SAILING YACHT PERFORMANCE WITH OPTIMIZATION

by

George S. Hazen

B.S.E., Princeton University
(1973)

Submitted in Partial Fulfillment of the Requirements
for the Degree of

Master of Science in Naval Architecture
and Marine Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

August 1975

Signature of Author: ____________________

Certified by: ____________________
Thesis Advisor

Accepted by: ____________________
Chairman
Departmental Committee on Graduate Students
Ocean Engineering Department

ACKNOWLEDGMENTS

The author wishes to thank his thesis advisor, Prof. J.E. Kerwin, for his encouragement and guidance in the writing of this thesis. He also wishes to express his gratitude to Prof. J.N. Newman and those others responsible for establishing the NAYRU/MIT Ocean Race Handicapping Project for which this study was initiated. And finally, the author wants to thank his wife for her support and aid in preparing this document.

ABSTRACT

A program is developed that computes the theoretical performance of sailing yachts given a mathematical model with which to evaluate the forces and moments acting upon the vessel. This system of equations comprising the mathematical model may incorporate as many as six independent variables. In addition, the program has capabilities that allow for the optimization of the performance equilibriums with respect to as many as three additional variables. No attempt is made to develop a new or improved model for the forces and moments affecting a sailboat's performance, instead the emphasis is placed on the development of a solution technique that can be used with any model involving six or fewer degrees-of-freedom. Examples of the program's output as well as the procedures used to evaluate the pertinent forces and moments are included in the appendices.

TABLE OF CONTENTS

Tables

TABLE OF CONTENTS

Figures

NOMENCLATURE

F - a principle force.

M - a principle moment.

$\vec{p}$ - vector of independent variables.

$\vec{q}$ - vector of optimization variables.

$\vec{R}$ - vector of errors in equilibrium equations.

S - matrix of "error sensitivities".

V - matrix of partial derivatives of boat speed.

V_A - apparent wind speed.

V_B - boat speed.

V_T - true wind speed.

β - apparent wind angle.

γ - true wind angle.

$\vec{\delta}$ - vector of increments to the independent variables.

$\vec{\Delta}$ - vector of increments used in formation of divided differences.

$\vec{\Delta q}$ - vector of increments to the optimization variables.

Subscripts

A - denotes aerodynamic.

H - denotes hydrodynamic.

i - denotes first dimension in 3-D space.

j - denotes second dimension in 3-D space.

k - denotes third dimension in 3-D space.

X - denotes direction parallel to x-axis.

NOMENCLATURE

Y - denotes direction parallel to y-axis.

Z - denotes direction parallel to z-axis.

Sailing Yacht Performance with Optimization

INTRODUCTION

There exist, for most vehicles of engineering interest, relatively straightforward and direct means of obtaining solutions to the problem of computing steady state performance equilibriums. One exception to this class of vehicles ,however, is surprisingly also one of man's most primitive, namely waterbourne craft deriving their propulsion from the relative motion between water and air at this "ancient interface". It is precisely this mode of operation, in the interface between two fluids, that causes the difficulty when attempting to analyze the performance of such vehicles.

In the absence of a direct means for obtaining the desired solutions, one must turn to an iterative procedure by which the forces and moments acting upon the vehicle in question may be brought into equilibrium. In the past, programs designed to perform this sort of iterative calculation (1),(2),(3),(4), have lacked generality. Thus it was desired to develop a program that would contain the numerical procedure necessary for bringing the required number of forces and moments into equilibrium, and to do so in as efficient a manner as possible.

It is the purpose of this paper to describe the program that was written to meet these requirements.

PROBLEM BACKGROUND

The equations describing the variables affecting the performance of sailing craft are decidedly nonlinear. Basic to even the simplest of these mathematical models is the apparent wind triangle. This series of equations relates the true wind and the boat's velocity to the apparent wind as seen in the moving reference frame of the sailboat. Figure 1 illustrates the vector addition of the velocities involved. The equations describing the trigonometric relationships depicted in Figure 1 are as follows: <1>

$$V_A = \text{Apparent wind velocity}$$
$$= \sqrt{(V_T \sin\gamma)^2 + (V_B + V_T \cos\gamma)^2}$$

$$\beta = \text{Apparent wind angle}$$
$$= \arctan[(V_T \sin\gamma)/(V_B + V_T \cos\gamma)]$$

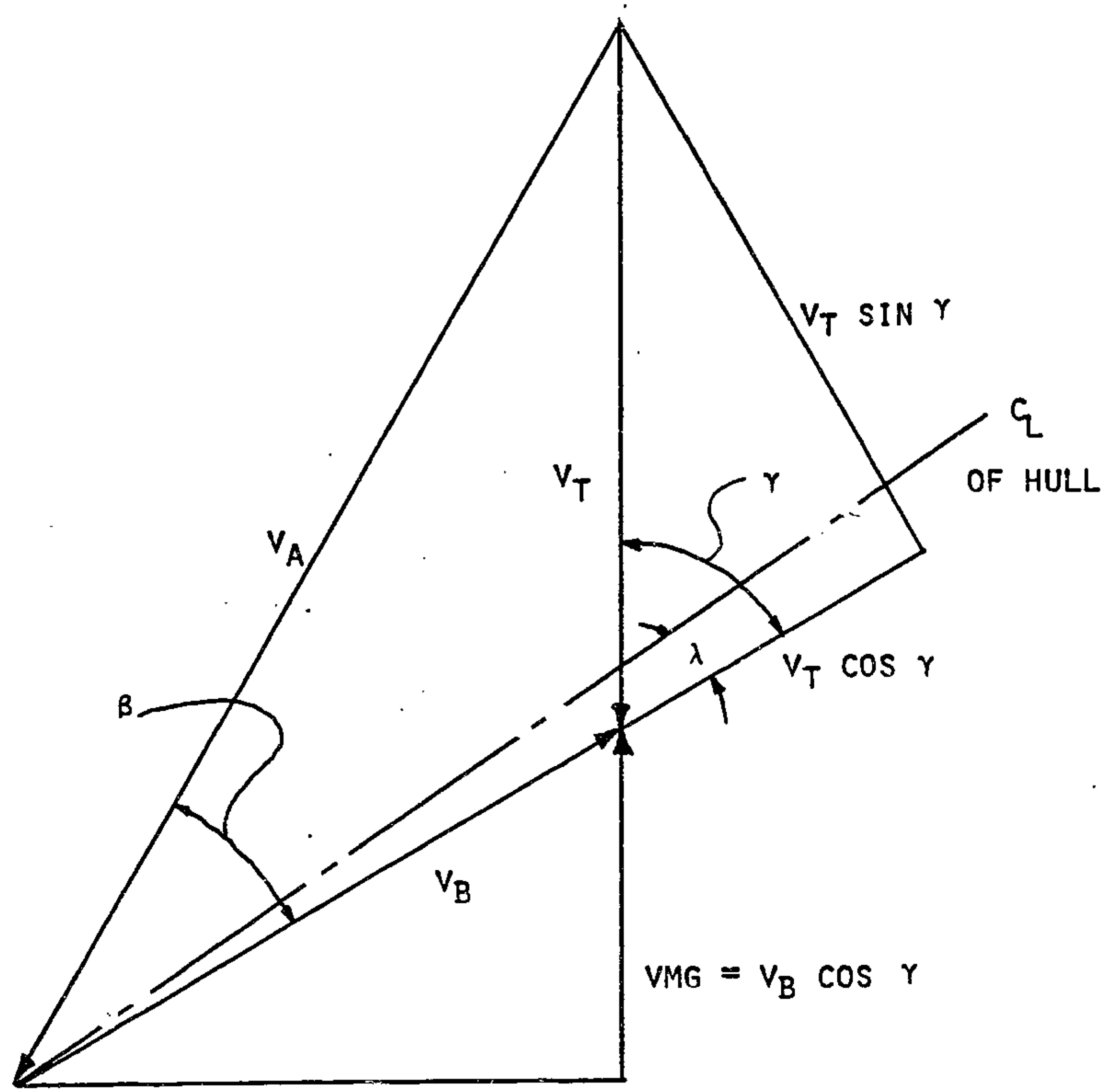
Thus it is apparent that irregardless of the simplifications made in any model chosen to describe the forces acting upon the sailboat, the aerodynamics and hydrodynamics of the vessel will always be coupled via this set of nonlinear relationships.

<1> Note should be taken of the fact that this definition of apparent wind angle is not the same as that usually used aboard a sailing vessel. Onboard instruments are intended to measure the angle between the apparent wind and the boat's centerline. Thus the definitions differ by landa, the leeway angle. The reason for using the preceding definition will become apparent in the discussion of the solution technique.

The simplest useful modeling of a sailboat's performance equations involves at least two-degrees-of-freedom (1). More typical models involve three- and four-degrees-of-freedom, but conceivably someone might want to extend their mathematical model to encompass a full six-degrees-of-freedom. When used in this context, each degree-of-freedom refers to an equation for one of the principle forces or moments acting upon the sailboat. In addition, there is a physical variable associated with each degree-of-freedom. For example, a model that requires the balancing of lateral and longitudinal forces as well as moments about the longitudinal axis would be a three-degree-of-freedom model. Here, the variables associated with each of these degrees-of-freedom would most likely be: leeway angle, boat speed and heel angle.

In order to compute a single equilibrium point, one must solve as many simultaneous equations as there are degrees-of-freedom; this is the essence of what the program developed in this paper does. If, however, there are more variables than equations to be solved, one discovers that the solution of the equations is a locus of equilibrium points, the particular solution depending upon the values of these additional variables. To illustrate this point, consider the three-degree-of-freedom model mentioned earlier. If in addition to boat speed, leeway angle and heel angle, the model includes the effect of reefing, one can imagine that for a given true wind speed and heading, a sailboat can achieve a new equilibrium for each value of this variable. Clearly, the solution which yields the maximum boat speed on the prescribed heading is an optimum solution for that heading in terms of the additional variable "reefing". Hence, when formulating the procedure described herein, a routine to enable the program to search out the optimum equilibrium in terms of "additional" variables was desired.

A final, but important, aspect of the requirements to be met by the program developed, is that its equilibriums should be computed within certain variable limits. These limits can be of a physical nature or they can be imposed by validity restrictions of one's model. Clearly, if positive "reefing" reduces sail area from its nominal value, one must not allow a solution that involves negative "reefing". This is an example of a physical limit on the variable "reefing". If, however, one's mathematical model produced unreliable values of side force for leeway angles greater than 8 degrees, a limit of 8 degrees on the variable leeway should be imposed for reasons of model validity. In the case of the latter restriction, if the solution sought lies outside the limits imposed, the program should inform the user of this condition and continue to process the next equilibrium.



APPARENT WIND TRIANGLE AND SPEED-MADE-GOOD TO WINDWARD

FIGURE 1

SOLUTION TECHNIQUE FOR COMPUTING EQUILIBRIUMS

There exist several numerical techniques for solving systems of nonlinear equations, all of them iterative in nature. Though each approach has its strengths and weaknesses, the Newton-Raphson minimization technique was chosen because of its speed of convergence.(6) The technique as adopted for use in this performance program is sometimes termed quasi-linearization (5). For small systems of equations, in this case a maximum of six, the Newton-Raphson minimization technique was vastly superior to the other techniques investigated when evaluated in terms of computation time.

The procedure is really a logical extension to systems with more than one variable of the well known Newton's method for finding roots of an equation. In this application the expressions to be minimized are the "errors" or differences between the principle aerodynamic and hydrodynamic forces and moments. Table 1 presents the mathematical formulation of the solution process for an assumed three-degree-of-freedom sailboat model. The elements of the vector $\underline{R}$ are the "errors" to be minimized, in other words, ideally the procedure will produce a vector $\underline{p}$ such that the vector of "errors" ,$\underline{R}$, is identically zero. This is rarely the case, however, so convergence tolerances are set in order that a criterion for determining a satisfactory solution may exist. Hence, if the incremental values calculated for the vector $\underline{p}$ are all less than their prescribed convergence tolerances, the procedure assumes that it has found a satisfactory solution to the system of equations.

The matrix "S" in Table 1 is a matrix of first-order partial derivatives of the vector $\underline{R}$ with respect to the independent variables contained in the vector $\underline{p}$. This can be thought of as a matrix of sensitivies of the vector $\underline{R}$ to changes in each of the independent variables.

In order that this minimization procedure might be used by the performance program, the aforementioned partial derivatives must be evaluated. While it is conceivable that certain of these derivatives exist in a readily available analytic form, clearly this is not the case for others. The programs developed in (1) and (3) made use of the analytically determined partial derivatives where possible, but had to concede to the need for the numerical evaluation of the others. One might argue convincingly that this is the most mathematically exact procedure to follow, but for the sake of computational simplicity, consistency, and efficiency, the author chose to evaluate all partial derivatives by numerical means. The premise that this method leads to increased simplicity and consistancy is easily defended, however the defense of its efficiency is somewhat less obvious. Here attention is drawn to the fact, that quite often the functions used to model a sailboat's forces and moments in terms of the independent variables are trigonometric in nature. Thus, partial differentiation of these functions often leads to increased numbers of trigonometric functions which must be evaluated. Since the expressions for such derivatives are often considerably longer than the functions from which they came, it is quite likely that it would be quicker to evaluate the parent function twice and form the numerical approximation to the desired partial derivative, than it would be to evaluate both the function and its exact partial derivative once, since both are required.

Again, for the sake of simplicity and speed, the forward difference approximation to the first-order derivatives was employed. Though this formulation is less exact than the central difference approximation for derivatives, it requires one less evaluation of the parent function. By choosing an appropriately small change in the value of the independent variable in question, the error in the approximation of the desired partial derivative can be kept within acceptable limits; consequently the program chooses a step size based on the required convergence tolerance for each variable. This procedure has worked very well in practice.

If at any time during the iterative search for a solution, the next approximation to the vector $\underline{p}$ places one or more of its elements outside the bounds of validity for that variable (recall the discussion of limits on the independent variables) the program simply sets that variable to its limit value. Since the solution may still lie within the required limits, this prevents the quasi-linearization of nonlinear functions from forcing the independent variables into undefined regions. For instance, an intermediate approximation to the solution of a sailboat's performance in light air might predict a negative boat speed. Since the hull drag term might well be undefined or wrongly defined for negative boat speed, setting this variable temporarily to a small positive value prevents the program from computing erroneous values of hull drag and its partial derivatives. If, however, a variable remains stuck against one of its limits for more than a specified number of iterations, it is assumed that the solution for that particular equilibrium lies outside the limits of confidence for the model, and the program refrains from any further attempts to seek convergence for that sailing condition.

TABLE 1

NEWTON-RAPHSON MINIMIZATION APPLIED TO
THE SOLUTION OF SAILBOAT THREE-DETREE-OF-FREEDOM EQUILIBRIUM

Equations to be minimized:

$$R_1 = F_{X_A} - F_{X_H}$$

$$R_2 = M_{X_A} - M_{X_H}$$

$$R_3 = F_{Y_A} - F_{Y_H}$$

Define:

$$\vec{P} = [\phi, \lambda, V_B]$$

$$S_{11} = \frac{\partial R_1}{\partial P_1}, \quad S_{12} = \frac{\partial R_1}{\partial P_2}, \quad S_{13} = \frac{\partial R_1}{\partial P_3},$$

$$S_{21} = \frac{\partial R_2}{\partial P_1}, \quad S_{22} = \frac{\partial R_2}{\partial P_2}, \quad S_{23} = \frac{\partial R_2}{\partial P_3},$$

$$S_{31} = \frac{\partial R_3}{\partial P_1}, \quad S_{32} = \frac{\partial R_3}{\partial P_2}, \quad S_{33} = \frac{\partial R_3}{\partial P_3}.$$

Then:

$$\begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{bmatrix} \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \end{bmatrix} = \begin{bmatrix} -R_1 \\ -R_2 \\ -R_3 \end{bmatrix}$$

Therefore:

$$[\delta_1, \delta_2, \delta_3]^T = \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{bmatrix}^{-1} \begin{bmatrix} -R_1 \\ -R_2 \\ -R_3 \end{bmatrix}$$

$$P_1(i+1) = P_1(i) + \delta_1, \quad P_2(i+1) = P_2(i) + \delta_2, \quad P_3(i+1) = P_3(i) + \delta_3.$$

SOLUTION TECHNIQUE FOR PERFORMANCE OPTIMIZATION

Where the solution of equilibrium conditions for a sailboat required that a number of "errors" as functions of several independent variables be minimized, the optimization of a sailboat's performance requires that the boat speed be maximized for a given sailing condition. <2> This requires that all of the first-order partial derivatives of boat speed with respect to the additional independent variables be zero. <3> The author chooses to refer to these additional variables as optimization variables.

This condition of having all the partial derivatives equal to zero is not sufficient to assure that the solution obtained has maximized, rather than minimized, boat speed. In practice, however, if the starting point for the solution process is sufficiently close to the optimum, then the solution determined subsequently will be of the desired nature. In any event, a sequence of equilibriums may be readily checked for a tendency towards decreased boat speed.

The solution technique employed for this optimization is again Newton-Raphson. In this application, however, one needs to use a second-order form of the Newton-Raphson

----

<2> Here sailing condition refers to a given true wind velocity and bearing relative to the direction in which the boat is travelling.

<3> These are variables not required in the determination of equilibrium conditions.

matrix equation. This is depicted in Table 2, and is illustrated graphically in Figure 2. Here the necessity for evaluating the first- and second-order partial derivatives of boat speed by numerical means is obvious. This time however, since the approximations to the second-order derivatives require three or four points for their evaluation, the first-order partial derivatives are approximated using their central difference form with no computational penality. In this manner the order of magnitude in the approximation error is the same for all of the partial derivatives appearing in the matrix equation. The formulation of the partial derivatives required for maximizing boat speed with respect to three "optimization variables" appears at the bottom of Table 2.

At this point, it is interesting to note the number of equilibriums required to evaluate all of the first- and second-order partial derivatives needed for one iteration in the optimization process. If only one "optimization variable" is used, then three equilibrium points are required to evaluate the first- and second-order partial derivatives of boat speed with respect to that variable. This is seen graphically in Figure 3. For two "degrees-of-optimization" nine equilibriums are needed to form all the partial derivatives, but the inclusion of a third optimization variable requires only nineteen, instead of the expected twenty-seven equilibriums. The reason for this is seen in the progression from Figure 4 to Figure 5. Since no partial derivative greater than second-order is ever required, the outermost corners in the three-dimensional space are never used. Beyond the three-dimensional case, graphic means for determining the number of required equilibriums break down, so one must turn to some sort of numerical series representation. It can be shown that the expression relating the required number of equilibriums to the number of optimization variables is of the form illustrated in Figure 6. Clearly the inclusion of more than two or three optimization variables has associated with it a very high computational price. For this reason it was decided to limit the present program to a maximum of three-degrees-of-optimization. It should be made clear however, that this is a restriction imposed by the author, rather than by any inherent shortcomings in the numerical procedure.

The process described thus far consists of an unconstrained optimization procedure, however, as stated earlier it is desirable to be able to place maximum and minimum attainable limits on the independent variables. One method for doing this is the method of sequential unconstrained minimization (optimization) (7). This method requires that a new function for minimization be defined. Typically this function consists of the original function minus a "logarithmic penality function". This later term is adjusted so that the gradients in the vicinity of a variable limit or boundary are such that they force the solution away from the boundary. A linear multiplier in front of this penalty function can then be iteratively reduced so as to allow the solution to approach the boundary (always from the same side), if that is where the optimum constrained solution lies. One of the great strengths of this method is the freedom to choose quite complex variable boundaries. Its greatest weakness from the standpoint of this program, however, is the number of iterations required to obtain an optimum solution. Consequently, a second approach was sought.

The approach adopted is similar to the one used in the equilibrium solution process. After each iteration, all of the optimization variables are checked to determine whether they have exceeded either their maximum or minimum bounds, and for any variables found to be outside their prescribed limits, two steps are taken. The first, as before, is to set the value of that variable to the value of the limit exceeded. The second step is to set an auxillary variable to a value of 1 or -1, depending on whether it was a maximum or minimum bound that was exceeded. After these steps are taken, each variable is checked for convergence.

There are two ways in which each variable can pass this convergence test. The first, and most obvious, is for the variable's value to change less than some prescribed amount.

The second way for a variable to be considered convergent upon its solution value, is for the sign of the first-order partial derivative of boat speed with respect to that variable and the sign of its auxilary varible, to be the same. In this case, the implication is that the true, or unconstrained, optimum solution lies outside the variable limits, and thus the constrained optimum lies on a boundary of the allowable variable space. As before, if all of the variables in the optimization process have not converged then another iteration towards the solution is initiated.

The process just described is equivalent to setting the appropriate element in the vector on the right-hand side of the matrix equation in Table 2 to zero when a variable exceeds one of its limits, but it requires less bookkeeping. This is due to the fact that a variable will occasionally hit against one of its limits during an intermediate step in the iterative solution process, only to reverse its direction in a later step. Thus the latter method would require that during every other iteration, the element in the "forcing" vector be restored to its true value so that any trend back into the interior of the variable space might be detected.

A final two points should be made about the efficiency of the second-order Newton-Raphson optimization process as it has been adopted for use in this program. The first has to do with the very nature of the technique. Since the matrix of "sensitivities" is composed of second-, rather than first-, order partial derivatives, as was the case in the equilibrium solution portion of this program, a convergent solution to the problem of optimization can be found with one iteration, if in the neighborhood of that solution, the change in boat speed with respect to each of the optimization variables can be approximated by a quadratic. This fact usually leads to an extremely low number of iterations required to determine the optimum values for the variables involved. The second point concerns constrained solutions that lie on variable boundaries. With the procedure adopted, an additonal iteration is not required in the case of the solution being reached by a variable exceeding one of its prescribed limits. For example, if the only optimization variable involved was "reefing", as discussed in the example given earlier, and the optimum solution always lay in the region of negative "reefing", then a second iteration would never be required in the determination of the constrained optimum, because the variable would have passed its second convergence check on the first iteration.

TABLE 2

OPTIMIZATION OF BOAT SPEED W.R.T. OPTIMIZATION VARIABLES

For three optimization variables, $\vec{q} = [q_1, q_2, q_3]$, the equation to solved can be written as follows:

$$\begin{bmatrix} V_{11} & V_{12} & V_{13} \\ V_{21} & V_{22} & V_{23} \\ V_{31} & V_{32} & V_{33} \end{bmatrix} \begin{bmatrix} \Delta q_1 \\ \Delta q_2 \\ \Delta q_3 \end{bmatrix} = \begin{bmatrix} -V_1 \\ -V_2 \\ -V_3 \end{bmatrix}$$

Where $V_i = \frac{\partial V_B}{\partial q_i}$ and $V_{ij} = \frac{\partial^2 V_B}{\partial q_i \partial q_j}$

Thus:

$$[\Delta q_1, \Delta q_2, \Delta q_3]^T = \begin{bmatrix} V_{11} & V_{12} & V_{13} \\ V_{21} & V_{22} & V_{23} \\ V_{31} & V_{32} & V_{33} \end{bmatrix}^{-1} \begin{bmatrix} -V_1 \\ -V_2 \\ -V_3 \end{bmatrix}$$

And:

$$q_1(i+1) = q_1(i) + \Delta q_1, \quad q_2(i+1) = q_2(i) + \Delta q_2, \quad q_3(i+1) = q_3(i) + \Delta q_3$$

Using finite differences:

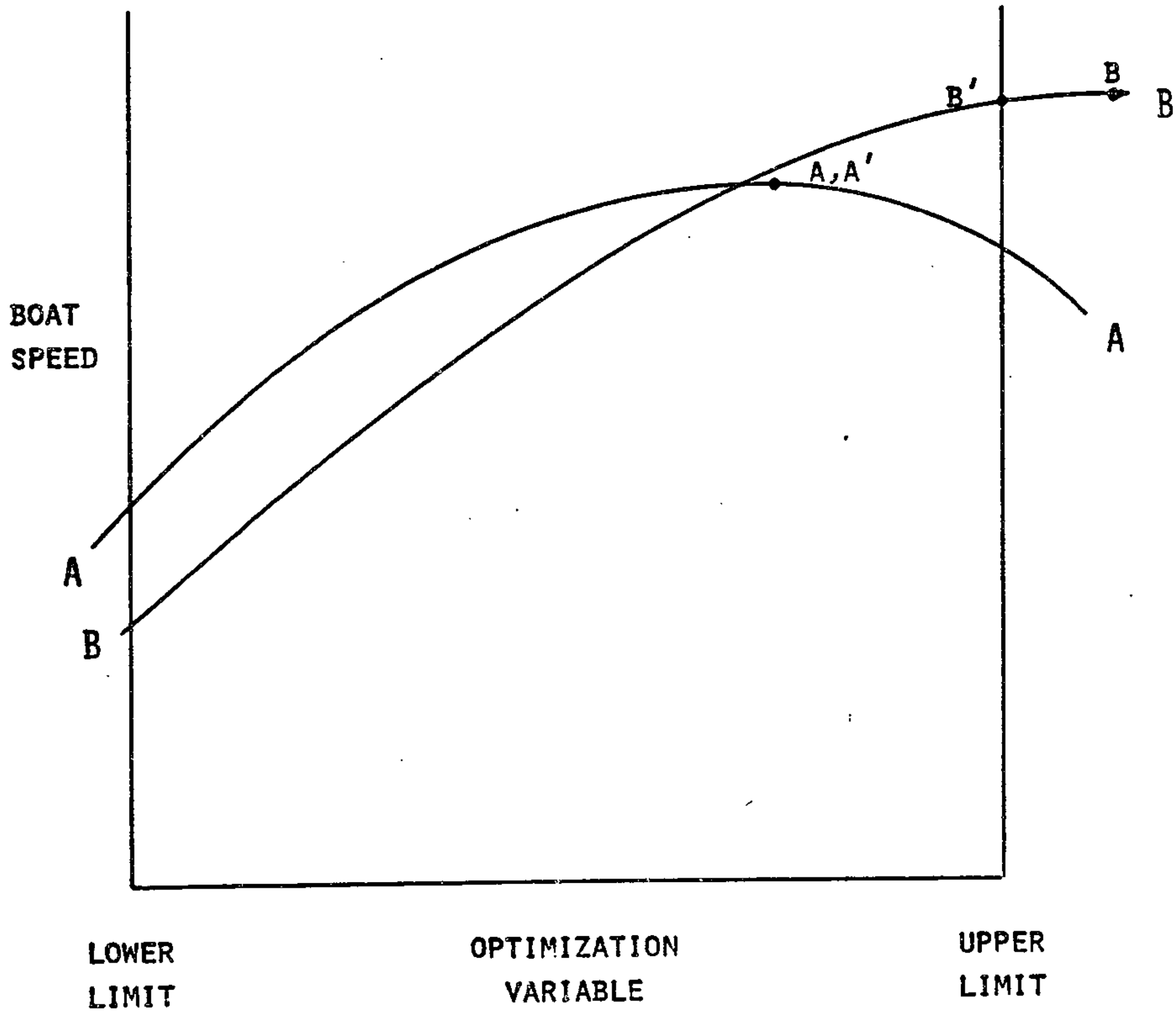
$$V(q_i, q_2, q_3) = V_{i,j,k}$$

$$V_i = (V_{i+1,j,k} - V_{i-1,j,k})/2\Delta_1 + O[(\Delta_1)^2]$$

$$V_{11} = (V_{i+i,j,k} - 2V_{i,j,k} + V_{i-1,j,k} + V_{i-\ ,j,k})/(\Delta_1)^2 + O[(\Delta_1)^2]$$

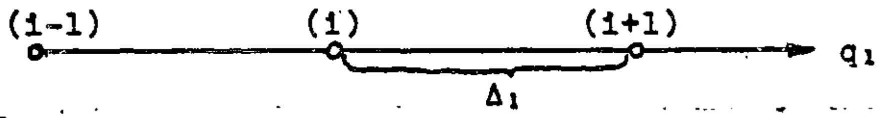
$$V_{12} = (V_{i+1,j+1,k} - V_{i-1,j+1,k} - V_{i+1,j-1,k} + V_{i-1,j-1,k})/4\Delta_1\Delta_2 + O[(\Delta_1+\Delta_2)^2]$$

.
.
.
etc.



POINT (A,A') REPRESENTS BOTH THE CONSTRAINED AND UNCONSTRAINED OPTIMUMS FOR THE LOCUS OF SAILING EQUILIBRIUMS REPRESENTED BY CURVE A. POINTS (B') AND (B) REPRESENT THE CONSTRAINED AND UNCONSTRAINED OPTIMUMS FOR THE LOCUS OF SAILING EQUILIBRIUMS COMPRISING CURVE B.

FIGURE 2

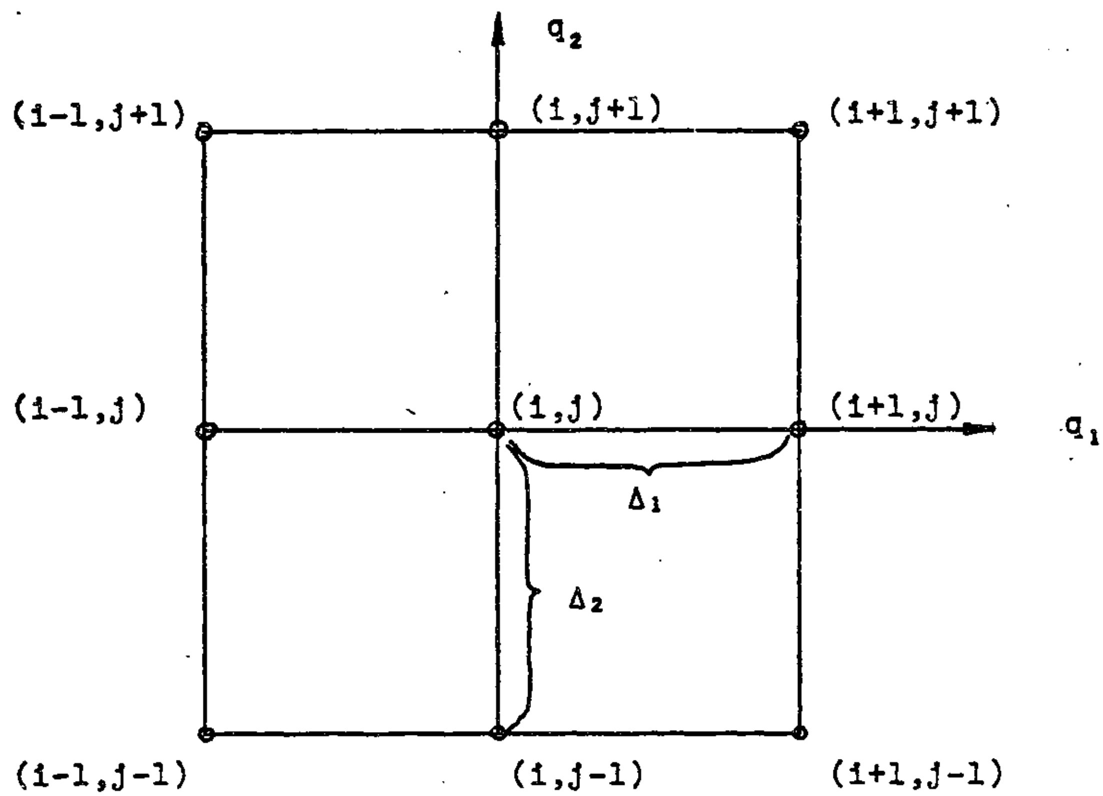


$$\frac{\partial V}{\partial q_1} \doteq (V_{i+1} - V_{i-1})/2\Delta_1$$

$$\frac{\partial^2 V}{\partial q_1^2} \doteq (V_{i+1} - 2V_i + V_{i-1})/(\Delta_1)^2$$

One dimensional derivation of first and second divided differences.

FIGURE 3

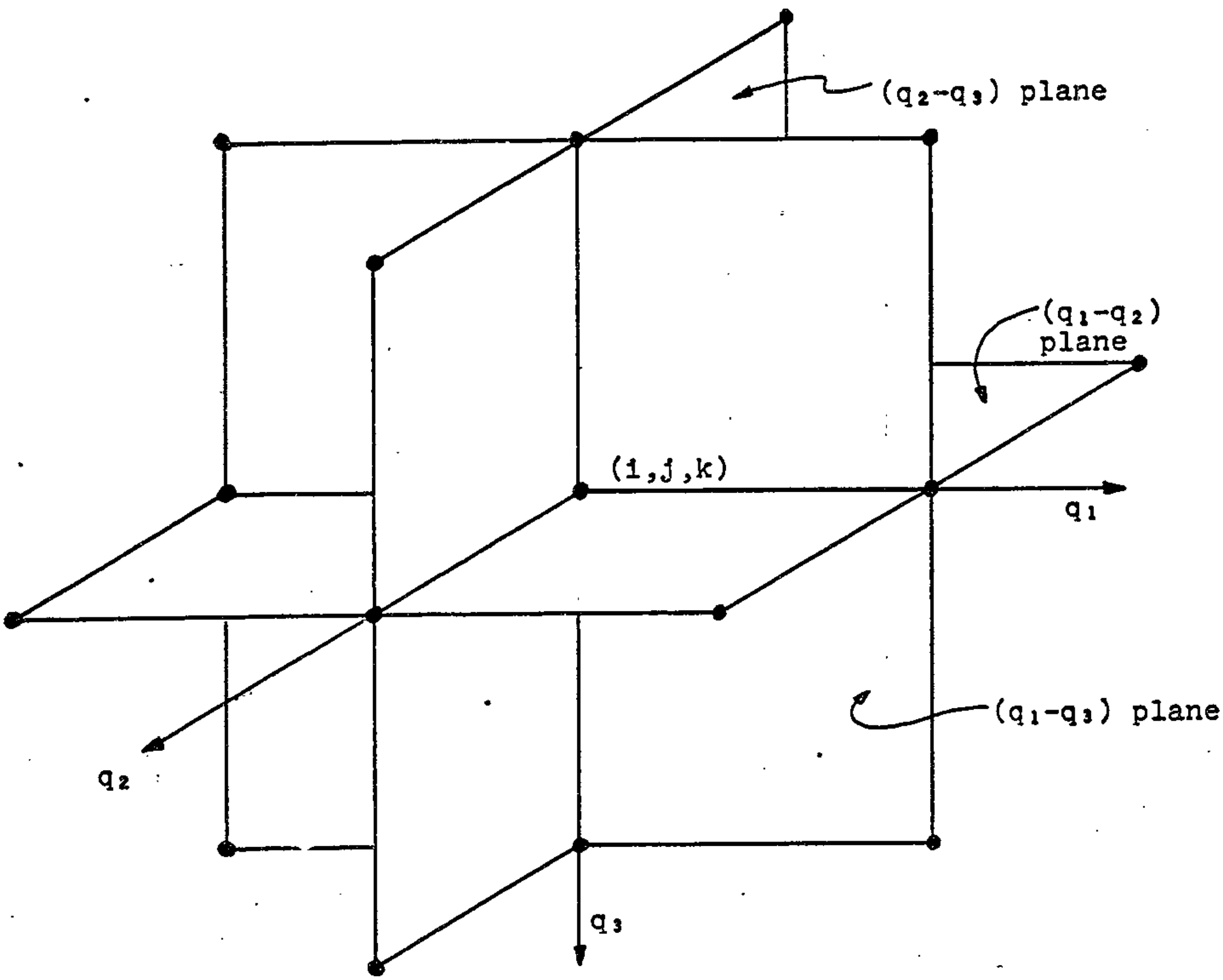


$$\frac{\partial^2 V}{\partial q_1 \partial q_2} \doteq (V_{i+1,j+1} - V_{i-1,j+1} - V_{i+1,j-1} + V_{i-1,j-1})/4\Delta_1\Delta_2$$

Others like the one dimensional case.

Two dimensional derivation of first and second divided differences.

FIGURE 4



$$\frac{\partial V}{\partial q_1} \doteq (V_{i+1,j,k} - V_{i-1,j,k})/2\Delta_1$$

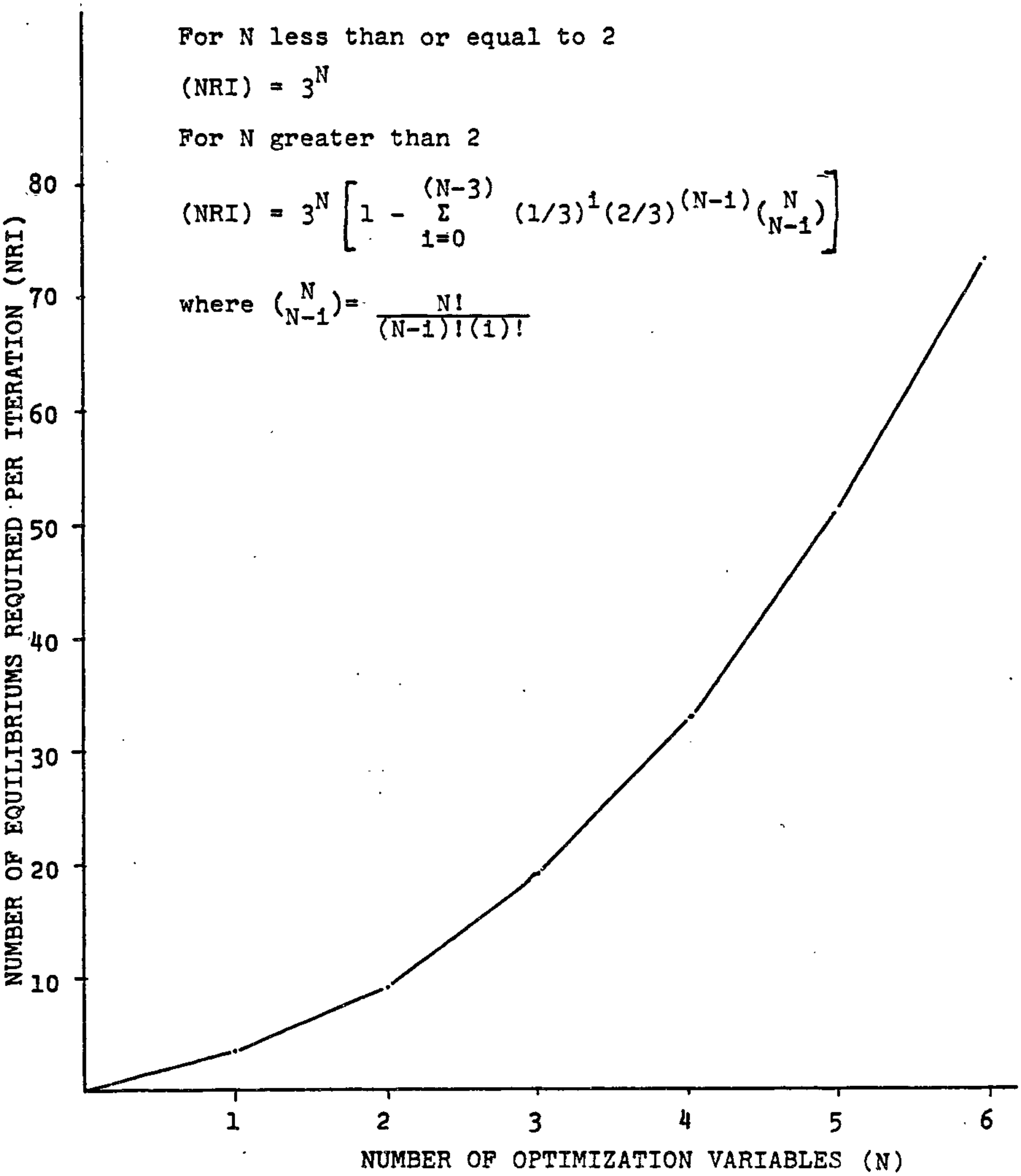
$$\frac{\partial^2 V}{\partial q_1^2} \doteq (V_{i+1,j,k} - 2V_{i,j,k} + V_{i-1,j,k})/(\Delta_1)^2$$

$$\frac{\partial^2 V}{\partial q_1 \partial q_2} \doteq (V_{i+1,j+1,k} - V_{i-1,j+1,k} - V_{i+1,j-1,k} + V_{i-1,j-1,k})/4\Delta_1\Delta_2$$

Other divided differences formed in same manner.

THREE DIMENSIONAL REPRESENTATION OF POINTS REQUIRED TO FORM APPROXIMATIONS TO THE FIRST- AND SECOND-ORDER PARTIAL DERIVATIVES.

FIGURE 5



Equilibriums required per iteration vs. number of optimization variables.

FIGURE 6

PROGRAMMING CONSIDERATIONS

Figure 7 contains a general schematic diagram of the performance program's structure. The blocks labeled PERFORM(MAIN), OPTIMIZE and MINIMIZE, are the three procedures that comprise the computational core of the program. The remaining block, PMODL, represents the user supplied procedure which contains the mathematical model to be used. <4>

Apart from the numerical considerations already discussed, certain operational guidelines were established while writing the program. First among these was the choice of a programming language. PL/1 was chosen for a number of reasons. A primary consideration was the need to handle considerable amounts of output formating. Because the program was designed to accept mathematical models with between two- and six-degrees-of-freedom, and to then optimize the performance equilibriums with respect to as many as three additional variables, the format of the printed output had to possess considerable flexibility. PL/1's many data types, and particularly its capability to manipulate string variables, made it well suited to handle the complex formating required. An equally important consideration was the efficient manner in which PL/1 handles

----

<4> A detailed description of the form requirements placed on PMODL by the main program appears in Appendix B.

array operations. The very nature of the solution process requires that operations such as matrix inversion and multiplication, not to mention element assignment, be executed repeatedly. The reason for PL/1's superiority over FORTRAN in this type of operation lies in the different manner in which the two compilers assign array variable addresses in core. A final, but somewhat less important reason for choosing PL/1, was its ability to allow the programmer to allocate variables at execution time. This feature was used to set array and vector dimensions at the time of execution, after their size requirements had been determined. In this manner, the program could readily adjust its solution procedure so as to exactly accomodate the size requirements of the particular mathematical model being used.

A second guideline set down for the program, was that it should be compatible with operations in both batch and time-sharing environments. In part, this requirement lead to the file structure chosen. Table 3 shows the seven files used by this program, and gives a brief statement concerning their contents.

As indicated, the input to the program was divided into three sections or files, in order that certain data which might be used repeatedly could be stored separately, (for instance on a magnetic disc), from the data that changed with each running of the program. Thus the coefficient values required by the mathematical model, and the table of desired sailing conditions, are kept in separate files from the "execution time options" which are entered in the standard input file, SYSIN.

Four files were allocated to program output. SYSPRINT, the standard PL/1 output file, is the "print" file which contains the majority of the program's output; this is a printed record of information concerning the models used, the execution time options employed and the optimized equilibriums computed. In addition, for the version currently running at M.I.T., a page is included that contains certain statistics to aid in the evaluation of the program's computational efficiency.

The second output file, TERM, is a file solely devoted to displaying input prompts at the time-sharing terminal (if the program is being run under TSO). These prompts pertain only to data entered via SYSIN. When the program is run in a batch mode, the file TERM is given a DUMMY assignment and hence no output operations to this file are performed.

File BUGS, as the name implies, contains any error or diagnostic messages generated by the program. Errors pertaining to the matrix inversion routine, as well as those related to a lack of convergence in either of the iterative procedures, are displayed in this file. <5>

The last output file, pertains to a third guideline set for the program. In order to remain as universally useful as possible, without carrying a lot of unnecessary programming overhead, it was decided not to try and second guess all of the secondary uses for the data generated by this program. Some examples of common secondary calculations performed by other more specialized programs (1),(2),(3) are: maximum (and minimum) speed made good to windward, tabulation of forces and moments acting on the rig or hull, and rating increases (or decreases) needed to sail at a speed assumed by a given rating rule relative to another "base boat". The decison was made that these or any other calculations desired, would be performed by user supplied auxillary programs with the file AUXOUT as input. The data in this file, like all of the input files used by the program, is in a free format. <6> This is accomplished by using PL/1's "list" directed input/output options. Consequently, most data items are separated by blanks, and/or commas, while string variables are additionally bracketted by single quotes. Basically, the contents of the file AUXOUT is the same as that of the file SYSPRINT, but where the latter is a "print" file, the data in AUXOUT would more likely be directed to punched cards or perhaps an on-line storage device. A final point in defense of the decision to use

----

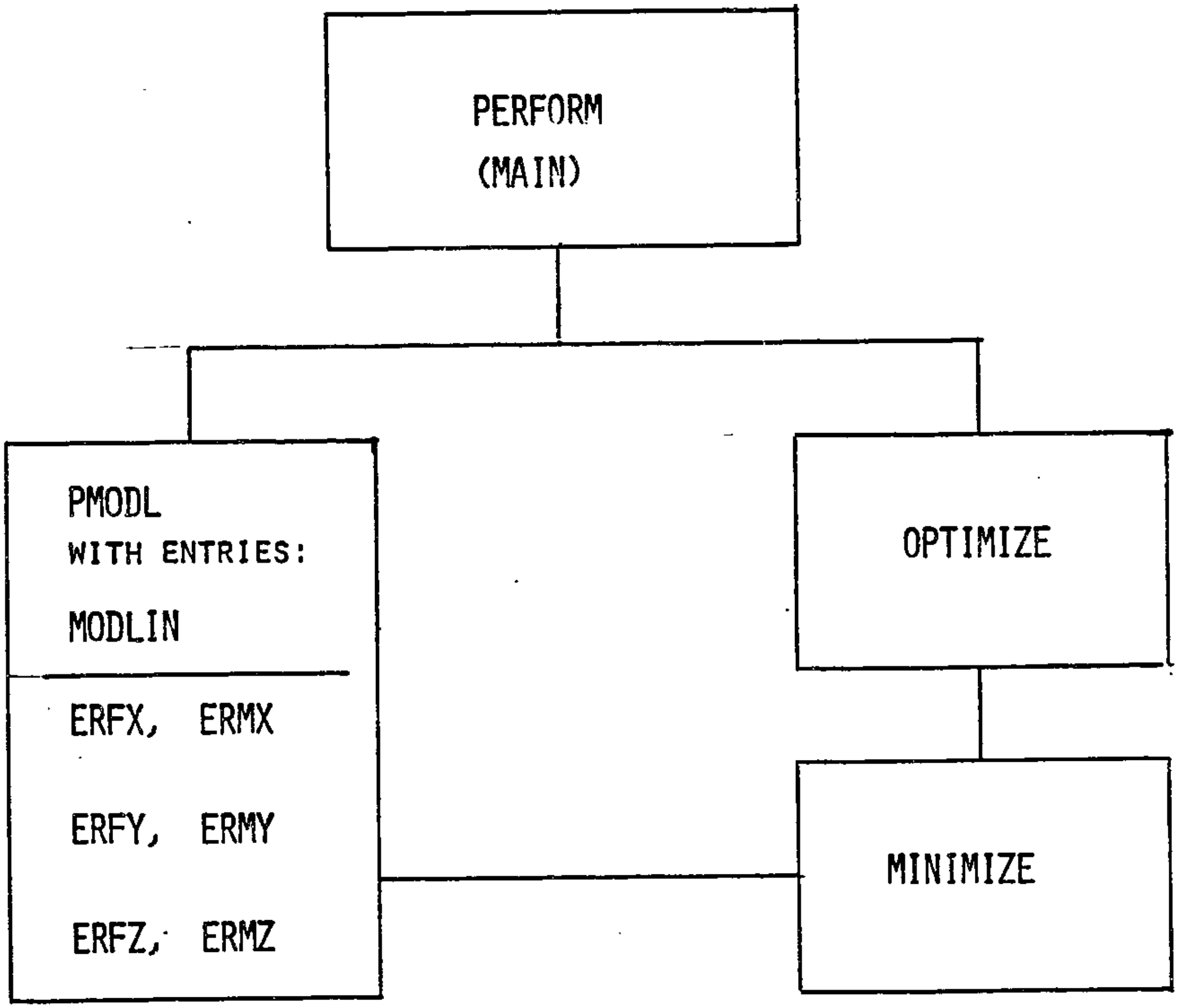
<5> Certain of these messages appear in abbreviated form to the right of the other variables displayed for each sailing condition in the file SYSPRINT. See Appendix A.

<6> By using this form of I/O any machine dependence is eliminated. Such files are said to be "stream" oriented, rather than "record" oriented.

AUXOUT as input to user supplied specialized programs, rather than including a number of these procedures in the main program, is the variety of plotting routines encountered as one goes from one computation center to another. Since in all likelyhood, some of the data computed by a program such as this will be most conveniently viewed in a graphical form, it would be short-sighted to assume that plotting procedures included for use at M.I.T. would be of use elsewhere.

A final decision that was made before programming began, concerned the variables to be associated with each of the degrees-of-freedom. These pairings are indicated at the bottom of Table 3. This predetermining of the order of inclusion for the six independent variables might be considered somewhat restrictive, but it was chosen to comply with known existing models. <7> It was felt that any restrictions imposed by this preset ordering were more than compensated for by the decrease in programming complexity, and hence computation time, that could be achieved.

<7> The exception to this was the relative placement of the last two variables. Since there was no precedent here, the author made an arbitrary decision concerning the variables and their order.



SCHEMATIC LAYOUT OF THE PERFORMANCE PREDICTION PROGRAM

FIGURE 7

TABLE 3

A. PROGRAM'S FILE STRUCTURE

INPUT			OUTPUT		
1.	SYSIN -	Title, execution time options	1.	SYSPRINT -	Standard print file
2.	WINDY -	Matrix of sailing conditions, i.e., true wind speeds and directions	2.	TERM -	TSO prompts for file SYSIN
			3.	BUGS -	Errors and diagnostic messages
3.	COEFF -	Coefficients used by the user supplied routine, PMODL	4.	AUXOUT -	Stream file like SYSPRINT, for use with auxillary programs

B. VARIABLES TO BE ASSOCIATED WITH EACH DEGREE-OF-FREEDOM

1. V_B - Boat speed - forces in x-direction
2. ϕ - Heel angle - moments about the x-axis
3. λ - Leeway (sideslip) angle - forces in y-direction
4. S_R - Rudder trim angle - moment about z-axis
5. Δz - Sinkage - Forces in z-direction
6. θ - Trim angle - moment about y-axis

DISCUSSION

To date, the program that has been discussed herein, has been used to predict the performance of three sailboats using two different four-degrees-of-freedom models. The printout for these runs has been included in Appendix A.

The first run was of a more or less conventional nature. The model used for the two boats in this test run had four-degrees-of-freedom, and was designed to roughly approximate the experimentally determined forces and moments for the yacht "Gincrack". (4) A single degree-of-optimization with respect to the variable "reef" was employed. In this case "reef" was defined in the same manner as the linear reefing function described in (1), with unity indicating no reefing and zero indicating total reefing, or no sail. All velocities and angles were in feet per second and degrees respectively. The procedure, PMODL, used for this run is reproduced in Appendix B, while the contents of the input files WINDY, COEFF and SYSIN may be found in Appendix C.

There are several things to note when viewing the output from this run. The first is the feature in the program that allows one to make use of the "geosim" principle to predict the performance of similar yachts of different sizes. In this case, two boats had their performance polars calculated using the same geosim model; one with a characteristic length of 23.8 feet and the other

25 feet. A second point of interest is the page summarizing the computation statistics. As indicated, the average time required to compute an equilibrium point was barely in excess of four hundredths of a second. This compares quite favorably with the program originally employing the mathematical model used (4), and its required five plus hundredths of a second per equilibrium. By comparing the total number of equilibriums computed with the number of optimized equilibriums one finds that on the average it required less than two iterations to find the optimum solution for a given sailing condition. This strongly supports the statements made earlier in the discussion on techniques for determining optimized equilibriums.

The second run of the performance perdiction program again utilized a four-degree-of-freedom model, in fact the hydrodynamic portion of the model was identical to that used in the first run. What made this second run unique was the aerodynamic model employed; polynomial approximations to a sailwing's aerodynamic characteristics.(9) The angle of attack of the wing, alpha, was the independent variable in the polynomial approximations, and so it was chosen as the optimization variable for the run. To the author's knowledge a system of this nature has never before been investigated in such a manner. Unlike the previous run, the optimum solution for this system nearly always occurred with the optimization variable between its upper and lower limits.

Here the number of iterations required to determine an optimum solution was somewhat higher than before. In addition, it is particularly interesting to note that once during each loop through the file of true wind headings, two adjacent optimized equilibruims had angles of attack differing by approximately fifty-five degrees, thus showing the power of the second-order Newton-Raphson technique and its ability to converge to a solution even after passing through a nearly discontinuous jump like the one indicated. Physically this jump corresponds to an abrupt change from lift- to drag-aerodynamics for the sailwing. The procedure, PMODL, and the input deck for this run are reproduced in Appendices B and C respectively.

CLOSING REMARKS

By all appearances, the program described in this paper has met each of the objectives set forth at the introduction. Despite the generality and application flexibility achieved by the program, the constant attention paid to computational efficiency throughout the program's development has yielded a procedure which is as fast, or faster, computationally than the more specialized programs described in (1) and (4). All of these programs were run on the same IBM 370-168 at M.I.T., so machine speed is not a factor in determining the time required to compute a performance equilibrium. Thus, the program which has been described herein, has escaped one of the greatest pitfalls common to most "general" programs.

In conclusion, it is the author's opinion that the generality of the program developed and the ease with which the mathematical model describing the sailboat can be altered, makes this a potentially powerful investigative tool for the yacht designer and researcher alike.

REFERENCES

1. Kerwin, J.E.; Preliminary Report on the Velocity Prediction Program, M.I.T., April 1975.

2. Myers, H.A.; "Theory of Sailing Applied to Ocean Racing Yachts", S.N.A.M.E., September 1974.

3. Letcher, J.S.; "Optimum Windward Performance of Sailing Craft", Advanced copy, April 1974.

4. Munger, E.C.; "Program Sailboat, a 13.71 Project", M.I.T., December 1974.

5. Windgrove, R.C.; "Quasi-linearization Technique for Estimating Aircraft States from Flight Data", Journal of Aircraft, Vol. 10, No. 5, May 1973.

6. Carnahan, B.; Luther, H.; Wilkes, J.; "Applied Numerical Methods", John Wiley & Sons, Inc., 1968.

7. Fiaco, A. and McCormick, G.; "Nonlinear Programming: Sequential Unconstrained Minimization Techniques", John Wiley & Sons, Inc., 1968.

8. Marchaj, C.A.; "Sailing Theory and Practice", Dodd, Mead & Co., 1964.

9. Mercer, J.E.; "Method for Determining the Spanwise Lift Distribution on a Wing from Bending Moment Data", Princeton Univ., Sailwing Technology, Vol. 4, 1970.

APPENDIX A

The following are reductions of the computer output produced for the two test runs described in this paper. They are presented here in the order in which they were discussed, namely the run with the more conventional four-degree-of-freedom model first, followed by the run used to investigate the effectiveness of a sailwing in lew of a more conventional rig.

SAMPLE FOUR DEGREE OF FREEDOM MODEL WITH ONE DEGREE OF OPTIMIZATION

MODEL INFORMATION

HYDRODYNAMIC MODEL: GIF MUNGER'S GIMCRACK HYDRO

THE FOLLOWING IS A LIST OF COEFFICIENTS, AND THEIR VALUES, USED BY THIS MODEL:

C1	C2	C3	C4	C5	CXR
8.6000E-04	3.4000E-03	8.4000E-04	7.1000E-02	1.7800E-02	3.4000E-01

AERODYNAMIC MODEL: GIF MUNGER'S GIMCRACK AERO

THE FOLLOWING IS A LIST OF COEFFICIENTS, AND THEIR VALUES, USED BY THIS MODEL:

CSA	CHCE
7.5000E-01	7.0000E-01

SAMPLE FOUR DEGREE OF FREEDOM MODEL WITH ONE DEGREE OF OPTIMIZATION

EXECUTION TIME OPTIONS

NUMBER OF BOATS = 2
WITH CHARACTERISTIC LENGTHS OF: 23.80 25.00

DEGREES OF FREEDOM = 4
THE INDEPENDENT VARIABLES ARE: VB, PHI, LAMDA, RUDDER

THE MAXIMUM AND MINIMUM VALUES ALLOWED FOR THESE VARIABLES ARE:
1.2000E+01 4.0000E+01 1.2000E+01 3.5000E+01
5.0000E-01 -1.0000E+01 -1.2000E+01 -3.5000E+01

THE REQUIRED CONVERGENCE TOLERENCES ARE:
1.0000E-03 1.0000E-01 1.0000E-01 1.0000E-01

THE VARIABLE TO BE OPTIMIZED IS: REEF

THE MAXIMUM AND MINIMUM VALUES ALLOWED FOR THE VARIABLE(S) ARE:
1.0000E+00
3.3000E-01

THE REQUIRED CONVERGENCE TOLERENCES ARE:
1.0000E-03

MAXIMUM ITERATIONS ALLOWABLE FOR CONVERGENCE IS 10

TABLE OF TRUE WIND VELOCITIES

1.0000E+01 2.0000E+01 3.0000E+01 4.0000E+01

TABLE OF TRUE WIND HEADINGS

1.8000E+02 1.6000E+02 1.4000E+02 1.2000E+02 1.0000E+02 8.0000E+01 6.0000E+01 4.0000E+01 3.5000E+01 3.0000E+01
2.5000E+01 2.0000E+01

VB/VT FOR START-UP IS 3.80000E-01

THE OPTIMIZATION VARIABLE(S) ARE SET TO THE FOLLOWING VALUES AT START-UP:

1.0000E+00

SAMPLE FOUR DEGREE OF FREEDOM MODEL WITH ONE DEGREE OF OPTIMIZATION

CHARACTERISTIC LENGTH = 23.800

VT	GAMMA	VA	BETA	VMG	VB	PHI	LAMDA	RUDDER	REEF
10.0	180.0	6.0	180.0	-4.0	4.026	0.000	0.000	0.000	1.000
10.0	160.0	6.2	146.5	-4.0	4.228	0.176	0.126	0.017	1.000
10.0	140.0	7.1	115.6	-3.5	4.584	0.703	0.417	0.088	1.000
10.0	120.0	8.7	90.3	-2.5	4.961	1.528	0.743	0.238	1.000
10.0	100.0	10.5	70.4	-0.9	5.247	2.464	1.025	0.446	1.000
10.0	80.0	12.1	54.2	0.9	5.374	3.217	1.241	0.619	1.000
10.0	60.0	13.4	40.1	2.6	5.290	3.514	1.402	0.650	1.000
10.0	40.0	14.1	27.1	3.8	4.907	3.212	1.549	0.492	1.000
10.0	35.0	14.1	23.9	3.9	4.740	3.042	1.595	0.428	1.000
10.0	30.0	14.1	20.8	3.9	4.528	2.835	1.656	0.357	1.000
10.0	25.0	14.0	17.6	3.9	4.252	2.592	1.747	0.281	1.000
10.0	20.0	13.7	14.5	3.6	3.872	2.309	1.912	0.202	1.000
20.0	180.0	14.2	180.0	-5.8	5.778	-0.000	0.000	0.000	1.000
20.0	160.0	14.5	151.9	-5.6	5.993	0.696	0.239	0.181	1.000
20.0	140.0	15.7	125.0	-4.9	6.334	2.684	0.646	0.800	1.000
20.0	120.0	17.6	100.9	-3.3	6.673	5.602	1.038	1.875	1.000
20.0	100.0	20.0	80.1	-1.2	6.906	8.733	1.281	3.129	1.000
20.0	80.0	22.3	62.1	1.2	6.970	11.123	1.452	4.026	1.000
20.0	60.0	24.1	45.8	3.4	6.824	11.953	1.688	4.083	1.000
20.0	40.0	25.2	30.7	4.9	6.369	10.847	2.106	3.129	1.000
20.0	35.0	25.3	27.0	5.1	6.174	10.274	2.258	2.752	1.000
20.0	30.0	25.3	23.3	5.1	5.927	9.592	2.443	2.332	1.000
20.0	25.0	25.2	19.6	5.1	5.605	8.804	2.687	1.878	1.000
20.0	20.0	24.9	15.9	4.8	5.159	7.908	3.057	1.398	1.000
30.0	180.0	22.9	180.0	-7.1	7.053	0.000	0.000	0.000	1.000
30.0	160.0	23.3	153.9	-6.8	7.282	1.560	0.207	0.690	1.000
30.0	140.0	24.7	128.6	-5.8	7.603	5.942	0.519	2.815	1.000
30.0	120.0	26.9	105.4	-3.9	7.846	12.173	0.581	5.976	1.000
30.0	100.0	29.7	84.8	-1.4	7.898	18.560	0.449	9.053	1.000
30.0	80.0	32.3	66.3	1.3	7.773	23.150	0.418	10.837	1.000
30.0	60.0	34.4	49.1	3.8	7.544	23.291	0.910	9.925	0.981
30.0	40.0	35.8	32.6	5.5	7.122	22.232	1.581	8.325	1.000
30.0	35.0	35.9	28.6	5.7	6.947	21.089	1.902	7.415	1.000
30.0	30.0	36.0	24.6	5.8	6.716	19.735	2.288	6.386	1.000
30.0	25.0	35.9	20.7	5.8	6.403	18.179	2.771	5.249	1.000
30.0	20.0	35.7	16.7	5.6	5.952	16.417	3.440	4.012	1.000
40.0	180.0	31.9	180.0	-8.1	8.091	0.000	0.000	0.000	1.000
40.0	160.0	32.3	154.9	-7.8	8.330	2.768	0.066	1.748	1.000
40.0	140.0	33.9	130.6	-6.6	8.579	10.455	-0.112	6.622	1.000
40.0	120.0	36.5	108.3	-4.3	8.534	21.050	-0.714	12.778	1.000
40.0	100.0	39.4	88.0	-1.4	8.314	23.188	0.114	11.695	0.901
40.0	80.0	42.2	69.1	1.4	8.110	23.301	0.863	10.292	0.838
40.0	60.0	44.5	51.2	3.9	7.862	23.303	1.389	9.341	0.823
40.0	40.0	45.9	34.0	5.7	7.430	23.211	1.936	8.337	0.853
40.0	35.0	46.1	29.8	5.9	7.259	23.176	2.139	7.987	0.869

SAMPLE FOUR DEGREE OF FREEDOM MODEL WITH ONE DEGREE OF OPTIMIZATION

CHARACTERISTIC LENGTH = 23.800

VT	GAMMA	VA	BETA	VMG	VB	PHI	LAMDA	RUDDER	REEF
40.0	30.0	46.2	25.6	6.1	7.039	23.162	2.409	7.561	0.889
40.0	25.0	46.2	21.5	6.1	6.745	23.148	2.804	6.998	0.914
40.0	20.0	46.0	17.3	5.9	6.324	23.206	3.450	6.241	0.945

SAMPLE FOUR DEGREE OF FREEDOM MODEL WITH ONE DEGREE OF OPTIMIZATION

CHARACTERISTIC LENGTH = 25.000

VT	GAMMA	VA	BETA	VMG	VB	PHI	LAMDA	RUDDER	REEF
10.0	180.0	5.9	180.0	-4.1	4.058	0.000	-0.000	0.000	1.000
10.0	160.0	6.2	146.3	-4.0	4.264	0.168	0.131	0.017	1.000
10.0	140.0	7.1	115.3	-3.5	4.627	0.671	0.432	0.086	1.000
10.0	120.0	8.7	89.9	-2.5	5.011	1.459	0.771	0.234	1.000
10.0	100.0	10.5	70.1	-0.9	5.303	2.355	1.066	0.440	1.000
10.0	80.0	12.2	53.9	0.9	5.433	3.076	1.292	0.612	1.000
10.0	60.0	13.5	39.9	2.7	5.348	3.361	1.460	0.643	1.000
10.0	40.0	14.2	27.0	3.8	4.960	3.073	1.610	0.487	1.000
10.0	35.0	14.2	23.8	3.9	4.791	2.910	1.657	0.423	1.000
10.0	30.0	14.1	20.7	4.0	4.576	2.713	1.719	0.352	1.000
10.0	25.0	14.0	17.6	3.9	4.297	2.480	1.813	0.277	1.000
10.0	20.0	13.7	14.4	3.7	3.911	2.208	1.983	0.199	1.000
20.0	180.0	14.2	180.0	-5.8	5.829	-0.000	0.000	0.000	1.000
20.0	160.0	14.5	151.8	-5.7	6.048	0.663	0.218	0.177	1.000
20.0	140.0	15.7	124.8	-4.9	6.394	2.557	0.679	0.784	1.000
20.0	120.0	17.6	100.7	-3.4	6.742	5.342	1.099	1.847	1.000
20.0	100.0	20.0	79.9	-1.2	6.983	8.335	1.369	3.095	1.000
20.0	80.0	22.3	61.9	1.2	7.053	10.625	1.558	3.994	1.000
20.0	60.0	24.2	45.7	3.5	6.908	11.424	1.805	4.056	1.000
20.0	40.0	25.3	30.6	4.9	6.445	10.368	2.228	3.107	1.000
20.0	35.0	25.4	26.9	5.1	6.247	9.820	2.380	2.730	1.000
20.0	30.0	25.4	23.2	5.2	5.996	9.168	2.567	2.312	1.000
20.0	25.0	25.3	19.6	5.1	5.668	8.414	2.815	1.860	1.000
20.0	20.0	25.0	15.9	4.9	5.215	7.556	3.192	1.383	1.000
30.0	180.0	22.9	180.0	-7.1	7.117	0.000	0.000	0.000	1.000
30.0	160.0	23.2	153.8	-6.9	7.350	1.486	0.223	0.675	1.000
30.0	140.0	24.6	128.4	-5.9	7.679	5.660	0.574	2.768	1.000
30.0	120.0	26.9	105.2	-4.0	7.937	11.608	0.681	5.914	1.000
30.0	100.0	29.7	84.6	-1.4	8.011	17.724	0.577	9.020	1.000
30.0	80.0	32.3	66.1	1.4	7.903	22.141	0.560	10.853	1.000
30.0	60.0	34.5	48.9	3.8	7.673	23.283	0.921	10.563	0.996
30.0	40.0	35.8	32.5	5.5	7.239	21.278	1.748	8.339	1.000
30.0	35.0	36.0	28.5	5.8	7.056	20.179	2.073	7.419	1.000
30.0	30.0	36.1	24.6	5.9	6.817	18.878	2.465	6.382	1.000
30.0	25.0	36.0	20.6	5.9	6.494	17.384	2.956	5.237	1.000
30.0	20.0	35.7	16.7	5.7	6.031	15.693	3.639	3.995	1.000
40.0	180.0	31.8	180.0	-8.2	8.167	0.000	0.000	0.000	1.000
40.0	160.0	32.2	154.9	-7.9	8.409	2.636	0.091	1.713	1.000
40.0	140.0	33.8	130.5	-6.6	8.676	9.961	-0.030	6.529	1.000
40.0	120.0	36.4	108.1	-4.3	8.670	20.090	-0.589	12.716	1.000
40.0	100.0	39.4	87.8	-1.5	8.454	23.183	0.079	12.448	0.915
40.0	80.0	42.2	68.9	1.4	8.249	23.299	0.868	10.961	0.851
40.0	60.0	44.5	51.1	4.0	7.997	23.302	1.424	9.949	0.836
40.0	40.0	46.0	33.9	5.8	7.557	23.210	2.004	8.877	0.866
40.0	35.0	46.2	29.7	6.0	7.383	23.175	2.219	8.504	0.882

SAMPLE FOUR DEGREE OF FREEDOM MODEL WITH ONE DEGREE OF OPTIMIZATION

CHARACTERISTIC LENGTH = 25.000

VT	GAMMA	VA	BETA	VMG	VB	PHI	LAMDA	RUDDER	REEF
40.0	30.0	46.3	25.6	6.2	7.159	23.161	2.505	8.049	0.903
40.0	25.0	46.3	21.4	6.2	6.859	23.148	2.923	7.449	0.928
40.0	20.0	46.1	17.3	6.0	6.430	23.205	3.606	6.641	0.960

SAMPLE FOUR DEGREE OF FREEDOM MODEL WITH ONE DEGREE OF OPTIMIZATION

COMPUTATION STATISTICS

THE TOTAL NUMBER OF EQUILIBRIUMS COMPUTED WAS 417, OF WHICH 96 WERE OPTIMUM.
TOTAL CPU TIME SPENT COMPUTING EQUILIBRIUMS WAS 1760 HUNDRETHS OF A SECOND.
THE AVERAGE TIME REQUIRED TO COMPUTE A SINGLE EQUILIBRIUM WAS 4.220623E+00 HUNDRETHS OF A SECOND.

A SAILWING USED AS PROPULSION FOR A SAILBOAT

MODEL INFORMATION

HYDRODYNAMIC MODEL: GIF MUNGER'S GIMCRACK HYDRO

THE FOLLOWING IS A LIST OF COEFFICIENTS, AND THEIR VALUES, USED BY THIS MODEL:

C1	C2	C3	C4	C5	CXR
8.6000E-04	3.4000E-03	8.4000E-04	7.1000E-02	1.7800E-02	3.4000E-01

AERODYNAMIC MODEL: SAILWING POLYNOMIAL AERO (PRINCETON)

THE FOLLOWING IS A LIST OF COEFFICIENTS, AND THEIR VALUES, USED BY THIS MODEL:

CSA	CHCE
7.5000E-01	7.0000E-01

A SAILWING USED AS PROPULSION FOR A SAILBOAT

EXECUTION TIME OPTIONS

```
NUMBER OF BOATS =          1
WITH CHARACTERISTIC LENGTHS OF:    23.80

DEGREES OF FREEDOM =          4
THE INDEPENDENT VARIABLES ARE: VB, PHI, LAMDA, RUDDER

THE MAXIMUM AND MINIMUM VALUES ALLOWED FOR THESE VARIABLES ARE:
   1.2000E+01   4.0000E+01   1.2000E+01   3.5000E+01
   5.0000E-01  -4.0000E+01  -1.2000E+01  -3.5000E+01

THE REQUIRED CONVERGENCE TOLERENCES ARE:
   1.0000E-03   1.0000E-01   1.0000E-01   1.0000E-01


THE VARIABLE TO BE OPTIMIZED IS: ALPHA

THE MAXIMUM AND MINIMUM VALUES ALLOWED FOR THE VARIABLE(S) ARE:
   9.0000E+01
  -1.0000E+01

THE REQUIRED CONVERGENCE TOLERENCES ARE:
   1.0000E-01

MAXIMUM ITERATIONS ALLOWABLE FOR CONVERGENCE IS        10
```

```
TABLE OF TRUE WIND VELOCITIES

   1.0000E+01   2.0000E+01   3.0000E+01   4.0000E+01

TABLE OF TRUE WIND HEADINGS

   2.0000E+01   2.5000E+01   3.0000E+01   3.5000E+01   4.0000E+01   5.0000E+01   6.0000E+01   7.0000E+01   8.0000E+01   9.0000E+01

   1.0000E+02   1.1000E+02   1.2000E+02   1.3000E+02   1.4000E+02   1.5000E+02   1.5500E+02   1.6000E+02   1.6500E+02   1.7000E+02

   1.7500E+02   1.8000E+02
```

```
VB/VT FOR START-UP IS  3.80000E-01

THE OPTIMIZATION VARIABLE(S) ARE SET TO THE FOLLOWING VALUES AT START-UP:

   0.0000E+00
```

A SAILWING USED AS PROPULSION FOR A SAILBOAT

CHARACTERISTIC LENGTH = 23.800

VT	GAMMA	VA	BETA	VMG	VB	PHI	LAMDA	RUDDER	ALPHA	
10.0	20.0	14.0	14.2	3.9	4.150	5.995	3.974	0.682	6.634	
10.0	25.0	14.2	17.3	4.1	4.473	6.291	3.519	0.820	7.620	
10.0	30.0	14.3	20.5	4.1	4.726	6.422	3.163	0.933	8.349	
10.0	35.0	14.3	23.6	4.0	4.931	6.424	2.865	1.020	8.915	
10.0	40.0	14.3	26.7	3.9	5.100	6.324	2.607	1.081	9.376	
10.0	50.0	14.1	33.0	3.4	5.353	5.885	2.169	1.124	10.089	
10.0	60.0	13.6	39.5	2.8	5.520	5.211	1.799	1.072	10.632	
10.0	70.0	13.0	46.1	1.9	5.614	4.386	1.473	0.942	11.072	
10.0	80.0	12.3	53.2	1.0	5.644	3.484	1.175	0.760	11.449	
10.0	90.0	11.5	60.7	0.0	5.616	2.573	0.896	0.556	11.786	
10.0	100.0	10.6	68.9	-1.0	5.535	1.715	0.632	0.357	12.104	
10.0	110.0	9.6	78.1	-1.8	5.405	0.956	0.381	0.187	12.420	
10.0	120.0	8.7	88.5	-2.6	5.231	0.325	0.143	0.058	12.757	
10.0	130.0	7.8	100.5	-3.2	5.012	-0.165	-0.081	-0.026	13.336	
10.0	140.0	7.1	114.4	-3.6	4.749	-0.535	-0.294	-0.073	13.886	
10.0	150.0	6.5	130.1	-3.9	4.457	-0.805	-0.507	-0.093	14.766	
10.0	155.0	6.4	138.4	-3.9	4.298	-0.911	-0.621	-0.096	15.494	
10.0	160.0	6.3	147.0	-3.9	4.128	-0.996	-0.740	-0.095	16.681	
10.0	165.0	6.3	155.6	-3.8	3.944	-1.027	-0.839	-0.088	19.049	
10.0	170.0	5.9	162.8	-4.2	4.240	-0.092	-0.066	-0.009	80.351	
10.0	175.0	5.8	171.4	-4.2	4.207	-0.127	-0.092	-0.012	90.000	NOT OPT.
10.0	180.0	5.8	180.0	-4.2	4.211	0.005	0.003	0.000	90.000	
20.0	20.0	25.0	15.9	5.0	5.283	18.275	5.144	3.525	5.689	
20.0	25.0	25.2	19.6	5.1	5.657	18.817	4.308	4.142	6.511	
20.0	30.0	25.3	23.2	5.2	5.958	19.020	3.677	4.668	7.193	
20.0	35.0	25.3	26.9	5.1	6.212	18.911	3.170	5.090	7.795	
20.0	40.0	25.3	30.6	4.9	6.430	18.545	2.748	5.404	8.347	
20.0	50.0	24.9	38.0	4.4	6.789	17.146	2.086	5.696	9.334	
20.0	60.0	24.3	45.4	3.5	7.060	15.080	1.601	5.548	10.172	
20.0	70.0	23.5	53.1	2.5	7.252	12.516	1.239	4.981	10.864	
20.0	80.0	22.5	61.2	1.3	7.362	9.673	0.957	4.071	11.430	
20.0	90.0	21.3	69.7	0.0	7.391	6.763	0.715	2.943	11.912	
20.0	100.0	20.1	78.9	-1.3	7.343	3.975	0.474	1.746	12.335	
20.0	110.0	18.8	88.8	-2.5	7.224	1.453	0.203	0.628	12.732	
20.0	120.0	17.6	99.7	-3.5	7.036	-0.700	-0.114	-0.284	13.326	
20.0	130.0	16.5	111.6	-4.4	6.781	-2.494	-0.465	-0.895	13.830	
20.0	140.0	15.6	124.5	-5.0	6.479	-3.920	-0.848	-1.228	14.520	
20.0	150.0	15.0	138.2	-5.3	6.129	-5.010	-1.281	-1.345	15.679	
20.0	155.0	14.8	145.3	-5.4	5.931	-5.405	-1.517	-1.331	16.682	
20.0	160.0	14.8	152.4	-5.4	5.715	-5.588	-1.740	-1.252	18.394	
20.0	165.0	14.2	158.6	-5.9	6.135	-0.571	-0.160	-0.158	77.445	
20.0	170.0	14.0	165.7	-6.0	6.089	-0.361	-0.104	-0.098	82.866	
20.0	175.0	14.0	172.8	-6.0	6.062	-0.443	-0.129	-0.119	90.000	NOT OPT.
20.0	180.0	13.9	180.0	-6.1	6.072	0.028	0.008	0.007	90.000	
30.0	20.0	35.4	16.9	5.3	5.648	27.076	4.748	6.122	1.144	
30.0	25.0	35.5	20.9	5.5	6.020	27.265	3.780	7.021	1.350	

A SAILWING USED AS PROPULSION FOR A SAILBOAT

CHARACTERISTIC LENGTH = 23.800

VT	GAMMA	VA	BETA	VMG	VB	PHI	LAMDA	RUDDER	ALPHA	
30.0	30.0	35.6	24.9	5.5	6.332	27.297	3.042	7.850	1.636	
30.0	35.0	35.6	28.9	5.4	6.607	27.226	2.436	8.627	2.023	
30.0	40.0	35.5	32.9	5.3	6.857	27.065	1.916	9.361	2.531	
30.0	50.0	35.2	40.8	4.7	7.313	26.521	1.036	10.724	3.989	
30.0	60.0	34.5	48.8	3.9	7.736	25.095	0.350	11.645	6.106	
30.0	70.0	33.7	56.9	2.8	8.117	21.850	-0.065	11.443	8.110	
30.0	80.0	32.5	65.2	1.5	8.420	17.391	-0.232	10.118	10.055	
30.0	90.0	31.2	74.0	0.0	8.610	11.947	-0.192	7.565	11.296	
30.0	100.0	29.7	83.3	-1.5	8.663	6.368	-0.073	4.294	12.155	
30.0	110.0	28.2	93.4	-2.9	8.578	1.104	0.001	0.774	12.852	
30.0	120.0	26.8	104.3	-4.2	8.356	-3.538	-0.073	-2.230	13.710	
30.0	130.0	25.6	116.1	-5.2	8.043	-7.480	-0.320	-4.064	14.540	
30.0	140.0	24.6	128.5	-5.9	7.665	-10.585	-0.725	-4.928	15.763	
30.0	150.0	24.0	141.3	-6.3	7.232	-12.459	-1.258	-4.924	18.032	
30.0	155.0	23.3	147.1	-6.9	7.591	-0.086	-0.009	-0.044	72.393	
30.0	160.0	23.1	153.6	-7.1	7.536	-0.974	-0.104	-0.477	75.285	
30.0	165.0	22.9	160.1	-7.2	7.480	-1.262	-0.142	-0.603	79.139	
30.0	170.0	22.7	166.7	-7.3	7.436	-0.351	-0.042	-0.167	84.395	
30.0	175.0	22.6	173.4	-7.4	7.418	-0.097	-0.012	-0.046	90.000	NOT OPT.
30.0	180.0	22.6	180.0	-7.4	7.425	0.072	0.009	0.034	90.000	
40.0	20.0	45.4	17.5	5.4	5.717	29.637	4.516	6.957	-2.173	
40.0	25.0	45.6	21.8	5.5	6.096	29.254	3.532	7.820	-2.135	
40.0	30.0	45.7	26.0	5.6	6.417	29.226	2.763	8.751	-2.017	
40.0	35.0	45.7	30.2	5.5	6.702	29.042	2.129	9.611	-1.824	
40.0	40.0	45.6	34.4	5.3	6.966	28.858	1.571	10.467	-1.549	
40.0	50.0	45.2	42.7	4.8	7.462	28.564	0.572	12.270	-0.701	
40.0	60.0	44.5	51.1	4.0	7.940	27.289	-0.263	13.757	0.344	
40.0	70.0	43.6	59.5	2.9	8.426	25.998	-1.099	15.508	2.689	
40.0	80.0	42.5	68.0	1.6	8.954	23.405	-1.751	16.460	6.301	
40.0	90.0	41.1	76.7	0.0	9.433	17.010	-1.642	13.773	9.613	
40.0	100.0	39.5	86.0	-1.7	9.703	8.435	-0.913	7.703	11.545	
40.0	110.0	37.8	96.1	-3.3	9.692	-0.076	0.008	-0.077	13.343	
40.0	120.0	36.2	107.1	-4.7	9.363	-7.883	0.606	-6.536	14.969	
40.0	130.0	35.0	118.8	-5.7	8.897	-13.348	0.581	-9.225	17.725	
40.0	140.0	33.8	130.4	-6.7	8.744	5.613	-0.102	3.919	63.882	
40.0	150.0	32.7	142.3	-7.6	8.735	0.788	-0.011	0.584	68.897	
40.0	155.0	32.3	148.5	-7.9	8.692	-0.396	0.004	-0.291	73.295	
40.0	160.0	32.0	154.7	-8.1	8.633	-1.406	0.006	-0.999	76.899	
40.0	165.0	31.8	161.0	-8.3	8.576	-1.297	-0.002	-0.906	80.973	
40.0	170.0	31.6	167.3	-8.4	8.535	-0.170	-0.001	-0.119	85.177	
40.0	175.0	31.5	173.7	-8.5	8.518	-0.183	-0.001	-0.128	89.333	NOT OPT.
40.0	180.0	31.5	180.0	-8.5	8.527	0.141	0.001	0.099	90.000	

A SAILWING USED AS PROPULSION FOR A SAILBOAT

COMPUTATION STATISTICS

THE TOTAL NUMBER OF EQUILIBRIUMS COMPUTED WAS 777, OF WHICH 88 WERE OPTIMUM.
TOTAL CPU TIME SPENT COMPUTING EQUILIBRIUMS WAS 4875 HUNDRETHS OF A SECOND.
THE AVERAGE TIME REQUIRED TO COMPUTE A SINGLE EQUILIBRIUM WAS 6.274131E+00 HUNDRETHS OF A SECOND.

APPENDIX B

The procedures containing the mathematical models used by the two runs discussed in this paper are reproduced in this appendix. Both procedures were given the name, PMODL, and were compiled and link-edited to the main program just prior to the time of execution.

The requirements placed on the format of the procedure, PMODL, and its entry points by the main program are as follows:

1. The procedure PMODL shall have eight entry points with the labels: PMODL, MODLIN, ERFX, ERFY, ERFZ, ERMX, ERMY and ERMZ.
2. The argument lists for each of the entry points should be of the form indicated by the two examples reproduced herein.
3. Routines that compute required quantities, eg. the apparent wind triangle, should be included in PMODL as internal procedures. Both examples show the use of the simple procedure TRI which can be used to compute the apparent wind speed and angle. (Note the use of the external variables VTCOSG and VTSING. These variables have been previously calculated by the main procedure.)

The following is a list of the required entry points and statements concerning each their arguements.

PMODL: PROCEDURE (MDLNAM,NHCOF,NACOF,DOF,DOP)

MDLNAM - vector of names given to the aerodynamic and hydrodynamic portions of the math model; each element declared as being a maximum of 40 characters in length.

NHCOF - number of coefficients to be input to the hydrodynamic model.

NACOF - number of coefficients to be input to the aerodynamic model.

DOF - degrees-of-freedom in model.

DOP - degrees-of-optimization in model. If equal to zero, the variables Q and OVARNAM should still be dimensioned as single element vectors.

MODLIN: ENTRY (HCOFNAM,HCOFVAL,ACOFNAM,ACOFVAL,OVARNAM)

HCOFNAM - vector of names for hydrodynamic coefficients, NHCOF elements.

HCOFVAL - vector of values read from file COEFF for the hydrodynamic coefficients.

ACOFNAM, ACOFVAL - like HCOFNAM and HCOFVAL, but for the aerodynamic model.

OVARNAM - vector of names given to the optimization variables. DOP elements, declared a maximum of six characters in length.

ERFX, ERFY, ERFZ, ERMX, ERMY,ERMZ: ENTRY (Q,P)

Q - vector of optimization variable values, dimensioned DOP.

P - vector of independent variable values, dimensioned DOF.

Note: Each of these entries returns a function value, in this case the value of ERR, or the difference in the aero and hydro components of the principle force or moment equation being evaluated.

U.M11098.12849.MODEL1.PLI

```
PMODL:PROCEDURE (MDLNAM,NHCOF,NACOF,DOF,DOP);
      /*****************************************************************/
      /*                                                               */
      /* HYDRODYNAMIC AND AERO DYNAMIC MODELS USED TO EVALUATE THE     */
      /* FORCES AND MOMENTS ON THE SAILBOAT.                           */
      /* GIF MUNGER'S HYDRO AND SAIL POLYNOMIAL AERO.                  */
      /*                                                               */
      /*****************************************************************/
      DCL ACOFNAM(2) CHAR(10),
          HCOFNAM(6) CHAR(10),
          (ERR,P(4),Q(1),ACOFVAL(2),HCOFVAL(6))
          FLOAT BIN(53),
          ACOFVAL_SAVE(2) FLOAT BIN(53) STATIC,
          HCOFVAL_SAVE(6) FLOAT BIN(53) STATIC,
          MDLNAM(2) CHAR(40) ,
          (NACOF,NHCOF,DOF,DOP)
          FIXED BIN,
          OVARNAM(1) CHAR(6) VARYING ,
          (CONST1,CONST2,CONST3,CONST4,ONE,TWO) BIN FLOAT(53) STATIC,
          (CONST5,CONST6,CONST7,CONST8,AFX,AFY,RFY)
          BINARY FLOAT(53) STATIC,

      /************************ EXTERNAL VARIABLES *****************/

          (XDIM,VA,BETA,VTSING,VTCOSG) BINARY FLOAT(53) EXTERNAL;

      ONE = 1.0E0; TWO = 2.0E0; CONST1 = 1.19E-3; CONST2 = 5.4E1;
      CONST3 = 3.0E1; CONST4 = 3.34E-2; CONST5 = 3.0E1;
      CONST6 = 1.17E-2; CONST7 = 7.0E-1; CONST8 = 1.25E0;
      NACOF = 2; NHCOF = 6; DOF = 4; DOP = 1;
      MDLNAM(1) = 'GIF MUNGER''S GIMCRACK HYDRO';
      MDLNAM(2) = 'GIF MUNGER''S GIMCRACK AERO';
      RETURN;
MODLIN:
      ENTRY (HCOFNAM,HCOFVAL,ACOFNAM,ACOFVAL,OVARNAM);
      HCOFNAM(1) = 'C1'; HCOFNAM(2) = 'C2'; HCOFNAM(3) = 'C3';
      HCOFNAM(4) = 'C4'; HCOFNAM(5) = 'C5'; HCOFNAM(6) = 'CXR';
      ACOFNAM(1) = 'CSA'; ACOFNAM(2) = 'CHCE';
      OVARNAM(1) = 'REEF';
      GET FILE(COEFF) LIST(HCOFVAL,ACOFVAL);
      ACOFVAL_SAVE = ACOFVAL;
      HCOFVAL_SAVE = HCOFVAL;
      RETURN;
ERFX: ENTRY (Q,P) RETURNS(FLOAT BIN(53));
      CALL TRI;
      PHI = ABS(P(2));
      AFX = CONST1*(VA**2)*ACOFVAL_SAVE(1)*(Q(1)**2)*(XDIM**2)*
      (SIND(BETA+CONST5) + BETA*CONST6 - CONST7)*(COSD(PHI))**2;
      ERR = AFX
      -XDIM*(HCOFVAL_SAVE(1)*ABS(P(1)**4.8)/(COSD(PHI)**2))
      -XDIM*(HCOFVAL_SAVE(2)*ABS(P(3))*(ONE+SIND(PHI))*P(1)**2)
      -XDIM*(HCOFVAL_SAVE(3)*ABS(P(4))*(ONE+SIND(PHI))*P(1)**2);
      RETURN (ERR);
ERMX: ENTRY (Q,P) RETURNS(FLOAT BIN(53));
      CALL TRI;
      PHI = ABS(P(2));
      AFY = CONST1*(VA**2)*ACOFVAL_SAVE(1)*(Q(1)**2)*(XDIM**2)*
```

```
        ((COSD(BETA/CONST8-CONST2))**2)*COSD(PHI)**2;
        ERR = XDIM*ACOFVAL_SAVE(2)*AFY/COSD(PHI)
        -(XDIM**4)*CONST4*(ONE+COSD(PHI))*SIND(P(2));
        RETURN (ERR);
 ERFY:  ENTRY (Q,P) RETURNS(FLOAT BIN(53));
        CALL TRI;
        PHI = ABS(P(2));
        AFY = CONST1*(VA**2)*ACOFVAL_SAVE(1)*(Q(1)**2)*(XDIM**2)*
        ((COSD(BETA/CONST8-CONST2))**2)*COSD(PHI)**2;
        RFY = XDIM*(HCOFVAL_SAVE(5)*P(4)*(ONE+SIND(PHI))*P(1)**2);
        ERR = AFY - RFY - XDIM*(HCOFVAL_SAVE(4)*P(3)*(ONE+SIND(PHI)) *
        P(1)**2);
        RETURN (ERR) ;
 ERMZ:  ENTRY (Q,P) RETURNS(FLOAT BIN(53));
        CALL TRI;
        PHI = ABS(P(2));
        AFX = CONST1*(VA**2)*ACOFVAL_SAVE(1)*(Q(1)**2)*(XDIM**2)*
        (SIND(BETA+CONST5) + BETA*CONST6 - CONST7)*(COSD(PHI))**2;
        RFY = XDIM*(HCOFVAL_SAVE(5)*P(4)*(ONE+SIND(PHI))*P(1)**2);
        ERR = XDIM*ACOFVAL_SAVE(2)*SIND(P(2))*AFX
        -XDIM*HCOFVAL_SAVE(6)*RFY;
        RETURN (ERR);
 ERFZ:  ENTRY (Q,P) RETURNS(FLOAT BIN(53));
        RETURN (0.0E0);
 ERMY:  ENTRY (Q,P) RETURNS(FLOAT BIN(53));
        RETURN (0.0E0);
        /******************** WIND TRIANGLE *************************/
 TRI:   PROCEDURE;
        VA = SQRT((P(1) + VTCOSG)**2 + VTSING**2);
        IF ABS(P(1) + VTCOSG) < 1.0E-8 THEN BETA = 90.0;
        ELSE BETA = ATAND(VTSING/(P(1) + VTCOSG));
        IF BETA <= 0.0 THEN BETA = BETA + 180.0;
        RETURN;
        END TRI;
        END PMODL;
```

U.M11098.12849.MODEL1.PLI

```
 PMODL:PROCEDURE (MDLNAM,NHCOF,NACOF,DOF,DOP);
       /*****************************************************************/
       /*                                                                */
       /* HYDRODYNAMIC AND AERO DYNAMIC MODELS USED TO EVALUATE THE     */
       /* FORCES AND MOMENTS ON THE SAILBOAT.                           */
       /* GIF MUNGER'S HYDRO AND SAIL POLYNOMIAL AERO.                  */
       /*                                                                */
       /*****************************************************************/
       DCL ACOFNAM(2) CHAR(10),
           HCOFNAM(6) CHAR(10),
           (ERR,P(4),Q(1),ACOFVAL(2),HCOFVAL(6))
           FLOAT BIN(53),
           ACOFVAL_SAVE(2) FLOAT BIN(53) STATIC,
           HCOFVAL_SAVE(6) FLOAT BIN(53) STATIC,
           MDLNAM(2) CHAR(40) ,
           (NACOF,NHCOF,DOF,DOP)
           FIXED BIN,
           OVARNAM(1) CHAR(6) VARYING ,
           (CONST1,CONST2,CONST3,CONST4,ONE,TWO) BIN FLOAT(53) STATIC,
           (CONST5,CONST6,CONST7,CONST8,AFX,AFY,RFY)
           BINARY FLOAT(53) STATIC,

       /************************ EXTERNAL VARIABLES *****************/

           (XDIM,VA,BETA,VTSING,VTCOSG) BINARY FLOAT(53) EXTERNAL;

      ONE = 1.0E0; TWO = 2.0E0; CONST1 = 1.19E-3; CONST2 = 5.4E1;
      CONST3 = 3.0E1; CONST4 = 3.34E-2; CONST5 = 3.0E1;
      CONST6 = 1.17E-2; CONST7 = 7.0E-1; CONST8 = 1.25E0;
      NACOF = 2; NHCOF = 6; DOF = 4; DOP = 1;
      MDLNAM(1) = 'GIF MUNGER''S GIMCRACK HYDRO';
      MDLNAM(2) = 'GIF MUNGER''S GIMCRACK AERO';
      RETURN;
MODLIN:
      ENTRY (HCOFNAM,HCOFVAL,ACOFNAM,ACOFVAL,OVARNAM);
      HCOFNAM(1) = 'C1'; HCOFNAM(2) = 'C2'; HCOFNAM(3) = 'C3';
      HCOFNAM(4) = 'C4'; HCOFNAM(5) = 'C5'; HCOFNAM(6) = 'CXR';
      ACOFNAM(1) = 'CSA'; ACOFNAM(2) = 'CHCE';
      OVARNAM(1) = 'REEF';
      GET FILE(COEFF) LIST(HCOFVAL,ACOFVAL);
      ACOFVAL_SAVE = ACOFVAL;
      HCOFVAL_SAVE = HCOFVAL;
      RETURN;
ERFX: ENTRY (Q,P) RETURNS(FLOAT BIN(53));
      CALL TRI;
      PHI = ABS(P(2));
      AFX = CONST1*(VA**2)*ACOFVAL_SAVE(1)*(Q(1)**2)*(XDIM**2)*
      (SIND(BETA+CONST5) + BETA*CONST6 - CONST7)*(COSD(PHI))**2;
      ERR = AFX
      -XDIM*(HCOFVAL_SAVE(1)*ABS(P(1)**4.8)/(COSD(PHI)**2))
      -XDIM*(HCOFVAL_SAVE(2)*ABS(P(3))*(ONE+SIND(PHI))*P(1)**2)
      -XDIM*(HCOFVAL_SAVE(3)*ABS(P(4))*(ONE+SIND(PHI))*P(1)**2);
      RETURN (ERR);
ERMX: ENTRY (Q,P) RETURNS(FLOAT BIN(53));
      CALL TRI;
      PHI = ABS(P(2));
      AFY = CONST1*(VA**2)*ACOFVAL_SAVE(1)*(Q(1)**2)*(XDIM**2)*
```

```
      ((COSD(BETA/CONST8-CONST2))**2)*COSD(PHI)**2;
      ERR = XDIM*ACOFVAL_SAVE(2)*AFY/COSD(PHI)
      -(XDIM**4)*CONST4*(ONE+COSD(PHI))*SIND(P(2));
      RETURN (ERR);
ERFY: ENTRY (Q,P) RETURNS(FLOAT BIN(53));
      CALL TRI;
      PHI = ABS(P(2));
      AFY = CONST1*(VA**2)*ACOFVAL_SAVE(1)*(Q(1)**2)*(XDIM**2)*
      ((COSD(BETA/CONST8-CONST2))**2)*COSD(PHI)**2;
      RFY = XDIM*(HCOFVAL_SAVE(5)*P(4)*(ONE+SIND(PHI))*P(1)**2);
      ERR = AFY - RFY - XDIM*(HCOFVAL_SAVE(4)*P(3)*(ONE+SIND(PHI)) *
      P(1)**2);
      RETURN (ERR) ;
ERMZ: ENTRY (Q,P) RETURNS(FLOAT BIN(53));
      CALL TRI;
      PHI = ABS(P(2));
      AFX = CONST1*(VA**2)*ACOFVAL_SAVE(1)*(Q(1)**2)*(XDIM**2)*
      (SIND(BETA+CONST5) + BETA*CONST6 - CONST7)*(COSD(PHI))**2;
      RFY = XDIM*(HCOFVAL_SAVE(5)*P(4)*(ONE+SIND(PHI))*P(1)**2);
      ERR = XDIM*ACOFVAL_SAVE(2)*SIND(P(2))*AFX
      -XDIM*HCOFVAL_SAVE(6)*RFY;
      RETURN (ERR);
ERFZ: ENTRY (Q,P) RETURNS(FLOAT BIN(53));
      RETURN (0.0E0);
ERMY: ENTRY (Q,P) RETURNS(FLOAT BIN(53));
      RETURN (0.0E0);
      /********************* WIND TRIANGLE **************************/
TRI:  PROCEDURE;
      VA = SQRT((P(1) + VTCOSG)**2 + VTSING**2);
      IF ABS(P(1) + VTCOSG) < 1.0E-8 THEN BETA = 90.0;
      ELSE BETA = ATAND(VTSING/(P(1) + VTCOSG));
      IF BETA <= 0.0 THEN BETA = BETA + 180.0;
      RETURN;
      END TRI;
      END PMODL;
```

APPENDIX C

A. Input deck for the first test run.

```
// EXEC PLIXG,PRCG='U.id.PERFORM.LOAD(MOD1)'
//G.AUXOUT DD DUMMY
//G.BUGS   DD SYSOUT=A
//G.TERM   DD DUMMY
//G.COEFF  DD *
0.00086 0.0034 0.00084 0.071 0.0178 0.34
0.75 0.70
```

These are coefficient values for inclusion with the mathematical model.

```
//G.WINDY  DD *
4 12 10. 20. 30. 40.
180. 160. 140. 120. 100. 80. 60. 40.
35. 30. 25. 20.
```

Number of true wind speeds and directions, followed by their values.

```
//G.SYSIN  DD *
SAMPLE FOUR DEGREE OF FREEDOM MODEL WITH ONE DEGREE OF
OPTIMIZATION
2 23.8 25.0
```

Number of lengths and their values.

12.0 0.5 40.0 -40.0 12.0 -12.0 35.0 -35.0

0.001 0.1 0.1 0.1

Maximum and minimum values for the independent variables and their convergence tolerances.

1.0 0.33 0.001

Maximum and minimum values for the optimization variable and its convergence tolerance.

10 0.38 1.0

Maximum allowed iterations per solution. Starting value for the ratio VB/VT and the optimization variable.

B. Input deck for the second test run. For explanation of input data see first run.

```
// EXEC PLIXG,PROG='U.id.SAIL.LOAD(MOD1)'
//G.AUXOUT DD DUMMY
//G.BUGS   DD SYSOUT=A
//G.TERM   DD DUMMY
//G.COEFF  DD *
0.00086 0.0034 0.00084 0.071 0.0178 0.34
0.75 0.70
//G.WINDY  DD *
4 22 10. 20. 30. 40.
20. 25. 30. 35. 40. 50. 60. 70. 80. 90.
100. 110. 120. 130. 140. 150. 155. 160.
165. 170. 175. 180.
//G.SYSIN  DD *
A SAILWING USED AS PROPULSION FOR A SAILBOAT
1 23.8
12.0 0.5 40.0 -40.0 12.0 -12.0 35.0 -35.0
0.001 0.1 0.1 0.1
90. -10.      0.1
10    0.38 0.0
```