The MIT Marine Industry Collegium
Opportunity Brief #13

Computer-Aided Preliminary Design of Ships

MIT
SEA
GRANT
PROGRAM

A Project of
The Sea Grant Program
Massachusetts Institute of Technology
MITSG 78-13

The MIT Marine Industry Collegium

COMPUTER-AIDED PRELIMINARY DESIGN OF SHIPS

Opportunity Brief #13

Revised Edition

July 1, 1978

Marine Industry Advisory Services

MIT Sea Grant Program

Cambridge, Massachusetts 02139

Report No. 78-13

Index No. 78-713-Ztb

PREFACE

This Opportunity Brief and the accompanying Workshop (held on May 23, 1978) were presented as part of the MIT/Marine Industry Collegium program, which is supported by the NOAA Office of Sea Grant, by MIT and by the more than 90 corporations and government agencies who are members of the Collegium. The underlying studies at MIT were carried out under the leadership of Professor C. Chryssostomidis, but the author remains responsible for the assertions and conclusions presented herein.

Through Opportunity Briefs, Workshops, Symposia, and other interactions the Collegium provides a means for technology transfer among academia, industry and government for mutual profit. For more information, contact the Marine Industry Advisory Services, MIT Sea Grant, at 617-253-4434.

Norman Doelling

1 July, 1978

TABLE OF CONTENTS

1. A Business Perspective

For nearly three decades, the computer has been a valuable although limited tool for the designers of ships, platforms, and other offshore structures. The computer's value has been principally as a high speed calculator for solving well-defined analytic problems encountered in the design process. The limitations of the computer are largely the result of its being a piece-meal add-on to the design process rather than a systematically integrated one.

This Opportunity Brief describes a computer-based system specifically organized and implemented to aid in the preliminary design of ships and other complex naval systems. As such, it greatly extends the computer's usefulness in exploring design alternatives and eases the burden of routine record-keeping and information handling.

The system is the result of a collaborative effort of faculty and students at the Massachusetts Institute of Technology, at the University of Michigan, and at the University of Colorado. In addition, practising naval architects have used parts of the system in a commercial environment and have provided ideas and criticisms based on their experience.

The system consists, in part, of a set of related computer programs (such as sea keeping programs, stability programs, calm water resistance programs, and many others), which have proven to be useful design tools. There are more than 15 distinct programs, and the number is growing. These programs are written in FORTRAN, and all are in the public domain. The only exception is the seakeeping program, which is available to U.S. companies only.

Of much greater importance than the family of programs is an operating system known as the Design Executive. DEX enables the designer to use all of the application programs in a uniquely convenient and rational way. DEX eliminates many of the rigidities inherent in using "batch" oriented computer programs and systems. It provides convenient storage and information retrieval, so that all or part of one program's output is available as another design program's input. DEX aids the designer in selecting which program to run next. It permits interruption of programs during execution, changing of variables values, and many other processes that are natural to designers but usually forbidden to computer users. These capabilities of DEX are usually associated with database management.

DEX is written in FORTRAN and is a time-sharing executive running under the control of another time-sharing monitor. While DEX cannot be said to be "machine independent," it is "transportable" to other computer systems and is currently running on IBM, Amdahl and CDC computers.

The focus of this Brief is the DEX system as applied to the preliminary design of ships. However, it is well worth noting that DEX could be equally applicable to the design of other complex systems in which the interactive, intuitive involvement of the designer is essential. The DEX system represents a unique tool that provides a combined facility, under the system designer's control, for design computation and database management.

In summary, the system described in this brief provides ship design tools that are valuable both as a set of useful programs and as an aid in the efficient control of the programs and data.

2. Some Comments on the Design Process

Given a set of design goals, the first step in designing ships or other complex systems is identifying the alternatives to be investigated. The only tools available for generating these alternatives are the designer's imagination, past experience, and knowledge of what has been tried and reported by others. Designers must create the alternatives to be investigated because it is only among these alternatives that an "optimum" will be found.

Ship design is an iterative process in which the designer converges toward an optimum system through a series of progressively refined estimates and calculations. There is no method for circumventing this process. Because of the indirect nature of design, the more alternatives that one investigates, the greater the likelihood of approximating a true "optimum". However, in real life, the number of alternatives that can be considered and the modes of investigation must be constrained by time and money.

The presence of these constraints forces a designer to use relatively crude models at the early stages of design to identify the most promising alternative and to define the areas that merit more detailed analysis. Using crude models in the early stages of design introduces some uncertainty into the conclusions. If subsequent and more detailed analysis yields values different from those assumed in earlier iterations, and if the differences affect the solution, the designer must iterate back and begin the solution process again.

The final step in design is selecting the "optimum" configuration of the alternative chosen in the previous step and producing its detailed design. Once again, a designer is forced to adopt an iterative solution whereby the detail increases progressively while the number of candidates studied in each iteration decreases.

From the above description one can draw at least the following conclusions about the design process. First, design (even for something as specific as a ship) is a process that cannot be formulated as an automatic or predetermined procedure; it is driven by a human designer, who must decide what needs to be studied and when and how it needs to be studied. Therefore, any design method worth consideration must allow the designer to assume complete control when necessary. Also, a good design method must allow rapid calculation of analytic problems encountered during execution. On both counts, computer aided design offers the most suitable solution.

However, computer aids have also introduced problems quite unknown to the ship designer of the past. There is a large and growing library of analytic programs to choose from. In addition, computers provide far more information than can be effectively processed with manual design methods. While such information is unquestionably of value, it cannot be effectively used unless it is integrated into, rather than appended to, the design process. Completely computerized design is almost certain to fail, since the human designer must remain the creative force behind any design procedure. Creative imagination and intuition cannot be quantified or incorporated into a computer program, and any procedure which excludes these peculiarly human aspects can only lead to a mediocre product.

In the following section we describe the DEX system and explain how it overcomes the limitation of the computer as a mere "add on" while insuring that the designer is given freedom to control the overall process and contribute all the uniquely human qualities necessary for good design.

3. Using DEX in the Preliminary Design of Ships

Two classes of problems have typically confronted persons using computers as an aid in the design of complex systems. The first relates to certain rigidities of procedure that have been an inherent part of computer usage. These problems are most notable in the batch processing mode, in which cards must be punched, organized, and submitted to perform any given calculation. However, even using a computer in a time-shared interactive mode may impose limitations. Operational restrictions can make it difficult to control the process or inconvenient to interrupt a program, perform some other calculation, and then continue execution of the interrupted program.

The second class of problems relates to the difficulties of handling the large number of programs and information involved in computer usage. To be an effective design tool, the computer should be able to provide convenient storage and information retrieval, easy communication between programs (so output from one program can easily become input for another), and the means for sharing data among users working on the same problem.

While these two classes of problems have by no means excluded the computer from the design process, they have prevented the development of a maximized partnership between human designers and computers.

A computer program known as the Design Executive (DEX) addresses these two classes of problems and resolves many of them in an efficient and effective manner. Reference 4 describes an early version of DEX. In the subsection below, we discuss how problems of computer-aided ship design are resolved by DEX in its current version.

3.1 Overcoming Procedural Rigidities

DEX makes a major step forward by operating in an interactive, time-shared environment. This change alone enhances designer control by eliminating the slow response time and many procedural restrictions imposed by batch processing methods. DEX incorporates a number of special features that make the computer easy to work with, flexible, and, as nearly as possible, "transparent" to the user.

For example, DEX allows the designer to interrupt a program in execution, perform another calculation or create necessary data, and then return to the point of interruption. A designer executing a program might discover the need for the value of shaft horsepower (SHP) and find that the DEX program module for computing SHP is inappropriate for this design. The designer wants SHP to be 20,000 horsepower. The program may be interrupted so the designer can CREATE a REAL variable called SHP, STORE into this variable the value 20000. and then resume the program by simply issuing the CONTINUE command (see Figure 1).

Another luxury that DEX has made available to the designer is the OPERSYS command. With this command, the designer may interrupt DEX, gain access to the operating system of the time-sharing computer on which DEX is running, and use all the services of the time-sharing system. For instance, if space is available one could write, edit, compile and execute a program and then resume operation with DEX. This is a parallel to the manual design operation in which the designer discovers a vital piece of information is missing and performs a side calculation to obtain it before proceeding with the original problem.

Other particularly attractive features of DEX from the user's point of view are:

a) DEX traps a wrong command (in actual fact any wrong input) and prompts the user to re-enter it correctly.

b) DEX allows abbreviations for each command, thus facilitating operations for the skilled user, but retains the ability to accept full names.

c) DEX prompts the user when it expects the next command.

d) DEX provides "HELP" facilities and facilities to inform the designer about the current status of a program.

3.2 Database Management

3.2.1. Selection of a Program Path

DEX allows the designer to select a course of action via menus. Just as in a restaurant where one is presented with a menu and asked to make a choice, DEX presents the designer with a DEX menu from which the desired program path may be selected. Figure 1 shows some of the DEX menus.

Menus are basically nodes in a decision tree, along which the designer travels. They are presented in a sequential fashion. DEX starts by presenting the designer with DEX.MAIN, which is the topmost menu. The designer now has a choice of ten commands listed in the menu. Selecting one item from DEX.MAIN will generally cause DEX to select another menu in order to offer the user further options. If the user selects DISPLAY, DEX will prompt the entry of a menu item from DEX.DISP. Once this is done, control will be returned to the calling menu because the user is unlikely to want to issue two successive commands from DEX.DISP. On the other hand, if the user selects EDITDB (i.e., change some data or add data to the database), the menu DBEDCMDS (Data Base EDit CoMmanDS) will appear, and the designer will be prompted to enter an item from DBEDCMDS. When this is carried out, control remains in DBEDCMDS until the user issues a DONE command, because it is expected that the user will wish to issue two or more successive commands from DBEDCMDS.

Use of menus will be described in more detail in Section 4.

MENU

DEX.MAIN

LIBRARY

HELP

DISPLAY

ALTER

TIDY

EDIT-DB

BEGIN

CONTINUE

OPER-SYS

QUIT-DEX

Figure 1

Menu DEX.MAIN

3.2.2 Information Handling

Convenience in accessing data, modifying data, using output data of one program as input to another, and storing data are essential capabilities incorporated into DEX for use by designers. The database in DEX is the place in which variables (including arrays) are stored under names meaningful to the designer. For example, the designer can store information about the beam of a ship in the database under the name BEAM rather than as the nth number in a sequential file. This information can be retrieved for any DEX compatible design program by requesting the value of BEAM. DEX even provides the capability of adding a comment to a variable name to make it more explicit. For instance,

BEAM (R) 105.   BEAM AT AMIDSHIP

where (R) stands for Real Number.

This capability of DEX is almost like making available to the designer a notebook in which all the relevant design information can be found. It takes very little imagination to see the side advantages that DEX's database has introduced to the design process. The database contains the latest information on the design for use by the different specialists involved, and thus helps resolve one of the most complicated human problems of design - namely, that of communication.

A number of subtle but important problems have also been dealt with by DEX. For example, if designer A changes the beam to improve seakeeping performance, how is designer B, who has already used the "old" beam value to calculate shaft horsepower, to be informed? Since designers C, D, and E also used the "old" value, should A be allowed to make the change at all? Part of the elegance of DEX is its ability to handle such issues flexibly.

4. An Example of DEX at Work

The menu approach and its convenience are best illustrated by an example. The designer, D, starts at DEX.MAIN (Figure 1), selects LIBRARY from the DEX.MAIN menu, and gets a second menu showing the library of programs. D wants to calculate shaft horsepower (SHP) using a technique well known to naval architects. D selects the shaft horsepower program, called SER60P.DEX (about 14 other major program modules handle other aspects of ship design such as sea keeping, stability, floatation, etc.), and a third menu, called MOD.MAIN, appears as shown in Figure 2. D can always jump back to DEX.MAIN by typing DONE.

MOD.MAIN, the topmost menu, offers the user nine choices, which are described below. Figure 3 graphically presents all nine choices and the subsequent command alternatives.

1. SOURCE allows D to define the source of the input. When D types SOURCE, a new menu presents four choices: DATABASE, USER, FILE, or DEFAULT. DATABASE means that the input will come from a DEX created database; USER means that the designer will supply the information from the keyboard of his terminal. FILE is a means of communicating with non DEX, batch oriented users. When this option is selected, DEX basically simulates a batch-like program and reads (and writes) sequential (batch) files. DEFAULT allows the designer to use "default" values for any input variable which is missing.
2. DESTINAT allows D to define the destination of the output as DATABASE, USER, or FILE, as described above.

3. STYLE allows D to define dialogue as TERSE (abbreviated command style, for skilled user) or VERBOSE (fully spelled out style), and to choose whether to print out database error messages (DBER-ON) or not (DBER-OFF).
4. UNITS allows selection of ENGLISH or METRIC units.
5. MODE allows review of the alternatives chosen from the above lists. When D types MODE, the module responds by telling which dialogue (TERSE or VERBOSE) is in effect; whether the database error flag inhibitor is activated; the SOURCE of the input; the DESTINATion of the output; the UNIT system (METRIC or ENGLISH) being used.
6. INPUT calls up a new menu with the options shown in Figure 3. Because the list of options is long and important, we describe it more completely below.
7. OUTPUT allows D to move output from any SOURCE or from any program to an output device, such as a line printer or graphics terminal. Its structure and subsidiary menus are similar in intent to those of INPUT.
8. COMPUTE directs the module to perform all the calculations necessary in determining the power needed to propel a given ship at a given speed.
9. DONE returns command to the calling menu, which is always DEX.MAIN.

For completeness, we describe the individual elements of the INPUT menu (Figure 3):

1. SOURCE is as described previously.
2. STYLE is as described previously

3. ALL instructs the machine to run the program and execute all commands using the specific parameters selected for the commands below (4 - 11). For example, if D runs SER60P.DEX for a particular design, selecting ALL in menu INPUT, and if SOURCE has been designated as USER, the program will proceed to prompt D for all input.
4. TITLE* reads alphanumeric information that distinguishes one run from another. (See below for the significance of the asterisk.)
5. UNITS* is as described previously.
6. VESSELI allows the designer to read in the geometric properties of the ship.
7. FLUIDI allows definition of the fluid properties.
8. GRAVITY* allows the user to read in a value for g, gravity acceleration.
9. CFLINE* allows the designer to select the ATTC or ITTC friction line for extrapolating the series results.
10. SPDEPINP calls up a new menu (see Figure 4) which allows the designer to define all the input that can be speed dependent.
11. TIDYSP allows the designer to organize the speeds and the dependent input (and output) in order of increasing speed. This reorganization is useful if the output is to be plotted.

The starred variables inform D of the database names of the variables already used in the program in question. For example, GRAVITY* signals the user that there exists a variable by the name GRAVITY in the database.

As Figures 4 and 5 indicate, D may continue to work through additional menus, although we will not go into the details here. The point of the example is to show the organization of menus, which allows logical and orderly progression from one set of subroutines and data to the next. By helping D home in on finer and finer details, DEX permits a systematic progression toward a solution, fully under the control of the designer.

MENU

MOD.MAIN

SOURCE

DESTINAT

STYLE

UNITS*

MODE

INPUT

OUTPUT

COMPUTE

DONE

Figure 2

Menu for MOD.MAIN

MENU	MENU	MENU	MENU	MENU
SOURCE	DESTINAT	STYLE	UNITS*	INPUT
DATABASE	DATABASE	TERSE	ENGLISH	SOURCE
USER	USER	VERBOSE	METRIC	STYLE
FILE	FILE	DBER-ON		ALL
DEFAULT		DBER-OFF		TITLE*
				UNITS*
				VESSELI
				FLUIDI
				GRAVITY*
				CFLINE*
				SPDEPINP
				TIDYSP
				DONE

Figure 3

Control Menus for Module SER60P.DEX

MENU	MENU	MENU	MENU
VESSELI	FLUIDI	CFLINE	SPDEPINP
SOURCE	SOURCE	ATTC	SOURCE
STYLE	STYLE	ITTC	STYLE
UNITS*	UNITS*		UNITS*
ALL	ALL		ALL
LWL*	RHO*		LOINDEX*
LBP*	GNU*		HIINDEX*
BEAM*	DONE		SPEEDS*
DRAFT			CORALL*
CB*			APPALL*
WTSRF*			SERVALL*
DONE			DONE

Figure 4

Examples of Menu Options Under Menu INPUT

MENU	MENU	MENU	MENU
SPEEDS*	APPALL*	SERVALL*	EFFIC
RANDOM	PTOTRES	PTOTARES	WTIORRSG
EQUAL-SP	PFRCRES	SERVRES	HUOPRRSH
	APPRES		QUASISH
			OVERALL*

Figure 5

Some Examples of Menu Options Under Menu SPDEP1NP

SUMMARY

Because of their reliance on menus, all DEX design programs have a highly modular structure which greatly facilitates expansion, modification and transportability of program segments. DEX does, however, introduce two relatively minor disadvantages in the execution and writing of design modules. First, DEX is an overhead that each design module must pay. Fortunately, DEX is relatively small and at present most computers are not memory bound, so the penalty is only a small increase in the cost of execution. Secondly, a DEX based design module is lengthier. Despite these minor drawbacks, the advantages of using DEX far outweigh the disadvantages.

Because of its flexible structure DEX should be adaptable to future computer-aided design systems. It is still lacking some important functions necessary in design, but its structure is adequate to accommodate these additions easily. Although every effort has been made to make DEX transportable, it still has some system dependencies which make it difficult to transport unless the recipient organization has a good system programmer to install it and support it. This problem is reduced because DEX is now available at Michigan University, Colorado University, and MIT, and is being installed at the U.S. Naval Academy.