No. 212
June 1979

# THE COMPUTER SYSTEM SPIRAL DOCUMENTATION FOR THE USER AND THE MODULE WRITER

John B. Woodward (Editor)

THE DEPARTMENT OF NAVAL ARCHITECTURE AND MARINE ENGINEERING

THE UNIVERSITY OF MICHIGAN

COLLEGE OF ENGINEERING

No. 212
June 1979

# The Computer System SPIRAL

## Documentation for the User and the Module Writer

John B. Woodward (editor)

Department of Naval Architecture
and Marine Engineering
College of Engineering
The University of Michigan
Ann Arbor, Michigan 48109

CONTENTS

# SOME SHIP DESIGN WORK BY DIGITAL COMPUTER

An Introduction for the UM Student of Naval Architecture and Marine Engineering

University of Michigan
Department of Naval Architecture
and Marine Engineering

John B. Woodward

June 1979

INTRODUCTION

Some of the important ship design work required of a student (as in NA 470, NA 475) can be done by the computer tools available (1979) in our SPIRAL computer library.[1] This memo outlines by brief abstracts what is available in that library, sketches how some of the principal tasks are accomplished, and lists the documentation needed when you actually try them. (All documentation that applies directly to SPIRAL is bound into this report.)

About that SPIRAL... It is a coordinated set of interactive programs intended to help students with their design work. It consists of an "executive routine" called DEX, a library of computational programs (called "modules" here), and a database that you can use to store information, such as, for example, the output of one module that is needed as input for another. It has some features that make computation very easy for even the inexperienced computer user. It isn't perfect -- you will surely be moved by frustration to mutter an occasional naughty word -- but much superior to older ways of using the computer (punching cards, and that kind of gruesome stuff).

Another chapter of this report titled USER'S GUIDE FOR THE INTERACTIVE COMPUTING SYSTEM SPIRAL tells you what a user needs to know about SPIRAL. Take a look at it before trying to do the things outlined here.

Each module is described by its own USER'S MANUAL, and you should have a copy for each module of interest to you. (If the module was completed before June 1979, its manual is in this report.)

SPIRAL is generally run from any remote terminal. However, the FAIRING module that is used for fairing of hull lines must be run from a graphics terminal, and the reading of offsets directly from a lines plan is done only by means of a digitizer. One graphics terminal and one digitizer were available in the NAEME Building at the time this was written.

---

[1] "Library" here does not mean a bunch of books, but refers to the programs that are part of the system, all stored in MTS files.

SUMMARY OF DOCUMENTATION[1]

1. SOME SHIP DESIGN WORK BY DIGITAL COMPUTER (That's this!)

2. ABOUT SPIRAL... (First draft of a paper Prof Woodward wrote for some educational journal, telling the world -- that part that reads educational journals, anyhow -- what a lucky him/her you are to have SPIRAL easing your struggles with na&me. Read it if you want to get some background.)

3. USER'S GUIDE FOR THE INTERACTIVE COMPUTING SYSTEM SPIRAL

4. USER MANUAL (One for each module in the SPIRAL library)

5. MODULE PROGRAMMER'S GUIDE (No help to the user -- needed only by someone writing a module.)

6. INTRODUCTION TO THE COMPUTING CENTER (A publication of the Computing Center. Says nothing about SPIRAL, but will help you with things you need to know about the UofM computer and MTS.)

---

[1]All items listed, except the last, are bound into this report.

BRIEF ABSRACTS OF SPIRAL MODULES (what they do)

1. ALTER

Develops hull offsets from several parent hulls

2. BCOST (Building Cost)

Calculates building costs for several types of ships. Should be used only for relative costs, since module has not been corrected for inflation.

3. DAMAGE (Damaged Stability)

Calculates data for cross curves of damaged stability.

4. DAMSS (Damaged Statical Stability)

Damaged statical stability calculations.

5. ECON (Economics)

Calculates operating costs and measures of merit for several types of ships.

6. FLOOD (Floodable Length)

Calculates data needed for plotting floodable length.

7. FAIRING

Interactive fairing of hull lines. Displays the lines, or sends to CALCOMP plotter for printing.

8. HYDRO (Hydrostatics)

Calculates all hydrostatic data for a hull. Sends to CALCOMP plotter for printing.

9. STABLE (Intact Stability)

Calculates data needed for plotting intact stability curves.

10. LIMIT (Limiting Draft)

Calculates forward and aft drafts necessary in order to survive subsequent flooding of specified compartments.

11. LINESGEN

Generates offsets by perturbing the offsets of a Series 60 hull.

12. STRENGTH (Longitudinal Strength)

Longitudinal strength calculations (bending moments).

13. POWER

Calculates calm water resistance and power from Taylor and Series 60 data.

14. SCREWOPT (Propeller Design)

Interactive optimizing of a Wageningen B-series propeller, with copious output.

15. VERIFY (Ship Hull Verification)

A pre-processor for offsets needed as input by STABLE, STRENGTH, TRIML, LIMIT.

16. TRIML (Trim Lines)

Calculates draft and trim resulting from flooding specified compartments.

17. SEAKEEP (Seakeeping)

Calculates seakeeping properties of ships similar to Series-60 models.

18. WEIGHT

Calculates weights for several ship types.

19. MORC (MORC Rating Calculation)

Calculates rating of sailyacht under current (1979) MORC rule.

20. PLANEOPT

Optimizes dimensions of a planing boat for best resistance in either smooth or rough water.

21. STMCYCLE (Steamcycle Analysis)

Analyzes regenerative steam power cycles (not capable of doing actual "heat balance").

22. MIDSHIP

Analyzes structure of a midship section. Input via graphics terminal or digitizer required. User manual is not here, but in report 213.

23. IOR

Calculates sailyacht rating under current (1979) IOR rule.

WHAT CAN YOU DO WITH SPIRAL?[1]

A hint or two may help if you've never done this sort of thing before.

Since each of the modules does something (that's a promise!), there seems to be at least one thing that can be done for every one just listed, and that thing is suggested by the title or by the abstract. However, the greatest utility appears when a task involves a number of different analyses, and you find a combination of modules that will handle the whole thing.

For example, note that two modules (ALTER and LINESGEN) produce hull offsets from parent hulls. You're roughing out a preliminary ship design, and one of the parents built into these modules is similar to your ship. You can make an estimate of the overall dimensions of the intended hull, call LINESGEN or ALTER to produce offsets, go immediately to HYDRO to get things like displacement and block coefficient (as well as anything hydrostatic, and plotted, too), then to POWER to find propulsive power, to STABLE (via VERIFY for reasons you will learn later) to check stability characteristics, and to WEIGHT to get an estimate of principal weights. You may be concerned with building the most ship for the least amount of money, so could also zip through BCOST to get an estimate of building cost (probably won't be in line with current costs, so use only for comparative purposes). Now you have a design roughed out. Many things may be unsatisfactory (too much power, too little deadweight??), but you have quickly gotten the key elements of one turn around the "design spiral," and after making adjustments, can just as quickly give it another trial.

Now, suppose you want to start such a process from a lines plan, even though it may be only a freehand sketch. You can DIGITIZE to put its offsets into a file or SPIRAL database, and proceed from there in much the same manner as above.

And other things...

---

[1] A student once tried to answer this question with words surly and shocking to his professor. The professor called the Dean, and the Dean called the offender's mother. She promised to slap him hard enough to break his medula oblongata the next time he came home. None of this would have happened if he had read the documentation carefully, and asked the professor to help him with things he didn't understand.

## INTRODUCTION

Naval architecture and marine engineering is the engineering discipline whose principal theme is the design of water craft. It has much in common with the other engineering fields, including the benefits of the digital computer, and it shares the burdens of continuing efforts to incorporate computing into both practice and teaching.

Although many enthusiasts have pursued computerization of naval architecture and marine engineering since the 1950s, the educational part received no organized attention until the advent of the project we discuss here. It was conceived in the early 1970s by Professors Horst Nowacki and Bertram Herzog of the University of Mighigan[1] in response to obvious needs and opportunities, funded by the National Science Foundation during 1974-79, and directed through most of this period by the author.

The project objective was to produce an integrated system of computer programs under the direction of a versatile executive routine. The programs (or MODULES as they shall be called here henceforth) were to cover all of the major calculational steps expected in preliminary ship design; the excutive was to handle the modules so that an interactive user would be insulated from most of the stringencies of the computer operating system. Since the modules would be necessarily written to mesh with the executive, they would have a standard form, making familiarity with one module tantamount to familiary with all.

The project reached its end just as this paper was written; bugs were pretty well eliminated, and the extent of success and failure distinguishable. Here, then, we'll outline the product, tell of its use, and critique its apparent virtues and faults.

## BACKGROUND

Naval architecture and marine engineering is characterized by numerous lengthy calculations that were thankfully turned over to the digital computer as soon as it was generally available, and new uses (optimization analyses, for example) soon were put into practice. Meanwhile, education in the use of computers developed apace, or so it seemed. Courses in programming and computer use were being taught to engineering students at the University of Michigan by the mid 1950s. A

---

[1]Nowacki is now professor at the Technical University of Berlin, and Herzog is director of the computing center at the University of Colorado.

graduate-level course entitled Computer-Aided Ship Design was added to the na&me curriculum at Michigan in 1968, and largely through the term projects assigned in this course, an extensive library of programs had been built up there within a few years.

However, this apparent accomplishment was actually of almost no permanent worth. A review in 1973, made in response to a scouring of the marine industry for computer programs by the Maritime Administration (U S Department of Commerce), revealed 80 programs in the library, but only 5 that were of quality (free of bugs, documentation adequate, etc) acceptable for general use. A few more might be usable locally with guidance from their authors who were still at the University. Programs and their documentation were written to differing standards (or to no discernable standards at all), making use by anyone other than the author almost a research project in itself. The typical undergraduate student, who could surely benefit from continued use of the computer in na&me courses, was virtually shut out by the difficulties and uncertainties.

Interactive computing was not available to na&me undergraduates, even though the University of Michigan had established a time-sharing system with numerous remote terminals in the mid 1960s. The major fault was lack of programs written for interactive use, although the general inaccessability of terminals to students also contributed.

These words describe only the state of computer use by students at the University of Michigan in the early 1970s, but they do pretty well fit the situation at a like time of other schools with similar engineering programs.

GOALS OF THE NATIONAL SCIENCE FOUNDATION PROJECT

The dismal situation just pictured was recognized by Professor Horst Nowacki, principal exponent of computer-aided ship design in the Michigan na&me department. With the assistance of Professor Bertram Herzog of the Michigan industrial engineering department, he developed the proposal to the National Science Foundation that was to become the 1974-79 project. Its principal goal was to develop an integrated system of programs that could be used readily by the undergraduate student, and used in a way that contributed to learning the ship design process taught at Michigan. This goal can be broken down into the following particulars:

1. There should be an executive routine to manage a library of immediately callable modules. This executive should protect the user against the strictures of the computer operating system (for example, absorb an input error, and offer an opportunity for correction, rather than aborting a run), should provide rapid transfer from one module to another

and between modules and databases, and should offer features that enhance the capability of the modules (for example, subroutines that manage menus of options offered to the user).

2. There should be a library of computational modules that cover the major calculations performed in preliminary ship design. These modules should not only be compatible with the executive, but should be in all feasible ways in standard form to improve user familiarity.

3. There should be user databases managed by the executive for storage of data between runs, and for transfer of data between associated modules.

4. The modules should be oriented toward a student designer rather than toward the experienced practicioner (for example, as much decision-making as possible should be left to the user).

5. The entire package should be readily transferable throughout the institutions teaching naval architecture and marine engineering.

6. Documentation should be complete, and readily understandable by a naive user. It should cover the principles of module programming so that the system could be enlarged in later years by people who had not been associated with the developmental project.

THE PRODUCT: COMPUTER SYSTEM "SPIRAL"

The project developed a computer system known at Michigan as SPIRAL,[1] consisting of the executive routine DEX (from Design EXecutive), the library of modules, and the user databases.

The heart of the executive is a menu of options (menu DEX.MAIN, shown by Figure 1) that it presents to the user as its first act. A reasonable way to describe DEX (as the user sees it) is to sketch what action each of these choices initiates, as following:

---

[1] Believe-it-or-not, this is not an acronym, but comes from the notion that design proceeds in spiral fashion toward finality, and that this computer tool gives a student designer the capability of making a quick advance around one loop of that spiral.

LIBRARY:
Displays the names of all modules in the SPIRAL library.

HELP:
Each of the modules is accompanied by a "help file" containing clarifying points that the module author feared a user might need.

DISPLAY:
Several menus are encountered in the running of each module. A user who has forgotten what the choices are can give this command to display the menu. Figure 1 was produced simply by making this choice through the command DISPLAY DEX.MAIN.

ALTER:
Used to switch input from keyboard to graphics screen, and vice-versa.

DATABASE:
Used to inform the executive of the name of a file to be used as a database, and to initialize it for that service.

EDIT-DB:
Leads to editing of a database. Variable names can be established, given numerical values, deleted, printed, etc.

BEGIN:
This choice followed by a module name loads and begins execution of that module.

CONTINUE:
Continues the execution of a module that has been interrupted by the user to consult DISPLAY, HELP, etc.

MTS:
Returns control from DEX to the computer operating system (MTS).

QUIT:
To quit! (Returns control to computer operating system.)

A valuable feature of DEX is that a user may return to this main menu at any place a module asks for an interactive input, make use there of any of the options just outlined, and then resume module execution at the point where it was supended. This detour is initiated simply by typing "&" instead of the requested input, then choosing the option CONTINUE when ready to resume the module. For example, a module may demand a choice from a menu unfamiliar to the user. A quick return to DEX.MAIN, followed by DISPLAY of the offending menu, is a quick way to resolve the uncertainty over what to type; it's usually a much

```
&     +----------+
&     +   MENU   +
&     + DEX.MAIN +
&     +----------+
&   1 + LIBRARY  +
&     +----------+
&   2 + HELP     +
&     +----------+
&   3 + DISPLAY  +
&     +----------+
&   4 + ALTER    +
&     +----------+
&   5 + DATABASE +
&     +----------+
&   6 + EDIT-DB  +
&     +----------+
&   7 + BEGIN    +
&     +----------+
&   8 + CONTINUE +
&     +----------+
&   9 + MTS      +
&     +----------+
&  10 + QUIT     +
&     +----------+
&
```

Figure 1  The Menu DEX.MAIN

```
&     +----------+
&     +   MENU   +
&     + MOD.MAIN +
&     +----------+
&   1 + ENTER-DB +
&     +----------+
&   2 + INPUT    +
&     +----------+
&   3 + DEFAULT  +
&     +----------+
&   4 + COMPUTE  +
&     +----------+
&   5 + FILL-DB  +
&     +----------+
&   6 + PRINT    +
&     +----------+
&   7 + SEND     +
&     +----------+
&   8 + STOP     +
&     +----------+
&
```

Figure 2  The Menu MOD.MAIN

handier route than stopping to paw through the documentation.

The executive manages input for the modules, does so in a format-free mode, and traps errors rather than allowing the computer operating system to terminate execution.

The modules are written in FORTRAN, and generally resemble any program in that language for performance of engineering calculations. However, they do make use of several useful functions that are part of the DEX executive. One is the input routine just mentioned. Another establishes the module's menus.

And there are others of value to a programmer, but not evident to the user. The feature most apparent to the user is the uniform format of the modules, enforced not by the executive, but by written standardization specifications. Every module[1] begins with the menu MOD.MAIN (Figure 2); a user familiar with one module thus can usually run other modules without study of the additional documentation. Typically, the running follows the order laid down by the main menu, as follows:

1. If input data lies in a database file, the first choice is ENTER-DB, upon which the model acquires all of its input without further action by the user. (It will have to ask for the name of the database.)

2. If the input, or some additional information, is to be given interactively, the choice is INPUT. The module asks for the required information.

3. There may be constants (for example, density of water) that are built into the module, but which the user may want to change (for example, seawater density to fresh water density). Choice of DEFAULT allows this to be done.

4. COMPUTE orders the performance of the module's calculations.

5. Some modules are linked to others via the database. If the module in question supplies output to another as input in this manner, then the user can elect FILL-DB to order the database filling.

6. Output to be printed at the user's location is ordered by PRINT.

---

[1]Not 100 percent true, for a few modules defy standardization.

7. A user may want output sent to a line printer at a remote location, stored in a file, or placed in a file for access by a plotter. The option SEND (followed by the specification of where) accomplishes this.

8. STOP sends control back to the executive routine.

Now, some of these choices may themselves be menus. For instance, choice of PRINT may lead to a menu (named PRINT) that offers the choices INPUT, RESULTS, or BOTH.

Any disk file supervised by the computer operating system may be designated as a database file under supervision of the DEX executive. The EDIT-DB choice on the menu DEX.MAIN offers the choices (since it is itself a menu) CREATE, STORE, DELETE, COMMENT, EXPLAIN, PRINT, DUMP, TITLE, SEND, DONE by which the user creates names in the database, stores their values, etc.

The executive also contains database manipulative subroutines that may be called by a module programmer. These are used by the module to read input directly (that is, without user intervention) from a database, or to send output directly to it.

USE AT THE UNIVERSITY OF MICHIGAN

Let's illustrate with one of the simpler tasks that a student user might attempt: calculate the righting arm of a particular ship when heeled 20 degrees (the moment arm tending to rotate the ship to the upright). A hand calculation is in principle quite simple, but is nettlesome because it requires calculation of the buoyant force on a rather odd-shaped (usually) body (the immersed part of the heeled hull), and so is one where the computer is a significant labor-saver. With the SPIRAL system, the module STABLE is appropriate. The user when confronted with the first menu (DEX.MAIN) would choose BEGIN STABLE, then at the first module menu, choose ENTER-DB. Presumably the ship in question is one which is the subject of a term project by the student, and so has its offsets (cartesian coordinates of the hull surface) stored in a database file. If so, then the ENTER-DB choice causes the module to acquire those offsets from the database. The next choice would be INPUT ALL, under which the user would type in the data specific to this problem; for example, the angle of heel of interest. The next choice would be COMPUTE, followed by PRINT. The results would appear at the user's terminal. The whole process would require essentially only that time necessary for the typing of the several instructions, plus the time required for output printing. If fumbling and bumbling were avoided, the whole process might be squeezed into 30 seconds. And a user of even moderate acquaintance with SPIRAL might not need documentation, since the computational path I have outlined is followed by most of the modules, and so is quite familiar to such a user.

The virtues of the SPIRAL system are much more evident when a design process involves several related tasks. To illustrate, let's outline an expanded version of the preceding example: a student is preparing a preliminary design of a cargo ship of specified size and speed, and with that righting arm at 20 degree heel having a specifided minimum value. Final selection of hull proportions is to be based on a minimum building cost with all specifications met. Appropriate modules can be called in turn, with transfer of key data handled via a common database. The exercise will proceed through the several modules as follows:

Module ALTER:
Estimated principal dimensions of the hull are supplied via the main menu choice INPUT. The module perturbs appropriate parent hull offsets to suit the input, and stores these offsets in a database file.

Module HYDRO:
Calculates essential hull characteristics, such as its displacement and block coefficient.

Module VERIFY:
Processes the database file containing offsets into the form required by several other modules, including the next one, STABLE.

Modules STABLE:
Calculates the righting arm as described previously.

Module POWER:
Calculates the required propulsive power.

Module WEIGHT:
Calculates the estimated weights of hull steel, machinery, outfitting equipment, etc.

Module BCOST:
Calculates an estimated building cost.

At this point, a great deal of information has been accumulated in a brief session at a terminal -- 2 or 3 minutes might suffice. But to continue, let's suppose that the specified righting arm was not obtained, and that the student then attempts to increase stability by widening the hull (increasing its beam). A likely procedure is to repeat that process just outlined, but with the new value of beam furnished to module ALTER at its menu INPUT. The new righting arm may be satisfactory, but the student will also have other new values to contemplate. For example, the ship is now bigger, hence has greater displacement, requires more power, and costs more to build. The student must therefore ponder the implications of these side consequences, ponder whether other means of meeting the specifications can be found at a lesser increase in building cost.

Here appears a distinct difference between SPIRAL -- intended as it is for the student -- and a design system intended for the practicioner. The latter would benefit from optimization routines that would perform the process of exploration of alternatives. The student fares otherwise; the professor is interested in seeing the best answer, yes, but is more interested in having the student do that pondering; to think about how the specifications can be met, and as a consequence of this human decision process, to probe for the combination of key design values that constitutes the best design. In other words, it is not so much the attainment of the "best" design, but the educational benefit of making the exploration. Note, though, that it is not SPIRAL that compels this exploration of alternatives -- it does not "teach." It is only a tool for the human teacher to manipulate to the student's benefit.

CRITIQUE

At the time this paper was written, no disinterested evaluation of SPIRAL had been attempted, but major virtues and likely criticisms are apparent even to the involved observer, and mention of a few of these can serve to round out this discussion.

The virtues have been laid out already -- we'll merely summarize here: the SPIRAL system is a tool by which a student may range widely and rapidly through several phases of ship design, avoiding the labor of lengthy hand calculation, avoiding the discouraging and time-wasting struggle with disparate individual programs, and skipping swiftly over many of the distracting demands of the computer operating system.

But criticisms... Viewed from close at hand, the most general and most serious one is likely to be that some design/analysis tasks that appear desireable are not included in the SPIRAL library ("you had 5 years and all that money, and you didn't...?!). Well, more of the time and money than a non-participating observer might suspect went into development of the executive routine DEX. It is now apparent that they who conceived the project regarded this part of the system as paramount, and felt that the computational modules would follow easily as soon as the executive was perfected. In particular, the effort to adapt the executive to the computers of other schools required much more effort than anticipated, so that effective emphasis could shift to module development late in the project.

Thus the excellence of the executive came at a price, and one may even question whether the executive is actually necessary -- whether all of the project resources should not have gone into module development. One is tempted to say yes, for computer programs can be written to standard format, and they can be written to communicate with one another via ordinary disk files, all without the benefit of an executive routine. On the other hand, the executive does greatly smooth the path of the naive computer user. The answer thus depends largely on the viewpoint of the observer.

The most serious criticism that I would expect from the National Science Foundation concerns the failure of this system to be adapted generally by other institutions teaching naval architecture and marine engineering. The obvious technical reason for this deficiency is the unforeseen difficulty in transferring it into the environment of other computers and other operating systems. Although such transfer is demonstrably possible, it requires a major effort on the part of the receiving institution, several hundred skilled manhours perhaps being necessary to overcome the many prickly inconsistencies between Michigan and the typical receiver. Those other institutions simply have not yet seen the advertised virtues as being worth the effort they would expend. Our only success with transfer has been the U S Naval Academy. Although the SPIRAL system has been successfully installed there, the price of such success was clearly demonstrated in the struggle for its attainment.

The system may also be criticized for its dependence on the massive clout of the main frame computer, this at a time when microcomputers are begining to innundate engineering schools. The eventual fate of SPIRAL under this threat is not yet predictable. I can only express the wish that the virtues it has demonstrated will either persist or be re-incarnated no matter how the computer environment evolves, and that its deficiencies will be swept away in that evolution.

# USER'S GUIDE
# FOR THE
# INTERACTIVE COMPUTING SYSTEM
# SPIRAL

University of Michigan

Department of Naval Architecture
and Marine Engineering

John B. Woodward

Bradley S. Winslow

May 1978

TABLE OF CONTENTS

# I. INTRODUCTION

SPIRAL is an interactive computing system intended for use by students of naval architecture and marine engineering. It consists of three parts, the Design Executive (DEX), the database, and a library of computational programs known as modules. The Design Executive handles all user commands via lists of options known as "menus". A user may request to see the options on a menu by issueing the command DISPLAY, followed by the appropriate menu name. DEX affords the user with format-free input eliminating the necessity for information to be typed in specific fields. In the event the user should type an illegal input, DEX will interrupt the user with an indication of the error discovered. DEX provides easy access to sources of auxiliary information ("help" files) and manages the database that facilitates data handling.

The database in the SPIRAL system is used to communicate information between modules of the system. The user may interact directly with the database via the "DEX.MAIN" menu option EDIT-DB. This in general is not required, however, as most of the information contained in the database forms a foundation on which various modules perform their calculations.

A library of computational modules comprises the third part of the SPIRAL system. All modules of the system library are written to standards specified by the system developers, so that each module confronts the user in a manner similar to all other modules. The user will therefore be able to apply experience gained in initial sessions to all subsequent work with the system.

Last it should be noted that SPIRAL is an interactive system, and so can only be run in a conversational mode from a terminal. Any terminal connected to the University of Michigan's computer may be used. Some modules involve graphics, thus requiring access via a terminal with graphics capability.

## II. GENERAL FEATURES OF SPIRAL

SPIRAL operates under the Michigan Terminal System (MTS). It may be invoked by a user after completing the SIGNON procedure with MTS with the command,

$RUN XXXX:SPIRAL T=5* (XXXX is the CCID under which SPIRAL is filed. May change occasionally, so check first.)

This command will load SPIRAL and cause its execution to begin. Successful loading is indicated by the appearance of this message

N.A. & M.E.
Educational System

USER:CCID

TODAY is: May 2, 1978 TIME:11:31:57

A new bulletin will then appear if the system management has information to diseminate to its users. The user will then encounter the instruction, "Enter item from menu: DEX.MAIN". An ampersand (&) will appear at the beginning of the next line as the DEX prompt symbol. The user is expected to respond with one of the items from the DEX.MAIN menu. This menu will not automatically appear, hence the user should be acquainted with it from this documentation. Items on the menu are:

LIBRARY

HELP

DISPLAY

ALTER

BEGIN

DATABASE

EDIT-DB

CONTINUE

MTS

QUIT

Each item on the menu leads into a task synonomous with its name. As an example, the frequent task of running a module is initiated by the menu choice BEGIN. The user types this command followed by the name of

---

*It is generally considered good practice to specify a time limit when running a program. A time limit of 5 seconds will usually suffice.

the desired module. The named module is loaded by DEX and its execution begins. Details of this and other tasks are given in section III.

The user should be aware of the significance of the prompt symbols, since they indicate whether MTS, DEX, or a module of the SPIRAL system is in command. These symbols are summarized in the table below:

| Symbol | Location of Control |
|---|---|
| # | MTS |
| & | DEX |
| ? | Module of SPIRAL* |

---

*The question mark (?) also appears as an MTS prompt symbol following an MTS error message. This is not to be confused with the question mark that appears as the Module prompting symbol.

## III. THE DESIGN EXECUTIVE

Control and use of SPIRAL is managed by choices among the ten items on the DEX.MAIN menu of the Design Executive. The ten commands and a brief summary of each are given below.

LIBRARY - displays the list of available modules
HELP - prints a brief set of instructions for the specified module
DISPLAY - lists the items on the menu specified
DATABASE - specifies and initializes a file as the current database
EDIT-DB - invokes the database editor
BEGIN - loads and runs a module
CONTINUE - resumes the running of the suspended module
MTS - returns control to the operating system
QUIT - terminates execution of SPIRAL

A more detailed explanation at each command follows.

1. LIBRARY

This command will effect the printing of the current list of modules available in the SPIRAL library.

2. HELP

A user may need on-the-spot information about a module that is running, or about to be run. The command HELP, followed by the name of the module, causes printing of the help file for this module. This file contains hints that the module author believes would be helpful to a user in such a situation. If the need arises during running, a user can return to menu DEX.MAIN by typing the ampersand (&) in reply to any question posed by the module. Return can subsequently be made to the point of interruption by use of the CONTINUE command.

3. DISPLAY

This command will effect the printing of a menu or several menus for the benefit of a user working with incomplete documentation. Only menus that have been encountered during the current session with SPIRAL will be available for DISPLAY. Menus that have not been encountered during the current session will be unavailable.

The user types DISPLAY, followed by one of these keywords:

| | |
|---|---|
| Name | the name of any individual menu |
| DEX | all menus encountered in the design executive |
| MOD | all menus encountered in the module currently being run |
| ALL | all menus encountered in the design executive and in the module currently being run |

The keyword "Name" refers to the name of the particular menu that the user wishes to display. The menu may be either a menu in DEX or a menu in the module.

4. ALTER

Alterations in the mode of input and output are afforded the user via the command ALTER, followed by one of these keywords:

| | |
|---|---|
| GRAPHIC | specifies operation in graphics mode |
| KEYBOARD | the opposite of GRAPHIC. This is the default case, i.e. KEYBOARD mode is in effect until GRAPHIC mode is specified. |

The user should be signed-on to a terminal with graphics capability in the event that he or she wishes to use the graphics capabilities of SPIRAL.

5. DATABASE

There are several uses for a database; for example, many of the modules are designed to take input from some other module. A user should examine module documentation to see if this is demanded (or helpful for any other reason), and then create an MTS file to serve the purpose. Once this is done, the file must be initialized by use of this menu item. The proper form is

DATABASE INITIAL Filename

where Filename refers to the name of the MTS file.

If an MTS file has previously been initialized as a database, it is not necessary to reinitialize the file again. To identify this file as the current database, the command is used as follows:

DATABASE Filename

This form of the command is useful in the event the user wishes to change the currently active database from one file to another. A common example would be the running of several modules during one session with SPIRAL where some of the modules may require information from a second database file.

The third form of this command releases the currently active database file without the necessity of loading another database file. This command takes the following form:

DATABASE UNLOAD

6. EDIT-DB

This menu choice summons the database editor, which in turn offers a menu having the following choices:

| | | |
|---|---|---|
| CREATE | COMMENT | TITLE |
| STORE | EXPLAIN | SEND |
| DELETE | PRINT | DONE |
| | DUMP | |

(a) CREATE is used to create variable names for storage in the database. It can also be used to assign a numerical value to variables at the same time. Examples below illustrate the possible uses of CREATE.

CREATE LBP - creates a variable named LBP in the database
CREATE LBP,BEAM,DRAFT - creates three variables as named
CREATE LBP=602,BEAM=85.6,DRAFT=29.734 - creates the variables named, and assigns numerical values

Variable names may consist of up to seven alphanumeric characters. Observe in the third example that it is permissible to omit the decimal point for whole numbers.

(b) STORE is used to assign numerical values to existing database variables. For example, the command

STORE LBP=605.1,BEAM=89.3

might be used to change two of the variable values originally stored by the CREATE command used as example in (a).

(c) DELETE is used to remove variable names and values from a database. For example

    DELETE LBP,BEAM,DRAFT

deletes the variables created and given values in (a).

(d) COMMENT is used to append a comment of up to 71 characters to a database variable. For example

    COMMENT LBP 'LENGTH BETWEEN PERPENDICULARS'

(e) EXPLAIN orders printing of the comment, if any, attached to a database variable. For example

    EXPLAIN LBP

causes the comment used as an example in (d) to be printed.

(f) PRINT orders the printing of the value of database variables. For example

    PRINT LBP,BEAM,DRAFT

orders printing of the values of the three named variables.

(g) DUMP orders printing of the database title, if any, and all variables and their values. The printing is done at the terminal.

(h) TITLE gives the database a title. For example

    TITLE 'NA 470 DESIGN DATA'

establishes the name indicated as the title.

(i) SEND is similar to DUMP, except that it accomplishes the printing at the place specified in the command. For example

| | |
|---|---|
| SEND *PRINT* | orders printing at the line printer |
| SEND Filename | orders storage of the dump in the user file Filename |

When *PRINT* is chosen, a routing command to either NUBS or the computing center must have been given while MTS was in control.

(j) DONE stops the editor, and returns the user to the DEX.MAIN menu.

7. BEGIN

This command is used to initiate running of a module. If the module is in the SPIRAL library, the command is

BEGIN MODNAME

where MODNAME is the exact designation of the module in the library. For example

BEGIN POWER    loads and runs the SPIRAL module POWER

BEGIN can also be used to initiate running of any module whose object code is in an MTS file. In this case, the command is

BEGIN TEST Filename

where Filename contains the object code. For example:

BEGIN TEST N719:PROB2.0    loads and runs the program whose object code is found in the file PROB2.0 belonging to user N719.

To repeat the running a module just terminated, give the command

BEGIN SAME

8. CONTINUE

This command returns control from the Design Executive to the current suspended module. For example, suppose you are being prompted to enter an item from a module menu, not having the documentation before you, you decide to use the DISPLAY command to view the module menu. Once you have seen the menu, you wish to return to the same point in the module, this is accomplished with the command CONTINUE.

9. MTS

This command returns control to MTS, but does not unload any part of the SPIRAL system. Return to SPIRAL is accomplished by the MTS command $RESTART.

10. QUIT

This command returns control to MTS and completely unloads SPIRAL.

&

This symbol (ampersand) is not found in any menu, but is a special key for returning to the menu DEX.MAIN. Its use is referred to at several appropriate spots in this memo.

## IV. LIBRARY MODULES

There is a standard format for SPIRAL modules, built around the menu MOD.MAIN. Whenever feasible, modules are written with this format, so that users should expect to confront the standard MOD.MAIN when running any module. This menu consists of the following commands:

ENTER-DB - enters database values into the module
INPUT - prompts user for input from the terminal
DEFAULT - prompts user for changes in default values
COMPUTE - performs the calculations within a module
FILL-DB - fills current database with module values
PRINT - prints module output at the terminal
SEND - sends module output to the file or device specified
DRAW - displays graphic output on the terminal
SAVE - sends graphic output to the file specified
STOP - terminates execution of module and returns control to DEX

The module normally leads the user to this menu at appropriate times during a run, but the user can order a return (you've made a mistake, perhaps, and want to back up) at any time by typing $ENDFILE or Control-C.

### 1. ENTER-DB

This command is chosen by the user if there is database information that the module needs as input. It in turn may lead to a menu whose items are the variable names to be input. This menu may also contain the choice ALL, which causes input of all the variables without individual action by the user.

### 2. INPUT

This command is given to initiate input from a terminal by the user. It in turn will lead to a menu whose items will include ALL, and REQUIRED. The command INPUT may be followed on the same line by ALL, REQUIRED, or a single item from the INPUT menu. If ALL is typed, the module asks the user for all of the inputs specified on the INPUT menu.

Many modules have REQUIRED data, which is general information that a user would not be expected to keep in a database. It therefore must be supplied by the user when running the module. Check module documentation to see if there is data in this category.

Whenever data is asked for by a module, you may be guided by the following rules:

(a) Whole numbers may be entered without decimal points whether they are integers or real.

(b) A number may be entered anywhere on the line, since there are no format specifications.

(c) When asked for more than one item, type them all on the same line, separated by either a comma or a space.

(d) If you make an error (e.g. typing only four values when five are asked for), a prompting message will advise how a correction is to be made.

3. DEFAULT

This word applies to those constants incorporated within the module that a user may change. The command DEFAULT may, in general, be followed on the same line by ALL, or by the name of the item to be changed, When ALL is given, the module asks the user for values of each of the constants that are in the DEFAULT category.

4. COMPUTE

The choice of COMPUTE causes the module to proceed with its calculations, and, of course, should be chosen only after the several categories of data discussed above are taken care of. In some modules, COMPUTE may designate a menu by which computing options are chosen. For example, the POWER module calculates effective hosepower by either a Series 60 or a Taylor Series method, and must be informed by the user which is desired.

5. FILL-DB

Some modules will store their output in a database file should this command be given. A user planning to take advantage of this feature should have a database file previously created and initialized. However, variable names need not be created in advance as this step is performed automatically.

6. PRINT

PRINT orders the printing of module output at the user's terminal. It may be followed on the same line by one of the items from menu PRINT. Typically the items of ALL or RESULTS are chosen. The former will cause the printing of all information available from the module while the latter will just cause the printing of the results of the modules' calculations.

7. SEND

This command may be used to print module output at a location other than the user's terminal. It takes on the following form.

SEND WHAT WHERE

WHAT refers to an item from the menu PRINT; WHERE refers to the destination, such as a file or device name. If the output is to be sent to the line printer (*PRINT*) the MTS routing commands should be given previous to the SEND command.

8. DRAW

This command causes the display of graphic output from the module. It may lead to a menu containing several options. The user must be signed-on to a terminal with graphics capability in order to achieve benefit from this command.

9. SAVE

This command is used to route graphic output to a file so the user may view it later or queue the file for a CAL-COMP plot. The command takes on the following form.

SAVE WHAT SCALE WHERE

WHAT refers to an item from the menu DRAW; SCALE refers to the size that the object is scaled to (i.e. 7.5 for 7.5 inches) and WHERE refers to the filename where the graphic output is to be routed.

10. STOP

This command causes termination of the module and returns control to the Design Executive. The user then must enter an item from the menu DEX.MAIN.

## V. STANDARD MENUS

There are 4 standard menus on SPIRAL. Items on other menus are different in each module. The standard menus are:

| MENU DEX.MAIN | | MENU MOD.MAIN | |
|---|---|---|---|
| 1. | LIBRARY | 1. | ENTER-DB |
| 2. | HELP | 2. | INPUT |
| 3. | DISPLAY | 3. | DEFAULT |
| 4. | ALTER | 4. | COMPUTE |
| 5. | DATABASE | 5. | FILL-DB |
| 6. | EDIT-DB | 6. | PRINT |
| 7. | BEGIN | 7. | SEND |
| 8. | CONTINUE | 8. | DRAW |
| 9. | MTS | 9. | SAVE |
| 10. | QUIT | 10. | STOP |

| MENU EDIT-DB | | MENU DEX.ALTR | |
|---|---|---|---|
| 1. | CREATE | 1. | KEYBOARD |
| 2. | STORE | 2. | GRAPHIC |
| 3. | DELETE | | |
| 4. | COMMENT | | |
| 5. | EXPLAIN | | |
| 6. | PRINT | | |
| 7. | DUMP | | |
| 8. | TITLE | | |
| 9. | SEND | | |
| 10. | DONE | | |

# USER'S MANUALS
# FOR THE
# INTERACTIVE COMPUTING SYSTEN
# SPIRAL

University of Michigan
Department of Naval Architecture
and Marine Engineering

June 1979

## ALTER

### I. ABSTRACT

This module develops a new set of offsets from the given offsets of a "parent hull" according to changes in length and/or depth and/or beam. These changes are given as input by the user, along with the "parent" offsets and station spacing. For his "parent hull", the user may choose from a given library of basic ship forms, or input his own SHCP offset file.

### II. Menu INPUT

ALL - Prompt the user for all input data

PARENT - Enter the offset location

Menu PARENT (See note VII)

| | | |
|---|---|---|
| MARINER | BARGE1 | TUGBOAT 2 |
| LAKER905 | BARGE2 | |
| SUPPLY | TUGBOAT1 | |

MYSHIP - User enters his own offset file and specifications.

REQUIRED - Desired LBP, BEAM, DEPTH.

### III. Constraints on Input

| | Minimum | Maximum |
|---|---|---|
| LBP | 0+ | 1500 |
| BEAM | 0+ | 200 |
| DEPTH | 0+ | 100 |
| STASPC | 0+ | LBP/10 |

### IV. Menu DEFAULT

LINEAR - linear longitudinal distribution of beam change

PARBOLIC - parabolic longitudinal distribution of beam change

### V. Menus PRINT and SEND

ALL - prints both of the following

VESSEL - Prints ship particulars

OFFSETS - Prints offsets

SHCPFILE - Offset file in SHCP format

If PRINT or SEND is invoked before COMPUTE, the parent offsets and/or specifications will be printed. After COMPUTE has been called, the current "offspring" offsets and/or specifications will be printed.

VI. Entering and Filling the Database

Offsets and specifications will be read from the database with ENTER-DB. The user will be prompted to input any values not found on the database. With FILL-DB, the "offspring" offsets and specifications will be stored on the database. The user must be careful to change databases after ENTER-DB if he does not want to replace the "parent" offsets with the "offspring" offsets.

VII. Library of Basic Hull Forms

Currently offered basic hull forms for menu PARENT are as follows:

| | | | |
|---|---|---|---|
| MARINER - | LBP = 520 | DEPTH = 46.22 |
| | BEAM = 76 | STASPC = 26 |
| LAKER905 - | LBP = 1000 | DEPTH = 50 |
| | BEAM = 105 | STASPC = 10 |
| SUPPLY - | LBP = 214 | DEPTH = 19.5 |
| | BEAM = 50 | STASPC = 17.833 |
| BARGE1 - | LBP = 539.75 | DEPTH = 40 |
| | BEAM = 85 | STASPC = 11.0153 |
| BARGE2 - | LBP = 525 | DEPTH = 45 |
| | BEAM = 75 | STASPC = 26.25 |
| TUGBOAT1 - | LBP = 96 | DEPTH = 14 |
| | BEAM = 33 | STASPC = 9.6 |
| TUGBOAT2 - | LBP = 123 | DEPTH = 15 |
| | BEAM = 31.852 | STASPC = 10.25 |

# BUILDING COST
(BCOST)

## I. ABSTRACT

This module computes the building cost of tankers, bulk carriers, dry cargo ships, container ships and SEABEE vessels. The inputs are the basic ship dimensions and the weights of hull steel, machinery, and outfitting. In addition, the hull engineering weight is required for dry cargo ships.

## II. Menu INPUT

ALL - Prompts the user for all input data

LBP - Length between perpendiculars

BEAM, DEPTH

SHP - Shaft horsepower

WSTEEL - Hull steel weight

WMACH - Propulsion machinery weight

WOUTFIT - Outfitting weight

WHULLENG - Hull engineering weight (dry cargo only)

REQUIRED -

- Menu POWER
  - STEAM - Steam propulsion machinery
  - DIESEL - Diesel propulsion machinery
- Menu SHIPTYPE
  - TANKER,BULKER,DRYCARGO,CONTANER,SEABEE

## II. Constraints on Input

| | minimum | maximum |
|---|---|---|
| LBP | 150 | 1500 |
| BEAM | 25 | 200 |
| DEPTH | 15 | 150 |
| SHP | 500 | 60,000 |
| WSTEEL | 0 | --- |
| WMACH | 0 | --- |
| WOUTFIT | 0 | --- |
| WHULLENG | 0 | --- |

IV. Menu DEFAULT

ALL - Standard

PSTEEL - Steel price, $/ton (325.)

POUTF+HE - Outfitting and hull engineering price, $/ton, tankers (2300.)

WAGERATE - Shipyard wage rate, $/hour (10.00)

PROFIT - Shipyard profit fraction (0.05)

PHULLENG - Hull engineering price, $/ton (3500.)

V. Menu PRINT

ALL - Prints all of the following

VESSEL - Prints input particulers

POWERING - Prints powerplant type and SHP

SHIPTYPE - Type of vessel

RESULTS - Results of cost calculations

VI. Menu SEND - Same as above

VII. Menu ENTER-DB

ALL,LBP,BEAM,DEPTH,SHP,WSTEEL,WOUTFIT,WMACHINERY,WHULLENG
(Same descriptions as INPUT)

VIII. Menu FILL-DB

ALL,LBP,BEAM,DEPTH,SHP

PRICE - Computed building cost

SHIPTYPE - Vessel type and powerplant type

WEIGHTS - Steel weight, machinery weight, outfit weight, and hull engineering weight

IX General notes concerning use of the database

This module is designed to use the same database as the Economics Module and the Weights Module. It should be run after the Weights Module, to make use of computed weights, and before the Economics Module, to supply the building cost for economic evaluations.

# DAMAGED STABILITY

(DAMAGE)

## I. ABSTRACT

The purpose of this module is to supply the user with the cross curves of damaged stability of a vessel. Results may also be obtained for the intact ship and individual compartments. This module is designed to use a user's database generated by the Ship Hull Verification module.

## II. Menu INPUT

<u>TRIM</u> - Trim of the vessel under the damaged condition

<u>VCG</u> - Vertical cg of the vessel

<u>DRAFTS</u> - Drafts of the vessel for which the cross curves of damage stability are desired (a maximum of 7 drafts are permitted)

<u>HEELS</u> - Heel angles that the calculations will be performed at. (A maximum of 10 heels are permitted. Heels greater than ±90° are prohibited)

<u>ADD</u> - Allows the user to add a compartment description

<u>ALTER</u> - Allows the user to alter a compartment description

<u>DELETE</u> - Allows the user to delete a compartment description

<u>GROUPS</u> - Allows the user to input the groups of flooded compartments for which the calculations are to be performed. Up to 5 groups of compartments are allowed in the module

## III. DEFAULT

The only available option under the default menu choice is the ship name, number and date.

## IV. ENTER-DB

The following information is extracted from the user's database through the ENTER-DB command.

| Variable | Array Size | Description |
|---|---|---|
| SDT | 7134 | Ship data table |
| STATN | 41 | Station numbers |
| NP | 41 | Number of points/station |
| X | 41 | Long station locations |
| DRAFT | NA | Design draft |
| LBP | NA | Length between perp. |
| TRIM | NA | Design trim |
| STASPC | NA | Station spacing |
| NSTAT | NA | Number of stations |
| CUFTON | NA | Cubic feet of water per ton |
| XMID | NA | Amidships |
| LOA | NA | Length overall |
| VCG | NA | Vertical cg |

The user will then encounter the menu CMP.DATA, and is prompted for the location of the compartment descriptions. The options are FILE or TERMINAL. If the user specifies a file for the location, the compartments should be in a file in the following format. (One per line.)

(I3, 6A4, I2, F4.2, 6F7.2, I2, I3)

| | |
|---|---|
| ID number | I3 |
| Compartment name | 6A4 |
| Symmetry indicator | I2 |
| Permeability | F4.2 |
| X1D | F7.2 |
| X2D | F7.2 |
| Y1D | F7.2 |
| Y2D | F7.2 |
| Z1D | F7.2 |
| Z2D | F7.2 |
| 1OFF = 0 | I2 |
| 1RNOFF = 0 | I3 |

The same compartment coordinates and format are used by DAMSS, and more detail is given in its user manual.

Note that the offsets submittal and the runoff option are not allowed in this module.

No additional information is saved by the FILL-DB menu choice.

## V. Menu PRINT

The options available under the Print option are:

VESSEL: Prints the design condition of the vessel

COMPTS: Prints the intact ship properties and the compartment properties at each specified condition

RESULTS: Prints the cross curves of Damaged Stability at the specified conditions

ALL: Prints all of the above

# DAMAGE STATICAL STABILITY

(DAMSS)

## I. ABSTRACT

With input of VCG, heel angles, compartment descriptions, and damaged compartment groups, this module performs damaged statical stability calculations. For each condition at every heel angle draft, trim, volume, LCB, righting arm, and compartment properties of the damaged ship are computed such that the remaining intact portion of the ship has displacement and LCB consistent with the design condition.

## II. Menu INPUT

<u>ALL</u> - Prompts user for all input data

<u>POLE</u> - Enter VCG (pole height)

<u>HEELS</u> - Enter all heel angles. Maximum of ten (10)

<u>ADD</u> - The user desires to enter a compartment description

<u>ALTER</u> - The user desires to change an existing compartment description

<u>DELETE</u> - The user desires to delete a compartment description

<u>GROUPS</u> - Enter damaged condition compartment groups

<u>REQUIRED</u> - Prompts user for all input data with the exception of menu choice POLE. The VCG (pole height) is defined in the database

## III. Menu DEFAULT

Through this MOD.MAIN menu choice the user enters the ship name, ship number, and the date.

## IV. Menu CMP.DATA

This menu appears in the module's database subroutine. It asks the user how he is to enter the compartment descriptions.

<u>TERMINAL</u>

<u>FILE</u> - The file must be in the format which follows, with one line for each compartment described:

| quantity | | format | columns |
|---|---|---|---|
| i.d. number | | I3 | 1-3 |
| compartment name | | 6A4 | 4-27 |
| symmetry indicator<br>-1 left side only<br>0 both sides<br>+1 right side only | | I2 | 28-29 |
| permeability | | F4.2 | 30-33 |
| forward bulkhead location | | F7.2 | 34-40 |
| after bulkhead location | | F7.2 | 41-47 |
| inboard bulkhead location | | F7.2 | 48-54 |
| outboard bulkhead location | | F7.2 | 55-61 |
| lower deck height | | F7.2 | 62-68 |
| upper deck height | | F7.2 | 69-75 |
| index for runoff | 0 = none<br>1 = runoff | I3 | 76-78 |

A maximum of forty (40) compartments can be submitted.

If no limiting longitudinal bulkhead or deck is present submit the quantity as 9999.99 .

The index for runoff applies only when a tank containing a liquid is damaged. For this case, submit the specific gravity of the liquid in percent. (i.e. 94 not 0.94)

## V. Constraints on Input

The user must supply the module with a database generated by the SHIPS HULL VERIFICATION module (VERIFY).

Other constraints are

(1) Maximum number of heel angles of 10

(2) Maximum number of damaged conditions of 5

(3) The number of compartment descriptions is limited to 5

(4) The compartment name is limited to 24 characters and can not include inbedded blanks

## VI. Database

The following information will be retrieved from the user's database upon the menu choice ENTER-DB.

SDT - Ship's database supplied by VERIFY module

STATN - Array of station numbers

NP - Number of points per station (interpolated offsets)

X - Array of longitudinal station distances (from FP)

DRAFT - Design draft of vessel

LBP - Length between perpendiculars

TRIM - Design trim of vessel

NSTAT - Number of stations

CUFTON - Cubic feet of water per ton of displacement

XMID - Midship location (distance from FP)

LOA - Length overall

VCG - Vertical center of gravity (pole height)

The FILL-DB menu choice does not save any additional database information.

## VII. Menu PRINT

INITIAL - Display of the input variables

COMPART - Input compartment descriptions

DAMAGED - Compartment and intact ship properties at balance condition

NET - Net damaged ship properties

RESULTS - The user is supplied all the information on the three menu items COMPART, DAMAGED, and NET

ALL - The user is supplied all the information on the three items COMPART, DAMAGED, and NET

# DIGITIZE

## I. ABSTRACT

This module is used with the digitizer to read offsets from a hull lines plan into a file or database.

## II. SPECIAL NOTE

This module is not in the standard SPIRAL format. But follow instructions given here, and you will do okay.

## III. MODULE ACCESS

1. At DEX.MAIN enter BEGIN DIGITIZE.
2. At MOD.MAIN enter INPUT ALL.
   Respond to prompts and locate the menu sheet and the bodyplan drawing on the digitizer tablet. Make sure that the Control Unit is set properly - that is, with the "POINT" button depressed only.
3. At MOD.MAIN enter COMPUTE.
4. At ID.MAIN enter CHOOSE DRAWING by placing the cursor ("mouse") on the proper square and depressing the 1 button.
5. At ID.VIEW enter BODYPLAN in a similar manner as above, but with the 3 button.
6. Enter the correct scale at the terminal (in./ft.).
7. Orient the axes with the cursor, using the Z button, as directed by the message on the terminal screen.
8. At ID.MAIN enter DEFINE STATION with the cursor.
9. Enter the station number at the terminal.
10. When the " " prompt appears on the terminal screen, begin digitizing the station contour. Use the cursor with the Z button to enter points at fairly even intervals along the contour, more closely spaced where the curvature is more extreme. If a point is a discontinuity, or knuckle, then after inputting that point, enter KNUCKLE at ID.FLAG, using the 2 button on the cursor, and then resume digitizing. After entering the last point on the contour, enter LAST POINT at ID.FLAG using the 2 button. If a point is mistakenly input, then immediately after, enter DELETE at ID.FLAG using the 2 button.
11. Repeat from step 8 for each station of the ship.
12. When all stations have been defined, enter EXIT at ID.MAIN, using the 1 button.

13. At MOD.MAIN enter DRAW at the terminal to check over the inputted stations. If corrections are needed, go back to step 3 and re-enter the station which is incorrectly defined. (This time you will not be prompted for scale and drawing locations.)
14. To send offsets to an MTS file, enter SEND SHCP filename at MOD.MAIN.
15. To send offsets to the database, enter FILL-DB SHCP at MOD.MAIN, or FILL-DB FAIRING if fairing module is to be used.
16. To send the bodyplan to the Calcomp plotter, enter SAVE filename. It will be drawn to fit on an 8 1/2 x 11 sheet of paper. Be sure to RUN*CCQUEUE before signing off. The filename must be a permanent file.
17. To obtain printed output, enter PRINT VESSEL and/or PRINT SHCP at MOD.MAIN. To get a hard copy, enter SEND VESSEL *PRINT* and/or SEND SHCP *PRINT* at MOD.MAIN.
18. At MOD.MAIN enter STOP.

IV. HELPFUL TIPS

1. If at any time while defining a station you would like to start that station over, simply enter DEFINE STATION at ID.MAIN with the 1 button and restart that station.
2. You can also "bail out" in the above case, should you get completely frustrated, by entering EXIT at ID.MAIN with the 1 button.
3. A station contour may be repeated for other station numbers in a parallel mid-body. After describing completely the typical station, the user should enter COPY STATION at ID.MAIN with the 1 button, and then input the new station number at the terminal. If a station by that number already exists, the user will be asked if he wants to replace it.
4. If at some later date you find you would like to alter or add stations to your ship, you may do so by having stored the offsets on a database previously, and by entering ENTER-DB SHCP at MOD.MAIN: then entering INPUT REQUIRED at MOD.MAIN: then proceeding from step 3. The offsets will not be read from an MTS file in SCHP format.
5. A bell will ring each time the ID-cursor is employed--wait for it!

Digitizer Menu Sheet

③ Menu ID.VIEW

| | | |
|---|---|---|
| *Body Plan* | *Half-Breadth Plan* | *Profile* |

② Menu ID.FLAG

| | |
|---|---|
| *Last Point* | *Delete* |

| | | | | |
|---|---|---|---|---|
| *Normal* | *Knuckle* | *Tangent* F→C | *Tangent* C→F | *Skip* |

① Menu ID.MAIN

| | | | | |
|---|---|---|---|---|
| *Choose Drawing* | *Define Station* | *Copy Station* | *Define Control Line* | *Exit* |

# ECONOMICS
## (ECON)

## I. ABSTRACT

This module computes operating costs and measures of merit for conventional tankers, dry cargo and bulk carriers, container ships, and SEABEE vessels. The user must supply ship particulars, building cost, some weight estimates, and basic economic expectations, i.e., lifetime, tax rate, and interest rate (yield). In some cases these economic factors are assumed. Route and cargo information is needed for all except tankers.

## II. Menu INPUT

- ALL - Prompts the user for all input data
- LBP - Length between perpendiculars
- BEAM, DRAFT, DEPTH - Standard definitions
- DISPL - Displacement in long tons
- CB - Block coefficient (fraction)
- SHP - Shaft horsepower
- SEASPEED - Average cruise speed in knots
- RANGE - Endurance at sea, nautical miles
- CREWSIZE - Number in crew
- POWRTYPE -
  - Menu POWER
    - STEAM - Steam propulsion machinery
    - DIESEL - Diesel propulsion machinery
- REQUIRED -
  - Menu SHIPTYPE
    - TANKER, BULKER, DRYCARGO, CONTANER, SEABEE
  - TANKER option:
    - Menu TANKDATA
      - ALL - Standard
      - LITESHIP - Lightship weight in long tons
      - WFUEL - Fuel weight, L.T.
      - WLUBE - Lube oil weight, L.T.
      - MISCDWT - Miscellaneous deadweight, L.T.
      - CARGODWT - Cargo deadweight, L.T.
      - TOTALDWT - Total deadweight, L.T.

ECONDATA - Ship life, yield, tax rate
PRICE - Building cost, $

BULKER and DRYCARGO options:
Menu BULKDATA
ALL - Standard
CM - Midship coefficient (fraction)
PRICE - Building cost, $
WSTEEL - Hull steel weight, L.T.
WMACH - Machinery weight, L.T.
WOUTFIT - Outfitting weight, L.T.
WHULLENG - Hull engineering weight
ROUTE - Route and cargo information - see item III

CONTANER and SEABEE options:
Menu CONTDATA
ALL - Standard
CM - Midship coefficient (fraction)
SEADAYS - Number of days at sea per year
PORTDAYS - Number of days in port per year
WCONTANR - Average container weight, L.T.
LIFE - Ship life expectation, yrs.
PRICE - Building cost, $
ROUTE - See item III

III. Route and Cargo Input (not needed for Tankers)

Menu PORTTYPE
OCEAN, LAKES, SEAWAY

Menu OCEAN
TOKYO, BUENOSAR, NAPLES, HAMBURG, CAPETOWN, NEW YORK,

Menu LAKES
BUFFALO, CLEVELAN, TOLEDO, DETROIT, BAYCITY, PORTHURO, DULUTH, MILWAUKE, CHICAGO, GARY

Menu SEAWAY
HALIFAX, SYDNEY, SEPTISL, BAIECOM, QUEBEC, MONTREAL, KINGSTON, ROCHESTE, TORONTO, HAMILTON, ST.CATH.

Menu CARGO (BULK and DRYCARGO only)

AUTO(-MOBILES), GRAIN, CEMENT, COAL, PIG-IRON, MACHINRY, OIL, TIMBER, LIMESTONE, BALLAST, EXTRA(NOT DEFINED)

Menu VESSEL (CONTAINER and SEABEE only)

OCEAN.O - Oceangoing vessel

OCEAN.S - Oceangoing vessel transisting the Seaway

OCEAN.SW - Oceangoing vessel transisting the Seaway and Welland Canal.

FEEDR.S - Feeder vessel transisting the Seaway

FEEDR.SW - Feeder vessel transisting the Seaway and Welland Canal

FEEDR.GL - Feeder vessel transisting the Great Lakes only

CONTAINER DATA (CONTAINER and SEABEE only)

1. Number of containers to be loaded at each port
2. Port of destination for each group of containers
3. Estimated time per container lift
4. Estimated cost per container lift
5. Number of containers to be unloaded at each port

The user is prompted for this data for each port given on the route description. Entry of ϕ for (1) above will signal for end of loading operation at that port.

IV. Constraints on Input

| | minimum | maximum | |
|---|---|---|---|
| LBP | 150 | 1500 | |
| BEAM | 25 | 200 | |
| DRAFT | 13 | 80 | |
| DEPTH | 15 | 150 | |
| DISPL | VOL/36.1 | VOL/34.9 | (VOL=L*B*T*CB) |
| CB | .50 | .90 | |
| SHP | 500 | 60,000 | |
| SEASPEED | 5 | 40 | |
| RANGE | -- | -- | |
| CREWSIZE | -- | -- | |
| LITESHIP | 1000 | 50,000 | |

| | minimum | maximum |
|---|---|---|
| WFUEL | 500 | 50,000 |
| WLUBE | 5 | 30,000 |
| MISCDWT | 0+ | 50,000 |
| CARGODWT | 1000 | 500,000 |
| ECONDATA | | |
| shiplife | 0+ | 50 |
| yield | 0+ | .50 |
| taxrate | 0+ | .99 |
| PRICE | 100,000 | 999,999,999 |
| CM | .60 | .99 |
| WSTEEL | 1000 | 60,000 |
| WMACH | 500 | 15,000 |
| WOUTFIT | 500 | 15,000 |
| WHULLENG | 0 | 15,000 |
| SEADAYS | 0+ | 365 |
| PORTDAYS | 0+ | 365 |
| WCONTANR | 0+ | 99 |

V. Menu DEFAULTS (BULKER and DRYCARGO)

ALL - Standard

LIFE - Ship life expectancy (20)

INTEREST - Interest rate, after tax (.10)

YEAR - Current year for evaluation (1978)

INFLRATE - Inflation rate, per year (.05)

OPSEASON - Operating and capital recovery days per season (365)

VI. Menu DEFAULTC (CONTANER and SEABEE)

ALL - Standard

PILOTFEE - Piloting fee, oceangoing vessels (6000)

SWCONFEE - Seaway container fee (.9)

SWTOLL - Seaway toll (.04)

WELLFEE - Welland Canal fee (1600)

FUELRATE - Average fuel price (80)

INTRATE - After-tax interest rate (.08)

TAXRATE - Corporate income tax (.48)

M&RFACT - Maintenance and repair cost factor (1.0)

INSFACT - Insurance cost factor (1.0)

OPERFACT - Operating cost factor (1.0)

DLAYFACT - Delay time factor (0.0)

VARWT - Variable weight (10.0)

VII. Menu PRINT

ALL - Prints all input, defaults, and output

INPUT - Prints input only

DEFAULTS - Prints defaults only

ECONOMIC - Prints computed economic evaluation results (see item XII)

VIII. Menu SEND

ALL, INPUT, DEFAULTS, ECONOMIC - same as above

IX. Menu ENTER-DB

ALL, LBP, BEAM, DRAFT, DEPTH, DISPL, CB, SHP, SEASPEED, RANGE, CREWSIZE, SHIPTYPE (Same descriptions as INPUT)

SPECIFIC - Branches to data for each shiptype.

Menu TANK-DB

ALL, LITESHIP, WFUEL, WLUBE, MISCDWT, CARGODWT, TOTALDWT, LIFE, INTEREST, TAX, PRICE - (Same descriptions as TANKDATA)

Menu BULK-DB

ALL, CM, PRICE, WSTEEL, WMACH, WOUTFIT, WHULLENG - (Same descriptions as BULKDATA)

Menu CONT-DB

ALL, CM, SEADAYS, PORTDAYS, WCONTANR, LIFE, PRICE - (Same descriptions as CONTDATA)

X. Menu FILL-DB

Same items as ENTER-DB

Menu TANK-FL

Same items as TANK-DB

Menu BULK-FL

Same items as BULK-DB

Menu CONT-FL

Same items as CONT-DB

XI. General notes concerning use of the database.

1) No information on routes or loading can be stored or retrieved from the database. These data must all be entered interactively.

2) The following items may be stored on the database by the Weights module, for use with this module:

LBP,BEAM,DRAFT,DEPTH,CB,CREWSIZE,DISPL,
WHULLENG,WSTEEL,WMACH,WOUTFIT,LITESHIP

3) The following items may be stored on the database by the Building Cost module, for use with this module:

LBP,BEAM,DEPTH,SHP,PRICE

4) After entering the database, when running this module, the user must input REQUIRED data (Menu INPUT) before computing.

XII. General notes concerning printed output.

1) All computed results are approximations based on generalized models of the various shiptypes.

2) Under the BULKER and DRYCARGO options, the economic figures listed under individual trade routes are for that particular leg only, for one year of regular operation. Comprehensive figures are listed under "Operating Information." In the case that there is a purely ballasted leg, the economic figures may need to be altered to reflect the loss of revenue on that leg. There is no attempt by the program to distribute this loss over other cargo legs.

# FAIRING
# (FAIR)

## I. Abstract

This module reads a set of preliminary design offsets and grid specifications to generate a graphical representation of the hull surface. The lines are drawn on the screen of the graphics terminal and the user is given the opportunity to alter them to smooth out any unfairness. The faired offsets may be stored on the database or in SHCP format for use with other modules; a table of offsets may be printed and drawings of any or all of the views may be sent to the Calcomp plotter at a user-specified scale. All dimensions are in decimal feet.

## II. Menu INPUT

ALL - Prompts the user for all items below.

SHIPNAME - Up to 20 characters, for identification.

LBP,BEAM,DRAFT - Standard definitions.

MIDSHIP - Distance from FP to amidships.

STASPC - Station spacing, which, when multiplied by the station number, gives the distance of the station from the FP.

OFFSETS - Offsets file name, preliminary offsets. These may be as generated by LINESGEN, in which case a profile and deck plan will be generated by this program. Otherwise the offsets must be prepared by the user as specified in section III below.

REQUIRED - Grid lines must be specified here:

MENU GRIDLINE

ALL - Define all gridlines below.

WATRLINE - Up to 10 waterline heights may be given.

BUTTOCK - Up to 10 half-breadths may be given.

DIAGONAL - Up to 5 diagonal angles may be given. These angles are given in degrees from the centerline to the diagonal plane and must be between 0 and 90 degrees, exclusive.

## III. Offsets Format

The input offsets, unless generated by LINESGEN, are read with the format: 10A1, 3F10.3, 3I1. The Alpha field contains the station number, in numerical digits, with a decimal point, or the contour name in the case of control lines (sheer line, profile, chines, tangents, etc.) This field is blank for all except the first point of each curve. For control lines an 8-character name is allowed; the first two characters of each must be one of the following pairs:

FC - for a fore-body curve which is not a knuckle line.

AC - for an aft-body curve which is not a knuckle line.

FD - for a fore-body curve which is a knuckle line.

AD - for an aft-body curve which is a knuckle line.

The remaining six characters are arbitrary, but should be descriptive of the curve being defined. Only these last six characters are printed in the table of offsets. Curve definition should lap midship by one station only.

The three F10.3 data fields are for the height, half-breadth, and longitudinal location for each data point, in that order. Longitudinal dimensions need not be given for offsets at stations.

The three integer fields are for flags to indicate the type of data-point. The first field relates to the transverse view, the second to the plan view, and the third to the elevation view. Only the first field need be used for station input. The following flags are to be used to identify data points:

0 - Continuous point.

1 - Discontinuous point.

2 - Tangency point, curve going from flat to curve.

3 - Tangency point, curve going from curve to flat.

4 - Skip point in curvefit for this view.

5 - Last point on curve.

A sample set of offsets may be seen in Appendix A of this documentation. Note that station definitions should precede control line definitions, forward stations should precede aft stations, and all forward control lines should precede aft control lines. A full centerline profile and deck plan <u>must</u> be included in the control lines so that buttocks, waterlines, and diagonals can be terminated. A maximum of 25 stations forward, 25 stations aft, 10 control lines forward, and 10 control lines aft may be defined.

IV. Constraints on Input

| | Minimum | Maximum |
|---|---|---|
| LBP | 0. | 1500. |
| STASPC | 0. | LBP/10. |

BEAM,DRAFT,Gridlines - Must be compatible with offsets given.

V. Menu DEFAULT

XMARGIN - width of blank paper desired on ends of Calcomp plots
PLOTSIZE - Dimension for a square region within which the user wants his

drawing plotted, useful for generating report drawings when the scale factor is not important. The user should then enter "Ø." for a scale factor when prompted under MOD.MAIN menu choice SAVE, Section XII; otherwise the scale given will override the PLOTSIZE specification.

DIAGORIG - Dimension at centerline from which the diagonal planes should emanate. Unless specified here, the draft dimension will be used.

VI. Menu COMPUTE

ALL - All of the following computations will be performed in sequence.

HULLFORM - The station and control line offsets input is interpolated and curvefitted.

LINES - Waterline, buttock, and diagonal planes are passed through the hull form and their intersection contours stored. The lines are then drawn on the screen and control passes to the fairing mode, section VII below.

AREAS - If the hullform has been successfully generated, the sectional area curve is computed with this menu choice.

OFFSETS - The user is prompted to specify approximately how many gridlines he wants for his loft offsets. The program then uses this as a guideline to slice the hullform up by 'nice' increments and to display the resulting lines on the screen. The user is again given control in fairing mode so that he may do any final smoothing that is needful. The user is then prompted to enter a file or device name on which to write offsets in SHCP-type format, for use with other SPIRAL modules.

VII. Fairing Mode

When control is passed to the fairing section, the screen is cleared, the lines are drawn, and the cross-hair cursor appears on the screen. While in the fairing section, interaction is only achieved via the cross-hair cursor. Command choices are made by depressing the key corresponding to the first letter of the command word. In most cases, this merely specifies the command choice, and further graphical input is necessary to complete the chosen task. There are 13 commands to choose from in the fairing section, some of them leading to an additional support command. When a major command word is expected by the program, a bell will ring at the terminal. The user must be very careful to wait for the cross-hair cursor to appear on the screen before depressing any key choice. Commands can not be "stacked" as with alphanumeric input. The following command choices are available to the user to modify his lines:

HELP - Gives brief instructions for operation in fairing mode.

VIEW - The lines are divided internally into six regions: forward body, aft body, forward plan, aft plan, forward elevation, aft elevation. With this command entry, the region within which the cursor lies will be blown up to larger scale to fill the screen. No addition input is required for this command. Note that it is only meaningful when the full lines drawing currently occupies the screen.

ZOOM - Following this command choice, the user is expected to position the cursor over two diagonally opposite corners of a rectangle, entering each with the <RETURN> key. The section within this rectangle is then scaled-up to fill the screen. This command may be chosen repeatedly and the scalings are comprehensive.

CURVE - Following this command choice, the user is expected to position the cursor on an unambiguous section of the curve which he intends to modify. If the positioning is clear enough for the program to identify the curve, then the data-points will be marked on the curve with asterisks. If the positioning is ambiguous, then the bell will again ring and the command must be chosen again, and a new position given on the curve. If it is impossible to find a good position, due to closeness of lines, then ZOOM should be used to scale up the drawing until the curves are sufficiently distinct. A view must have been chosen prior to calling this command.

ADD - Allows the user to add an additional data point to a curve which he has selected with command CURVE above. After selecting this command, he is expected to position the cursor over the location of the point to be added, and <RETURN>. The resulting contour will then be drawn on the screen, without erasing the current picture. Points may be added with repeated calls to this command until no more space is available in the associated arrays.

DELETE - Allows the user to delete a point from the curvefitting of the data points of a curve previously selected with command CURVE. Following selection of this command, the user must position the cursor over the point to be deleted and <RETURN>. The resulting contour will then be drawn on the screen, without erasure. Then the user must decide whether to approve or undo this alteration. If the altered contour looks acceptable, the user should <RETURN>; otherwise he should enter UNDO, which will restore the curve to its former configuration.

MOVE - Allows the user to move an already existing data point on a curve selected with command CURVE. After selecting this command, the user must position the cursor over the point to be moved, <RETURN>, and then position the cursor over its new location, again <RETURN>. The resulting contour will be drawn on the screen, without erasure. The user must then decide whether to approve or undo this alteration, just as in DELETE.

UPDATE - When the user has made all the necessary alterations to the curve he has chosen to work with, this command will revise the data structure to reflect his alterations. The revised lines drawing will then be drawn on the screen. Unless this command is chosen immediately following the desired ADD, DELETE, and MOVE commands, the alterations will not be stored at all. No further input is needed for this command.

REDRAW - The current picture is redrawn on the terminal screen. No further input is needed for this command.

INSERT - Allows the user to insert a new station into the data structure. After choosing this command word, the user must position the cross-hairs over desired station location and <RETURN>. One of the plan view or elevation view regions must be on the screen when this command is chosen. The station will be added and the revised lines drawn on the screen.

GETOUT - Returns the user to MOD.MAIN. No further input is required for this command.

STOP - Terminates the fairing section and sets a flag to indicate that fairing was successful. Sectional areas and offsets may then be computed, depending on the menu item chosen from menu COMPUTE.

LINES - Redraws the lines drawing on the screen. No further input is needed for this command choice.

The efficiency of this algorithm is very dependent upon the ability of the user to make wise decisions in altering lines to fair the hull. Though a high degree of fairness is quite possible the user should consider the small scale at which he is working and not demand precision out of proportion to scale. It is generally better to begin with a few intelligently placed gridlines, to fair those nicely, and then add more when prompted for them under the COMPUTE menu item OFFSETS. Control will automatically be returned to Fairing to re-check the fairness of the hull.

## VIII. The Database

Filling the database from the Fairing Module results in the storage of the basic hull from at the time of the call to FILL-DB. The ship name, station and control line offsets, and overall dimensions are stored, but grid information is not stored. The database may then be entered for further lines alterations at some later date, in which case the only further input required is the grid information. (At MOD.MAIN, enter INPUT REQUIRED.)

## IX. Menu PRINT

ALL - All of the following are printed in sequence. Note that this would be extremely long - probably should be sent to the line printer rather than the terminal. Also, some of the format is dependent upon carriage control which is not recognised by the terminal printer.

VESSEL - Prints a dated title page with the ship name and ship particulars centered on the page.

OFFSETS - If the fairing was completed and loft offsets generated, then these will be printed in traditional tabular format. Otherwise the offsets will be printed station-by-station.

DRAWSEG - The actual incremental movements made in drawing all of the lines will be printed: stations, waterlines, buttocks, diagonals, and control lines. This is a lot of printing and should be used only when very close data are needed on curve descriptions.

X. Menu SEND

The same items as for PRINT above are available. Here the output will be sent to the specified file or device.

XI. MENU DRAW

LINES - The full lines drawing will be drawn, centered on the screen.

BODY - Only the body plan is drawn, but at the largest scale that will fit on the screen.

PLAN - The half-breadth plan is drawn, with expanded diagonals opposite to the waterlines.

ELEV - The elevation view is drawn.

AREAS - The sectional area curve is drawn.

XII. Menu SAVE

The same items as for DRAW above are available. The user is prompted for a file in which to store information to produce a Calcomp plot of the selected item, and for the scale (in./ft.) at which the plot is to be drawn. The scale must be chosen so that the width of the scaled drawing is within the range of the plotter - approximately 29 inches. The length of the plot is practically unlimited, as long as the corresponding width will fit on the paper. If the scale is given as zero (Ø.Ø), then the plot will be drawn to the default size - either 18" or as specified by the user under the DEFAULT menu.

## Appendix A

### Input Offsets File for a 26-foot Sloop

| | | | | |
|---|---|---|---|---|
| 1 | 0.00 | 3.5 | 0.0 | 1 |
| 3 | | 4.0 | 0.37 | 0 |
| 4 | | 4.97 | 0.85 | 5 |
| 5 | 2.00 | 1.78 | 0.0 | 1 |
| 6 | | 2.0 | 0.25 | 0 |
| 7 | | 3.0 | 0.95 | 0 |
| 8 | | 4.0 | 1.4 | 0 |
| 9 | | 4.92 | 1.77 | 5 |
| 10 | 4.00 | 1.36 | 0.0 | 1 |
| 11 | | 1.83 | 1.0 | 0 |
| 12 | | 2.00 | 1.22 | 0 |
| 13 | | 3.00 | 1.95 | 0 |
| 14 | | 4.00 | 2.38 | 0 |
| 15 | | 4.85 | 2.67 | 5 |
| 16 | 6.00 | 1.07 | 0.0 | 1 |
| 17 | | 1.38 | 1.0 | 0 |
| 18 | | 2.00 | 2.06 | 0 |
| 19 | | 3.00 | 2.80 | 0 |
| 20 | | 4.00 | 3.16 | 0 |
| 21 | | 4.82 | 3.38 | 5 |
| 22 | 8.00 | 0.87 | 0.0 | 1 |
| 23 | | 1.09 | 1.0 | 0 |
| 24 | | 1.45 | 2.0 | 0 |
| 25 | | 2.00 | 2.81 | 0 |
| 26 | | 3.00 | 3.47 | 0 |
| 27 | | 4.00 | 3.81 | 0 |
| 28 | | 4.80 | 3.95 | 5 |
| 29 | 10.00 | 0.80 | 0.0 | 1 |
| 30 | | 0.93 | 1.0 | 0 |
| 31 | | 1.22 | 2.0 | 0 |
| 32 | | 1.73 | 3.0 | 0 |
| 33 | | 2.00 | 3.36 | 0 |
| 34 | | 3.00 | 3.97 | 0 |
| 35 | | 4.00 | 4.26 | 0 |
| 36 | | 4.78 | 4.36 | 5 |
| 37 | 12.00 | 0.75 | 0.0 | 1 |
| 38 | | 0.87 | 1.0 | 0 |
| 39 | | 1.11 | 2.0 | 0 |
| 40 | | 1.51 | 3.0 | 0 |
| 41 | | 2.00 | 3.67 | 0 |
| 42 | | 3.00 | 4.24 | 0 |
| 43 | | 4.00 | 4.45 | 0 |
| 44 | | 4.77 | 4.53 | 5 |
| 45 | 14.00 | 0.86 | 0.0 | 1 |
| 46 | | 0.93 | 1.0 | 0 |
| 47 | | 1.12 | 2.0 | 0 |
| 48 | | 1.46 | 3.0 | 0 |
| 49 | | 2.00 | 3.69 | 0 |
| 50 | | 3.00 | 4.23 | 0 |
| 51 | | 4.00 | 4.40 | 0 |
| 52 | | 4.74 | 4.43 | 5 |
| 53 | 16.00 | 1.02 | 0.0 | 1 |
| 54 | | 1.10 | 1.0 | 0 |
| 55 | | 1.27 | 2.0 | 0 |
| 56 | | 1.68 | 3.0 | 0 |
| 57 | | 2.00 | 3.4 | 0 |
| 58 | | 3.00 | 3.93 | 0 |
| 59 | | 4.00 | 4.12 | 0 |
| 60 | | 4.73 | 4.16 | 5 |

| | | | | | |
|---|---|---|---|---|---|
| 61 | 18.00 | 1.29 | 0.0 | | 1 |
| 62 | | 1.32 | 1.0 | | 0 |
| 63 | | 1.56 | 2.0 | | 0 |
| 64 | | 2.27 | 3.0 | | 0 |
| 65 | | 3.00 | 3.39 | | 0 |
| 66 | | 4.00 | 3.63 | | 0 |
| 67 | | 4.72 | 3.70 | | 5 |
| 68 | 20.00 | 1.55 | 0.0 | | 1 |
| 69 | | 1.65 | 1.0 | | 0 |
| 70 | | 2.12 | 2.0 | | 0 |
| 71 | | 3.00 | 2.63 | | 0 |
| 72 | | 4.00 | 2.94 | | 0 |
| 73 | | 4.71 | 3.04 | | 5 |
| 74 | 22.00 | 1.84 | 0.0 | | 1 |
| 75 | | 2.15 | 1.0 | | 0 |
| 76 | | 3.00 | 1.73 | | 0 |
| 77 | | 4.00 | 2.10 | | 0 |
| 78 | | 4.74 | 2.23 | | 5 |
| 79 | FCPROFIL | 5.02 | 0.0 | -1.4 | 111 |
| 80 | | 2.00 | 0.0 | 1.45 | 442 |
| 81 | | 1.78 | 0.0 | 2.0 | 440 |
| 82 | | 1.36 | 0.0 | 4.0 | 440 |
| 83 | | 1.07 | 0.0 | 6.0 | 440 |
| 84 | | 0.87 | 0.0 | 8.0 | 440 |
| 85 | | 0.80 | 0.0 | 10.0 | 440 |
| 86 | | 0.75 | 0.0 | 12.0 | 440 |
| 87 | | 0.86 | 0.0 | 14.0 | 555 |
| 88 | FCSHEER | 5.02 | 0.0 | -1.4 | 111 |
| 89 | | 4.97 | 0.85 | 0.0 | 000 |
| 90 | | 4.92 | 1.77 | 2.0 | 000 |
| 91 | | 4.85 | 2.67 | 4.0 | 000 |
| 92 | | 4.82 | 3.38 | 6.0 | 000 |
| 93 | | 4.80 | 3.95 | 8.0 | 000 |
| 94 | | 4.78 | 4.36 | 10.0 | 000 |
| 95 | | 4.77 | 4.53 | 12.0 | 100 |
| 96 | | 4.74 | 4.43 | 14.0 | 555 |
| 97 | ACPROFIL | 0.80 | 0.0 | 10.0 | 111 |
| 98 | | 0.75 | 0.0 | 12.0 | 440 |
| 99 | | 0.86 | 0.0 | 14.0 | 440 |
| 100 | | 1.02 | 0.0 | 16.0 | 440 |
| 101 | | 1.29 | 0.0 | 18.0 | 440 |
| 102 | | 1.55 | 0.0 | 20.0 | 440 |
| 103 | | 1.84 | 0.0 | 22.0 | 440 |
| 104 | | 2.16 | 0.0 | 24.27 | 555 |
| 106 | ACSHEER | 4.78 | 4.36 | 10.0 | 111 |
| 107 | | 4.77 | 4.53 | 12.0 | 100 |
| 108 | | 4.74 | 4.43 | 14.0 | 000 |
| 109 | | 4.73 | 4.16 | 16.0 | 000 |
| 110 | | 4.72 | 3.70 | 18.0 | 000 |
| 111 | | 4.71 | 3.04 | 20.0 | 000 |
| 112 | | 4.74 | 2.23 | 22.0 | 111 |
| 113 | | 4.0 | 1.8 | 22.64 | 004 |
| 114 | | 3.0 | 1.085 | 23.525 | 004 |
| 115 | | 2.16 | 0.0 | 24.27 | 555 |
| 116 | ADTANGNT | 0.80 | 0.01 | 10.01 | 111 |
| 117 | | 0.75 | 0.01 | 12.0 | 440 |
| 118 | | 0.86 | 0.01 | 14.0 | 440 |
| 119 | | 1.02 | 0.01 | 16.0 | 440 |
| 120 | | 1.29 | 0.01 | 18.0 | 440 |
| 121 | | 1.55 | 0.01 | 20.01 | 440 |
| 122 | | 1.84 | 0.01 | 22.0 | 440 |
| 123 | | 2.16 | 0.01 | 24.27 | 555 |

## HYDROSTATICS (HSTAT)

## I. ABSTRACT

This module performs the required calculations to obtain the hydrostatic properties of the ship to each given waterline, including sectional areas at each station, to each specified waterline.

The following hydrostatic data will be computed:

1) Change in displacement for one foot trim by stern
2) Displacement, if draft is specified
3) Draft, if displacement is specified
4) KB - height of center of buoyancy about baseline
5) LCB - longitudinal center of buoyancy
6) LCF - longitudinal center of flotation
7) Longitudinal BM
8) Longitudinal KM
9) Moment to trim one inch
10) Prismatic coefficient
11) Tons per inch immersion
12) Transverse BM
13) Transverse KM
14) Displaced Volume
15) Wetted Surface
16) Waterplane area
17) Waterplane coefficient
18) Inertia coefficient

## II. Menu INPUT

ALL - Prompts the user for all input data.

TRIM - The trim of the vessel (defaults to design trim) only one trim per run.

REQUIRED - Prompts the user to enter the waterlines for the calculations (21 max.)

III. Menu DEFAULT

DESIGN - Used to alter the design condition of the ship.

TITLE - Used to give the ship name, ship number and the date.

DEFLN - Hull deflection in feet (+ SAG) Distribution is assumed parabolic between perpendiculars.

IV. Constraints on Input

The user must supply the module with a database generated by the Ships Hull Verification Module (VERIFY), in order to successfully calculate the hydrostatics. The user must "ENTER-DB" before "COMPUTE".

V. Menu ENTER-DB

The following information will be retrieved from the user's database (stored by VERIFY)

SDT - Ship's data table

STATN - Array of station numbers

X - Array of longitudinal station distances from FP.

NP - Number of points per station

CUFTON - Cubic Feet per ton of water

DRAFT - Design draft of the vessel

LBP - Length between perpendiculars

STASPC - Station spacing

LOA - Length overall

TRIM - Design trim of the vessel

XMID - Distance from FP to amidships

NSTAT - Number of stations

VI. Menu PRINT

VESSEL - Design condition of the vessel

WLINES - Given waterlines of the vessel

UNIT/DEF - Units and definitions

HSTATIC1 - Hydrostatics - Part 1

HSTATIC 2 - Hydrostatics - Part 2

AREAS - Sectional Areas at given waterlines

RESULTS - VESSEL + WLINES + UNIT/DEF + HSTATIC1 + HSTATIC2 + AREAS

## VII. Menu FILL-DB

This menu choice does not save any additional database information.

# FLOODABLE LENGTH
(FLOOD)

## I. ABSTRACT

This module computes data necessary for plotting curves of floodable length. The desired permeabilities must be input and the margin line heights can be entered through the default menu. Output includes; margin line heights, minimum permeability, entered permeabilities and floodable lengths for each input permeability.

## II. Menu INPUT

ALL - Prompts user for all input data

REQUIRED - Prompts user to enter permeabilities. A maximum of seven (7) are permitted.

## III. Menu DEFAULT

ALL - Prompts user to enter all default data

MARGIN - The user is to submit all the stations of which he is going to enter margin line heights. He then enters all the margin line heights corresponding to those stations.

LONGI - Submit the number of increments at which floodable length is calculated.

TITLE - Used for entering the ship name, ship number, and date

## IV. Constraints on Input

The user must supply the module with a database generated by the Ships Hull Verification module (VERIFY) in order to successfully calculate the vessels floodable length.

Other constraints are:

(1) NUMBER OF LONGITUDINAL INCREMENTS - Has a maximum value of forty (40)

(2) NUMBER OF PERMEABILITIES - Has a maximum value of seven (7)

## V. Database

The following information will be retrieved from the user's database upon the menu choice ENTER-DB.

<u>NP</u> - Number of points per station (interpolated offsets)

<u>SDT</u> - Ship's database supplied by VERIFY module

<u>STATN</u> - Array of station numbers

<u>X</u> - Array of longitudinal station distances (from FP)

<u>CUFTON</u> - Cubic feet of water per ton of displacement

<u>DISPL</u> - Design displacement of vessel

<u>DRAFT</u> - Design draft of vessel

<u>LBP</u> - Length between perpendiculars

<u>LCG</u> - Longitudinal center of gravity

<u>LOA</u> - Length overall

<u>TRIM</u> - Design trim of vessel

<u>XMID</u> - Distance between the FP and midship

<u>NSTAT</u> - Number of stations

The FILL-DB menu choice does not save any additional database information.

## VI. Menu PRINT

<u>INITIAL</u> - Output of input parameters, calculated on input margin line heights and the vessel's minimum permeability

<u>FLOOD</u> - Output of the calculated floodable lengths at every input permeability

<u>RESULTS</u> - The items of both INITIAL and FLOOD are printed

<u>ALL</u> - The items of both INITIAL and FLOOD are printed

# HYDROSTATICS
# &
# CURVES OF FORM

# (HYDRO)

## I. ABSTRACT

This module performs the required calculations to obtain the hydrostatic properties and curves of form of a vessel. Input consists of INITIAL offsets, principal dimensions and waterlines at which information is desired. Output consists of values at each waterline and graphs of the data over this range.

This module works directly off of an SHCP initial offset hull description and does not require that the user run the VERIFY module first. (See the VERIFY module to find out how the initial offsets are to be set up).

## II. REQUIRED INPUT

A. Initital offsets location
   1. In a file in standard SHCP format, or
   2. In a data base (e.g. generated by LINESGEN)

B. Station spacing
   If the offsets were obtained from LINESGEN, use 1.0 for station spacing.

C. Vessel name
   Up to 32 characters. If imbedded blanks are used, put the character string in primes.

D. Vessel number
   Up to eight (8) characters

E. LOA
   Length overall

F. LBP
   Length between perpendiculars

G. LENGTH (XL)
   The length to be used for calculating valves such as CB & CP (See METHODS)

H. BEAM

I. DEPTH

J. XMID
   The distance to amidships from the F.P. It is the location used for calculating values such as LCB. (See METHODS).

K. KG
   The vertical center of gravity above the baseline

## III. Menu INPUT

All of the following information is required for each run:

ALL
Prompts the user for all of the input data

UNITS
Defines which units the module will use for calculations.
a.) Length units (feet or meters)
b.) Weight units (tons, LBS or KG)
Only consistent units will be used by the module.

TRIM
The trim of the vessel (One trim per run)
(+ Down by the stern) It is defined as follows:
DRAFT AT STATION = DRAFT AT ⊗ -TRIM*(XMID-X(ISTA))/XLBP

WLINES
Up to 16 waterlines may be entered in any order. Results will be sorted in order of increasing drafts. For vessels with trim this value is the draft at amidship (XMID).

IV. CONSTRAINTS ON INPUT

The initial offsets must be in the data base or in a MTS file in standard SHCP format.

V. Menu ENTER-DB

Since the module doesn't require interpolated offsets, this command does nothing.

VI. Menu DEFAULT

The are no default values.

VII. Menu COMPUTE

There are no options for this command

VIII. Menu FILL-DB

None of the results are stored.

IX. Menu PRINT

Prints any or all of the following at the terminal:

ALL
Prints all of the options below

DESIGN
Title page and principal dimensions. (INPUT)

SECTION
Prints sectional information and Bonjean's curves. Note: one page is printed per station.

HYDRO1
Prints the following for each waterline:
a.) Volume
b.) Displacement (S.W. & F.W.)
c.) Wetted surface
d.) Surface area & KG
e.) Longitudinal center of buoyancy (LCB)
f.) KB

HYDRO2

Prints the follwoing for each waterline:

a.) Waterplane area

b.) Longitudinal Inertia Coefficient

c.) Transverse Inertia Coefficient

d.) Longitudinal center of flotation (LCF)

e.) BML

f.) BMT

HYDRO3

Prints the following for each waterline:

a.) KML

b.) KMT

c.) Moment to trim one inch (MTI) (or cm)

d.) Tons per inch immersion (TPI) (or cm)

e.) Change in displacement per trim of one foot (meter) aft.

HYDRO4

Prints the following for each waterline:

a.) Block coefficient ($C_b$)

b.) Prismatic coefficient ($C_p$)

c.) Midship coefficient ($C_m$)

d.) Waterplane coefficient ($C_{wp}$)

e.) Vertical prismatic coefficient ($C_{vp}$)

X. Menu SEND

This command sends the printed results to an MTS file. It has the same menu as PRINT but requires an output file or device name

XI. Menu DRAW (Only for graphics terminals)

Draws any or all of the following at the terminal:

ALL

Draws all of the options below

BODY

Body plan for the vessel

AREAS

Graphs of sectional areas & waterplanes at each input waterline

MOMENTS

Graphs of transverse & vertical moments at each input waterline.

GIRTHS

Graphs of wetted surface, girth area & KG of girth area at each input waterline.

NOTE: The following graphs are available <u>only</u> if the number of input waterlines is greater than 3.

<u>VOLUMES</u>
Graphs over the range of input waterlines for:
a.) Volume
b.) Displacement
c.) Longitudinal center of bouyancy
d.) Vertical center of bouyancy

<u>LCF</u>
Graphs over the range of input waterlines for:
a.) Waterplane areas
b.) Longitudinal center of flotation
c.) Transverse coefficient of inertia
d.) Longitudinal coefficient of inertia

<u>CENTERS</u>
a.) BML
b.) BMT
c.) KML
d.) KMT

<u>FACTORS</u>
Graphs over the range of input waterlines for:
a.) Moment to trim one onch (or cm)
b.) Tons per inch immersion (or cm)
c.) Change in displacement per trim of one foot (meter) aft
d.) Waterplane coefficient

<u>COEFFS</u>
Graphs over the range of input waterlines for:
a.) Block coefficient
b.) Prismatic coefficient
c.) Midship coefficent
d.) Vertical prismatic coefficient

<u>SURFACES</u>
Graphs over the range of input waterlines for:
a.) Surface area
b.) KG of surface area
c.) Wetted surface

XII. Menu SAVE

This command saves the graphs in an MTS file. These can be queued later for plotting. It has the same menu as DRAW but requires a SIZE in inches (width) of the plot and a file for storage.
e.g. ?SAVE ALL 7.5 PLOTS

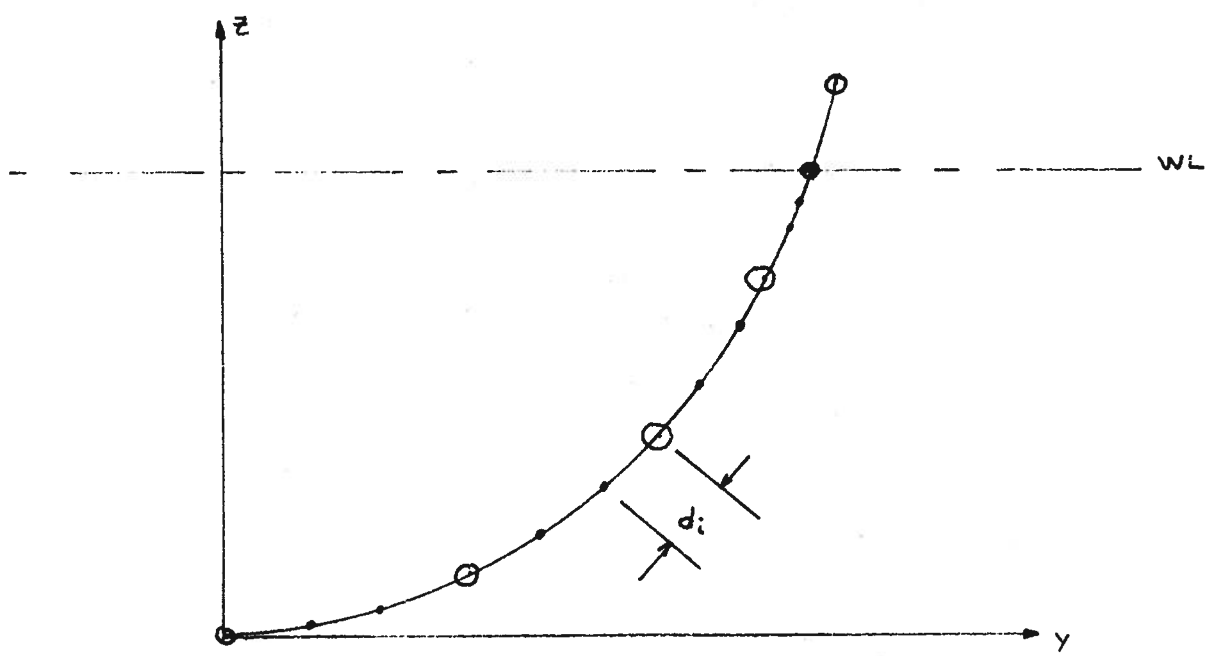
# DISCUSSION OF METHODS EMPLOYED IN HYDROSTATICS

## GENERAL

The program employs a modified cubic spline curve-fitting routine to interpolate points and generate all of the curves. Areas, volumes, moments and moments and products of inertia are obtained by exact integration of these curves by using line integrals.

## EXAMPLE

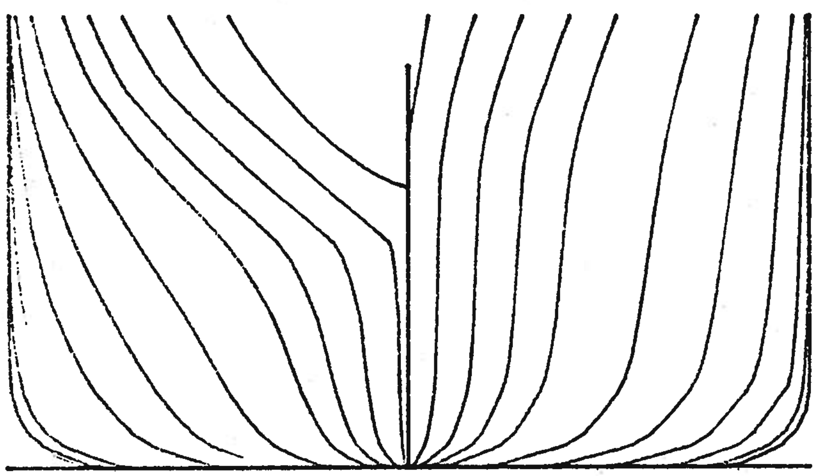
SUBMERGED AREAS & GIRTHS




$$\text{AREA} = -\int_L y\,dz$$

$$\text{GIRTH*} = \sum_{i=1}^{n} d_i$$

⊙ – Data point - Supplied or taken off body plan
• – Data point introduced by curve-fitting



DEC 14, 1978

PRINCIPAL DIMENSIONS

LOA - 525.00

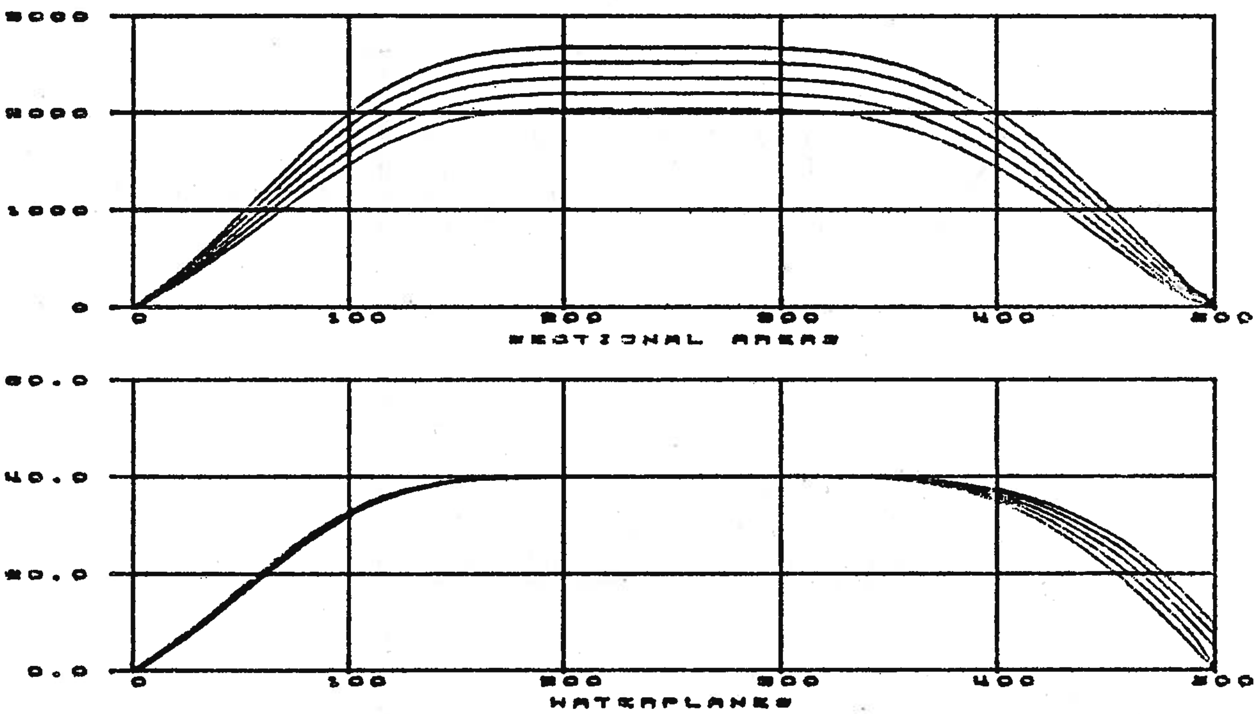
LBP - 500.00

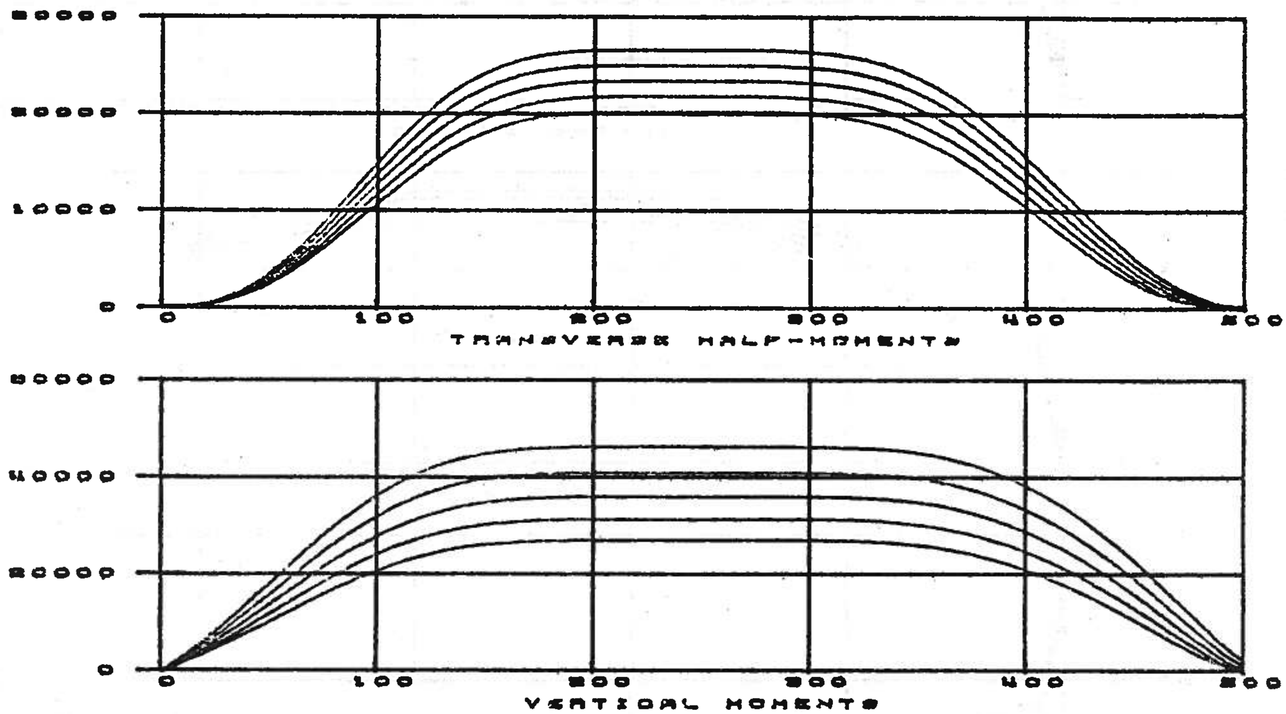
BEAM - 80.00

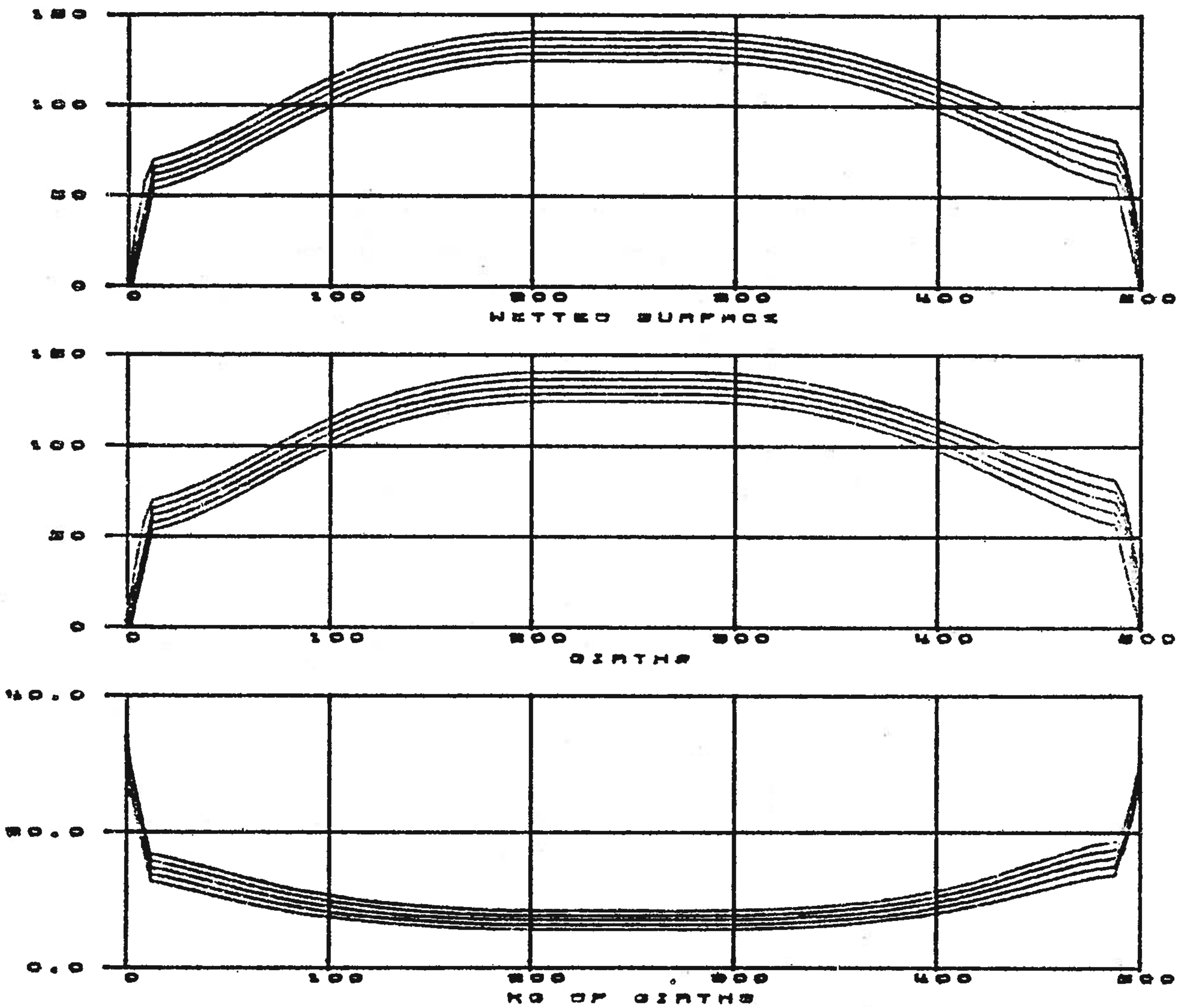
DEPTH - 45.00

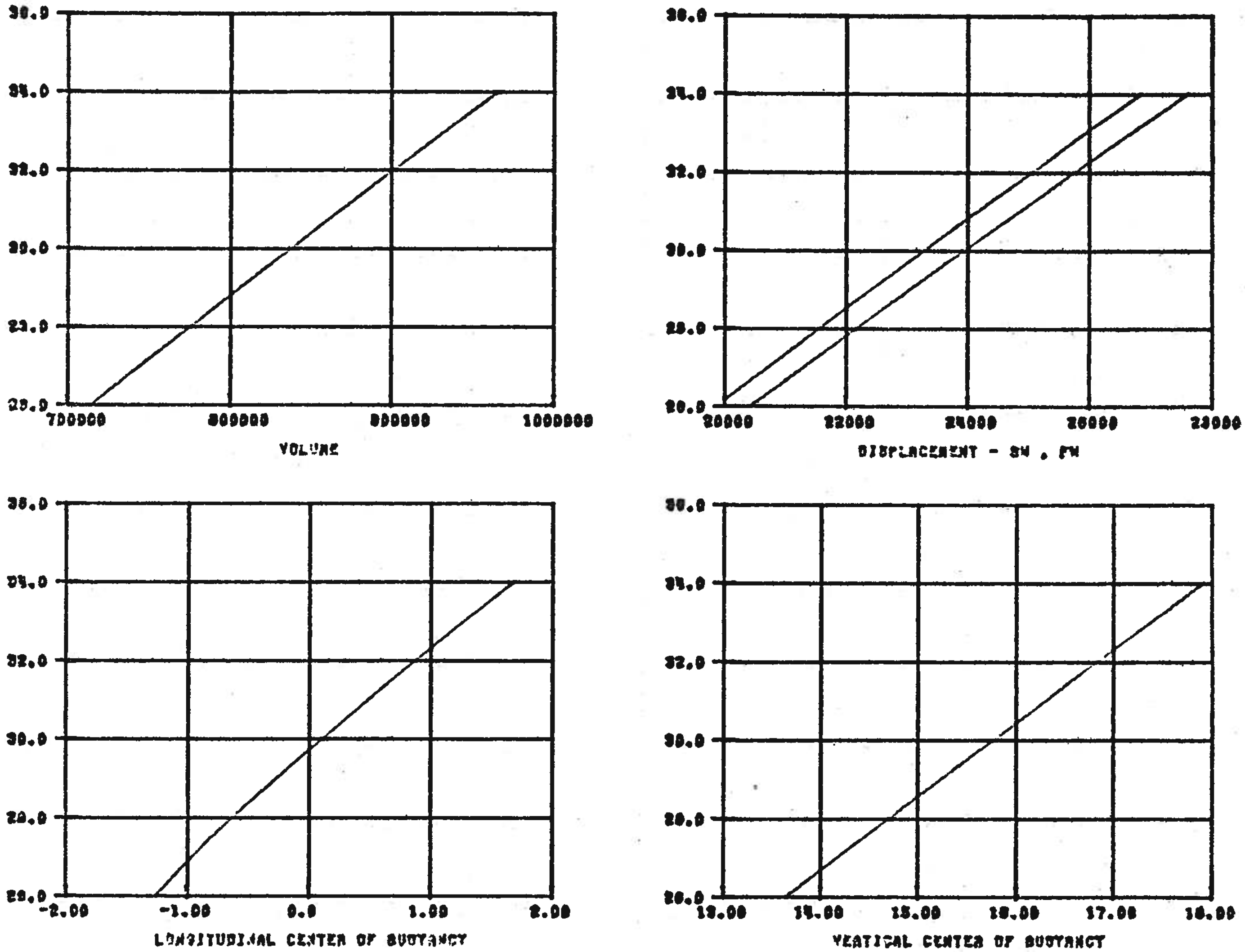
*** JBM EXPRESS

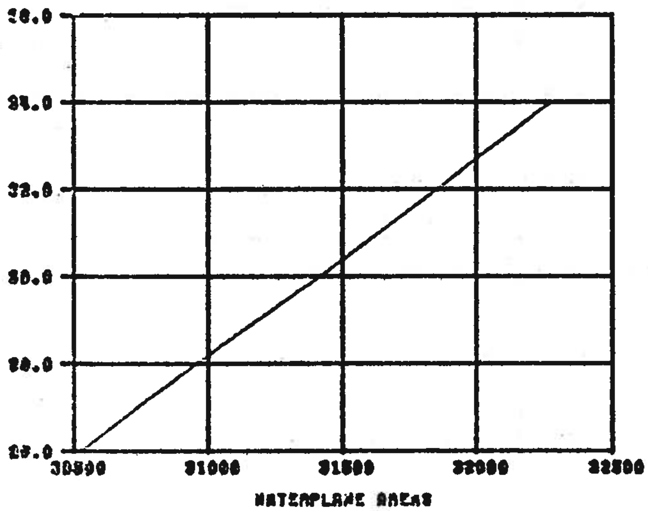
DESIGN NO: 100

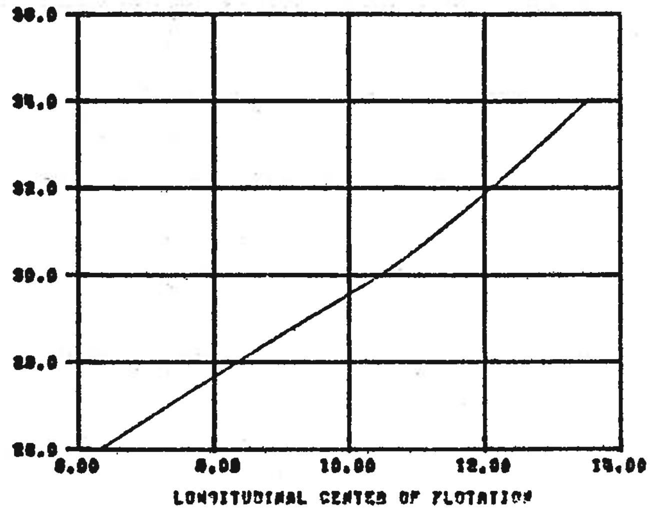

3000
2000
1000
0
0
100
200
300
400
500
SECTIONAL AREAS
60.0
40.0
20.0
0.0
0
100
200
300
400
500
WATERPLANES

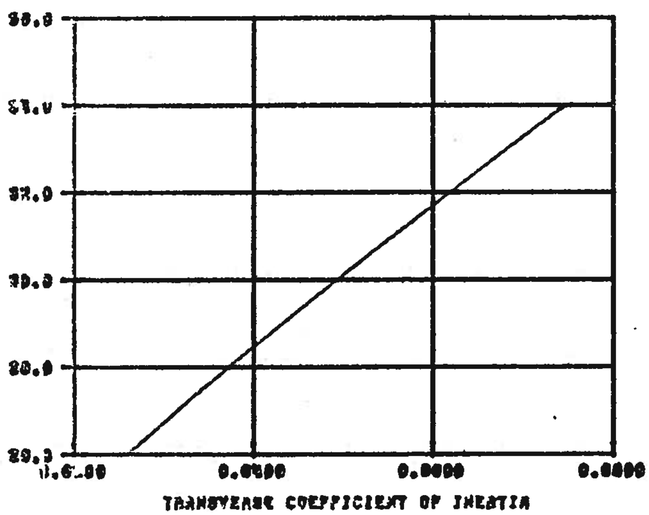

30000
20000
10000
0
0
100
200
300
400
500
TRANSVERSE HALF-MOMENTS
60000
40000
20000
0
0
100
200
300
400
500
VERTICAL MOMENTS

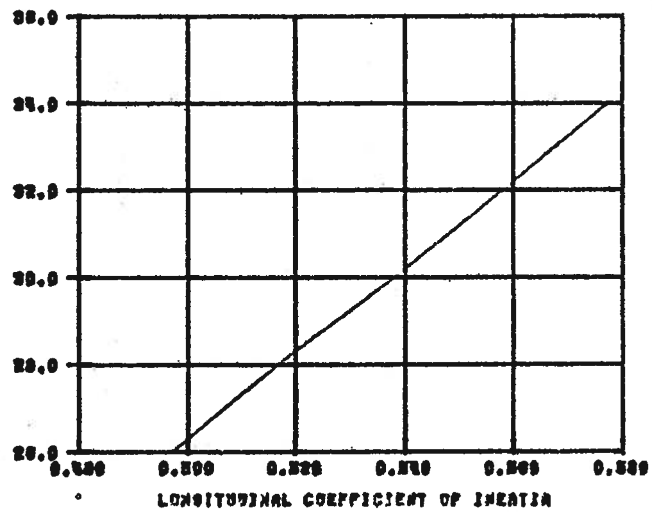

150
100
50
0
0
100
200
300
400
500
WETTED SURFACE
150
100
50
0
0
100
200
300
400
500
GIRTHS
40.0
20.0
0.0
0
100
200
300
400
500
KG OF GIRTHS

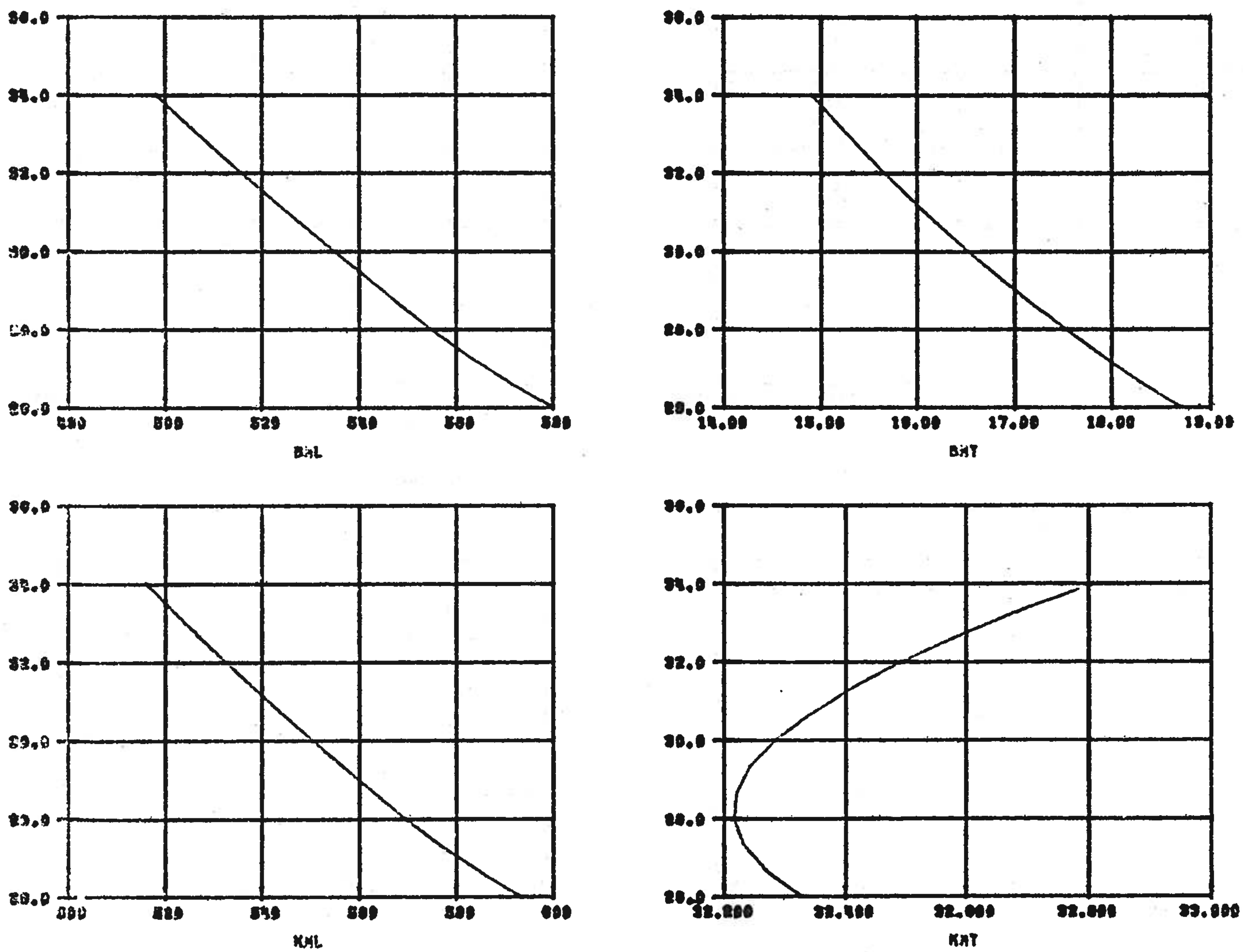

700000
800000
900000
1000000
VOLUME
20000
22000
24000
26000
28000
DISPLACEMENT - SW , FW
-2.00
-1.00
0.0
1.00
2.00
LONGITUDINAL CENTER OF BUOYANCY
14.00
15.00
16.00
17.00
18.00
VERTICAL CENTER OF BUOYANCY

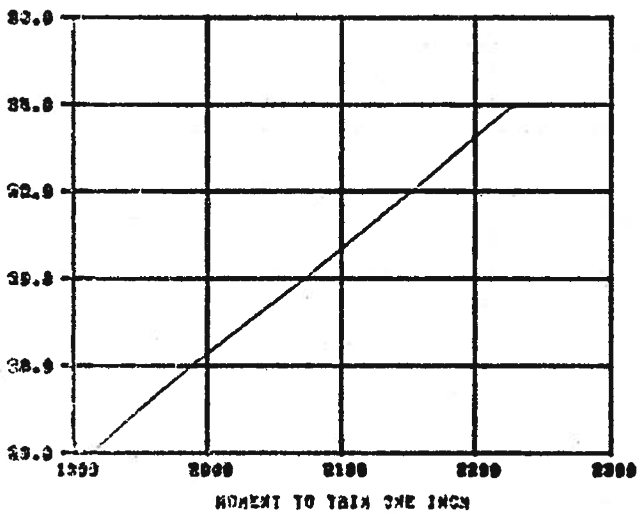

30500
31000
31500
32000
32500
WATERPLANE AREAS

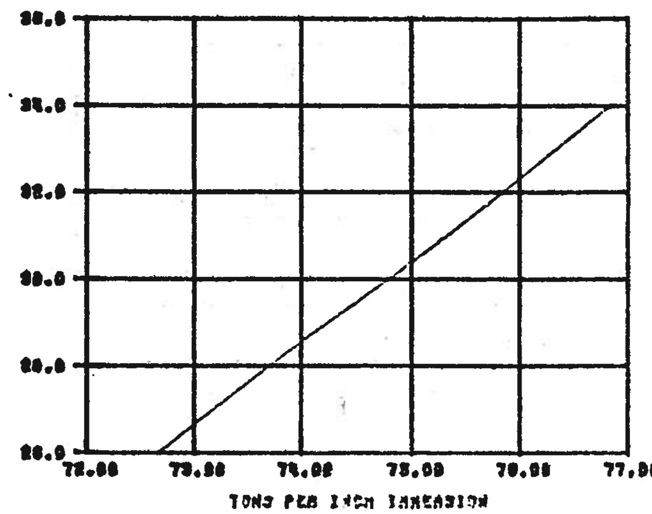

10.00
12.00
LONGITUDINAL CENTER OF FLOTATION

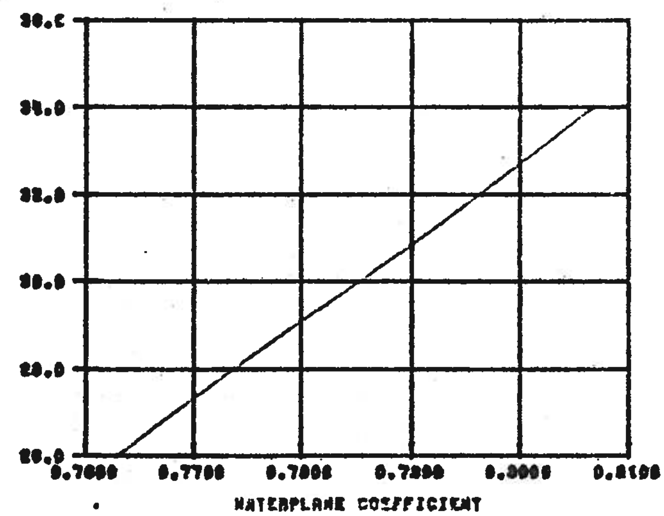

TRANSVERSE COEFFICIENT OF INERTIA

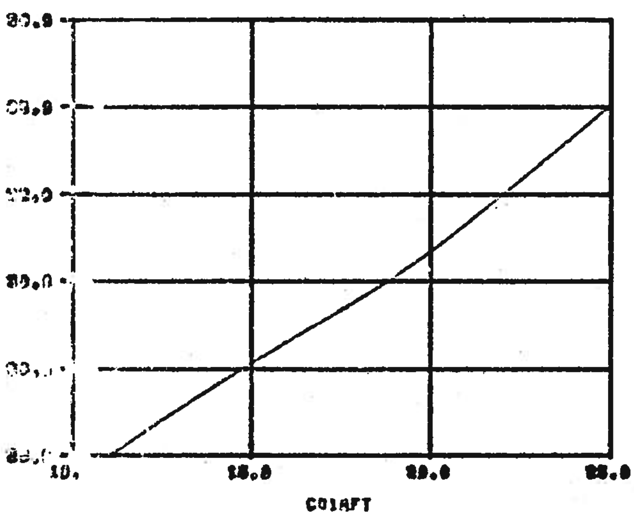

LONGITUDINAL COEFFICIENT OF INERTIA

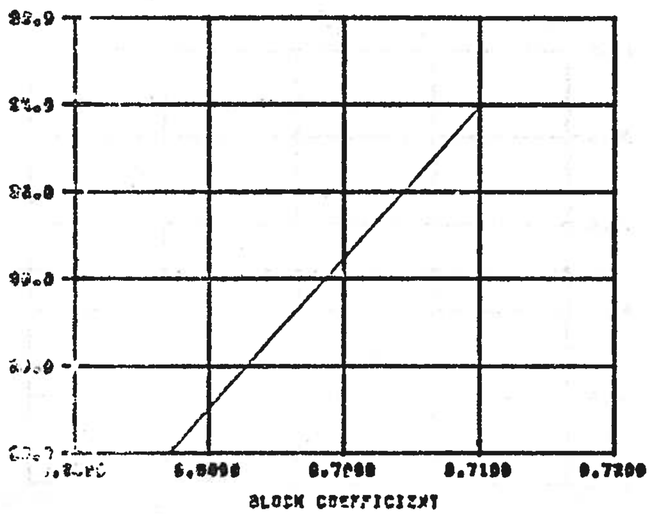

BHL
BHT
KHL
KHT

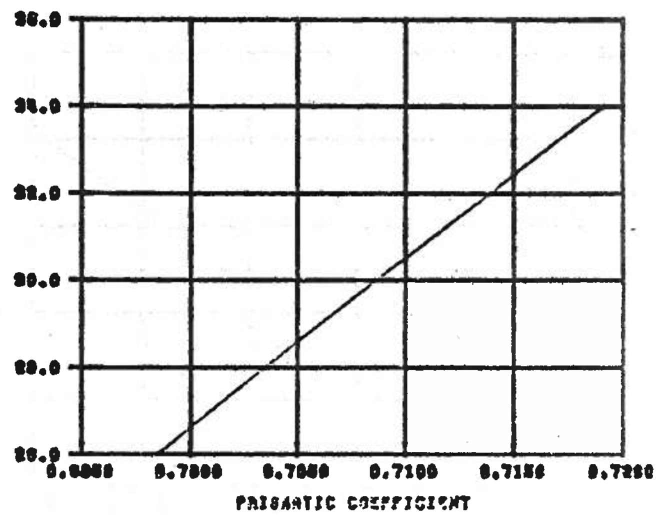

2000
2100
2200
MOMENT TO TRIM ONE INCH

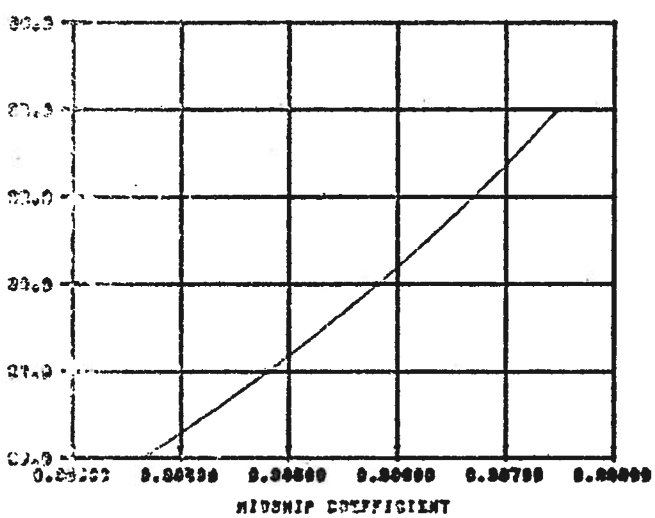

77.00
TONS PER INCH IMMERSION

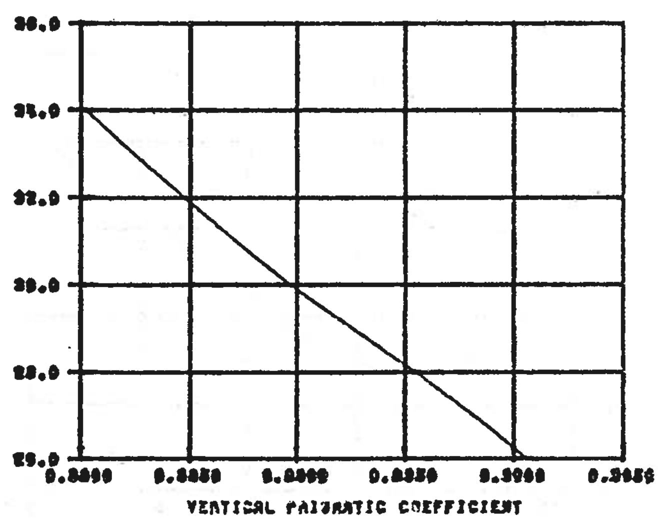

WATERPLANE COEFFICIENT

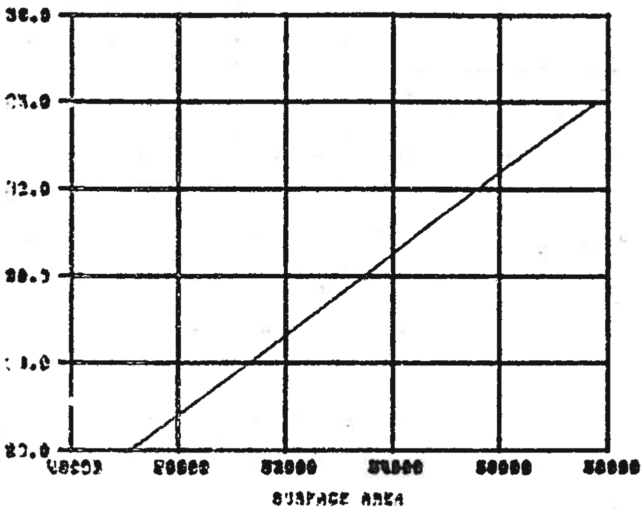


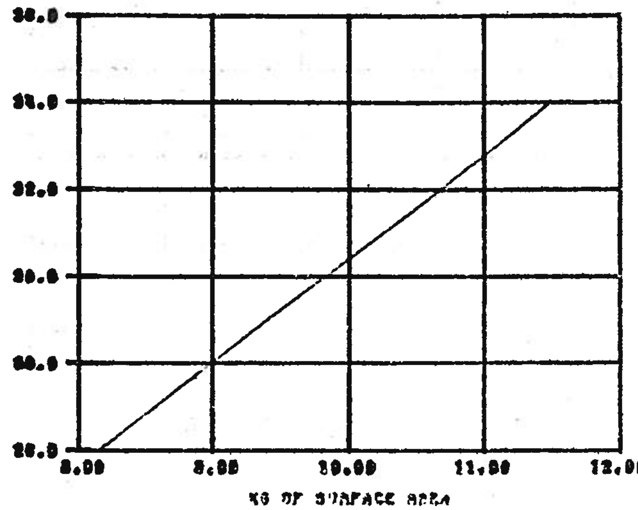

0.7100
0.7200
BLOCK COEFFICIENT

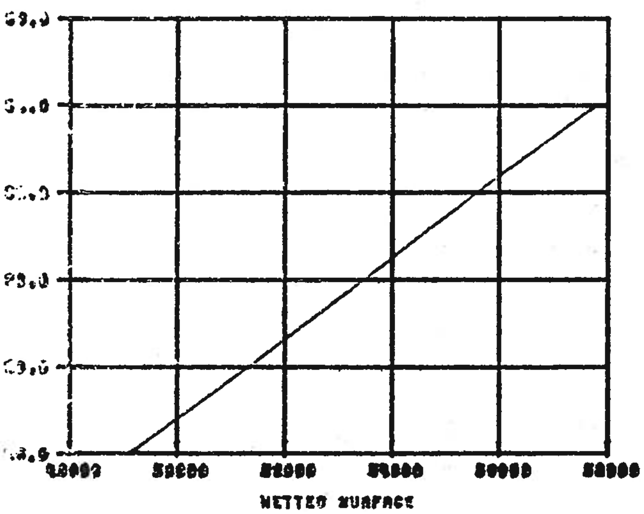

PRISMATIC COEFFICIENT


MIDSHIP COEFFICIENT

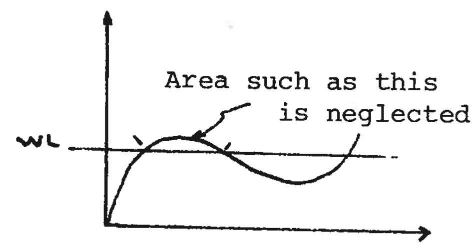

VERTICAL PRISMATIC COEFFICIENT


SURFACE AREA


% OF SURFACE AREA


WETTED SURFACE

## EXPLANATION OF TERMS & SYMBOLS IN HYDROSTATICS

| SYMBOL OR TERM | DEFINITION | FORMULA |
|---|---|---|
| DRAFT | Mean waterline draft at (XMID) | - |
| VOLUME | Volume of Submerged hull at specified draft | $VOL. = \int_L S.A.\, dx$<br>S.A. = Sectional Area Curve |
| DISPLACEMENT | Weights of displaced volume of water in long tons (2240#) (or in desired units) | Salt water = $Vol./\gamma_{SW}$<br>Fresh Water = $Vol./\gamma_{FW}$ |
| SURFACE AREA | Surface area of hull up to the given draft. | $\text{Surface Area} = \int_L \text{Girth}\, dx$ |
| WETTED SURFACE | Wetted surface of hull up to the given draft |  |
| LCB | Longitudinal Center of bouyancy in ft. from (t+aft) | $LCB = \dfrac{\int_L S.A.\ {*}X dx}{Vol.} - XMID$ |
| KB | Vertical Center of bouyancy in ft. from the baseline (t+UP) | $KB = \dfrac{\int_L S.A.\ {*}Z dx}{Vol.}$ |
| WATERPLANE AREA | Area of waterplane at the given draft | $W.P.\ AREA = 2\int_L HB\, dx$<br>HB = Half breadths at the given draft |
| LCF | Longitudinal center of flotation if ft. from (+ AFT) | $LCF = \dfrac{2\int_L HB{*}x dx}{W.P.\ AREA} - XMID$ |

| SYMBOL OR TERM | DEFINITION | FORMULA |
| --- | --- | --- |
| LONGITUDINAL INERTIA | Longitudinal moment of inertia of the waterplane about LCF | $I_L = \int_L HB\, x^2 dx - Awp*\bar{x}^2$ |
| LONGITUDINAL INERTIA COEFFICIENT | | where $\bar{x}$ = XMID+LCF<br>$= I_L / \frac{(BEAM * XL^3)}{12.}$ |
| TRANSVERSE INERTIA | Transverse moment of inertia of the waterplane about C | $I_T = \int_L HB\, y^2\, dx$ |
| TRANSVERSE INERTIA COEFFICIENT | | $= I_T / \frac{(XL*BEAM^3)}{12.}$ |
| BML | Longitudinal Metacentric height (ft)(m) | $I_L$/Vol. |
| BMT | Transverse Metacentric height (ft)(m) | $I_T$/Vol. |
| KML | Longitudinal Metacentric Radius (ft)(m) | KML=BML+KB |
| KMT | Transverse Metacentric Radius (ft)(m) | KMT=BMT+KB |
| MT1 | Moment to trim vessel 1 inch (foot-tons) (1 cm) | $MT1 = \frac{GM_1*Vol}{12*XL*\ SW}$<br>(100.)<br>XL=Design length<br>GML=KML-KG |
| TPI | Tons per inch immersion (cm) | $TPI = \frac{Awp}{12*\ SW}$<br>(100.) |
| CD1AFT | Change in displacement per foot of trim by the stern (m.) | $CD1AFT + \frac{Awp}{XL} \frac{LCF}{SW}$ |
| CB | Block Coefficient | CB=Vol/XL Bwp · Draft |
| CM | Midship Coefficient | $CM = S.A._{MAX}/(B_{wp} \cdot Draft)$ |
| CP | Prismatic Coefficient | $CP = Vol/(XL\ S.A._{MAX})$ |
| CWP | Waterplane coefficient | $CWP = A_{WP}/(XL\ B_{wp})$ |
| CVP | Vertical prismatic coefficient | CVP-CB/CWP |

# INFLATION
## (INFLATE)

## I. ABSTRACT

This module calculates present worth at a given yield or actual yield on a capital investment, while accounting for inflation. Using economic life equal to tax life, it produces a year-by-year table of depreciation allowance, tax base, tax, cash flow, and present worth.

## II. Menu INPUT

<u>ALL</u> - User will be prompted for all input variables

<u>LIFE</u> - Economic and tax life in years

<u>INVEST</u> - Initial investment

<u>DVALUE</u> - Disposal value

<u>AREVENUE</u> - Annual revenue

<u>AOCOST</u> - Annual operation cost given in decimal form

<u>TAXRATE</u> - Tax rate given in decimal form

<u>AIRATE</u> - Annual inflation rate

<u>YIELD</u> - Given in decimal form

## III. Menu DEFAULT

There is no DEFAULT menu.

## IV. Menu PRINT

<u>INPUT</u> - Display of input variables

<u>PARTIAL</u> - This enables display of select years of the year-by-year table of results

<u>YIELD</u> - The calculated or input yield is printed

<u>SUMPW</u> - The sum of present worth is printed

<u>RESULTS</u> - All the above are printed, including the full year-by-year table

## V. Constraints on Input

(1) ECONOMIC AND TAX LIFE - Must be an integer

(2) YIELD - Must be greater than zero

VI. COMPUTE options

Menu YIELD

ACTUAL - The estimated yield will be corrected as necessary to converge on the actual yield

PRESENT - The user desired present worth at his entered yield

Menu DEP.TYPE

LINE - Straight line depreciation method

SUMDIGIT - Sum of digit depreciation method

VII. Database

This module does not operate with a database.

VIII. Method

The module was adapted to DEX from a program written by R. J. Cunuccio.

# INTACT STABILITY
## (STABLE)

## I. ABSTRACT

This module performs the required calculations to produce the Curves of Intact Statical Stability and the Cross Curves of Stability. Inputs are the draft (or displacement), trim (or LCG), and poleheight of the vessel, and the desired heel angles for which the calculations are to be performed.

## II. Menu INPUT

<u>ALL</u> - Prompts user for all input data

<u>TRIM</u> - Vessel's trim for which the calculations are to be performed

<u>WL(S)</u> - Either the desired waterline draft for the Intact Statical Stability Curves or up to seven (7) waterline drafts for the Cross Curves of Stability

<u>HEELS</u> - The heel angles for which the Curves of Stability will be evaluated. A maximum of ten (10) heel angles are permitted.

<u>REQUIRED</u> - Prompts the user to enter the desired waterline drafts and the heel angles

## III. Menu DEFAULT

<u>LCG-DISP</u> - Prompts the user to enter a desired combination of displacement and LCG for which the Curves of Intact Statical Stability will be calculated.

<u>TRIMDISP</u> - A trim and displacement combination for which the Curves of Statical Stability will be calculated

The user will also be prompted to enter the heel angles for which the calculations are desired. A maximum of ten (10) heel angles are permitted.

## IV. CONSTRAINTS ON INPUT

The user must supply the module with a database generated by the Ships Hull Verification Module (VERIFY) in order to successfully calculate the vessel's stability.

Other constraints are:

(1) heel angle must lie between 0° and 90° inclusive, and

(2) the highest waterline draft must be lower than the sheer of the vessel.

V. DATABASE

The following information will be retrieved from the user's database upon the menu choice ENTER-DB.

SDT: Ship's data base supplied by VERIFY module

STATN: Array of station numbers

X: Array of longitudinal station distances (from FP)

NP: Number of points per station (interpolated offsets)

NSTAT: Number of stations

STASPC: Station spacing

DRAFT: Design draft of vessel

TRIM: Design trim of vessel

VCG: Vertical center of gravity

CUFTON: Cubic feet of water per ton of displacement

The FILL-DB menu choice does not save any additional database information.

VI. Menu COMPUTE

Menu choices are STATICAL and CROSS

VII. Menu PRINT

VESSEL - Design conditions

INPUT - Input variables

RESULTS - Results of calculations

ALL - All the above information

Other constraints are:

(1) heel angle must lie between 0° and 90° inclusive, and

(2) the highest waterline draft must be lower than the sheer of the vessel.

V. DATABASE

The following information will be retrieved from the user's database upon the menu choice ENTER-DB.

SDT: Ship's data base supplied by VERIFY module

STATN: Array of station numbers

X: Array of longitudinal station distances (from FP)

NP: Number of points per station (interpolated offsets)

NSTAT: Number of stations

STASPC: Station spacing

DRAFT: Design draft of vessel

TRIM: Design trim of vessel

VCG: Vertical center of gravity

CUFTON: Cubic feet of water per ton of displacement

The FILL-DB menu choice does not save any additional database information.

VI. Menu COMPUTE

Menu choices are STATICAL and CROSS

VII. Menu PRINT

VESSEL - Design conditions

INPUT - Input variables

RESULTS - Results of calculations

ALL - All the above information

# IOR Rating Program
# (IOR)

## ABSTRACT

This program calculates the rating for a yacht under the International Offshore Rule MARK III and MARK IIIA amended to Nov., 1978. Input consists of all measured parameters (listed on the certificate) and output consists of the two page rating certificate and/or intermediate calculated values. Note that this program calculates results only for sloops.

## USER GUIDELINES

A file should be established by the user prior to the execution of the program. The data file should contain all the measurements in the following format. See the IOR MARK III rule book for definitions and details.

| LINE | INPUT | FORMAT | COMMENT |
|---|---|---|---|
| 1 | Yacht name | 6A4 | |
| 2 | Sail number | 6A4 | |
| 3 | Owner's name | 7A4 | |
| 4 | Owner's street | 7A4 | |
| 5 | Owner's City, State & Zip | 7A4 | |
| 6 | Designer | 3A4 | |
| 7 | Builder | 3A4 | |
| 8 | Hull number | 3A4 | |
| 9 | Year built | I6 | |
| 10 | Hull month & year | 2I6 | |
| 11 | Rig month & year | 2I6 | |
| 12 | Rig type | 7A4 | |
| 13 | Keel type | 7A4 | |
| 14 | Prop. type | 7A4 | |
| 15 | Prop. Installation type | 7A4 | |
| 16 | IUNITS | I6 | 1-feet, 2-meters |
| 17 | IBOARD | I6 | 0-Keel, 1-Board |
| 18 | LOA | F15,14 | |
| 19 | FGO | " | |
| 20 | AGO | " | |
| 21 | LBG | " | |

| LINE | INPUT | FORMAT | COMMENT |
| --- | --- | --- | --- |
| 22 | GSDA | F15.14 | |
| 23 | GSDF | " | |
| 24 | FD | " | |
| 25 | CMD | " | |
| 26 | MD | " | |
| 27 | OMD | " | |
| 28 | GDFI | " | |
| 29 | EW | " | |
| 30 | ESL | " | |
| 31 | CDA | " | Board No. 1 |
| 32 | CDB | " | Board No. 2 |
| 33 | BMAX | " | |
| 34 | B | " | |
| 35 | BWL | " | |
| 36 | BF | " | |
| 37 | BFI | " | |
| 38 | BAI | " | |
| 39 | BA | " | |
| 40 | GD | " | |
| 41 | Y | " | |
| 42 | DMS | " | DM-Standard |
| 43 | GLAI | " | |
| 44 | EWD | " | |
| 45 | ESC | " | |
| 46 | WCBA | " | Board No. 1 |
| 47 | FFS | " | FF-Standard |
| 48 | FFIS | " | FFI- " |
| 49 | FFDS | " | FFD- " |
| 50 | FMDS | " | FMD- " |
| 51 | FAIS | " | FAIS- " |
| 52 | FAS | " | FA- " |

| LINE | INPUT | FORMAT | COMMENT |
| --- | --- | --- | --- |
| 53 | VHAI | F15.14 | |
| 54 | VHA | " | |
| 55 | BHAI | " | |
| 56 | BHA | " | |
| 57 | ESDS | " | ESD-Standard |
| 58 | PDS | " | PD- Standard |
| 59 | ST1 | " | |
| 60 | WCBB | " | Board No. 2 |
| 61 | AW | " | |
| 62 | BW | " | |
| 63 | CW | " | |
| 64 | DW | " | |
| 65 | AWD | " | |
| 66 | BWD | " | |
| 67 | CWD | " | |
| 68 | DWD | " | |
| 69 | MAW1 | " | |
| 70 | MACG1 | " | |
| 71 | MACL | " | |
| 72 | PRD | " | |
| 73 | ST2 | " | |
| 74 | CBDA | " | Board No. 1 |
| 75 | CBLDA | " | Board No. 1 |
| 76 | APD | " | |
| 77 | BPD | " | |
| 78 | CPD | " | |
| 79 | DPD | " | |
| 80 | PL | " | |
| 81 | SBMAX | " | |
| 82 | SPD | " | |
| 83 | SDM | " | |
| 84 | MAW2 | " | |

| LINE | INPUT | FORMAT | COMMENTS |
|---|---|---|---|
| 85 | MACG2 | F15.14 | |
| 86 | MACO | " | |
| 87 | PBW | " | |
| 88 | ST3 | " | |
| 89 | CBDB | " | Board No. 2 |
| 90 | CBLDB | " | Board No. 2 |
| 91 | J | " | |
| 92 | LPG | " | |
| 93 | LPIS | " | |
| 94 | FSP | " | |
| 95 | FJS | " | FJ-Standard |
| 96 | FBIS | " | |
| 97 | IG | " | |
| 98 | SPL | " | |
| 99 | SPS | " | |
| 100 | SL | " | |
| 101 | SMW | " | |
| 102 | HBS | " | |
| 103 | SFJ | " | |
| 104 | ISP | " | |
| 105 | P | " | |
| 106 | E | " | |
| 107 | BAS | " | |
| 108 | BAL | " | |
| 109 | BD | " | |
| 110 | MW | " | |
| 111 | GO | " | |
| 112 | HB | " | |
| 113 | BL1 | " | |
| 114 | BL2 | " | |
| 115 | BL3 | " | |
| 116 | BL4 | " | |
| 117 | BL5 | " | Largest intermediate batten |

| LINE | INPUT | FORMAT | COMMENT |
|---|---|---|---|
| 118 | BLP | F15.14 | |
| 119 | PF | " | Corrections must be added in. |
| 120 | SMF | " | |
| 121 | LRP | " | |
| 122 | FFM | " | FF-Measured |
| 123 | FAM | " | FA-Measured |
| 124 | FBIM | " | FBI- " |
| 125 | FJM | " | FJ- " |

NOTES

1. The order of this input follows the ordering of the data on the certificate.
2. See Part X of the IOR rule for complete description of Standard vs. Measured values. If the measured value differs from the standard value, the SINK and TRIM of the yacht is calculated and corrections to other values are made.
3. The input file name will be requested by the program

## RUNNING THE PROGRAM

After the input file is read correctly, the user is prompted with the message "OPTIONS?". At this point the user specifies which print option is desired, along with any modification to input values. A proper response to the prompt consists of a list of assignments of the form variable=value, e.g.

```
OPTIONS? B=12.5  J=10.2  PRINT =1
```

For this example, the rating of the yacht will be recalculated using the new values for B and J. Note that a change in B would normally cause the girth stations to change, but this program doesn't take this into account. The user must adjust the values to compensate for this.

The values which can be assigned new values in this manner consist of the values located on lines 18 through 125, including the variable PRINT. All of the measurements must be assigned real values, except PRINT, which must be an integer between Ø and 5 inclusive. The variable PRINT is used to control the printed form of the results.

| | |
|---|---|
| PRINT=0 | No output. This is the default value for PRINT |
| PRINT=1 | Prints the two page rating certificate and intermediate results. |
| PRINT=2 | Prints the two page rating certificate. |
| PRINT=3 | Prints just the final factors and rating. |
| PRINT=4 | Prints final rating and intermediate results. |

The program will always return the user to the prompting message "OPTIONS?" when a task is completed.

## EXAMPLE

The following example shows the rating certificate for the yacht "DOWNTOWN" before and after remeasurement. Note that the measurements are in meters and that the intent of the remeasurement was to get the yacht to rate in at one ton (27.5).

The yacht was remeasured in fresh water (floatation test) and the "P" measurement was reduced to 12.884 meters. The following measurements were taken and used to calculate the new rating.

| | | | |
|---|---|---|---|
| P | = 12.884 | APD | = 53.0 |
| PL | = 2000. | BPD | = 106.5 |
| AW | = 20.5 | CPD | = 51.5 |
| BW | = 43.4 | DPD | = 105.5 |
| CW | = 20.5 | FFM | = 1.030 |
| DW | = 43.3 | FAM | = 1.167 |
| AWD | = 5.53 | FBIM | = 1.057 |
| BWD | = 5.53 | FJM | = 1.030 |
| CWD | = 5.53 | BWL | = 2.678 |
| DWD | = 5.53 | | |

Note that the values for PL and AW - DW remained the same.

#R IOR.O 2=IORTEST T=5
#EXECUTION BEGINS



OPTIONS? PRINT=1

INTERNATIONAL OFFSHORE RULE
MARK III AMEND. TO NOV. 1978
DESIGNER FARR
BUILDER COOKSON BUILT 1978
HULL# HULL DT 8/78
RIG DT 5/79
RIG: SLOOP
KEEL: FIXED KEEL W/O MOV. APPEND.
PROP: FOLDING
INST: OUT OF APERTURE - EXP. SHAFT

RATING CERTIFICATE NO.
MEASUREMENTS IN METERS AND KGS.

| |
|---|
| YACHT DOWNTOWN |
| SAIL# KZ-3776 |
| OWNER: |
| William C. Martin |
| 115 Depot St. |
| Ann Arbor, MI 48104 |

----HULL----

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| LOA | 11.289 | BMAX | 3.839 | FF | 1.051 | AW | 20.50 | APD | 42.000 |
| FGO | 0.559 | B | 3.461 | FFI | 1.050 | BW | 43.30 | BPD | 89.000 |
| AGO | 0.487 | BWL | 2.654 | FFD | 1.060 | CW | 20.50 | CPD | 42.000 |
| LBG | 10.243 | BF | 0.437 | FMD | 1.090 | DW | 43.30 | DPD | 90.000 |
| GSDA | 0.571 | BFI | 0.780 | FAI | 1.148 | AWD | 4.900 | PL | 2000.000 |
| GSDF | 0.501 | BAI | 2.788 | FA | 1.159 | BWD | 4.900 | SBMAX | 7.180 |
| FD | 1.450 | BA | 2.498 | VHAI | 0.861 | CWD | 4.900 | SPD | 7.580 |
| CMD | 1.547 | GD | 0.645 | VHA | 0.748 | DWD | 4.900 | SDM | 5.680 |
| MD | 1.409 | Y | 0.487 | BHAI | 0.0 | MAW1 | 0.0 | MAW2 | 0.0 |
| OMD | 1.089 | DM | 2.034 | BHA | 0.0 | MACG1 | 0.0 | MACG2 | 0.0 |
| GDFI | 0.0 | GLAI | 0.0 | ESD | -0.0 | MACL | 0.0 | MACO | 0.0 |
| EW | 210.0 | EWD | 2.256 | PD | 0.651 | PRD | 0.510 | PBW | 0.130 |
| ESL | 0.800 | ESC | 0.285 | ST1 | 0.029 | ST2 | 0.110 | ST3 | 0.112 |
| CD | 0.0 | WCBA | 0.0 | WCBB | 0.0 | CBDA | 0.0 | CBDB | 0.0 |
| | | | | | | CBLDA | 0.0 | CBLDB | 0.0 |

| ------FORETRIANGLE------ | | | | ---------MAINSAIL---------- | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| I | 12.010 | SPL | 3.530 | P | 13.110 | HB | 0.211 | PC | 13.1100 |
| J | 3.530 | SPS | 1.741 | E | 5.450 | BL1 | 0.850 | IC | 12.0102 |
| LPG | 5.250 | SL | 11.890 | BAS | 1.566 | BL2 | 0.850 | JC | 3.5300 |
| LPIS | 5.250 | SMW | 6.350 | BAL | 0.150 | BL3 | 0.959 | EC | 5.4500 |
| FSP | 0.040 | HBS | 0.0 | BD | 0.155 | BL4 | 0.959 | MXSL | 11.8923 |
| FJ | 1.086 | TCI | 0.124 | | | BL5 | 0.0 | MXSMW | 6.3540 |
| FBI | 1.072 | SFJ | 0.634 | MW | 0.147 | BLP | 2.800 | MXLP | 5.2950 |
| IG | 11.939 | ISP | 12.023 | GO | 0.172 | | | | |

| | | | | | |
|---|---|---|---|---|---|
| L | 9.5967 | DC | 0.0076 | SC | 7.6004 |
| B | 3.4610 | FC | -0.0272 | MR | 9.0281 |
| D | 1.0622 | EPF | 0.9671 | R | 8.4701 |
| CGF | 0.9701 | MAF | 1.0000 | RA | 0.0 |
| SMF | 1.0000 | DLF | 1.0207 | CBF | 1.0000 |
| LRP | 1.0000 | | | | |

| RATINGS | MK III | MK IIIA |
|---|---|---|
| FEET: | 27.8 | 0.0 |
| METERS: | 8.47 | 0.0 |

JUN 20, 1979

```
                    SAIL# KZ-3776     MK III =    8.4701  PAGE 2
                                      MK IIIA=    0.0
YACHT DOWNTOWN                        ISSUED:   JUN 20, 1979
---- FREEBOARDS -------------------- RM -------- PROP & CB
FJS  1.086 DMS   2.034 RSBS1 0.0   ARM    83.71  PF  0.8500
FFS  1.051 PDS   0.651 RSBS2 0.0   BRM    83.44  PS  0.5100
FFIS 1.050 ESDS  0.0   RSBS3 0.0   CRM    83.71  DF  0.0242
FFDS 1.060 DMT   0.0   RSBS4 0.0   DRM    82.51  EMF 0.0087
FBIS 1.072 PDT   0.0   RSBS5 0.0   RM     83.34  PRDC .0
FMDS 1.090             RSBS6 0.0   RMC    83.34  PDC 0.6510
FAIS 1.148 SINK  0.0   BSC   0.0   TR   33.4460  RD  2.0340
FAS  1.159 TRIM -0.0   BBS   0.64  SV   -0.2343
--- L/FC -------------------- D/DC ------- MK IIIA -------
FDC    0.1026  AGSL  3.3725  FDI  0.3900  SCA        0.0
ADC    0.5437  APSL  4.6048  FDIC 0.3638  DLFA       0.0
ADCC   0.5437  ACG1  0.1289  CMDI 0.4570  MRA        0.0
ADCP   0.9730  ACG2  0.1145  MDI  0.3190  CBFA       0.0
ADCC   0.1145  HGLA  1.6204  DMDI -.0010  CGFA       0.0
FB     0.9130  HGLI  1.8367  MDIA 0.2510  AGE DATE        1978
FM     1.0942  BDR   1.0403  DB   2.0111  SERIES DATE
               LWL   0.0     DD   0.0229  HULL CHANGE   8/78
                             DSPL  4276.  RIG  CHANGE   5/79
SAIL AREAS & PENALTIES
-- FORETR/MAIN --- PEN ----- MIZZEN & PEN -- SCHOON & PEN
LP           5.29  SL    0.0    RSAY      0.0   RSAB     0.0
RSAF        31.15  SPS   0.0    RSAK      0.0   RSAG     0.0
RSAM        27.42  HBS   0.0    YSAC      0.0
RSAM MIN    13.56  HB    0.0    RSAC      0.0   HBF    0.0
SATC        -0.81  BD    0.0                    BDF    0.0
RSAT        57.77  BAS   0.0    HBY    0.0      BADS   0.0
RSAL        57.77  I/P   0.0    BDY    0.0      I/PSF  0.0
SPIN        42.40  BLP   0.0    BADY   0.0      BLPS   0.0
SHR       14.6983  BTNS 0.0000  IY/PY  0.0      BTNSF  0.0
SCF        1.0000               BLPY   0.0
S          7.6004               BTNSY  0.0

--- COMMENTS -------------------------------------------
```

# ***** Calculated Values *****

## *** Hull ***

| | | | |
|---|---|---|---|
| LBG | 10.24 | LBGC | 11.09 |
| BFC | 0.437 | | |
| FOCNUM | 0.039 | FOCDEN | 0.382 |
| FMAX | 0.751 | FOC | 0.1026 |
| AGSL | 3.372 | APSL | 4.605 |
| BAC | 2.498 | APSL1 | 4.605 |
| ACG1 | 0.129 | AOCP | 0.973 |
| ACG2 | 0.114 | AOCP1 | 0.211 |
| AOCG | 0.114 | AOCY | 1.031 |
| AOC1 | 0.0 | AOC2 | 0.544 |
| AOC | 0.5437 | AOCC | 0.5437 |
| HGLA | 1.6204 | HGLI | 1.8367 |

## *** Depth ***

| | | | |
|---|---|---|---|
| FDI | 0.390 | MDI | 0.319 |
| CMDI | 0.457 | DMDI | -0.001 |
| MDIA | 0.2510 | BDR | 1.0403 |

## *** Draft & Freeboard Corrections ***

| | | | |
|---|---|---|---|
| CD | 0.0 | WCBC | 0.0 |
| DB | 2.0111 | DSPL | 4275.770 |
| RDADD | 0.603 | RD | 2.0340 |
| DD | 0.023 | FB | 0.913 |
| FM | 1.094 | | |

## *** Engine & Propeller Factor ***

| | | | |
|---|---|---|---|
| EM | 473.76 | EMF | 0.0087 |
| PS | 0.510 | PSL | 0.050 |
| DF | 0.024 | PDC | 0.6510 |

## *** Center of Gravity Factor ***

| | | | |
|---|---|---|---|
| ARM | 83.71 | CRM | 83.71 |
| BRM | 83.44 | DRM | 82.51 |
| RM | 83.3411 | RMC | 83.3411 |
| TR | 33.4460 | SV | -0.2343 |

## *** Sails ***

| | | | |
|---|---|---|---|
| I | 12.010 | ISPA | 11.886 |
| TCI | 0.1244 | LP | 5.29 |
| SLPEN | 0.0 | SPSPEN | 0.0 |
| HBSPEN | 0.0 | IC | 12.0102 |
| JC | 3.5300 | SPIN | 42.3994 |
| RSAFA | 2.1843 | RSAF | 31.1547 |
| BALPEN | 0.0 | EC | 5.4500 |
| BASPEN | 0.0 | HDPEN | 0.0 |
| HBPEN | 0.0 | BDPEN | 0.0 |
| BLPEN | 0.0 | BL1P | 0.0000 |
| BL2P | 0.0000 | BL3P | 0.0000 |
| BL4P | 0.0000 | BL5P | 0.0 |
| BTNPEN | 0.0000 | PC | 13.1100 |
| RSAMAR | 2.4089 | RSAMA | 27.4162 |
| RSAM | 27.4162 | RSAMM | 13.5589 |
| SATC | -0.8050 | RSAT | 57.7658 |
| RSALA | 35.6410 | RSAL | 57.7658 |
| S2 | 57.7658 | SHR | 14.6983 |
| SCF | 1.0000 | | |

```
OPTIONS? AWD=5.53 BWD=5.53 CWD=5.53 DWD=5.53 APD=53. BPD=106.5 CPD=51.5
OPTIONS? DPD=105.5 FFM=1.030 FAM=1.167 FBIM=1.057 FJM=1.030 BWL=2.678
OPTIONS? P=12.884 PRINT=1


INTERNATIONAL OFFSHORE RULE             RATING CERTIFICATE NO.
MARK III AMEND. TO NOV. 1978            MEASUREMENTS IN METERS AND KGS.
DESIGNER FARR                           +-------------------------------+
BUILDER  COOKSON           BUILT 1978  |YACHT DOWNTOWN                 |
HULL#                   HULL DT  8/78  |SAIL# KZ-3776                  |
                         RIG DT  5/79  |OWNER:                         |
RIG:  SLOOP                             |WILLIAM C. MARTIN              |
KEEL: FIXED KEEL W/O MOV. APPEND.       |115 DEPOT ST.                  |
PROP: FOLDING                           |ANN ARBOR, MI  48104           |
INST: OUT OF APERTURE - EXP. SHAFT      +-------------------------------+


 ----HULL-------------------------------------------------------------------
LOA   11.289 BMAX      3.839 FF         1.030  AW       20.50  APD     53.000
FGO    0.559 B         3.461 FFI        1.030  BW       43.30  BPD    106.500
AGO    0.487 BWL       2.678 FFD        1.046  CW       20.50  CPD     51.500
LBG   10.243 BF        0.437 FMD        1.083  DW       43.30  DPD    105.500
GSDA   0.571 BFI       0.780 FAI        1.154  AWD       5.530 PL    2000.000
GSDF   0.501 BAI       2.788 FA         1.167  BWD       5.530 SBMAX    7.180
FD     1.450 BA        2.498 VHAI       0.861  CWD       5.530 SPD      7.580
CMD    1.547 GD        0.645 VHA        0.748  DWD       5.530 SDM      5.680
MD     1.409 Y         0.487 BHAI       0.0    MAW1      0.0   MAW2     0.0
OMD    1.089 DM        2.041 BHA        0.0    MACG1     0.0   MACG2    0.0
GDFI   0.0   GLAI      0.0   ESD        0.003  MACL      0.0   MACO     0.0
EW   210.0   EWD       2.256 PD         0.652  PRD       0.510 PBW      0.130
ESL    0.800 ESC       0.285 ST1        0.029  ST2       0.110 ST3      0.112
CD     0.0   WCBA      0.0   WCBB       0.0    CBDA      0.0   CBDB     0.0
                                               CBLDA     0.0   CBLDB    0.0
------FORETRIANGLE------   ---------MAINSAIL----------  -------------
I     12.051 SPL       3.530 P         12.884  HB        0.211 PC      12.8840
J      3.530 SPS       1.741 E          5.450  BL1       0.850 IC      12.0512
LPG    5.250 SL       11.890 BAS        1.566  BL2       0.850 JC       3.5300
LPIS   5.250 SMW       6.350 BAL        0.150  BL3       0.959 EC       5.4500
FSP    0.040 HBS       0.0   BD         0.155  BL4       0.959 MXSL    11.9296
FJ     1.030 TCI       0.165                   BL5       0.0   MXSMW    6.3540
FBI    1.057 SFJ       0.634 MW         0.147  BLP       2.800 MXLP     5.2950
IG    11.939 ISP      12.023 GO         0.172


------------------------------------- +-RATINGS---MK III--+-MK IIIA--+
L     9.5978  DC    0.0098  SC    7.5706 |  FEET:     27.5   |   0.0    |
B     3.4610  FC   -0.0258  MR    8.9656 +-------------------+----------+
D     1.0715  EPF   0.9672  R     8.3936 |  METERS:    8.39  |   0.0    |
CGF   0.9680  MAF   1.0000  RA    0.0    +-------------------+----------+
SMF   1.0000  DLF   1.0185  CBF   1.0000
LRP   1.0000                                               JUN 20, 1979

----------------------------------------------------------
```

```
                SAIL# KZ-3776     MK III =    8.3936  PAGE 2
                                  MK IIIA=    0.0
YACHT DOWNTOWN                    ISSUED:  JUN 20, 1979
---- FREEBOARDS -------------------- RM -------- PROP & CB
FJS  1.051 DMS   2.034 RSBS1 0.0     ARM    74.86  PF   0.8500
FFS  1.051 PDS   0.651 RSBS2 0.0     BRM    78.69  PS   0.5100
FFIS 1.050 ESDS  0.0   RSBS3 0.0     CRM    77.04  DF   0.0242
FFDS 1.060 DMT   0.0   RSBS4 0.0     DRM    79.44  EMF  0.0086
FBIS 1.068 PDT   0.0   RSBS5 0.0     RM     77.51  PRDC .0
FMDS 1.090             RSBS6 0.0     RMC    77.51  PDC  0.6521
FAIS 1.148 SINK  0.021 BSC   0.0     TR   36.9513  RD   2.0405
FAS  1.159 TRIM  0.029 BBS   0.64    SV   -0.0104
--- L/FC -------------------- D/DC ------- MK IIIA -------
FDC    0.0753  AGSL  3.3406  FDI  0.4037  SCA        0.0
ADC    0.5698  APSL  4.5455  FDIC 0.3735  DLFA       0.0
ADCC   0.5698  ACG1  0.1544  CMDI 0.4635  MRA        0.0
ADCP   0.9968  ACG2  0.1428  MDI  0.3255  CBFA       0.0
ADCC   0.1428  HGLA  1.6204  OMDI 0.0055  CGFA       0.0
FB     0.9131  HGLI  1.8367  MDIA 0.2559  AGE DATE       1978
FM     1.0848  BDR   1.0378  DB   2.0113  SERIES DATE
               LWL   0.0     DD   0.0292  HULL CHANGE    8/78
                             DSPL  4361.  RIG  CHANGE    5/79
SAIL AREAS & PENALTIES
-- FORETR/MAIN --- PEN ----- MIZZEN & PEN -- SCHOON & PEN
LP           5.29  SL    0.0     RSAY     0.0    RSAB     0.0
RSAF        31.27  SPS   0.0     RSAK     0.0    RSAG     0.0
RSAM        26.74  HBS   0.0     YSAC     0.0
RSAM MIN    13.65  HB    0.0     RSAC     0.0    HBF    0.0
SATC        -0.70  BD    0.0                     BDF    0.0
RSAT        57.31  BAS   0.0     HBY    0.0      BADS   0.0
RSAL        57.31  I/P   0.0     BDY    0.0      I/PSF  0.0
SPIN        42.53  BLP   0.0     BADY   0.0      BLPS   0.0
SHR       14.5597  BTNS  0.0000  IY/PY  0.0      BTNSF  0.0
SCF        1.0000                BLPY   0.0
S          7.5706                BTNSY  0.0

--- COMMENTS --------------------------------------------
```

## ♦♦♦♦♦ CALCULATED VALUES ♦♦♦♦♦



### ♦♦♦ HULL ♦♦♦

| | | | |
|---|---|---|---|
| LBG | 10.24 | LBGC | 11.09 |
| BFC | 0.437 | | |
| FOCNUM | 0.029 | FOCDEN | 0.381 |
| FMAX | 0.751 | FOC | 0.0753 |
| AGSL | 3.341 | APSL | 4.546 |
| BAC | 2.498 | APSL1 | 4.546 |
| ACG1 | 0.154 | AOCP | 0.997 |
| ACG2 | 0.143 | AOCP1 | 0.219 |
| AOCG | 0.143 | AOCY | 1.057 |
| AOC1 | 0.0 | AOC2 | 0.570 |
| AOC | 0.5698 | AOCC | 0.5698 |
| HGLA | 1.6204 | HGLI | 1.8367 |

### ♦♦♦ DEPTH ♦♦♦

| | | | |
|---|---|---|---|
| FDI | 0.404 | MDI | 0.326 |
| CMDI | 0.464 | OMDI | 0.005 |
| MDIA | 0.2559 | BDR | 1.0378 |

### ♦♦♦ DRAFT & FREEBOARD CORRECTIONS ♦♦♦

| | | | |
|---|---|---|---|
| CD | 0.0 | WCBC | 0.0 |
| DB | 2.0113 | DSPL | 4360.602 |
| RDADD | 0.603 | RD | 2.0405 |
| DD | 0.029 | FB | 0.913 |
| FM | 1.085 | | |

### ♦♦♦ ENGINE & PROPELLER FACTOR ♦♦♦

| | | | |
|---|---|---|---|
| EM | 473.76 | EMF | 0.0086 |
| PS | 0.510 | PSL | 0.050 |
| DF | 0.024 | PDC | 0.6521 |

### ♦♦♦ CENTER OF GRAVITY FACTOR ♦♦♦

| | | | |
|---|---|---|---|
| ARM | 74.86 | CRM | 77.04 |
| BRM | 78.69 | DRM | 79.44 |
| RM | 77.5095 | RMC | 77.5095 |
| TR | 36.9513 | SV | -0.0104 |

### ♦♦♦ SAILS ♦♦♦

| | | | |
|---|---|---|---|
| I | 12.051 | ISPA | 11.886 |
| TCI | 0.1654 | LP | 5.29 |
| SLPEN | 0.0 | SPSPEN | 0.0 |
| HBSPEN | 0.0 | IC | 12.0512 |
| JC | 3.5300 | SPIN | 42.5327 |
| RSAFA | 2.2023 | RSAF | 31.2717 |
| BALPEN | 0.0 | EC | 5.4500 |
| BASPEN | 0.0 | HDPEN | 0.0 |
| HBPEN | 0.0 | BDPEN | 0.0 |
| BLPEN | 0.0 | BL1P | 0.0000 |
| BL2P | 0.0000 | BL3P | 0.0000 |
| BL4P | 0.0000 | BL5P | 0.0 |
| BTNPEN | 0.0000 | PC | 12.8840 |
| RSAMAR | 2.1626 | RSAMA | 26.7388 |
| RSAM | 26.7388 | RSAMM | 13.6517 |
| SATC | -0.6965 | RSAT | 57.3140 |
| RSALA | 34.7604 | RSAL | 57.3140 |
| S2 | 57.3140 | SHR | 14.5597 |
| SCF | 1.0000 | | |

```
 OPTIONS? PRINT=5
#EXECUTION TERMINATED
#
#LIST IORTEST
>       1       DOWNTOWN
>       2       KZ-3776
>       3       William C. Martin
>       4       115 Depot St.
>       5       Ann Arbor, MI  48104
>       6       FARR
>       7       COOKSON
>       8
>       9       1978,
>      10       8,78,
>      11       5,79,
>      12       SLOOP
>      13       FIXED KEEL W/O MOV. APPEND.
>      14       FOLDING
>      15       OUT OF APERTURE - EXP. SHAFT
>      16       2,
>      17       0,
>      18       11.289,
>      19       .559,
>      20       .487,
>      21       10.243,
>      22       .571,
>      23       .501,
>      24       1.450,
>      25       1.547,
>      26       1.409,
>      27       1.089,
>      28       0.,
>      29       210.,
>      30       .80,
>      31       0.,
>      32       0.,
>      33       3.839,
>      34       3.461,
>      35       2.654,
>      36       .437,
>      37       .780,
>      38       2.788,
>      39       2.498,
>      40       .645,
>      41       .487,
>      42       2.034,
>      43       0.,
>      44       2.256,
>      45       .285,
>      46       0.,
>      47       1.051,
>      48       1.050,
>      49       1.060,
>      50       1.090,
>      51       1.148,
>      52       1.159,
>      53       .861,
>      54       .748,
>      55       0.,
>      56       0.,
>      57       0.,
>      58       .651,
>      59       .029,
>      60       0.,
```

```
>     61      20.5,
>     62      43.3,
>     63      20.5,
>     64      43.3,
>     65      4.90,
>     66      4.90,
>     67      4.90,
>     68      4.90,
>     69      0.,
>     70      0.,
>     71      0.,
>     72      .510,
>     73      .110,
>     74      0.,
>     75      0.,
>     76      42.,
>     77      89.,
>     78      42.,
>     79      90.,
>     80      2000.,
>     81      7.180,
>     82      7.580,
>     83      5.68,
>     84      0.,
>     85      0.,
>     86      0.,
>     87      .130,
>     88      .112,
>     89      0.,
>     90      0.,
>     91      3.53,
>     92      5.25,
>     93      5.25,
>     94      .040,
>     95      1.086,
>     96      1.072,
>     97      11.939,
>     98      3.53,
>     99      1.741,
>    100      11.89,
>    101      6.35,
>    102      0.,
>    103      .634,
>    104      12.023,
>    105      13.11,
>    106      5.45,
>    107      1.566,
>    108      .150,
>    109      .155,
>    110      .147,
>    111      .172,
>    112      .211,
>    113      .850,
>    114      .850,
>    115      .959,
>    116      .959,
>    117      0.,
>    118      2.8,
>    119      .85,
>    120      1.,
>    121      1.,
>    122      1.051,
>    123      1.159,
>    124      1.072,
>    125      1.086,
*END OF FILE
*
```

## CALCULATED VALUES DEFINITIONS

Only those values not defined in the IOR rule book are defined here.

| PARAMETER | DEFINITION |
|---|---|
| **HULL** | |
| BFC | = BF or BFI, whichever is less |
| FOCNUM | = the numerator of FOC. |
| FOCDEN | = the denominator of FOC. |
| FMAX | = 1.5*GSDF*.25*B/(.25*B+GDFI) See 300.3 |
| BAC | = BA or BAI, whichever is less. |
| APSL1 | = GSDA/(VHAI-VHA+FA-FAI) See 332.5 |
| AOCP1 | = (FA-VHA-.018*LBGC) See 332.7 |
| AOCY | = AOC+Y |
| AOC1 | = ABS(.6*AOC) See 333. |
| AOC2 | = 1.25*GSDA See 333. |
| **DRAFT & FREEBOARD CORRECTIONS** | |
| RDADD | = .3*DB See 510. |
| **SAILS** | |
| ISPA | = (IG+(IG*(GO-MW))/(J-GO+MW))-.04*B See 829.1 |
| SLPEN | = SL penalty |
| SPSPEN | = SPS penalty |
| HBSPEN | = HBS penalty |
| RSAFA | = .125*JC*(IC-2.*JC) See 830. |
| BALPEN | = BAL penalty |
| BASPEN | = BAS penalty |
| HDPEN | = Mainsail Head penalty. See 846.2 |
| HBPEN | = HB penalty |

| | |
|---|---|
| BDPEN | = BD penalty |
| BLPEN | = BLP penalty |
| BL1P-BL5P | = BL1 - BL5 penalties |
| BTNPEN | = Total batten penalties |
| RSAMAR | = .2*EC*(PC-2.*E) See 849.1 |
| RSAMA | = .35*(EC*PC)+RSAMAR See 849.1 |
| RSAMM | = .094*(IC*IC) See 850. |
| RSALA | = 1.3*RSAM See 890.1 |
| S2 | = S**2 See 891. |

# LIMITING DRAFT
## (LIMIT)

I. ABSTRACT

This module performs the required calculations to obtain the drafts fore and aft in the undamaged condition at which the ship must be proceeding in order to survive flooding at each specified group of compartments.

These drafts are calculated for survival with the waterline tangent to the margin line at each of seven arbitrary trim angles.

II. Menu INPUT

ALL - Prompts the user for all input data

ADD - Add a new compartment to the list

ALTER - Change the description of a given compartment

DELETE - Subtract a compartment from the list

REQUIRED - Prompts the user to enter the groups of flooded compartments

III. Menu DEFAULT

DESIGN - Used to alter the design condition of the ship

TITLE - Used to give the ship name, ship number and the date

MARGIN - Used to change the default margin line locations (3 inches below the deck)

IV. Constraints on Input

The user must supply the module with a database generated by the Ships Hull Verification Module (VERIFY) in order to successfully calculate the limiting drafts. The user must "ENTER-DB" before "COMPUTE".

Other constraints are:

(1) The number of separate compartments is limited to 41

(2) Each group is limited to 15 compartments

V. Menu ENTER-DB

The following information will be retrieved from the user's database. (stored by VERIFY)

SDT - Ship's data table

STAIN - Array of station numbers

X - Array of longitudinal station distances from FP

NP - Number of points per station

CUFTON - Cubic feet per ton of water

DRAFT - Design draft of the vessel

LBP - Length between perpendiculars

STASPC - Station spacing

LOA - Length overall

TRIM Design trim of the vessel

XMID - Distances from FP to amidships

NSTAT - Number of stations

After this data is read in from the database the user will be asked where the compartment descriptions are located; in a file, in the database, or will be inputted from the terminal.

For input from a file, the following is the proper format:

Comp. No., Fwd. Bkhd. Location, Aft Bkhd. Location, Permeability,
(I3,3F10.3)

... where the locations are given in feet from the FP, positive aft.

For input from the database, the following are the proper database variable names:

NCMPS - The number of compartments

CMPNO - The compartment's number (NCMPS numbers)

CMPFWD - The forward bulkhead locations (NCMPS locations)

CMPAFT - The aft bulkhead locations (NCMPS locations)

PERMS - The compartment permeabilities (NCMPS perms)

VI. Menu PRINT

ALL - All output

DESIGN - The design condition of the vessel

<u>MARGIN</u> - The margin line heights of the stations

<u>INPUT</u> - The input compartment descriptions

<u>FLOODED</u> - The group of flooded compartments

<u>RESULTS</u> - The results of computation for given compartments flooded

VII. Menu FILL-DB

This menu choice does not save any additional database information.

# LINESGEN

## I. ABSTRACT

LINESGEN alters the offsets of a Series 60 parent hull to generate offsets for a ship being designed by the user. It is essentially the same as the module ALTER, except that it is limited to this single parent.

## II. Menu INPUT

- ALL - Prompts the user for all input data
- LBP - Length between perpendiculars
- BEAM - Beam of the vessel
- DRAFT - Draft of the vessel
- CB - Block coefficient
- LCB - Longitudinal center of buoyancy
- REQUIRED - There is none

## III. Constraints on Input

The input for the program must satisfy the following constraints:

| | |
|---|---|
| LENGTH/BEAM ratio | between 5.4 and 8.6 |
| Block coefficient | between .55 and 0.9 |
| Beam/Draft ratio | between 1.9 and 4.1 |

The location of the LCB is given as a percent of the LBP. For example, if the ship is 500 feet long and the LCB is located 5 feet aft of amidships, the correct entry would be -1.0. Also note that LCB is positive forward of amidship and negative aft.

The allowable location of the LCB is dependent upon the block coefficient. For ranges of block coefficients, the allowable ranges of the LCB are:

| Block Coefficient | LCB Location |
|---|---|
| .55 - .60 | -3.5 to .53 |
| .601 - .65 | -2.5 to 2.5 |
| .651 - .75 | -2.5 to 3.5 |
| .751 - .80 | .47 to 3.51 |
| .801 - .85 | .75 to 3.51 |
| .851 - .90 | .75 to 3.05 |

When entrance or run is too short for Series 60 offsets, a message will appear.

IV. Menu ENTER-DB

This option obtains the following variables from the user's database.

LBP BEAM DRAFT CB LCB

V. Menu DEFAULT

There are none.

VI. Menu PRINT

ALL - Outputs all of the following options

INPUT - The input variables

NORMAL - Normalized, Series 60 offsets

TABLE - Full scale results in a table format

OFFSETS - Full scale results in SHCP format

VII. Menu FILL-DB

This option fills the user's database with the following variables:

LBP BEAM DRAFT CB LCB

WL - The waterlines of the vessel

NSTAT - The number of stations

MAXP - The maximum number of points per station

STATN - The station numbers

NPF - The number of points per station

YF - The half-breadths

ZF - The heights above the baseline

JTEST - Breakpoint indicators

STASPC - Station spacing

# LONGITUDIONAL STRENGTH

## (STRENGTH)

I. ABSTRACT

STRENGTH calculates load, shear, and bending moment calculations for a ship in still water, in a hogging wave, and in a sagging wave. Hull information is taken from the Ship Data Table (in user's database) prepared by VERIFY; a weight curve is supplied by the user from a file, or interactively.

II. Menu INPUT

ALL - Prompts user to enter ship's weight curve

REQUIRED - The user is prompted to enter the weight curve if it has not already been entered

WEIGHTS - Weight curve

III. Menu DEFAULT

TITLE - Ship name, number and the date

W.HEIGHT - Prompts the user to enter an item from menu HEIGHT (default is 1.1 x SQRT (LBP))

W.CENTER - Wave center (default is Midship)

W.LENGTH - Wave length as a fraction of LBP (default is 1.0)

ALL - Prompts the user for all the above default information

IV. Menu HEIGHT

This menu is encountered when using the default menu choices W.HEIGHT and ALL.

LBP/20 - Wave height is set as LBP/20

INPUT - Enter wave height in feet

V. Menu WGT.DATA

This menu appears in the module's database subroutine. It asks the user how the weight curve is to be entered.

TERMINAL

FILE - The file must be in the format which follows.

| Line No. | Quantity | Format | Columns |
|---|---|---|---|
| 1 | forward limit of weight curve | F 10.3 | 1-10 |
| 2-42 | after limit of weight curve segment | F 10.3 | 1-10 |
| | weight between segments (long tons) | F 10.3 | 11-20 |
| | LCG of weight referred to midships (+ Fwd) | F 10.3 | 21-30 |
| | section modulus (optional) | F 10.3 | 31-40 |

A maximum of forty one (41) weight segments can be submitted. One segment is defined on each of the lines 2-42.

VI. Constraints on Input

The user must supply the module with a database generated by the SHIPS HULL VERIFICATION module (VERIFY).

Other constraints are

(1) Maximum number of weight curve segments of 41

(2) The weight distribution curve must extend at least as far forward as the first ship station, and at least as far aft as the last ship station

(3) The LCG of each segment is restricted to the middle 1/3 of its fore and aft boundaries.

VII. Database

The following information will be retrieved from the user's database upon the MOD.MAIN menu choice ENTER-DB.

SDT - Ship's database supplied by VERIFY module

STATN - Array of station numbers

NP - Number of points per station (interpolated offsets)

X - Array of longitudinal station distances (from FP)

<u>DRAFT</u> - Design draft of vessel

<u>LBP</u> - Length between perpendiculars

<u>TRIM</u> - Design trim of vessel

<u>NSTAT</u> - Number of stations

<u>CUFTON</u> - Cubic feet of water per ton of displacement

<u>XMID</u> - Midship location (distance from FP)

<u>LOA</u> - Length overall

The FILL-DB menu choice does not save any additional database information.

VIII. Menu PRINT

<u>INITIAL</u> - Display of input & default information

<u>STILL</u> - Still water results only

<u>SAG</u> - Results for a sagging wave only

<u>HOG</u> - Results for a hogging wave only

<u>RESULTS</u> - The user is supplied the input and default information, as well as longitudinal strength calculations for hogging, sagging and still water conditions

<u>ALL</u> - The user is supplied the input and default information, as well as longitudinal strength calculations for hogging, sagging and still water conditions

# MORC Rating Program
(MORC)

## ABSTRACT

This program performs all the necessary calculations required to determine a sailboat's rating under the current M.O.R.C. rating rule. The program allows the user to modify any of the input values and examine the effect of the rating. The results of the calculations may be printed in several forms.

## USER GUIDELINES

A file should be established by the user prior to the execution of this program. The data file should contain all the measurements in the following format.

| | |
|---|---|
| Line 1 | General information FORMAT(20A4) |
| Lines 2-43 | Measurements FORMAT(F15.5) |

| Line # | Measurement | Line # | Measurement |
|---|---|---|---|
| 2 | LOA | 23 | J |
| 3 | OHF | 24 | SL |
| 4 | OHA | 25 | SPL |
| 5 | 4%OHF (FPOHF)* | 26 | SMW |
| 6 | 4%OHA (FPOHA) | 27 | LP |
| 7 | 4%TW (FPTW) | 28 | BFO |
| 8 | TH | 29 | FTO |
| 9 | OHT | 30 | WT |
| 10 | BXM | 31 | LKWT |
| 11 | BWL | 32 | IKWT |
| 12 | DR | 33 | IBW |
| 13 | FBF | 34 | ESW |
| 14 | FBA | 35 | CBW |
| 15 | P | 36 | CBA(s) (CBA) |
| 16 | B | 37 | SG |
| 17 | HDBD | 38 | BLADES |
| 18 | UP-BAT (BL1) | 39 | PS |
| 19 | UI-BAT (BL2) | 40 | DPROP |
| 20 | LI-BAT (BL3) | 41 | PF |
| 21 | LO-BAT (BL4) | 42 | PMIZ |
| 22 | P2W | 43 | BMIZ |

*The measurement variable name is used when prompted for an option except in the case where variable name is in parenthesis.

The program will ask the user for the name of the input file.

The user will be prompted with the message "OPTIONS?". The legal options are:

1) 0-42 input values to be charged
2) How the results will be printed

A proper response to the prompt consists of a list of assignments of the form variable=value, e.g.

OPTIONS? LOA=25. PRINT=1

All the measurements are legal variables and all their values are real numbers except for the variable PRINT where the value is a integer between 1 and 5 inclusive. The variable PRINT is used to control the form that the results to be printed.

| | |
|---|---|
| PRINT=1 | Prints Rating Certificate |
| PRINT=2 | Prints calculated values, corrections & rating |
| PRINT=3 | Prints corrections & rating |
| PRINT=4 | Prints just the rating (default) |
| PRINT=5 | Terminates program |

The program will always return the user to the prompting message "OPTIONS?" when a task is completed.

Reference "The MIDGET OFFSHORE RACING CLUB HANDBOOK" Section 5

# PLANEOPT

## I. ABSTRACT

PLANEOPT is a preliminary design computer program for planing boats. An efficient direct search routine allows the user to optimize the dimensions of a planing boat yielding a vessel with the best resistance characteristics for operation in either smooth or rough water. Alternately, the user can by-pass optimization and perform resistance computations for a vessel of given dimensions.

The following performance data will be computed:

1) Running trim angle
2) Displacement (may be input or computed)
3) Maximum stable (porpoising) trim
4) Total resistance
5) Hull viscous drag
6) Flap drag
7) Added wave drag
8) Effective horsepower
9) Wetted keel length
10) Wetted chine length

In addition, if optimization is performed, the following dimensions* are computed:

11) Optimum beam
12) Optimum deadrise angle
13) Optimum flap angle
14) Optimum length**
15) Optimum LCG position
16) Optimum VCG position

## II. Menu INPUT

ALL - User will be prompted for all input parameters in sequence

---

* optimization may be performed for any or all dimensions

** length optimization not recommended

DIMENSNS - User will be prompted to reply to menu DIMENSNS for input of vessel dimensions

Menu DIMENSNS

BEAM - beam in feet

DEADRISE - deadrise angle in degrees

FLAPANGL - flap angle in degrees

LENGTH - length on calm waterline in feet

LCG - longitudinal center of gravity position in feet forward of transom

VCG - vertical center of gravity position in feet above keel

SPEED - speed in feet per second

RANGE - User will be prompted to reply to menu YES/NO as to whether optimization is to be performed. If optimization is desired, the user will be prompted to input maximum and minimum dimensions available.

WEIGHTS - User will be prompted to reply to menu WTSUB for weight estimation method.

Menu WTSUB

BOTCONST - bottom weight per square foot is to be held constant. User will be prompted to input data for base boat from which weight data will be derived

WTCONST - vessel weight held constant. User is prompted to input displacement

USERSUPP - a user supplied subroutine will be used to estimate weight (SUBROUTINE WTEST2)(LENGTH, BEAM,DEADRISE,DISPLACEMENT))

OBJECT - User is prompted to respond to menu OBJECT to determine objective to be minimized and to menu CALM/WAV to determine if calm or rough water computations are to be performed

Menu OBJECT

RESIST - total resistance is to minimized

RES/LB - total resistance per pound displacement is to be minimized

RES/FTZ - total resistance per square foot of projected bottom area is to be minimized

Menu CALM/WAV

CALM - calm water computations are to be performed

ROUGH - rough water computations are to be performed

WATER - significant wave height in feet

GEOMETRY - user is prompted to reply to menu GEOMET for appendage and flap geometric data

Menu GEOMET

ALL - user is prompted to input all appendage and flap geometric data. Enter 0 for dimensions of non-existant elements

APPEND - appendage geometry

OUTDRIV - outdrive of stern drive geometry

FLAPS - flap geometry

SHAFT - exposed shaft and propeller geometry

REQUIRED - User is prompted exactly as OBJECT above

TITLE - ship name (up to 72 characters)

III. Menu DEFAULT

ALL - user will be prompted to change all default values

WATER - density and kinematic viscosity of water (values assumed for salt water @59°F)

CORREL - correlation allowance (assumed zero)

ALPHA - stopping condition for optimization (assumed 0.001)

IV. Menu COMPUTE

OPT - optimization is to be performed

NO.OPT - resistance computations without optimization are to be performed

V. Menu FORM

SHRTFORM - Savitsky's shortform (least accurate and least expensive method) procedure for resistance computations

LONGFORM - Savitsky's longform procedure for resistance computations

## VI. Constraints on Input

The range of validity of the input data is checked everytime the INPUT menu is invoked. In addition validity of empircal equations is checked against computed values when menu PRINT is invoked. In each case, a message is printed to indicate necessary changes to input.

## VII. DATABASE

The following items will be stored in the Database with the MOD.MAIN menu choice FILL-DB:

AAPP - frontal area of appendage ($ft^2$)

ABOTB - base boat bottom area ($ft^2$)

ADRIVE - frontal area of outdrive ($ft^2$)

BEAM - beam (ft)

BBASE - base beat beam (ft)

DDRIS - deadrise angle (degrees)

DDRISO - non-optimum deadrise angle (degrees)

BEAMO - non-optimum beam (ft)

BOTPCT - percentage of displacement represented by bottom structure

CAPP - chord length of appendage (ft)

CFLAP - chord length of flap (ft)

FLPANG - flap angle (degrees)

FLPANO - non-optimum flap angle (degrees)

RFRIC - viscous drag (lbs)

DISHFT - diameter of exposed shaft (ft)

R - total resistance (lbs)

SHFTAN - shaft angle (degrees)

HSIG - significant wave height (ft)

PAROPT - control variable for optimization

IWT - control variable for weight estimation

LWL - length on waterline (ft)

LWLO - non-optimum length (ft)

LAPP - longitudinal position of appendage

LBASE - length of base boat (ft)

LCG - LCG position (ft)

LCGO - non-optimum LCG position (ft)

LDRIVE - longitudinal position of outdrive (ft)

LPROP - longitudinal position of propeller (ft)

LSHFT - length of exposed shaft (ft)

NPAR - number of variables to be optimized

PAR - vectors of dimensions

TAPP - thickness of appendage (ft)

TRIM - equilibrium running trim angle (ft)

SPEED - speed (ft/sec)

VAR - vector of dimensions

VARHI - vector of maximum dimensional constraints

VARLO - vector of minimum dimensional constraints

K6 - vertical cg position (ft)

KG0 - non-optimum vertical cg position (ft)

VPROP - vertical position of propeller (ft)

WFLAP - width of flaps (ft)

DISPL - vessel displacement (lbs)

WTBASE - base boat weight (lbs)

YAPP - vertical position of appendage (ft)

YDRIVE - vertical outdrive position (ft)

EHP - effective horsepower

TITLE - vessel name

All of the above items are inputted from the Database with MOD.MAIN menu choice ENTER-DB.

## VIII. ERROR MESSAGES

PLANEOFT is equipped to print error messages indicating the nature of any problems occurring in execution. In one such case where convergence of an equation is not satisfied, the user is prompted for input. The following message will be printed:

```
?  LAMDA NOT CONVERGING, USE GRAPH
?  ENTER LAMDA FOR CV & CLO/TAU**1.1
?  CLO/TAU**1.1 = some no.  CV= some no.
?  ENTER LAMDA FROM GRAPH
?
```

The user should refer to the attached graph of Lift coefficients of a flat planing surface. Entering with the value of CV on the contours, and the value of CLO/TAU**1.1 on the ordinate axis, a value of LAMDA is read from the abscissa when the user inputs this value of LAMDA, execution will continue as normal.

# POWER

## I. ABSTRACT

This module computes calm water resistance using Taylor A, Taylor B or Series 60 resistance data. The user must supply the ship's particulars to obtain output of speed to length ratio, Reynolds number, resistance coefficients, resistance, EHP and SHP.

## II. Menu INPUT

ALL - Prompts user for all input data

SHIPID - Ship identification

DWL - Length on design waterline

LBP

BEAM

DRAFT

CB - Block coefficient

CP - Prismatic coefficient

PROPUL - Propulsive efficiency

SWET - Wetted surface. If SWET is entered as zero (0) it will be computed

REQUIRED - Prompts user only to enter the ship's speeds

## III. Menu DEFAULTS

RO - Water mass density 1.9905 ($^{lb - sec^2}/ft^4$)

GAMMA - Water weight density 64.0423 ($^{lb}/ft^3$)

GNU - Water kinematic viscosity 1.2817 E-05 ($^{ft^2}/sec$)

CFCORRCT - Roughness allowance coefficient 0.00040

APPALL - % of appendage allowance 3(%)

TOWTANK - Enter Ø for ITTC or 1 for ATTC

## IV. Menu TABLE

Selection of resistance table.

ATAYLOR

BTAYLOR

SERIES60

V. Menus ENTER-DB and FILL-DB

ALL - The user is to enter or fill all database information

SHIPID - Ship identification

DWL - Length on design waterline

LBP

BEAM

DRAFT

CB - Block coefficient

CP - Prismatic coefficient

PROPUL - Propulsive efficiency

SWET - Wetted surface

VI. Constraints on Input

| | minimum | maximum |
|---|---|---|
| (1) DWL | 40.0 | 1600.0 |
| (2) LBP | 40.0 | 1600.0 |
| (3) BEAM | 20.0 | 250.0 |
| (4) DRAFT | 4.0 | 120.0 |
| (5) CB | 0.50 | 0.95 |
| (6) CP | 0.50 | 0.95 |
| (7) PROPUL | 0.30 | 1.00 |

VII. Database

The modules database subroutines contain two menus, ENTER-DB and FILL-DB. When the user gains database access through MOD.MAIN menu choice ENTER-DB any or all of the following information will be retrieved.

SHIPNAME - Ship identification

DWL - Length on design waterline

<u>LBP</u>

<u>BEAM</u>

<u>DRAFT</u>

<u>CB</u> - Block coefficient

<u>CP</u> - Prismatic coefficient

<u>PROPUL</u> - Propulsive efficiency

<u>SWET</u> - Wetted surface

When the user gains database access through MOD.MAIN menu choice FILL-DB any or all of the above information will be saved.

VIII. Menu PRINT

<u>VESSEL</u> - Ship parameters

<u>FLUID</u> - Water parameters, roughness allowance coefficient, appendage allowance, and propulsive efficiency

<u>POWER</u> - Table of speed, speed to length ratio, Reynolds number, resistance coefficients, resistance, EHP and SHP

<u>RESULTS</u> - The user is supplied all the information included of the three menu items above.

<u>ALL</u> - The user is supplied all the information of menu items VESSEL, FLUID, and POWER.

# PROPELLER DESIGN
(SCREWOPT)

## I. ABSTRACT

The intent of this program is to provide the user with a reliable tool in determining the best propeller design to suit his or her particular needs. The program has the capability of optimizing the design based upon the 4 or 5 bladed Wageningen B-Screw Series and providing the user with a detailed description of the optimum wheel, ready for draftsmen. Standard output consists of wheel design particulars, thrust loading for cavitation requirements, weight and polar moment of inertia,expanded blade outline, and detailed blade section definition.

## II. Menu INPUT

ALL - Includes all the required data

BLADES - No. of blades

THRUST - Thrust per shaft (LBS)

SHP - Shaft horsepower per shaft (HP)

RAKE - Propeller rake in inches(+aft,-fwd.)

INHUB - Inner diameter of hub (FT)

OUTHUB - Outer diameter of hub (FT) (at directrix)

LENHUB - Length of hub (FT)

SPEED-A - Speed of advance (KNOTS)

DEPTH - Depth of centerline of shaft below water surface

BACKCAV - % back cavitation; either 5% or 10%

REQUIRED

a. ALL
b. DMIN - Min. prop. diameter
c. DMAX - Max. prop. diameter
d. SECTION - NACA or Wageningen (not now available)
e. AE/AO - Expanded area ratio
f. PITCH - (FT) (at the tip)
g. RPM
h. DIAM - (FT)

## III. Menu DEFAULT

DELTA - Stepsize for the search (.05)

EPSI - Stopping criterion for variables (.0001)

PRINT - Expanded print for optimization (off)

IV. Menu ENTER-DB

Inputs all data, except REQUIRED, from the database

- NBLD - No. of blades
- THRUST - Thrust per shaft
- SHP - SHP per shaft
- RAKE - Prop. rake in feet
- HUBID - Inner hub diameter
- HUBOD - Outer hub diameter
- HUBLEN - Length of the hub
- VA - Speed of advance
- HWL - Prop. depth
- BACK% - % back cavitation

V. Menu OPTYPE (in compute)

- NONE - No optimization is desired
- HJ - Hooke & Jeeves
- NM - Nelder & Mead

IV. Menu PRINT

- ALL - All of the output
- OPTDATA - Input condition and final optimized values for:
  - a. AE/AO
  - b. PITCH
  - c. RPM
  - d. DIAMETER
- DESIGN - Design data and areas
  Values are based on new advance coeff. of .95(of optimum)
  (Optimum advance coefficient) (see METHODS)
- DIM - Blade dimension and thickness
- WEIGHT - Weight estimates
- SECTION - Blade section dimensions

VII. Menu FILL-DB

Puts the following data into the data base:

All of the input data, except REQUIRED

- PPR - Propeller pitch ratio
- RPMS - RPM for the shaft
- POEFF - Optimized open water efficiency

DPROP - Prop. diameter

PITCH - Pitch (FT)

PSLIP - Prop. slip

AE/AO - Expanded area ratio

KT - Thrust coefficient

KQ - Torque coefficient

CAVN - Cavitation number

AO - Disk area (FT**2)

AD - Developed blade area (FT**2)

AP - Projected blade area (FT**2)

TLOAD - Thrust loading on blade (PSI)

WBLADE - Weight of the blades (LBS)

WHUB - Weight of the hub (LBS)

WPROP - WBLADE+WHUB

POLAR - Polar moment of inertia of prop.

## VIII. METHODS

Two optimization methods can be used to maximize open water efficiency. The four independent variables are:

1. Expanded area ratio
2. Pitch
3. Rpm
4. Diameter

The open water efficiency given by

$$\eta = \frac{J\,K_T}{2\pi K_Q}$$

is the objective function which is maximized.

$$K_T = \Sigma C_{x,y,z}[AE/AO]^x[P/D]^y[J]^z \qquad J = SLIP$$

$$K_Q = \Sigma D_{x,y,z}[AE/AO]^x[P/D]^y[J]^z \qquad J = \frac{V_A}{nD}$$

where $C_{x,y,z}$ ,X,Y,Z and
$D_{x,y,z}$ ,X,Y,Z

are coefficients of the polynomial as taken from:

Lammeren, Van Manen and Oosterveld, "The Wageningen B-Screw Series" SNAME Trans., No. 8 Nov. 14, 1969 page 15

for 4 and 5 bladed B-screws

A. Constraints imposed on the optimization:

Area ratio:

$.4 \leq AE/AO \leq 1.0$, for 4 bladed screws

$.45 \leq AE/AO \leq 1.05$, for 5 bladed screws

Diameter

$Dmin \leq Diameter \leq Dmax$

Pitch/Diameter ratio:

$.60 \leq P/D \leq 1.40$

Advance coefficient:

$0.0 \leq J \leq 1.6$

B. Thrust loading

5% cavitation

$$\tau_c = .0095692 + .4623*CAVN - .1941*CAVN^{2.0}$$

10% cavitation

$$\tau_c = -.0080547 + .654*CAVN - .30965*CAVN^{2.0}$$

Giving an upper limit for thrust loading of;

$$TOA = \tau_c*PRES2$$

where -

$$PRES2 = [V_A/7.12]^{2.0}$$

The actual thrust loading is programmed not to exceed TOA.

C. The blade dimensions are generated to the appropriate expanded area ratio using the formulation given in the above ref.

D. The required blade thickness at .25R is given by:

Am. Bureau of Shipping, "Rules for Building and classing steel Vessels" Sec. 37, ref. 6

E. A good propeller design keeps the advance coefficient below the peak value given by the optimum open water efficiency. This provides the design with some margin on RPM, to prevent the fast reduction of open water efficiency for decreases in RPM, and in turn, increases in advance coefficient.
- a new RPM is calculated at .95 (of optimum)

F. The weights and polar moment of inertia estimates are taken from:

Principles of Naval Architecture, SNAME, 1967

- based upon the propeller design parameters, and should be used only as preliminary estimates. (or use dimension to calculate)

G. The section dimensions are based on an NACA airfoil section shape, ref:

Abbott & Von Doenhoff, "Theory of Wing Sections" Dover, NY 1959

The dimensions are based on parent blade section given in this ref. Using lifting line theory, the section should exhibit good properties with regard to cavitation. A nearly constant pressure distribution is available from the leading edge to the .70 chord.

H. The Wageningen option for base section offsets is not available at this time.

I. NOTES !

To verify the optimum open water efficiency, it is a good idea to rerun the module with a different initial starting point. (change AE/AO, pitch, rpm and diameter) If approximately the same open water efficiency is obtained, then the user can put greater faith in the results. Also, the default stopping criteria can be changed to .00001 or less for (possibly) more accurate results.
The Nelder & Mead optimization seems to provide a solution closer to the optimum than the Hooke & Jeeves. Try both and compare .
***Negative open water efficiencies have been obtained, so check results carefully to see if they are reasonable.

J. This program was developed by Volker Elste for N.A. 574 (Winter '77) It was converted to "DEX" Std. format in May '77.

# SEAKEEPING
(SEAKEEP)

## I. ABSTRACT

This module calculates, dimensionally and non-dimensionally seakeeping properties of ships based on Series 60 models. (i.e. single screw merchant ships)

It is necessary to input the length between perpendiculars, beam, draft, block coefficient, significant wave height, and the speed(s) to be run. The resulting output will then be: heave at midship; pitch; bending at midship; mean added resistance; acceleration at stations 0, 5, 10, 15, and 20; the relative motion and relative velocity at stations 1, 2, 3, 4, and 20.

If the draft at station 1, the freeboard at station 1, and the propeller clearance near station 20 are inputted, then keel emergence, deck wetness, and propeller racing will also be calculated.

## II. Menu INPUT

ALL - Prompts user for all input data in sequence

SHIPNAME - Up to 20 characters to identify ship (No imbedded blanks)

LBP - Length between perpendiculars

BEAM, DRAFT

CB - Block coefficient

SWH - Significant wave height

TSTN1 - Draft at Station 1 (optional)

FSTN1 - Freeboard at Station 1 (optional)

PSTN20 - Distance from propeller disk to water surface (optional)

REQUIRED - Up to 20 speeds may be given, with or without even increment. If increment is not even, give negative number for number of speeds.

If TSTN1, FSTN1, and PSTN20 are given, keel emergence, deck wetness, and propeller racing will be computed.

## III. Constraints on Input

Block coefficient

greater than 0.55 and less than 0.90

Length to Beam Ratio

greater than 5.5 and less than 8.5

Beam to Draft Ratio

greater than 2.0 and less than 4.0

Wave Height to Length Ratio

greater than .015 and less than 0.100

IV. Menu DEFAULT

ALL - to change all default items

UNITS - to indicate English or Metric units of length. English is assumed.

SPEED - to indicate units of speed

Menu SPEED

KNOTS (assumed)

MPS - Meters per second

FPS - Feet per second

MPH - Miles per hours

FORMAT - indicate dimensional or non-dimensional output format. Dimensional is assumed.

V. Database Items - see menu INPUT for descriptions

| | | |
|---|---|---|
| SHIPNAME | DRAFT | TSTN1 |
| LBP | CB | FSTN1 |
| BEAM | SWH | PSTN20 |

VI. Output Form

A. English

RMS heave at midship in feet

RMS pitch in degrees

RMS bending at midship in foot-long tons

Mean added resistance in long tons

RMS acceleration in feet/sec$^2$

RMS relative motion in feet

RMS relative velocity in feet/sec

B. Metric

RMS heave at midship in meters

RMS pitch in degrees

RMS bending at midship in tonne-meters

Mean added resistance in tonnes
(1 tonne = 1000 kilograms = 9087 newtons)

RMS acceleration in meters/sec$^2$

RMS relative motion in meters

RMS relative velocity in meters/sec

C. Non-Dimensional

RMS heave at amidship * 1000/LBP

RMS pitch in degrees * 100

RMS midship(bending/RHO*G*LBP$^4$)*10,000,000

Mean→Added Resistance*10,000,000/RHO*G*LBP$^3$

RMS acceleration*1000/G

RMS relative motion/LBP*1000

RMS relative velocity*1000/G*LBP

# SHAFT ALIGNMENT ANALYSIS (SHAFT)

## I. ABSTRACT

This module finds bearing reactions and bearing reaction influence coefficients for a propulsion shaft. The user supplies information on shafting, such as: diameters, bearing spacings, weight of bull gear, weight of propeller, etc. The module produces:

1) bearing reactions for both hot and cold reduction gear

2) vertical movements from straight-line alignment, for both temperature conditions

3) bearing influence coefficients

An option allows the user to alter the solution by raising or lowering individual bearings.

## II. Menu INPUT

Input units are in inches and pounds.

ALL - Prompts user to enter all the necessary input parameters

BEARINFO - 1) Number of bearings
2) Bearing types, identified by the following number code:

| | |
|---|---|
| Reduction gear or line shaft bearing, pressure oil lubricated | = 1 |
| Reduction gear or line shaft bearing, not pressure oil lubricated | = 2 |
| Stern tube, oil lubricated | = 3 |
| Stern tube, water lubricated | = 4 |
| Stern tube, oil lubricated, adjacent to propeller (i.e. propeller isn't way out there supported by a strut) | = 5 |
| Stern tube, water lubricated, adjacent to propeller | = 6 |
| Intermediate strut, water lubricated | = 7 |
| Propeller support strut, water lubricated | = 8 |

3) Bearing warm-up rise, given in thousanths of an inch.

WEIGHTS - 1) Weight of bull gear, not including weight of bull gear shaft
2) Propeller weight in air

3) Weight of thrust bearing - This includes the weight of the collars, and other parts supported by it but does not include the shaft weight, or any part not supported by the shaft or collar.

<u>POSITION</u> - 1) Distance from center of gravity of propeller to center of load of most aft bearing.

2) Location of each bearing, longitudinally from a reference point - If the thrust bearing is forward of the reduction gear, reference point is the center of mass of the thrust collar. Otherwise, it is the midpoint of the forward bull gear bearing. For best results, give location of a bearing to be its center of load. Number bearings from forward to aft, with forward bull gear bearing being <u>one</u>.

3) Location of thrust bearing - This is the distance from the reference point to the center of mass of the thrust collar.

<u>DIAMETER</u> - 1) Bull gear shaft

2) Line shaft (i.e. between bull gear and stern tube)

3) Stern tube shaft

4) Tail shaft

Any outboard shafting in a multi-screw ship is assumed to have the diameter of the stern tube shaft, so you won't be asked. If any of the four types of shaft is not present, enter any non-zero number in response to the module's request.

<u>COUPLIFO</u> - 1) Number of couplings - If none, enter zero

2) Coupling locations, as measured from the reference point, beginning with the forward-most coupling - If zero has been entered as the number of couplings, the module skips this question.

3) Coupling weights, in the same order as (2) - Include only that part of the coupling lying outside the shaft diameter. This question is also skipped by the module if there are no couplings.

<u>MATERIAL</u> - Density ($lb/in^3$) of the propeller material

III. Menu DEFAULT

<u>ALL</u> - Prompts user to enter all default data

MODULUS - Youngs modulus

SHFTDENS - Shaft density (lb/in$^3$)

## IV. Constraints on Input

1) No more than 15 bearings
2) Maximum of 5 couplings
3) Maximum of one couplings per span between bearings
4) Maximum of one thrust bearing
5) Thrust bearing must be located either forward of the reduction gear, or in the first span of shafting aft of the gear.

The module has no way of accomodating a hollow shaft, nor of treating more or less than two bull gear bearings. What if there is no bull gear? Such would imply a direct-drive (diesel, most likely). The module should not be used for a direct-drive system, since the module cannot account for the several crankshaft bearings and the crankshaft, that would typically be part of the system.

## V. Database

The following items will be retrieved from the users database on menu choice ENTER-DB, or filled into it by the menu choice FILL-DB

NBEAR - Number of bearings

TYPE - Array of bearing types

BWARMRIS - Bearing warm-up rise

WTBG - Weight of bull gear

WTPROP - Propeller weight

WTTB - Thrust bearing weight

PROPDIST - Distance from center of gravity of propeller to center of load of most aft bearing

BEARLOC - Array of bearing locations

DIA - Array of shaft diameters

NCOUP - Number of couplings

XCOUP - Array of coupling locations

PROPDENS - Density of propeller material

SFTDENS - Density of shaft material

YMOD - Young's modulus

IALIGN - Database code stating whether alignment data will be entered

BR - Array of cold bearing reaction values

BRH - Array of hot bearing reaction values

RINS - Array of reaction influence numbers

CVBA - Array of cold bearing adjustment values

HVBA - Array of hot bearing adjustment values

NNN - Array of bearing numbers

VI. Menu COMPUTE

ALIGNED - Calculations for the aligned system are desired

UNALIGNED - Used if bearing alignment is to be changed

The only option is that of raising or lowering bearings after the straight-line solution is given. The menu name is RAISEBRN and the options are YES or NO. Option YES causes the module to ask what the change will be, and amount of change. Amount is to be in thousanths, and is positive for a raise, negative for a lower (e.g. 1.0 indicates up 0.001 inch; -1.0 down 0.001 inch).

VIII. Menu PRINT

ALL - All output

RESULTS - Bearing reactions and cumulative vertical bearing adjustments

INPUT - Prints input data

BRG.RCTN - Prints bearing reactions

VERT.ADJ - Output of cumulative vertical bearing adjustments

RCN.INFL - Output of reaction influence numbers

# SHIP HULL VERIFICATION

## (VERIFY)

## I. ABSTRACT

VERIFY processes lines data that has been placed in a database file by the modules LINESGEN, ALTER, or DIGITIZE, or lines data produced by any other process and placed in an MTS file. If an MTS file is to be processed by VERIFY, then data in that file must be in SHCP format. An appendix to this memo describes that format.

VERIFY is used before lines data is given to any module that is derived from SHCP (Ship Hull Verification Program, furnished by the U S Navy a long time ago), such as STABLE or DAMAGE (check the User Manual of any module you plan to run to see if "SHCP format" or prior use of VERIFY is specified). It establishes the Ship Data Table (SDT) used by these modules, checks for syntax and formatting errors in the lines file, and checks that the slope requirements (see Appendix) of SHCP have not been violated. It can also produce a body plan and station plots for graphic display, and provide them to a file for later input to the CALCOMP plotter.

## II. TYPICAL USE

The typical user will have entered offsets into a database file and is using VERIFY to prepare them for further analysis, just as noted in the abstract. Beginning at menu MOD.MAIN, the user's likely moves are

ENTER-DB
INITIAL
COMPUTE SLOPE
COMPUTE INTERP
FILL-DB

On the other hand, if the offsets are in an MTS file, the first move from MOD.MAIN is INPUT ALL. The user is led to COMPUTE from this point.

## III. Menu INPUT

WHERE - The user will be asked whether the offsets are located in a file or in a database

If DB is chosen, VERIFY obtains the following variables from the database

STATN - The station numbers

YF - Half-breadths of the vessel

ZF - Heights above the baseline

JTEST - Breakpoint indicators

LBP - Length between perpendiculars

STASPC - Station spacing

If FILE is chosen, it obtains the offsets from a file which must have the standard SHCP format: (see Appendix)

Additional inputs are

LBP - Length between perpendiculars

STASPC - Station spacing

If the offsets come from LINESGEN or ALTER via a database file, the two inputs above may be omitted.

DESIGN - The user will be prompted to enter the design conditions of the vessel. Draft and trim, displacement and trim, and displacement and LCG are the possible choices.

IV. Menu DEFAULT

The user is allowed to enter the following under the DEFAULT menu choice.

SHIPID - Ship name, number, and date

CUFTON - Cubic feet of water/ton

SCALE - Y and Z scale factors for the offsets

If the English units are used the scale factor should not be altered. To determine the proper scale factor, use the equation below.

$$\text{SCALE} = \frac{0.005}{\text{No. of feet/unit of length}}$$

i.e. for metres

$$\text{SCALE} = \frac{0.005}{3.28\ \text{(feet/metres)}}$$

V. ENTER-DB

The user is allowed to enter the offsets and/or SDT through the ENTER-DB command on MOD.MAIN by selecting the appropriate menu choice on menu OFFSETS.

INITIAL - Enters the initial offsets, (user defined)

INTERPOLATED - Enters only the interpolated offsets (SDT) prepared by VERIFY. Should therefore be called only on second and subsequent runs.

BOTH - Enters INITIAL and INTERPOLATED offsets

VI. Menu COMPUTE

Two options are allowed on the COMPUTE menu. SLOPE performs a check on the initial offsets to see if any slope violations exist. INTERP computes the interpolated offsets and establishes the SDT.

VII. Menu DRAW (only for a graphics terminal)

Two menus are encountered under this command. The user first specifies if he wishes to see the initial (INITIAL) or the interpolated (INTERP) offsets plotted. The user is then prompted for an option from menu DRAW.

These options are:

STATION - Draw the station, station breakpoints, and accompanying table of station offsets requested by the user.

BODYPLAN - Draw the bodyplan as specified by the offsets.

DONE - Permits the user to return to MOD.MAIN

VIII. FILL-DB

The FILL-DB option saves the following information in the user's database.

| VARIABLE | ARRAY SIZE | DESCRIPTION |
|---|---|---|
| NPF | 41 | No. of point/station (initial) |
| YF | 1189 | Half-breadths (initial) |
| ZF | 1189 | Heights (initial) |
| JBRKF | 328 | Breakpoints |
| JTEST | 1189 | Breakpoints |
| X | 41 | Station distances |
| STATN | 41 | Station numbers |
| NP | 41 | Number of points/station (interp) |
| SDT | 7134 | Ship data table |
| STASPC | N.A. | Station spacing |
| LOA | N.A. | Length Overall |
| LBP | N.A. | Length between perpendiculars |

| | | |
|---|---|---|
| XMID | N.A. | Amidships |
| TRIM | N.A. | Design trim |
| DRAFT | N.A. | Design draft |
| DISPL | N.A. | Design displacement |
| CUFTON | N.A. | Cubic feet of water/ton |
| VCG | N.A. | Vertical CG |
| LCG | N.A. | Longitudinal CG |
| NSTAT | N.A. | Number of stations |

## IX. Menu PRINT

The user may select any of the following options

<u>INITIAL</u> - Prints a table of the initial offsets

<u>INTERP</u> - Prints a table of the interpolated offsets

<u>DESIGN</u> - Prints the design conditions of the vessel

<u>ALL</u> - Prints INITIAL, INTERP and DESIGN

One additional option is available on the SEND command. The option, BODYPLAN, permits the user to create a bodyplan plot file of the vessel for the CAL-COMP plotter.

## X. APPENDIX - DESCRIPTION OF SHCP FORMAT

The format is (F6.3,2F7.3,I5), where the fields are

| | |
|---|---|
| STATION NUMBER | (F6.3) |
| HALF-BREADTH | (F7.3) |
| WATERLINE | (F7.3) |
| BREAKPOINT INDICATOR | (I5) |

Station distance from FP is obtained by multiplying station number by station spacing, thus a station forward of the FP must have a negative station number.

There must always be an odd number of stations submitted to describe the hull form, each of which must have a non-zero sectional area when entirely immersed. The minimum station consists of two offsets having vertical and horizontal spacings of at least 0.01 feet.

Longitudinal breakpoints (end of raised forecastle, end of skeg, etc.) are represented by 3 very closely spaced stations. These stations must be numbered so that all three are included in a single integration step. For example, if stations are numbered 0,1,2,3,4,5,6,7,..., with 5 being a breakpoint, integration will proceed correctly because Simpson's Rule takes groups 0,1,2; 2,3,4; 4,5,6,; etc. thus placing 5 and its two closely-spaced neighbors together.

41 stations are permitted.

Offsets must be submitted in ascending order by height above BL. Successive heights must increase, therefore tunnel sterns may not be represented. Horizontal straight lines are also ruled out except from the centerline to the first point and from the last point to the centerline. Any other line must have a slope of at least 0.002.

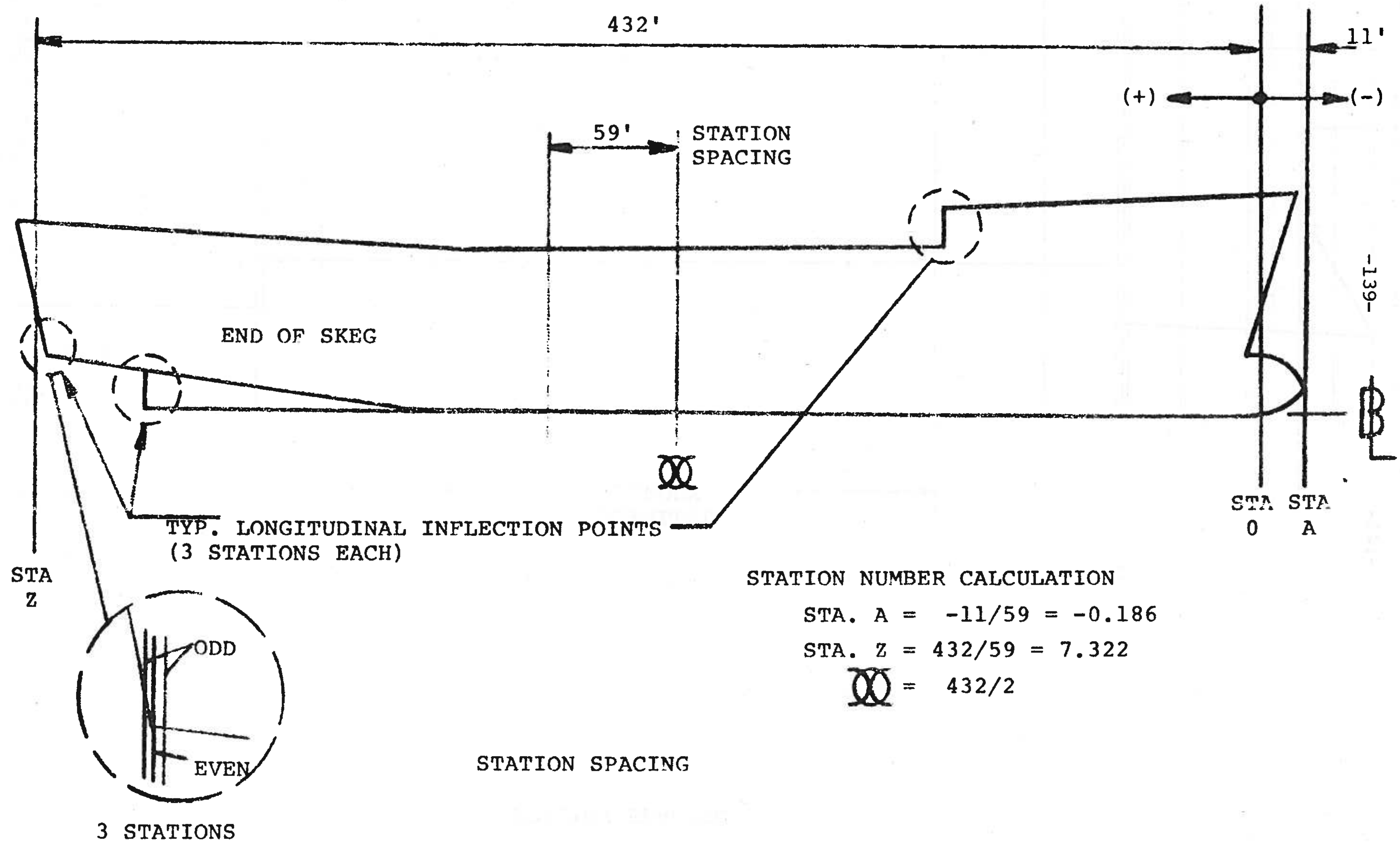
Breakpoint indicators are the following:

Normal points (knuckles, keels, etc) = +7777

Last point on each station = +8888

Last point on last station = +9999

Sketches are included in this manual to aid visualization of the instructions above. Further information is in SHCP documentation; to obtain this, users at UM can $Copy N.A.:SHCP.U




STATION NUMBER CALCULATION

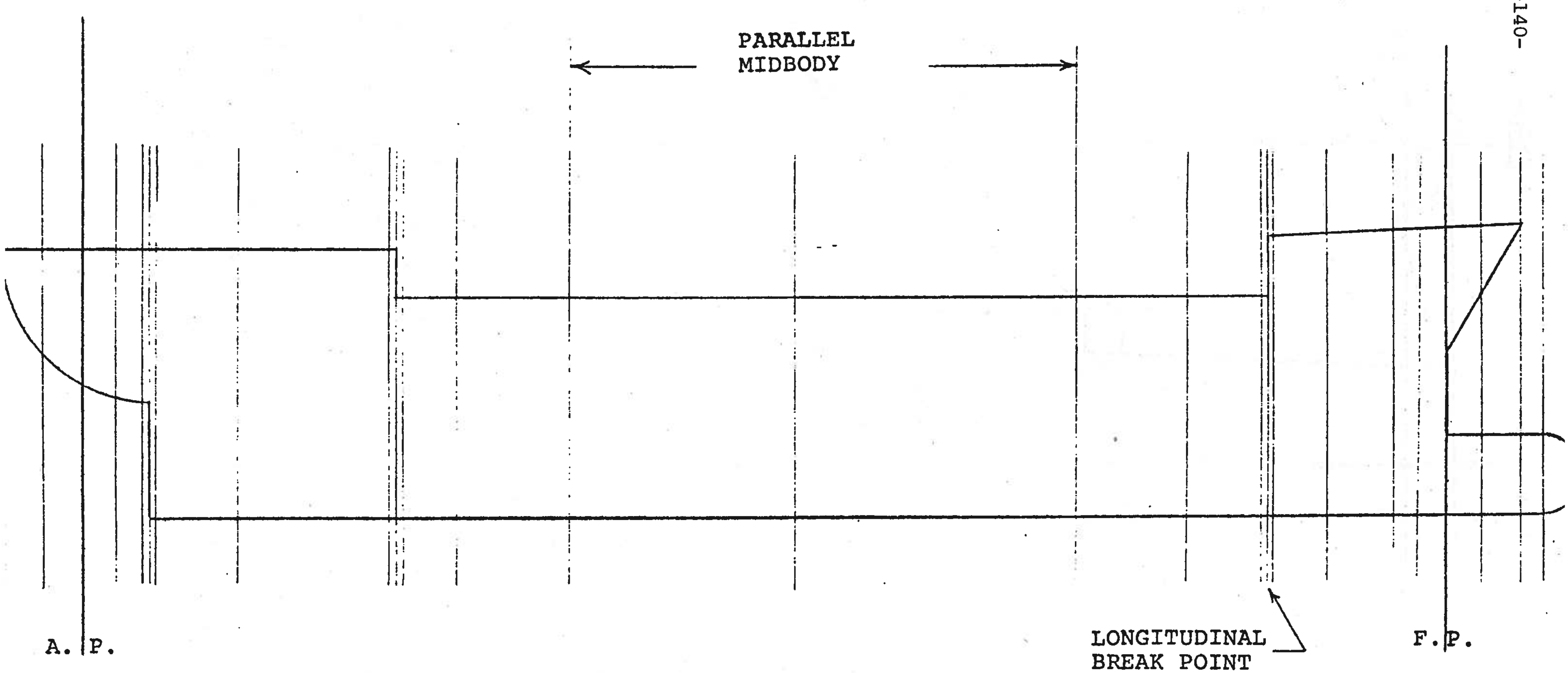
STA. A = -11/59 = -0.186

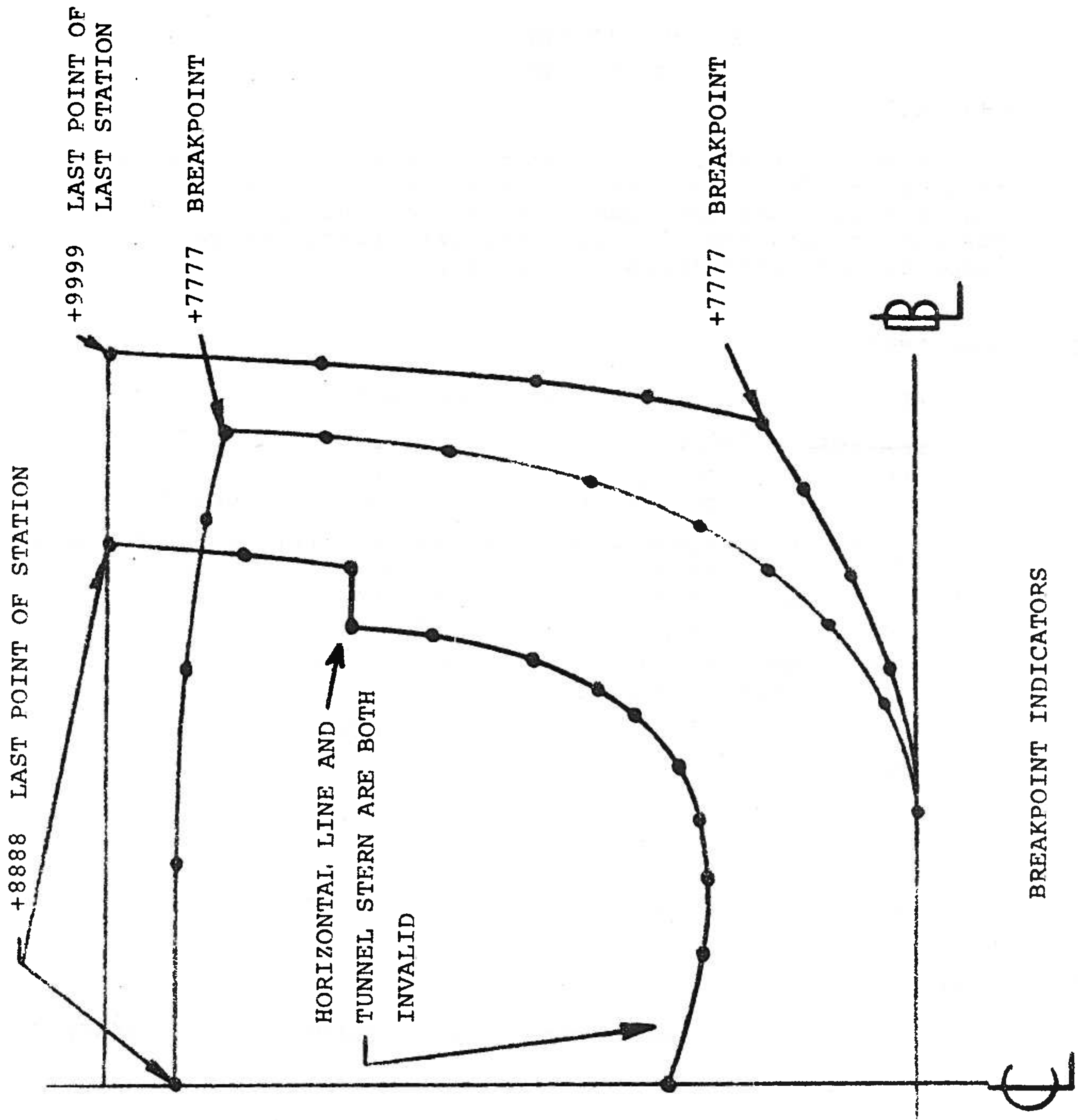
STA. Z = 432/59 = 7.322

ꝏ = 432/2

STATION SPACING

STATION SPACING




+9999 LAST POINT OF LAST STATION
+7777 BREAKPOINT
+7777 BREAKPOINT
+8888 LAST POINT OF STATION
HORIZONTAL LINE AND TUNNEL STERN ARE BOTH INVALID
BREAKPOINT INDICATORS

# STEAMCYCLE ANALYSIS
(STMCYCLE)

## I. ABSTRACT

STMCYCLE analyzes regenerative steam power cycles but does not perform detailed heat balances of practical plants. The user supplies the number of heaters and cycle conditions. The output consists of fuel rate, efficiency, flows, temperatures, enthalpies and pressure.

## II. Menu INPUT

<u>ALL</u> - Prompts user for all input data

<u>REQUIRED</u> - Prompts user for all input data

<u>HEATERS</u> - 1) Total number of heaters
2) Type of heaters i.e. DFT or SURFACE

<u>TIN</u> - Temperatures (F) of feedwater entering the heaters. The first entry should be that of the heater nearest the boiler, second the second nearest, third the third nearest, etc.

<u>TOUT</u> - Temperature (F) of feedwater leaving the heaters. Order as for TIN.

<u>TTD</u> - Terminal temperature difference. Order as for TIN

<u>INLET</u> - Pressure (psia) and temperature (F) of the steam at the turbine inlet.

<u>EXHAUST</u> - Pressure (in. Hg) at turbine exhaust (Not vacuum.)

<u>TSUPERH</u> - Temperature (F) at super heater outlet.

## III. Menu TYPE

This is a subsidiary menu of menu INPUT for item HEATERS.

<u>DFT</u>

<u>SURFACE</u>

## IV. Menu DEFAULT

<u>ALL</u> - Prompts user to enter all default data

<u>HHV</u> - Higher heating value of the fuel (Btu/lb)

<u>SHP</u> - Shaft horsepower

EFFBLR - Efficiency of boiler
EFFPMP - Efficiency of pump
VOLFEED - Specific volume of feedwater at pump (cuft/lb)
PRESDROP - Pressure drop, boiler to turbine (psi)
EFFTUR - Efficiency of turbine

V. Constraints on Input

(1) Steam conditions at the turbine inlet range from 600 psia to 1400 psia and 600 F to 1400 F.

(2) Turbine exhaust pressure in range 1 to 30 inches of mercury.

(3) Feedwater temperatures must be in the proper order, i.e. Temperatures decreasing as one gets farther away from the boiler.

(4) Correct output is dependent on correct input. The user must therefore provide protection against folly. For example, boiler efficiency can be entered as greater than 1.0, or a negative TTD can be specified, and will not be recognized by the module as erroneous.

VI. Database

The following information will be retrieved from the user's database upon MOD.MAIN menu choice ENTER-DB.

N - Number of heaters

NDFT - Number of DFT type heaters. This must equal N if NHTR = Ø, or must equal Ø if NHTR = N

NHTR - Number of SURFACE type heaters. This must equal N if NDFT = Ø, or must equal Ø if NDFT = N

PEXT - Turbine exhaust pressure (in. Hg)

PIT - Pressure at turbine inlet (psia)

TIT - Temperature at turbine inlet (F)

TSH - Temperature at super heater outlet (F)

TIN - Array of feedwater temperatures (F) entering the heaters. The first entry is for the heater nearest the boiler, second the second nearest, third the third nearest, etc.

TOUT - Array of feedwater temperatures (F) leaving the the heaters. Order as for TIN.

TTD - Array of terminal temperature differences. Order as for TIN.

The MOD.MAIN menu choice FILL-DB does not save any additional database information.

VII. Menu PRINT

<u>RESULTS</u> - Output of all calculated information

<u>TURBINE</u> - Turbine and compresser states

<u>FUEL</u> - HHV and fuel rate

<u>HEATERS</u> - All calculated heater specifics

<u>PUMP</u> - Pump heat loss and horsepower

<u>FLOW</u> - Feedwater flow rate

<u>CYCLEEFF</u> - Cycle efficiency

<u>ALL</u> - Output of all calculated information

# TRIM LINES (TRIML)

## I. ABSTRACT

This module performs the required calculations to obtain the final draft and trim conditions the ship will assume after flooding the specified compartments. Drafts fore and aft, quantity and LCG of flooded water are also available.

## II. Menu INPUT

ALL - Prompts the user for all input data

ADD - Add a new compartment to the list (See Section V)

ALTER - Change the description of a given compartment

DELETE - Subtract a compartment from the list

REQUIRED - Prompts the user to enter the groups of flooded compartments

## III. Menu DEFAULT

DESIGN - Used to alter the design condition of the ship

TITLE - Used to give the ship name, ship number and the date

## IV. Constraints on Input

The user must supply the module with a database generated by the Ships Hull Verification module (VERIFY) in order to successfully calculate the trim lines. The user must "ENTER-DB" before "COMPUTE".

Other constraints are:

(1) The number of separate compartments is limited to 41

(2) Each group of flooded compartments is limited to 15

## V. Menu ENTER-DB

The following information will be retrieved from the user's database (stored by VERIFY)

SDT - Ship's data table

STATN - Array of station numbers

X - Array of Longitudinal station distance from .FP.

NP - Number of points per station

CUFTON - Cubic feet per ton of water

DRAFT - Design draft of the vessel

LBP - Length between perpendiculars

STASPC - Station spacing

LOA - Length overall

TRIM - Design trim of the vessel

XMID - Distance from FP to amidships

NSTAT - Number of stations

After this data is read in from the database, the user will be asked where the compartment descriptions are located; in a file, in the database, or will be input from the terminal.

For input from a file, the following is the proper format:

Comp. No., FWD.BKHD.LOCATION,AFT BKHD.LOCATION,Permeability,

...All terminated by a comma. The locations are given in feet from the FP, positive aft.

For input from the database, the following are the proper database variable names:

NCMPS - The number of compartments

CMPNO - The compartment's number (NCMPS numbers)

CMPFWD - The forward bulkhead locations (NCMPS locations)

CMPAFT - The aft bulkhead locations (NCMPS locations)

PERMS - The compartment permeabilities (NCMPS perms)

VI. Menu PRINT

ALL - All output

DESIGN - The design condition of the vessel

INPUT - The input compartment descriptions

FLOODED - The group of flooded compartments

RESULTS - The results of computation in flooded condition

VII. Menu DBFILL

This menu choice does not save any additional database information.

## WEIGHTS
(WEIGHT)

I. ABSTRACT

This module computes hull steel, machinery, and outfitting weights for tankers, bulk carriers, dry cargo ships, container ships, and SEABEE vessels. In addition, it computes hull engineering weight for dry cargo ships only. The inputs are basic ship particulars, type of powerplant, and a choice of computing methods for tankers only.

II. Menu INPUT

- ALL - Prompts the user for all input data
- LBP - Length between perpendiculars
- BEAM,DRAFT,DEPTH - Standard
- DISPL - Displacement in long tons
- CB - Block coefficient (fraction)
- SHP - Shaft horsepower
- CREWSIZE - Number in crew
- REQUIRED -
  - Menu POWER
    - STEAM - Steam propulsion machinery
    - DIESEL - Diesel propulsion machinery
  - Menu SHIPTYPE
    - TANKER,BULKER,DRYCARGO,CONTANER,SEABEE
  - TANKER option:
    - Menu METHOD
      - BUXTON,SATO
    - BUXTON option
      - Menu BULKHEAD
        - PLAIN,CORRUGTD

III. Constraints on Input

| | minimum | maximum |
|---|---|---|
| LBP | 150 | 1500 |
| BEAM | 25 | 200 |
| DRAFT | 13 | 80 |

| | minimum | maximum | |
|---|---|---|---|
| DEPTH | 15 | 150 | |
| DISPL | VOL/36.1 | VOL/34.9 | (VOL=L*B*T*CB) |
| CB | .50 | .90 | |
| SHP | 500 | 60,000 | |

IV. Menu DEFAULTS

There are no defaulted inputs with this module.

V. Menu PRINT

ALL - Prints all of the following

VESSEL - Prints input particulars

POWERING - Prints powerplant type and SHP

METHOD - Method of calculation (Tankers only)
Type of bulkheads (BUXTON method only)

SHIPTYPE - Type of vessel

RESULTS - Results of weight calculations

VI. Menu SEND - Same as above

VII. Menu ENTER-DB

ALL,LBP,BEAM,DRAFT,DEPTH,DISPL,CB,SHP,CREWSIZE (Same descriptions as INPUT)

VIII. Menu FILL-DB

ALL,LBP,BEAM,DRAFT,DEPTH,DISPL,CB,SHP,CREWSIZE,

SHIPTYPE - Vessel type and powerplant type

WEIGHTS - Steel weight, machinery weight, outfit weight, hull engineering weight, and light ship weight.

IX. General notes concerning use of the database

This module is designed to use the same database as the Economics Module and the Building Cost Module. The weight results stored from this module can be used by the Building Cost Module, and the results from these two for the Economics Module.

# MODULE PROGRAMMER'S GUIDE

to the

Computer-Aided Design System

SPIRAL

The University of Michigan
College of Engineering
Department of Naval Architecture
and Marine Engineering

Bradley S. Winslow
John B. Woodward

June 26, 1978

# CONTENTS

I. *Introduction*

SPIRAL is an interactive computing system designed for use by the students of the Department of Naval Architecture and Marine Engineering. It consists of a design executive called DEX, database management routines to control the storage of data and to facilitate communication between individual computing tasks, and a number of computation programs known as modules. New modules may be added to the system as they are developed, and this manual is offered for the guidance of programmers preparing modules for the system.

SPIRAL offers several advantages to the interactive FORTRAN programmer. The design executive contains routines available to the module programmer to handle program control and data acquisition. Database management routines and high-level graphing routines are also available to the module programmer to enhance the operation of the program.

The principal purpose of SPIRAL is to enable the student designer to readily perform calculations of the "design spiral", hence the name. To help achieve this goal, the designers of the system have set forth guidelines for module programmers. All modules are designed to confront the user in a manner similar to all other modules. Programmers are therefore required to adhere to the method and style recommended by the system designers. The information needed to follow these structures is covered in this manual. It is assumed that the programmer has a prior working knowledge of FORTRAN and its use in interactive programming. A novice SPIRAL programmer will benefit from prior study of the sample program included in this manual, and from prior experience as a user of the system.

## II. *Discussion of Module Structure*

The designers of the SPIRAL system have developed a structure to which all modules of the system conform. It is believed that this structure provides the needed flexability to the programmer while providing a standard man-machine interface to a user of the system. A user feels more secure if he is confronted in a similar manner by all modules. To achieve this goal the module programmer incorporates into the module one or more of the routines furnished by DEX. These routines are catagorized into six groups: control, input, output, graphic manipulation, character manipulation and database. To introduce the routines, they are listed under those catagories.

| Control | Input | Output | Graphic Manipulation | Character Manipulation | Database |
|---|---|---|---|---|---|
| MENUIN | XYIN | MESOUT | GRAF | LFANYC | IGET |
| MENUXY | INTIN | | GLINE | LENOTC | RGET |
| MODERR | REALIN | | DSPLAY | LMATCH | AGET |
| ENDIT | CHARIN | | | LMOVEC | IPUT |
| WAIT | | | | LEQLCH | RPUT |
| SETDEV | | | | LNRCVT | APUT |

Routines most frequently used are discussed in detail in the next section. All are detailed in the appendix.

The construction of a module is straight-forward. Control through the module is handled by menus established by the module programmer and by the designers of the system. All modules contain the same "main program" which contains the main module menu "MOD.MAIN." The main program consists simply of this menu, a computed GO TO statement and the associated subroutine calls and statement labels. Its purpose is to transfer control to the part of the module specified by the menu item chosen by the user. The transfer of control from the menu "MOD.MAIN" should lead the user to another menu if applicable whose name is associated with the item choice from the "MOD.MAIN" menu. (Fig. 1)

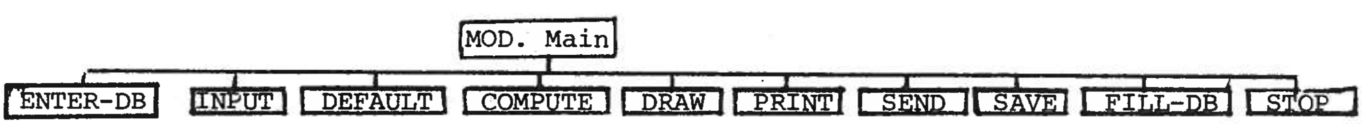



Fig. 1 The Ten Menu Options of MOD.MAIN

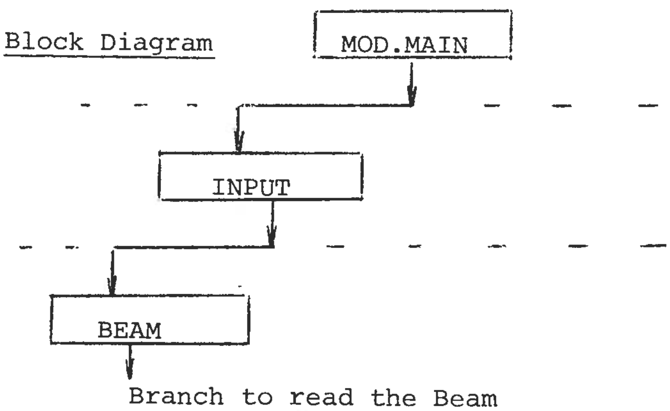
If the user should select the menu option INPUT from the menu MOD.MAIN, control should transfer from the main program (containing MOD.MAIN) to a subroutine called INPUT containing the menu "INPUT." The user would then select an item from the input menu. Through the use of a computed GO TO statement, the programmer could then branch within the INPUT subroutine to accept the desired user input. Figure 2 demonstrates a) The user terminal session and, (b) a block diagram of the control flow within the program.

Fig. 2

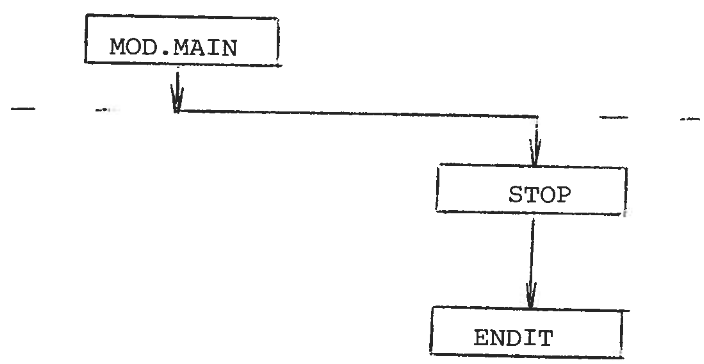
a) <u>User Terminal Session</u>

```
? Enter an item from Menu MOD.MAIN
? Input
? Enter an item from Menu INPUT
? Beam
? Enter Beam:
? 80
```

b) <u>Block Diagram</u>




Control is passed down through the module by successive levels of menus until control is sufficiently localized to perform an operation. The STOP command on the MOD.MAIN menu obviously need not have a menu associated with its name since its only purpose is to terminate the module (Fig. 3).



Each of the ten menu options on the MOD.MAIN menu is designed to perform a set of tasks associated with its name. Although the level of complexity will vary between modules the purpose of the commands should be the same. The purpose of each item on the MOD.MAIN menu are outlined below.

ENTER-DB - This command is used to *enter* the database values into the program variables. The choice may or may not lead the user to a menu.

INPUT - This then leads to a menu by which the user selects which inputs he has for the program. The menu choices ALL and REQUIRED should appear on the input menu.

DEFAULT - In the event that default values are to be changed when running the module, a user will select this menu choice to change the value. A typical example would be the value of cubic feet per ton for water. Default value of CUFTON is 35 for salt water, and may be changed to 36 for fresh water. Numerical constants that aren't normally inputted would appear on the DEFAULT menu.

COMPUTE - A user selects the command COMPUTE to perform the calculations involved in the module. The programmer should perform a check on the input data to assure that it both exists and is within valid ranges for the module. This menu choice may or may not have a menu associated with it.

DRAW - Should the module incorporate graphic displays, the user could request the displays to be drawn with this menu choice. The command will typically have a menu associated with its name.

PRINT - This command directs the user to a menu entitled PRINT. Its purpose is to cause the printing of either the input data, the results of the calculations, or both at the users terminal. The formatting details are left to the programmer.

SEND - This command is used to route the printing of information to the user-specified file or device. The command should lead the user to specify what device or file will be used; then it should lead the user to the menu PRINT as described above.

<u>SAVE</u> - The SAVE command performs essentially the same function for graphic output as the SEND command does for printed output. It should prompt the user for a file to store the graphic output and lead the user to the menu DRAW.

<u>FILL-DB</u> - This command is used to fill the database with values from the module. Information may be transferred from one module to another by the use of this command and the command ENTER-DB.

<u>STOP</u> - Terminates the use of the module.

The logical functions (routines) supplied by DEX that acquire information from the user return logical values associated with their names. Typically the value of .TRUE. is returned when the user has entered the appropriate information. These functions are designed to perform error checking on the input (i.e. if an integer value was requested, it will prompt the user for an integer value until an integer value is entered). These routines are designed to return a value of .FALSE. only if an end of file is encountered (Control-C or $ENDFILE). The Control-C or $ENDFILE is entered by the user in the event that he has made a mistake and wishes to return to the MOD.MAIN level. This can be accomplished with an IF statement where the FORTRAN statement RETURN is the contingent statement.

The novice module programmer will benefit from the study of the sample module program and sample terminal session in the appendix at this manual.

## III. *Discussion of Frequently Used Routines*

### Control Routines

1. ITEM=MENUIN(DNAME,ICOUNT,ITEMS,MESSAG)

MENUIN creates a menu having the name DNAME, a number of options equal to ICOUNT, and it prints a MESSAG (i.e. a message) to prompt the user. The names of the options on the menu are stored in the array ITEMS. When MENUIN is encountered during program execution, execution stops until the user enters one of the menu options. Having received such an input, MENUIN returns an integer value corresponding to the position of the option in the array ITEMS. This integer directs the subsequent flow of the program. An example illustrates:

```
      .
      .
      .
      LOGICAL NOTALL
      DOUBLE PRECISION INMEN(6)
      DATA INMEN/8HLBP     ,8HDRAFT   ,8HBEAM    ,
     +          8HCB      , 8HLCB     , 8HALL     ,
      .
      .
      .
   10 CONTINUE
      ITEM=MENUIN('INPUT   ',6,INMEN,'READY@'
      NOTALL=ITEM.LT.6
      GO TO (100,200,300,400,500,100),ITEM
      RETURN
      .
      .
      .
```

The menu named INPUT offers the user 6 options, LBP,DRAFT,BEAM,CB, LCB, and ALL. Each menu choice is an 8 character (including trailing blanks) array element from the double precision array INMEN. The user is prompted with the standard menu prompt plus the word READY desired by the programmer. Note the manatory @ appended to READY. The integer variable ITEM takes on the valve associated with the location of the INMEN item chosen. If ALL is chosen, ITEM assumes the value of 6, if LBP is chosen, ITEM is given the value of 1, etc. The value of ITEM is used to direct the program flow via the FORTRAN computed GO TO statement. Note that the logical variable NOTALL is used to determine if the item ALL was chosen. This variable is used to suppress the RETURN until all data has been entered.

2. CALL ENDIT

The call to ENDIT is used in the main program in place of the FORTRAN command STOP. This ensures that upon completion of the module control is returned to DEX, rather than to the computer's operating system.

Input Routines

1. IF(REALIN(ICOUNT,ARRAY,MESSAG))STATEMENT

REALIN generates a call to the user to enter ICOUNT number of items of real data for ARRAY, and prints MESSAG to prompt the user. REALIN is a logical function, and will therefore have the values TRUE or FALSE. Normally, only TRUE is expected, but FALSE is generated by a user response of CONTROL-C or $ENDFILE. Entry of faulty data (e.g. a character) merely produces an error message from DEX instructing the user to try again. An example illustrates:

```
DIMENSION WAVE(2)
LOGICAL REALIN,NOTALL
.
.
.
IF(.NOT.REALIN(2,WAVE,'ENTER WAVE HEIGHT and LENGTH@'))RETURN
IF(NOTALL)RETURN
.
.
.
```

Here the user enters the values for wave height and wave length as requested by the MESSAG. The values for wave height and length are stored in WAVE(1) and WAVE(2) respectively. Any occurance causing the value of REALIN to be .FALSE. causes a RETURN from the subroutine.

2. IF(INTIN(ICOUNT,ARRAY,MESSAG))STATEMENT

INTIN is the integer equivalent to REALIN. The two input routines are frequently used together, for example:

```
LOGICAL REALIN,INTIN,NOTALL
DIMENSION HEELS(20)
.
.
.
IF(.NOT.INTIN(1,NHEEL,'Enter number of heel angles:@'))RETURN
IF(.NOT.REALIN(NHEEL,HEELS,'Enter all heel angles:@')) RETURN
IF(NOTALL)RETURN
.
.
.
```

3. IF(CHARIN(ICOUNT,ARRAY,MESSAG))STATEMENT

CHARIN is similar to REALIN and INTIN. It generates a call to the user to enter character values for ARRAY. MESSAG is the prompting message, and ICOUNT is the maximum number of characters that may be entered. An example illustrates:

```
DIMENSION TEMP(5)
LOGICAL CHARIN
.
.
.
IF(.NOT.(CHARIN(20,TEMP,'ENTER FILE NAME@'))RETURN
```

Here the array TEMP is dimensioned to have 5 elements. Each element of the array stores 4 characters. If, for example, the user supplies SEQA:SHIPDATA the following storage is produced:

```
TEMP(1)=SEQA
TEMP(2)=:SHI
TEMP(3)=PDAT
TEMP(4)=A
TEMP(5)=
```

Output Routines

1. CALL MESOUT(MESSAG)

MESOUT causes printing of MESSAG. An example follows:

```
CALL MESOUT('THIS PROGRAM CALCULATES POWER@')
```

Note that MESOUT does not require a FORMAT statement as does the FORTRAN command WRITE(DEVICE,FORMAT) LIST. It therefore lacks the carriage-control capability of WRITE, making the latter preferred in some situations.

Database Routines

SPIRAL enables the user to create, modify, and use databases. A database can be an efficacious means of storing data, especially data produced by one module that is to be used by another module. The programmer is therefore expected to include options that permit acquisition of input from a database, and the sending of output to a database.

A database is a MTS line file that has been initialized by the menu item "DATABASE INITIAL" of the DEX.MAIN menu (see USER'S GUIDE). In the database editor the names of variables (e.g. BEAM,DRAFT) may be created; subsequent linkage between module and database is via these names. In consequence, it is essential that module documentation include a list of data names used, and that all modules using the same data use the identical names for the same data.

The database file name is specified by the user via the menu item "DATABASE" in DEX.MAIN module or the first call to the database routine during execution causes DEX to query the user for this information. The database file name is therefore not the responsibility of the programmer.

1. 
```
CALL RGET ('DBNAME ',RVAR)
CALL RPUT('DBNAME ',RVAR)
CALL IGET('DBNAME ',IVAR)
CALL IPUT('DBNAME ',IVAR)
```

These four routines respectively call a real variable (named DBNAME in database, RVAR in module) from the database, send a real variable to the database, call an integer variable from the database, and send an integer variable to the database.

The argument DBNAME is the name of the variable in the database. It may be up to eight characters long, using any alphanumeric characters; if less than eight characters are used, a trailing blank must be included as shown above. The argument IVAR or RVAR is used for the variable in the module. If IPUT or RPUT has a DBNAME not previously created in the database by the user, the DBNAME will be created.

2. 
```
CALL AGET('DBNAME ',NUM,ARRAY)
CALL APUT('DBNAME ',NUM,ARRAY)
```

These routines respectively call array data from the database, and send array data to it. The argument DBNAME is the name of the array in the database, and ARRAY is the name of the array in the module. NUM is the number of elements in the array (e.g. if ARRAY is dimensioned 8,5 then NUM=40). The array variable DBNAME is created in the database by APUT during execution.

IV. *Module Documentation*

Each module is to be supported by four items of explanatory literature, these being,

(1) Abstract

(2) User Manual

(3) Help file

(4) Documentation

1. ABSTRACT

Purpose is to inform a potential user of the tasks accomplished by the module, and to guide the user to further information. It is to contain the following:

(a) Name of the module

(b) Purpose of the module, i.e. the tasks performed by the module

(c) The input required. This should not cover details of input, but merely inform a potential user of the general level or type of information that must be in hand before the module can be used.

2. USER MANUAL

Purpose is to supply a user with all the information needed to run the module. It is to contain the following

(a) through (c). Same as ABSTRACT.

(d) Details of each of the module menus associated with MOD.MAIN.

  (1) INPUT

    (a) Description of input

    (b) Constraints on input

  (2) ENTER-DB

    (a) Information expected to be in the database and DBNAME for each such item.

  (3) DEFAULT

    (a) Module default values with associated menu choices.

  (4) COMPUTE

    (a) Options available in computation

(5) DRAW

(a) Options available for graphic display

(6) PRINT

(a) Description of module output with associated menu choices.

(7) SAVE

(a) Options available if different from DRAW

(8) SEND

(a) Options available if different from PRINT

(9) FILL-DB

(a) Information to be saved in the database with associated menu choices.

3. HELP FILE

Purpose is to assist user during running of module. It is to contain the following:

(a) through (c). Same as ABSTRACT.

(d) Changes, cautions, hints, from recent action of module author.

(e) Further items from user experience.

# APPENDIX A

## Description of

## Routines

# APPENDIX A

## Statements for Module Programmers

### Standard Definitions

Certain arguments occur frequently in the twenty-two DEX statements intended for module users. The description of these consistent arguments is given here.

DNAME: A *character string* used to identify the menu. The character string *must be* exactly eight characters long and *must not* include any leading or imbedded blanks. It may be passed either as a string in the call, e.g., 'LONGNAME", or it may be a double precision variable that has been initialized in a data statement. DNAME must be different from all other DNAME's in the module.

ICOUNT: An *integer* constant, variable, or arithmetic expression giving the maximum number of items to be read or examined by DEX.

ITEMS: A list of the items in the menu DNAME. Each item *may be* up to eight characters long. Leading and imbedded blanks are prohibited. The first character must be an alphabetic character. Each item's first three characters may be different from the first three characters of every other item in *this* menu. This list of items may be defined in two ways. One way is to pass a single string as a constant where each item is separated from the others by an at sign (@). The other way is to pass a DOUBLE PRECISION array. Each element of this array should be initialized to the name of a menu item.

MESSAG: Any useful string of characters. This message can have a maximum of 64 characters and must be terminated by an at sign (@). A null message may be indicated by a single @.

ITEM: Any *integer* variable name. It may be subscripted.

STRING: An array variable which has or will contain a string of characters.

SPOSN: An *integer* constant, variable, or arithmetic expression giving the position of the character in "STRING" where the operation is to begin.

SLEN: An *integer* constant, variable, or arithmetic expression giving the number of characters in "STRING" to be scanned or moved.

PLEN: An *integer* constant, variable, or arithmetic expression giving the length (number of characters) of "PATTERN".

LOGVAL: A *logical* variable name.

STATEMENT: A FORTRAN statement to transfer to if the value of the logical function is .TRUE.

DBNAME: A *character string* of from one to eight characters in length. If less than eight, a trailing blank must be included. Used in the database routines as the name of an item in the database.

RVAR: A *real* variable name.

IVAR: An *integer* variable name.

ARRAY: A *real* array.

NUM: An *integer* constant; number of items in ARRAY

NAME: MENUIN (integer function)

PURPOSE: To define and use a menu

CALLING SEQUENCE: ITEM=MENUIN(DNAME, ICOUNT,ITEMS, MESSAG)

ARGUMENTS:

DNAME: Standard definition

ICOUNT: Standard definition

ITEMS: Standard definition

MESSAG: Standard definition

ITEM: Standard definition

COMMENTS: MENUIN requests *only* command input for the accompanying menu. It returns as its value the integer place corresponding to the position in the list of items of the command obtained as input. The value returned will always be between one and the maximum number of items in the given menu inclusively. The value returned will usually be used in a FORTRAN computed GO TO statement.

EXAMPLE:

```
ILOC=MENUIN('WEIGHT  ',2,'FILE@DATABASE@',
+'WHERE IS THE WEIGHT DATA@')
```

or

```
DOUBLE PRECISION DEFMEN(3)
DATA DEFMEN/8HNAME    ,8HNUMBER  ,8HDATE    /
.
.
.
ITEM=MENUIN('DEFAULT ',3,DEFMEN,'@')
```

NAME: MENUXY (integer function)

PURPOSE: To define and use a menu *or* obtain X,Y screen coordinates

CALLING SEQUENCE: ITEM=MENUXY(X,Y,DNAME,ICOUNT,ITEMS,MESSAG)

ARGUMENTS: X,Y: The entered x,y screen coordinates will be returned in these variables *if* x,y coordinates were entered. Otherwise, they are both set to 0.

DNAME: Standard definition

ICOUNT: Standard definition

ITEMS: Standard definition

MESSAG: Standard definition

COMMENTS: This routine is very similar to MENUIN, but will accept x,y screen coordinates as well as command input. It should be used if either form of input is appropriate. If only commands or only x,y coordinates are desired, then MENUIN or XYIN should be used. If a menu command item was entered, the value returned by MENUXY will have the same meaning as the value returned by MENUIN. If x,y coordinates were entered, MENUXY will be returned as zero.

```
ITEM=MENUXY(X,Y,'BODYPLAN',2,'MOVE@DISPLAY@','@'
GO TO (100,200),ITEM
CALL POINTS(X,Y)
.
.
.
```

NAME: SETDEV

PURPOSE: To set an I/0 unit to some file or device

CALLING SEQUENCE: CALL SETDEV(UNIT,FDNAME)

ARGUMENTS: UNIT: The *integer* value of the device to be set.

FDNAME: The file or device to which the unit is to be set. This is an arbitrary length character string or an array variable containing a character string. FDNAME's character string must be terminated by at least one blank. Leading and imbedded blanks are prohibited.

COMMENTS: The purpose of this routine is to allow a module to dynamically assign a FORTRAN I/0 device to a file or device. The range of "UNIT" is from 0 to 9 inclusively.

EXAMPLE: If a data set was stored in a file named SHIP.DATA, and it was desired to read this data from device 3 this could be accomplished by:

```
    CALL SETDEV(3,'SHIP.DATA ')
    READ(3,10)x,y,z
10  FORMAT(3F10.2)
```

| | |
|---|---|
| NAME: | MODERR |
| PURPOSE: | To terminate a module in an abnormal fashion after it detects an error. |
| CALLING SEQUENCE: | CALL MODERR(MESSAG) |
| ARGUMENTS: | MESSAG: Standard definition |
| COMMENTS: | This routine provides an easy exit back to DEX after detecting an internal error in the module. The module may then be repeated from its starting point, but not CONTINUEd. After MODERR is invoked a "return" must be issued in order to procede. |
| EXAMPLE: | IF(ERROR)CALL MODERR('Divide by zero in routine CAL@') |

| | |
|---|---|
| NAME: | ENDIT |
| PURPOSE: | To terminate execution of a module running under DEX. |
| CALLING SEQUENCE: | CALL ENDIT |
| ARGUMENTS: | none |
| COMMENTS: | This routine is provided to replace the FORTRAN STOP statement. After returning to DEX with this routine, the module may then be repeated by using the DEX command BEGIN SAME. |

| | |
|---|---|
| NAME: | XYIN |
| PURPOSE: | To obtain x,y screen coordinates |
| CALLING SEQUENCE: | CALL XYIN(X,Y,MESSAG) |
| ARGUMENTS: | X,Y: The x,y screen coordinates entered |
| | MESSAG: Standard definition |
| COMMENTS: | This routine should be used only when x,y screen coordinates are acceptable to the module. |
| EXAMPLE: | CALL XYIN(X,Y,'LOCATE CURSOR ON POINT TO BE MOVED@') |

NAME: REALIN (logical function)

PURPOSE: To obtain a real type input

CALLING SEQUENCE:

```
LOGVAL=REAL(ICOUNT,ARRAY,MESSAG)
```

or

```
IF(REALIN(ICOUNT,ARRAY,MESSAG))STATEMENT
```

ARGUMENTS:

ICOUNT: Standard definition

ARRAY: A *real* array to receive the real items read.

MESSAG: Standard definition

LOGVAL: Standard definition

STATEMENT: Standard definition

COMMENTS: This routine reads an arbitrary number of real values into the array of the calling sequence. This routine, in addition to printing the supplied message, will print the message:

ENTER ICOUNT REAL NUMBERS:

At least one blank or a comma must separate the values. Otherwise, there are no restrictions on formatting the input.

REALIN is given the value .TRUE. if the read operation is completed successfully. It is .FALSE. if the read operation encounters an end-of-file or the word $ENDFILE.

EXAMPLE:

```
COMMON/HEEL/NHEEL,HEELS(20)
LOGICAL REALIN
.
.
.
IF(.NOT.REALIN(NHEEL,HEELS,'Enter all heel angles@'))RETURN

.
.
.
```

NAME: INTIN (logical function)

PURPOSE: To obtain integer input

CALLING SEQUENCE: LOGVAL=INTIN(ICOUNT,ARRAY,MESSAG)

or

IF(INTIN(ICOUNT,ARRAY,MESSAG))STATEMENT

ARGUMENTS:

ICOUNT: Standard definition

ARRAY: An *integer* array to receive the integer items read.

MESSAG: Standard definition

LOGVAL: Standard definition

STATEMENT: Standard definition

COMMENTS: This routine performs exactly like REALIN, except that it reads integers. This routine, in addition to printing the supplied message, will print the message:

ENTER ICOUNT INTEGERS:

The logical value returned follows the same rules as in REALIN.

EXAMPLE:

```
 COMMON/HEEL/NHEEL,HEELS(20)
 LOGICAL INTIN
 .
 .
 .
 IF(.NOT.INTIN(1,NHEEL,'Enter no. of heels(<20)@'))
+GO TO 150
 .
 .
 .
```

NAME: CHARIN (logical function)

PURPOSE: To obtain character string input

CALLING SEQUENCE: LOGVAL=CHARIN(LEN,STRING,MESSAG)

or

IF(CHARIN(LEN,STRING,MESSAG))STATEMENT

ARGUMENTS: LEN: An *integer* constant, variable, or arithmetic expression giving the number of characters in "STRING". The maximum value allowed is 72. LEN is less than or equal to the number of bytes in the STRING.

STRING: Standard definition

MESSAG: Standard definition

LOGVAL: Standard definition

STATEMENT: Standard definition

COMMENTS: This is very similar to REALIN and INTIN, except that it reads string data. The supplied message, as well as the following message, will be printed;

ENTER A CHARACTER STRING:

The logical value returned follows the same rules as in REALIN.

EXAMPLE:

```
INTEGER FILE(8)
LOGICAL CHARIN
.
.
.
IF(.NOT.CHARIN(32,FILE,'ENTER PLOT FILENAME@'))RETURN

.
.
.
```

NAME: GRAF

PURPOSE: This routine sets up the axes, grid, etc. for the IPLOT graph in the IDISP display.

CALLING SEQUENCE: GRAF(IDISP,IPLOT,XL1,XL2,YL1,YL2,XP1,XP2,YP1,YP2, IOWNX,IOWNY,NTICKX,NTICKY,NGRIDX,NGRIDY,NDASH,IAXCOL, LABELX,LENX,IXCOL,LABELY,LENY,IYCOL,ITITLE,LENT,ITCOL)

ARGUMENTS:

1. IDISP - The number of the display in which the graph is to be placed. (1 to 10)

2. IPLOT - The number of the graph in the IDISP display that this call is describing. (1 to 10)

3. XL1, XL2, YL1, YL2 - Logical extremes of the axes of the graph.
   1 implies minimum
   2 implies maximum

4. XP1, XP2, YP1, YP2 - Physical location of the graph in the display if the display has a lower left corner of (-1,-1) and an upper right corner of (1,1)
   Note: for Tektronix screens, the bounds of the display are (-1,-.75) and (1,+.75). Therefore for a display to cover the entire Tektronix screen the following physical coordinates should be given.
   XP1=-1.0 YP1=-.75
   XP2=+1.0 YP2=+.75

5. IOWNX - If the value is 1, then the routine will
   IOWNY - draw that axis using the logical maximum specified. If the value is anything else, the routine will trace these values and find its own "nice" axis extremes and grid interval.

6. NTICKX, NTICKY - This is the number of "tick mark" intervals on the axis.
   If the corresponding IOWNX or IOWNY is 1, then the axis will have exactly that many intervals. Otherwise it will be approximate.

7. NGRIDX, NGRIDY - Specifies how many grid lines to be drawn for each axis.
   - 0 - None (Just X & Y axes)
   - -1 - Just outside
   - +1 - Every tick mark
   - +2 - Every 1/2 tick mark
   - +4 - Every 1/4 tick mark
   - +5 - Every 1/5 tick mark
   - +10 - Every 1/10 tick mark

8. NDASH - If this is 1, then the grid lines will be dashed lines. Otherwise they will be solid lines.

9. IAXCOL - This specifies the color of the axis, grid, tick marks and tick mark labels.

   1 - black
   2 - red
   3 - blue

   The default value (i.e. if the value is other than 1,2, or 3) is 1(black)

10. LABELX - The label for the x-axis. The maximum number of characters allowed is 50. If more is specified, it is truncated to 50. For the greatest ease, put the label right in the calling argument list enclosed in primes. (e.g.: ...IAXCOL, 'THIS IS THE X-LABEL',21,...)

11. LENX - This is the length of LABELX in characters.

12. IXCOL - Color of LABELX (same values as IAXCOL)

13. LABELY - Same as above for the Y-axis, with 40 characters maximum.

14. LENY - Length of LABELY in characters

15. IYCOL - Color of LABELY (same values as IAXCOL)

16. ITITLE - Title for the graph, 30 characters maximum. 1 1/2 times the size of LABELX and LABELY and placed below LABELX. Like the axis labels, this is also centered.

17. LENT - Length of the title in characters.

18. ITCOL - Color of ITITLE (same values as IAXCOL)

   Note: For the title or either of the two labels, if its length parameter is $\leq 0$ then no label or title will be drawn.

COMMENTS: This routine is called prior to GLINE and DSPLAY to establish the graph.

EXAMPLE:

```
CALL GRAF(1,1,VMIN,VMAX,0.,PEMAX,-1,1.,-.75,.70,0,0,
+         5,5,2,2,0,0,'VELOCITY (KTS)',20,2
+         'EFFECTIVE HORSEPOWER',20,2
+         'EHP vs. VK',10,3
```

NAME: GLINE

PURPOSE: This routine puts out a set of X,Y data on the IPLOT graph in the IDISP display. How the data is displayed depends on the MODE chosen.

CALLING SEQUENCE: GLINE(IDISP,IPLOT,ILINE,XIN,YIN,NPTS,MODE,IRTYPE,NORDER, LDASH,IRDASH,ICOL,ICTYPE,ICHAR)

ARGUMENTS:

1. IDISP - The display in which the data is to be drawn.
2. IPLOT - The graph in which the data is to be placed. (Should have been set up already by a call to GRAF)
3. ILINE - The number of the data set to be put into IPLOT in IDISP.
4. XIN YIN - The X,Y data set to be put into IPLOT.
5. NPTS - The number of X,Y data points in XIN & YIN. (200 points maximum)
6. MODE - A code describing how the data is to be displayed.

   1 - Just draw the points
   2 - Just a straight line between points
   3 - Just interpolate a curve between points
   4 - Points and line (1&2)
   5 - Points and curve (1&3)
   6 - Regressed line only
   7 - Points and regressed line (1&6)
   8 - Points, line and regressed line (1&2&6)

   An error message is written if any other mode value is given.
7. IRTYPE - If MODE = 6,7 or 8, then IRTYPE is used to determine which type of gregressed line to put through the data.

   1 - $N^{th}$ order polynomial least squares curvefit
   2 - Exponential curvefit of the form: Y=AO*X**Al
   3 - Power curvefit of the form: Y=AO*X**Al
   4 - Logrithmic curvefit of the form: Y=AO+Al*ALOG(X)
8. NORDER - If IRTYPE = 1, then NORDER is used to determine the order of the polynomial fit. NORDER must be less than NPTS and also (arbitrarily) less than or equal to 10.
9. LDASH - Type of line to be drawn (if one is to be drawn) for either line or curve.

1 - Solid line ————————
2 - Short dashed line --------
3 - Long-short-long line —— - —— -

10. IRDASH - Same as above except this specifies the line type for the regressed line. For either line type specification, 1 is used if the value is neither 2 or 3.

11. ICOL - This specifies the color with which the data is to be drawn. (Same values as IAXCOL)

12. ICTYPE - If points are to be drawn, then ICTYPE is used to determine which point type is to be used.

1 - put out an alphanumeric character
2 - "Right-side up" triangle Δ
3 - "Upside-down" triangle ∇
4 - Box □
5 - Diamond ◇

13. ICHAR - This is the character to be used if ICTYPE=1. Typically, one would put the character in the calling argument in primes. e.g.: ...ICOL,1,'X')

COMMENTS: GLINE is used to plot data on the display and graph specified.

EXAMPLE: Call GLINE (1,1,VK,PE,N,7,3,NORDER,IDASH,3,1,3,ICHAR,2)

NAME: DSPLAY

PURPOSE: This routine controls the deletion, display and saving of the previously created displays.

CALLING SEQUENCE: DSPLAY(IGO,IDISP,IPLOT,ILINE,SIZE)

ARGUMENTS:

1. IGO - This is a code which specifies what to do with the display(s) generated by the previous calls to GRAF & GLINE.
   - 1 - Delete ILINE in IPLOT in IDISP
   - 2 - Delete IPLOT in IDISP
   - 3 - Delete IDISP
   - 4 - Draw IDISP on the terminal
   - 5 - Put IDISP in a file for Calcomp plot
   - 6 - Put IDISP in a file as an object
   - 7 - Put all IDISP in a file for Calcomp plot
   - 8 - Put all IDISP in a file as objects
   - 9 - Delete all IDISP
2. IDISP - Which display in question
3. IPLOT - Which plot in question
4. ILINE - Which line in question
5. SIZE - Size of the Calcomp plot. (i.e. the width of the display in inches)

COMMENTS: IDISP,IPLOT,ILINE and SIZE need only be given values for the IGO type requiring that information.

The routine DSPLAY does not pause after drawing the display on the terminal. Control is returned immediately to the calling routine. The programmer should put in some sort of pause or stop before another call to DSPLAY. In that way the user is given a chance to view the display before it is erased.

EXAMPLE: Call DSPLAY(4,1,IPLOT,ILINE,SCALE)

NAME: MESOUT

PURPOSE: To print a message

CALLING SEQUENCE: CALL MESOUT(MESSAG)

ARGUMENTS: MESSAG: Standard definition

COMMENTS: This routine prints the message in a controlled manner with some screen management. It should be used instead of FORTRAN WRITE statements for simple messages. No explicit carriage control possible.

EXAMPLE:

```
.
.
.
150 continue

Call MESOUT('more than 20 heels are not allowed @')
.
.
.
```

| | |
|---|---|
| NAME: | LFANYC (integer function) |
| PURPOSE: | To search a character string for the first occurrence of any character specified in a specific pattern. |
| CALLING SEQUENCE: | LOC=LFANYC(STRING,SPOSN,SLEN,PATTERN,PLEN) |
| ARGUMENTS: | 1. STRING - Standard definition |
| | 2. SPOSN - Standard definition |
| | 3. SLEN - Standard definition |
| | 4. PATTERN - An array variable containing a character string, or a Hollerith literal, indicating the specific characters to be sought. |
| | 5. PLEN - Standard defintion |
| | 6. LOC - An integer variable name. "LOC" will be set to the position in "STRING" at which the first character indentical to any in "PATTERN" is found. If there is no character common to both "STRING" and "PATTERN", "LOC" will be set equal to zero. "LOC" may be subscripted. |
| COMMENTS: | LFANYC is the complement of LFNOTC. |

| | |
|---|---|
| NAME: | LMATCH ( integer function) |
| PURPOSE: | To search a character string for the first occurrence of a specific pattern. |
| CALLING SEQUENCE: | LOC=LMATCH(STRING,SPOSN,SLEN,PATTERN,PLEN) |
| ARGUMENTS: | 1. STRING - Standard definition |
| | 2. SPOSN - Standard definition |
| | 3. SLEN - Standard definition |
| | 4. PATTERN - An array variable containing a character string, or a Hollerith literal, indicating the specific pattern sought. |
| | 5. PLEN - Standard definition |
| | 6. LOC - An integer variable name. "LOC" will be set to the position in "STRING" at which "PATTERN" starts. If "PATTERN" is not found, "LOC" is set to zero. "LOC" may be subscripted. |

NAME: LMOVEC

PURPOSE: To copy a character string from one array to another.

CALLING SEQUENCE: CALL LMOVEC(STRING,SPOSN,SLEN,TO,TOPOSN)

ARGUMENTS:

1. STRING - Standard definition
2. SPOSN - Standard definition
3. SLEN - Standard definition
4. TO - The array variable which is to receive the characters copied from "STRING". "TO" and "STRING" may be the same variables.
5. TOPOSN - An <u>integer</u> constant, variable, or arithmetic expression giving the position of the character in "TO" at which insertion of "STRING's" character string is to being.

NAME: LEQLCH (logical function)

PURPOSE: To determine if two characters are identical

CALLING SEQUENCE: LOGVAL=LEQLCH(FIRST,SECOND)

or

IF(LEQLCH(FIRST,SECOND))STATEMENT

ARGUMENTS:

1. FIRST,SECOND - Array variables of length one, single array elements, or single character Hollerith literals.
2. LOGVAL - Standard definition
3. STATEMENT - Standard definition

COMMENTS: If "FIRST" is identical to "SECOND", then "LEQLCH" will be .TRUE.; otherwise, it will be .FALSE.

NAME: LNRCVT (logical function)

PURPOSE: To convert a string of characters into a numeric value.

CALLING SEQUENCE: LOGVAL=LNRCVT(VALUE,MODE,SLEN,STRING,SPOSN)) STATEMENT

ARGUMENTS:

1. VALUE - A single variable or element of an array which is to be assigned the numeric value resulting from the conversion of "STRING".
2. MODE - An array variable or Hollerith literal giving the mode of value. "MODE" must be:

   'I' for integer
   'F' for floating point (real),
   'E' for scientific notation, or
   'D' for double precision
3. SLEN - Standard definition
4. STRING - Standard definition
5. SPOSN - Standard definition
6. LOGVAL - Standard definition
7. STATEMENT - Standard definition

COMMENTS: If conversion is successful then "LNRCVT" will be .TRUE.; otherwise it will be .FALSE.

| | |
|---|---|
| NAME: | IGET |
| PURPOSE: | To obtain an integer value from a database and assign it to an integer variable. |
| CALLING SEQUENCE: | CALL IGET(DBNAME,IVAR) or<br>IF(IGET(DBNAME,IVAR))STATEMENT |
| ARGUMENTS: | 1. DBNAME - Standard definition<br>2. IVAR - Standard definition |
| COMMENTS | None |
| EXAMPLES | CALL IGET('CREWNUM',NCREW)<br>CALL IGET('CREW', N(3)) |

| | |
|---|---|
| NAME: | IPUT (logical function) |
| PURPOSE: | To place the value of an integer variable in a database. |
| CALLING SEQUENCE: | CALL IPUT(DBNAME,IVAR) or<br>IF(IPUT(DBNAME,IVAR))STATEMENT |
| ARGUMENTS: | 1. DBNAME - Standard definition<br>2. IVAR - Standard definition |
| COMMENTS: | None |
| EXAMPLES: | CALL IPUT('CREWNUM',NCREW)<br>CALL IPUT('CREW',N(3)) |

| | |
|---|---|
| **NAME:** | RGET (logical function) |
| **PURPOSE:** | To obtain a real value from a database and assign it to a real variable. |
| **CALLING SEQUENCE:** | CALL RGET(DBNAME,RVAR) or<br>IF(RGET(DBNAME,RVAR))STATEMENT |
| **ARGUMENTS:** | 1. DBNAME - Standard definition<br>2. RVAR - Standard definition |
| **COMMENTS:** | None |
| **EXAMPLES:** | CALL RGET('DISPLACE',X(4))<br>CALL RGET('BEAM',BEAM) |

| | |
|---|---|
| NAME: | RPUT (logical function) |
| PURPOSE: | To place the value of a real variable in a database |
| CALLING SEQUENCE: | CALL RPUT(DBNAME,RVAR) or<br>IF(RPUT(DBNAME,RVAR))STATEMENT |
| ARGUMENTS: | 1. DBNAME - Standard Definition<br>2. RVAR - Standard Definition |
| COMMMENTS: | None |
| EXAMPLES: | CALL RPUT('DISPLACE',X(4))<br>CALL RPUT('BEAM',BEAM) |

**NAME:** AGET (logical function)

**PURPOSE:** To obtain data from a database and assign it to the elements of an array.

**CALLING SEQUENCE:** CALL AGET(DBNAME,NUM,ARRAY) or
IF(AGET(DBNAME,NUM,ARRAY))STATEMENT

**ARGUMENTS:**

1. DBNAME - Standard definition
2. ARRAY - Standard definition
3. NUM - Standard definition

**COMMENTS:** All arrays in a database are saved as real arrays. Integer arrays in a module may also be saved in a database as real arrays allowing DEX to perform the conversion.

**EXAMPLE:**

```
REAL LINES(11,8)
INTEGER NPTS(11)
.
.
.
CALL AGET('LINES',88,LINES)
CALL AGET('NPTS',11,NPTS)
```

| | |
|---|---|
| NAME: | APUT (logical function) |
| PURPOSE: | To place values of an array in a database. |
| CALLING SEQUENCE: | CALL APUT(DBNAME,NUM,ARRAY) or<br>IF(ABUT(DBNAME,NUM,ARRAY))STATEMENT |
| ARGUMENTS | 1. DBNAME - Standard definition<br>2. ARRAY - Standard definition<br>3. NUM - Standard definition |
| COMMENTS: | See comments on AGET |
| EXAMPLE: | REAL LINES(11,8)<br>INTEGER NPTS(11)<br>.<br>.<br>.<br>CALL APUT('LINES',88,LINES)<br>CALL APUT('NPTS',11,NPTS) |

APPENDIX B

# MODEL-SHIP RESISTANCE EXTRAPOLATION MODULE

Library Name: RESIST

## ABSTRACT

The module performs the model to ship resistance extrapolation calculations. Froude's method is used to extrapolate the model data. Either the ATTC or ITTC model-ship extrapolator line may be used to calculate the frictional resistance component. Two plots are available within the module, 'Effective Horsepower vs Velocity' and 'Resistance vs Velocity'. The user is expected to supply the model data and scale ratio for the ship.

## Details of Module Operation

The user is expected to enter either all the information under the command ENTER-DB or to enter all the information under the command INPUT-REQUIRED prior to initialing the commands COMPUTE, PRINT, DRAW, etc.

(1) menu INPUT

The inputs of the module consist of:

(a) CORR-ALL- Correlation allowance, a value of 0.0004 is assumed if not specified during input.

(b) TANK- The water conditions in the towing tank, The water density (slugs/$ft^3$) and kinematic viscosity ($ft^2$/sec) are requested.

(c) MODEL- The model's length waterline (LWL) in feet and the model's wetted surface in square feet.

(d) LAMBDA- Linear scale ratio between the model and the ship. If a value of 0 is entered a request for the ship's LWL will be made.

(e) RES-DATA- The location of the resistance data. The user should have the data in either a database or a file. The format for the file is;

| Line # | Items(s) | Format |
|---|---|---|
| 1 | NRUNS | I5 |
| 2 | VM(1),RTM(1) | 2F15.5 |
| ⋮ | ⋮ | ⋮ |
| NRUNS+1 | VM(NRUNS),RTM(NRUNS) | 2F15.5 |

where

NRUNS = number of runs

VM = model velocity (ft/sec)

RTM = model resistance (lbs)

(f) REQUIRED- items (b) through (e) above

(g) ALL- items (a) through (e) above

(2) ENTER-DB

The following data will be retrieved from the database via the ENTER-DB command.

XLM - model's length waterline (ft)

SURFM - model's wetted surface ($ft^2$)

RHOTK - tank water density ($slugs/ft^3$)

GNUTK - tank water kinenatic viscosity ($ft^2/sec$)

N - number of runs (up to 30)

VM - model velocity (ft/sec), up to 30 values

RTM - model resistance (lbs), up to 30 values

(3) Menu DEFAULT

The defaults of the module consist of;

(a) MODEL - prompts the user for the model I.D. (name and number)

(b) SHIP - prompts the user for the ship I.D. (name and number)

(c) DATE - prompts the user for the date

(d) OCEAN - allows the user to change from the standard ocean condition of,

| | |
|---|---|
| water density | 1.9905($slugs/ft^3$) |
| kinenatic viscosity | $1.27908 \times 10^{-5}$($ft^2/sec$) |

(e) LINE - allows the user to establish which extrapolator line is to be used. The default is the ITTC line.

(4) COMPUTE

No options are available under the COMPUTE command.

(5) menu DRAW

Two plots are available to the user through the DRAW or SAVE commands.

(a) POWER - a plot of Effective Horsepower vs Velocity for the ship is produced.

(b) RESIST - a plot of the Resistance vs Velocity for the ship is produced. The resistance is broken down into the frictional and residual components.

(c) ALL - both plots are produced.

If the SAVE command is used a request for the plot file-name and plot size will be made.

(6) menu PRINT

The PRINT and SEND commands have five options associated with them.

(a) INPUT - the input data will be printed for verification.

(b) MODEL - the model data and results will be printed.

(c) SHIP - the ship results will printed.

(d) RESULTS - prints the model data and results plus the ship results, similar to (b) and (c) above.

(e) ALL - prints (a) through (c) above.

If the SEND command is used a prompt for the file or device where the output is to be sent will be made.

(7) FILL-DB

The command FILL-DB saves the following information in the database

| DBNAME | Size | Description |
|---|---|---|
| LWL | 1 | length waterline for ship |
| SWET | 1 | wetted surface of ship |
| N | 1 | number of runs |
| VS | 30 | velocity in feet/sec |
| VK | 30 | velocity in knots |
| CFS | 30 | Frictional Resistance Coefficient |
| CR | 30 | Residual Resistance Coefficient |
| RTS | 30 | Total Resistance of ship |
| PE | 30 | Effective Horsepower of ship |

METHODS

Froude's method for extrapolating model data is used to compute the resistance and powering of the ship. The resistance of the model is broken down into two components, frictional and residual. The residual component is assumed to be the same for both the model and the ship. The frictional component is assumed to be a function of the Reynold's number and an extrapolator line such as the ATTC or ITTC line is used to compute it. A correlation allowance is used to improve the agreement between the model tests and the full scale results.

REF: P.N.A. Chapter VIII

```
#RU SEQA:SPIRAL T=5
#EXECUTION BEGINS



 N. A. & M. E.
 Educational System

 USER: SEQG

 TODAY IS: JUN 30, 1978    TIME: 14:32:17


 **** RECENT NEWS on LINESGEN - see Brad or Steve for details *****
&
&Enter item from menu: DEX.MAIN
&R TEST NR.O+SEQA:CRAFPAX.O+*IG




?***  MODEL-SHIP EXTRAPOLATION MODULE   ***
?
?Enter item from menu: MOD.MAIN
?INPUT ALL
?ENTER CORRELATION ALLOWANCE:
?.0004
?ENTER TANK WATER DENSITY AND KIN. VISC.:
?Enter a total of     2 real number(s)
?1.9905 1.2705E-05
?ENTER LWL AND WETTED SURFACE OF THE MODEL:
?Enter a total of     2 real number(s)
?15.095 42.0
?ENTER LINEAR SCALE RATIO, LAMBDA:
?34.667
?
?Enter item from menu: WHERE
?WHERE IS THE RESISTANCE DATA:
?FILE
?ENTER FILE NAME:
?REDAT
?*** DATA READ CORRECTLY ***
?
?Enter item from menu: MOD.MAIN
?COMPUTE
?
?
?*** CALCULATIONS COMPLETED ***
?
?
?
?Enter item from menu: MOD.MAIN
?PRINT ALL
```

INPUT VALUES FOR MODEL AND SHIP      DATE:

| MODEL: | | SHIP: | |
|---|---|---|---|
| NUMBER: | 0 | NUMBER: | 0 |
| LWL : | 15.095 | LWL : | 523.298 |
| WETTED SURFACE: | 42.000 | WETTED SURFACE: | 50475.652 |
| SCALE RATIO : | 34.667 | | |

WATER PROPERTIES

| | | | |
|---|---|---|---|
| DENSITY : | 1.99050 | DENSITY : | 1.99050 |
| KIN. VIS.: | 0.12765E-04 | KIN. VIS.: | 0.12791E-04 |

MODEL RESULTS

| VM | CFM | CR | CTM | RTM |
|---|---|---|---|---|
| 2.410 | 0.37757E-02 | 0.17845E-03 | 0.39542E-02 | 0.96 |
| 2.860 | 0.36528E-02 | 0.17857E-03 | 0.38314E-02 | 1.31 |
| 3.300 | 0.35547E-02 | 0.33368E-03 | 0.38883E-02 | 1.77 |
| 3.790 | 0.34634E-02 | 0.46714E-03 | 0.39305E-02 | 2.36 |
| 4.320 | 0.33803E-02 | 0.41409E-03 | 0.37944E-02 | 2.96 |
| 4.840 | 0.33106E-02 | 0.43739E-03 | 0.37480E-02 | 3.67 |
| 5.230 | 0.32642E-02 | 0.46161E-03 | 0.37258E-02 | 4.26 |
| 5.300 | 0.32564E-02 | 0.51648E-03 | 0.37729E-02 | 4.43 |
| 5.540 | 0.32304E-02 | 0.51881E-03 | 0.37492E-02 | 4.81 |
| 5.750 | 0.32089E-02 | 0.66948E-03 | 0.38784E-02 | 5.36 |
| 6.060 | 0.31788E-02 | 0.89918E-03 | 0.40780E-02 | 6.26 |
| 6.260 | 0.31604E-02 | 0.13083E-02 | 0.44687E-02 | 7.32 |
| 6.420 | 0.31463E-02 | 0.14275E-02 | 0.45738E-02 | 7.88 |

? SHIP RESULTS PART 1

?

| ? | VS | CFS | CR | CA | CTS |
|---|---|---|---|---|---|
| ? | 14.190 | 0.16394E-02 | 0.17845E-03 | 0.00040 | 0.22178E-02 |
| ? | 16.839 | 0.16039E-02 | 0.17857E-03 | 0.00040 | 0.21825E-02 |
| ? | 19.430 | 0.15752E-02 | 0.33368E-03 | 0.00040 | 0.23088E-02 |
| ? | 22.315 | 0.15481E-02 | 0.46714E-03 | 0.00040 | 0.24152E-02 |
| ? | 25.436 | 0.15231E-02 | 0.41409E-03 | 0.00040 | 0.23372E-02 |
| ? | 28.497 | 0.15019E-02 | 0.43739E-03 | 0.00040 | 0.23393E-02 |
| ? | 30.794 | 0.14877E-02 | 0.46161E-03 | 0.00040 | 0.23493E-02 |
| ? | 31.206 | 0.14853E-02 | 0.51648E-03 | 0.00040 | 0.24017E-02 |
| ? | 32.619 | 0.14772E-02 | 0.51881E-03 | 0.00040 | 0.23961E-02 |
| ? | 33.955 | 0.14706E-02 | 0.66948E-03 | 0.00040 | 0.25400E-02 |
| ? | 35.680 | 0.14612E-02 | 0.89918E-03 | 0.00040 | 0.276[illegible]4E-02 |
| ? | 36.858 | 0.14555E-02 | 0.13083E-02 | 0.00040 | 0.31637E-02 |
| ? | 37.800 | 0.14511E-02 | 0.14275E-02 | 0.00040 | 0.32786E-02 |

? SHIP RESULTS PART 2

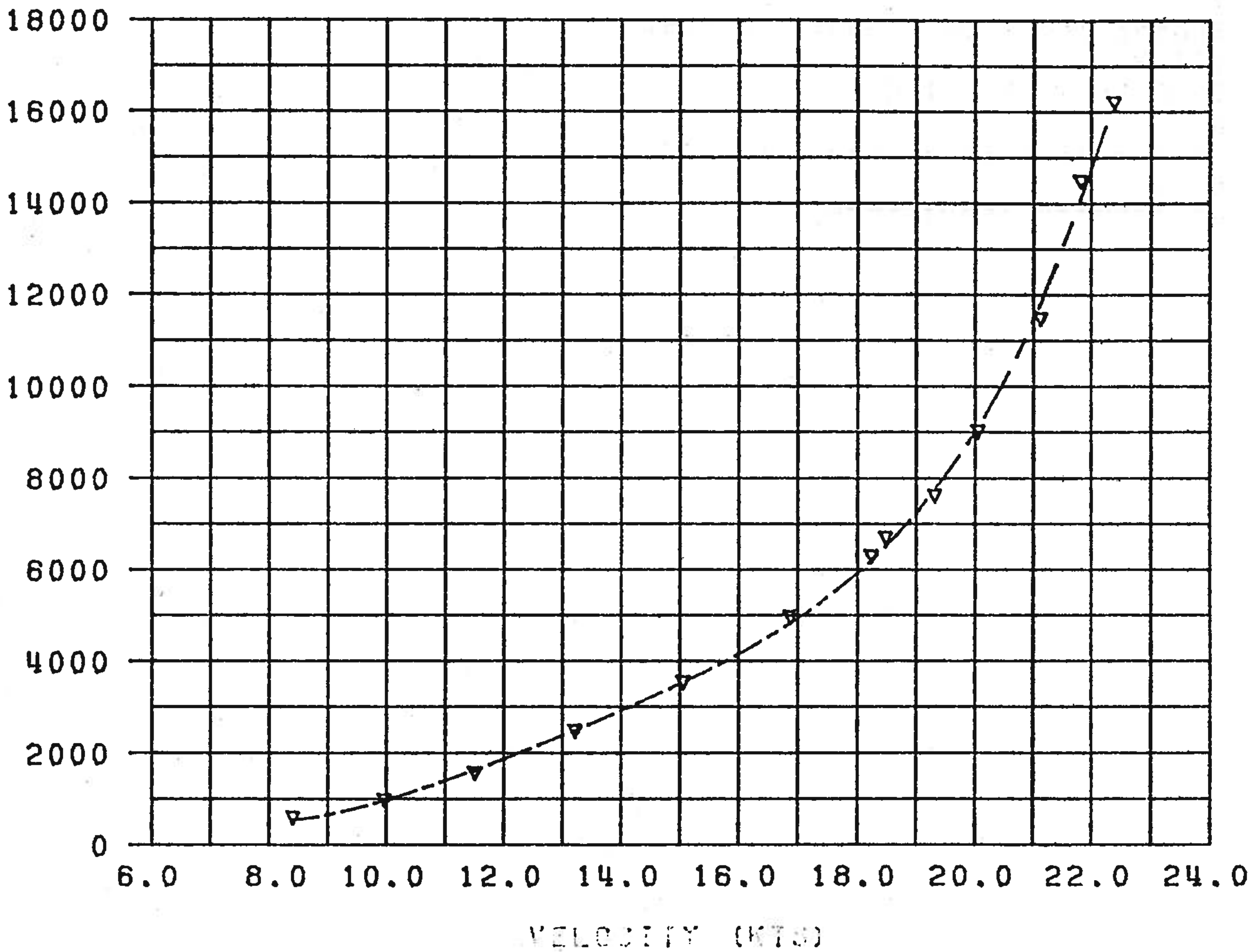
?

| ? | VS | RTS | EHP | VK |
|---|---|---|---|---|
| ? | 14.190 | 0.22433E+05 | 578.76563 | 8.402 |
| ? | 16.839 | 0.31089E+05 | 951.86060 | 9.971 |
| ? | 19.430 | 0.43787E+05 | 1546.88867 | 11.505 |
| ? | 22.315 | 0.60417E+05 | 24[illegible]1.29077 | 13.213 |
| ? | 25.436 | 0.75960E+05 | 3512.90527 | 15.060 |
| ? | 28.497 | 0.95433E+05 | 4944.70703 | 16.873 |
| ? | 30.794 | 0.11191E+06 | 6265.64063 | 18.233 |
| ? | 31.206 | 0.11749E+06 | 6666.18359 | 18.477 |
| ? | 32.619 | 0.12807E+06 | 7595.41797 | 19.314 |
| ? | 33.855 | 0.14625E+06 | 9002.64453 | 20.046 |
| ? | 35.680 | 0.17654E+06 | 11452.91016 | 21.126 |
| ? | 36.858 | 0.21591E+06 | 14469.35547 | 21.824 |
| ? | 37.800 | 0.23533E+06 | 16173.82031 | 22.381 |

? FRICTIONAL COEFFICIENTS FROM THE ITTC LINE

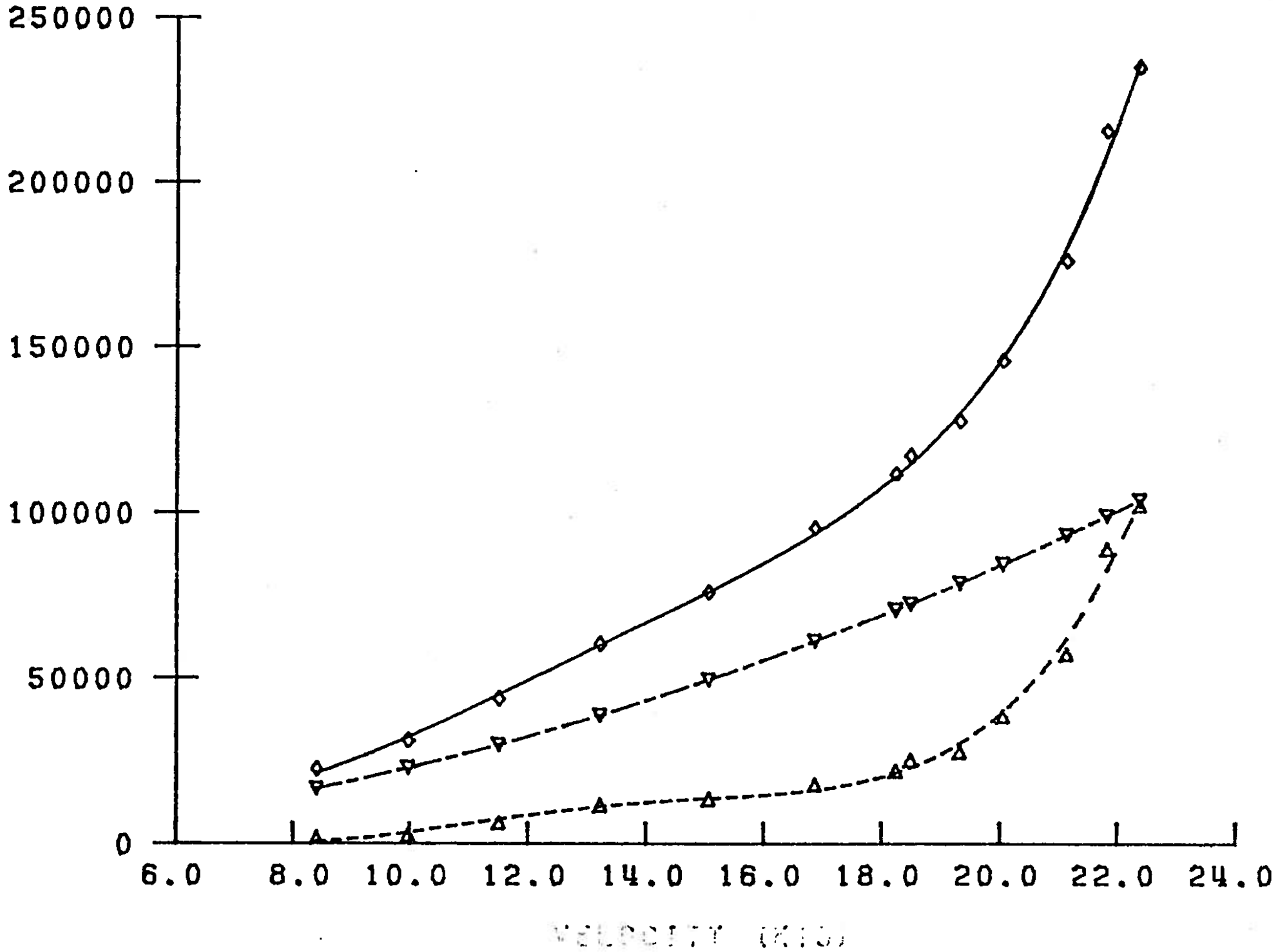
```
?
?Enter item from menu: MOD.MAIN
?SAVE
?
?Enter item from menu: DRAW
?ALL
?ENTER PLOT SIZE (INCHES):
?7.5
?ENTER PLOT FILENAME:
?-PLOT


?PDS: PLOT DESCRIPTION GENERATION BEGINS.
?
?Enter item from menu: MOD.MAIN
?STOP
?MODULE TERMINATION
&
&Enter item from menu: DEX.MAIN
&QUIT
#EXECUTION TERMINATED
#
```



18000
16000
14000
12000
10000
8000
6000
4000
2000
0
6.0
8.0
10.0
12.0
14.0
16.0
18.0
20.0
22.0
24.0
VELOCITY (KTS)


EHP VS. VK




RESISTANCE VS. VELOCITY

```
C       .............. MAIN PROGRAM................
        DOUBLE PRECISION MASITM(10)
        DATA MASITM/8HENTER-DB,8HINPUT   ,8HDEFAULT ,8HCOMPUTE ,
     1            8HFILL-DB ,8HDRAW    ,8HPRINT   ,8HSEND    ,
     2            8HSAVE    ,8HSTOP    /
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C   ENTER-DB - ENTERS THE PROGRAM INPUT FROM A DATABASE
C   INPUT    - INPUTTING OF THE MAIN PROGRAM VARIABLES
C   DEFAULT  - ENABLES THE CHANGING OF VALUES OF VARIABLES THAT
C              ARE GIVEN DEFAULT VALUES DURING LOADING
C   COMPUTE  - COMPUTATIONS IN THE MODULE ARE DONE
C   FILL-DB  - FILLS THE DATABASE WITH THE PROGRAMS INPUT AND OUTPUT
C   DRAW     -DISPLAYS GRAPHIC OUTPUT AT THE TERMINAL
C   PRINT    - PRINTS MODULE INPUT AND OUTPUT AT THE TERMINAL
C   SEND     - SENDS THE INPUT AND OUTPUT TO DEVICE SUCH AS
C              *PRINT*, A TEMPORARY FILE, OR A PERMANENT FILE
C   SAVE     -SAVES GRAPHIC OUTPUT IN A FILE FOR LATER USE
C   STOP     - RETURN TO DEX WHEN FINISHED WITH THE MODULE
C
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
        CALL INIT
        WRITE(6,100)
  100   FORMAT('1***  MODEL-SHIP EXTRAPOLATION MODULE  ***')
  500   CONTINUE
        ITEM=MENUIN('MOD.MAIN',10,MASITM,'@')
        GO TO (1000,2000,3000,4000,5000,6000,7000,8000,9000,10000),ITEM
        CALL MCDERR('error in MOD.MAIN@')
 1000   CONTINUE
C       enter the database
        CALL DBASE
        GO TO 500
 2000   CONTINUE
C       call to take input
        CALL INPUT
        GO TO 500
 3000   CONTINUE
C       call to set default variables
        CALL DEFLT
        GO TO 500
 4000   CONTINUE
C       call for computation
        CALL CALC
        GO TO 500
 5000   CONTINUE
C       call to store variables in the database
        CALL DBFILL
        GO TO 500
 6000   CONTINUE
C       call to draw at the terminal
        CALL DRAW
        GO TO 500
 7000   CONTINUE
C       call to print at the terminal
        CALL PRINT
        GO TO 500
 8000   CONTINUE
```

```
C     call to route cutput
      CALL DEVICE
      GO TO 500
 9000 CONTINUE
C     call tc save grapnics
      CALL SAVE
      GO TO 500
10000 CONTINUE
C     return control to DEX
      CALL ENDIT
      END
```

```
      SUBROUTINE INIT
C
C     Initialize program variables and set default
C     constants.
C
      INTEGER MNAME(5),SHIP(5),SNUM,DATE(2)
      REAL LAMBDA
      LOGICAL TEST(5)
      COMMON /TESTS/TEST
      COMMON /CONDTN/RHOTK,RHOSW,GNUTK,GNUSW,CA,LINE,LAMBDA
      COMMON /MODEL/XLM,SURFM,N,VM(30),RTM(30),CR(30),CFM(30),CTM(30)
      COMMON /SHIPS/XLS,SURFS,VS(30),VK(30),CFS(30),RTS(30),PE(30),
     +             CTS(30),RNS(30),RFS(30),RRS(30)
      COMMON /TITLES/MNAME,MNUM,SHIP,SNUM,DATE
C
      INTEGER BLANK
      DATA BLANK/4H    /
C
C...  TEST(1)=resistance data (file)
C...  TEST(2)=plot test to delete plots
C...  TEST(3)=resistance data (database)
C...  TEST(4)=check for all data before computing
C...  TEST(5)=DRAW & PRINT check
C
      CALL IGINIT
      DO 10 I=1,5
         TEST(I)=.FALSE.
         MNAME(I)=BLANK
         SHIP(I) =BLANK
   10 CONTINUE
         RHOTK=-10.
         GNUTK=-10.
         XLM=-10.
         XLS=-10.
         SURFM=-10.
         SURFS=-10.
         SNUM=0
         MNUM=0
         DATE(1)=BLANK
         DATE(2)=BLANK
         RHOSW=1.9905
         GNUSW=1.27908E-05
         CA   =0.0004
         LINE=1
      RETURN
      END
```

```
      SUBROUTINE DBASE
C
C     Call to enter database values into program variables
C
      INTEGER MNAME(5),SHIP(5),SNUM,DATE(2)
      REAL LAMBDA
      LOGICAL TEST(5)
      COMMON /TESTS/TEST
      COMMON /CONDTN/RHOTK,RHOSW,GNUTK,GNUSW,CA,LINE,LAMBDA
      COMMON /MODEL/XLM,SURFM,N,VM(30),RTM(30),CR(30),CFM(30),CTM(30)
      COMMON /SHIPS/XLS,SURFS,VS(30),VK(30),CFS(30),RTS(30),PE(30),
     +              CTS(30),RNS(30),RFS(30),RRS(30)
      COMMON /TITLES/MNAME,MNUM,SHIP,SNUM,DATE
C
C...  get values from the database
C
         CALL RGET('XLM ',XLM)
         CALL RGET('SURFM ',SURFM)
         CALL RGET('RHOTK ',RHOTK)
         CALL RGET('GNUTK ',GNUTK)
C
         CALL IGET('N ',N)
C
         CALL AGET('VM ',30,VM)
         CALL AGET('RTM ',30,RTM)
C
C...  set TEST(3) to .TRUE. to show that db was entered
C
      TEST(3)=.TRUE.
         WRITE(6,00001)
00001 FORMAT('*** DATABASE ENTERED ***')
      RETURN
      END
```

```
      SUBROUTINE INPUT
C
C     Data aquisition routine.  This routine interfaces
C     with the user to accept input.
C
      INTEGER MNAME(5),SHIP(5),SNUM,DATE(2)
      REAL LAMBDA,VIN(30),RIN(30)
      LOGICAL TEST(5)
      COMMON /TESTS/TEST
      COMMON /CONDTN/RHOTK,RHOSW,GNUTK,GNUSW,CA,LINE,LAMBDA
      COMMON /MODEL/XLM,SURFM,N,VM(30),RTM(30),CR(30),CFM(30),CTM(30)
      COMMON /SHIPS/XLS,SURFS,VS(30),VK(30),CFS(30),RTS(30),PE(30),
     +                CTS(30),RNS(30),RFS(30),RRS(30)
      COMMON /TITLES/MNAME,MNUM,SHIP,SNUM,DATE
C
C...  establish input menu choises
C
      DOUBLE PRECISION INMEN(7)
      DATA INMEN/8HCORR-ALL,8HTANK    ,8HMODEL   ,8HLAMBDA  ,
     +           8HRES-DATA,8HREQUIRED,8HALL     /
      REAL VEC(2)
      INTEGER FILE(5)
      LOGICAL INTIN,REALIN,CHARIN,NOTALL
C
   10 CONTINUE
C...  input menu
      ITEM=MENUIN('INPUT   ',7,INMEN,'@')
C...  set switch according to item choisen
C...  NOTALL=.TRUE. causes a return after the specified input
      NOTALL=ITEM .LT. 5
C...  branch according to item choisen
      GO TO (1000,2000,3000,4000,5000,2000,1000),ITEM
      CALL MODERR('error in INPUT@')
C
 1000 CONTINUE
C...  Corralation allowance
         IF(.NOT. REALIN(1,CA,'ENTER CORRELATION ALLOWANCE:@')) RETURN
            IF(NOTALL) RETURN
C
 2000 CONTINUE
C...  water density and viscosity for towing tank
         IF(.NOT. REALIN(2,VEC,
     +      'ENTER TANK WATER DENSITY AND KIN. VISC.:@')) RETURN
            RHOTK=VEC(1)
            GNUTK=VEC(2)
            IF(NOTALL) RETURN
C
 3000 CONTINUE
C...  model dimensions
         IF(.NOT. REALIN(2,VEC,
     +      'ENTER LWL AND WETTED SURFACE OF THE MODEL:@')) RETURN
            XLM=VEC(1)
            SURFM=VEC(2)
            IF(NOTALL) RETURN
C
 4000 CONTINUE
C...  scale ratio (lambda), if 0 is entered a request
C...  for the LWL of the ship is made.
```

```
      IF(.NOT. REALIN(1,LAMBDA,'ENTER LINEAR SCALE RATIO, LAMBDA:@')) RETUR
         IF (LAMBDA .EQ. 0.) GO TO 4100
C...  compute ship properties
         XLS=LAMBDA*XLM
         SURFS=LAMBDA**2*SURFM
         GO TO 4200
 4100    CONTINUE
C...  branch to here if Lambda=0
      IF(.NOT. REALIN(1,XLS,'ENTER LWL OF SHIP:@')) RETURN
C...  compute scale ratio and wetted surface of the ship
         LAMBDA=XLS/XLM
         SURFS=LAMBDA**2*SURFM
 4200 CONTINUE
         IF(NOTALL) RETURN
C
 5000 CONTINUE
C...  establish a menu to locate the resistance data
      ILOC=MENUIN('WHERE    ',2,'FILE@DATABASE@',
     +            'WHERE IS THE RESISTANCE DATA:@')
C...  branch accordingly
         GO TO (5100,5200),ILOC
      RETURN
 5100 CONTINUE
C...  prompt for resistance filename and read the file
      IF(.NOT. CHARIN(20,FILE,'ENTER FILE NAME:@')) RETURN
         CALL SETDEV(1,FILE)
            READ(1,00001,ERR=5150,END=5150) N
            READ(1,00002,ERR=5150,END=5150) (VIN(I),RIN(I),I=1,N)
00001          FORMAT(I5)
00002          FORMAT(2F15.5)
      CALL SORT(N,VIN,RIN,VM,RTM)
      WRITE(6,00003)
00003    FORMAT('*** DATA READ CORRECTLY ***')
C...  flip the switch for a successful read
      TEST(1)=.TRUE.
      GO TO 5400
 5150 CONTINUE
C...  warn the user of a read error
      WRITE(6,00004)
00004    FORMAT('*** ERROR IN READING FILE! -PLEASE CHECK ***')
C...  don't flip the switch!
       TEST(1)=.FALSE.
      GO TO 5400
 5200 CONTINUE
C...  if the user hasn't entered the database,
C...  prompt him to do so.
      IF(.NOT. TEST(3)) WRITE(6,00005)
00005   FORMAT('*** PLEASE ENTER-DB BEFORE COMPUTE ***')
 5400 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DEFIT
C
C     Default input routine.
C     Allows the user to alter program default values.
C
      INTEGER MNAME(5),SHIP(5),SNUM,DATE(2)
      REAL LAMBDA
      LOGICAL TEST(5)
      COMMON /TESTS/TEST
      COMMON /CONDTN/RHOTK,RHOSW,GNUTK,GNUSW,CA,LINE,LAMBDA
      COMMON /MODEL/XLM,SURFM,N,VM(30),RTM(30),CR(30),CFM(30),CTM(30)
      COMMON /SHIPS/XLS,SURFS,VS(30),VK(30),CFS(30),RTS(30),PE(30),
     +             CTS(30),RNS(30),RFS(30),RRS(30)
      COMMON /TITLES/MNAME,MNUM,SHIP,SNUM,DATE
C
      REAL VEC(2)
      LOGICAL REALIN,INTIN,CHARIN
C
C...  establish the default menu choises
      DOUBLE PRECISION DFMEN(5)
      DATA DFMEN/8HMODEL   ,8HSHIP    ,8HDATE    ,8HOCEAN   ,
     +           8HLINE    /
C
  500 CONTINUE
C...  prompt the user with the menu and branch accordingly
         ITEM=MENUIN('DEFAULT ',5,DFMEN,'@')
            GO TO (1000,2000,3000,4000,5000),ITEM
               CALL MODERR('error in DEFAULT@')
 1000 CONTINUE
C...  model id
         IF(.NOT. CHARIN(20,MODEL,'ENTER MODEL NAME:@')) RETURN
         IF(.NOT. INTIN(1,MNUM,'ENTER MODEL NUMBER:@')) RETURN
            GO TO 9999
 2000 CONTINUE
C...  ship id
         IF(.NOT. CHARIN(20,SHIP,'ENTER SHIP NAME:@')) RETURN
         IF(.NOT. INTIN(1,SNUM,'ENTER SHIP NUMBER:@')) RETURN
            GO TO 9999
 3000 CONTINUE
C...  date
         IF(.NOT. CHARIN(8,DATE,'ENTER DATE; MM/DD/YY:@')) RETURN
            GO TO 9999
 4000 CONTINUE
C...  ocean conditions
         IF(.NOT. REALIN(2,VEC,'ENTER WATER DENSITY AND KINEMATIC VISCOSITY:@
     +                        ')) RETURN
            RHOSW=VEC(1)
            GNUSW=VEC(2)
               RETURN
 5000 CONTINUE
C...  extrapolater line
         LINE=MENUIN('LINE    ',2,'ITTC@ATTC@',
     +               'WHICH EXTRAPOLATOR LINE:@')
 9999 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CALC
C
C      CALC controls the computations in the module
C      First check to see if all the data is available,
C      then perform all the calculations.
C
      INTEGER MNAME(5),SHIP(5),SNUM,DATE(2)
      REAL LAMBDA
      LOGICAL TEST(5)
      COMMON /TESTS/TEST
      COMMON /CONDTN/RHOTK,RHOSW,GNUTK,GNUSW,CA,LINE,LAMBDA
      COMMON /MODEL/XLM,SURFM,N,VM(30),RTM(30),CR(30),CFM(30),CTM(30)
      COMMON /SHIPS/XLS,SURFS,VS(30),VK(30),CFS(30),RTS(30),PE(30),
     +              CTS(30),RNS(30),RFS(30),RRS(30)
      COMMON /TITLES/MNAME,MNUM,SHIP,SNUM,DATE
C
C...  check to see that all the information is available
      CALL CHECK(&9999)
C
C...  call EXTRAP to perform all the calculations
      CALL EXTRAP
C...  call PLOT to prepare the graphic displays
      CALL PLOT
C...  set test(5) to .TRUE. so user may PRINT or DRAW
      TEST(5)=.TRUE.
C
C...  print appropriate message
         WRITE(6,00001)
00001       FORMAT(//,'*** CALCULATIONS COMPLETED ***',//)
         GO TO 10000
 9999 CONTINUE
         WRITE(6,00002)
00002       FORMAT(//,'*** CALCULATIONS INTERRUPTED ***',//)
C
10000 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CHECK(*)
C
C     check to see that all data is available
C
      INTEGER MNAME(5),SHIP(5),SNUM,DATE(2)
      REAL LAMBDA
      LOGICAL TEST(5)
      COMMON /TESTS/TEST
      COMMON /CONDTN/RHOTK,RHOSW,GNUTK,GNUSW,CA,LINE,LAMBDA
      COMMON /MODEL/XLM,SURFM,N,VM(30),RTM(30),CR(30),CFM(30),CTM(30)
      COMMON /SHIPS/XLS,SURFS,VS(30),VK(30),CFS(30),RTS(30),PE(30),
     +              CTS(30),RNS(30),RFS(30),RRS(30)
      COMMON /TITLES/MNAME,MNUM,SHIP,SNUM,DATE
C
      TEST(4)=.FALSE.
C...  tank conditions?
         IF(RHOTK .GT. 0.) GO TO 10
            WRITE(6,00001)
00001          FORMAT('**ERROR -TANK CONDITIONS UNDEFINED**')
         TEST(4)=.TRUE.
   10 CONTINUE
C...  model dimensions?
         IF(XLM .GT. 0.) GO TO 20
            WRITE(6,00002)
00002          FORMAT('**ERROR -MODEL DIMENSIONS UNDEFINED**')
         TEST(4)=.TRUE.
   20 CONTINUE
C...  ship dimensions?
         IF(XLS .GT. 0.) GO TO 30
            WRITE(6,00003)
00003          FORMAT('**ERROR -SHIP DIMENSIONS UNDEFINED**')
         TEST(4)=.TRUE.
   30 CONTINUE
C...  resistance data?
         IF(TEST(1).OR.TEST(3)) GO TO 40
            WRITE(6,00004)
00004          FORMAT('**ERROR -NO RESISTANCE DATA**')
         TEST(4)=.TRUE.
   40 CONTINUE
C...  if TEST(4) .EQ. .TRUE. don't allow user to COMPUTE
         IF(TEST(4)) RETURN 1
      RETURN
      END
```

```
      SUBROUTINE EXTRAP
C
C     This routine performs the extrapolation of the model
C     data via Froude's method.  See module documentation
C     for details.
C
      INTEGER MNAME(5),SHIP(5),SNUM,DATE(2)
      REAL LAMBDA
      LOGICAL TEST(5)
      COMMON /TESTS/TEST
      COMMON /CONDTN/RHOTK,RHOSW,GNUTK,GNUSW,CA,LINE,LAMBDA
      COMMON /MODEL/XLM,SURFM,N,VM(30),RTM(30),CR(30),CFM(30),CTM(30)
      COMMON /SHIPS/XLS,SURFS,VS(30),VK(30),CFS(30),RTS(30),PE(30),
     +              CTS(30),RNS(30),RFS(30),RRS(30)
      COMMON /TITLES/MNAME,MNUM,SHIP,SNUM,DATE
C
C
C...  develop model coefficients of resistance
      XKM=0.50*RHOTK*SURFM
         DO 1000 I=1,N
            CTM(I)=(RTM(I)/VM(I)**2)/XKM
            CFM(I)=CF(VM(I),XLM,GNUTK,LINE)
            CR(I) =CTM(I)-CFM(I)
 1000    CONTINUE
      XKS=0.50*RHOSW*SURFS
      SQRTLM=SQRT(LAMBDA)
C...  develop ship coefficients of resistance
         DO 2000 I=1,N
            VS(I)=VM(I)*SQRTLM
            CFS(I)=CF(VS(I),XLS,GNUSW,LINE)
            CTS(I)=CFS(I)+CR(I)+CA
            RTS(I)=CTS(I)*XKS*(VS(I)**2)
            RFS(I)=CFS(I)*XKS*(VS(I)**2)
            RRS(I)=CR(I)*XKS*(VS(I)**2)
            PE(I)=(RTS(I)*VS(I))/550.
            VK(I)=VS(I)/1.6889
 2000    CONTINUE
C
      RETURN
      END
```

```
      FUNCTION CF(V,L,GNU,LINE)
C
C     CF computes the frictional coefficients by either the
C     ATTC or ITTC formula.
C
      REAL L
C
      RENO=(V*L)/GNU
C...  ITTC line
         CFI=.075/((ALOG10(RENO)-2.)**2)
            I=1
         IF(LINE .EQ. 2) GO TO 1000
         CF=CFI
      RETURN
1000 CONTINUE
C...  ATTC line
C...  A NEWTON-RAPHSON iterative solution is used to solve
C...  for CF here.
         SQRTCF=SQRT(CFI)
         F=ALOG10(RENO*CFI)*SQRTCF- 0.242
         DF=(.4342945/(RENO*CFI))*SQRTCF
     +      +0.5*ALOG10(RENO*CFI)/SQRTCF
         CFN=CFI-F/DF
C...  stopping criteria i=10 or epsilon=1.e-07
            IF(ABS(CFN-CFI) .LE. 1.E-07) GO TO 1100
         CFI=CFN
            I=I+1
         IF(I .GT. 10) GO TO 1200
         GO TO 1000
 1100 CONTINUE
         CF=CFN
      RETURN
 1200 CONTINUE
C... set CF=0.0 and notify user
      CF=0.0
         WRITE(6,00001)
00001 FORMAT('*** CF NOT CONVERGING ***')
      RETURN
      END
```

```
      SUBROUTINE PLOT
C
C     PLOT creates the displays for the graphic output.
C
      LOGICAL TEST(5)
      COMMON /TESTS/TEST
      COMMON /CONDTN/RHOTK,RHOSW,GNUTK,GNUSW,CA,LINE,LAMBDA
      COMMON /MODEL/XLM,SURFM,N,VM(30),RTM(30),CR(30),CFM(30),CTM(30)
      COMMON /SHIPS/XLS,SURFS,VS(30),VK(30),CFS(30),RTS(30),PE(30),
     +              CTS(30),RNS(30),RFS(30),RRS(30)
C
C
      IF(TEST(2)) CALL DSPLAY(9,IDISP,IPLOT,ILINE,SIZE)
      TEST(2)=.TRUE.
C...  prepare graph for EHP VS. VK
C
C...  find the maximum and minimum values for the graph
      CALL MAXMIN(N,PE,PEMAX,PEMIN)
      CALL MAXMIN(N,VK,VKMAX,VKMIN)
C...  prepare the graph
      CALL GRAF(1,1,VKMIN,VKMAX,1.,PEMAX,-1.,1.,-.75,.70,0,0,
     +          5,5,2,2,0,1,'VELOCITY (KTS)',14,2,
     +          'EFFECTIVE HORSEPOWER',20,2,'EHP VS. VK',10,3)
C...  plot PE vs. VK
      CALL GLINE(1,1,1,VK,PE,N,7,1,3,IDASH,3,1,3,ICHAR)
C...  prepare graph for coef. of resistance
C
C...  find the maximum and minimum values for the graph
      CALL MAXMIN(N,RTS,RTMAX,RTMIN)
C
C...  prepare the graph
      CALL GRAF(2,1,VKMIN,VKMAX,1.0,RTMAX,-1.,1.,-.75,.70,
     +          0,0,5,5,0,0,NDASH,1,'VELOCITY (KTS)',14,2,
     +          'RESISTANCE',10,2,'RESISTANCE VS. VELOCITY',
     +          23,3)
C
C...  plot resistance of the ship vs. velocity in knots
      CALL GLINE(2,1,1,VK,RFS,N,7,1,3,IDASH,3,1,3,ICHAR)
      CALL GLINE(2,1,2,VK,RRS,N,7,1,3,IDASH,2,1,2,ICHAR)
      CALL GLINE(2,1,3,VK,RTS,N,7,1,3,IDASH,1,1,5,ICHAR)
C
      RETURN
      END
```

```
      SUBROUTINE DBFILL
C
C     Fill the database with the ship data.
C
      INTEGER MNAME(5),SHIP(5),SNUM,DATE(2)
      REAL LAMBDA
      LOGICAL TEST(5)
      COMMON /TESTS/TEST
      COMMON /CONDTN/RHOTK,RHOSW,GNUTK,GNUSW,CA,LINE,LAMBDA
      COMMON /MODEL/XLM,SURFM,N,VM(30),RTM(30),CR(30),CFM(30),CTM(30)
      COMMON /SHIPS/XLS,SURFS,VS(30),VK(30),CFS(30),RTS(30),PE(30),
     +              CTS(30),RNS(30),RFS(30),RRS(30)
      COMMON /TITLES/MNAME,MNUM,SHIP,SNUM,DATE
C
      CALL RPUT('LWL ',XLS)
      CALL RPUT('SWET ',SURFS)
C
      CALL IPUT('N ',N)
      CALL APUT('VS ',30,VS)
      CALL APUT('CFS ',30,CFS)
      CALL APUT('CR ',30,CR)
      CALL APUT('CTS ',30,CTS)
      CALL APUT('RTS ',30,RTS)
      CALL APUT('PE ',30,PE)
      CALL APUT('VK ',30,VK)
C
      RETURN
      END
```

```
      SUBROUTINE DRAW
C
C     This routine shows the user the graphic displays
C      on a graphics terminal.
C
      INTEGER MNAME(5),SHIP(5),SNUM,DATE(2)
      REAL LAMBDA
      LOGICAL TEST(5)
      COMMON /TESTS/TEST
      COMMON /CONDTN/RHOTK,RHOSW,GNUTK,GNUSW,CA,LINE,LAMBDA
      COMMON /MODEL/XLM,SURFM,N,VM(30),RTM(30),CR(30),CFM(30),CTM(30)
      COMMON /SHIPS/XLS,SURFS,VS(30),VK(30),CFS(30),RTS(30),PE(30),
     +              CTS(30),RNS(30),RPS(30),RRS(30)
      COMMON /TITLES/MNAME,MNUM,SHIP,SNUM,DATE
      LOGICAL NOTALL
      DOUBLE PRECISION DRWMEN(3)
      DATA DRWMEN/8HPOWER   ,8HRESIST  ,8HALL     /
C
C...  test to see if plots were generated
      IF(.NOT. TEST(5)) GO TO 1000
      IGO=4
   50 CONTINUE
      ITEM=MENUIN('DRAW    ',3,DRWMEN,'@')
      NOTALL=ITEM .NE.3
      GO TO (100,200,100),ITEM
      GO TO 50
  100 CONTINUE
C...  display EHP vs. VK
      CALL DSPLAY(IGO,1,IPLOT,ILINE,SCALE)
      IF (NOTALL) RETURN
      IF(IGO .EQ. 4) CALL WAIT('@')
  200 CONTINUE
C...  display RESISTANCE vs. VK
      CALL DSPLAY(IGO,2,IPLOT,ILINE,SCALE)
      IF (NOTALL) RETURN
      IF(IGO .EQ. 1) CALL WAIT('@')
      RETURN
 1000 CONTINUE
C...  write appropriate message
      WRITE(6,00001)
00001 FORMAT('*** MUST compute before draw ***')
      RETURN
      END
```

```
      SUBROUTINE PRINT
C
C     This routine handles all the printing of information
C     in the module.
C
      INTEGER MNAME(5),SHIP(5),SNUM,DATE(2)
      REAL LAMBDA
      LOGICAL TEST(5)
      COMMON /TESTS/TEST
      COMMON /CONDTN/RHOTK,RHOSW,GNUTK,GNUSW,CA,LINE,LAMBDA
      COMMON /MODEL/XLM,SURFM,N,VM(30),RTM(30),CR(30),CFM(30),CTM(30)
      COMMON /SHIPS/XLS,SURFS,VS(30),VK(30),CFS(30),RTS(30),PE(30),
     +           CTS(30),RNS(30),RFS(30),RRS(30)
      COMMON /TITLES/MNAME,MNUM,SHIP,SNUM,DATE
C
      LOGICAL NOTALL
      INTEGER NAME(2)
      DATA NAME/4HITTC,4HATTC/
C
      DOUBLE PRECISION PRIMEN(5)
      DATA PRIMEN/8HINPUT    ,8HMODEL    ,8HSHIP     ,8HRESULTS ,
     +           8HALL      /
C
      ITEM=MENUIN('PRINT   ',5,PRIMEN,'@')
      IWRITE=6
C



        ENTRY PRIDEV(ITEM,IWRITE)
C
      NOTALL=ITEM .LT. 4
C...  test to see that computations were performed
      IF(.NOT. TEST(5) .AND. ITEM .GT. 1) GO TO 100
      GO TO (1000,2000,3000,2000,1000),ITEM
      CALL MODEER('error in PRINT@')
  100 CONTINUE
      WRITE(6,00020)
00020 FORMAT('*** MUST compute before print ***')
      RETURN
C...  print the input
C
 1000 CONTINUE
      WRITE(IWRITE,00001) DATE
00001 FORMAT('1',14X,'INPUT VALUES FOR MODEL AND SHIP',5X,
     +        'DATE: ',2A4,///)
      WRITE(IWRITE,00002) MNAME,SHIP,MNUM,SNUM
00002 FORMAT(10X,'MODEL: ',5A4,7X,'SHIP: ',5A4,/,
     +         9X,'NUMBER:',I5,21X,'NUMBER:',I5,///)
      WRITE(IWRITE,00003) XLM,XLS,SURFM,SURFS,LAMBDA
00003 FORMAT(10X,'LWL',11X,':',F12.3,6X,'LWL',11X,':',F12.3,/,
     +        10X,'WETTED SURFACE:',F12.3,6X,'WETTED SURFACE:',F12.3,//,
     +        10X,'SCALE RATIO    :',F12.3,///)
      WRITE(IWRITE,00004) RHOTK,RHOSW,GNUTK,GNUSW
00004 FORMAT(32X,'WATER PROPERTIES',//,
     +        10X,'DENSITY  :',F12.5,12X,'DENSITY  :',F12.5,/,
     +        10X,'KIN. VIS.:',E12.5,12X,'KIN. VIS.:',E12.5)
      IF(NOTALL) RETURN
```

```
 2000 CONTINUE
C...  print the model results
      WRITE(IWRITE,00005)
00005 FORMAT('1',32X,'MODEL RESULTS',//,
     +         16X,'VM',10X,'CFM',13X,'CR',12X,'CTM',10X,'RTM')
      WRITE(IWRITE,00006) (VM(I),CFM(I),CR(I),CTM(I),RTM(I),I=1,N)
00006 FORMAT(10X,F10.3,3X,E12.5,3X,E12.5,3X,E12.5,3X,F10.3)
      IF(NOTALL) RETURN
 3000 CONTINUE
C...  print the ship results
      WRITE(IWRITE,00007)
00007 FORMAT('1',33X,'SHIP RESULTS   PART 1',//,
     +         16X,'VS',10X,'CFS',12X,'CR',11X,'CA',10X,'CTS')
      WRITE(IWRITE,00008) (VS(I),CFS(I),CR(I),CA,CTS(I),I=1,N)
00008 FORMAT(10X,F10.3,3X,E12.5,3X,E12.5,F10.5,E15.5)
      WRITE(IWRITE,00009)
00009 FORMAT('2',33X,'SHIP RESULTS   PART 2',//,
     +         16X,'VS',12X,'RTS',14X,'EHP',15X,'VK')
      WRITE(IWRITE,00010) (VS(I),RTS(I),PE(I),VK(I),I=1,N)
00010 FORMAT(10X,F10.3,5X,E12.5,5X,F12.5,5X,F10.3)
      WRITE(IWRITE,00011) NAME(LINE)
00011 FORMAT('-',9X,'FRICTIONAL COEFFIENTS FROM THE ',A4,' LINE')
C
      RETURN
      END
```

```
      SUBROUTINE DEVICE
C
C     This routine sets a device or filename to where the
C     output is to be sent.
C
      INTEGER MNAME(5),SHIP(5),SNUM,DATE(2)
      REAL LAMBDA
      LOGICAL TEST(5)
      COMMON /TESTS/TEST
      COMMON /CONDTN/RHOTK,RHOSW,GNUTK,GNUSW,CA,LINE,LAMBDA
      COMMON /MODEL/XLM,SURFM,N,VM(30),RTM(30),CR(30),CFM(30),CTM(30)
      COMMON /SHIPS/XLS,SURFS,VS(30),VK(30),CFS(30),RTS(30),PE(30),
     +                CTS(30),RNS(30),RFS(30),RRS(30)
      COMMON /TITLES/MNAME,MNUM,SHIP,SNUM,DATE
C
      LOGICAL CHARIN
      INTEGER OUTPUT(5)
      DOUBLE PRECISION PRIMEN(5)
C...  establish the menu choises
      DATA PRIMEN/8HINPUT    ,8HMODEL    ,8HSHIP    ,8HRESULTS ,
     +            8HALL      /
C
      ITEM=MENUIN('PRINT    ',5,PRIMEN,'@')
C...  read the FDNAME and set it to device #7
      IF(.NOT.CHARIN(20,OUTPUT,'WHERE?@')) RETURN
      CALL SETDEV(7,OUTPUT)
      CALL PRIDEV(ITEM,7)
C
      RETURN
      END
```

```
      SUBROUTINE SAVE
C
C     SAVE displays for future reference or to plot
C     on the CALCOMP.
C
      INTEGER MNAME(5),SHIP(5),SNUM,DATE(2)
      REAL LAMBDA
      LOGICAL TEST(5)
      COMMON /TESTS/TEST
      COMMON /CONDTN/RHCTK,RHOSW,GNUTK,GNUSW,CA,LINE,LAMBDA
      COMMON /MODEL/XLM,SURFM,N,VM(30),RTM(30),CR(30),CFM(30),CTM(30)
      COMMON /SHIPS/XLS,SURFS,VS(30),VK(30),CFS(30),RTS(30),PE(30),
     +               CTS(30),RNS(30),RFS(30),RRS(30)
      COMMON /TITLES/MNAME,MNUM,SHIP,SNUM,DATE
      INTEGER FILE(8)
      LOGICAL REALIN,NCTALL,CHARIN
C...  the same menu as DRAW
      DOUBLE PRECISION DRWMEN(3)
      DATA DRWMEN/8HPOWER   ,8HRESIST  ,8HALL     /
C...  test to see that plots were generated
      IF (.NOT. TEST(5)) GO TO 100
C
      IDISP=MENUIN('DRAW    ',3,DRWMEN,'@')
      IGO=5
      IF(IDISP .EQ. 3) IGO=7
C...  Scale the plot size
      IF(.NOT.REALIN(1,SCALE,'ENTER PLOT SIZE (INCHES):@'))RETURN
C...  establish the plotfile
      IF(.NOT. CHARIN(32,FILE,'ENTER PLOT FILENAME:@')) RETURN
      CALL SETDEV(9,FILE)
      CALL DSPLAY(IGO,IDISP,IPLOT,ILINE,SCALE)
      RETURN
  100 CONTINUE
C...  write appropriate message
      WRITE(6,00001)
00001 FORMAT('*** MUST compute before save ***')
      RETURN
      END
```