M C Meyer

MICROCOMPUTER APPLICATIONS IN YACHT DESIGN

by

John S. Letcher, Jr.
Naval Architect
Letcher Offshore Design
Southwest Harbor, Maine

INTRODUCTION

It is commonplace today to hear that the world is in the midst of a computer revolution. Computers are dramatically changing the ways of working in countless occupations, and most participants in the revolution show enthusiasm for the changes and excitement about the prospects for future developments. In general, computers are seen as taking over the routine, repetitive tasks in any business, performing them with speed, precision and reliability, and freeing the human mind to focus on the creative, varied challenges at which it excels. Further, partnership between a computer and a human mind can be greater than the sum of its parts -- creativity can flourish best when it has access to the right information, organized and presented in the right way.

Naval architecture, though undoubtedly a vocation full of creative and varied challenges, has its share of tedious repetition. The current generation of microcomputers, equipped with the necessary programs, can go a long way toward eliminating tedium and error in fairing lines, hydrostatic calculations, weight and c.g. estimates, and rating calculations. These are things we all know how to do, but they take a lot of time and as a consequence either are not done properly or absorb an inordinate amount of the designer's time. But there's another category of things the computer can do for the naval architect, things which were really not possible to attempt before the computer age. In this category I would put the Velocity Prediction Program, which simulates the balance of forces and moments acting on a sailing yacht, allowing prediction of actual sailing speeds, heel angles, etc. and assessment of the impact of design changes on them.

We'll first review the current state of microcomputer development and availability with particular attention to the needs of the naval architect; then outline various application areas with the characteristics of computers and programs needed to serve in each.

Microcomputers and Peripherals -- October 1982

Please note the precise dating in the title of this section; it indicates that developments are taking place not year by year but month by month. Any information given here about the current status is very likely to be out of date within a year as we see new generations of machines introduced, new manufacturers entering the market, and lower prices through increased production levels and competition. I recommend BYTE magazine as a voluminous monthly source of up-to-date information on the whole range of microcomputers.

The core of any computer is the central processing unit (CPU), the "brain" where arithmetic and logical operations are actually performed. In order to make a useful computer we also need input/output (I/O) devices to communicate with the CPU, and memory to store data, instructions and intermediate results. Memory comes in two levels -- random access memory (RAM) and read-only memory (ROM), accessible to the CPU in times on the order of a microsecond, and disk memory with access times on the order of seconds. Memory is measured in Kilobytes or K's -- about 1000 characters, or 500 5-digit numbers. Computers today are roughly classified into mainframes, minicomputers and microcomputers (Table 1). Mainframes

Table 1. Classes of Computers

Class	Mainframe	Minicomputer	Microcomputer
CPU bits	32-64	16-32	8-16
RAM	1M & up	64K-1M	4K-128K
Disk	10M & up	.5M-10M	100K-1M
Operations/sec	10^8 & up	10^7	10^6
Price	\$$10^6$ & up	\$50,000-\$$10^6$	\$100-\$10,000

typically serve the computing needs of hundreds or thousands of users through time-sharing or batch processing. A minicomputer typically serves the accounting needs of a moderate-sized business. Although microcomputers have very limited speed and memory by comparison, their capabilities are growing all the time, and a micro of the current generation can serve the single user with presently available naval architecture applications very adequately.

Comparing the scores of microcomputers now available, what features are important to the small-craft designer, and how can a rational selection be made?

(1) Random access memory. Though a lot can be done in 16K memory, additional memory is very cheap now (about $50 per 16K) and any 8-bit machine should be filled out to 64K (total ROM and RAM). Even with only 16K of memory, larger programs can be overlaid (new sections of program read in from disk as they are needed) in most machines, making up for a lack of memory. One gauge on memory requirements is to see how much RAM is required for a very detailed table of offsets -- say z and y components for 400 points on one side of a hull. This requires 1.6K to 6.4K on various machines, depending on the format in which numbers are stored. Thus the whole shape of a hull can be contained in 10 to 40% of RAM even in a 16K machine, leaving lots of room for a program to operate on it (e.g. hydrostatics).

(2) CPU characteristics. A 16-bit CPU can address more than 64K of memory locations, allowing memory to be expanded over 64K, which is not ordinarily possible with an 8-bit CPU. (I doubt if this is a real advantage, because of the possibility of using overlays as mentioned above.) It also can process twice as many channels and, in theory, can work twice as fast as an 8-bit CPU. Clock speed also directly affects the speed of arithmetic operations and RAM storage. There are some calculations we do (e.g., hydrostatic equilibration, velocity prediction) in which a faster response time would be desirable -- finding equilibrium flotation for one loading, or equilibrium speed for one wind speed and one heading might take 2-4 minutes. But this time has not proved tc be a serious limitation; typically we set up 20 cases or so for the machine and then go away and do something else for an hour.

(3) Keyboard and screen. These basic I/O devices are more or less equivalent on various microcomputers. Some people prefer the feel of one keyboard over another, or prefer to look at small sharp letters on a small screen rather than big fuzzy letters on a big screen. I have not noticed that it makes much difference.

(4) Disk memory. Compared with cassette storage, which we found very unreliable and slow, disk memory is a blessing. One disk drive is the first thing I'd buy

to add to a basic 16K machine. A $5\frac{1}{4}$" diskette can hold about 85K (single density) or 170K (double density). For example 170K can hold about 25-100 of those detailed offset tables, or 17 good-sized (10K) programs, or the information in about 70 pages of a typical book. Diskettes cost about $2.50 each in quantities of 10. A second disk drive is expecially useful for maintaining backups (reserve copies) of programs and data.

(5) RS-232 interface. This is a blessedly standard input/output channel allowing microcomputers to talk to each other and to a wide variety of I/O devices such as printers, plotters and digitizers. With the addition of a modem and a terminal program, the microcomputer can act as a terminal and communicate with time-sharing systems, information services and other microcomputers via telephone. The RS-232 is essential if you want to do any of these things.

(6) Printer. Though the screen allows quite a bit of information (0.5 to 1K) to be presented at one time, and it's possible to get along for a long time without a printer by copying what you need off the screen (or by storing it on disk rather than on paper), printed output is a great convenience. Neat tables of output can be printed; program listings are a great help in program development; word processing becomes possible. Some printers can act as a plotter as well, either

(e.g. several dot-matrix printers) by placing dots in a row to make a line, or (e.g. Radio Shack 26-1190) by moving a ball-point pen on the paper for both plotter and printer functions.

(7) Plotters. A plotter is a graphic output device which moves a pen over paper, or moves the paper under the pen, under computer control. Since a great deal of the output of a naval architect is graphic material, the plotter can be an extremely valuable device. Plotters are classified into two categories -- <u>flatbed</u> plotters in which the paper is stationary and therefore of limited size, and <u>drum</u> plotters in which the plotting medium is a strip having effectively unlimited length in one direction. Since naval architects are accustomed to working on rather large drawings, say 24" x 36", a small ($8\frac{1}{2}$" x 11" or 11" x 17" flatbed or 9" - 11" drum) plotter could be rather limiting; however, drawings can be segmented by a program and drawn on several strips or numerous sheets which fit together to form the complete drawing. It is likely that the plotter quality, and programming and computing time needed to create a finished, reproducible drawing directly on the plotter will not be available in the immediate future. But as any draftsman knows, producing a neat tracing from a drawing takes only a fraction of the time it takes to create the drawing, and this is

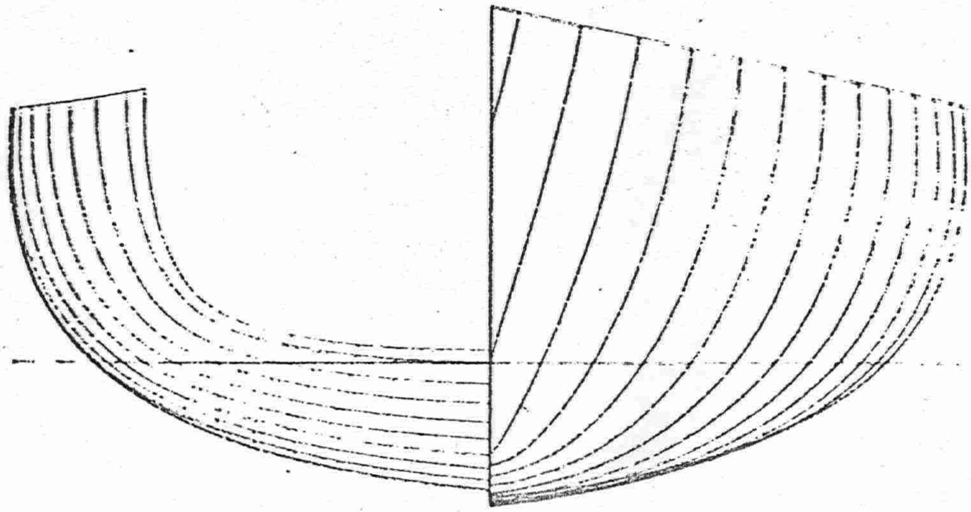
a practical way to utilize the small plotters available at present for around $1000.

Although most plotters are driven via an RS-232 port, they vary widely in the signals they require to do a particular plotting job. Therefore plotters are not interchangeable, unless the appropriate parts of the program are changed to produce the proper plotter instructions.

(8) Screen graphics. High-resolution graphics allow useful information to be displayed graphically on the screen, for example, a body plan to give a quick visual check of hydrostatics offsets or a new hull definition. 256 x 192 (e.g. Apple II) is about the minimum resolution for this purpose. Even with 512 x 192 I have been disappointed in the quality of lines drawings -- they are good enough only for rough, qualitative judgments -- and in the speed of plotting from a BASIC program. I think the impressive 3-D projections of complex surfaces we see in microcomputer advertising take a long time to draw -- like hours -- and the beautiful images of the space shuttle we see in so many CAD ads today are done on far bigger computers and more expensive screens, with ten times this resolution in both directions. I don't attach much importance to having the present level of screen graphics. A plotter

gives far better images, in a hard-copy form suitable for reproduction and close comparisons (Figure 1).

Fig. 1. Body plan drawn by Radio Shack 26-1190 plotter.



(9) Languages. All microcomputers offer the user one or more high-level languages, which allow programming with English-like statements rather than machine-code instructions. The high-level program is then translated into machine instructions by another program called a compiler or interpreter.

BASIC is the _lingua franca_ of the microcomputer world, and there are about as many dialects of it as there are of French -- or, at least, as there are microcomputer makes and models. Do not expect a BASIC program written for one computer to run on another without extensive modification. However, the recent widespread adoption of the CP/M operating system among manufacturers and users of high-end microcomputers with 64K RAM is a bright star on this horizon; any microcomputer with this standard operating system can run a standard version of BASIC (BASIC-80 or MBASIC), making programs portable from one make to another.

The various BASICs vary rather widely in their features, flexibility and convenience. This is only significant if you plan to do a considerable amount of programming yourself.

A distinction needs to be made between an interpreter and a compiler for running a high-level language. An interpreter, which is the standard form of BASIC supplied with most machines, usually in 8-12K of ROM, translates the program line by line and immediately forgets the translation as soon as the instructions are executed. This form of BASIC is relatively slow, because it spends so much time re-translating repeated instructions; however it is ideal for program development because of its interactive character -- you can run and stop the program in small segments, examining intermediate results, examining portions of the program, tracing the sequence of execution and modifying the program freely in searching for errors. A compiler translates the whole program once and for all before running it; the program then runs much faster -- typically 3 to 10 times faster -- but is far less accessible to the programmer for locating and fixing errors.

Software.

"Software" refers to the programs a computer runs, as opposed to "hardware", the physical components. A program is a set of instructions, either in machine language or a high-level language, which is loaded from disk into part of RAM and then executed to carry out a desired sequence of calculations or other operations. "Documentation", an important part of most software, is the instruction manual that comes with the program telling how to use it, defining input and output quantities and explaining how to interpret the output.

Software can come from several sources:

(1) Some comes with the computer -- for example, the disk operating system and parts of the BASIC interpreter are ordinarily supplied as software. These programs are intended for very wide distribution and are normally extensively tested and well documented.

(2) Program packages distributed by the computer manufacturer are normally of the same high quality as (1).

(3) Third-party software suppliers are independent distributors of programs. The subject matter, broadness of appeal, quality and prices vary widely. Most microcomputer manufacturers publish lists of available third-party software. Naturally, far more software is

available for the more popular microcomputers and for CP/M machines than for the more obscure models.

(4) Numerous free programs are exchanged through magazines, users groups and information networks. Documentation is usually brief or nonexistent. The quality of anything that comes for free is open to serious question.

(5) Many microcomputer users are able to learn BASIC programming and find it an interesting and rewarding experience to develop specialized programs for their own needs.

Unless you expect to develop all your own software, a question of paramount importance is the availability of applications programs for the jobs you want to do. This one factor is more important in selecting a computer than all the hardware considerations mentioned in the preceding section.

NAVAL ARCHITECTURE APPLICATIONS

In this section we outline the principles of a number of computing tasks in which microcomputers can serve the needs of the naval architect.

Hull surface definition.

The drafting and fairing-up of lines is a tedious job which many naval architects would like to automate. According to Kinney [1], "It takes between 5 and 10 working days for an experienced man to produce a good set of lines, whether the boat is 25' over-all or 95' over-all."

There have been three basic approaches to using the computer to aid this process.

(1) The first is essentially a mechanization of the draftsman's procedure: representing the hull by a table of offsets for, say, the intersection of stations and waterlines, let the computer perform a smoothing operation on this table by alternate curve fitting along columns and rows, iterating until the required adjustments become sufficiently small. Though this approach can be demonstrated to converge and successfully smooth data in simple test cases, it has not been very successful with the more complex surfaces of real ships. Furthermore, it does not result in a complete surface definition, but rather a network of intersecting curves which still requires interpolation to locate most points on the surface.

(2) The second approach is that of fitting a mathematical equation systematically to a surface which has been previously defined to fairly high precision graphically [2]. This approach, requiring storage of a large number of offsets and solution of large sets of simultaneous equations with many variables and constraints,

may well strain the capabilities of microcomputers. Although many such schemes have been worked on and commercially developed, it is my impression that results have been disappointing and that real success here remains elusive. Further, it doesn't eliminate the initial graphical fairing of the lines but merely facilitates full-size lofting, or numerical control of production machinery.

(3) The third approach is for the hull designer to directly manipulate the mathematical formulas for the surface until it achieves a shape which satisfies him [3]. That this is well within the capabilities of microcomputers has been demonstrated by a very flexible system on the TI-59 calculator (Fairline/1) [4]. This has led to a more powerful and conceptually simpler Fairline/2 system for 32-64K microcomputers which is now being licensed to users worldwide. Here is a brief description of how a design proceeds using Fairline/2:

1. Rough lines are first sketched at any convenient scale -- profile, plan and body plan views with perhaps a few waterlines and diagonals, and stations agreeing roughly with these. A desired displacement and distribution of section area can be imposed (roughly) at this stage.

2. Offsets are lifted from the sketched lines to define a small number (3 to 6, typically) of "Master Curves" in the surface of the hull. (These are typically stations, but can include the bow and/or stern profiles.) These offsets are keyed into the computer in response to prompts. The addition of a few other numbers completes the mathematical "representation" of this particular hull, and the system immediately passes a fair spline surface through these specified curves. The representation can be stored on disk and read back in at any later time to retrieve this particular hull.

3. At this point the program can provide detailed offsets for any place on the surface, generating offsets along any plane section, which can be manually plotted or displayed on the screen or plotter as a lines drawing for the designer's approval. Upright hydrostatic calculations can be carried out in a few minutes with another part of the program, to verify intended displacement, section areas, etc.

4. Modifications, if desired, are made by adjusting specific numbers in the representation. Such adjustments have a local effect, moving the surface near where the change is made but blending into the rest of the surface in a way that preserves fairness. In addition, dramatic changes of size and proportion can be made by adjusting a few global parameters in the representation.

5. The program can write a disk offset file for the hull which can be used immediately as input to the velocity prediction program or a program to analyze inclined hydrostatics.

6. A full-size plotting service is available which starts from a Fairline/1 or /2 representation and produces a full-size body plan on mylar or paper for far less than competitive surface-fitting lofting services.

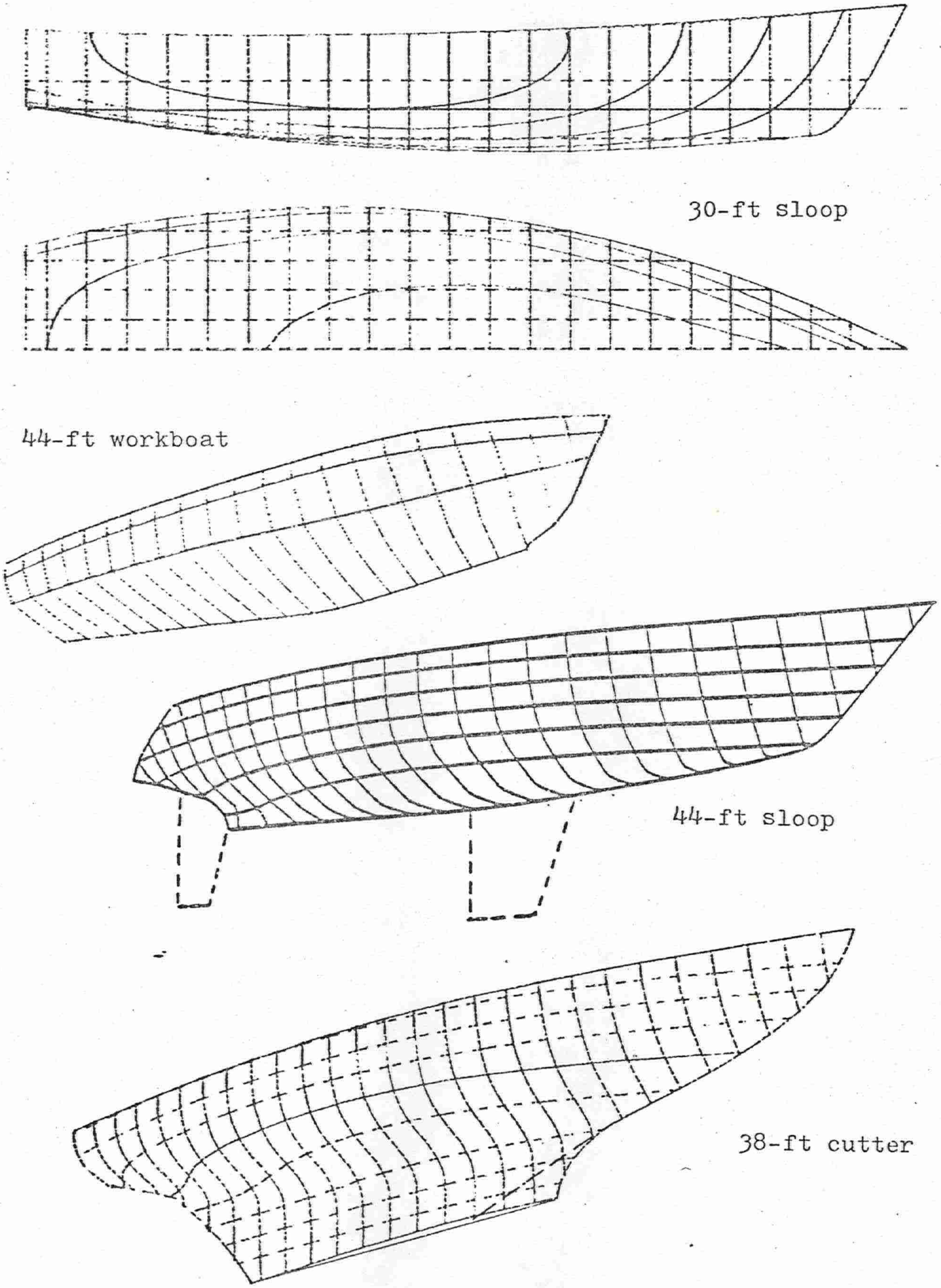
Figure 2 shows several hulls that have been designed or fitted with Fairline/1 and /2.

The design of developable hull surfaces (i.e. those that can be formed from flat material by pure bending rather than stretching) is a separate topic. A recent computer-based approach is outlined in Ref. [5]. The mathematics is well within microcomputer capabilities.

Drafting/rendering.

Several minicomputers or mainframe-based drafting systems are on the market. These store drawing information in a numerical form which can be changed interactively using a light pen or keyboard input. Although development of these systems has been intense, their penetration into industry is not very large; a great deal of drafting is still being done with pencil and paper by draftsmen.

Figure 2. An assortment of Fairline/1 and /2 hulls



I think the price of these drafting systems is beyond the range of most boat designers.

Rendering, the production of accurate pictorial views of a design, can be greatly assisted by a microcomputer. If hull (and, generally, deck and rig) offsets are stored in a disk file, perspective projections can be made from various distances and viewpoints. The transformations involved in perspective projection are simple and can be found in many books on computer graphics, for example 6 . A plot of such a projection gives the artist a dimensionally correct foundation from which to create a rendering of near-photographic accuracy. (Fig. 3)

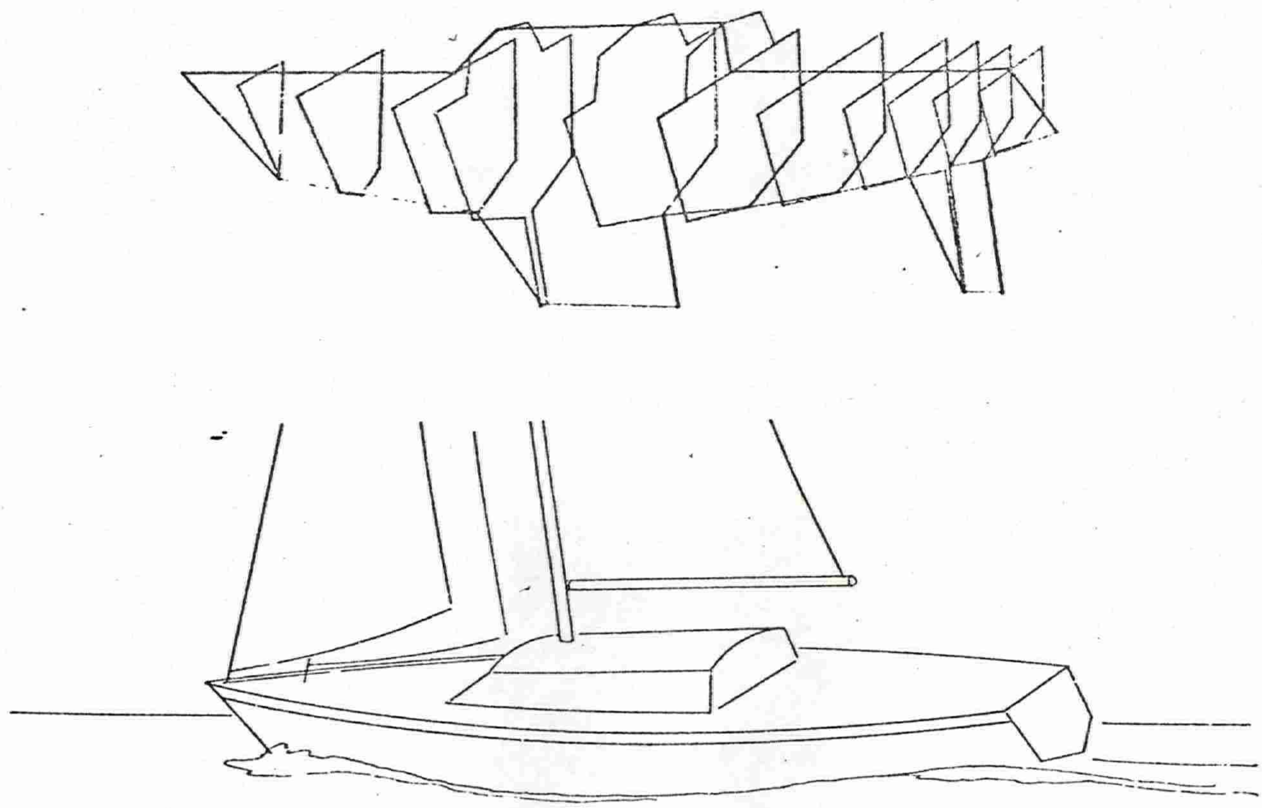


Fig. 3. Rendering based on perspective projection.

Hydrostatics.

A very small program can be written to accept planimeter readings and perform the longitudinal integrations by Simpson's or trapezoidal rules. A more advanced application, well within microcomputer capabilities, is to eliminate the planimeter, representing the hull by a table of offsets and letting the computer calculate the immersed volume and waterplane area and their centers with the hull in any position. There are two possible ways to represent the hull (Fig. 4): by a series of transverse stations (appropriate to elongated shapes like

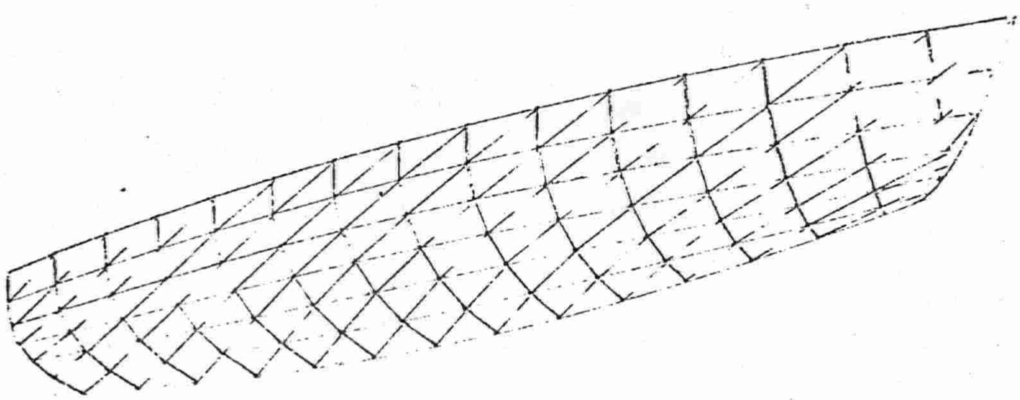
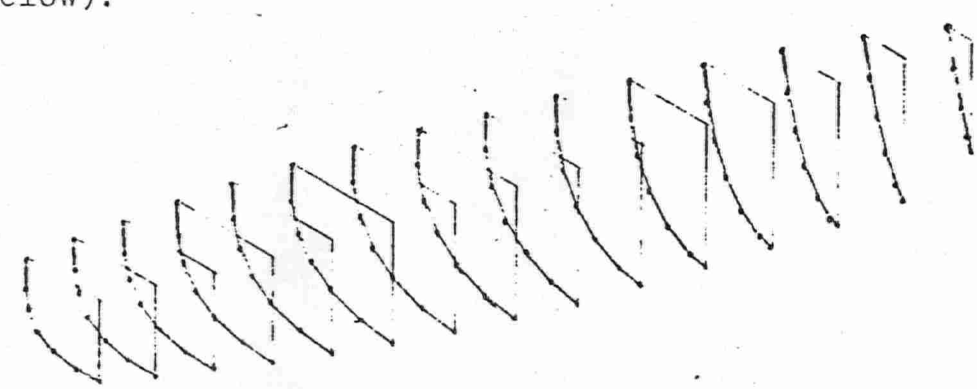


Figure 4. Representation of hull surface for hydrostatic calculations by triangular panels (above) or transverse sections (below).

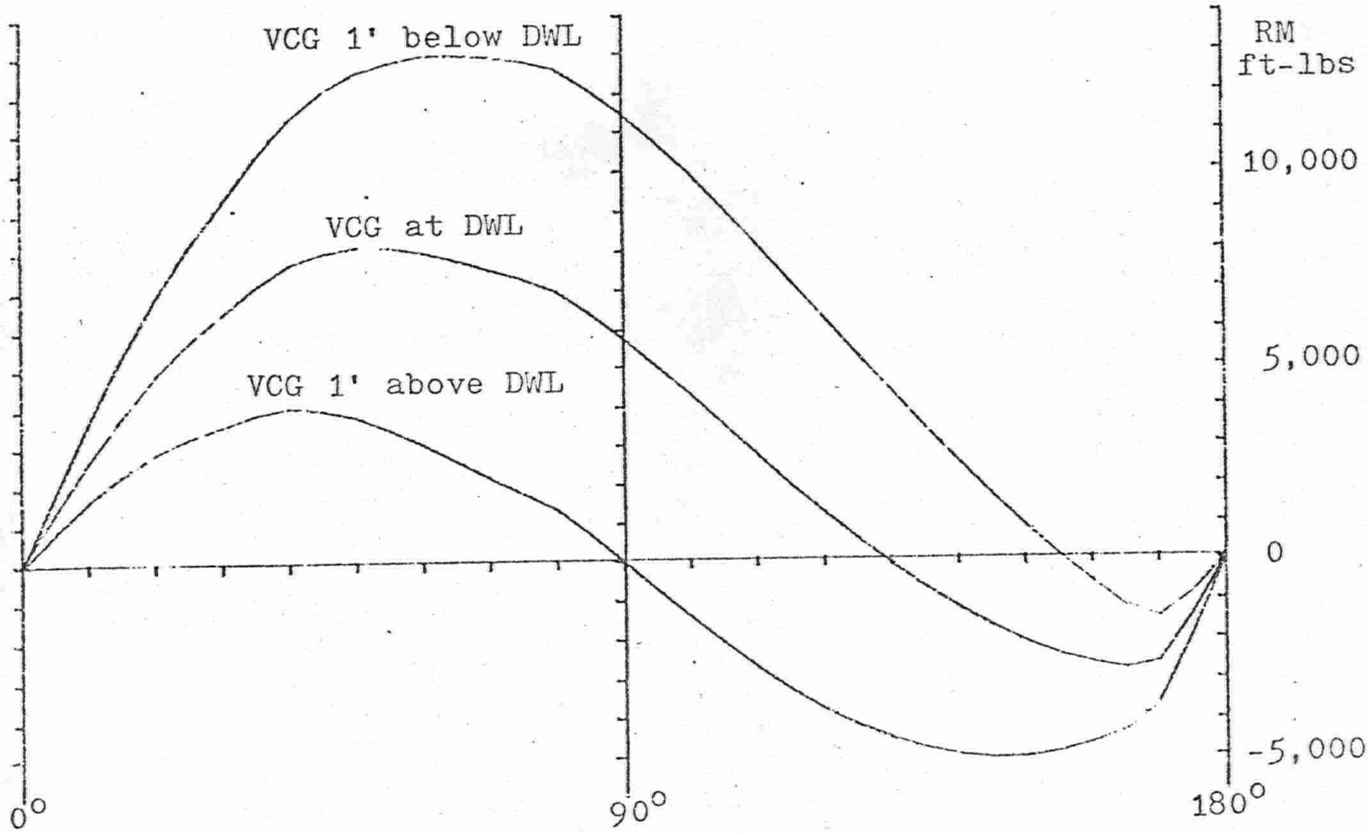


most boats), or by an assemblage of triangular panels (appropriate to more general shapes). In either case coordinate transformation have to be provided to allow the "boat" to be positioned arbitrarily with respect to the waterplane, and the core calculation is of either the immersed area and centroid of a transverse section, or the resultant force and force center on the wetted portion of a panel. Summations over sections or panels produce the resultant buoyant force and center of buoyancy.

In problems of most interest to naval architects the inclined waterplane is unknown, and the loading and heel angles are instead specified. Finding the sinkage and trim which will yield the correct immersed volume and longitudinal center is a process of either trial and error or of systematic search, which the computer can also patiently carry out. All this can be done with just 16K RAM if the offset table is limited to around 400 points. It is one of the slower routine calculations we carry out. Typical output is shown in Fig. 5, for a hard-chine boat represented by just 14 stations and 63 total points; the calculations for this graph require 30 min. in TRS-80 Disk BASIC, or 14 min. in TRS-80 Compiler BASIC.

The offsets for a hydrostatics calculation can be keyed directly into the program, but we have found it highly advantageous to put them on a disk file, using a separate

Figure 5. Righting moment vs. angle of heel for three VCG's.



program to create, edit, store and retrieve the offsets. This we call the "Offset File Editor". Some graphics capability is desirable for displaying the offsets, to guard against gross errors in data entry.

A program that can calculate the volume and centroid of irregularly-shaped bodies also has obvious applications in getting weight and c.g. of ballast or tank capacity.

Weight Schedule.

Weight calculations are straightforward bookkeeping exercises. The main problem is that as a design progresses, or as a boat is built, weights are added or subtracted (most likely added!), locations are shifted, and the weight schedule needs to be periodically updated, requiring a lot of tedious, error-prone additions and multiplications.

The obvious solution is to maintain the list on a disk file which can be read in, edited and printed out with the new totals and centers automatically calculated as items are changed. The file can either be kept right on the disk as a random-access file, with items being transferred individually on and off the disk as they are needed, or, with somewhat better response time and flexibility they can be read in in entirety and stored in arrays to be accessed by the program. One item with an identifying string of, say, 35 characters, and its weight and 3 c.g. coordinates requires 51 to 71 bytes, so you can store 15-20 items per K of RAM or disk storage.

Velocity Prediction Programs

A VPP is a mathematical model, at some level of abstraction and idealization, of a complete sailboat and its motion under the influence of aerodynamic and hydrodynamic forces acting on it. The inputs to the program are information about the hull shape (e.g. table of offsets), stability and rig (e.g. sail measurements). The output is the speed and angle of heel at which the boat will sail, as functions of the wind speed and true wind angle.

A variety of VPP's have been put together by various investigators over the past decade, motivated by both design applications and handicapping problems. These differ in the level of detail at which they attempt to model the yacht, in the specific approximations introduced to describe the various forces mathematically, and in the algorithm employed to find optimal solutions to the resulting equations of equilibrium. For a very basic VPP showing the principles of boat modeling and solution see [7]. The one that has attracted the most public attention and acceptance is the VPP developed in the Pratt Project at MIT in 1974-78 and used as the foundation of the Measurement Handicap System [8]. In a 1980 project for USYRU I showed that this complete VPP could be implemented on a 16K microcomputer, the Texas Instruments 99/4.

This, too, is one of our slower programs, yet with an efficient solution algorithm it runs plenty fast enough to be useful. Initial processing of the hull offsets to produce a resistance curve (based on the Delft systematic series of model tests) and heeled stabilities takes about 15 minutes in TRS-80 Disk BASIC; after that it takes 1 to 3 minutes to reach equilibrium for each combination of wind speed and angle, so in an hour you can have the polars pretty well mapped out. (Figure 6)

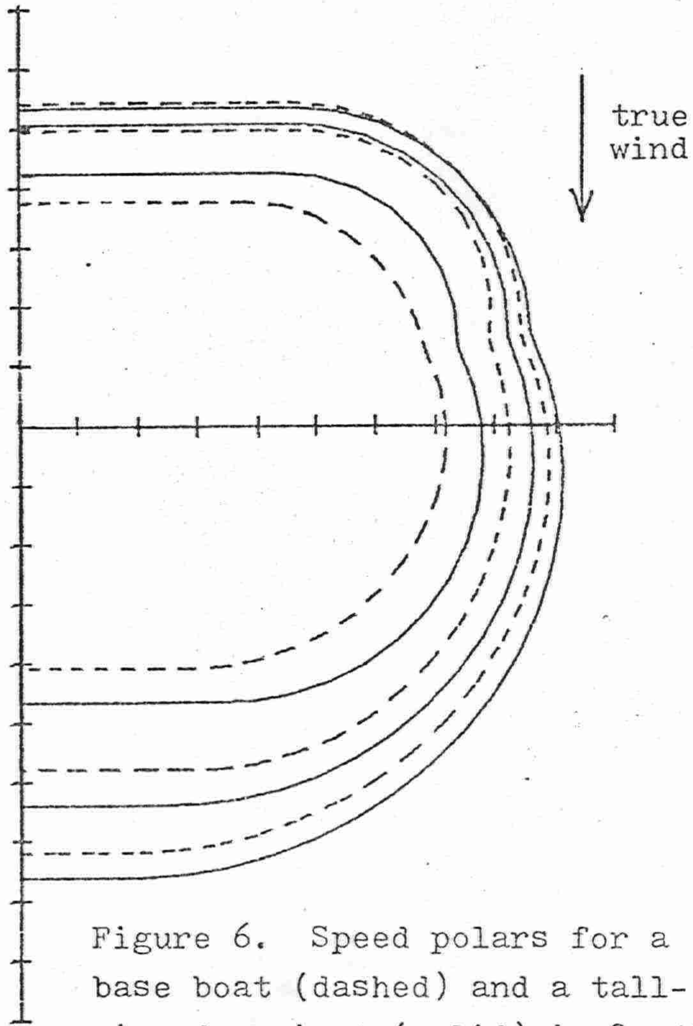


Figure 6. Speed polars for a base boat (dashed) and a tall-rigged variant (solid) in 8, 12 and 16 kt. true wind -- per VPP.

It has proven very difficult to obtain reliable speed data from real yachts to validate the VPP. But there have been many qualitative indications that it is good enough to be a useful tool for many design and handicapping problems. Typical reasonable design applications today are evaluation of the impact on performance of proposed changes in rig dimensions, stability, keel draft, or propeller installation. Areas which the MHS VPP does not attempt to treat, or in which its treatment is suspected of being shaky, include design factors which affect motion in waves (e.g. pitching moment of inertia), minimum keel area, keel-centerboard performance, rigging size and mechanical properties, and fine details of the hull design.

Rating.

The rating rules (e.g. IOR, MORC) contain a series of formulas which can be converted quite directly to BASIC statements, to make a program for calculating rating from measurements. This is a useful tool for a designer who is concerned with these rules, allowing quick evaluation of the rating impact of various measurements.

A program to take IOR hull measurements [9] from a mathematical representation of a hull, such as an offset table, is no mean task. I have developed such a program in a project for the Offshore Racing Council. The official version of the program is in Fortran and run on minicomputers at USYRU and RORC, but an equivalent BASIC program was implemented on a 48K TRS-80; it runs in about 10 minutes. The basic operations are interpolations between measured points on stations and between stations to infer the location of measurement points not on the measured stations.

Rigging calculations.

The standard methods of rigging strength calculations (e.g. [1]) are easy to reduce to a BASIC program. This can include tables of wire strengths to streamline selection of wire sizes; however this doesn't have to be done very often and it seems a good example of a calculation just as well left off the computer. The standard methods are rather conservative and no doubt are satisfactory for conventional rigs, moderately tuned.

However, one modern area where computers could be advantageously applied is the analysis of bendy rigs. The IOR's prohibition against "mechanically bent spars" has been stretched to the breaking point by designers seeking mainsail draft control and low windage. Once a mast is bent out of column and subjected to heavy preload stresses by tuning, the conventional Euler-buckling calculation loses all validity and a much more complete analysis must be undertaken to assure structural integrity of the rig. A foundation for such an advanced analysis is provided in [10]. Although this work does not recognize the essential limitation on mast loads imposed by the yacht's finite stability, it analyzes deflections, effect of preload, modes of failure and distribution of stress throughout the rig with more rationality and completeness than any other approach I am aware of. The level of mathematics involved is easily within the microcomputer range (15 x 15 linear equations).

Hull Scantlings.

The scantling rules covering various forms of construction are couched in formulas and tables which could rather easily be coded into BASIC programs. A somewhat more sophisticated program would combine the scantling rules with structural weight estimates, allowing the designer to explore alternatives within the scantling rules to minimize weight.

Another simple but useful program relating to scantlings would assemble some of the formulas and data from Gibbs and Cox [11] and from other sources relative to structural properties and weights of various laminates and core materials into a quick laminate design program.

A more fundamental approach to hull structure could be taken using finite element methods, wherein a continuous structure is modeled by an assemblage of a large number of simple beam or plate elements [12]. However, loads are probably not presently known sufficiently well to justify such an approach. I suspect the memory requirements may be far outside the microcomputer range, for finite-element-methods are usually mainframe stuff, but it's amazing what you can do within 48K if you're willing to wait until tomorrow morning to see the answer.

Word processing, business programs.

We should not overlook the more conventional and mundane applications of computers in the office. Word processing is particularly attractive for specifications, which tend to be very repetitive and often just need 5-10% of changes to become applicable to a new boat. With a word processing program you can keep one or more standard specification documents on file, reading one in and going through it paragraph by paragraph to make all necessary changes. Then the edited copy can be printed, rather than having to be retyped.

CONCLUSIONS

It should be apparent from the above that the current generation of microcomputers can do a great deal to assist the small-craft designer. Undoubtedly there are more wonders to come, but the hardware and software are already available to practically revolutionize boat design. The naval architect who embraces this opportunity can expect relief from the routine, repetitive tasks of lines drafting, hydrostatic and weight calculations; freedom to consider a wider range of alternatives in optimizing a design for a particular purpose; ability to respond more quickly to the needs of his clients; and opportunity to do his chosen work in a more thorough, scientific and professional fashion.

With the rapid improvements we are seeing in price and performance of microcomputers there is a tendency for many people to think "I'm glad I didn't buy one a year ago -- so maybe I should wait another year, to get more for my money." The trouble is, that logic could go on and on. What we have right now are marvellous machines and programs, worth far more than their price to any naval architect who is short of time; and the person who learns to use them in 1983 is going to be a long way ahead when 1984's opportunities come around.

REFERENCES

1. F. S. Kinney, Skene's Elements of Yacht Design, 8th Ed., Dodd, Mead & Co., New York (1973)

2. C. Kuo, Computer Methods for Ship Surface Design, Longman Group Ltd., London (1971)

3. J. S. Letcher, Jr. "A New Approach to Numerical Fairing and Lofting", Marine Technology, v. 9, pp. 223-231 (1972)

4. J. S. Letcher, Jr. "Mathematical Hull Design for Sailing Yachts", 5th Chesapeake Sailing Yacht Symposium, S.N.A.M.E., pp. 77-92 (1981)

5. J. C. Clements "A Computer System to Derive Developable Hull Surfaces and Tables of Offsets", Marine Technology, v. 18, pp. 227-233 (1981)

6. W. M. Newman and R. Sproul, Principles of Interactive Computer Graphics, 2nd. Ed., McGraw-Hill, New York

7. J. S. Letcher, Jr. "Optimum Windward Performance of Sailing Craft", J. of Hydronautics, v. 10, pp. 140-144 (1976)

8. J. E. Kerwin "A Velocity Prediction Program for Ocean Racing Yachts", M.I.T. Ocean Engineering Rep. 78-11 (1978)

9. International Offshore Rule, Offshore Racing Council

10. R. B. Nelson and R. A. Westmann "Structural Analysis and Failure of Rigging", 3rd A.I.A.A. Symposium on Aero/Hydronautics of Sailing, pp. 1-9 (1971)

11. Gibbs & Cox, Inc. Marine Design Manual for Fiberglass Reinforced Plastics, McGraw-Hill, New York (1960)

12. D. Liu and Y. K. Chen, "Fundamentals and Applications of the Finite-Element Method in Analyzing Structural and Non-structural Marine Problems", Marine Technology, v. 19, pp. 272-292 (1982)