCS(K,L) = LCS
    EN = EN + Q*(RHO/2)*AREA*LCS*YHAT*YHAT*WPHI*ENCON
40  CONTINUE
50  CONTINUE
60  CONTINUE
    DO 70 IV=1,NVK
    ENPL(IV) = 0
    IF (NSBSET .GT. 0) ENPL(IV) = EN*VFS(IV)
70  CONTINUE

    RETURN
END

C DECK SCB2
SUBROUTINE SCB2 (DELHDG,B2,PI,NLCH)

* This routine pre-computes the shortcrested weighting
* constants, B2, for variable spreading angles.
* W.G.MEYERS, DTNSRDC, 072977

    INTEGER DELHDG
    DIMENSION B2(NLCH)

    N = 180/(2*DELHDG)
    CON1 = 1./N

```

```

CON2 = PI/(2*N)
I = - N
DO 10 K=1,NLCH
I = I + 1
COSI = COS(I*CON2)
B2(K) = CON1*COSI*COSI
10 CONTINUE

RETURN
END

C DECK SECT1
SUBROUTINE SECT1

* determines section type (ITSK) and bilge radius (RDK)
* ITSK = 1 bow sections - narrow v or u
* ITSK = 2 full sections
* ITSK = 3 shallow v or u (destroyer stern)
* ITSK = 4 very rounded destroyer midship section - no eddymaking

COMMON /APPEND/ NBKSET,NBKSTN(2),BKIMAG(2),BKFS(2),BKAS(2),
2 BKWD(2),BKSTN(10,2),BKHB(10,2),BKLNTH,BKWDTH,
2 BKWL(10,2),BKAN(10,2),NSKSET,SKIMAG(2),SKFLS(2),SKALS(2),
2 SKAUS(2),SKHB(2),SKFLWL(2),SKALWL(2),SKAUWL(2),NEDSET,RDIMAG(2),
2 RDRFS(2),RDRAS(2),RDRHB(2),RDRFWL(2),RDRAWL(2),RDTFS(2),RDTAS(2),
2 RDTHB(2),RDTFWL(2),RDTAWL(2),NSBSET,SBIMAG(2),SOBRFS(2),SOBRAS(2),
2 SOBRHB(2),SOBRFW(2),SOBRAW(2),SIBRFS(2),SIBRAS(2),SIBRHB(2),
2 SIERFW(2),SIBRAW(2),SBTFS(2),SBTAS(2),SBTHB(2),SBTFWL(2),
2 SBTAWL(2),NFNSET,FNIMAG(2),FNRFS(2),FNFRAS(2),
2 FNRHB(2),FNRFWL(2),FNRAWL(2),FNTFS(2),FNTAS(2),FNTHB(2),
2 FNTFWL(2),FNTAWL(2),NEXPRD,ENRDO(8),ENRDS(8)

COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
2 LRAOPR,ADRPR,ORGOPTN,GMNOM,KG,STATN(25),NSOFST(25),
2 NLEWF(25),HLFBTH(10,25),WTRLNE(10,25),BLEWF(25),TLEWF(25),
2 AREALI(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFB(10),YPTFB(10),
2 ZPTFB(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
2 STATNM,STATIS
CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,FBCODE,
2 FBNUMB,PTNUMB,ORGOPTN
REAL KG

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL
INTEGER SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL

COMMON /RLDBK/ PSUR(25),BMK(25),DK(25),CAK(25),HQ,HSPAN,HMNCHD,
2 HAREA,HXCP,HYCP,HZCP,HGAMMA,HYHAT,HEAR,HLCS,RQ(2),RSPAN(2),
2 RMNCHD(2),RAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),
2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
2 SYCP(2),SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2),BQ(2),
2 BSPAN(2),BMNCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),

```

```

2 PQ(2,2),PSPAN(2,2),PMNCHD(2,2),PAREA(2,2),FXCP(2,2),FYCP(2,2),
2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),PLCS(2,2),
2 STADM(10),SHPDMP(10,8),ENCON,WPHI,TPHI,WMELM(4,9,8),SFELM(4,9,8),
2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
2 ENWN,ENSF(8,8),ENRE(8),ENPE(8),ENFE(8),ENHE(8),ENBE(8),
2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(8),ENSL(8),ENBL(8),
2 ENSHP(8,8),RELM(4,9),ITS(25),RD(25),EDDY(6,25),RGB(25)
2 REAL RDBLK(2692)
EQUIVALENCE (PSUR(1),RDBLK(1))

DIMENSION AA(3,4),AR(10)

M = NSTATN + 1
DO 100 K=1,NSTATN
M = M - 1
ITSK=4
RDK=1.
IF (NOFSET(K) .LT. 2) GO TO 21
NNODES = NOFSET(K)
BLOCAL = BMK(K)
TLOCAL = DK(K)
ORG = TLOCAL - VCG
CAC = CAK(K)
GDB = ABS(ORG)/(2.*BLOCAL)
RMIN=1.E36
NNM=NNODES-1
DO 31 I=2,NNM
DO 32 J=1,3
IDX=I+J-2
AA(J,1)=Y(IDX,K)**2+Z(IDX,K)**2
AA(J,2)=Y(IDX,K)
AA(J,3)=Z(IDX,K)
AA(J,4)=1.0
32 CONTINUE
A=CMINR(1,AA)
B=-CMINR(2,AA)
C=CMINR(3,AA)
D=-CMINR(4,AA)
IF (A .EQ. 0) GO TO 33
DY=Y(I+1,K)-Y(I-1,K)
IF (ABS(DY) .EQ. 0.) GO TO 33
ZT=Z(I-1,K)+(Z(I+1,K)-Z(I-1,K))*(Y(I,K)-Y(I-1,K))/DY
IF (ZT.LE.Z(I,K)) GO TO 33
YC=-B/(2.*A)
ZC=-C/(2.*A)
R=SQRT(ABS(YC*YC+ZC*ZC-D/A))
AR(I) = R
IF (R .LT. RMIN) RMIN=R
33 CONTINUE
31 CONTINUE
RDK=RMIN

* SERE not used (triangular sections)
* IF (BDG.GT.0.8 .AND. BDG.LE.2.25) ITSK = 3
    IF (CAC .GT. 0.55) ITSK = 4
    IF (CAC .GE. 0.95) ITSK = 2
    IF (GDB .GE. 1.2) ITSK = 1
* no eddymaking (TANAKA) for stations with bilgekeels
    IF (NBKSET .EQ. 0) GO TO 40
DO 30 I=1,NBKSET
NBKS = NBKSTN(I)
DO 20 J=1,NBKS
IF (.NOT.(STATN(M).EQ.BKSTN(J,I))) GO TO 20
YBK = BKHB(J,I)
ZBK = BKWL(J,I) - DBLWL
      WRITE (IPRIN,1000) BKSTN(J,I),YBK,ZBK
1000 FORMAT (/3F10.2)

```

```

M1=2
M2=NNM
DO 11 NN=2,NNM
IF (Z(NN,K).LT.ZBK) GO TO 11
M2=NN
M1=NN-1
IF (Z(NN,K).EQ.ZBK) M2=NN+1
GO TO 12
11 CONTINUE
12 CONTINUE

L = NNODES
DO 13 NN=2,NNM
L = L - 1
R = AR(L)
1010 WRITE (IPRIN,1010) Y(L,K),Z(L,K),AR(L)
FORMAT (2F10.2,1PE12.2)
13 CONTINUE

1011 WRITE(IPRIN,1011) M1,M2
FORMAT (' M1, M2 = ',2I5)

* search for minimum radius of the bilge starting from the waterline

RMIN = AR(M2)
L = M2+1
DO 15 NN=M1,M2
L = L - 1
R = AR(L)
IF (R .GT. RMIN) GO TO 17
RMIN = R
15 CONTINUE
17 RDK = RMIN

1020 WRITE (IPRIN,1020) RMIN
FORMAT (8H RMIN = ,1PE12.2)

ITSK = 4
GO TO 21
20 CONTINUE
30 CONTINUE
40 CONTINUE

* SERE used for sections with skegs

IF (NSKSET .EQ. 0) GO TO 60
DO 50 I=1,NSKSET
IF (STATN(M) .LE. SKAUS(I) .AND. STATN(M) .GE. SKFLS(I)) ITSK = 3
50 CONTINUE
60 CONTINUE
21 CONTINUE
RD(K)=RDK
ITS(K)=ITSK
100 CONTINUE

RETURN
END

C DECK SERAB
SUBROUTINE SERAB (K,ROLANG,BLOCAL,TLOCAL,ORG,RD,EDDY,RGB)
EXTERNAL EXP

* calculates eddy-making roll damping data for TANAKA series A and B
* REF- TANAKA, J. ZOSEN KIOKAI, VOL. 109, 1961

RGB = SQRT(ORG*ORG + BLOCAL*BLOCAL) - RD*(SQRT(2.)-1.)
BDG = 2.*BLOCAL/ABS(ORG)
C = FIG56(ROLANG,BDG)*EXP(-FIG7(ROLANG)*RD/ABS(TLOCAL))
C = C*FTWO(K,TLOCAL,RD)
EDDY = C

```

```

RETURN
END

C DECK SERD
SUBROUTINE SERD (K,ROLANG,BLOCAL,TLOCAL,ORG,EDDY,RGB)
EXTERNAL EXP

* calculates eddy-making roll damping data for TANAKA series D
* REF- TANAKA, J. ZOSEN KIOKAI, VOL. 109, 1961

RGB = ABS(ORG)
IF (BLOCAL .LE. 0.) C = 0.63
IF (BLOCAL .LE. 0.) GO TO 10
GDB = RGB/(2.*BLOCAL)
REQ = FIG10(GDB)*BLOCAL
BDG = 1./GDB
C = FIG56(ROLANG,BDG)*EXP(-FIG7(ROLANG)*REQ/ABS(TLOCAL))
10 CONTINUE
C = C*FTWO(K,TLOCAL,REQ)
EDDY = C

RETURN
END

C DECK SERE
SUBROUTINE SERE (BLOCAL,ORG,EDDY,RGB)

* calculates eddy-making roll damping data for TANAKA series E
* REF- TANAKA, J. ZOSEN KIOKAI, VOL. 109, 1961

RGB = ABS(ORG)
BDG = 2.*BLOCAL/ABS(ORG)
C = FIG11(BDG)
EDDY = C

RETURN
END

C DECK SETSEV
SUBROUTINE SETSEV (NSVRSP,LSVRSP)

COMMON /RESPN/ NRESP,IPOINT(182),IMOTN(182),ITYPE(182),
2 ILIN(182),ISYM(182)
LOGICAL ILIN,ISYM

DIMENSION LSVRSP(NSVRSP)

DO 160 LR=1,NSVRSP
DO 140 IR=1,NRESP
IP = IPOINT(IR)
IM = IMOTN(IR)
IT = ITYPE(IR)
GO TO (10,20,30,40,50,60,70,80,90,100,110,120,130),LR

10 IF (.NOT. (IP.EQ.0 .AND. IM.EQ.3 .AND. IT.EQ.1)) GO TO 140
* heave
    GO TO 150

20 IF (.NOT. (IP.EQ.0 .AND. IM.EQ.5 .AND. IT.EQ.1)) GO TO 140
* pitch
    GO TO 150

30 IF (.NOT. (IP.EQ.0 .AND. IM.EQ.2 .AND. IT.EQ.1)) GO TO 140
* sway
    GO TO 150

40 IF (.NOT. (IP.EQ.0 .AND. IM.EQ.4 .AND. IT.EQ.1)) GO TO 140

```

```

*    roll
      GO TO 150
50     IF (.NOT. (IP.EQ.0 .AND. IM.EQ.6 .AND. IT.EQ.1)) GO TO 140
*
*    yaw
      GO TO 150
60     IF (.NOT. (IP.EQ.1 .AND. IM.EQ.3 .AND. IT.EQ.3)) GO TO 140
*
*    vertical acceleration at point 1 (p1)
      GO TO 150
70     IF (.NOT. (IP.EQ.1 .AND. IM.EQ.2 .AND. IT.EQ.3)) GO TO 140
*
*    lateral acceleration at point 1 (p1)
      GO TO 150
80     IF (.NOT. (IP.EQ.2 .AND. IM.EQ.3 .AND. IT.EQ.3)) GO TO 140
*
*    vertical acceleration at point 2 (p2)
      GO TO 150
90     IF (.NOT. (IP.EQ.2 .AND. IM.EQ.2 .AND. IT.EQ.3)) GO TO 140
*
*    lateral acceleration at point 2 (p2)
      GO TO 150
100    IF (.NOT. (IP.EQ.3 .AND. IM.EQ.3 .AND. IT.EQ.3)) GO TO 140
*
*    vertical acceleration at point 3 (p3)
      GO TO 150
110    IF (.NOT. (IP.EQ.3 .AND. IM.EQ.2 .AND. IT.EQ.3)) GO TO 140
*
*    lateral acceleration at point 3 (p3)
      GO TO 150
120    IF (.NOT. (IP.EQ.4 .AND. IM.EQ.3 .AND. IT.EQ.3)) GO TO 140
*
*    vertical acceleration at point 4 (p4)
      GO TO 150
130    IF (.NOT. (IP.EQ.4 .AND. IM.EQ.2 .AND. IT.EQ.3)) GO TO 140
*
*    lateral acceleration at point 4 (p4)
      GO TO 150
140    CONTINUE
150    LSVRSP(LR) = IR
160    CONTINUE
      RETURN
      END

C DECK SEVMOT
SUBROUTINE SEVMOT (NSVRSP,RSPNME,HDNG,IMODL)

COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
2 LRAOPR,ADRPR,ORGOPTN,GMNOM,KG,STATN(25),NSOFST(25),
2 NLEWF(25),HLFBTH(10,25),WTRLNE(10,25),BLEWF(25),TLEWF(25),
2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
2 ZPTFBD(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
2 STATNM,STATIS
CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
2 FBNUMB,PTNUMB,ORGOPTN
REAL KG

COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMUDAL,NTMUD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS

```

```

INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELG,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREED,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELG,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /INDEX/ PFIDX,LPFIDX,RMIDX,LRMIDX,SVIDX,LSVIDX
INTEGER LPFIDX,LRMIDX,LSVIDX
REAL PFIDX(235),RMIDX(183),SVIDX(3)

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEIFL
INTEGER SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEIFL

COMMON /PHYSICO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

COMMON /SEVERE/ NRSIND,RSINDEX,NSWIND,SWINDEX,RSVTOE,RV,RH
REAL RSINDEX(14),SWINDEX(5),RSVTOE(402)
INTEGER RV(13),RH(13)

COMMON /SMPSYS/ FIS,AS,SIS,SOS,SDS,HALOS,DEV,PRN,SMPPS,SMPIS,
2 SMPOS,SMPDS,SHPTYPS,SHIPS,VARS,CYCLS,TITLES,OPTION,LSIS,LSOS,
2 LSDS,LHALOS,LDEV,LPRN,LSMPPS,LSMPI,LSMPOS,LSMPDS,LSHPTYPS,
2 LSHIPS,LTITLES
CHARACTER*160 AS
CHARACTER*80 FIS,SIS,SOS,SDS,TITLES
CHARACTER*20 HALOS,DEV,PRN,SMPPS,SMPIS,SMPOS,SMPDS,SHPTYPS
CHARACTER SHIPS*6,VARS*2,CYCLS*2
INTEGER*2 OPTION

DIMENSION RSV(13,13),TOE(13,13),TEMV(13),TEMH(13),TEMR(13),
2 TEMT(13),LSVRSP(13),HDNG(24),IMODL(4)
CHARACTER*4 RSPNME(2,13)
INTEGER TEMT
CHARACTER*4 METER

DATA METER /'METE'/
DATA LSVRSP /2,4,1,3,5,7,6,9,8,11,10,13,12/

FIS = SDS(1:LSDS) //'SEV'
OPEN (UNIT=SEVFIL,FILE=FIS,STATUS='UNKNOWN',
2 ACCESS='DIRECT',RECL=1620)

NHEAD = 24
N1 = NHEAD + 1
NDATA = 2 + N1*NVK*2
DO 500 IC=1,2
DO 400 IS=1,NSIGWH
LT = IMODL(IS)
DO 300 IR=1,NSVRSP
DO 200 JR=1,NSVRSP
LR = LSVRSP(JR)
INDEX = NSIGWH * NSVRSP * (IC - 1) + NSIGWH * (LR - 1) + IS

```

```

READ (SEVFIL,REC=INDEX) RSVTOE
* CDC    CALL FETCH (IC,LR,IS,RSVTOE,SVIDX,RSINDX,SWINDX,NDATA,LVIDX,
* CDC  2 NSVRSP,NSIGWH,SEVFIL)

      IF (IR .GT. 1) GO TO 10
      RV(JR) = RSVTOE(1) + .001
      RH(JR) = RSVTOE(2)+ .001
10     IF (JR .GT. 1) GO TO 20
      IV = RV(IR)
      IH = RH(IR)
20     IE = 3 + (IH-1)*2 + (IV-1)*NHEAD*2
      RSV(JR,IR) = RSVTOE(IE)
      TOE(JR,IR) = RSVTOE(IE+1)
200    CONTINUE
300    CONTINUE
      WRITE (IPRIN,1000) TITLE
1000   FORMAT (1H1/,28X,20A4,///,48X,28HS E V E R E   M O T I O N ,
2 9HT A B L E)
      IF (IC .EQ. 1) WRITE (IPRIN,1010)
      IF (IC .EQ. 2) WRITE (IPRIN,1020)
1010   FORMAT (//,60X,11HLONGCRESTED)
1020   FORMAT (//,60X,12HSHORTCRESTED)
      IF (PUNITS(1) .NE. METER) WRITE (IPRIN,1030) SIGWH(IS)
      IF (PUNITS(1) .EQ. METER) WRITE (IPRIN,1040) SIGWH(IS)
1030   FORMAT (/,42X,37HSEA STATE: SIGNIFICANT WAVE HEIGHT =
2 ,F6.2,7H FEET )
1040   FORMAT (/,42X,37HSEA STATE: SIGNIFICANT WAVE HEIGHT =
2 ,F6.2,7H METERS)
      WRITE (IPRIN,1050) TMODAL(LT)
1050   FORMAT (54X,19HMODAL WAVE PERIOD =,F4.0,8H SECONDS)
      IF (NSVRSP .EQ. 5) GO TO 60
      NP = NSVRSP - 5
      NP = NP / 2
      WRITE (IPRIN,1025)
1025   FORMAT (//,54X,16HPOINT LOCATIONS:)
      DO 50 IP=1,NP
      WRITE (IPRIN,1026) IP,(PTNAME(I,IP),I=1,8),XPTLOC(IP),
2 YPTLOC(IP),ZPTLOC(IP)
1026   FORMAT (22X,1H#,I1,3H- ,8A4,2X,5HXFP =,F7.2,2X,5HYCL =,F7.2,2X,
2 5HZBL =,F7.2)
50     CONTINUE
60     CONTINUE
      WRITE (IPRIN,1055) (STATNM(I),I=1,3)
1055   FORMAT (/.40X,3A4,39H VALUE / ENOUNTERED MODAL PERIOD (TOE))
      WRITE (IPRIN,1060) ((RSPNME(I,IR),I=1,2),IR=1,NSVRSP)
1060   FORMAT (/,48X,32HMAXIMUM RESPONSES AND CONDITIONS,/,1X,
2 130(1H-),/,14H RESPONSE ,13(4X,A4,A1))
      DO 310 IR=1,NSVRSP
      IV = RV(IR)
      IH = RH(IR)
      TEMV(IR) = VK(IV)
      TEMH(IR) = HDNG(IH)
      TEMR(IR) = RSV(IR,IR)
      IF (IR .GT. 5) TEMR(IR) = TEMR(IR) * 100
      TEMT(IR) = TOE(IR,IR)
      IF (TEMT(IR) .GE. 99) TEMT(IR) = 99
310    CONTINUE
      WRITE (IPRIN,1070) (TEMR(IR),TEMT(IR),IR=1,NSVRSP)
1070   FORMAT (/,14H (MAX.RSV)/TOE,13(1X,F5.2,1H/,I2))
      WRITE (IPRIN,1080) (TEMV(IR),IR=1,NSVRSP)
1080   FORMAT (17H AT SPEED (KNOTS),F6.1,12F9.1)
      WRITE (IPRIN,1090) (TEMH(IR),IR=1,NSVRSP)
1090   FORMAT (17H AT HEADING (DEG),F6.0,12F9.0)
      WRITE (IPRIN,1100) ((RSPNME(I,JR),I=1,2),JR=1,NSVRSP)
1100   FORMAT (/,54X,20HASSOCIATED RESPONSES,/,1X,130(1H-),/,
2 15H MAX SPEED /,15H RESPN. HEADING,3X,A4,A1,12(4X,A4,A1))
      WRITE (IPRIN,1110)
1110   FORMAT (1X)
      DO 330 IR=1,NSVRSP
      IV = RV(IR)
      IH = RH(IR)

```

```

MV = VK(IV) + .001
MH = HDNG(IH) + .001
IF (IR.EQ.6 .OR. IR.EQ.8 .OR. IR.EQ.10 .OR. IR.EQ.12)
2 WRITE (IPRIN,1110)
DO 320 JR=1,NSVRSP
TEMR(JR) = RSV(JR,IR)
IF (JR .GT. 6) TEMR(JR) = TEMR(JR) * 100
TEMT(JR) = TOE(JR,IR)
IF (TEMT(JR) .GE. 99) TEMT(JR) = 99
320 CONTINUE
WRITE (IPRIN,1120) (RSPNME(I,IR),I=1,2),MV,MH,(TEMR(JR),TEMT(JR),
2 JR=1,NSVRSP)
1120 FORMAT (1X,A4,A1,2X,I2,1H/,I3,13(F6.2,1H/,I2))
330 CONTINUE
WRITE (IPRIN,1130)
1130 FORMAT (//,2X,42HNOTES: 1) RESPONSES ARE IN PHYSICAL UNITS:.,,
2 22X,50HHEAVE AND SWAY ARE IN WAVE HEIGHT UNITS; PITCH, ,
2 29HROLL, AND YAW ARE IN DEGREES;.,22X,23HAND THE POINT VERTICAL,
2 53HAND LATERAL ACCELERATIONS ARE IN UNITS OF G-S * 100.)
WRITE (IPRIN,1140)
1140 FORMAT (9X,51H2) POINT LOCATIONS: XFP IS IN STATION NUMBERS;
2 ,37HYCL AND 2BL ARE IN WAVE HEIGHT UNITS.)
WRITE (IPRIN,1150)
1150 FORMAT(9X,52H3) HEADING CONVENTION: 0 DEG=HEAD, 90 DEG=STBD BEAM,
2 ,24H 180 DEG=FOLLOWING SEAS.)
400 CONTINUE
500 CONTINUE

CLOSE (UNIT=SEVFIL)

RETURN
END

C DECK SKFRSP
FUNCTION SKFRSP (WE,LPP,V,SFD)

REAL LPP

SKFRSP = SFD*(1. + 4.1*V/(WE*LPP))

RETURN
END

C DECK SKLIFT
SUBROUTINE SKLIFT

COMMON /APPEND/ NBKSET,NBKSTM(2),BKIMAG(2),BKFS(2),BKAS(2),
2 BKWD(2),BKSTM(10,2),BKHB(10,2),BKLNTM,BKWDTH,
2 BKWL(10,2),BKAN(10,2),NSKSET,SKIMAG(2),SKFLS(2),SKALS(2),
2 SKAUS(2),SKHB(2),SKFLWL(2),SKALWL(2),SKAUWL(2),NRDSET,RDIMAG(2),
2 RDRFS(2),RDRAS(2),RDRHB(2),RDRFWL(2),RDRAWL(2),RDTFS(2),RDTAS(2),
2 RDTHB(2),RDTFWL(2),RDTAWL(2),NSBSET,SBIMAG(2),SOBRFS(2),SOBRAS(2),
2 SOBRHB(2),SOBRFW(2),SOBRAW(2),SIBRFS(2),SIBRAS(2),SIBRHB(2),
2 SIBRFW(2),SIBRAW(2),SBTFS(2),SBTAS(2),SBTHE(2),SBTFWL(2),
2 SBTAWL(2),NFNSET,FNIMAG(2),FNRFS(2),FNRAS(2),
2 FNRHB(2),FNRFWL(2),FNRAWL(2),FNTFS(2),FNTAS(2),FNTHB(2),
2 FNTFWL(2),FNTAWL(2),NEXPRD,ENRDO(8),ENRDS(8)

COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF
1 VCG,GM,DELM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,

```

```

2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

COMMON /RLDBK/ PSUR(25),BMK(25),DK(25),CAK(25),HQ,HSPAN,HMNCHD,
2 BAREA,HXCP,HYCP,HZCP,HGAMMA,HYHAT,HEAR,HLCS,RQ(2),RSFAN(2),
2 RMNCHD(2),RAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),
2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
2 SYCP(2),SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2),BQ(2),
2 BSPAN(2),BMNCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),
2 PQ(2,2),PSPAN(2,2),PMNCHD(2,2),PAREA(2,2),PXCP(2,2),PYCP(2,2),
2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),PLCS(2,2),
2 STADMP(10),SHPDMP(10,8),ENCON,WPHI,TPHI,WMELM(4,9),SFELM(4,9,8),
2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
2 ENWM,ENSF(8,8),ENRE(8),ENPE(8),ENFE(8),ENHE(8),ENBE(8),
2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(8),ENSL(8),ENBL(8),
2 ENSHP(8,8),RELM(4,9),ITS(25),RD(25),EDDY(8,25),RGB(25)
REAL RDBLK(2692)
EQUIVALENCE (PSUR(1),RDBLK(1))

REAL LCS,MCHORD

IF (NSKSET .EQ. 0) GO TO 20
EN = 0
STASPC = LPP/20
DO 10 K=1,NSKSET
XSKF = LCB - SKFLS(K)*STASPC
XSKAU = LCB - SKAUS(K)*STASPC
XSKAL = LCB - SKALS(K)*STASPC
YSKG = SKHB(K)
ZSKF = SKFLWL(K) - (DBLWL+VCG)
ZSKAU = SKAUWL(K) - (DBLWL+VCG)
ZSKAL = SKALWL(K) - (DBLWL+VCG)
Q = SKIMAG(K)
GAMMA = - 90
SPAN = ZSKAU - ZSKAL
MCHORD = (XSKF - XSKAL)/2

* area
AREA = SPAN*MCHORD

* center of pressure
XCP = XSKAL + (XSKF - XSKAL)/3
YCP = YSKG
ZCP = ZSKF + (ZSKAU - ZSKF)/6

* moment arm
GAM = GAMMA*DEGRAD
YHAT = YCP*COS(GAM) + ZCP*SIN(GAM)

* effective aspect ratio
EAR = 2*SPAN/MCHORD

* lift curve slope
LCS = (PI/2)*EAR
SQ(K) = Q
SSPAN(K) = SPAN

```

```

SMNCHD(K) = MCHORD
SAREA(K) = AREA
SXCP(K) = XCP
SYCP(K) = YCP
SZCP(K) = ZCP
SGAMMA(K) = GAMMA
SYHAT(K) = YHAT
SEAR(K) = EAR
SLCS(K) = LCS
EN = EN + Q*(RHO/2)*AREA*LCS*YHAT*YHAT*WPHI*ENCON
10 CONTINUE
20 CONTINUE
DO 30 IV=1,NVK
ENSL(IV) = 0
IF (NSKSET .GT. 0) ENSL(IV) = EN*VFS(IV)
30 CONTINUE

RETURN
END

C DECK SKNFRC
SUBROUTINE SKNFRC

COMMON /CH3D/ ISIGMA,SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI,
2 IMMIN,IMMAX,IMDEL,LMIN,LMAX
REAL SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI(4)
INTEGER ISIGMA,IMMIN,IMMAX,IMDEL,LMIN,LMAX

COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
INTEGER NVK,NMU,OMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(26),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(26),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /PHYSICO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMTR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMTR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

COMMON /RLDBK/ PSUR(25),BMK(25),DK(25),CAK(25),HQ,HSPAN,HMNCHD,
2 HAREA,HXCP,HYCP,HZCP,HGAMMA,HYHAT,HEAR,HLCS,RQ(2),RSPAN(2),
2 RMNCHD(2),RAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),
2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
2 SYCP(2),SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2),BQ(2),
2 BSPAN(2),BMNCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),
2 PQ(2,2),PSPAN(2,2),PMNCHD(2,2),PAREA(2,2),PXCP(2,2),PYCP(2,2),
2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),PLCS(2,2),
2 STADMP(10),SHPDMP(10,8),ENCON,WPHI,TPHI,WMELM(4,9),SFELM(4,9,8),
2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
2 ENWM,ENSF(8,8),ENRE(8),ENPE(8),ENFE(8),ENHE(8),ENBE(8),
2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(8),ENSL(8),ENBL(8),
2 ENSHP(8,8),RELM(4,9),ITS(25),RD(25),EDDY(8,25),RGB(25)
REAL RDBLK(2692)
EQUIVALENCE (PSUR(1),RDBLK(1))

```

```

DATA RNT/3.E5/
DO 10 IA=1,NRANG
DO 10 IS=1,NSIGMA
SHPDMP(IS,IA) = 0
10 CONTINUE
DO 40 K=1,NSTATN
IF (NOFSET(K) .LT. 2) GO TO 40
RS = 1./PI*((0.887+0.145*CAK(K))*(1.7*ABS(DK(K))+CAK(K)*2*BMK(K))
2 + 2.*VCG)
CON = 4./(3.*PI)*RHO*PSUR(K)*RS**3
DO 30 IA=1,NRANG
DO 20 IS=1,NSIGMA
PERE = TPI/SIGMA(IS)
RN = (3.22*(RS*RANG(IA))**2 / (PERE*GNU)) * REYSCL

* laminar flow
CF = 1.328/SQRT(RN)

* turbulent flow
IF (RN .GE. RNT) CF = CF + 0.014*RN**(-0.114)
STADMP(IS) = CON*SIGMA(IS)*RANG(IA)*CF
STADMP(IS) = SIGMA(IS)*STADMP(IS)
SHPDMP(IS,IA) = SHPDMP(IS,IA) + STADMP(IS)
20 CONTINUE
30 CONTINUE
40 CONTINUE
DO 50 IA=1,NRANG
CALL SPFIT (SIGMA,SHPDMP(1,IA),SFELM(1,1,IA),NSIGMA)
ENSFO = ENCON*REVAL (SFELM(1,ISIGMA,IA),WTSI)
DO 45 IV=1,NVK
ENSF(IV,IA) = SKFRSP (WPHI,LPP,VFS(IV),ENSFO)
45 CONTINUE
50 CONTINUE

RETURN
END

C DECK SLENTH
SUBROUTINE SLENTH (AS,K)

CHARACTER*(*) AS
L=LEN(AS)
K=L+1
DO 10 M=1,L
K=K-1
IF (AS(K:K).NE.CHAR(32)) GO TO 20 ! Test for trailing blanks
10 CONTINUE
20 CONTINUE

RETURN
END

C DECK SMP93 - Standard Ship Motion Program (SMP93)
PROGRAM SMP93

* Standard Ship Motion Program (SMP93)
* for Personal Computers

* Operating system MS-DOS Version 4.01
* FORTRAN 77 using Lahey Fortran
* Overlay linking using PLINK86

* Hull plot and Speed Polar/Density plots
* done in separate programs
* using HALO graphics language

```

```

COMMON /APPEND/ NBKSET,NBKSTN(2),BKIMAG(2),BKFS(2),BKAS(2),
2 BKWD(2),BKSTN(10,2),BKHB(10,2),BKLNTB,BKWDTH,
2 BKWL(10,2),BKAN(10,2),NSKSET,SKIMAG(2),SKFLS(2),SKALS(2),
2 SKAUS(2),SKHB(2),SKFLWL(2),SKALWL(2),NRDSET,RDIMAG(2),
2 RDRFS(2),RDRAS(2),RDRHB(2),RDRFWL(2),RDRAWL(2),RDTFS(2),RDTAS(2),
2 RDTHB(2),RDTFWL(2),RDTAWL(2),NSESET,SBIMAG(2),SOBRFS(2),SOBRAS(2)
2 SOBRHB(2),SOBRFW(2),SOBRAW(2),SIBRFS(2),SIBRAS(2),SIBRHB(2),
2 SIBRFW(2),SIBRAW(2),SIBTFS(2),SBTAS(2),SB1HB(2),SBTFWL(2),
2 SBTAWL(2),NFNSET,FNIMAG(2),FNRFWS(2),FNRAS(2),
2 FNRRHB(2),FNRFWL(2),FNRAWL(2),FNTFS(2),FNTAS(2),FNTHB(2),
2 FNTFWL(2),FNTAWL(2),NEXPRD,ENRDO(8),ENRDS(8)

COMMON /CH3D/ ISIGMA,SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI,
2 IMMIN,IMMAX,IMDEL,LMIN,LMAX
REAL SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI(4)
INTEGER ISIGMA,IMMIN,IMMAX,IMDEL,LMIN,LMAX

COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
2 LRAOPR,ADRPR,ORGOPTN,GMNOM,KG,STATN(25),NSOFST(25),
2 NLEWF(25),HLFBTH(10,25),WTRLNE(10,25),BLEWF(25),TLEWF(25),
2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFB(10),YPTFB(10),
2 ZPTFB(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
2 STATNM,STATIS
CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
2 FBNUMB,PTNUMB,ORGOPTN
REAL KG

COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGHW,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8),
REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(6),S(30,6),FRNUM(6),VFS(8)

COMMON /FJNCON/ IACTFN,IFCLCS,FGAIN(8),FK(3),FA(3),FB(3),
2 FCCLCS(8,2)

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FEDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /HULL/ A26

COMMON /INDEX/ PFIDX,LPFIDX,RMIDX,LRMIDX,SVIDX,LSVIDX
INTEGER LPFIDX,LRMIDX,LSVIDX
REAL PFIDX(235),RMIDX(183),SVIDX(3)

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL
INTEGER SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL

COMMON /LOADS/ MLOADS,SWGHT(25),SMASS(25),XLDSTN(10),XLDXPT(25),
2 LSTATN(25)

COMMON /PELEM/ PELEM
COMPLEX PELEM(4,1000)

```

```

COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

COMMON /RDGEO/ BKLEN,WBKMAX,DLBKEL(25),SRBS(25),PHIS(25),CPS(25),
2 BKT(25),RKS(25),SSTR(25)

COMMON /RESPN/ NRESP,IPOINT(182),IMOTN(182),ITYPE(182),
2 ILIN(182),ISYM(182)
LOGICAL ILIN,ISYM

COMMON /RLDBK/ PSUR(25),BMK(25),DK(25),CAK(25),HQ,HSI,AN,HMNCHD,
2 HAREA,HXCP,HYCP,HZCP,HGAMMA,HYHAT,HEAR,HLCS,RQ(2),RSPAN(2),
2 RMNCHD(2),RAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),
2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
2 SYCP(2),SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2),BQ(2),
2 BSPAN(2),BMNCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),
2 PQ(2,2),PSPAN(2,2),PMNCHD(2,2),PAREA(2,2),PXCP(2,2),PYCP(2,2),
2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),PLCS(2,2),
2 STADMP(10),SHPDMP(10,8),ENCON,WPHI,TPHI,WELM(4,8),SFELM(4,9,8),
2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
2 ENWM,ENSF(8,8),ENRE(8),ENPE(8),ENFE(8),ENHE(8),ENBE(8),
2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(8),ENSL(8),ENBL(8),
2 ENSHP(8,8),RELM(4,9),ITS(25),RD(25),EDDY(8,25),RGB(25)
REAL RDBLK(2692)
EQUIVALENCE (PSUR(1),RDBLK(1))

COMMON /SEVERE/ NRSIND,RSINDEX,NSWIND,SWINDEX,RSVTOE,RV,RH
REAL RSINDEX(14),SWINDEX(5),RSVTOE(402)
INTEGER RV(13),RH(13)

COMMON /SMPSYS/ FIS,AS,SIS,SOS,SDS,HALOS,DEV,PRN,SMPPS,SMPIS,
2 SMPPOS,SMPDS,SHPTYPS,SHIPS,VARS,CYCLS,TITLES,OPTION,LSIS,LSOS,
2 LSDS,LHALOS,LDEV,LFRN,LSMPPS,LSMPIS,LSMPOS,LSMPDS,LSHPTYPS,
2 LSHIPS,LTITLES
CHARACTER*160 AS
CHARACTER*80 FIS,SIS,SOS,SDS,TITLES
CHARACTER*20 HALOS,DEV,PRN,SMPPS,SMPIS,SMPPOS,SMPDS,SHPTYPS
CHARACTER SHIPS*6,VARS*2,CYCLS*2
INTEGER*2 OPTION

COMMON /STATE/ LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL
LOGICAL LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL

COMMON /STELEM/ STELEM
COMPLEX STELEM(4,9,250)

COMMON /TELEM/ TELEM
COMPLEX TELEM(4,9,10)

COMMON /TWOD/ YY,ZZ,ENN,ISTA
INTEGER ISTA
REAL YY(10,25),ZZ(10,25),ENN(4,10,25)

COMMON /WGHTS/ WTDL,NORM
REAL WTDL(10,25),NORM(4,10,25)

CHARACTER*20 DS,TS,ES,T1S,T2S

```

* START

* set underflow to zero
* CALL UNDERO (.TRUE.)
CALL UNDFL (.TRUE.)

```

AS='CLS'
CALL SYSTEM (AS)

CALL PRELIM

CALL RDSMPSYS

FIS = SOS(1:LSOS) //''.TEX'
OPEN (TEXFIL,FILE=FIS,FORM='FORMATTED',STATUS='UNKNOWN')

2 AS = '(/19X,"STANDARD SHIP MOTION PROGRAM, SMP93"/25X,'//
      '"FOR PERSONAL COMPUTERS")'
      WRITE (*,AS)
      WRITE (TEXFIL,AS)

AS = '(/28X,"DTRC    CODE 1561")'
      WRITE (*,AS)
      WRITE (TEXFIL,AS)

CALL DATE (DS)
AS = '(/28X,"DATE = ",A20)'
      WRITE (*,AS) DS
      WRITE (TEXFIL,AS) DS

CALL TIME (TS)
T1S=TS
AS = '(/28X,"TIME = ",A8)'
      WRITE (*,AS) TS
      WRITE (TEXFIL,AS) TS

AS = '(/2X,"Running - ")'
      WRITE (*,AS)
      WRITE (TEXFIL,AS)

AS = '(/" CALL INPUT")'
      WRITE (*,AS)
      WRITE (TEXFIL,AS)

CALL INPUT

CALL TIME (T2S)
CALL ELTIME (T1S,T2S)
T1S=T2S

IF (OPTN .EQ. 1) GO TO 10

AS = '(/" CALL REGWAV")'
      WRITE (*,AS)
      WRITE (TEXFIL,AS)

CALL REGWAV

CALL TIME (T2S)
CALL ELTIME (T1S,T2S)
T1S=T2S

AS = '(/" CALL IRGSEA")'
      WRITE (*,AS)
      WRITE (TEXFIL,AS)

CALL IRGSEA

CALL TIME (T2S)
CALL ELTIME (T1S,T2S)
T1S=T2S

AS = '(/" CALL OUTPUT")'
      WRITE (*,AS)
      WRITE (TEXFIL,AS)

CALL OUTPUT

```

```

CALL TIME (T2S)
CALL ELTIME (T1S,T2S)

* QUIT

10 CONTINUE

AS = '(//2X,"Finished ! ")'
WRITE (*,AS)
WRITE (TEXFIL,AS)

CALL TIME (ES)
CALL ELTIME (TS,ES)

CLOSE (UNIT=TEXFIL)
CLOSE (UNIT=IPRIN)

STOP
END

C DECK SOLVE
SUBROUTINE SOLVE (N,COFF,EXC,MOTN,UL,IP,IPRIN)

*   This routine obtains a solution of the lateral or vertical
*   equations of motion.
*   W.G.MEYERS, DTNSRDC, 072977

COMPLEX COFF,EXC,MOTN,UL
INTEGER N,IP
DIMENSION COFF(N,N),EXC(N),MOTN(N),UL(N,N)
DIMENSION IP(N)

CALL CDDCOMP(N,N,COFF,UL,IP)
IF (IP(N) .EQ. 0) WRITE (IPRIN,1000)
1000 FORMAT (42H SOLVE -- PROGRAM STOP. MATRIX SINGULAR.)
IF (IP(N) .EQ. 0) STOP
CALL CSOLVE(N,N,UL,EXC,MOTN,IP)

RETURN
END

C DECK SPFIT
SUBROUTINE SPFIT (X, Y, ELEMS, NPTS)

*   SPFIT created from SPLINE      E N HUBBLE      JUNE 19
*   fits cubic non-parametric spline segments
*   to set of real data points

*   INPUTS
*   X     = array of real independent variables
*   Y     = array of real dependent variables
*   NPTS = number of (X,Y) data points

*   RETURN
*   ELEMS = array of (NPTS-1) segments in following form
*           ( Y(I), D(I), Y(I+1), D(I+1) ) , where
*           D = array of second derivatives at data points

*   arrays A,B,C are mainly sub diag., diagonal, and super diag.
*   D array is the right hand side of matrix equation
*   second derivatives at nodes are placed in D array after solution
*   solution technique is gaussian elimination
*   boundary conditions set by extrapolation of second derivatives

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPFIL,LACFIL,LAFFIL
INTEGER      SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,

```

```

2 SPTFIL,LAFIL,LAEFIL

DIMENSION X(NPTS),Y(NPTS),ELEMS(4,NPTS)
DIMENSION A(100), B(100), C(100), D(100)

N = NPTS
NL1 = N - 1
NL2 = N - 2
DO 50 I=2,N
IF (X(I) .GT. X(I-1)) GO TO 50
WRITE (IPRIN,888) X(I-1),X(I)
GO TO 88888
50 CONTINUE
IF (N .LE. 100) GO TO 100
WRITE (IPRIN,999)
N = 100
100 CONTINUE
IF (N .GT. 2) GO TO 125
D(1) = 0.0
D(2) = 0.0
GO TO 376
125 CONTINUE
IF (N .GT. 3) GO TO 150
YDD = 2.*((X(3)-X(2))*Y(1)+(X(2)-X(1))*Y(3)-(X(3)-X(1))*Y(2))
2 /((X(3)-X(2))*(X(2)-X(1))*(X(3)-X(1)))
D(1) = YDD
D(2) = YDD
D(3) = YDD
GO TO 375
150 CONTINUE
DO 200 I=1,N
A(I) = 0.0
B(I) = 0.0
C(I) = 0.0
D(I) = 0.0
200 CONTINUE
*      set up matrices(a tridiagonal structure)
A(1) = (X(3)-X(2))/(X(3)-X(1))
C(1) = 2.0
B(1) = 1.0 - A(1)
D(1) = 6.0*((Y(3)-Y(2))/(X(3)-X(2))-(Y(2)-Y(1))/(
1 ((X(2)-X(1))/(X(3)-X(1))))
H = X(3) - X(2)
DO 250 I=3,NL1
HP = X(I+1) - X(I)
C(I) = HP / (H+HP)
B(I) = 2.0
A(I) = 1.0 - C(I)
D(I) = 6.0*((Y(I+1)-Y(I))/HP-(Y(I)-Y(I-1))/H)/(HP+H)
H = HP
250 CONTINUE
*      set boundary conditions
C(2) = (X(2)-X(1))/(X(3)-X(2))
A(2) = 1.0
B(2) = -1.0-C(2)
D(2) = 0.0
C(N) = -A(2)*A(1)/B(1) + C(2)
C(N) = (X(N)-X(N-1))/(X(N-1)-X(N-2))
A(N) = -1.0 - C(N)
B(N) = 1.0
D(N) = 0.0
*      solve equations
II = 1
DO 300 I=1,NL2
I1 = I + 1
I2 = I + 2

```

```

AUGH = ABS (B(I))
IF (AUGH .LT. 1.0E-06) GO TO 275
CONST = A(I1) / B(I)
B(I1) = B(I1) - CONST*C(I)
D(I1) = D(I1) - CONST*D(I)
IF (I .NE. NL2) GO TO 300
A(N) = A(N) - C(N)*C(I) / B(I)
D(N) = D(N) - C(N)*D(I) / B(I)
GO TO 300
275 CONTINUE
II = I + 1
D(I) = D(I) / C(I)
D(I1) = D(I1) - B(I1)*D(I)
B(I1) = A(I1)
A(I1) = 0.0
D(I2) = D(I2) - A(I2)*D(I)
A(I2) = 0.0
IF (I .NE. NL2) GO TO 300
A(N) = C(N)
300 CONTINUE
DET = B(NL1)*B(N) - C(NL1)*A(N)
STORE = D(N)
D(N) = (B(NL1)*D(N) - D(NL1)*A(N)) / DET
D(NL1) = (D(NL1)*B(N) - C(NL1)*STORE) / DET
IP = 0
DO 350 I=2,NL2
JI = N - I
IF (JI .EQ. IP) GO TO 350
IF (JI .EQ. II) GO TO 325
D(JI) = (D(JI)-C(JI)*D(JI+1))/B(JI)
GO TO 350
325 CONTINUE
IP = JI-1
STORE = D(JI)
D(JI) = D(IP)
D(IP) = (STORE - C(IP)*D(JI+1))/B(IP)
350 CONTINUE
D(1) = (D(1) - A(1)*D(3) - C(1)*D(2)) / B(1)

*      set up spline segments

375 CONTINUE
DC 400 I=1,NL1
I1 = I + 1
ELEMS(1,I) = Y(I)
ELEMS(2,I) = D(I)
ELEMS(3,I) = Y(I1)
ELEMS(4,I) = D(I1)
400 CONTINUE
99999 CONTINUE

RETURN
88888 CONTINUE

STOP
888 FORMAT ('0 SPFIT -- X VALUES NOT ASCENDING', 2E16.8)
999 FORMAT ('0 SPFIT -- NPTS EXCEEDS 100. ONLY 99 SEGMENTS RETURNED')

END

C DECK SPINT2
SUBROUTINE SPINT2 (SEGS, NSEGS, AREA, NS, TS, NE, TE, IWAY)

* evaluates the integral of a function given as a parametric spline

* INPITS
* SEGS = spline segments generated by SPLNT2
* NSEGS = number of spline segments
* NS = index of segment for start of integration
* TS = t parameter for start of integration
* NE = index of segment for end of integration
* TE = t parameter for end of integration

```

```

*      IWAY = -1 , if integral of y dx is to be evaluated
*      IWAY = 0 , if integral of x dy is to be evaluated
*
*      RETURN
*      AREA = INTEGRAL (AREA UNDER CURVE) FROM (NS+TS) TO (NE+TE)
*
*      COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2      SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2      SPTFIL,LACFIL,LAEFIL
*      INTEGER SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2      SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2      SPTFIL,LACFIL,LAEFIL
*
*      DIMENSION SEGS(8,NSEGS),CC(14),T(2),A(2)
*
*      AREA = 0.0
IF (NS.GE.1 .AND. NS.LE.NSEGS) GO TO 100
WRITE (IPRIN,991) NS
GO TO 99999
100 CONTINUE
IF (NE.GT.NS .AND. NE.LE.NSEGS) GO TO 150
WRITE (IPRIN,992) NE
GO TO 99999
150 CONTINUE
IF (TS.GE.0.0 .AND. TS.LE.1.0) GO TO 200
WRITE (IPRIN,993) TS
GO TO 99999
200 CONTINUE
IF (TE.GE.0.0 .AND. TE.LE.1.0) GO TO 250
WRITE (IPRIN,994) TE
GO TO 99999
250 CONTINUE
IF (IWAY .EQ. 0) GO TO 350
K = 1
J = 2
GO TO 400
350 CONTINUE
K = 2
J = 1
400 CONTINUE
J2 = J + 2
J4 = J + 4
J6 = J + 6
K4 = K + 4
K8 = K + 8
K10 = K + 10
DO 600 I=NS,NE
T(1) = 0.0
T(2) = 1.0
IF (I .EQ. NS) T(1) = TS
IF (I .EQ. NE) T(2) = TE
CALL CUBCO2 (SEGS(1,I), CC)
DD1 = (CC(J)*CC(K8)) / 6.0
DD2 = (CC(J)*CC(K10) + CC(J2)*CC(K8)) / 5.0
DD3 = (CC(J)*CC(K4) + CC(J2)*CC(K10) + CC(J4)*CC(K8)) / 4.0
DD4 = (CC(J2)*CC(K4) + CC(J4)*CC(K10) + CC(J6)*CC(K8)) / 3.0
DD5 = (CC(J4)*CC(K4) + CC(J6)*CC(K10)) / 2.0
DD6 = CC(J6)*CC(K4)
DO 550 L=1,2
IF (T(L) .GT. 0.0) GO TO 450
A(L) = 0.0
GO TO 550
450 CONTINUE
IF (T(L) .LT. 1.0) GO TO 500
A(L) = DD1 + DD2 + DD3 + DD4 + DD5 + DD6
GO TO 550
500 CONTINUE
A(L) = ((((( DD1 * T(L) + DD2) * T(L) + DD3) * T(L) + DD4)
2 * T(L) + DD5) * T(L) + DD6) * T(L)
550 CONTINUE
AREA = AREA + A(2) - A(1)
600 CONTINUE

```

```

99999 CONTINUE
      RETURN
991  FORMAT ('0 SPINT2 -- NS =', I5, ' OUT OF RANGE' )
992  FORMAT ('0 SPINT2 -- NE =', I5, ' OUT OF RANGE' )
993  FORMAT ('0 SPINT2 -- TS =', E12.5, ' OUT OF RANGE' )
994  FORMAT ('0 SPINT2 -- TE =', E12.5, ' OUT OF RANGE' )

      END

C DECK SPINTG
SUBROUTINE SPINTG (XA, XB, X, NPTS, ELEMS, A, CINTG, SINTG)

*      SPINTG created from SUMSPL and SPLFIT
*      evaluates the integral of a real function defined by
*      non-parametric spline segments

*      INPUTS
*      XA      = lower limit of integration
*      XB      = upper limit of integration
*      X       = array of independent variables
*      NPTS   = number of values in x-array
*      ELEMS  = non-parametric spline segments generated by SPFIT
*      A       = constant for specific integral to be evaluated

*      RETURNS
*      CINTG  = INTEGRAL OF F(X) * COS(A*X)
*      SINTG  = INTEGRAL OF F(X) * SIN(A*X)
*      IF A = 0.0 , THEN CINTG = INTEGRAL OF F(X), AND SINTG = 0.

      DIMENSION X(NPTS),ELEMS(4,NPTS)

      CINTG = 0.0
      SINTG = 0.0
      CALL SPLVAL (X, NPTS, ELEMS, XA, YA, SA, IA)
      CALL SPLVAL (X, NFTS, ELEMS, XB, YB, SB, IB)
      A2 = A * A
      A3 = A * A2
      A4 = A * A3
      DO 500 I=IA,IB
      IF (I .GT. IA) GO TO 100
      X1 = XA
      X2 = X(I+1)
      Y1 = YA
      Y2 = ELEMS(3,I)
      S1 = SA
      S2 = ELEMS(4,I)
      GO TO 300
100   CONTINUE
      IF (I .LT. IB) GO TO 200
      X1 = X(I)
      X2 = XB
      Y1 = ELEMS(1,I)
      Y2 = YB
      S1 = ELEMS(2,I)
      S2 = SB
      GO TO 300
200   CONTINUE
      X1 = X(I)
      X2 = X(I+1)
      Y1 = ELEMS(1,I)
      Y2 = ELEMS(3,I)
      S1 = ELEMS(2,I)
      S2 = ELEMS(4,I)
      GO TO 300
300   CONTINUE
      XX = X2 - X1
      IF (A .NE. 0.0) GO TO 400
      SEGINT = (Y2+Y1) * XX / 2. - (S2+S1) * XX**3 / 24.
      CINTG = CINTG + SEGINT
      GO TO 500
400   CONTINUE

```

```

ZAA = (S2-S1) / (XX * 6.)
ZBB = S1 / 2.
ZCC = (Y2-Y1) / XX - (S2 + 2.*S1) * XX / 6.
AXX = A * XX
E = SIN (AXX)
F = COS (AXX)
XX2 = XX * XX
XX3 = XX * XX2
P = (3.*A2*XX2 - 6.) / A4
Q = (A2*XX3 - 6.*XX) / A3
AA1 = F*P + E*Q + 6./A4
AA2 = E*P - F*Q
PP = (2.*XX) / A2
QQ = (A2*XX2 - 2.) / A3
BB1 = F*PP + E*QQ
BB2 = E*PP - F*QQ - 2./A3
XXA = XX / A
CC1 = (F-1.)/A2 + E*XXA
CC2 = E/A2 - F*XXA
DD1 = E/A
DD2 = (1.-F)/A
AX1 = A * X1
VV = COS (AX1)
UU = SIN (AX1)
PPP = (AA1*ZAA + BB1*ZBB + CC1*ZCC + DD1*Y1)
QQQ = (AA2*ZAA + BB2*ZBB + CC2*ZCC + DD2*Y1)
SISEG = UU*PPP + VV*QQQ
CISEG = VV*PPP - UU*QQQ
CINTG = CINTG + CISEG
SINTG = SINTG + SISEG
500 CONTINUE
      RETURN
      END

C DECK SPLNAR
SUBROUTINE SPLNAR (P,NPTS,SPAREA,PSEGS,NS)

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL
      INTEGER   SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL

DIMENSION P(2,10),PSEGS(8,9)
DIMENSION NDI(2),ENDI(2,2)

DATA ZERO,ONE /0.0,1.0/
DATA NDI,ENDI /2*1,4*0.0/

CALL SPLNT2 (PSEGS,P,NPTS,NDI,ENDI)
CALL SPINT2 (PSEGS,NS,SPAREA,1,ZERO,NS,ONE,0)

      RETURN
      END

C DECK SPLNFT
SUBROUTINE SPLNFT

*      routine used to write offsets to HPLFIL for graphics

COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
2 LRAOFR,ADRPR,ORGOPTN,CMNOM,KG,STATN(25),NSOFST(25),
2 NLEWF(25),HLFBTH(10,25),WTLINE(10,25),BLEWF(25),TLEWF(25),
2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
2 ZPTFBD(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
2 STATNM,STATIS
      CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
      INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
2 FBNUMB,PTNUMB,ORGOPTN

```

```

REAL KG

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FRDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL
INTEGER SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL

COMMON /PHYSOO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

COMMON /SMPSSYS/ FIS,AS,SIS,SOS,SDS,HALOS,DEV,PRN,SMPPS,SMPIS,
2 SMPOS,SMPDS,SHPTYPS,SHIPS,VARS,CYCLS,TITLES,OPTION,LSIS,LSOS,
2 LSDS,LHALOS,LDEV,LPRN,LSMPPS,LSMPIS,LSMPOS,LSMPDS,LSHPTYPS,
2 LSHIPS,LTITLES
CHARACTER*160 AS
CHARACTER*80 FIS,SIS,SOS,SDS,TITLES
CHARACTER*20 HALOS,DEV,PRN,SMPPS,SMPIS,SMPOS,SMPDS,SHPTYPS
CHARACTER SHIPS*6,VARS*2,CYCLS*2
INTEGER*2 OPTION

DIMENSION P(2,10),PSEG(8,9),CC(14),AY(900),AZ(900),
2 HFB(10,25),WTR(10,25),NDI(2),ENDI(2,2)
CHARACTER*6 SNAME(6)
CHARACTER*80 ATITL
CHARACTER STSP*30

DATA SNAME /'YFWD','ZFWD','YAFT','ZAFT','HLFBTH','WTRLNE'/
DATA NDI,ENDI /2*1,4*0.0/
```

- DO 30 K=1,NSTATN
- NPTS = NSOFST(K)
- DO 10 I=1,NPTS
- HFB(I,K) = HLFBTH(I,K)
- WTR(I,K) = WTRLNE(I,K)
- 10 CONTINUE
- IF (NPTS.EQ. 1 .AND. STATN(K).GT.10.0) HFB(1,K) = - HFB(1,K)
- NPT = 10 - NPTS
- IF (NPT .EQ. 0) GO TO 30
- DO 20 I=1,NPT
- IPT = I + NPTS
- HFB(IPT,K) = HFB(NPTS,K)
- WTR(IPT,K) = WTR(NPTS,K)
- 20 CONTINUE
- 30 CONTINUE
- DX = LPP/20
- 1000 WRITE(STSP,1000) DX,PUNITS(1),PUNITS(2)
- FORMAT ('STATION SPACING =',F6.2,1X,A4,A2)
- 1010 WRITE(ATITL,1010) TITLE
- 1010 FORMAT (20A4)

* open file for hull offset plotting

```

FIS = SDS(1:LSDS)//'.HPL'
OPEN (UNIT=HPLFIL,FILE=FIS,STATUS='UNKNOWN')

1020 WRITE (HPLFIL,1020) ATITL
      FORMAT (A80)
      WRITE (HPLFIL,1030) STSP
      FORMAT (A30)

      NOS = 10
      L = 0
      KOUNT = 0
40   IK = KOUNT + 1
      DO 100 K=IK,NSTATN
      KOUNT = KOUNT + 1
      NPTS = NSOFST(K)
      IF (NPTS .EQ. 1) GO TO 100
      L = L + 1
      AY(L) = 0.
      AZ(L) = WTR(NOS,K) - DRAFT
      DO 50 J=1,NOS
      IF (STATN(K) .GT. 10.0) HFB(J,K) = - HFB(J,K)
      WTR(J,K) = WTR(J,K) - DRAFT
      P(1,J) = HFB(J,K)
      P(2,J) = WTR(J,K)
50   CONTINUE
      NS = NOS - 1
      CALL SPLNTZ (PSEGS,P,NOS,NDI,ENDI)
      DO 70 J=1,NS
      CALL CUBC02 (PSEGS(1,J),CC)
      NT = 7
      DT = 1./(NT-1)
      DO 60 I=1,NT
      L = L + 1
      T = (I-1)*DT
      T2 = T*T
      T3 = T*T2
      AY(L) = CC(1)*T3 + CC(3)*T2 + CC(5)*T + CC(7)
      AZ(L) = CC(2)*T3 + CC(4)*T2 + CC(6)*T + CC(8)
60   CONTINUE
70   CONTINUE
      IF (STATN(K) .EQ. 10.0) GO TO 110
100  CONTINUE

      WRITE (HPLFIL,1040) SNAME(3),SNAME(4)
      WRITE (HPLFIL,1050) L
      DO 210 I=1,L
      WRITE (HPLFIL,1060) AY(I),AZ(I)
210  CONTINUE
      GO TO 120

110  WRITE (HPLFIL,1040) SNAME(1),SNAME(2)
1040 FORMAT (A6,4X,A6)
      WRITE (HPLFIL,1050) L
1050 FORMAT (215)
      DO 220 I=1,L
      WRITE (HPLFIL,1060) AY(I),AZ(I)
1060 FORMAT (10F7.2)
220  CONTINUE
      L = 0
      GO TO 130

120  WRITE (HPLFIL,1040) SNAME(5),SNAME(6)
      WRITE (HPLFIL,1050) NOS,NSTATN
      DO 230 K=1,NSTATN
      WRITE (HPLFIL,1060) (HFB(I,K),I=1,NOS)
      WRITE (HPLFIL,1060) (WTR(I,K),I=1,NOS)
230  CONTINUE
130  IF (KOUNT .LT. NSTATN) GO TO 40

      CLOSE (UNIT=HPLFIL)

      RETURN

```

```

      END

C DECK SPLNT2
SUBROUTINE SPLNT2 ( SEGS, P, NP, NDI, ENDI )

*     SPLNT2 created from SPLNT ( NAVSEC-N000 ) - A M REED JULY 1976
*     fits cubic parametric spline segments through set of data points

*     INPUTS
*       P      = array of (X,Y) points
*       NP     = number of points
*       NDI(1) = 1, if initial slope not specified at first point
*       NDI(1) = 2, if initial slope is specified at first point
*       NDI(2) = 1, if initial slope not specified at final point
*       NDI(2) = 2, if initial slope is specified at final point
*       ENDI(1,1) = DX/DT at first point -- not required if NDI(1)=1
*       ENDI(2,1) = DY/DT at first point -- not required if NDI(1)=1
*       ENDI(1,2) = DX/DT at final point -- not required if NDI(2)=1
*       ENDI(2,2) = DY/DT at final point -- not required if NDI(2)=1

*     RETURNS
*       SEGS = array of (NP-1) segments in endpoint/tangent form
*              X(I),Y(I),DX(I),DY(I),X(I+1),Y(I+1),DX(I+1),DY(I+1)

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL
INTEGER   SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL

DIMENSION SEGS(8,NP),P(2,NP),NDI(2),ENDI(2,2)
DIMENSION DS(2,70),INDEX(70),R1(70),R2(70),R3(70),R4(70),
2 CS(70),T10(2),T21(2)

      DATA T10 / 1.0, 0.0 /
1      T21 / 2.0, 1.0 /

*     initialize segs array. determine deltas, chord lengths and
*     indices of non-zero length segments.

      M = 1
      N = NP
      N1 = N - 1
      IF (N1 .LE. 69) GO TO 1000
      N1 = 69
      WRITE (IPRIN,999)
1000  CP = 0.0
      DO 1120 J = 1, M1
      INDEX(J) = J
      C = 0.0
      DO 1100 I = 1, 2
      P1 = P(I,J)
      P2 = P(I,J+1)
      DELTA = P2 - P1
      C = C + DELTA*DELTA
      DS(I,M) = 3.0*DELTA
      SEGS(I,J) = P1
      SEGS(I+4,J) = P2
      SEGS(I+2,J) = DELTA
      SEGS(I+6,J) = DELTA
1100  CONTINUE
      IF (C .LE. 0.000001) GO TO 1110
      C = SQRT( C )
      CS(M) = C
      R1(M) = C
      R3(M) = CP
      INDEX(M) = J
      M = M + 1
      CP = C

```

```

1110  CONTINUE
1120  CONTINUE
N = M
M = N - 1
*
*      check for degenerate case (only 2 points)
IF (N .GT. 2) GO TO 1300
*
*      degenerate case.  set single segment tangent vectors.
J = INDEX(1)
C = CS(1)
DO 1240 I = 1, 2
IF (NDI(1) .GT. 1) SEGS(I+2,J) = ENDI(I,1)*C
IF (NDI(2) .GT. 1) SEGS(I+6,J) = ENDI(I,2)*C
1240  CONTINUE
GO TO 99999
1300  CONTINUE
*
*      set end conditions of tri-diagonal matrix
I = NDI(1)
R2(1) = T21(I)
R3(1) = T10(I)
I = NDI(2)
R1(N) = T10(I)
T2 = T21(I)
*
*      solve matrix for tangent vectors
DO 1340 I = 1, 2
R4(1) = DS(I,1)/CS(1)
IF (NDI(1) .GT. 1) R4(1) = ENDI(I,1)
DO 1315 J = 2, M
R = CS(J-1)/CS(J)
R2(J) = 2.0*(CS(J) + CS(J-1))
R4(J) = DS(I,J)*R + DS(I,J-1)/R
1315  CONTINUE
R2(N) = T2
R4(N) = DS(I,M)/CS(M)
IF (NDI(2) .GT. 1) R4(N) = ENDI(I,2)
DO 1330 J = 1, M
R = R1(J+1)/R2(J)
R2(J+1) = R2(J+1) - R3(J)*R
R4(J+1) = R4(J+1) - R4(J)*R
1330  CONTINUE
DN = R4(N)/R2(N)
DO 1335 L = 1, M
J = N - L
K = INDEX(J)
DJ = (R4(J) - R3(J)*DN)/R2(J)
SEGS(I+2,K) = DJ*CS(J)
SEGS(I+6,K) = DN*CS(J)
DN = DJ
1335  CONTINUE
1340  CONTINUE
99999  CONTINUE
*
RETURN
999 FORMAT('0 SPLNT2 -- NP EXCEEDS 70. ONLY 69 SEGMENTS RETURNED.')
END
C DECK SPLVAL
SUBROUTINE SPLVAL (X, NPTS, ELEMS, X0, Y0, S0, IELM)
*
*      SPLVAL created from SPLFIT
*      evaluates a real non-parametric spline
*
*      INPUTS

```

```

*      X      = array of independent variables
*      NPTS   = number of values in x-array
*      ELEMS  = spline segments generated by SPF1T
*      X0     = x-value at which spline is to be evaluated

*      RETURNS
*      Y0     = F(X0) = y-value evaluated at x0
*      S0     = second derivative evaluated at x0
*      IELM   = index of spline segment containing x0

      COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2      SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2      SPTFIL,LACFIL,LAEFIL
      INTEGER   SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2      SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2      SPTFIL,LACFIL,LAEFIL

      DIMENSION X(NPTS),ELEMS(4,NPTS)

      N = NPTS
      IF (X0.GE.X(1) .AND. X0.LE.X(N)) GO TO 100
      WRITE (IPRIN,999) X0
      GO TO 99999
100    CONTINUE
      DO 200 I=2,N
      IF (X0 .GT. X(I)) GO TO 200
      GO TO 300
200    CONTINUE
300    CONTINUE
      I = I - 1
      XX = X(I+1) - X(I)
      X1 = X0 - X(I)
      X2 = X(I+1) - X0
      XX6 = XX * XX / 6.0
      Y1 = ELEMS(1,I)
      Y2 = ELEMS(3,I)
      S1 = ELEMS(2,I)
      S2 = ELEMS(4,I)
      Y0 = (S1 * X2**3 + S2 * X1**3) / (6.0 * XX) +
2      ((Y1 - S1*XX6) * X2 + (Y2 - S2*XX6) * X1) / XX
      S0 = (S1 * X2 + S2 * X1) / XX
      IELM = I

      RETURN

99999  CONTINUE
      STOP

999    FORMAT ('0 SPLVAL -- EXTRAPOLATION NOT ALLOWED. X0 =', E16.8)
      END

C DECK SPPLV2
      SUBROUTINE SPPLV2 (V, P, SEGS, NSEGS, PT, NINT, TINT, INT)

*      SPPLV2 created from LNPLI2 and LNPLI
*      finds intersection between a curve defined by a parametric spline
*      and a plane defined by a point and a direction vector

*      INPUTS
*      P(1)   = X-COORDINATE OF POINT USED TO DEFINE THE PLANE
*      P(2)   = Y-COORDINATE OF POINT USED TO DEFINE THE PLANE
*      V(1)   = X-COMPONENT OF VECTOR PERPENDICULAR TO THE PLANE
*      V(2)   = Y-COMPONENT OF VECTOR PERPENDICULAR TO THE PLANE
*      SEGS   = SPLINE SEGMENTS IN ENDPOINT-TANGENT FORM, FROM SPLNT2
*      NSEGS  = NUMBER OF SPLINE SEGMENTS

*      RETURNS
*      PT(1)  = X-COORDINATE OF THE INTERSECTION
*      PT(2)  = Y-COORDINATE OF THE INTERSECTION
*      NINT   = INDEX OF SEGMENT IN WHICH INTERSECTION LIES

```

```

*      TINT = VALUE OF T PARAMETER AT INTERSECTION
*      INT = 1, IF INTERSECTION FOUND AND WITHIN TOLERANCE
*      INT = 2, IF INTERSECTION NOT WITHIN TOLERANCE
*      INT = 3, IF NO INTERSECTION FOUND
*      INT = 4, IF SEGMENT LIES WITHIN THE PLANE

DIMENSION V(2),P(2),SEGS(8,NSEGS),PT(2),CC(14),U(2)

1 EQUIVALENCE (U1,U(1)), (U2,U(2)), (CC1,CC(1)), (CC2,CC(2)),
  (CC3,CC(3)), (CC4,CC(4)), (CC5,CC(5)), (CC6,CC(6)), (D,DPS)

DATA TOLER, IMAX / 0.001, 10 /

INT=1

*      unitize plane's direction vector
CALL VUNIT2 (U, S, V)

*      determine the segment number n which contains the intersection
DO 140 N=1,NSEGS
DPS=0.0
DPE=0.0
DO 1000 I = 1, 2
DPS = DPS + (SEGS(I,N) - P(I))*U(I)
DPE = DPE + (SEGS(I+4,N) - P(I))*U(I)
1000 CONTINUE

*      check if segment lies within plane. if so, set int and return.
IF ( ABS( DPS ) .GT. TOLER .OR.
1 ABS( DPE ) .GT. TOLER ) GO TO 130
INT=4
GO TO 99999
130 CONTINUE

*      check if dot product changes sign within segment
NSEG=N
IF ( DPS*DPE .LT. 0.0 ) GO TO 200
IF ( DPS*DPE .EQ. 0.0 ) GO TO 145
140 CONTINUE
NSEG=NSEGS
N=1
145 CONTINUE

*      check if intersection occurs at either end of line
T=0.0
DO 1170 J = 1, 5, 4
DIST = 0.0
DO 1150 I = 1, 2
K = I + J - 1
PT(I) = SEGS(K,N)
DIST = DIST + (PT(I)-P(I)) * U(I)
1150 CONTINUE
IF ( ABS(DIST) .LE. TOLER ) GO TO 1440
N = NSEG
T = 1.0
1170 CONTINUE

*      no intersection found. set int and return.
INT=3
GO TO 99999
200 CONTINUE

*      fetch segment polynomial coefficients
CALL CUBCO2 (SEGS(1,N), CC)

```

```

*      determine scalar polynomial coefficients
A = CC1*U1 + CC2*U2
B = CC3*U1 + CC4*U2
C = CC5*U1 + CC6*U2
A3=A*3.0
B2=B*2.0

*      iterate for t at which the scalar polynomial becomes zero
ITER=0
T=DPS/(DPS-DPE)
CONTINUE
300   FT=((A*T+B)*T+C)*T+D
DT=FT/((A3*T+B2)*T+C)
T=T-DT
IF ( ABS( DT ) .LE. 0.0000001 ) GO TO 400
ITER=ITER+1
IF ( ITER .LE. IMAX ) GO TO 300
IF ( ABS( FT ) .GT. TOLER ) INT = 2
400   CONTINUE

*      set intersection coordinates, n and t parameters
DO 1420 I = 1, 2
COORD = ((CC(I)*T + CC(I+2))*T + CC(I+4))*T + CC(I+6)
IF ( ABS( COORD - P(I) ) .LE. TOLER ) COORD = P(I)
PT(I) = COORD
1420   CONTINUE
1440   CONTINUE
NINT=N
TINT=T
99999  CONTINUE

      RETURN
      END

C DECK T2DAMD
SUBROUTINE T2DAMD (K,PHI2D,T2D,T3D)

* calculates added mass and damping forces on a 2-d section given
* the potentials

COMMON /CH3D/ ISIGMA,SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI,
2 IMMIN,IMMAX,IMDEL,LMIN,LMAX
REAL SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI(4)
INTEGER ISIGMA,IMMIN,IMMAX,IMDEL,LMIN,LMAX

COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(28),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /PHYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOP,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,

```

```

1 RHOF,GNUS,GNUF,FTMETR

COMMON /WGHTS/ WTDL,NORM
REAL WTDL(10,25),NORM(4,10,25)

COMPLEX PHI2D(10,10,4),CTEMP,T2D(10,10),T3D(10,10)
DIMENSION IDX(10),JDX(10)
DIMENSION T(25),ELEMS(4,25)

DATA IDX/1,3,5,3,2,4,6,2,2,4/
DATA JDX/1,3,5,5,2,4,6,4,6,6/

NNODES=NOFSET(K)
IF(NNODES.LE.0) RETURN
DO 3 I=1,NSTATN
T(I)=0.0
3 CONTINUE
T(K)=1.0
CALL SPFIT (X,T,ELEMS,NSTATN)
CALL SPINTG (X(1),X(NSTATN),X,NSTATN,ELEMS,0.0,WTL1,DUM)
DO 10 ISIGMA=1,NSIGMA
DO 1 L=LMIN,LMAX
CTEMP = (0.,0.)
I=IDX(L)
IN = I
IF (I .EQ. 5) IN = 3
IF (I .EQ. 6) IN = 2
J=JDX(L)
JP=J
IF (J .EQ. 5) JP=3
IF (J .EQ. 6) JP=2
XFCTR=1.0
IF (I .EQ. 5) XFCTR=-XFCTR*X(K)
IF (I .EQ. 6) XFCTR= XFCTR*X(K)
IF (J .EQ. 5) XFCTR=-XFCTR*X(K)
IF (J .EQ. 6) XFCTR= XFCTR*X(K)
DO 2 M=1,NNODES
CTEMP = CTEMP + WTDL(M,K)*NORM(IN,M,K)*PHI2D(ISIGMA,M,JP)
2 CONTINUE
T2D(ISIGMA,L) = 2.0*III*RHO*SIGMA(ISIGMA)*XFCTR*CTEMP
T3D(ISIGMA,L) = T3D(ISIGMA,L) + WTL1*T2D(ISIGMA,L)
1 CONTINUE
10 CONTINUE

RETURN
END

C DECK T3DAMD
SUBROUTINE T3DAMD

COMMON /CH3D/ ISIGMA,SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI,
2 IMMIN,IMMAX,IMDEL,LMIN,LMAX
REAL SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI(4)
INTEGER ISIGMA,IMMIN,IMMAX,IMDEL,LMIN,LMAX

COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
2 LRAOPR,ADRPR,ORGOPTN,CMNOM,KG,STATN(25),NSOFST(25),
2 NLEWF(25),HLFBTH(10,25),WTRLNE(10,25),BLEWF(25),TLEWF(25),
2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
2 ZPTLOC(10),NBB,FBNUMB(10),FBNAME,XPTFBD(10),YPTFBD(10),
2 ZPTFED(10),FBCODE(10),FBTYPE,RDOT(10),VKDES,FNDES,
2 STATNM,STATIS
CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
2 FBNUMB,PTNUMB,ORGOPTN
REAL KG

COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

```

```

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,VCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /INDEX/ PFIDX,LPFIDX,RMIDX,LRMIDX,SVIDX,LSVIDX
INTEGER LPFIDX,LRMIDX,LSVIDX
REAL PFIDX(235),RMIDX(183),SVIDX(3)

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL
INTEGER SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL

COMMON /PELEM/ PELEM
COMPLEX PELEM(4,1000)

COMMON /STATE/ LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL
LOGICAL LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL

COMMON /STELEM/ STELEM
COMPLEX STELEM(4,9,250)

COMMON /TELEM/ TELEM
COMPLEX TELEM(4,9,10)

COMMON /WGHTS/ WTDL,NORM
REAL WTDL(10,25),NORM(4,10,25)

COMPLEX T3D(10,10),PHI2D(10,10,4)
EQUIVALENCE (PELEM(1,1),T3D(1,1)),(PELEM(1,26),PHI2D(1,1,1))
COMPLEX T2D(10,10)

READ (SCRFIL) WTDL,NORM
BACKSPACE SCRFIL
IMMIN = 1
IF (.NOT. VRT) IMMIN = 2
IMMAX = 4
IF (.NOT. LAT) IMMAX = 3
IMDEL = 2
IF (VRT .AND. LAT) IMDEL = 1
LMIN = 1
IF (.NOT. VRT) LMIN = 5
LMAX = 10
IF (.NOT. LAT) LMAX = 4
DO 20 I=1,10
DO 10 J=1,10
T3D(I,J) = (0.0,0.0)
10 CONTINUE
20 CONTINUE
DO 30 K=1,NSTATN
NPT = NOFSET(K)
IF (NPT .LT. 2) GO TO 30
CALL RPHI2D (K,PHI2D)
CALL T2DAMD (K,PHI2D,T2D,T3D)
M = (K-1)*10
DO 25 L=LMIN,LMAX
M = M + 1
CALL CPFIT (SIGMA.T2D(1,L),STELEM(1,1,M),NSIGMA)
25 CONTINUE
30 CONTINUE

```

```

DO 40 L=LMIN,LMAX
CALL CPPFIT (SIGMA,T3D(1,L),TELEM(1,1,L),NSIGMA)
40 CONTINUE
REWIND COFFIL
WRITE (COFFIL) TELEM
REWIND COFFIL
IF (RLDMPR .GT. 0) CALL AMDPRN (SIGMA,NSIGMA)

RETURN
END

C DECK TANAKA
SUBROUTINE TANAKA

* calculates coefficient C (=EDDY(K)) and RADIUS (=RGB(K))
* for calculating eddy-making roll damping by the method of
* TANAKA, J.ZOSEN KIOKAI, V. 109, 1961

COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8),
REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAWL,KYAWRL,AWF,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAWL,IYAWRL,CHEAVE,CWIDTH,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAWL,KYAWRL,AWF,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CWIDTH,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /RLDBK/ PSUR(25),BMK(25),DK(25),CAK(25),HQ,HSPAN,HMNCHD,
2 HAREA,HXCP,HYCP,HZCP,HGAMMA,HYHAT,HEAR,HLCS,RQ(2),RSPAN(2),
2 RMNCHD(2),HAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),
2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
2 SYCP(2),SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2),BQ(2),
2 BSPAN(2),BMNCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),
2 PQ(2,2),PSpan(2,2),PMNCHD(2,2),PAREA(2,2),PXCP(2,2),PYCP(2,2),
2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),PLCS(2,2),
2 STADMP(10),SHPDMP(10,8),ENCON,WPHI,TPHI,WMELM(4,9,8),SFELM(4,9,8),
2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
2 ENWM,ENSF(8,8),ENRE(8),ENPE(8),ENFE(8),ENHE(8),ENBE(8),
2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(8),ENSL(8),ENBL(8),
2 ENSHP(8,8),RELM(4,9),ITS(25),RD(25),EDDY(8,25),RGB(25)
REAL RDBLK(2692)
EQUIVALENCE (PSUR(1),RDBLK(1))

DO 20 IA=1,NRANG
DO 10 K=1,NSTATN
EDDY(IA,K) = 0
RGB(K) = 0.
IF (NOFSET(K) .LT. 2) GO TO 10
BLOCAL = BMK(K)
TLOCAL = DK(K)
ORG = TLOCAL - VCG
IF (ITS(K) .EQ. 1) CALL SERD (K,RANG(IA),BLOCAL,TLOCAL,ORG,
2 EDDY(IA,K),RGB(K))
IF (ITS(K) .EQ. 2) CALL SERAB (K,RANG(IA),BLOCAL,TLOCAL,ORG,
2 RD(K),EDDY(IA,K),RGB(K))
IF (ITS(K) .NE. 3) GO TO 10

* stations with skegs
ORG = TLOCAL - VCG

```

```

      CALL SERE (BLOCAL,ORG,EDDY(IA,K),RGB(K))
10    CONTINUE
20    CONTINUE

      RETURN
      END

C DECK TEPEAK
      SUBROUTINE TEPEAK (NWEVN,WEVN,ERS,XTOE,TPI)
*
*   this routine obtains the period of max energy of an encounter
*   spectrum.
*   W.G.MEYERS, DTNSRDC, 072877

      DIMENSION WEVN(NWEVN      'NWEVN)
      PEAK = 0.
      XTOE = TPI/WEVN(1)
      DO 10 I=1,NWEVN
      TE = TPI/WEVN(I)
      IF (ERS(I).GT.PEAK) XTOE = TE
      IF (ERS(I).GT.PEAK) PEAK = ERS(I)
10    CONTINUE

      RETURN
      END

C DECK TFNFFIT
      SUBROUTINE TFNFFIT (RLANG,NRANG,RLANS,MOTL,JM,IW,CTFN)

      DIMENSION RLANG(8)
      COMPLEX MOTL(3,30,8),CANS(8),CELM(4,8),CTFN,CDUM

      IF (RLANS .GE. RLANG(1)) GO TO 10
      CTFN = MOTL(JM,IW,1)
      GO TO 40
10    IF (RLANS .LE. RLANG(NRANG)) GO TO 20
      CTFN = MOTL(JM,IW,NRANG)
      GO TO 40
20    DO 30 IA=1,NRANG
      CANS(IA) = MOTL(JM,IW,IA)
30    CONTINUE
      CALL CPFIT (RLANG,CANS,CELM,NRANG)
      CALL CPLVAL (RLANG,NRANG,CELM,RLANS,CTFN,CDUM,IELM)
40    CONTINUE

      RETURN
      END

C DECK TOE
      SUBROUTINE TOE (KREC,AOMGE,RAO1,RAO2,JA,IT,R,B2,NPREDH,NLCH,N1,
2 N2,NBETA,DELBET,NWEVN,WEVN,IV,DATA)
2
      DIMENSION KREC(13),AOMGE(30,13),RAO1(30,8,13),RAO2(30,8,11),
2 R(30),B2(35),WEVN(100),DATA(432),DUM1(30),DUM2(30),ARLC1(100),
2 ARLC2(100),ARLC3(100),RLC(100,24)

      COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
      INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
      REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

      COMMON /PHYSICO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
      COMPLEX II
      CHARACTER*4 PUNITS(2)
      REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

      INTEGER DELBET

      DO 50 IH=1,NMU

```

```

HDNG = (IH-1)*DELBET
I1 = N1 + IH
I2 = N2 - IH
IF (I2 .LE. 0) I2 = I2 + NBETA
IF (KREC(IH) .GT. 0) GO TO 20
DO 10 I=1,NWEVN
10 RLC(I,I1) = 0.
GO TO 50
20 CALL PSPLC (NOMEGA,OMEGA,AOMGE(1,IH),VK(IV),HDNG,DEGRAD,GRAV,
2 VKMETR,DUM1,DUM2,RAO1(1,JA,IH),S(1,IT),R,NWEVN,WEVN,ARLC1,ARLC2,
2 ARLC3,RLC(1,I1))
IF (KREC(IH) .EQ. 2) GO TO 40
DO 30 I=1,NWEVN
30 RLC(I,I2) = RLC(I,I1)
GO TO 50
40 KH = IH - 1
CALL PSPLC (NOMEGA,OMEGA,AOMGE(1,IH),VK(IV),HDNG,DEGRAD,GRAV,
2 VKMETR,DUM1,DUM2,RAO2(1,JA,KH),S(1,IT),R,NWEVN,WEVN,ARLC1,ARLC2,
2 ARLC3,RLC(1,I2))
50 CONTINUE

L = 0
DO 60 IPH=1,NPREDH
CALL PSPSC (NWEVN,WEVN,RLC,NBETA,B2,NLCH,IPH,ARLC1,ARLC2,TOELC,
2 TOESC,TPI)
L = L + 1
DATA(L) = TOELC
L = L + 1
DATA(L) = TOESC
60 CONTINUE

RETURN
END

```

C DECK TRIM
SUBROUTINE TRIM

* This subroutine provides the correction of zero-speed freeboard
* for the sinkage and trim induced by forward speeds. Reference-
* RICHARD C. BISHOP and NATHAN K. BALES, "A SYNTHESIS OF BOW
* WAVE PROFILE AND CHANGE OF LEVEL DATA FOR DESTROYER-TYPE HULLS
* WITH APPLICATION TO COMPUTING MINIMUM REQUIRED FREEBOARDS,"
* DTNSRDC REPORT 78-SPD-811-01, JAN. 1978. The formulae for
* sinkage, Z0, were developed in units of feet. Conversion
* to meters is provided. The formulae for trim, ang, were
* developed in units of degrees. Conversion to radians is made.
* ship speed is in knots. NBB=0 means a ship without a bow dome.

```

COMMON /DATINP/ OPTN,MOTN,BSCFIL,VLACPR,RAOPR,RLDMPR,DISPLMT,
2 LRAOPR,ADRPR,ORGOPTN,GMNOM,KG,STATN(25),NSOFST(25),
2 NLEWF(25),HLFBTH(10,25),WTRLNE(10,25),TLEWF(25),ZPTLOC(10),
2 AREALF(25),NPTLOC,PTNUMB(10),PTNAME,XPTLOC(10),YPTLOC(10),
2 ZPTLGC(10),NBB,FBNUMB(10),FBNAME,XPTFB(10),YPTFB(10),
2 ZPTFB(10),FBCODE(10),FBTYPE,RD0T(10),VKDES,FNDES,
2 STATNM,STATIS
CHARACTER*4 PTNAME(8,10),FBNAME(8,10),STATNM(5),FBTYPE(3,10)
INTEGER OPTN,MOTN,BSCFIL,VLACPR,RAOPR,ADRPR,RLDMPR,FBCODE,
2 FBNUMB,PTNUMB,ORGOPTN
REAL KG

```

```

COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8),
REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

```

```

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS

```

```

REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBODY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),TITLE(20),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /PRYSCO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

REAL LO
CHARACTER*4 METER

DATA METER /'METE'/

CON = 1
IF (PUNITS(1) .EQ. METER) CON = FTMETR
DO 1 I=1,NVK

* speed is FROUDE scaled to LO ship
LO = 480.*CON
VO = SQRT(LO/LPP) * VK(I)
V2 = VO*VO
V3 = V2*VO
IF (NBB .EQ. 0) GO TO 20

* ship with bow dome
Z0 = (.007848*VO + .001321*V2) * CON
ANGO = (.015422*VO - .0021752*V2 + 5.957E-5*V3) * DEGRAD
GO TO 30
CONTINUE

* ship without bow dome
Z0 = (-.005292*VO + .001855*V2) * CON
ANGO = (.0092648*VO - .0015692*V2 + 4.2912E-5*V3) * DEGRAD
30 CONTINUE

* sinkage FROUDE scaled from LO ship to LPP ship.
* sinkage and trim both defined positive.
* freeboard correction = F - SINKAGE + FBDX*TRIM

DO 5 J=1,NFREBD
SNK = Z0 * LPP/LO
TRM = ANGO
FBDZV(I,J) = FBDZ(J) - SNK + FBDX(J)*TRM
5 CONTINUE
1 CONTINUE

RETURN
END

C DECK TRNLAT
SUBROUTINE TRNLAT (VCG,TL,EXCL,TLG,EXCLG)

COMPLEX TL(3,3),EXCL(3),TLG(3,3),EXCLG(3)

TLG(1,1) = TL(1,1)
TLG(1,2) = TL(1,2) + VCG*TL(1,1)
TLG(1,3) = TL(1,3)
TLG(2,1) = TLG(1,2)
TLG(2,2) = TL(2,2) + VCG*(TL(1,2) + TL(2,1) + VCG*TL(1,1))
TLG(2,3) = TL(2,3) + VCG*TL(1,3)
TLG(3,1) = TL(3,1)
TLG(3,2) = TL(3,2) + VCG*TL(3,1)
TLG(3,3) = TL(3,3)

```

```

EXCLG(1) = EXCL(1)
EXCLG(2) = EXCL(2) + VCG*EXCL(1)
EXCLG(3) = EXCL(3)

RETURN
END

C DECK TWODPT
SUBROUTINE TWODPT (KSTA,YSTA,ZSTA,NPT,PHI2D)

* This subroutine provides two-dimensional velocity potentials for
* oscillating cylinders of arbitrary cross section in a free surface
* four velocity potentials associated with the individual modes
* of oscillation, surge, sway, heave, and roll, are obtained which
* are stored in PHI2D (frequency, offset point, mode).

COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8),
REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL
INTEGER SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL

COMMON /PHYSICO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,CNUS,GNUF,FTMETR

COMMON /STATE/ LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL
LOGICAL LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL

COMMON /TWOD/ YY, ZZ, ENN, ISTA
INTEGER ISTA
REAL YY(10,25),ZZ(10,25),ENN(4,10,25)

COMPLEX RHS1(10), RHS2(10), RHS3(10), RHS4(10), Q1(10), Q2(10),
Q3(10), Q4(10), GREENV(10,10), GREENL(10,10), CTV(10,10),
CTL(10,10), UV(10,10), UL(10,10), SIGIM, FAC,
PHI2D(10,10,4)
DIMENSION POTLOG(2,10,10), PTNLOG(2,10,10), CN(10), SN(10)
DIMENSION YS(11), ZS(11), IPV(10), IPL(10)
DIMENSION YSTA(10),ZSTA(10)
DIMENSION SP(10),SQ(10),W1(10),W2(10)
LOGICAL LID

ISTA = KSTA
FACTOR = SQRT(GRAV*LPP)
SQRLG = SQRT(LPP/GRAV)
DO 60 I=1,NSIGMA
SIGMA(I) = SIGMA(I)*SQRLG
60 CONTINUE

```

```

DO 70 J=1,NPT
ENN(4,J,ISTA) = ENN(4,J,ISTA)/LPP
YS(J) = YSTA(J)/LPP
ZS(J) = ZSTA(J)/LPP
YY(J,ISTA) = YY(J,ISTA)/LPP
ZZ(J,ISTA) = ZZ(J,ISTA)/LPP
70 CONTINUE
SQ(1) = 0.
DO 72 N=2,NPT
NM = N - 1
YINT = YS(N) - YS(NM)
ZINT = ZS(N) - ZS(NM)
GIR = SQRT(YINT*YINT+ZINT*ZINT)
SQ(N) = SQ(NM) + GIR
72 CONTINUE
NON = NPT - 1
YINT = YY(1,ISTA) - YS(1)
ZINT = ZZ(1,ISTA) - ZS(1)
GIR = SQRT(YINT*YINT+ZINT*ZINT)
SP(1) = GIR
DO 74 N=2,NON
NM = N - 1
YINT = YY(N,ISTA) - YY(NM,ISTA)
ZINT = ZZ(N,ISTA) - ZZ(NM,ISTA)
GIR = SQRT(YINT*YINT+ZINT*ZINT)
SP(N) = SP(NM) + GIR
74 CONTINUE
DO 76 N=2,NON
NM = N - 1
DEN = SP(N) - SP(NM)
W1(N) = (SP(N) - SQ(N))/DEN
W2(N) = (SQ(N) - SP(NM))/DEN
76 CONTINUE
DEN = SP(2) - SP(1)
W1(1) = (SP(2) - SQ(1))/DEN
W2(1) = (SQ(1) - SP(1))/DEN
NM = NON - 1
DEN = SP(NON) - SP(NM)
W1(NPT) = (SP(NON) - SQ(NPT))/DEN
W2(NPT) = (SQ(NPT) - SP(NM))/DEN

* test for LID

LID = .TRUE.
IF (ABS(YS(NPT)) .LE. 1.0E-5) LID = .FALSE.
NARG = NPT
IF(.NOT.LID) NARG = NPT-1
NZRO = NPT + 1

* below two cards are to introduce one more segment on the free
* surface inside a cross section for removing irregular frequencies.

YS(NZRO) = 0.
ZS(NZRO) = 0.

CALL GRNLOG( YS, ZS, NARG, POTLOG, PTNLOG, CN, SN)
DO 10 K=1,NSIGMA
SIGMA2 = SIGMA(K)**2
SIGIM=III*SIGMA(K)
DO 1 I=1,NON
RHS1(I) = -ENN(1,I,ISTA)*SIGIM
RHS2(I) = -ENN(2,I,ISTA)*SIGIM
RHS3(I) = -ENN(3,I,ISTA)*SIGIM
RHS4(I) = -ENN(4,I,ISTA)*SIGIM
1 CONTINUE

* the following four cards are to impose a rigid wall condition on
* the waterline segment inside the section.

IF(.NOT. LID) GO TO 25
RHS1(NPT) = (0.0, 0.0)
RHS2(NPT) = (0.0, 0.0)

```

```

RHS3(NPT) = {0.0, 0.0}
RHS4(NPT) = {0.0, 0.0}
25 CONTINUE

CALL GRNFRQ( YS, ZS, NARG, SIGMA2, POTLOG, PTNLOG, CN, SN,
              CTV, CTL, GREENV, GREENL)

* for the algebraic equation AX=B, CDCOMP makes an inversion of
* the matrix A, and CSOLVE provides the solution vector X by
* X=(INVERTED A)B

CALL CDCOMP( NARG, 10, CTV, UV, IPV)
IF (IPV(NARG) .EQ. 0) GO TO 17
CALL CSOLVE( NARG, 10, UV, RHS1, Q1, IPV)
CALL CSOLVE( NARG, 10, UV, RHS3, Q3, IPV)
IF (.NOT. LAT) GO TO 20
CALL CDCOMP (NARG, 10, CTL, UL, IPL)
IF (IPL(NARG) .EQ. 0) GO TO 17
CALL CSOLVE (NARG, 10, UL, RHS2, Q2, IPL)
CALL CSOLVE (NARG, 10, UL, RHS4, Q4, IPL)
20 CONTINUE
DO 2 I=1,NON
PHI2D(K,I,1) ={0., 0.}
PHI2D(K,I,3) ={0., 0.}
DO 2 J=1,NARG
FAC=GREENV(I,J)*FACTOR
PHI2D(K,I,1) = PHI2D(K,I,1)+Q1(J)*FAC
PHI2D(K,I,3) = PHI2D(K,I,3)+Q3(J)*FAC
2 CONTINUE

* PHI2DS are to be interpolated or extrapolated linearly from the
* midpoint of the segments to the offset points.
* QI arrays are to be used for temporary storage for PHI2DS

DO 150 N=2,NON
NM = N - 1
Q1(N) = W1(N)*PHI2D(K,NM,1) + W2(N)*PHI2D(K,N,1)
Q3(N) = W1(N)*PHI2D(K,NM,3) + W2(N)*PHI2D(K,N,3)
150 CONTINUE
NM = NON - 1
Q1(1) = W1(1)*PHI2D(K,1,1) + W2(1)*PHI2D(K,2,1)
Q3(1) = W1(1)*PHI2D(K,1,3) + W2(1)*PHI2D(K,2,3)
Q1(NPT) = W1(NPT)*PHI2D(K,NM,1) + W2(NPT)*PHI2D(K,NON,1)
Q3(NPT) = W1(NPT)*PHI2D(K,NM,3) + W2(NPT)*PHI2D(K,NON,3)
DO 90 I=1,NPT
PHI2D(K,I,1) = Q1(I)
90 PHI2D(K,I,3) = Q3(I)
IF(.NOT. LAT) GO TO 10
DO 5 I=1,NON
PHI2D(K,I,2) ={0., 0.}
PHI2D(K,I,4) ={0., 0.}
DO 5 J=1,NARG
FAC=GREENL(I,J)*FACTOR
PHI2D(K,I,2) = PHI2D(K,I,2)+Q2(J)*FAC
5 PHI2D(K,I,4) = PHI2D(K,I,4)+Q4(J)*FAC
DO 160 N=2,NON
NM = N - 1
Q2(N) = W1(N)*PHI2D(K,NM,2) + W2(N)*PHI2D(K,N,2)
Q4(N) = W1(N)*PHI2D(K,NM,4) + W2(N)*PHI2D(K,N,4)
160 CONTINUE
NM = NON - 1
Q2(1) = W1(1)*PHI2D(K,1,2) + W2(1)*PHI2D(K,2,2)
Q4(1) = W1(1)*PHI2D(K,1,4) + W2(1)*PHI2D(K,2,4)
Q2(NPT) = W1(NPT)*PHI2D(K,NM,2) + W2(NPT)*PHI2D(K,NON,2)
Q4(NPT) = W1(NPT)*PHI2D(K,NM,4) + W2(NPT)*PHI2D(K,NON,4)
DO 97 I=1,NPT
PHI2D(K,I,2) = Q2(I)
97 PHI2D(K,I,4) = Q4(I)
10 CONTINUE
GO TO 19
17 WRITE (IPRIN,18) K
18 FORMAT (//// 10X,'TWODPT -- SINGULAR MATRIX AT K=', I3)

```

```

      STOP
19  CONTINUE

* patch to obtain correct potential

      DO 32 K=1,NSIGMA
      DO 30 I=1,NPT
      DO 31 J=1,4
      PHI2D(K,I,J)=-CONJG(PHI2D(K,I,J))
31  CONTINUE
      PHI2D(K,I,4)=LPP*PHI2D(K,I,4)
30  CONTINUE
32  CONTINUE
      DO 75 I=1,NSIGMA
      SIGMA(I) = SIGMA(I)/SQRLG
75  CONTINUE
      DO 80 J=1,NPT
      ENN(4,J,ISTA) = ENN(4,J,ISTA)*LPP
      YY(J,ISTA) = YY(J,ISTA)*LPP
      ZZ(J,ISTA) = ZZ(J,ISTA)*LPP
80  CONTINUE

      RETURN
      END

C DECK VELACC
      SUBROUTINE VELACC (IM,IT,GRAV,NL,NU,OMEGA,E,RAO1,PHS1,RAO2,PHS2,
2 NOMEA,NPLANE,IPHS)

* This routine obtains the velocity and acceleration rao's and
* phase angles for motions at the origin and at a point.
* W.G.MEYERS, DTNSRDC, 100477

      DIMENSION OMEGA(E),RAO1(NOMEA),PHS1(NOMEA),RAO2(NOMEA),
2 PHS2(NOMEA)

      GRAV2 = GRAV*GRAV
      DO 20 I=NL,NU
      OMEGE2 = OMEGA(I)*OMEGA(E)
      OMEGE4 = OMEGE2*OMEGE2
      DO 10 J=1,NPLANE
      IF (IT.EQ.2 .AND. J.EQ.1) RAO1(I) = RAO1(I)*OMEGE2
      IF (IT.EQ.2 .AND. J.EQ.2) RAO2(I) = RAO2(I)*OMEGE2
      IF (IT.EQ.3 .AND. J.EQ.1) RAO1(I) = RAO1(I)*OMEGE4
      IF (IT.EQ.3 .AND. J.EQ.2) RAO2(I) = RAO2(I)*OMEGE4
      IF (IT.EQ.3 .AND. IM.LT.4 .AND. J.EQ.1) RAO1(I) = RAO1(I)/GRAV2
      IF (IT.EQ.3 .AND. IM.LT.4 .AND. J.EQ.2) RAO2(I) = RAO2(I)/GRAV2
      IF (IPHS .EQ. 0) GO TO 10
      IF (IT.EQ.2 .AND. J.EQ.1) PHS1(I) = PHS1(I) + 90.
      IF (IT.EQ.2 .AND. J.EQ.2) PHS2(I) = PHS2(I) + 90.
      IF (IT.EQ.3 .AND. J.EQ.1) PHS1(I) = PHS1(I) + 180.
      IF (IT.EQ.3 .AND. J.EQ.2) PHS2(I) = PHS2(I) + 180.

10  CONTINUE
20  CONTINUE

      RETURN
      END

C DECK VISC
      SUBROUTINE VISC

      COMMON /CH3D/ ISIGMA,SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI,
2 IMMIN,IMMAX,IMDEL,LMIN,LMAX
      REAL SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI(4)
      INTEGER ISIGMA,IMMIN,IMMAX,IMDEL,LMIN,LMAX

      COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
      INTEGER NVK,NMU,NOMEA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8),
      REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

```

```

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /PHYSICO/ II,TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADDEG,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

COMMON /RLDBK/ PSUR(25),BMK(25),DK(25),CAK(25),HQ,HSPAN,HMNCHD,
2 HAREA,HXCP,HYCP,HZCP,HGAMMA,HYHAT,HEAR,HLCS,RQ(2),RSPAN(2),
2 RMNCHD(2),RAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),
2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
2 SYCP(2),SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2),BQ(2),
2 BSPAN(2),BMNCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),
2 PQ(2,2),PSPAN(2,2),PMNCHD(2,2),PAREA(2,2),PXCP(2,2),PYCP(2,2),
2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),PLCS(2,2),
2 STADHP(10),SHPDMP(10,8),ENCON,WPHI,TPHI,WHELM(4,9),SFELM(4,9,8),
2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
2 ENWM,ENSF(8,8),ENRE(8),ENPE(8),ENFE(8),ENHE(8),ENBE(8),
2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(8),ENSL(8),ENBL(8),
2 ENSHP(8,8),RELM(4,9),ITS(25),RD(25),EDDY(8,25),RGB(25)
REAL RDBLK(2692)
EQUIVALENCE (PSUR(1),RDBLK(1))

DO 10 IA=1,NRANG
DO 10 IS=1,NSIGMA
SHPDMP(1S,IA) = 0
10 CONTINUE
DO 40 K=1,NSTATN
IF (NOFSET(K) .LT. 2) GO TO 40
CON = 4./ (3.*PI)*RHO*PSUR(K)*RGB(K)**3
DO 30 IA=1,NRANG
DO 20 IS=1,NSIGMA
STADMP(IS) = CON*SIGMA(IS)*RANG(IA)*EDDY(IA,K)
STADMP(IS) = SIGMA(IS)*STADMP(IS)
SHPDMP(1S,IA) = SHPDMP(1S,IA) + STADMP(IS)
20 CONTINUE
30 CONTINUE
40 CONTINUE
DO 50 IA=1,NRANG
CALL SPFIT (SIGMA,SHPDMP(1,IA),HEELM(1,1,IA),NSIGMA)
ENHE(IA) = ENCON*REVAL(HEELM(1,ISIGMA,IA),WTSI)
50 CONTINUE

RETURN
END

C DECK VUNIT2
SUBROUTINE VUNIT2 (V1, S1, V2)

*   VUNIT2 created from VUNIT ( NAVSEC-N065 ) - A M REED JULY 1976
*   unitizes plane direction vector

DIMENSION V1(2), V2(2)

S = SQRT( V2(1)*V2(1) + V2(2)*V2(2) )
IF (S .LE. 0.000001*(ABS(V2(1))+ABS(V2(2)))) GO TO 2000
S1=S

```

```

V1(1)=V2(1)/S
V1(2)=V2(2)/S
GO TO 99999
2000  CONTINUE
      S1=0.0
      V1(1)=0.0
      V1(2)=0.0
99999  CONTINUE

      RETURN
      END

C DECK WAVMAK
SUBROUTINE WAVMAK

COMMON /CH3D/ ISIGMA,SIGMIN,SIGMAX,V,SINMU,COSMU,WTSI,
2 IMMIN,IMMAX,IMDEL,LMIN,LMAX
REAL S,GMIN,SIGMAX,V,SINMU,COSMU,WTSI(4)
INTEGER ISIGMA,IMMIN,IMMAX,IMDEL,LMIN,LMAX

COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELG,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWF,VCB,FBDX,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAW,IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELG,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWF,VCB,
2 FBDX(10),FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
6 IYAWRL,CHEAVE,CPITCH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /INDEX/ PFIDX,LPFIDX,RMIDX,LRMIDX,SVIDX,LSVIDX
INTEGER LPFIDX,LRMIDX,LSVIDX
REAL PFIDX(235),RMIDX(163),SVIDX(3)

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL
INTEGER SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFIL,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL

COMMON /PHYSICO/ II,TPI,PI,PIOT,DEGRAD,RADdeg,VKMETR,METRVK,GRAV,
2 RHO,GNU,RHOS,RHOF,GNUS,GNUF,FTMETR,PUNITS,REYSCL
COMPLEX II
CHARACTER*4 PUNITS(2)
REAL TPI,PI,PIOT,DEGRAD,RADdeg,VKMETR,METRVK,GRAV,RHO,GNU,RHOS,
1 RHOF,GNUS,GNUF,FTMETR

COMMON /RLDBK/ PSUR(25),BMK(25),DK(25),CAK(25),HQ,HSPAN,HMNCHD,
2 HAREA,HXCP,HYCP,HZCP,HGAMMA,HYHAT,HEAR,HLCS,RQ(2),RSPAN(2),
2 RMNCHD(2),RAREA(2),RXCP(2),RYCP(2),RZCP(2),RGAMMA(2),RYHAT(2),
2 REAR(2),RLCS(2),SQ(2),SSPAN(2),SMNCHD(2),SAREA(2),SXCP(2),
2 SYCP(2),SZCP(2),SGAMMA(2),SYHAT(2),SEAR(2),SLCS(2),BQ(2),
2 BSPAN(2),BMNCHD(2),BAREA(2),BXCP(2),BYCP(2),BZCP(2),BGAMMA(2),
2 BYHAT(2),BEAR(2),BLCS(2),FQ(2),FSPAN(2),FMNCHD(2),FAREA(2),
2 FXCP(2),FYCP(2),FZCP(2),FGAMMA(2),FYHAT(2),FEAR(2),FLCS(2),
2 PQ(2,2),PSPAN(2,2),PMNCHD(2,2),PAREA(2,2),PXCP(2,2),PYCP(2,2),
2 PZCP(2,2),PGAMMA(2,2),PYHAT(2,2),PEAR(2,2),FLCS(2,2),
2 STADMP(10),SHPDMP(10,8),ENCON,WPHI,TPHI,WMELM(4,9,8),SFELM(4,9,8),
2 REELM(4,9,8),PEELM(4,9,8),FEELM(4,9,8),HEELM(4,9,8),BEELM(4,9,8),
2 ENUM,ENSF(8,8),ENRE(8),ENPF(8),ENFE(8),ENHE(8),ENBE(8),
2 ENEMV(8,8),ENRL(8),ENPL(8),ENFL(8),ENHL(8),ENSL(8),ENBL(8),
2 ENSHP(8,8),RELM(4,9),ITS(25),RD(25),EDDY(8,25),RGB(25)

```

```

REAL RDBLK(2692)
EQUIVALENCE (PSUR(1),RDBLK(1))

COMMON /SMPSYS/ FIS,AS,SIS,SOS,SDS,HALOS,DEV,PRN,SMPPS,SMPIS,
2 SMPDS,SHPTYPS,SHIPS,VARS,CYCLS,TITLES,OPTION,LSIS,LSOS,
2 LSOS,LHALOS,LDEV,LPRN,LSMPPS,LSMPIS,LSMPOS,LSMPDS,LSHPTYPS,
2 LSHIPS,LTTITLES
CHARACTER*160 AS
CHARACTER*80 FIS,SIS,SOS,SDS,TITLES
CHARACTER*20 HALOS,DEV,PRN,SMPPS,SMPIS,SMPDS,SHPTYPS
CHARACTER SHIPS*6,VARS*2,CYCLS*2
INTEGER*2 OPTION

COMMON /TELEM/ TELEM
COMPLEX TELEM(4,9,10)

COMPLEX T22,T24,T42,T44,T44G(10),CELM(4,9),CT44G,CDUM
REAL IROLLG,I44G
DATA EPS /0.25/

FIS = SDS(1:LSDS) //''.COF'
OPEN (UNIT=COFFIL,FILE=FIS,FORM='UNFORMATTED',STATUS='UNKNOWN')
READ (COFFIL) TELEM
CLOSE (UNIT=COFFIL)

* wavemaking (origin at VCG)
DO 10 IS=1,NSIGMA
JS = IS
J = 1
IF (IS .EQ. NSIGMA) JS = IS - 1
IF (IS .EQ. NSIGMA) J = 3
T44 = TELEM(J,JS,6)
T22 = TELEM(J,JS,5)
T24 = TELEM(J,JS,8)
T42 = T24

* translate to VCG
T44G(IS) = T44 + VCG*(T24 + T42 + VCG*T22)
SHPDMP(IS,1) = AIMAG(T44G(IS))
10 CONTINUE
CALL CPFIT (SIGMA,T44G,CELM,NSIGMA)

* find natural roll frequency
C44 = CROLL
IROLLG = MASS*(KROLL*BEAM)**2
I44G= IROLLG
WPHI = SQRT(C44/I44G)
TPHI = TPI/WPHI
IDONE = 0
DO 20 I=1,10
IT = I
TS = TPHI
CALL CPLVAL (SIGMA,NSIGMA,CELM,WPHI,CT44G,CDUM,1SIGMA)
A44G = REAL(CT44G)/(-WPHI**2)
I44G= IROLLG + A44G
IF (IDONE .EQ. 1) GO TO 30
WPHI = SQRT(C44/I44G)
TPHI = TPI/WPHI
IF (ABS(TPHI-TS) .LT. EPS) IDONE = 1
20 CONTINUE
30 CONTINUE
CALL FINTSP (WPHI)
CALL SPFIT (SIGMA,SHPDMP,WMELM,NSIGMA)
ENCON = 1. / (2.*C44)
ENWM = ENCON + REVAL(WMELM(1,1SIGMA),WTSI)

RETURN
END

```

C DECK WEDEFN
SUBROUTINE WEDEFN (NWEVN,WEVN)

* This routine calculates the evenly-spaced encounter wave
* frequencies over which the response spectra are calculated.
* The number of frequencies must be set equal to 100.
* W.G.MEYERS, DTNSRDC, 072877

```
DIMENSION WEVN(NWEVN)
K = 0
DWE = 0.01
DO 110 J=1,54
K = K + 1
110  WEVN(K) = 0.05 + (J-1)*DWE
DWE = 0.02
DO 120 I=1,21
K = K + 1
120  WEVN(K) = WEVN(54)+I*DWE
DWE = 0.10
DO 130 I=1,10
K = K + 1
130  WEVN(K) = WEVN(75)+I*DWE
DWE = 0.2
DO 140 I=1,10
K = K + 1
140  WEVN(K) = WEVN(85)+I*DWE
DWE = 0.4
DO 150 I=1,5
K = K+1
150  WEVN(K) = WEVN(95)+I*DWE
```

RETURN
END

C DECK WTPELM
SUBROUTINE WTPELM(ISTATN, PELEM)

* writes out spline elements for 2-d potential and forces
* W. R. MCCREIGHT DTNSRDC JULY, 1977

```
COMMON /ENVIOR/ VK,NVK,MU,NMU,OMEGA,NOMEGA,SIGMA,NSIGMA,SIGWH,
1 NSIGWH,TMODAL,NTMOD,NRANG,RANG,RLANG,S,NNMU,FRNUM,VFS
INTEGER NVK,NMU,NOMEGA,NSIGMA,NSIGWH,NTMOD,NRANG,NNMU(8)
REAL VK(8),MU(37,8),OMEGA(30),SIGMA(10),SIGWH(4),TMODAL(8),
2 RANG(8),RLANG(8),S(30,8),FRNUM(8),VFS(8)

COMMON /GEOM/ X,NSTATN,Y,Z,NOFSET,LPP,BEAM,DRAFT,LCF,
1 VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,FBDY,
2 FBDZ,NFREBD,XPT,YPT,ZPT,NPTS,LCB,GML,ASTAT,BSTAT,TITLE,MASS,
2 DISPLM,IPITCH,IROLL,IYAWL,CHEAVE,CPTICH,CHEAPI,CROLL,
2 AREAMX,WSURF,GIRTH,FBDZV,DBLWL,TLCB
INTEGER NSTATN,NOFSET(25),NFREBD,NPTS
CHARACTER*4 TITLE(20)
REAL X(25),Y(10,25),Z(10,25),FBDZV(8,10),LPP,BEAM,DBLWL,TLCB,
2 DRAFT,LCF,VCG,GM,DELGM,NEBLA,KPITCH,KROLL,KYAW,KYAWRL,AWP,VCB,
2 FBDY(10),FBDZ(10),XPT(10),YPT(10),ZPT(10),LCB,GML,
4 ASTAT(25),BSTAT(25),MASS,DISPLM,IPITCH,IROLL,IYAW,
5 IYAWRL,CHEAVE,CPTICH,CHEAPI,CROLL,AREAMX,WSURF,GIRTH(25)

COMMON /INDEX/ PFIDX,LPFIDX,RMIDX,LRMIDX,SVIDX,LSVIDX
INTEGER LPFIDX,LRMIDX,LSVIDX
REAL PFIDX(235),RMIDX(183),SVIDX(3)

COMMON /IO/ SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL
INTEGER SYSFIL,POTFIL,COFFIL,LCOFIL,ICARD,TEXFIL,IPRIN,
2 SCRFILE,HPLFIL,LRAFIL,ORGFIL,RAOFIL,RMSFIL,SEVFIL,SPDFIL,
2 SPTFIL,LACFIL,LAEFIL

COMMON /STATE/ LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL
LOGICAL LAT,VRT,LOADS,ADDRES,SALT,HEAD,EXROLL,BKEEL
```

```

DIMENSION DATA(320)
COMPLEX PELEM(4,0.40)

IF (NOFSET(ISTATN) .LE. 0) RETURN
IMMIN=1
IF (.NOT. VRT) IMMIN=2
IMMAX=4
IF (.NOT. LAT) IMMAX=3
IMDEL=2
IF (VRT .AND. LAT) IMDEL=1
ISGMX=NSIGMA-1
DO 1 ISIGMA=1,ISGMX
NEXT=1
NNODE=NOFSET(ISTATN)
DO 2 J=1,NNODE
DO 3 IMODE=IMMIN,IMMAX,IMDEL
DO 4 I=1,4
IDX=(IMODE-1)*10+J
DATA(NEXT)=REAL(PELEM(1,ISIGMA,IDX))
DATA(NEXT+1)=AIMAG(PELEM(1,ISIGMA,IDX))
NEXT=NEXT+2
4 CONTINUE
3 CONTINUE
2 CONTINUE
NDATP=NEXT-1
INDEX=(ISIGMA-1)*NSTATN+ISTATN

*      change for VAX-11 version.
* CDC      CALL WRITNS(POTFIL,DATA,NDATP,INDEX)
        WRITE (POTFIL,REC=INDEX) DATA
1 CONTINUE
RETURN
END

C DECK XMSSC
SUBROUTINE XMSSC (IPH,B2,MSLC,NLCH,RMSLC,RMSSC)
DIMENSION B2(NLCH)

REAL MSLC(24),MSSC
MSSC = 0.
LH = IPH - 1
DO 10 IH=1,NLCH
LH = LH + 1
IF (LH .GT. 24) LH = LH - 24
MSSC = MSSC + B2(IH)*MSLC(LH)
10 CONTINUE
KH = IPH + 5
IF (KH .GT. 24) KH = KH - 24
RMSLC = MSLC(KH)
RMSSC = MSSC

RETURN
END

```


REFERENCES

1. Meyers, W. G., T. R. Applebee, and A. E. Baitis, "User's Manual for the Standard Ship Motion Program, SMP," DTNSRDC/SPD-0936-01 (Sep 1981).
2. Meyers, W. G. and A.E. Baitis, "SMP84: Improvements to Capability and Prediction Accuracy of the Standard Ship Motion Program, SMP81," DTNSRDC/SPD-0936-04 (Sep 1985).
3. Smith, T. C. and C. J. Bennett, "User's Manual for the Ship Motions Applications Manager - PREDICT," CRDKNSWC/HD-0936-06 (Feb 1994).
4. Graham, R., A.E. Baitis, and W.G. Meyers, "On the Development of Seakeeping Criteria," Naval Engineers Journal, Vol 104, No 4, pg 259-275 (May 1992)
5. Meyers, W. G., C. J. Bennett, and T. R. Applebee, "User's Manual for the Simulation Time History and Access Time History Programs," CARDEROCKDIV, NSWC/SHD-1297-01 (Feb 1993).