U.S.N.A. --- Trident Scholar project report; no. 315 (2003)

Performance Prediction of the Mk II Navy 44 Sail Training Craft with respect to Tank Testing, Velocity Prediction Programs, and Computational Fluid Dynamics

by

Midshipman Jon P. Silverberg, Class of 2003
United States Naval Academy
Annapolis, Maryland

Certification of Adviser Approval

Assistant Professor Paul H. Miller
Naval Architecture and Ocean Engineering Department

Acceptance for the Trident Scholar Committee

Professor Joyce E. Shade
Deputy Director of Research & Scholarship

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE **05 MAY 2003**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**

4. TITLE AND SUBTITLE **Performance Prediction of the Mk II Navy 44 Sail Training Craft with respect to Tank Testing, Velocity Prediction Programs, and Computational Fluid Dynamics**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **United States Naval Academy Annapolis, Maryland 21402**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S) 11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**UU**	**153**	

ABSTRACT

The powering requirement of a ship is one of the most important aspects of naval architecture. Traditionally, ships have been tested for hull resistance using hydrodynamic tank testing. Tank testing evaluates models of ships by measuring their resistance in a tow tank. This has proved to be useful and accurate, but is very time consuming, expensive, and has inherent scaling errors. Because of these reasons, today many vessels are sold on the market without any model testing. Another set of design tools is parametric predictions. Parametric predictions contain acquired data for a specific family of hull forms and use key hull parameters to evaluate a particular design. Parametric predictions run quickly, but cannot be used to evaluate any hull outside of the limits of previously tested hulls. Computational Fluid Dynamics (CFD) codes are a much newer method of determining the powering requirement for ships. CFD uses numerical modeling to simulate the forces which act on the ship. Unlimited testing takes a fraction of the time and effort as compared to tank testing. However, it is not yet wholly proven in its accuracy.

An investigation has been made using tank testing and CFD to predict the new David Pedrick Mk II Navy 44 Sail Training Craft's (STC) performance. The performance prediction process proved most accurate when using tank data to aid the CFD calculations. Using this combined method along with an aerodynamic calculation for the Mk II Navy 44 STC, a Velocity Prediction Program (VPP) was constructed. Finally, the new VPP as well as previous parametric VPPs were used to predict a full set of performance figures. These tools were also used to improve the rudder design of the Mk II Navy 44 STC.

Keywords:

Ship Resistance

Computational Fluid Dynamics

Tow Tank Testing

Velocity Prediction Program

Sail Craft

ACKNOWLEDGMENTS

Many thanks are to be given to the United States Naval Academy's Department of Naval Architecture and Ocean Engineering. Professor Paul Miller, my Trident Scholar Advisor, provided invaluable aid in dealing with sailing craft dynamics and in keeping me on track through the year. Professors Greg White and Rameswar Bhattacharyya also gave invaluable advice on resistance testing. The Naval Academy's Technical Services Department provided a great amount of help from Mr. John Zseleczky, Mr. John Hill, and Mr. Bill Beaver. Without the aid of Mr. Zselczky, I would still be doing tests in the tow tank. Special thanks extend to Dr. Dane Hendrix, Joe Laiosa, and Bruce Rosen for my use of their CFD codes. Furthermore, Joe Laiosa and Bruce Rosen both supplied hours of assistance as I struggled to get their code working. I would also like to thank Professor Shade and the Trident Subcommittee for giving me this opportunity to do this study.

TABLE OF CONTENTS

LIST OF TABLES AND FIGURES

PERFORMANCE PREDICTION

Performance prediction has always been an important parameter in ship design. From cruising boats to America's Cup yachts to naval surface ships, the speed of the vessel will greatly influence its mission effectiveness. Today there are three methods of making performance predictions of vessels:

- Parametric predictions
- Hydrodynamic tank testing
- Computational fluid dynamics

What this study entailed was an examination of these methods of performance prediction. In order to research these tools extensively, a sailing craft was evaluated because of the complexity in its hydrodynamic characteristics. For example, a naval vessel or a cargo ship is typically only evaluated in its forward motion. A sailing vessel, however, must be evaluated not only in forward motion, but also for varying amounts of heel angle, yaw angle, rudder angle, and trim angle. Furthermore, the lifting surfaces on a sailing craft such as the keel and rudder have tremendous effect on the performance of the vessel and so must also be evaluated.

For sailing craft, parametric predictions often take the form of Velocity Prediction Programs (VPPs). These parametric VPPs are the result of the testing of many hull forms in the tow tank. VPPs are computer programs which predict the hydrodynamic as well as aerodynamic configurations of sailing craft. Parametric VPPs take key measurements of the hull and sailing rig and then interpolate through supplied data to make predictions for any vessel. However, parametric VPPs are restricted in their ability to extrapolate beyond the limits of their hydrodynamic data. For each wind condition, the program will iterate boat speed, heel angle, and sail conditions so that the sum of the aerodynamic and hydrodynamic forces and moments on the vessel is zero. The speed of VPPs is what makes these tools so valuable. Today, VPPs can be run even through personal computers in a matter of minutes. Parametric VPPs are used mostly for preliminary yacht designs. They will give as output the estimated speed of the boat for any wind condition as well as the angle of heel, side force, and drag.

Over many years, naval architecture has refined the method of hydrodynamic tank testing. Simply defined, tank testing consists of measuring the forces on a geometrically scaled model which is being towed through a tank of water. The measured forces from the tank tests can then be extrapolated to predict the forces on the full-sized vessel. For sailing craft, an elaborate testing program is necessary. Due to aerodynamic forces, the towing rig must be equipped to change the vessel's yaw, heel, trim, and rudder angles. Although drag is the primary output from the tests, lift and yawing moment are also important to a sailboat. Besides the complexity of the testing rig, tank testing has further inherent difficulties in scale effects. Though the model can be geometrically scaled down, the water molecules cannot, leading to inherent flow differences between the ship and the model. For this reason, there are many numerical adjustments and engineering assumptions which add to the extrapolation of the data.

Computer codes known as Computational Fluid Dynamics (CFD) use fundamental fluid dynamics equations to predict the resistance of the ship moving through water. Using a digitized hull, a computer puts the ship into a virtual tow tank and calculates the lift, drag, yawing moment, and trim on the body. However, the accuracy of the simulation is primarily a function of the quality of the code and the quality of the digitized hull. Furthermore, there are complexities due to viscous effects and the free-surface.

THE PERFORMANCE PREDICTION PROCESS

The intention of this study was to compare these different performance prediction tools and create a process which will accurately evaluate the Mk II Navy 44 Sail Training Craft (STC). This study compared the parametric predictions of the International Measurement System (IMS) and PCSail VPPs to small model tow tank testing and to the FKS and SPLASH CFD codes. The IMS VPP results were supplied by the Mk II Navy 44 designer, David Pedrick. The PCSail VPP code was developed by David Martin and Robert Beck from the University of Michigan and was distributed at the 15th Chesapeake Sailing Yacht Symposium. The tank testing was conducted in the 120' tow tank at the United States Naval Academy. The FKS code was developed at the David Taylor Model Basin, Naval Surface Warfare Center - Carderock, by Dr. F. Noblesse, in cooperation with C. Yang of George Mason University. The SPLASH code was developed by Bruce Rosen and Joe Laiosa of South Bay Simulations.

One important aspect of this study was to compare the influence of turbulent and laminar flow on the lifting appendages of the model. The model has a waterline length of 45" with the keel having a span of less than 6" and the rudder's span of less than 2". The model's length was a function of the size of the tooling equipment in the model shop and the size of the tow tank. Furthermore, both CFD codes were inviscid – the frictional component of the fluid was not calculated in the initial drag prediction. Both codes were chosen based on their speed of calculation and their availability.

The testing program made performance predictions of the sailboat at four wind speeds and five wind angles as well as the upright condition which was equivalent to the vessel under mechanical propulsion. The IMS VPP, Tank Testing, FKS, and SPLASH evaluated the boat in each performance condition. The PCSail VPP evaluated the boat’s initial sailing characteristics in each condition.

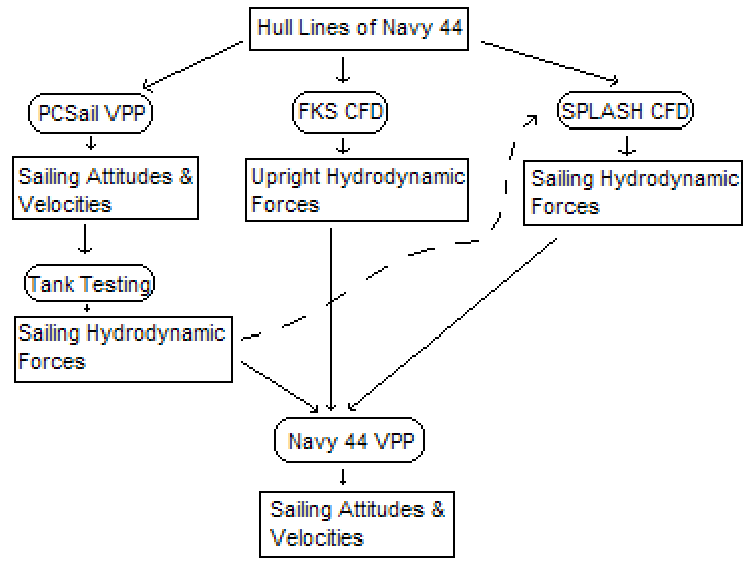

Hull Lines of Navy 44
PCSail VPP
FKS CFD
SPLASH CFD
Sailing Attitudes & Velocities
Upright Hydrodynamic Forces
Sailing Hydrodynamic Forces
Tank Testing
Sailing Hydrodynamic Forces
Navy 44 VPP
Sailing Attitudes & Velocities

Figure 1 – Performance prediction flow diagram

An initial performance prediction process shown in Figure (1) was developed through this study. The hull lines and rig diagram of the Mk II Navy 44 STC were the initial data necessary for all calculations. From these diagrams of the Navy 44, the PCSail VPP was able to compute the sailing attitudes and velocities of the Navy 44. These positions were used to determine which conditions would be tested in the tank. The tank testing gave both upright and sailing hydrodynamic forces as output.

As in PCSail, the diagrams of the Navy 44 were the only necessity to run the FKS CFD code. The primary output from FKS was upright hydrodynamic forces. Sailing hydrodynamic conditions were evaluated, but the accuracy of FKS in evaluating sailing conditions was poor.

In order for the SPLASH CFD to be more accurate, some of the tank testing data was used as input to SPLASH. SPLASH made predictions as a virtual tow tank, being able to predict both upright and sailing hydrodynamic forces. Finally, all of these hydrodynamic forces were used by a custom-made VPP for the Mk II Navy 44. The output of this custom VPP was the optimum sailing attitudes and velocities of the Mk II Navy 44.

PCSail Velocity Prediction Program

VPPs operate by the principle that a sailing vessel is in equilibrium. Therefore, the summation of forces and moments on the vessel must be zero. In this way, a VPP is divided into two parts: hydrodynamic and aerodynamic. The hydrodynamic forces are a hull form's reaction to the vessel being driven through the water. These forces include hydrodynamic drag, side force, yawing moment, and righting moment (Figures (2), (3)). The hydrodynamic forces resist the aerodynamic forces on the boat due to sails. These forces are aerodynamic drive, side force, yawing moment, and heeling moment. When each of the hydrodynamic forces equals the opposite aerodynamic forces, the boat is in equilibrium and is considered a possible sailing condition.[1] After a VPP solves for possible sailing conditions, the VPP finds the sailing conditions which are fastest around a course and considers them the optimum conditions.[2]

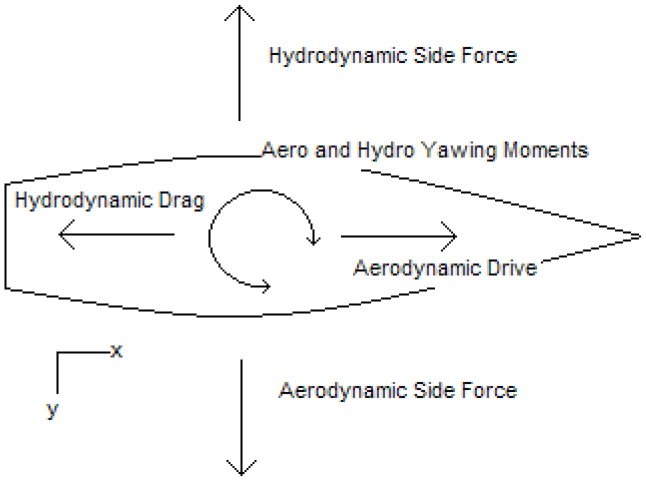


Figure 2 - Z-plane force and moments diagram

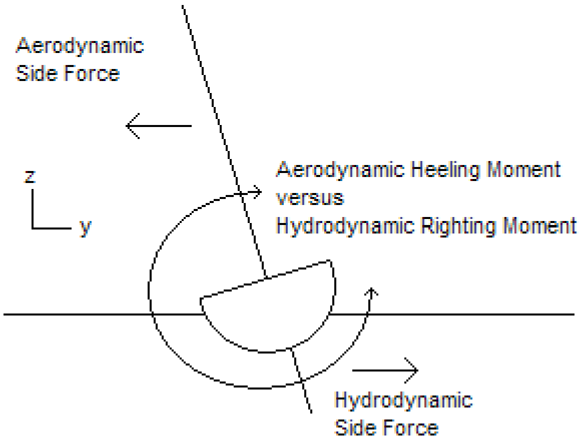


Figure 3 - X-plane force and moments diagram

PCSail was supplied as freeware at the 15th annual Chesapeake Sailing Yacht Symposium. PCSail is a spreadsheet for Microsoft Excel® which predicts the forces on the boat and the optimum sailing conditions.[9]

PCSail predicts the hydrodynamic forces on a vessel from the Delft series of sailing yacht model tests. The Delft series is a set of hydrodynamic data taken from a series of different

[1] Longitudinal trimming moments are typically ignored in a VPP since its calculations have limited effects on velocity. In the same way, Sinkage and buoyant forces are also assumed to have negligible effects on velocity.

sailing hull forms. By giving key hull parameters, PCSail interpolates through the Delft series to predict the hydrodynamic qualities of a given hull form. [9]

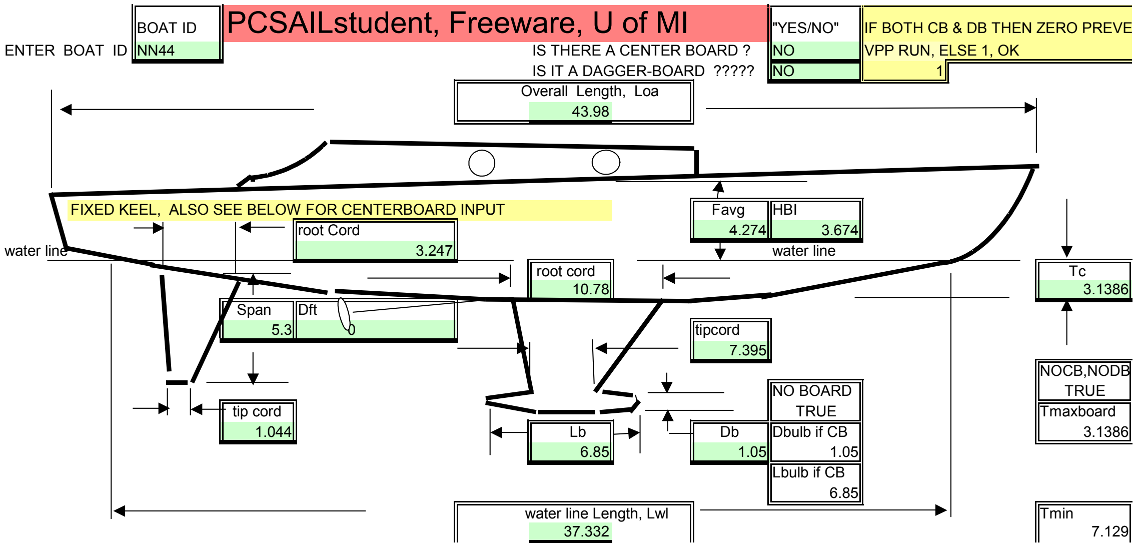
The Hazen method documented by Larsson and Eliasson calculates the aerodynamic forces generated by the sailing rig. The characteristic lift and drag of sails for PCSail were found via experimental data as well as theoretical calculations. After giving sail shape inputs, PCSail predicts the lift and drag on the sails specific to a vessel. Finally, lift and drag in the sail-axis are corrected to aerodynamic drive and lift in the vessel-axis.[5]

PCSail only solves to equate aerodynamic drive to hydrodynamic drag and heeling moment to righting moment. Forces which are ignored included hydrodynamic side force and yawing moments. Aerodynamic side force is considered only for calculating heeling moment.[9]

PCSail Setup

Designer David Pedrick supplied the IMS VPP predictions for the Mk II Navy 44' STC. The IMS VPP serves as the standard of performance prediction and a rating tool for racing yachts. However, since the IMS VPP output did not give sailing characteristics such as sail forces and heel angles, the PCSail VPP was used. The boat characteristics from the IMS VPP were used as inputs to the PCSail VPP.

The initial inputs into the PCSail program were very straightforward as in Figure (4). Some modifications had to be made to the program so that the IMS VPP and PCSail VPPs agreed in their output.



PCSAILstudent, Freeware, U of MI

ENTER BOAT ID — BOAT ID: NN44

"YES/NO"
IS THERE A CENTER BOARD ? NO
IS IT A DAGGER-BOARD ????? NO

IF BOTH CB & DB THEN ZERO PREVE
VPP RUN, ELSE 1, OK
1

Overall Length, Loa: 43.98

FIXED KEEL, ALSO SEE BELOW FOR CENTERBOARD INPUT

root Cord: 3.247
Span: 5.3
Dft: 0
tip cord: 1.044
root cord: 10.78
tipcord: 7.395
Lb: 6.85
Db: 1.05
Favg: 4.274
HBI: 3.674
Tc: 3.1386
NO BOARD: TRUE
Dbulb if CB: 1.05
Lbulb if CB: 6.85
NOCB,NODB: TRUE
Tmaxboard: 3.1386
Tmin: 7.129
water line Length, Lwl: 37.332

Figure 4 - PCSail sample hull characteristics input

Modifications to PCSail

One modification was that of GM, the metacentric height. The metacenter is a calculation which helps define initial stability in a vessel. PCSail computed GM as:

$$\overline{GM} = \frac{RM_2 \cdot {}^{180}\!/_{\pi}}{\Delta} \tag{1}$$

While acceptable for trend studies, this approximation did not provide the righting moment accuracy within the tolerance needed for this study. The hydrostatics program FASTSHIP® evaluated the stability of the Mk II Navy 44 STC. GM was calculated for use in the VPP, and 4.94 ft was used instead of the PCSail calculated GM of 4.27 ft. This indicated that the vessel was more stable than predicted by the trend from PCSail.

Secondly, yaw angle and rudder angle determinations were added to the VPP. PCSail had already calculated the total aerodynamic lift from each sail. From the supplied rig diagrams, the longitudinal center of effort of each sail was approximated at the geometric center of the sail. From these two values, the yawing moment on the boat due to aerodynamic forces was found. From the lines plan, the center of effort on each appendage was approximated so that the lift force required on each appendage could be found. From these values, the necessary angle of attack of each appendage was found.

Another small modification was that the VPP used the average freeboard height as the height of the base of the mast. PCSail initially calculated that the total height of the rig was approximately 6 inches less than the true height. This made the righting moments and wind speeds at the height of the sails lower than they should have been. The height of the base of the mast was modified to reflect the true base mast height.

VPP Results

The VPP was run at wind speeds ranging from 6 to 24 knots. The PCSail output corresponded extremely well with most of the IMS VPP output. There was a fairly consistent discrepancy at low-wind speeds. This was expected since the PCSail aerodynamic algorithm is not as rigorous as the IMS algorithm at these low wind speeds. Figure (5) shows this discrepancy in light air especially for heel angles at reaching angles (120-170 degrees)[2]. This sailing condition is very sensitive to apparent wind effects as the sails are often stalled and requires more complex wind calculations for more accurate results. However, most of the range had to be checked against their respective velocities-made-good (VMG), which is the speed of the vessel on the wind-axis. For instance, if leeway is neglected at a true wind direction of 90 degrees, the VMG is zero. By choosing the maximum absolute value of velocity-made-good, the wind angle at which a boat can most optimally travel upwind or downwind can be found. Most of the deep-reaching angles were at angles past the best VMG, and therefore were out of the range of the optimum sailing angle.[9]

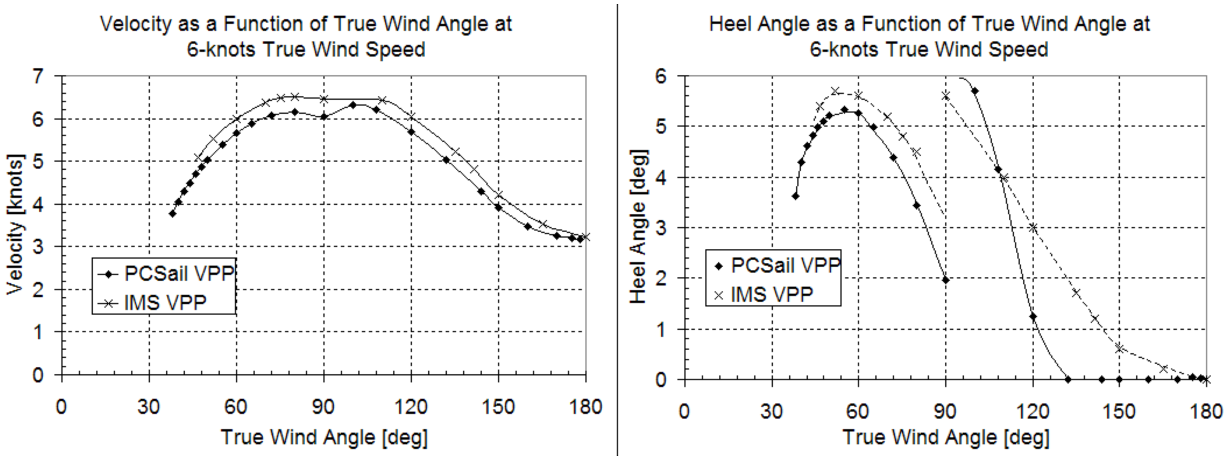


Figure 5 - PCSail VPP compared to IMS VPP at 6-knots wind speed

The PCSail VPP seemed to disagree with the IMS VPP for heel around a true wind angle of 90 degrees. This disagreement was based on the two algorithm's different interpretations on when to raise and lower the jib and spinnakers. Therefore, the heel angle was not a continuous function of true wind angle since a spinnaker and jib have much different effects on heel. Velocity for jib and spinnaker, however, was nearly equal at the point which the VPP chose to switch sails. Velocity was considered continuous as a function of true wind angle even while performing sail changes.

[2] Wind angles are reported as the true wind angle aft of the bow. The bow is an angle of 0 degrees, wind from the beam is at 90 degrees, and wind from the stern is at 180 degrees.

For medium to high wind speeds as in Figure (6), the PCSail's data came very near the IMS VPP's data. Even heel angle showed much better correlation between the two VPPs.

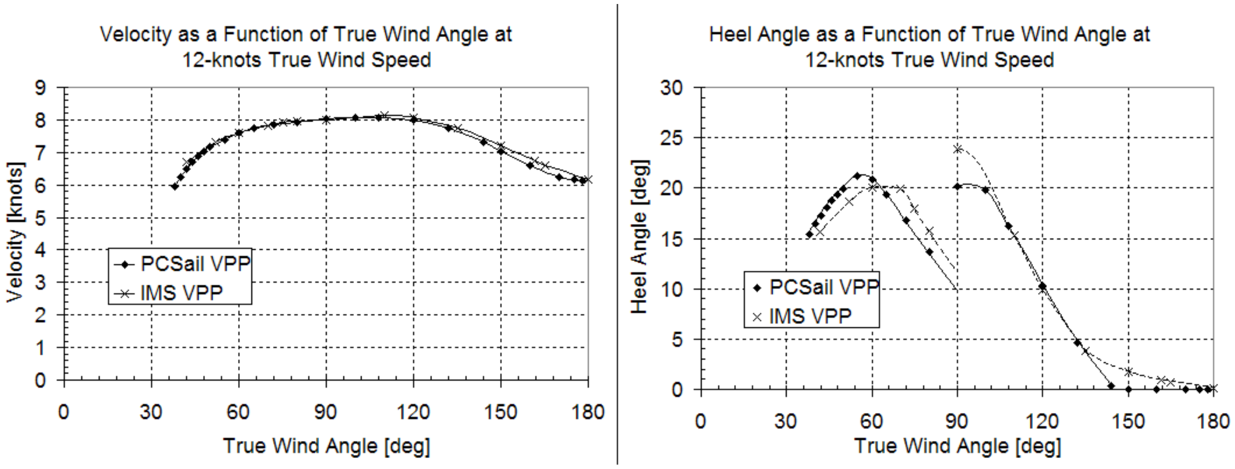


Figure 6 - PCSail VPP compared to IMS VPP at 12-knots wind speed

TANK TESTING

Tank testing is the traditional method of predicting the performance of vessels. Since it is far too expensive to build full-size ships to test different hull forms, models are built for testing instead.

Performance Prediction Theory

Performance prediction tank testing accomplishes one goal: determining the forces on the hull at different attitudes and speeds. To accurately predict forces, the model and ship need dynamic similarity. There are three parts to dynamic similarity: scaling of model sizes, model speeds, and model forces. [4]

Geometric similarity is the linear scaling of the model to the ship (Eqn (2)). This is met when all characteristic lengths of the model (L_{model}) are made proportional to the ship (L_{ship}) by the scaling factor (λ).[4]

$$\frac{L_{ship}}{L_{model}} = \lambda \tag{2}$$

Kinematic similarity is the scaling of speeds (Eqn (3)). Kinematic similarity[4] requires that the ratio of velocities of the model (V_{model}) and ship (V_{ship}) are the same.[14]

$$\frac{V_{ship}}{V_{\text{mod}el}} = \frac{\sqrt{L_{ship}}}{\sqrt{L_{model}}} = \sqrt{\lambda} \tag{3}$$

Dynamic similarity is the scaling of forces (Eqn (4)). In the tank, hull resistance is the force of the water acting on the ship. All forces (F) can be decomposed into their non-dimensional coefficient (C).[14]

$$C = \frac{F}{½\rho S V^2} \tag{4}$$

Besides force, a coefficient is dependent on the mass density of the fluid (ρ), the velocity (V) of the fluid relative to the characteristic length (L) of the body, and the characteristic wetted surface area of the hull (S).

Total resistance coefficient (C_T) is the summation of viscous, wavemaking, air, and added wave resistances (Eqn (5)). The primary resistances are viscous (C_V) and wavemaking (C_W), and the others are omitted for simplicity. [14]

$$C_T = C_W + C_V \tag{5}$$

The viscous resistance coefficient (C_V) is a function of Reynolds number (R_e) and the frictional resistance coefficient (C_f). Reynolds number defines the speed of the vessel relative to the condition of fluid flow. It is the ratio of inertia forces to viscous forces (Eqn (6)).[4]

$$R_e = \frac{VL}{\nu} \tag{6}$$

Reynolds number is dependent on the velocity of the fluid, the characteristic distance of fluid flow over the body (L), and the kinematic viscosity of the water (ν).

There is one modification to the Reynolds number calculation which is exclusive to sailing craft. For normal non-sailing vessels, the flow over the body (L) is typically the length of the waterline. For sailing craft, the keel and rudder make a large difference in frictional resistance. Moreover, the local Reynolds number for both keel and rudder are much less than the Reynolds number across the entire hull. Therefore, the flow conditions across much of the wetted surface area for sailing craft is at a lesser Reynolds number than if the whole body was the characteristic flow length. The result is an approximation to Reynolds number (R_n) for sailboats: a factor of 0.7 is added to the formula for Reynolds number (Eqn (7)).

$$R_n = 0.7\frac{VL}{\nu} \tag{7}$$

The frictional resistance coefficient (C_f) can be closely approximated without modeling by using the standard 1957 International Towing Tank Conference (ITTC) ship-model correlation equation (Eqn (8)). The frictional resistance equation is a function created through the curve fits of many years of experimental data.[14]

$$C_f = \frac{0.075}{\left(\log_{10} R_n - 2\right)^2} \tag{8}$$

The viscous resistance equation includes the skin friction term and a form factor (k). The form factor accounts for added resistance from the shape of the vessel. This reduces to an equation for the viscous resistance coefficient (Eqn (9)).[14]

$$C_V = \left(1 + k\right) \cdot C_f \tag{9}$$

The wavemaking resistance coefficient (C_W) is a function of the non-dimensional Froude number (F_n). Froude number defines the speed of the vessel relative to the type of wave systems created. It is the ratio of inertial forces to gravity forces (Eqn (10)). [4]

$$F_n = \frac{V}{\sqrt{gL}} \tag{10}$$

Froude number is dependent on the velocity of the fluid, the acceleration due to gravity (g), and the characteristic distance of fluid flow over the body.

There is no equation for the wavemaking resistance coefficient, and it remains a function of only Froude number (Eqn (11)).[13]

$$C_W = f\left(F_n\right) \tag{11}$$

Since it is assumed that only viscous and wavemaking resistances are significant, and there exists an equation for the viscous resistance coefficient, an equation of resistance can be defined in Eqn (12). Wavemaking becomes the only unknown variable in the resistance equation and is found experimentally through tank testing or CFD. [14]

$$R_T = \frac{1}{2}\rho S V^2\left((1+k)\left(\frac{0.075}{\left(\log 10\left(R_n\right)-2\right)^2}\right)+C_w\right) \tag{12}$$

Hydrodynamic Fluid Flow Theory

The flow condition around a ship is composed of three basic parts: turbulent flow, laminar flow, and separated flow. [5]

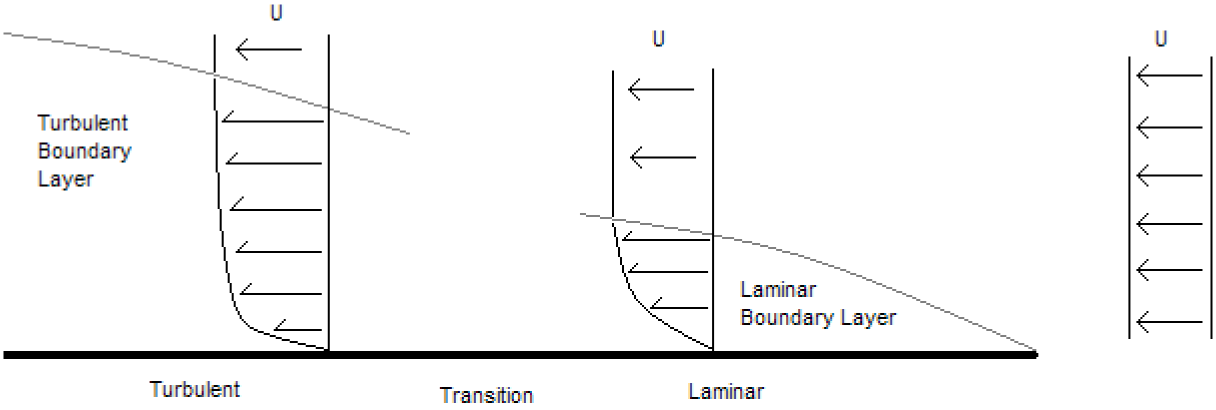


Figure 7 - Boundary layers[3]

The aerodynamicist, Ludwig Prandtl, developed the concept of the boundary layer for fluid flow (Figure (7)). Flow in the region surrounding a body can be divided into the boundary layer and anywhere outside of the boundary layer. Outside of the boundary layer, the viscosity on the body is negligible, and the flow may be treated as inviscid. Therefore, only flow inside the boundary layer is important for determining frictional resistance.[3]

The laminar boundary layer can be assumed to have a parabolic velocity profile "U" as shown in Figure (7). At the body, the velocity of the fluid is zero. Reynolds number, pressure gradient, and surface roughness determine the length of the laminar boundary layer and disturbed flow. For a flat, smooth plate in undisturbed flow, the flow typically moves from laminar flow to the transition zone around Rn = 500,000.[1] A description of laminar flow is that it is unmixable, meaning that if the flow were to oscillate between positive and negative flow directions, a fluid particle would always remain on the same streamline.[3] The coefficient of skin friction on a flat plate in laminar flow is: [3]

$$C_f = \frac{0.730}{\sqrt{Rn}} \tag{13}$$

Fluctuating velocities and eddying motion characterize the turbulent boundary layer. Compared to the laminar boundary layer, the turbulent boundary layer is thicker, and the velocity profile of the flow increases much faster away from the surface. However, before the flow enters into the turbulent region, it passes through the transition zone (Figure (7)). The transition zone contains sudden spots of turbulence generation and a general breakdown of laminar flow. Introduction into the transition zone is primarily a function of Reynolds number, but surface roughness and disturbed flow can also trip the condition from laminar to turbulent. [1] The coefficient of skin friction on a flat plate in turbulent flow is: [3]

$$C_f = \frac{0.0576}{\sqrt[5]{Rn}} \tag{14}$$

A special region inside the turbulent boundary layer is called the viscous sublayer (Figure (8)). The viscous sublayer is mainly laminar flow, but for a typical sailboat, is only 0.1 mm thick. The importance of the layer's size is that it is affected by surface roughness, and the layer is liable to trip back to the turbulent boundary layer in enough disturbance.[5] When laminar, the viscous sublayer significantly decreases the skin friction on the vessel.

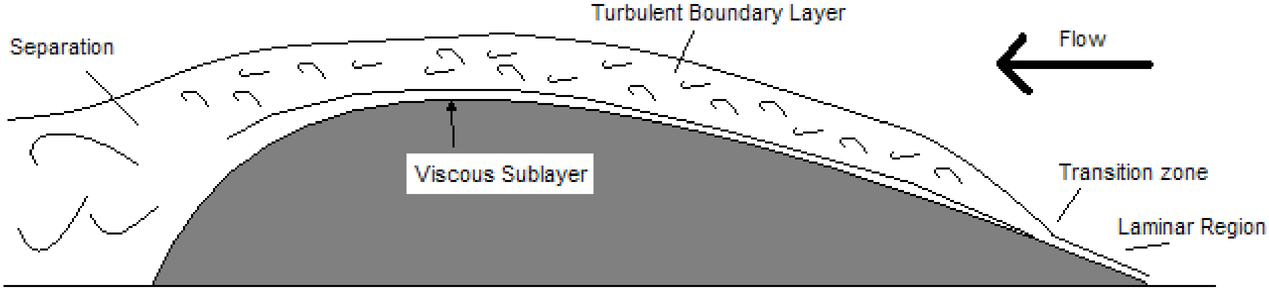


Figure 8 - Exaggerated flow conditions on a typical sailing vessel[5]

The separated zone is typically found near the stern of a vessel and is a function of the pressure gradient. For a body that is not a flat plate, the pressure gradient will remain positive as the form of the body increases (from the bow to after amidships on a typical sailboat). However, as the form of the body bends inward, the velocity profile will still try to cling to the surface. The resultant is that the flow will invert under this negative pressure gradient. The inverted flow will cause large eddies much larger than the eddies found in normal turbulent flow. [5] Separated flow drag is significantly higher than turbulent flow drag.

For tank testing, the flow condition is a function of Reynolds number, the roughness of the surface of the hull, and the turbulence in the surrounding water. For the full-size Mk II Navy 44 STC, the flow condition along most of the hull is turbulent as shown in Figure (8). The flow condition around most of the model would be laminar since the model is small, the tank water is generally stable, and the model surface is very clean. Although a model can be scaled geometrically from a ship, the water and therefore the flow condition cannot. If resistance data with a model in mostly laminar flow were scaled to full-scale, the predicted resistance would be significantly lower than actual.

Keeping Reynolds and Froude numbers constant defines dynamic similarity in terms of viscous and wavemaking resistance respectively.[4] Therefore, if both Reynolds and Froude numbers of the model matched those of the ship, then the model's behavior would be scaled identically to the ship's behavior. However, because the acceleration due to gravity and the kinematic viscosity of water cannot be scaled at the same time, perfect modeling conditions cannot exist. In some cases, Reynolds number similarity is used in a wind tunnel for testing submerged appendages. For most hydrodynamic tank testing, however, Reynolds number similarity is neglected because it would require the model to travel at speeds much greater than a tow tank could provide. Froude number similarity is used because the model's speed is slower than the ship's speed. Furthermore, viscous resistance is fairly well predicted using skin based methods (such as the given ITTC equation) while the wavemaking resistance is generally more sensitive to the details of the hull form and thus must be determined for each new design. Using the Froude hypothesis, the forces on the model can be predicted for the full-scale ship.[14]

The Froude hypothesis states that the wavemaking coefficient is the only variable which can be directly equated between the ship and model. In accordance, for each test performed in the tank, the Froude number of the extrapolated ship's speed is equal to the Froude number of the model. Extrapolation is begun by decomposing the tank test data into the coefficient of wavemaking. Thereafter, the Reynolds number as well as the coefficient of viscous resistance is recalculated for the ship. The new viscous resistance term, the wavemaking term, and a correlation allowance are summed to produce a new coefficient of total resistance for the ship. The correlation allowance mainly takes into effect the discrepancy between the smooth-painted hull of the model and the rough, biologically coated hull of the ship. Typically, the correlation allowance is 0.0004, although this varies in accordance with the assumed roughness of the ship's hull. [14]

Once the coefficient of total resistance for the ship is calculated, the total resistance force is calculated, and the ship prediction is completed. According to dimensional analysis, geometric force scaling can be performed by multiplying the model force by the cubed scaling factor. If this is done for resistance tests, the ship prediction will be over-estimated. The reason for this is that although there is a greater amount of viscous resistance for the ship due to its larger wetted surface area, the coefficient of viscous resistance is smaller. Turbulent flow theory predicts smaller coefficients of viscous resistance for larger Reynolds numbers. Since the ship has a much larger length and travels at larger velocities, the Reynolds number for the ship is larger than for the model. The ITTC resistance equation which is used in this extrapolation reflects the differences in the viscous coefficient. [14]

Model Preparation

The tank used in testing the Mk II Navy 44 STC was the 120' tow tank located in USNA's Hydromechanics Laboratory. The towing rig on the 120' tank required extensive preparation. The attitude of the model needed to be set in almost any condition of heel, yaw angle, trim, heave, and rudder angle (Figure (9)).

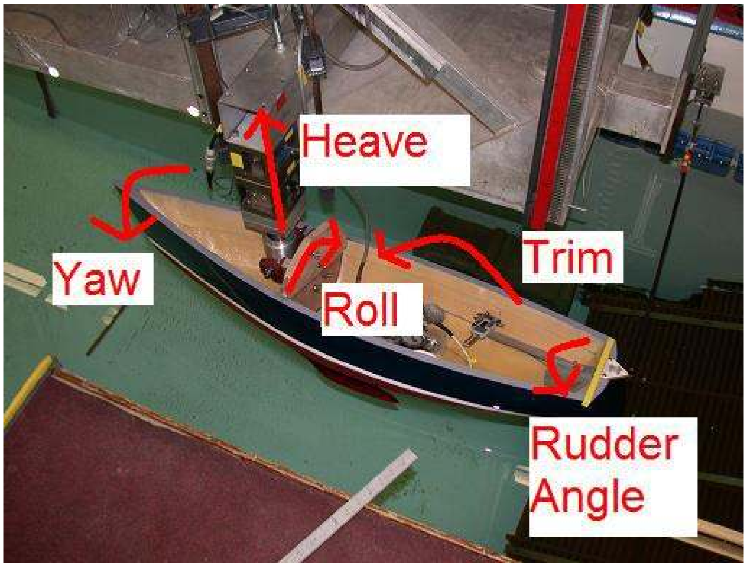


Figure 9 - Degrees of freedom in model testing

The requirements on the rig were therefore:

1) Infinitely adjustable in yaw from 0 to 10 degrees. Yaw is the angle at which the bow points when traveling through the water.
2) Infinitely adjustable in heel from 0 to 35 degrees (or deck submersion). Heel is the angle at which the boat rolls around the longitudinal axis of the boat.
3) Infinitely adjustable in rudder angle from 0 to 10 degrees.
4) Infinitely adjustable in initial trim until deck submersion. Trim is an angular measurement of how either the bow or stern is submerged relative to a lateral axis through midships.
5) Free to trim.
6) Free to heave. Heave is the vertical movement of the entire boat into or out of the water.

Because of the forces and attitudes specific to sailing craft, more instrumentation than just drag was needed. For the Navy 44, the recorded data included:

1) Drag parallel to motion of travel
2) Lift perpendicular to motion of travel
3) Yawing moment
4) True speed
5) degrees of change in trim

The model was received from the USNA Model Shop built to the lines supplied by Pedrick Yacht Design. The model was measured and put to a position of 0 degrees heel and trim to find midships and the centerline. A small aluminum plate was mounted to be flat at this baseline upright condition. Lines of centerline and midships were etched into the plate.

A bulkhead was installed in the model a few inches forward of midships. Attached to the aft side of the bulkhead were two aluminum plates which could be adjusted so that the model was set in a position of heel. Forward of the aft plate was a cylinder of bearings which housed the heave post and left the model free to trim (Figure (10)).

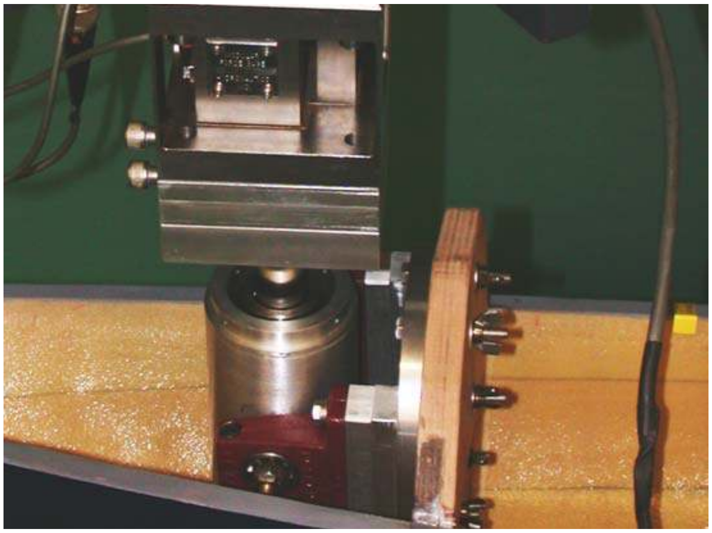


Figure 10 - Bulkhead and heave post

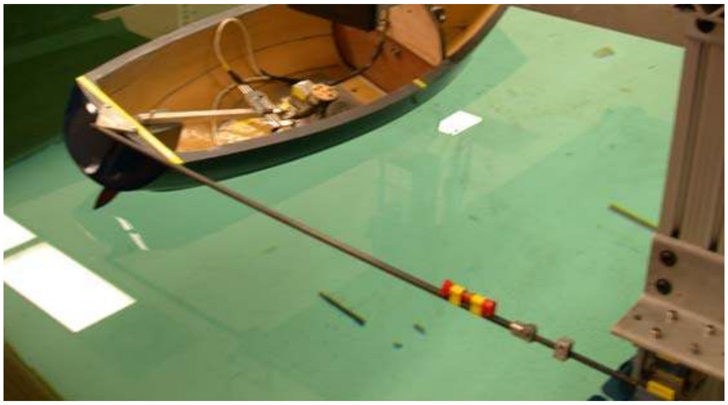


Figure 11 - Yaw adjuster

A pin was attached to the transom on centerline. This connected to a stiff carbon fiber rod which could be extended and was itself connected to the tow rig. The rod allowed the model to be set in a yawed position (Figure (11)).

A 4" tiller was attached to the rudderpost. At the end of the rudder was a pin which connected to an electronic caliper. The other end of the caliper was pinned to the hull. The caliper allowed for extremely precise measurements in rudder angle (Figure (12)).

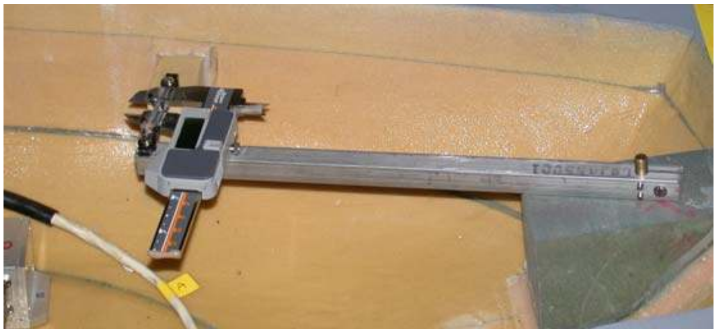


Figure 12 - Rudder adjuster

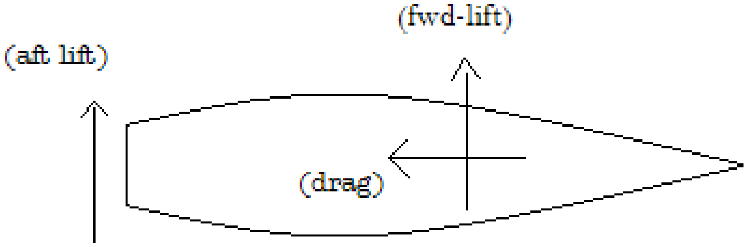


Figure 13 - Schematic indicating measured forces

Next, three force blocks and two inclinometers were calibrated. Force blocks measured resistance in the direction of travel (drag), lift perpendicular to motion at the heave post (fwd-lift), and lift perpendicular to motion at the aft pin (aft-lift). The directions of these forces are shown in Figure (13). Inclinometers would measure change in trim and heel.[3] The aft-lift block would be used to measure the longitudinal (yawing) moment. All force blocks were calibrated using standard weights up to twice the expected loads. The inclinometers were calibrated using a stand-alone electronic inclinometer on an adjustable surface. Both inclinometers were calibrated to their rated maximum incline.

The drag and lift force blocks were both rated to a 25 lb capacity. This was to minimize the effect of cross-moments interfering with the force acquisition. The aft-lift block did not have any interfering moments and was rated at 5 lbs. In the calibration for all force blocks, the maximum linear deviation was 0.116% on the 5 lb aft-lift block. This corresponded to a standard deviation of 0.0037 lbs, which was considered very acceptable.

The heave post was attached to the drag block, mounted underneath the lift block, and finally secured to a square aluminum beam on the carriage. The aft-lift block was mounted to an "L"-shaped beam, which was clamped onto the after end of the carriage. The inclinometers were mounted to the aluminum plate inside the model. The speed of the carriage had been previously calibrated and was measured at the motor's gears.

After the inclinometers were added, the model was weighed and trimming weights were added in order to scale the model geometrically. Placing the model in the water, the weights were moved so that the model had no trim. The position of these weights marked 0-degrees of trim, and a ruler was drawn on the centerline to measure the trimming arm induced by the weights. Figure (14) shows the final model setup:

[3] The heel of the vessel was set through thumb screws mounted to the bulkhead. An inclinometer was set to measure any change in the heel of the vessel – which would not be desired. The acquired data from the tests showed that the heel did not change.

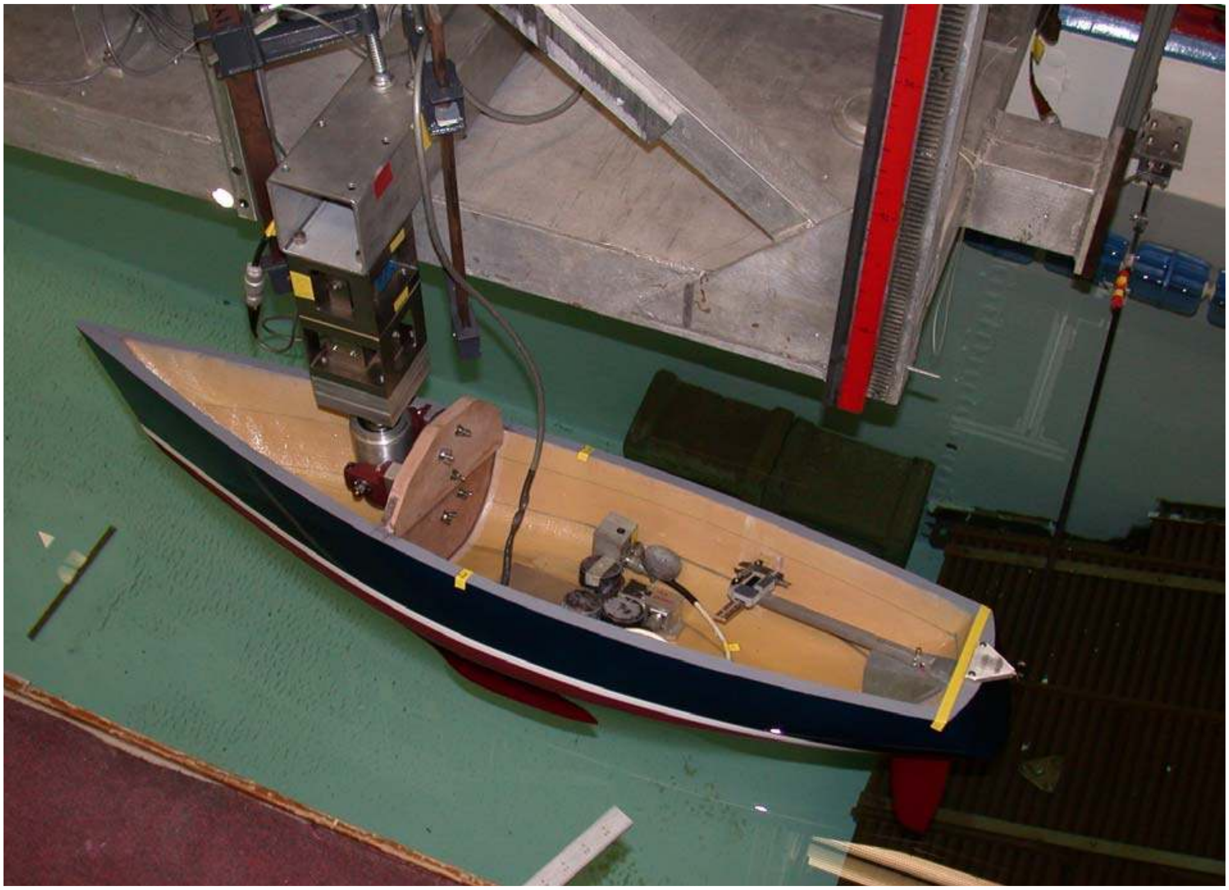


Figure 14 – Model ready for testing

All force blocks and inclinometers had wires which ran to two amplifiers mounted on the tow rig (Figure (15)). The data then ran from the amplifiers to an analog-digital converter and finally through a USB connection to a PC. Before gage calibrations, the amplifier was set to return values in the linear region of output.

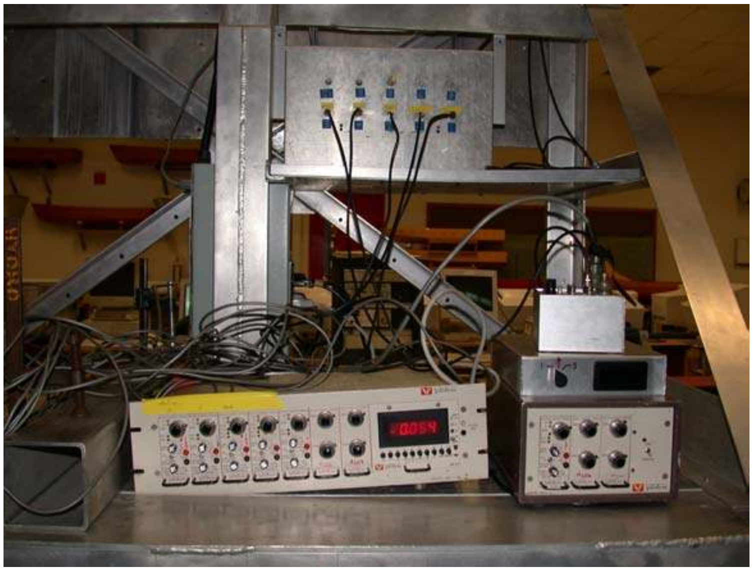


Figure 15 - Amplifiers mounted on the tow rig

The final stage in testing was to find the position of 0 degrees of yaw. The model was attached to the rig without the rudder and with 0-degrees heel and trim. The model was then run down the tank a minimum of four times where each trial varied by yaw. Model speed was held at a constant 3.2 ft/s. This speed was to ensure typical flow conditions across the appendages, but with minimal wavemaking interference.[4] Plots were constructed of inches of yaw adjustment versus pounds of side force. The position of 0 degrees of yaw was marked at the point of no side force.

Next the rudder was added, and the process of finding 0 degrees of rudder was identical to the above process, except the rudder angle was adjusted vice yaw. Again, the point of no side force marked the position of 0 degrees of rudder. This position of 0-rudder, 0-yaw, 0-trim, and 0-heel was considered the standard upright condition.

The typical standard deviation in calibrating the yaw and rudder angles was 0.019 lbs as measured by both lift blocks. These calibrations were checked every time the model was taken out of the water, which was at least after every day of testing.

[4] At higher speeds, the wavemaking effects would have produced large waves, causing the model to heave and trim substantially. This would disrupt the measurements taken in calibration.

Tank Tests

There were three parts of tank testing:

1) Unstimulated Upright Conditions
2) Stimulated Upright Conditions
3) Stimulated Sailing Conditions

The goal of unstimulated testing was to find a baseline for the evaluation of the later-added turbulence stimulators.

Stimulators were added to the model to try to create turbulent flow condition around the model. These stimulators were placed at a position to trip the flow where turbulent flow would begin on the ship. Details of these stimulators are discussed in the "Stimulated Upright Condition" section. Additional stimulators were added after all other testing was completed in order to factor out the added resistance due to the stimulators.

The stimulated upright condition provided results which represent the motoring condition of the vessel. The upright resistance of a vessel is also typically used to judge its overall resistance against other vessels. In the upright condition, heel angle, yaw angle, and rudder angle were all set at zero degrees. The vessel was trimmed to its designated waterline and was left free to trim and heave while at speed due to hydrodynamic forces.

Stimulated sailing conditions allowed for calculations of the vessel at different attitudes. For sailing conditions, the vessel had a set heel angle, yaw angle, rudder angle, and trim angle. The heel, yaw, and rudder angles were set initially from a chosen matrix. Trim angle was set through trim weights which approximate the aerodynamic trimming moment. This trimming moment was determined experimentally through an initial run for each condition and equaled the countering measured hydrodynamic trimming moment. The model was free to heave in the sailing condition.

Upright Unstimulated Tests

The unstimulated testing was conducted at the standard upright condition. The coefficient of total resistance is shown in Figure (16). The generated curve maintained coherency above Rn=400,000 signaling turbulent flow. Below Rn=400,000, there was significant disorder in the resistance curve, showing that the flow condition was moving from laminar through the transition zone.

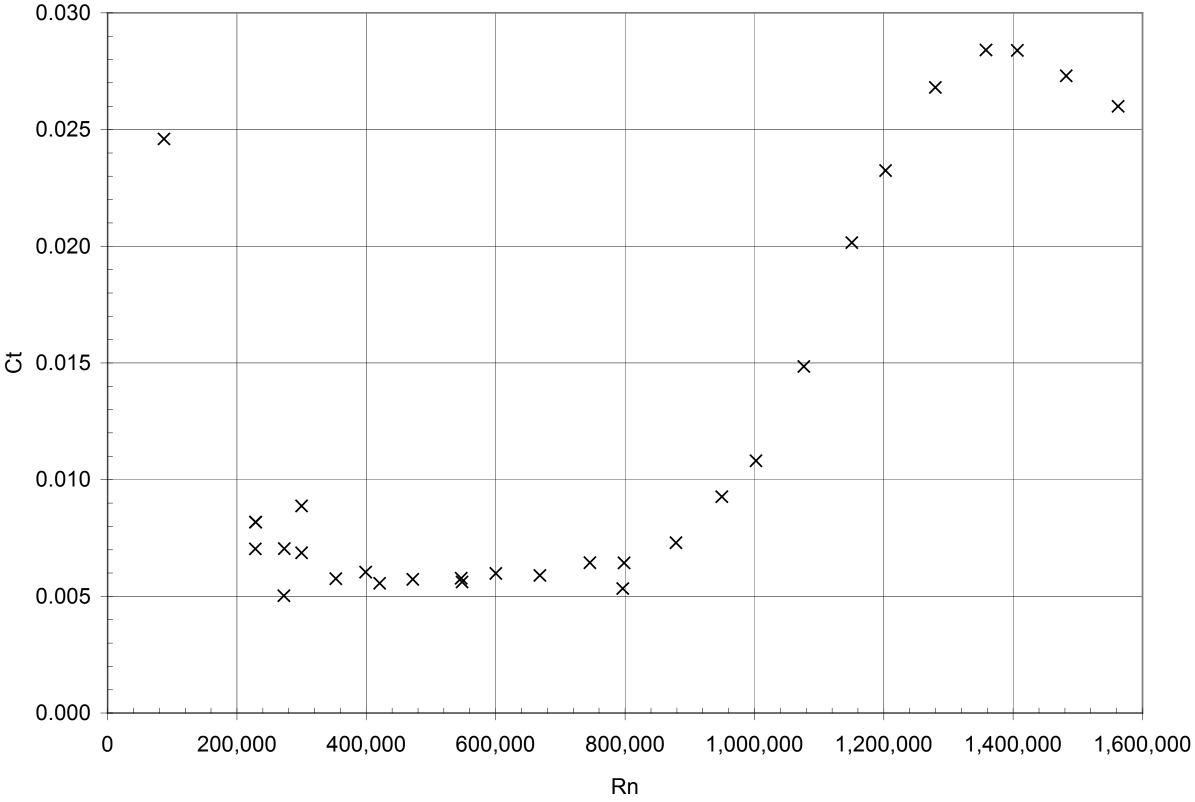


Figure 16 - Drag in upright unstimulated flow

Upright Stimulated Tests

After the unstimulated flow testing was completed, sand strips were applied to the hull to act as turbulent stimulators. Sand strips were placed on the hull 1" aft of the stem and on the keel and rudder at 25% of their chords' girth. A grain size of 0.5mm to 1.0mm was used and at a width of a 0.25". The grain size was chosen to be as large as the laminar boundary layer in order to disrupt the flow sufficiently. This disturbed boundary layer should change from laminar to turbulent flow. The upright stimulated testing was conducted at the standard condition.

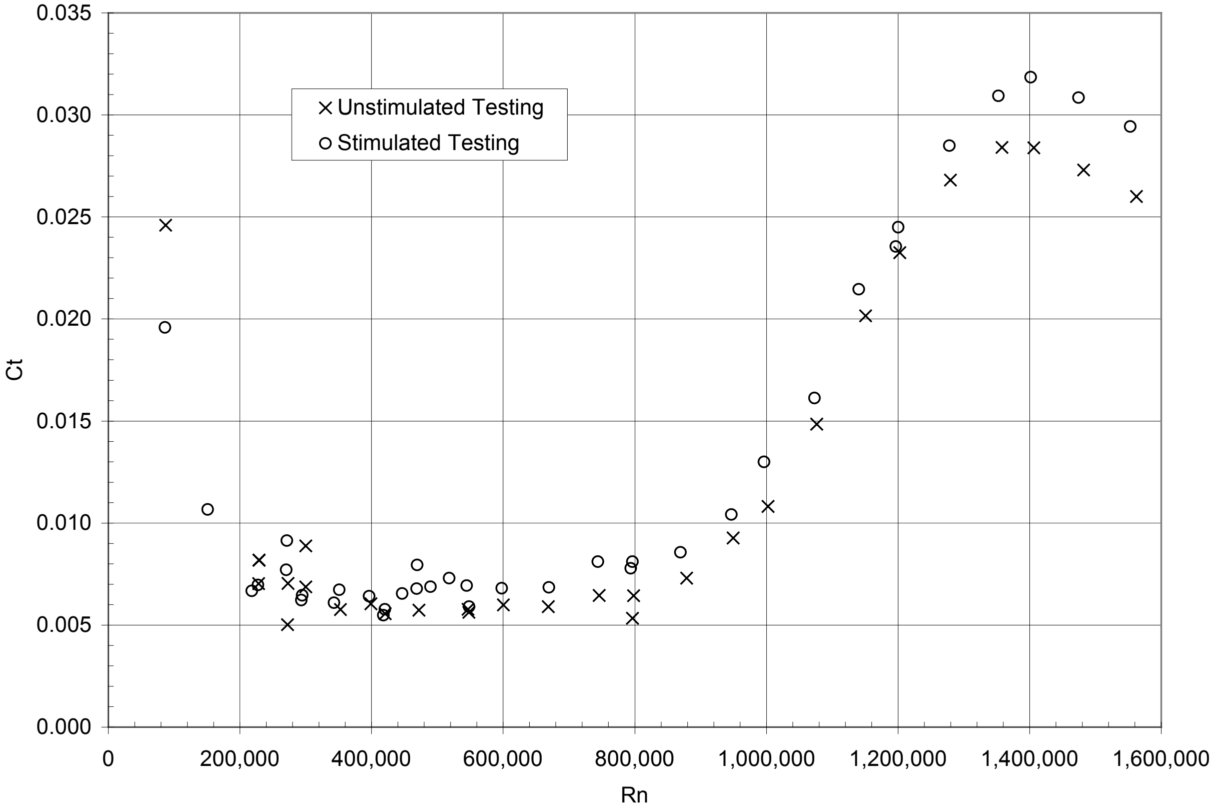


Figure 17 - Drag in upright conditions

The results of both unstimulated and stimulated conditions are shown in Figure (17). The stimulated condition showed a uniform increase in total resistance. The added resistance was from the friction caused by the sand strips and also indicated that the flow condition did not substantially change at Rn > 400,000. The meaning of this was that it was considered safe to assume ship-like turbulent flow above Rn = 400,000. Stimulated tests performed below Rn=400,000 showed that the fluid flow was still in the transition zone on at least some part of the vessel.

Additional sand strips were added to the model to calculate the added friction from the strips. The first set of additional strips doubled the width of the previous strips and was located aft of and adjacent to the previous strips. The second set of strips brought the width up to three times the width of the original strips. The third set of strips was aft of and adjacent to the secondary strips. Testing of the strips populated only the lower Reynolds number conditions since the strips were used only for Prohaska analysis of the data.[5]

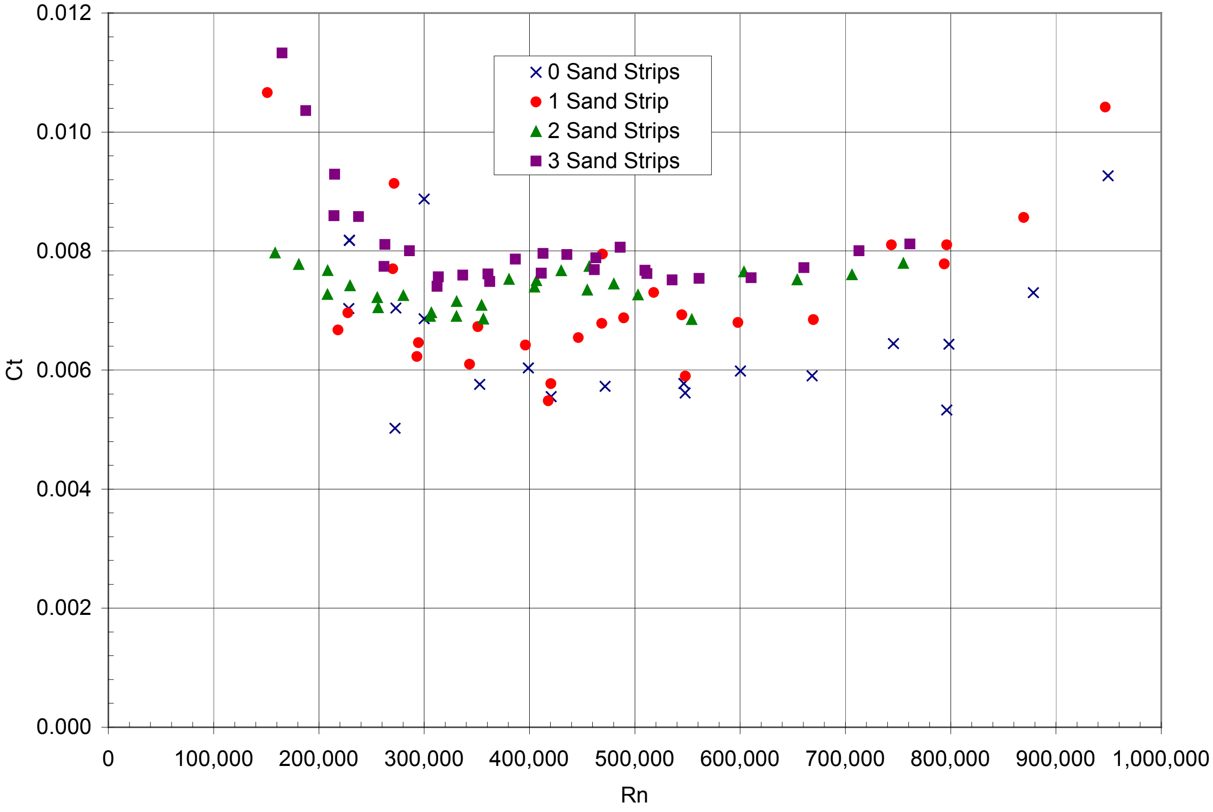


Figure 18 – Drag in the upright conditions with additional sand strips

Again, the additional sand strips show a general increase in resistance (Figure (18)). The trend in the resistance curve was maintained after Rn = 300,000. Below Rn = 300,000, separation between the second and third sand strip increased. This showed the presence of laminar flow even with the extra turbulence stimulators. Because of this, results below Rn = 300,000 with additional sand strips were questionable.

[5] Prohaska analysis will be discussed in the "Prohaska Theory" section.

Stimulated Sailing Tests

The sailing conditions were estimated from attitudes and conditions predicted by the PCSail VPP. This matrix varied by speed, heel angle, yaw angle, and rudder angle.

PCSail was run at true wind speeds of 6, 12, 18, and 24 knots and at 40, 60, 90, 120, and 170 degrees to the wind. The VPP determined the values of boat speed and heel angle for each of these 20 conditions. The values of speed and heel are shown in Figure (19).

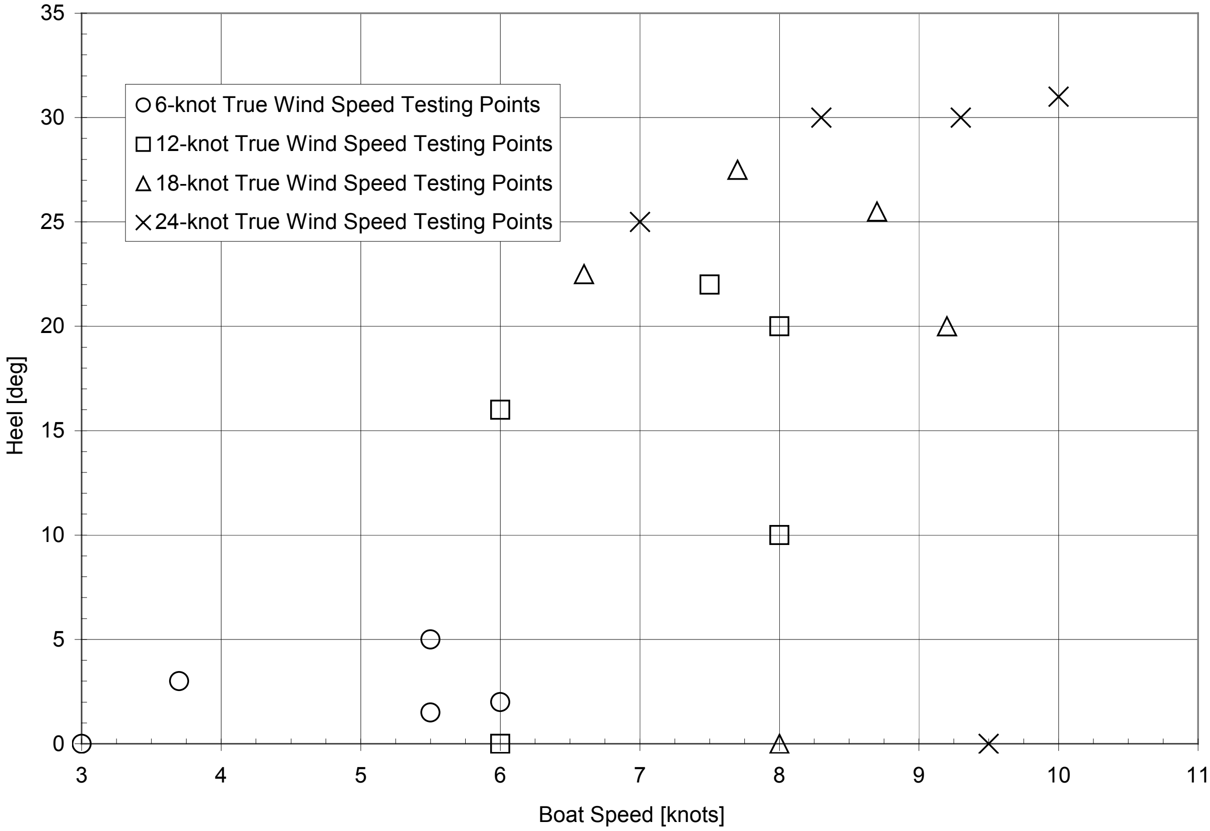


Figure 19 – Sailing tests matrix

The points picked for positions of speed and heel were near to the VPP's prediction. These points were picked so that the true sailing conditions could be interpolated from the matrix once the testing had concluded. Furthermore, the speed of the carriage could be set only approximately at its input dial. Therefore, the desired speed of each test differed slightly from the resulting speed of the carriage.

Each sailing test had a predicted yaw and rudder angle from calculations added to the PCSail VPP. However, as the sailing testing was conducted at first in unstimulated flow, it was found that the aft-lift block was recording negative forces on the model which was opposite of the predicted aerodynamic force. The modifications to the PCSail VPP had estimated the angle of attack on the keel and rudder from their calculated coefficients of lift, while the lifting force due to the canoe body was assumed zero. The PCSail modifications to find rudder angle and keel angle proved to be incorrect. This was due either to the lift caused by the canoe body itself or from unusual flow patterns at the rudder possibly due to laminar separation.

The result of this finding was two-fold. First, the unstimulated testing was adjusted to include only the upright condition. Secondly, the test-matrix was reevaluated so that true sailing conditions could be found.

A test was performed to find how yaw affected the boat's characteristics. From this test of yaw shown in Figure (20), it was determined that for small angles of changes in yaw the lift would change linearly. In the new test matrix, instead of trying to predict the yaw and rudder angles, set angles were chosen. Yaw angles of 0 and 4 degrees and rudder angles of 0, 3, and 6 degrees were used for each sailing condition. Ultimately, for the sailing matrix with 20 variations of speed and heel, there were a total of 120 tests which included rudder and yaw angle.

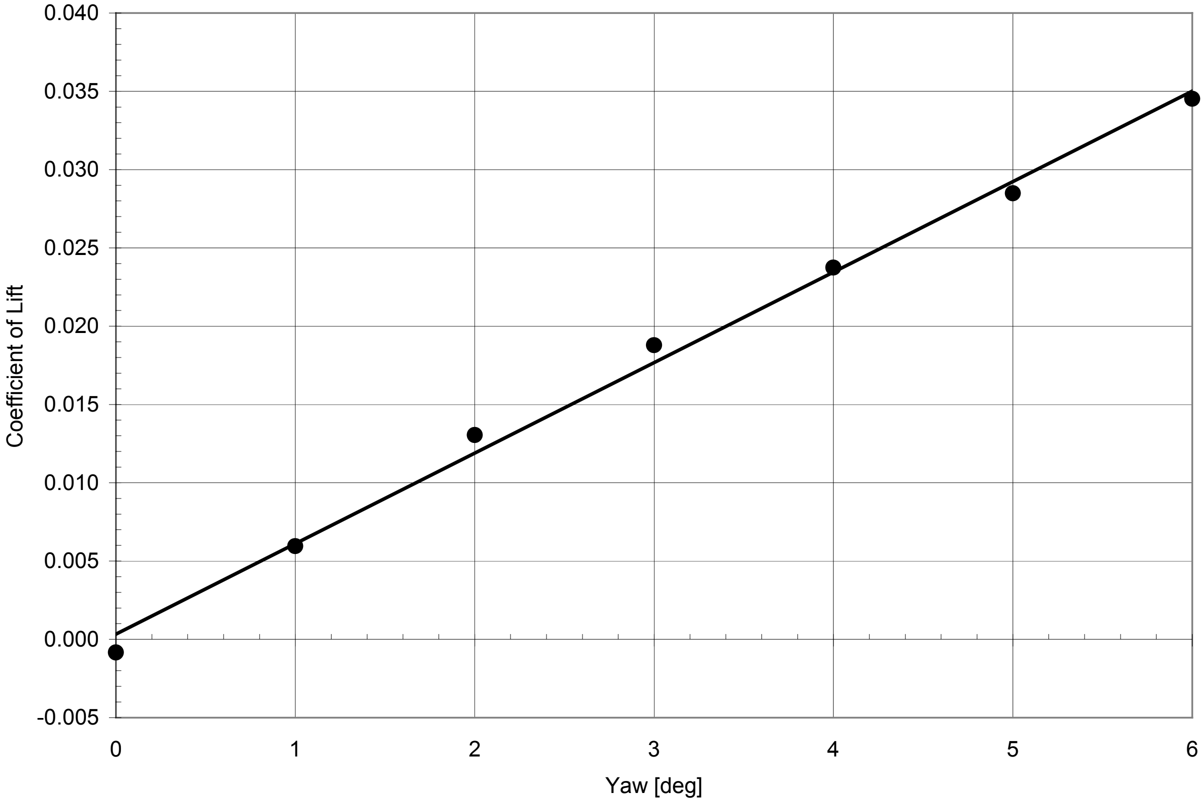


Figure 20 – Lift linearity for small changes in yaw

Prohaska Theory

The total resistance is assumed to be composed of viscous and wavemaking factors (Eqn (12)). Viscous resistance is composed of frictional resistance multiplied by a form factor (Eqn (9)). Because of Bernoulli's principle, an incompressible fluid has to change velocity while moving across a form which is not flat. The increased velocity of the fluid creates increased frictional resistance across the form, and that component is calculated in the viscous resistance term as the form factor, k.[14]

To calculate the form factor, a process similar to the Prohaska method is used. Prohaska suggested that the form factor could be reduced from the equation: [14]

$$C_T \big/ C_F = (1+k) + F_n^{\ 4} \big/ C_F \tag{15}$$

The International Tank Testing Committee in 1978 recommended modifying Prohaska's method to use the equation: [14]

$$C_T \big/ C_F = (1+k) + F_n^{\ n} \big/ C_F \tag{16}$$

where n is a power of Froude number being: $4 \le n \le 6$.

Typically, a plot is constructed where C_T / C_F are ordinal values and $F_n^{\ n} / C_F$ are abscissa values. The form factor is derived as an ordinal-intercept of the best-fit linear regression. Values used for computation should be at as slow a speed as possible without reaching into the scale-effects region.

Prohaska Analysis

The unstimulated and stimulated upright tests were analyzed using Prohaska plots to find the form factor of the model.

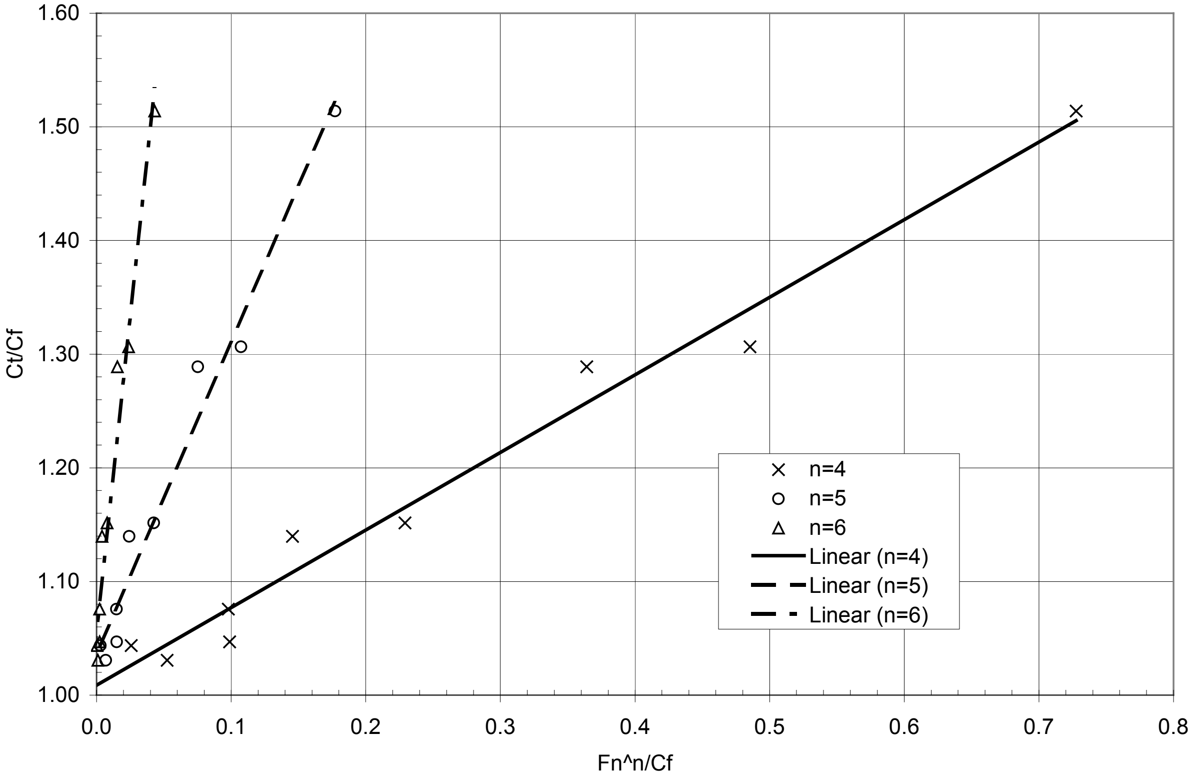


Figure 21 - Prohaska plot of unstimulated data

The data which showed linearity without scale effects was used for the Prohaska plots in Figure (21). The best-fit lines produced ordinal-intercepts of 1.009, 1.035, and 1.054 for n values of 4, 5, and 6, respectively. The $F_n^{\ 4} / C_F$ Prohaska plot produced the best-fit line, so the value of 1+k for the form factor in unstimulated flow was 1.009.

The same method was used for stimulated flow.

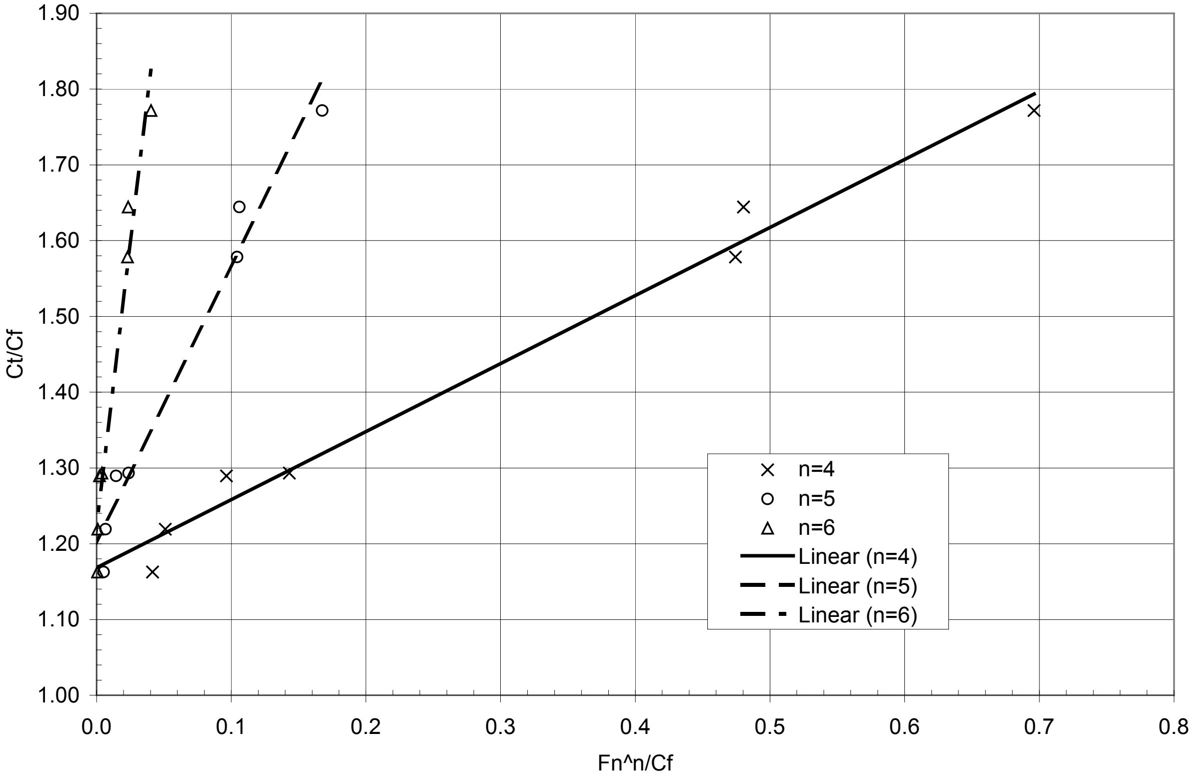


Figure 22 - Prohaska plot of stimulated data

The data was sorted to find good points for the Prohaska plots in Figure (22). The best-fit lines produced ordinal-intercepts of 1.168, 1.181, and 1.209 for n values of 4, 5, and 6, respectively. Again, the $F_n^{\,4}/C_F$ Prohaska plot produced the best-fit line, so the value of 1+k for the form factor in stimulated flow was 0.168. The value of 0.168 included both form factor and the effect of one strip of sand. This form factor was used to compute the viscous friction from the model to the coefficient.

Finally, the sand-strip stimulation data was analyzed.

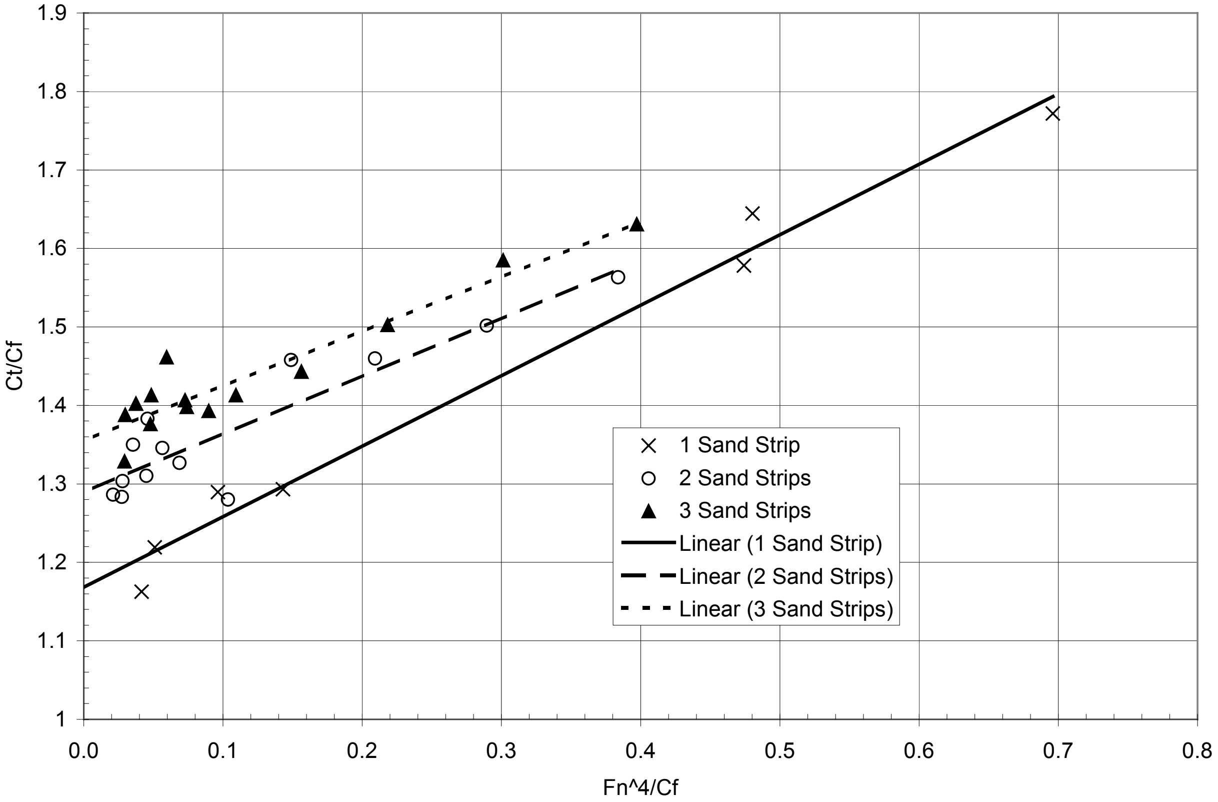


Figure 23 - Prohaska plot of additional sand-strips data

The data for both extra sand strips were analyzed for n values of 4, 5, and 6 (Figure (23)). The $F_n^{\,4}/C_F$ Prohaska plot fit the best line through both sets of data. The ordinal-intercept of the two sand strips plot was 1.290 and of the three sand strips plot was 1.356.

The difference in form factors between the additional sand strips gave the added form factor due to the sand strips. The difference between these values was 0.065. Assuming that the testing with only one sand strip indicated the true flow conditions around a full-size vessel, the difference from the sand strips was subtracted from the stimulated form factor. Therefore, the adjusted form factor for the hull was 0.103. The form factor of 0.103 was used to scale up the coefficient data to the ship scale.

An interesting look into flow affects was seen by superimposing the viscous coefficient over the total resistance coefficient for each sand strip condition (Figure (24)). The points to notice were at the intersections of each total resistance curve and viscous resistance curve. Most of these interactions occurred around Rn = 375,000. Data acquired at Reynolds numbers lower than 375,000 had significantly lower resistance than at points where Rn > 375,000. The data having lower resistance indicated a large amount of laminar flow, and did not properly model the flow conditions found on the full-size vessel.

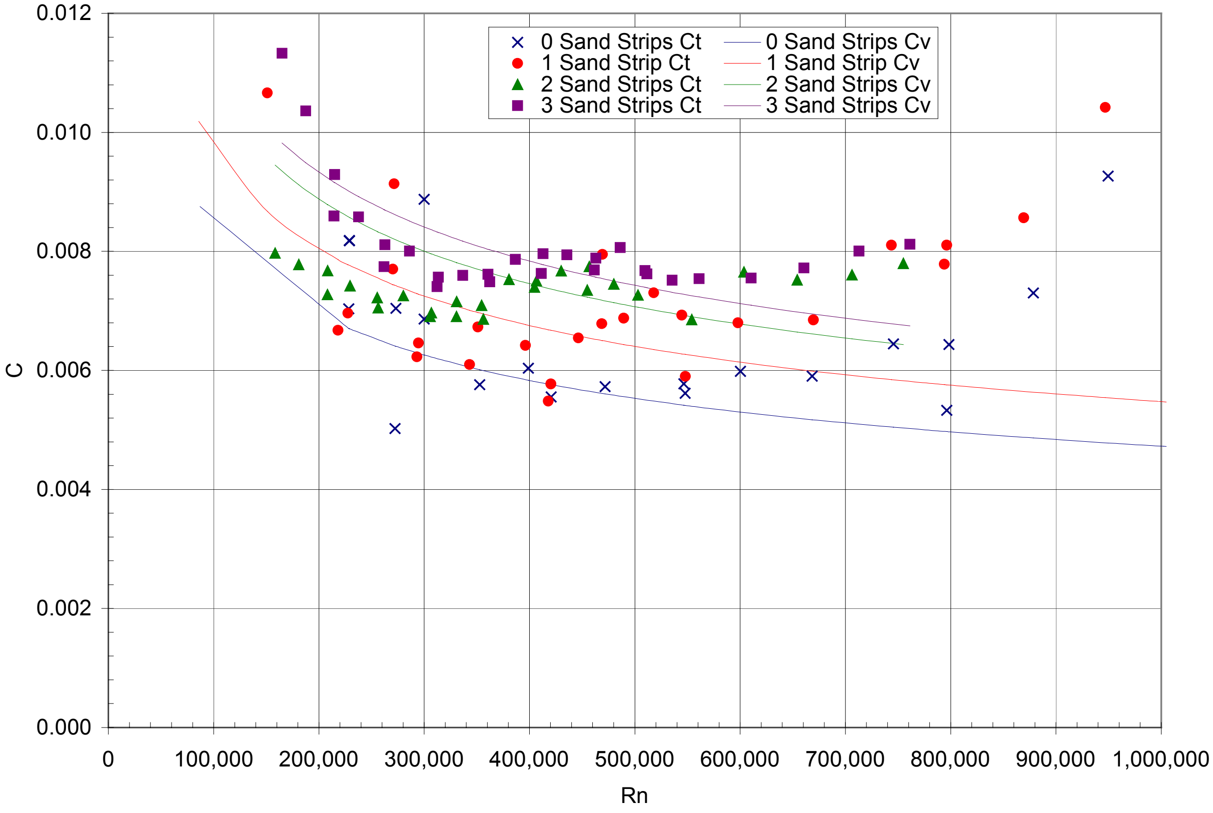


Figure 24 – Total and viscous resistance for all upright data

The exception to the rule of a decreasing coefficient of total resistance with decreasing velocity is below Rn = 250,000. These very-slow test points had entirely laminar flow, and the laminar friction coefficient dominated resistance. Eventually, these slow test points would drive the total resistance coefficient to a limit of infinity at low velocities (Eqns (8), (13)).

Since the form factor was found, the viscous component of resistance was subtracted from the total resistance to leave wavemaking resistance (Figure (25)). Notice that below a Froude number of 0.18 (Rn<400,000), there was a considerable amount of data which had a negative wavemaking coefficient. The negative wavemaking coefficient indicated that there was a considerable amount of laminar flow across the model at those points.

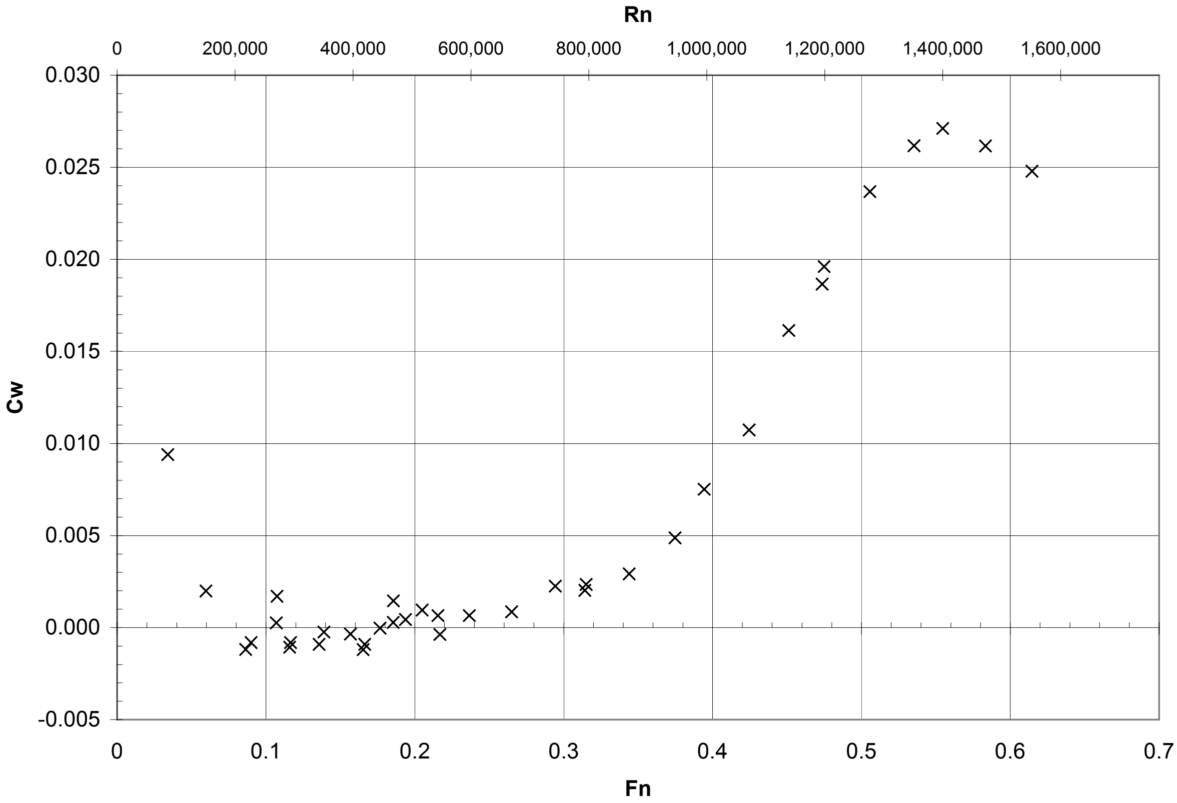


Figure 25 – Wavemaking resistance in the upright condition

Froude proposed that ship predictions could be made by scaling only the wavemaking resistance from model tests. The viscous component of resistance for the ship would be calculated using the form factor which was found for the model.

Powering required to pull a ship through the water is known as effective horsepower. Power is the product of force over a velocity, therefore, for English units power is:

$$ehp = {}^{R_T V}\!\big/_{326} \qquad (17)$$

where velocity is expressed in knots, and resistance is in pounds.

From the acquired upright data, both the resistance and effective horsepower were calculated for the full-size Mk II Navy 44 STC (Figure (26)).

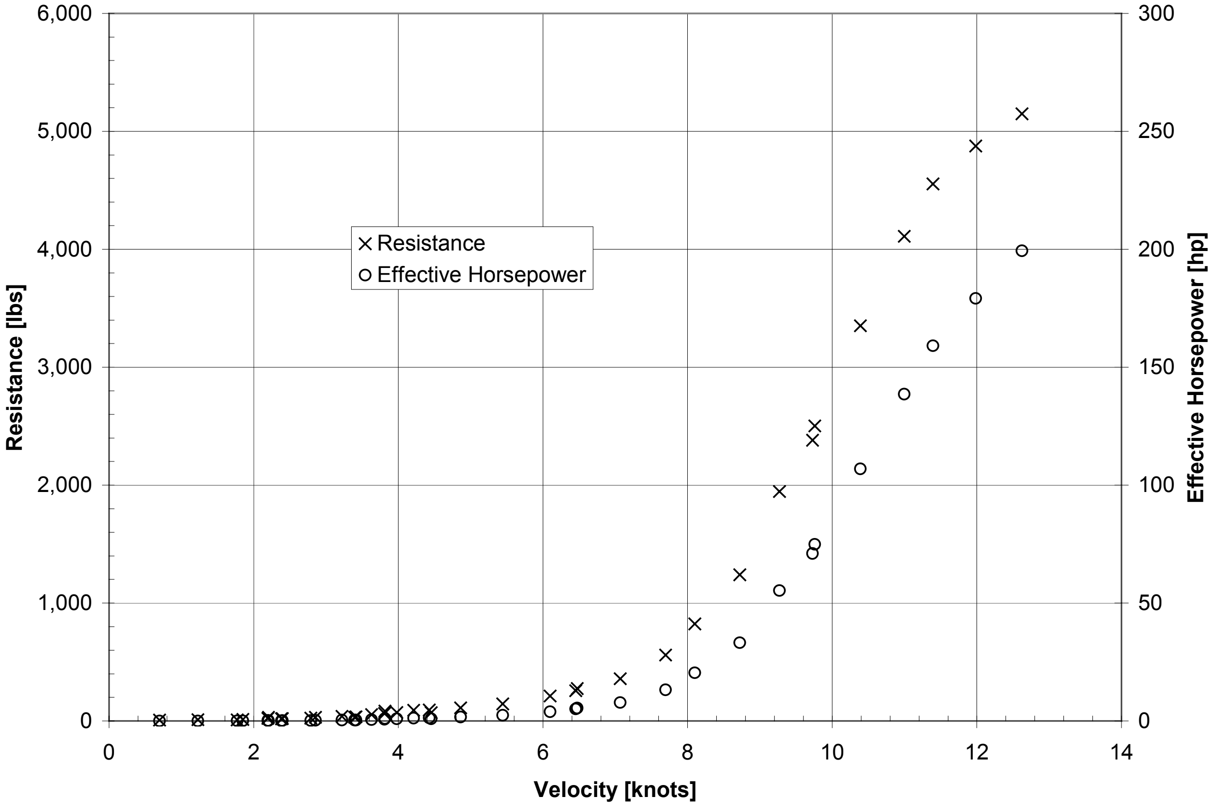


Figure 26 – Upright resistance and horsepower

Difficulties in Tank Testing

From the beginning of tank testing, there were two major problems. The first problem consisted in realignment. After the initial alignment, the rig was assumed straight. However, after checking the alignment again after some of the initial unstimulated flow cases, it was found that the square beam was not rigidly attached to the carriage. To solve this, both the square beam and the "L" beam were screwed into the carriage – not just clamped.

Another major difficulty throughout the project was the presence of transient noises in the system (Figure (27)). The result of this noise was to multiply the raw forces to many times the actual forces. Many Fourier-transforms were performed on the data to try to identify the source of the errors. The Fourier-transforms could not find a frequency of the noise which was distinguishable from the standard data frequencies.

Ultimately, it was shown from an oscilloscope that the noise had a beat signal. The source of the beat could not be found. However, the data acquisition system was modified to truncate the data near the zero-crossings of the beats. This method adjusted the starting and ending points of the data to find the lowest standard deviation. In between the truncation limits, the data was averaged to find the steady-state force without the influence of noise.

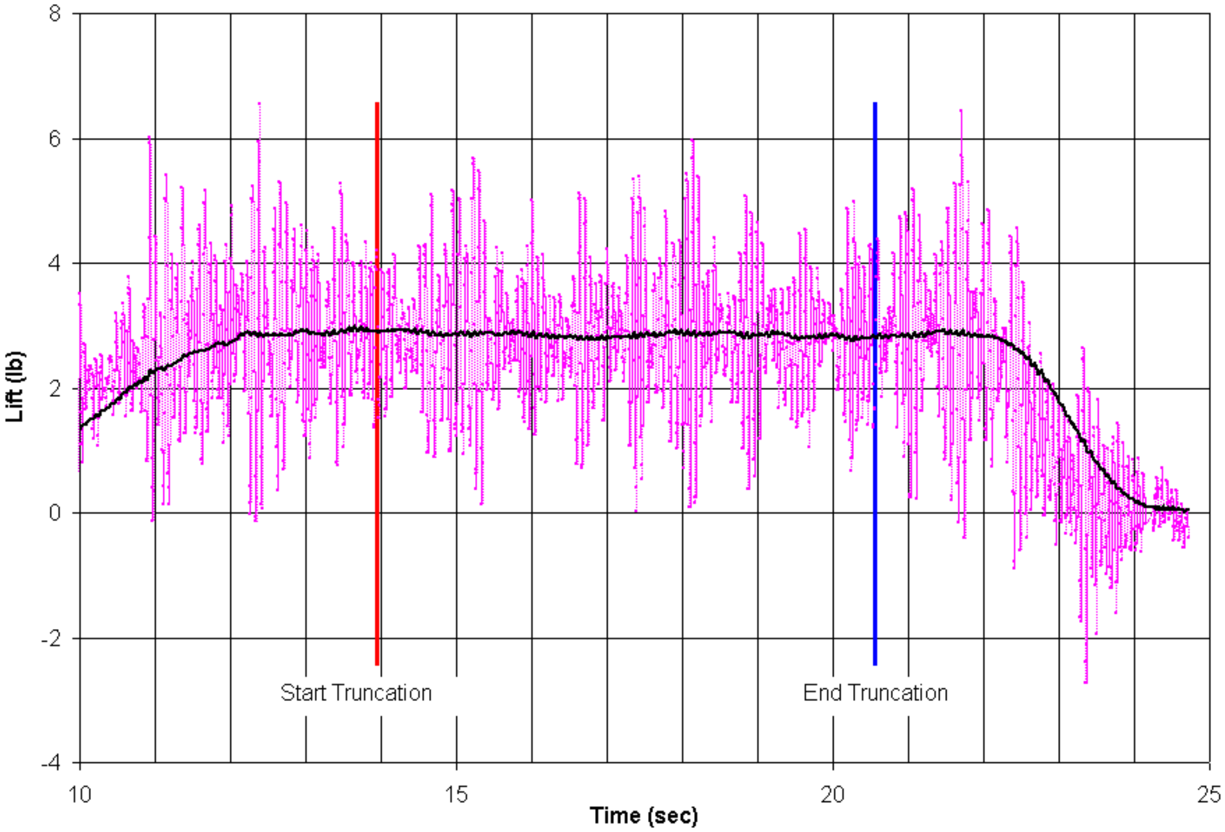


Figure 27 – Fwd-lift force block raw data for typical sailing test

Another work-around to reduce the error associated with the transient noise was to drastically increase the rate of sampling. Using this method, a greater percentage of the noise averaged itself out as sampling was increased from 25 Hz to 143 Hz. This method required moving to a laptop instead of a PC. This was because the PC's data acquisition was connected by a USB connection which could not handle the high sampling rate. The laptop used a parallel connection for data acquisition which could receive high data rates. On the higher data sampling system, the beat signal was visible in the data (Figure (27)).

FKS

FKS is a CFD code developed primarily from the work of Dr. Noblesse from NSWC-CD. FKS is called a "free-surface code" which describes its goal of predicting wave making resistance. The free-surface is the boundary between two fluids. For ships, this boundary is between the seawater and the air and mathematically is modeled as a continuous surface. In real life at the free-surface, non-continuum effects can take place. For example, a bow wave from a ship can often roll over on itself, or spray can form and break off free from the rest of the surface. Furthermore, free-surface codes are difficult to construct because little is known about the viscous relationship of wavemaking. FKS is an inviscid code, and this viscous interference is ignored, leading to some error in the solution. Determining this error was a major focus of the testing.

FKS Theory

As waves are formed near the surface of the hull, they radiate towards infinity. However, as they radiate, the waves interfere with each other. As an example of this interference at Fn=0.40, the bow wave has constructive interference with the stern wave which creates larger waves. Since waves are created from the energy of the boat driving through the water, it requires more force for the boat to create the larger bow and stern waves. Therefore, total resistance increases at Froude number 0.40. For ships, Fn=0.40 is considered the "hull speed." Without extra powering, most ships will not be able to travel at faster Froude numbers.

Farfield waves are the summation of waves at a great distance away from the ship. Individual waves created by the hull merge a short distance away from the ship, creating a uniform wave pattern that propagates outward. In the absence of viscosity, these would travel to infinity. By inspecting the farfield waves, the total resistance from all of the wave interference can be calculated.[10]

The Havelock formula (Eqn (18)) calculates the resistance caused by waves through a calculation of the farfield wave-spectrum (S). In Eqn (18), the wave-spectrum is broken into its real (S_r) and imaginary parts (S_i). [11]

$$C_W = \frac{\nu}{\pi} \int_{-\infty}^{\infty} \frac{d\beta\, \kappa}{\kappa - \nu} \left(S_r^2 - S_i^2 \right) \tag{18}$$

The Havelock formula is a function of Froude number, wavenumber (κ), the Fourier variable (β), and the wave-spectrum. Froude number is incorporated into the equation via Eqn (19) which acts as a description for velocity.

$$\nu = \frac{1}{2{F_n}^2} \tag{19}$$

Wavenumber is a description of the wavelength (λ) of the radiated waves (Eqn (20)) and therefore acts as a function of wave celerity (c), or the speed of the waves (Eqn (21)). [6]

$$\kappa = \frac{2\pi}{\lambda} \tag{20}$$

Wave celerity (Eqn (21)) is also a function of the frequency of the waves (ω). The wave frequency, along with wavelength can be used to calculate the energy in each wave. [6]

$$c = \frac{\omega}{\kappa} \tag{21}$$

Moreover, wavenumber is a function of both velocity and the Fourier variable in Eqn (22). [11]

$$\kappa(\beta) = \nu + \sqrt{\nu^2 + \beta^2} \tag{22}$$

The farfield wave-spectrum can be calculated as a function of disturbance velocity, or the velocity of water particles due to the ship's velocity, along the hull. This calculation, the Fourier-Kochin representation of farfield waves, was used by Dr. Noblesse to develop FKS.[11]

FKS calculates the disturbance velocity at the hull using a slender-ship approximation. This approach treats a hull as a set of cylinders without end conditions. The benefit of this approach is that calculations can occur on limited resources. The negative aspect of this method is that the approximation breaks down once the ship's beam is too large for its length. Slender-ship theory's assumptions will fail when the beam is much greater than the total wavelength.[6]

FKS calculates the velocity at the hull of the ship via the slender-ship approximation. Using these velocities, the farfield waves can be calculated using the Fourier-Kochin method. Finally, the Havelock formula calculates the wave resistance from the farfield waves. [11]

FKS Process

In order to calculate the disturbance velocity at the hull, FKS needed to be able to make calculations along a numerical representation of the hull. FKS used a three-dimensional surface model divided into panels to form a structured grid, or mesh (Figure (28)). The model itself had to be scaled to a non-dimensional length of one unit.

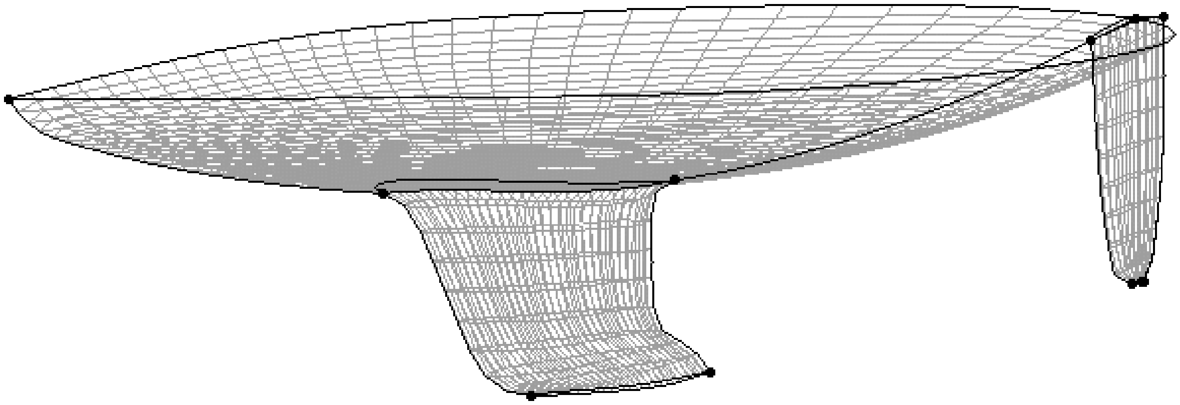


Figure 28 – Nondimensional Mk II Navy 44 STC with trapezoidal mesh

Care had to be included in the formation of the meshed hull. [6] FKS used only the underwater-body of the hull as the input mesh. Therefore, the mesh was constructed of the body below the waterplane, or z=0. Since the initial free-surface at z=0 moves to both above (+z) and below (-z) the meshed waterplane, FKS calculates the hull at z>0 as a linear continuation of the hull from below the surface of the water. If the meshed hull had protrusions at the waterplane which were not characteristic of the continued hull, the calculated free-surface would be inaccurate. An example would be if the boat was at a maximum heel where the deck touched the free-surface. At this point, FKS would perform its calculations as if there was still more hull for the free-surface to attach.

[6] Before the beginning of this Trident Project, the author went to "Gridgen training" by the Gridgen Corporation in Ft Worth, TX. This training taught how to form usable meshes using the Gridgen program which were used extensively in this project.

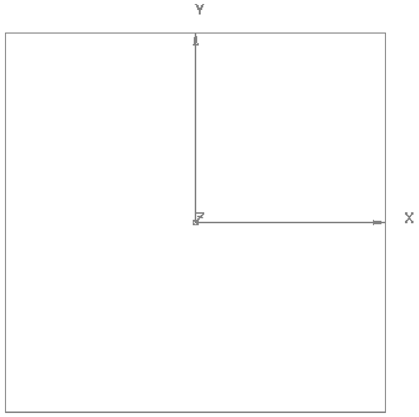


Figure 29 – Sample x-y surface in Gridgen

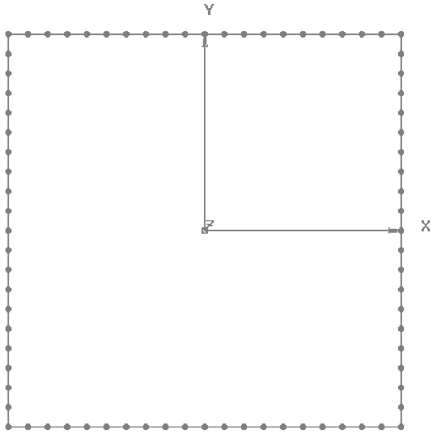


Figure 30 –Dimensionalized connectors

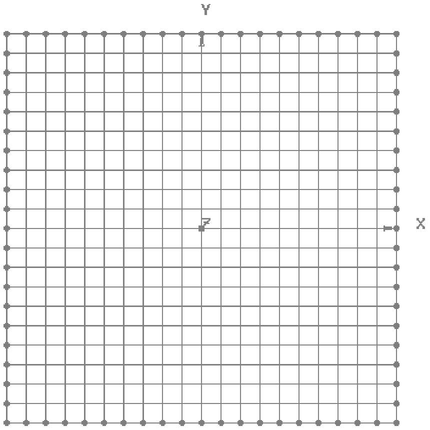


Figure 31 – Gridgen with domain over surface

The CFD meshing program, "Gridgen" was used to create a grid of panels over the underwater surface of the hull.

The first step in using Gridgen was to create items called "connectors." These connectors were placed over the boundaries of the hull. Connectors act as limits to the area being analyzed. They also determine the sizes and amounts of panels used for analysis. Sizes of panels were determined through putting nodes on the connectors in a process called "dimensionalizing."

Finally, an area called a "domain" was constructed by arranging adjacent connectors into a four-sided surface. The order in which the connectors were picked was important since the normal of a domain determines if the domain is an inside or outside piece. A domain with an inverted normal would tell FKS that water was flowing through the inside of the hull, not over the outside.

After exporting the domains which represent the hull from Gridgen, a preprocessor was used to cut the trapezoidal panels in half to create triangular panels. The preprocessor, included in FKS, called "g2FKS" created the final model ready for computations.

Besides the hull itself, the only other input to FKS was the Froude number. By running a set of Froude numbers, an upright resistance curve could be completed.

FKS Tests

To validate FKS, a sample hull known as the Wigley wedge-hull was used. The Wigley wedge-hull is a mathematical form which has wavemaking data published in Principles of Naval Architecture.

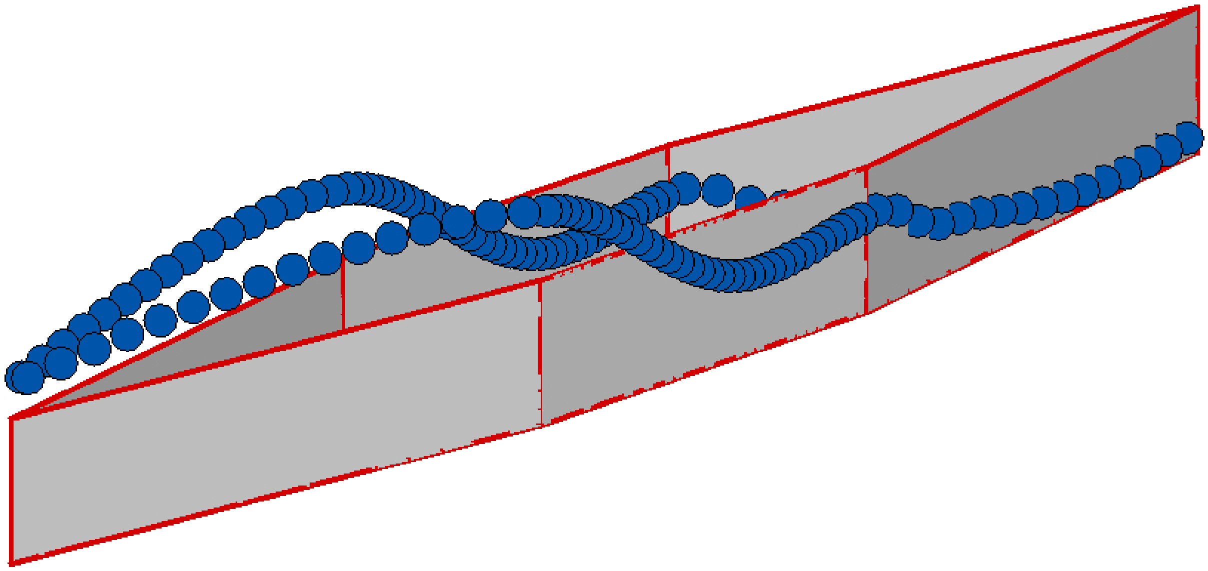


Figure 32 – Wigley hull in FKS At Fn=0.60

The Wigley wedge-hull was created in Gridgen, and a range of speeds were tested. The result from FKS was the variation of the wavemaking coefficient as a function of Froude number. Table (1) indicates that FKS performed well compared to the published calculated coefficient of wavemaking.

		Values of Fn			
Wigley Calculations	Minima C_w	0.23	—	0.35	—
	Maxima C_w	—	0.27	—	0.48
FKS Calculations	Minima C_w	0.25	—	0.33	—
	Maxima C_w	—	0.29	—	0.48

Table 1 – FKS comparison of the Wigley wedge-hull to published data[14]

The published data was of a simple closed-form calculation of the coefficient of wavemaking. Like FKS, the published solution was also inviscid. As a consideration of the published data, the Wigley calculations were from the wave patterns from only four points on the hull. However, at Fn=0.48, the bow wave and the stern wave dominated the resistance curve. That FKS exactly matched the closed-form wavemaking solution at the higher Froude number showed that FKS can properly evaluate a simple hull. It was considered that since FKS was more complex and calculated wave patterns from all parts of the hull, the lower Froude numbers were more accurate in the FKS model than the published solution. At these lower Froude numbers, the resistance curve was not dominated by just one point, but considered the wavemaking of the entire surface. [14]

For the Mk II Navy 44, a script[7] was written in the TCL language to facilitate generating grids in Gridgen. This script received as input any heel, yaw, rudder, or trim angle on the Navy 44. Furthermore, heave and the number of panels on the hull could be set internally in the script. Two subsequent scripts[8] were written in the BASH shell which would automatically run FKS over a set range of speeds.

Two sets of testing were performed on the Mk II Navy 44: upright and sailing conditions. For the upright condition, all attitude angles were set at zero degrees. The range of speeds was set from Fn = 0.05 to 0.70. Each condition took approximately 20 seconds to analyze on a Pentium 4 laptop running Linux. Sixty-six conditions were analyzed.

Sailing conditions were composed of a square test matrix which varied speed, heel angle, yaw angle, and rudder angle. Speed ranged between Fn = 0.1 and 0.5. Heel angle varied between 0 and 27.5 degrees. Yaw angle and Rudder angle was set as if in the tow tank where yaw was set at 0 and 4 degrees. Rudder angle was similarly set at 0, 3, and 6 degrees. Five hundred forty sailing conditions were analyzed.

FKS Upright Analysis

To post-process FKS, several corrections had to be applied. The main problem with the raw FKS data was that the characteristic length used in FKS's Froude number calculations was not representative of the true length of the fluid in question.

A variable called "virtual length" was created to revise the Froude number scaling. Virtual length was created to help visualize the speed of the boat relative to the virtual tow tank in FKS's code.

[7] See Appendix B for the TCL/CL Code
[8] See Appendix B for the BASH Codes

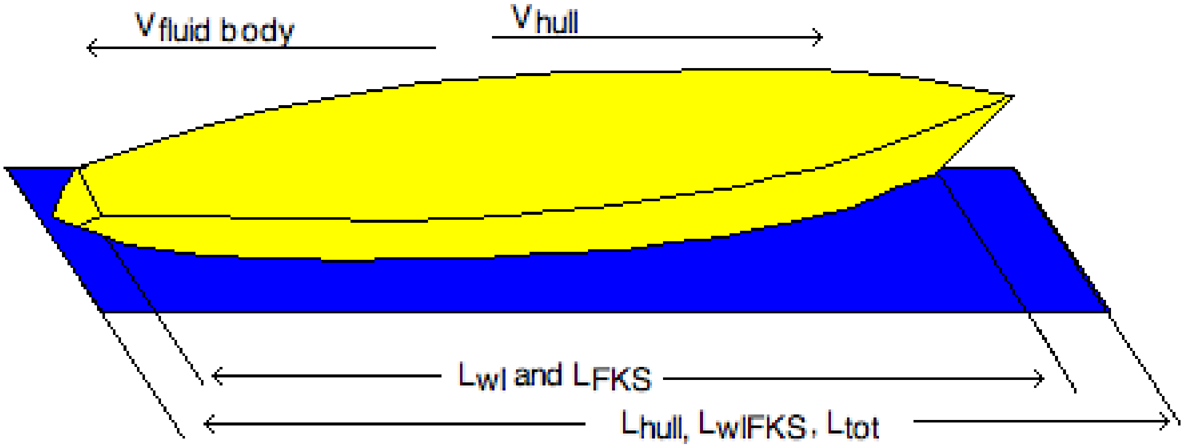


Figure 33 – FKS corrections

The first step in correcting the Froude number was to use the entire model length as the fluid body (Figure (33). This allowed the body of water to travel as fast as the entire hull (L_{hull}). The second correction was to divide the nondimensional length (L_{wlFKS}) by the length of body used in FKS (L_{FKS}). However, the second correction is not reducible to a value of one. One more correction was needed to reduce the terms to one. The nondimensional length was a waterline length, while the term L_{FKS} was a total length. Therefore, the preceding term had to be multiplied by the value of the total length (L_{tot}) over the waterline length (L_{wl}).

$$V_{hull} = V_{water} = F_{n\,FKS}\sqrt{L_{hull} \cdot g \cdot \frac{L_{wlFKS}}{L_{FKS}} \frac{L_{tot}}{L_{wl}}} \qquad (23)$$

The upright resistance curve could now be evaluated and compared to the tank testing curve. The total resistance curve (Figure (34)) for FKS included the wavemaking coefficient and a viscous coefficient based off the ITTC viscous equation (Eqns (8), (9)). In the viscous formulation, the form factor used was k=0.103.

FKS did not use the typical hydrodynamic wavemaking coefficients as in Eqn (4). Instead, FKS uses the Eqn (24) where wavemaking resistance is defined as (R_w) and the velocity of the fluid is (U). [11]

$$C_W = \frac{R_W}{\rho U^2 L^2} \qquad (24)$$

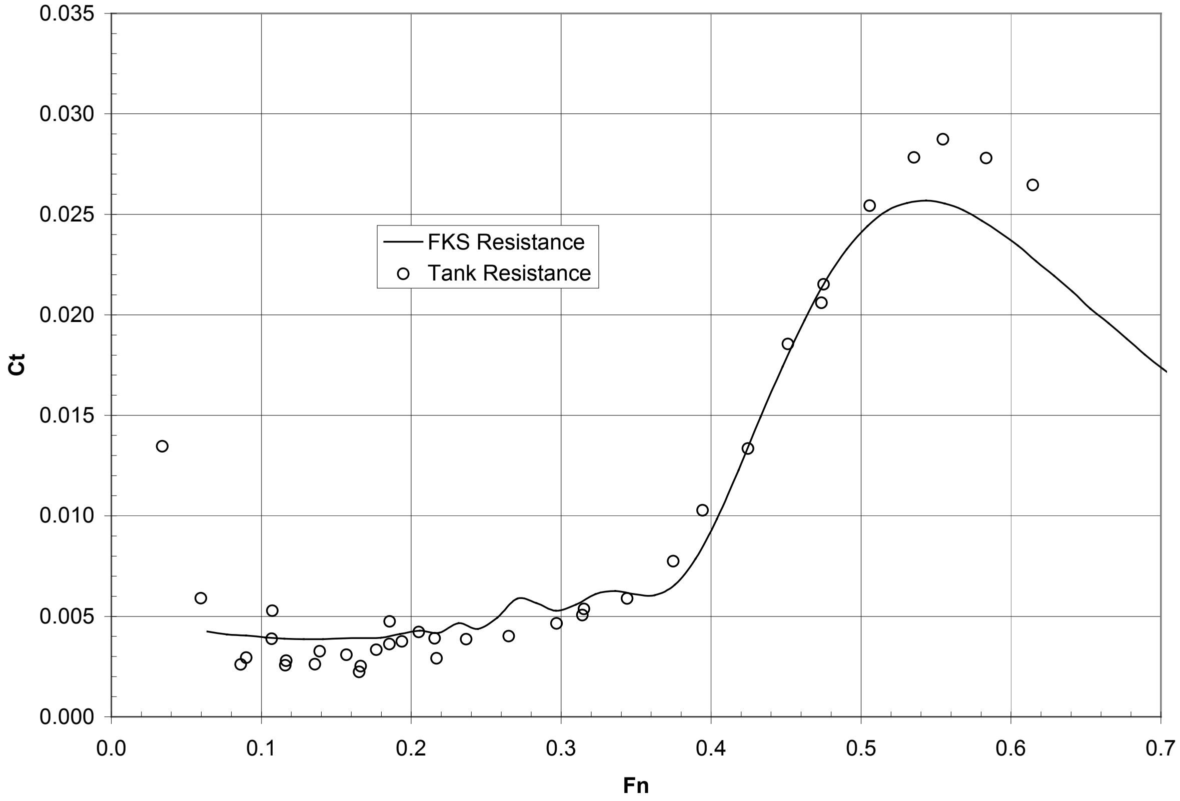


Figure 34 – Upright resistance from FKS and tank testing

FKS compared favorably to a complex hull form like the Mk II Navy 44. However, as Froude number increased after Fn=0.35, spray would develop as the assumptions from continuum mechanics break down in that the free-surface would roll over on itself. This effect became prominent after Fn=0.5. In Figure (34), FKS helped to show where laminar flow existed on the tank model in areas mostly below Fn = 0.2.

Furthermore, the local velocities at the hull calculated using slender-ship theory did not account for the interaction of the waves generated upstream (near the bow). While slender-ship theory is a very quick method of determining the local velocities, its approximations can break down for a beamy hull like the Mk II Navy 44. At high speeds, the hull's wider entry angle would cause local velocity disturbances which are not accounted for in slender ship theory. The presence of this effect contributed to the error seen at the high Froude numbers (Fn > 0.5).

Finally, the humps and hollows of the coefficient of total resistance curve also existed in the coefficient of wavemaking curve. Because FKS performed a pure inviscid wavemaking calculation, the output was thought of as a numerical approximation to a closed form inviscid wavemaking solution. In this solution, the humps and hollows would be magnified since viscosity would normally decrease wavemaking effects.

Further explanations of FKS results are located in the conclusions.

SPLASH

SPLASH is a free-surface CFD code which was developed especially for sailboats. It has been used extensively in the America's Cup and is considered one of the leading CFD codes for sail craft. SPLASH is a panel-based code, meaning that it calculates potential flow, and much revision has gone into the code to evaluate the effectiveness of lifting surfaces.[12]

SPLASH Theory

SPLASH stands for Small Perturbation Linearized Analysis of free-Surface Hydrodynamics. SPLASH is inviscid, meaning that shear stresses from the fluid are not calculated. Secondly, SPLASH calculates the flow of incompressible fluids. Incompressibility combined with inviscidness creates the conditions for an ideal fluid meaning that the potential flow of the fluid can more easily be calculated.[12]

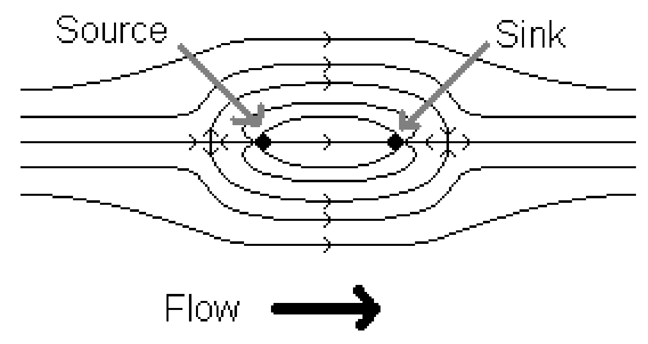


Figure 35 – Potential flow[10]

Potential flow can be calculated from singularities called sources and sinks. To visualize this concept, sources can be thought of as points which radiate fluid symmetrically, while sinks suck fluid (Figure (35)). In reality, sources and sinks simulate the flow around a body by acting as either receptors or transmitters of velocity potential (Φ). The underlying property of potential flow is found in Laplace's equation for valid fluid flow where the velocity potential (Φ) must reach equilibrium: [12]

$$\nabla^2 \Phi = 0 \tag{25}$$

From the velocity potential (Φ), the nondimensional velocity vector (V) can be calculated: [12]

$$V = \nabla \Phi = \Phi_x i + \Phi_y j + \Phi_z k \tag{26}$$

SPLASH Testing

SPLASH was compiled on an IRIX SGI machine with eight processors. Because of the many differing versions of POSIX based systems and the complexity of the Mk II Navy 44, the setup of SPLASH exclusively took almost two months.

As in FKS, the first step to running the code was to generate a mesh around the body. The current version of SPLASH was ideally suited to handle International America's Cup Class (IACC) Yachts. SPLASH normally automatically creates the mesh required to set up a hull for testing. However, the Mk II Navy 44 differed quite a bit from an IACC boat. Changes first had to be made to the Fortran setup files in SPLASH to eliminate the calculations for a keel-bulb and winglets.

Elements known as "iges surfaces" defined the hull shape for the Mk II Navy 44 three-dimensionally. SPLASH normally had the ability to determine how to place meshes onto the iges surfaces of the hull. However, the iges surface of the Mk II Navy 44 hull was of an irregular format, and automatic meshing failed. Instead, Gridgen was again used to generate an initial mesh for SPLASH.

SPLASH was made more accurate by incorporating sail forces into the setup files. SPLASH is a valuable tool for sailing vessels because in its internal calculations, SPLASH will change the attitude of the boat based on an equilibrium found with hydrodynamic and aerodynamic forces. Sail forces could be determined either from a VPP or from tank testing data. In the tank, aerodynamic forces are equal and opposite of the hydrodynamic forces. Therefore, the tank data of the Mk II Navy 44 was modified to become aerodynamic data and was inserted into SPLASH's sail forces file.

Finally, the actual conditions to be analyzed in SPLASH had to be coded into FORTRAN. Conditions were chosen based on a method that SPLASH needed to post-process the data. For the upright condition, 40 points were picked. For the sailing condition, heel angle was set at 0, 10, 15, 20, and 25 degrees while speed ranged from 0 to 10.5 knots. Yaw and rudder angle varied as in FKS and tank testing – using six tests for each heel and speed combination.

SPLASH Upright Analysis

As in both tank testing and FKS, the output from SPLASH consisted of upright and sailing data. SPLASH did not require any scaling of the output since forces were computed at the full-scale. Furthermore, it was not necessary to calculate the viscous drag of the ship in the post-processing. This was performed as a part of SPLASH's computations.

Figure (36) shows how SPLASH compared the upright resistance of the Mk II Navy 44 to tank testing and FKS predictions. At high velocities, SPLASH predicted slightly greater resistance for the ship compared to both tank testing and FKS. One explanation for this is in the viscous drag calculation, called "viscous-stripping." In tank testing and FKS, viscous drag was computed using a standard wetted surface area. However, since SPLASH calculated the changed wetted surface area for each sailing condition, the updated area was used in the calculation. The result of this is that SPLASH more accurately predicted viscous drag than the other two methods.

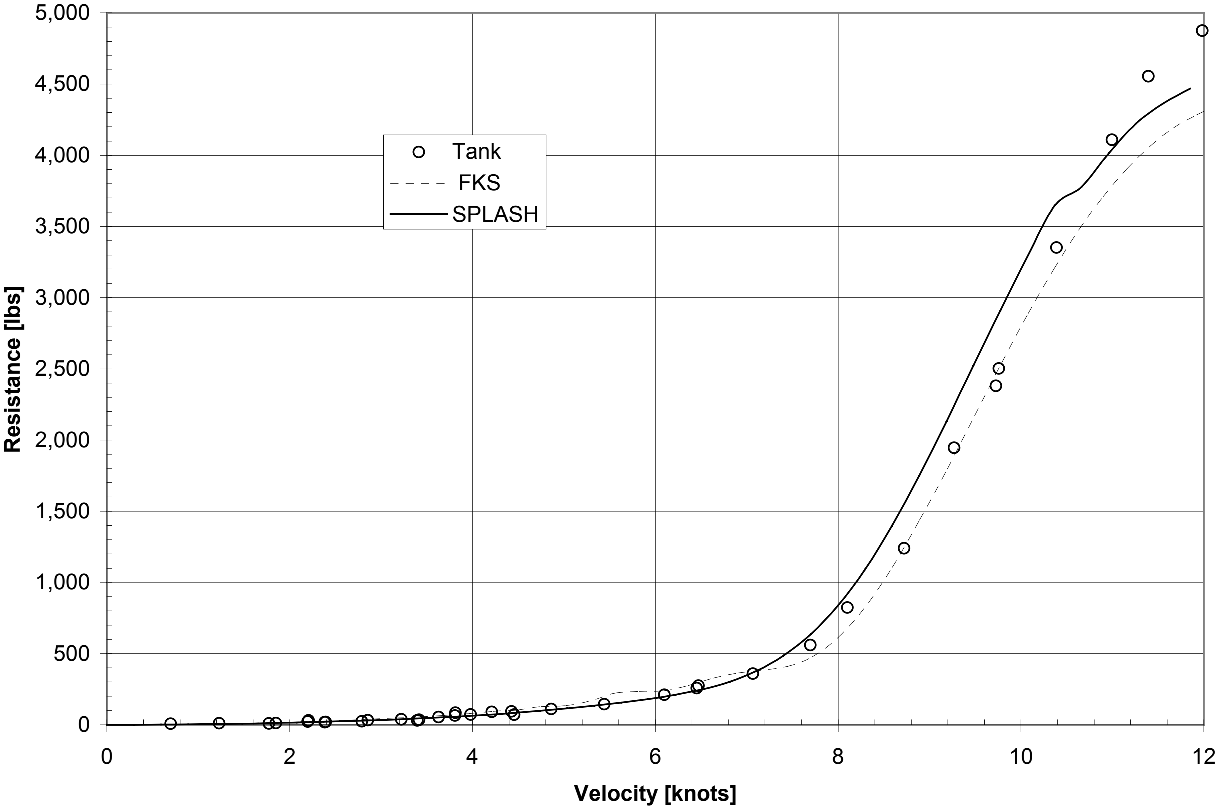


Figure 36 – Resistance of all upright data

Interestingly, at a speed of 10.5 knots, SPLASH predicted a significant drop in resistance. At this speed, SPLASH also predicted that the pitch angle of the boat would jump as well. As boat speed would accelerate through 10.5 knots, the vessel would actually begin to ride over the bow wave (a similar principle to breaking the sound barrier in aerodynamics). Moving over the bow wave would create trim by the stern of about 5 degrees. The effect of this is that the bow would rise out of the water, decreasing the waterline. At this velocity, the bow wave would begin to disintegrate, leaving only a partial bow wave and a stern wave. Without the massive bow wave, the coefficient of wavemaking would drop. Furthermore, with a decreased wetted surface area, viscous resistance would decrease as well. This speed, known as semi-planing is characterized by a decrease in the coefficient of total resistance and facilitates high-powered boats to reach speeds much larger than their own hull speeds. That the tank test data did not clearly indicate such a condition of semi-planing, was not cause for questioning SPLASH's predictions. On the contrary, there were not enough upright high-speed tests performed in tank testing to sufficiently populate this region.

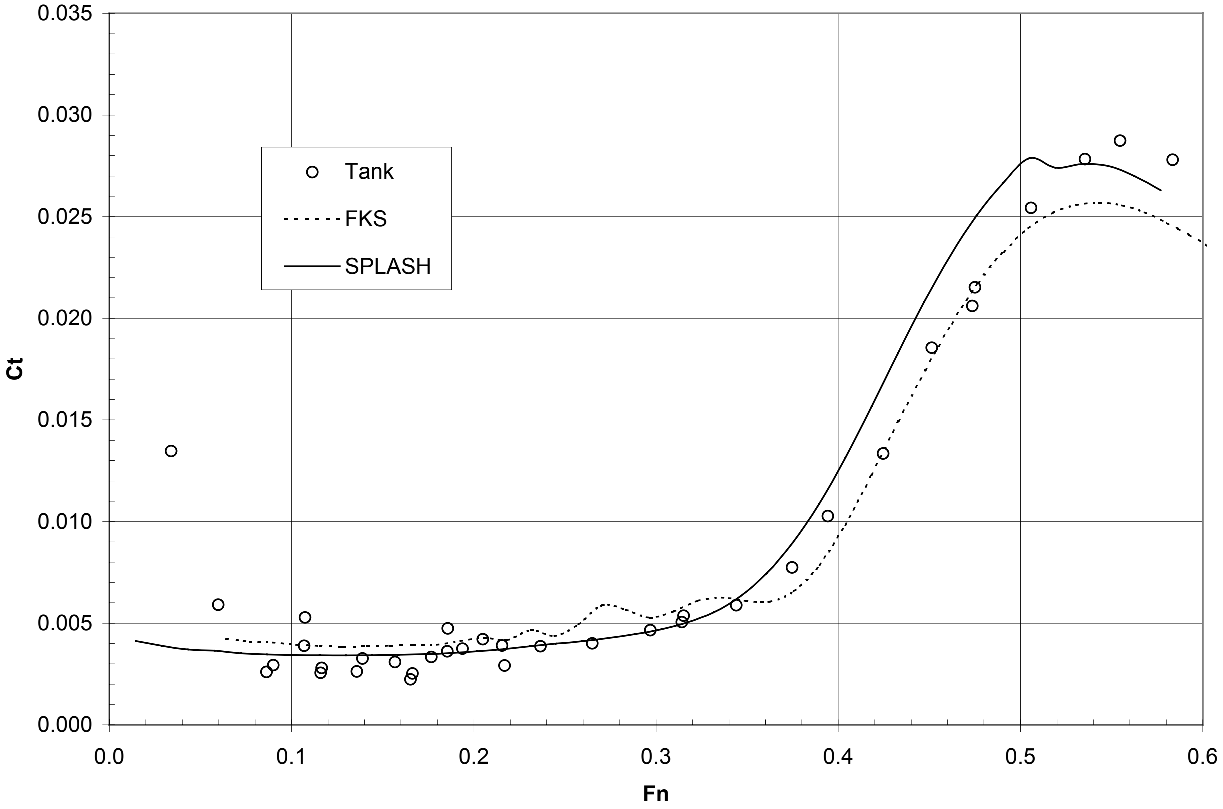


Figure 37 – Total resistance coefficient of all upright data

Noticeably, SPLASH did not have the minor humps and hollows in the coefficient curves that were present from FKS (Figure (37)). Since SPLASH calculated only the potential flow, some of the wave interference effects were minimized when compared to FKS. However, many of the humps and hollows found in FKS would not be as great if viscous interference were to occur. Therefore, the actual coefficient of wavemaking curve was estimated to be between the curves of SPLASH and FKS. The use of a viscous CFD code could help to identify where discrepancies exist.

Furthermore, it was noticed that at $0.1 < Fn < 0.2$ there was a difference in the coefficient of total resistance between FKS and SPLASH. At this speed, viscous resistance dominated the resistance on the boat and this again indicated that the viscous resistance calculation differed between both methods.

SAILING DATA ANALYSIS

Tank testing provided 120 data points for sailing conditions. Each point varied by boat speed, heel angle, rudder angle, and yaw angle. FKS also gave force outputs for a much larger matrix of over 500 data points. SPLASH generated a matrix of 150 data points. The difficult part of analyzing the sailing conditions was that each method generated results for slightly different sailing conditions.

Many attempts were made at making use of the sailing data before a sure method was created. The main problem with the sailing data was that the sailing conditions tested in the tank, FKS, and SPLASH, did not exactly match the predicated conditions from the PCSail VPP. In the tank, this was because the velocity input for the tank could only be set approximately to the desired velocity. Furthermore, the internal data analysis in SPLASH confined tests from being singular test conditions. In SPLASH, the PCSail VPP predicted conditions could not be individually tested efficiently.

The first attempt of calculations with the tank data was to perform a regression of all of the data points to find functions for drag, lift, and the aft-lift terms. These functions would vary by velocity, heel angle, yaw angle, and rudder angle. Several multivariate regressions were made, but ultimately, a polynomial expression for the data with acceptable accuracy could not be found.

A second approach used MATLAB to interpolate between the acquired test points.[9] The interpolation technique took the sparsely populated data matrix and transformed it into a square matrix using splines to calculate intermediate points. Since the transformed matrix was both square and dense, linear interpolation through this matrix could be used to find the required values for any velocity and heel angle. Each yaw angle and rudder angle condition created six matrices each for the drag, fwd-lift, and aft-lift variables.

Making square matrices of the tank data provided not only a method to perform interpolation, but also visualization and analysis of the acquired sailing data. Figure (38) shows the collected data laid onto the matrix used in calculations.

[9] See Appendix C for the MATLAB script

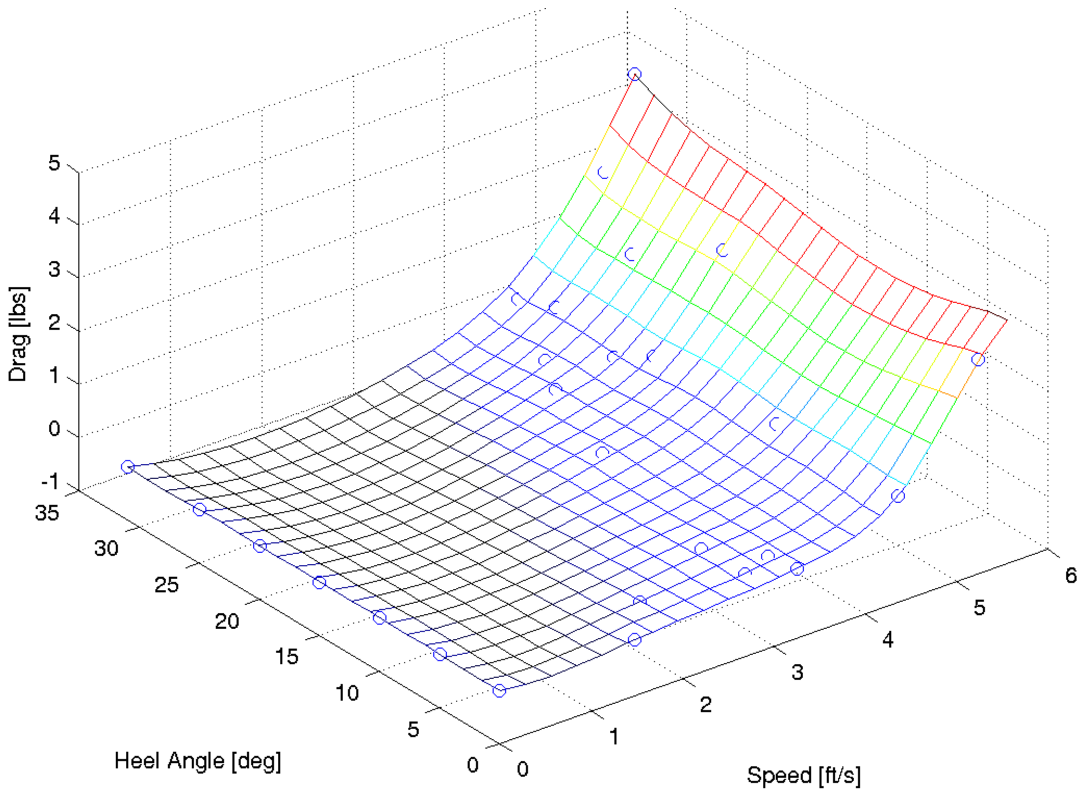


Figure 38 – Tank testing drag with 0 degrees rudder and 0 degrees yaw

Figure (38) indicates that for a given heel angle, drag did not change dramatically. Upon further inspection of the data, two things became apparent. First, values chosen for calculations were required to be inside the range of tested values. Furthermore, between the testing matrix and velocity = 0, the splines created a very slight negative drag in order to smooth their curves. At speeds and heels outside the range of the testing matrix, there was no data and extrapolation would be inappropriate. This applied for all data collected (drag, fwd-lift, and aft-lift). Notice that the square matrix in Figure (39) modified for calculations reported null answers for any data which was not in the range of interpolation.

Secondly, drag did not change dramatically between heel angles, but Figure (39) shows that there was a significant amount of change in drag across a range of heel angles with constant model speed. This shows where tank testing and CFD have great resolution. The change in drag across a range of heels is a function of only the shape of any particular hull. A VPP would not be able to detect these fine changes in drag.

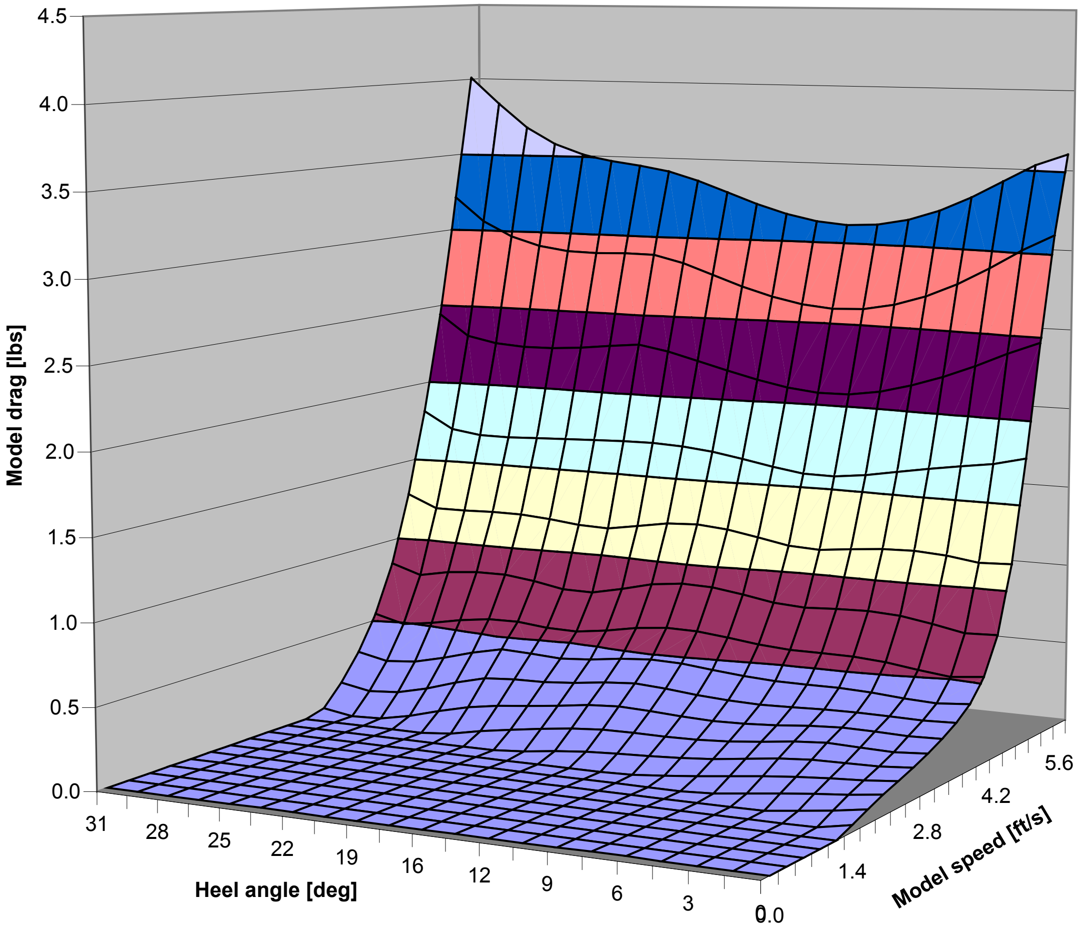


Figure 39 – Closer inspection of tank testing drag with 0 degrees rudder and 0 degrees yaw

For tank testing, CFD, and FKS, six drag matrices were made for all rudder angles and yaw angles. One of the disadvantages of the method of making square matrices for the sailing conditions had to deal again with resolution. Individual points, such as the point at model speed 4.2 fps and heel of 10 degrees (Figure (38) had a great effect on the shape of the entire matrix. Notice that above 4.2 fps, around 10 degrees of heel, a dip was present (Figure (39). The precision of the test at this point could be drawn into question, especially since it had far reaching affects on the rest of the matrix.

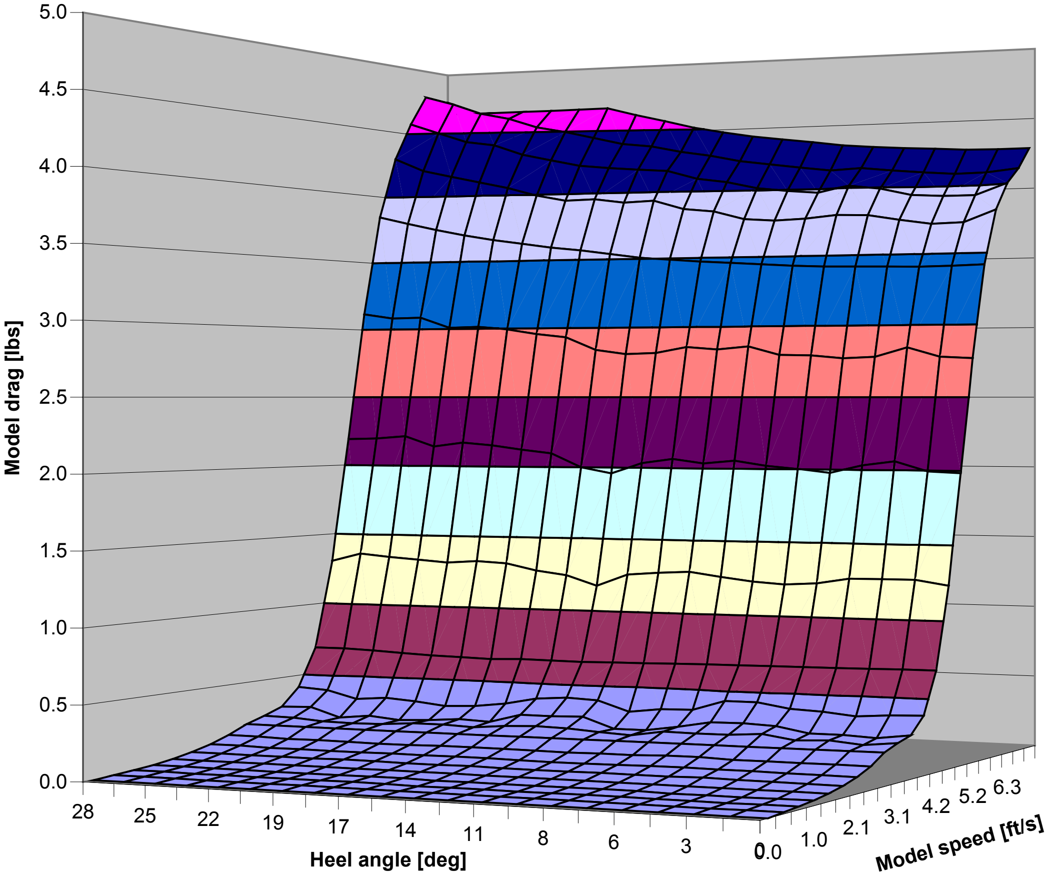


Figure 40 – FKS drag with 0 degrees rudder and 0 degrees yaw

FKS did not show a dip in drag at 10 degrees of heel at all (Figure (40)). However, FKS showed interesting humps right underneath 0.5 pounds of drag. These, unfortunately were in error, and the sailing data for FKS will later be shown to be unrealistic.

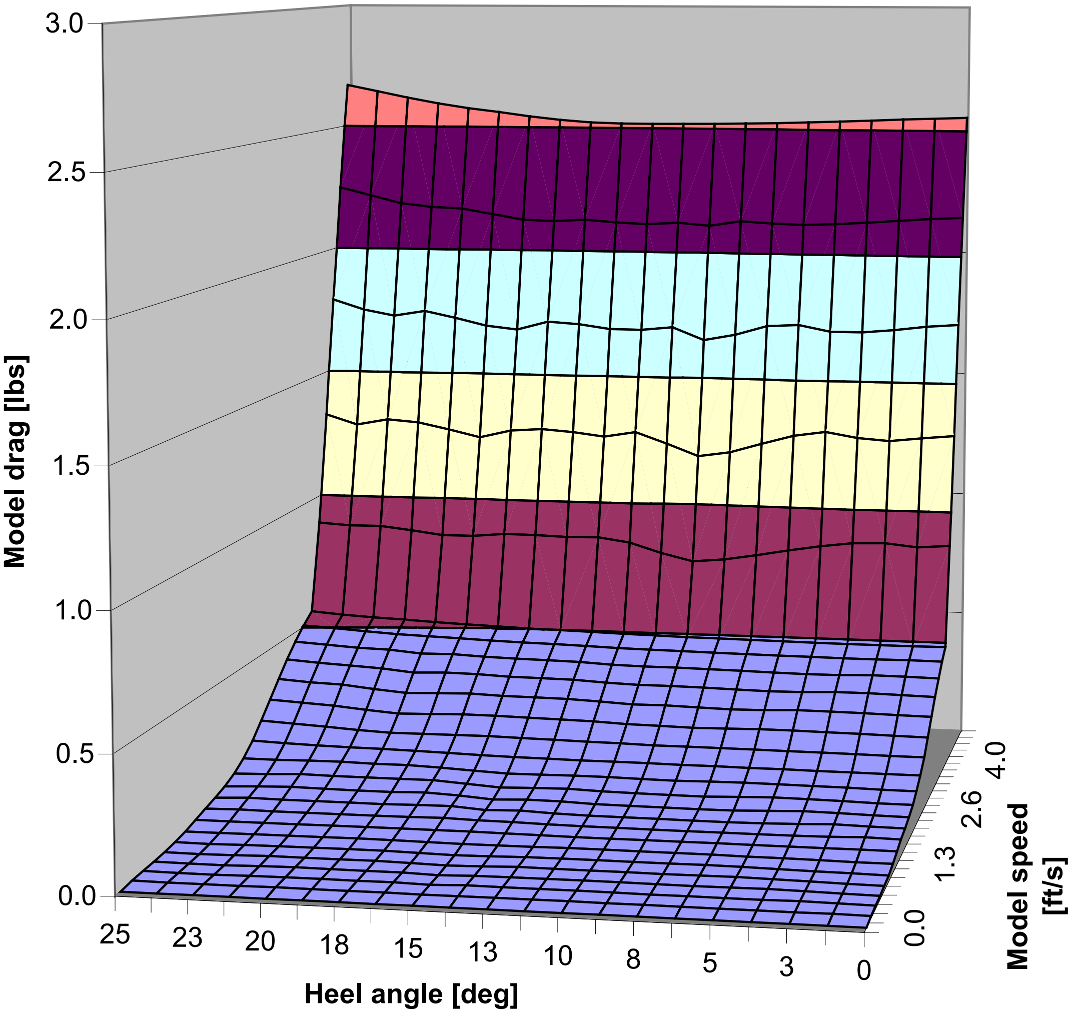


Figure 41 – SPLASH drag with 0 degrees rudder and 0 degrees yaw

Since SPLASH results were given in the full-size condition, the results presented in Figure (41) were calculated to model scale to facilitate the comparison between the three experimental methods. First, total model drag in Figure (41) did not have as high a limit in SPLASH as in tank testing or FKS. The speed of the tests in SPLASH were limited to 5 fps, unlike tank testing which proceeded to 5.6 fps for the sailing conditions.

Secondly, SPLASH did not show as much variation in drag as a function of heel angle as when compared to tank testing. This further draws into question the precision[10] of individual points in tank testing. A quality of CFD which was exposed is that its precision, unlike tank

[10] In engineering terminology, "precision" is defined as repeatability of tests while "accuracy" is defined as absolute error of the tests from a datum.

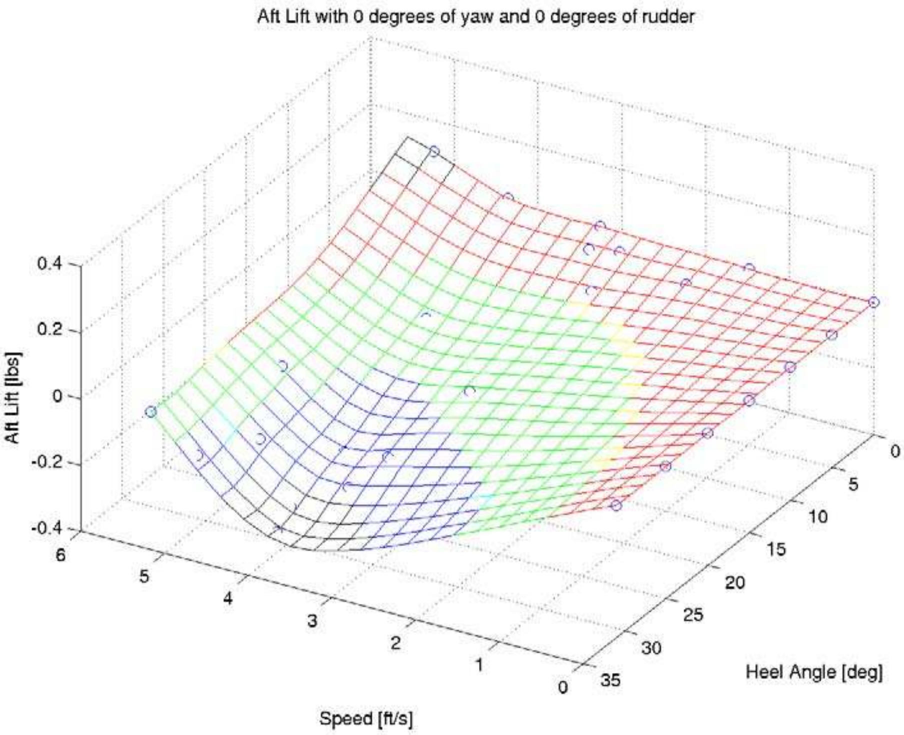
testing, can always be 100%. This made matrices such as these very able to obtain reasonable results. However, in CFD, if the meshing of a model at certain conditions was incorrect, the accuracy of the entire grid would be very low.

The questionable accuracy was the case for FKS. When forming the meshes for the sailing conditions, FKS suffered from two items. First, as the model was heeled in Gridgen, it was not heaved at corresponding amounts to control the displacement. This created a mesh which was not an accurate representation of the hull. Secondly, FKS did not iteratively remesh the hull during its calculations. This tended to leave the hull in a position which was also not accurate. Finally, FKS was not programmed to make lift circulation calculations. The meaning of this is that FKS could not accurately model lifting forces or induced drag due to lifting surfaces. Therefore, the use of FKS for resistance calculations was only appropriate for the upright condition with no heel angle.

SPLASH also had accuracy problems. First, although the hull was remeshed during the calculations, the data which provided the aerodynamic force balance was calculated from tank testing data. The aerodynamic force balance provided only a small input into the trim of the vessel, but the realization is that the accuracy of the remeshing was only as high as the precision and accuracy of the tank testing.

Interesting plots were seen as a series through different rudder and yaw angles. Figure (42) shows sailing data recorded from the aft-lift force block from the tank. These forces, when multiplied by the distance between the aft-pin and the heave post gave the yawing moment. From this data, the effect of yaw angle, and rudder angle, and heel angle on the yawing moment were investigated.

Similarly, Figure (43) is the yaw moment measured in SPLASH. The moment was around the heave post at its placement on the model. Although Figure (42) is force data and Figure (43) is moment data, both plots are proportional.



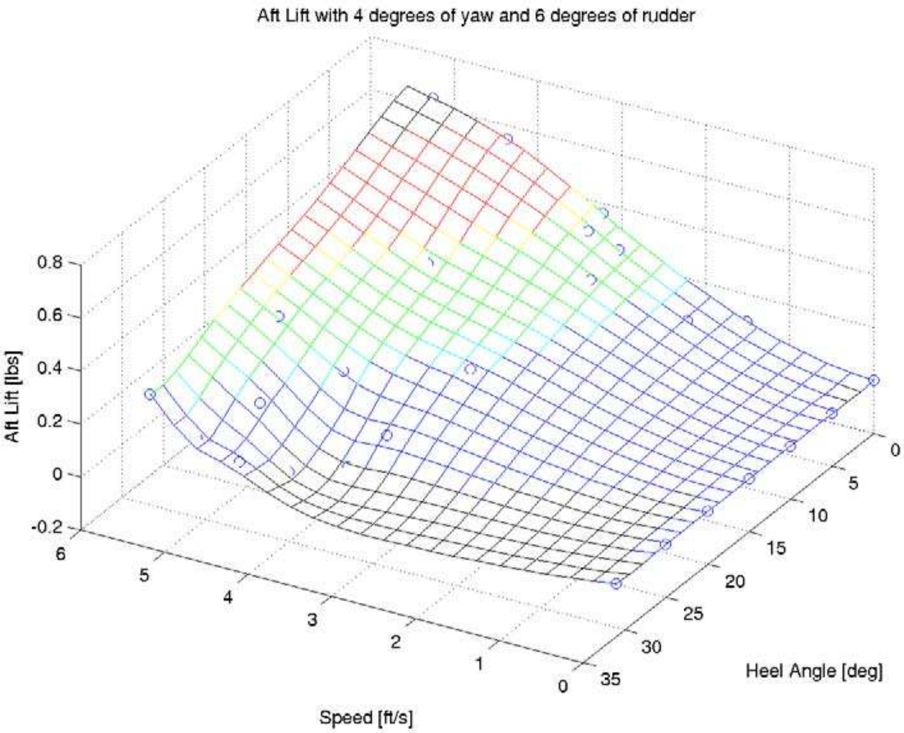


Figure 42 – Data from the aft-lift force block in tank testing

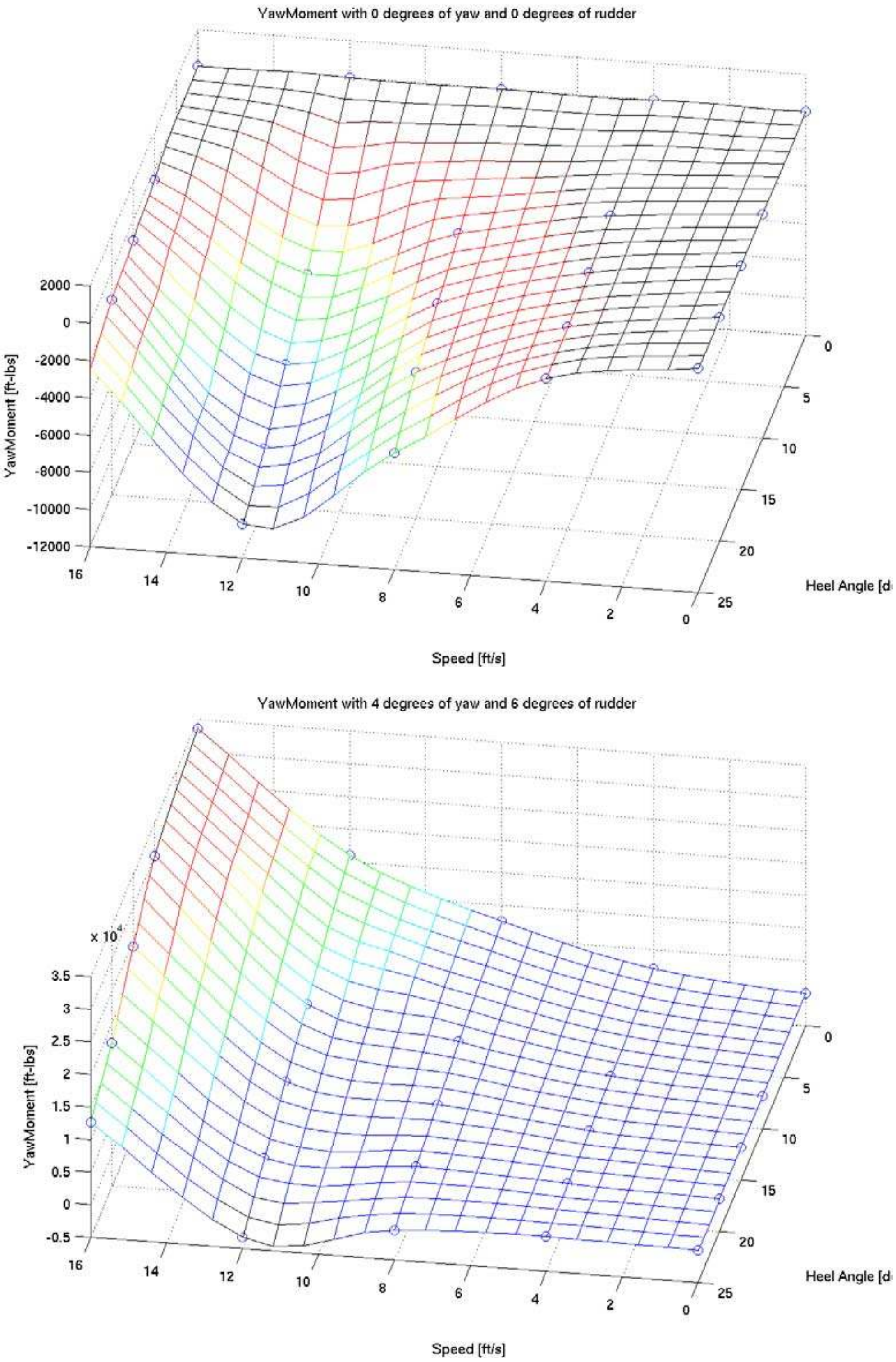


Figure 43 – Full-scale yaw moment around the heave post in SPLASH

Both tank test data and SPLASH data showed very similar results with respect to yawing moment. The similarity in both acted to validate each other by arriving at the same answer from two very different methods. The yawing moment was a difficult measurement to make in the tank because of scaling errors. Because the size of the rudder was so small, it had a high chance of laminar flow thereby not creating the necessary lift across the rudder or yaw moment for the entire hull. The yawing moment was also a difficult computation to make in CFD but for different reasons. On a sailboat, there is a strong interaction of flow between the keel, the hull, and the rudder. If some part of the mesh around an appendage, or between an appendage and the hull, was flawed, then the entire lift circulation calculated by the appendage would have tremendous error.

However, there were differences. For instance, in the tank, especially at the condition of 0 degrees yaw, heel, and rudder, there was a measurable yaw moment at high velocities. This moment was attributed to large amounts of heave and trim in the model at high speeds. The effect of this movement was that the distance between the aft-pin and the force block tried to lengthen and therefore put tension on the aft-lift force block. The presence of this effect is visible in Figure (44).

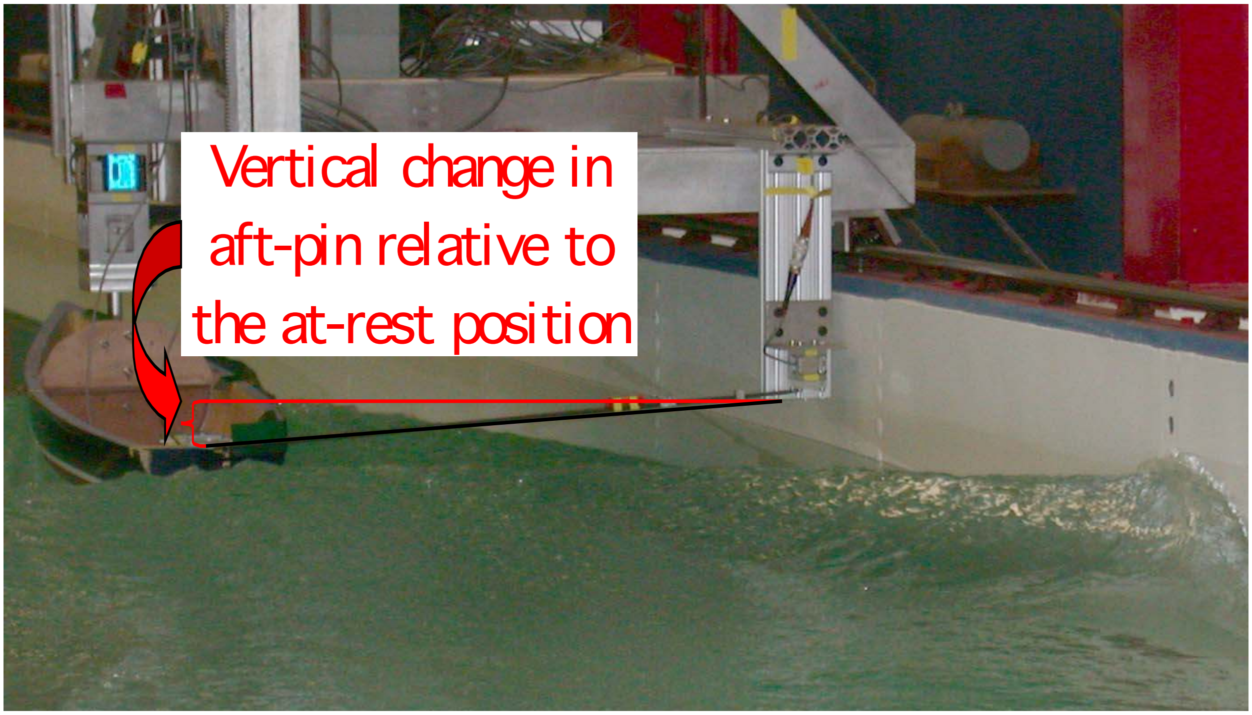


Figure 44 – High speed 0 degrees yaw, heel, rudder tank test

The conclusion is that although both methods had good abilities at measuring very specific forces on the hull, SPLASH was far more able to accurately measure these minute forces without any induced testing error.

From a sailing-analysis perspective, both methods showed interesting results for yaw-moment in that heel moment created a tremendous amount of inverse yaw. The meaning of this is that the more the Mk II Navy 44 heeled over, the more it wanted to put its bow into the wind. Although this was already known to sailors who have experienced this effect of "rounding up," the quantification of this principle showed that it occurred most at high boat speeds. Notably, both SPLASH and tank testing predicted the same speed for the maximum rounding up effect: 7 to 7.5 knots.

One of the tremendously useful abilities of SPLASH was in its visualization of each test. Figure (45) shows the result of the flow calculations visually which helped analysis in two ways. First, visualization helped ensure that there were no meshing errors associated with the calculations. Secondly, Figure (45) and other angles of the image show the contour plot of the pressure distribution across the entire hull and free-surface. Labeled Cp for coefficient of pressure, this pressure diagram could be integrated to calculate the center of lateral resistance for each appendage or for the entire hull. Not only did Cp show the result of the calculations, but Cp could aid with the development of the hull form or even the design of the sail arrangements. Stream vectors along with Cp also showed areas where a design may not be fair. For instance, at the rudder tip, large vectors existed indicating non-continuous flow calculations. Although some of these vectors may be the result of induced drag vortices, it also indicated that the tip of the rudder was not a smooth surface.[11]

[11] After analyzing the grid near the rudder, Joe Laiosa and Bruce Rosen, the creators of SPLASH, reported that the tip of the rudder created from the iges surface itself was not fair

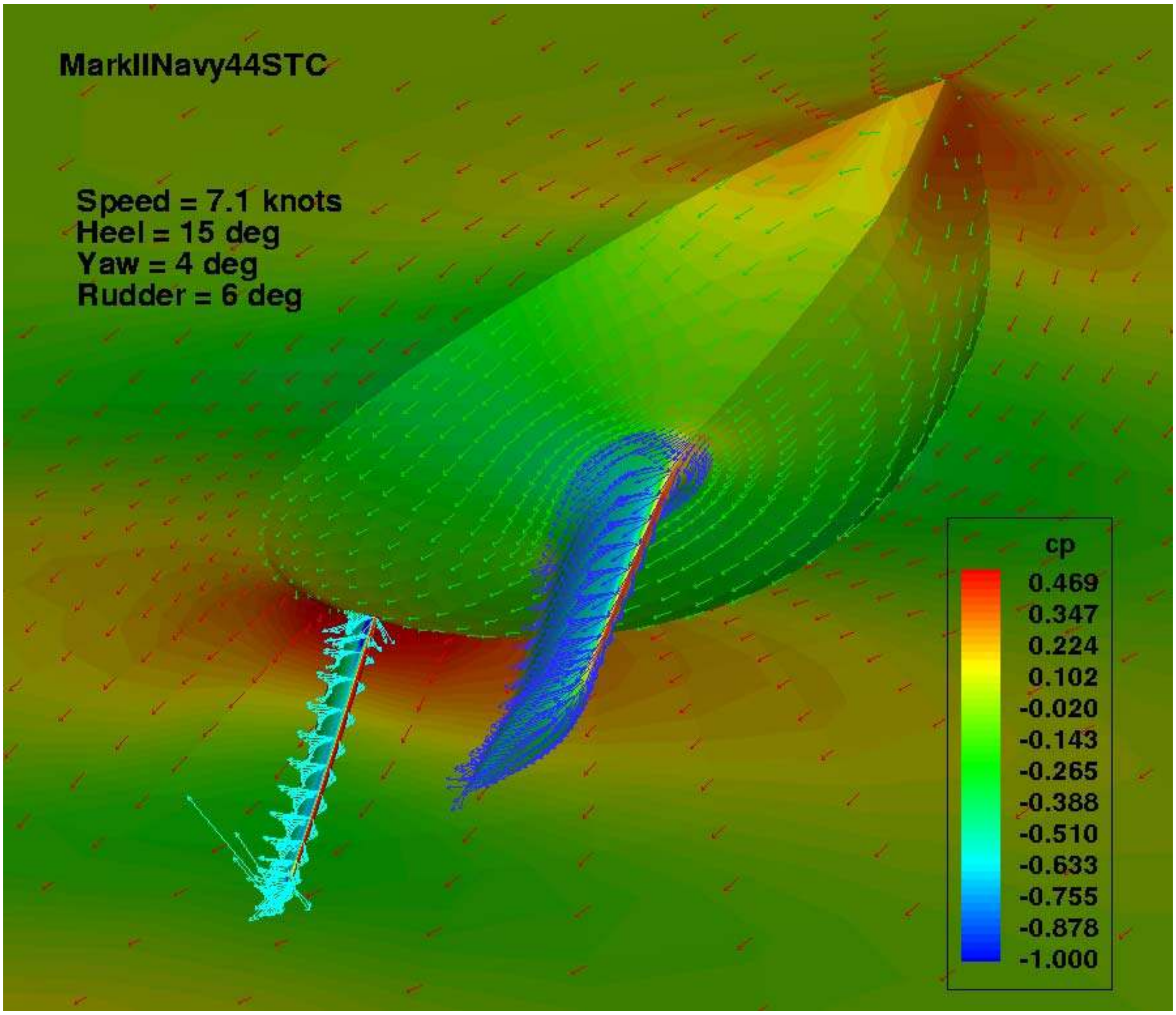


Figure 45 – Potential flow visualization in SPLASH

MAPLE was used to process data between the yaw and rudder angles.[12] Since all variables were considered to behave linearly in yaw, an equation for yaw angle was formed which could interpolate between each matrix. Similarly, there were three rudder angles tested for each point. A quadratic regression was used to solve between different yaw angles. Ultimately, a function was created which treated yaw and rudder as two axes of a three-dimensional surface. The surface (ζ) was defined in terms of rudder angle (γ) and yaw angle (λ) by the function:

$$\begin{aligned}\zeta(\lambda,\gamma) = {} & \tfrac{1}{72}\gamma^2\cdot\lambda\cdot\zeta(4,0) - \tfrac{1}{72}\gamma^2\cdot\lambda\cdot\zeta(0,0) + \tfrac{1}{36}\gamma^2\cdot\lambda\cdot\zeta(0,3) - \tfrac{1}{36}\gamma^2\cdot\lambda\cdot\zeta(4,3) \\ & - \tfrac{1}{72}\gamma^2\cdot\lambda\cdot\zeta(0,6) + \tfrac{1}{72}\gamma^2\cdot\lambda\cdot\zeta(4,6) + \tfrac{1}{18}\gamma^2\cdot\zeta(0,0) - \tfrac{1}{9}\gamma^2\cdot\zeta(0,3) \\ & + \tfrac{1}{18}\gamma^2\cdot\zeta(0,6) - \tfrac{1}{8}\gamma\cdot\lambda\cdot\zeta(4,0) + \tfrac{1}{8}\gamma\cdot\lambda\cdot\zeta(0,0) - \tfrac{1}{6}\gamma\cdot\lambda\cdot\zeta(0,3) \\ & + \tfrac{1}{6}\gamma\cdot\zeta(4,3) + \tfrac{1}{24}\gamma\cdot\lambda\cdot\zeta(0,6) - \tfrac{1}{24}\gamma\cdot\lambda\cdot\zeta(4,6) - \tfrac{1}{2}\gamma\cdot\zeta(0,0) \\ & + \tfrac{2}{3}\gamma\cdot\zeta(0,3) - \tfrac{1}{6}\gamma\cdot\zeta(0,6) - \tfrac{1}{4}\lambda\cdot\zeta(0,0) + \tfrac{1}{4}\lambda\cdot\zeta(4,0) + \zeta(0,0)\end{aligned} \tag{27}$$

An example of this interpolation between matrices is shown in Figure (46) where surface (ζ) is model drag at 2.5 feet per second and 10 degrees heel with the results from tank data.

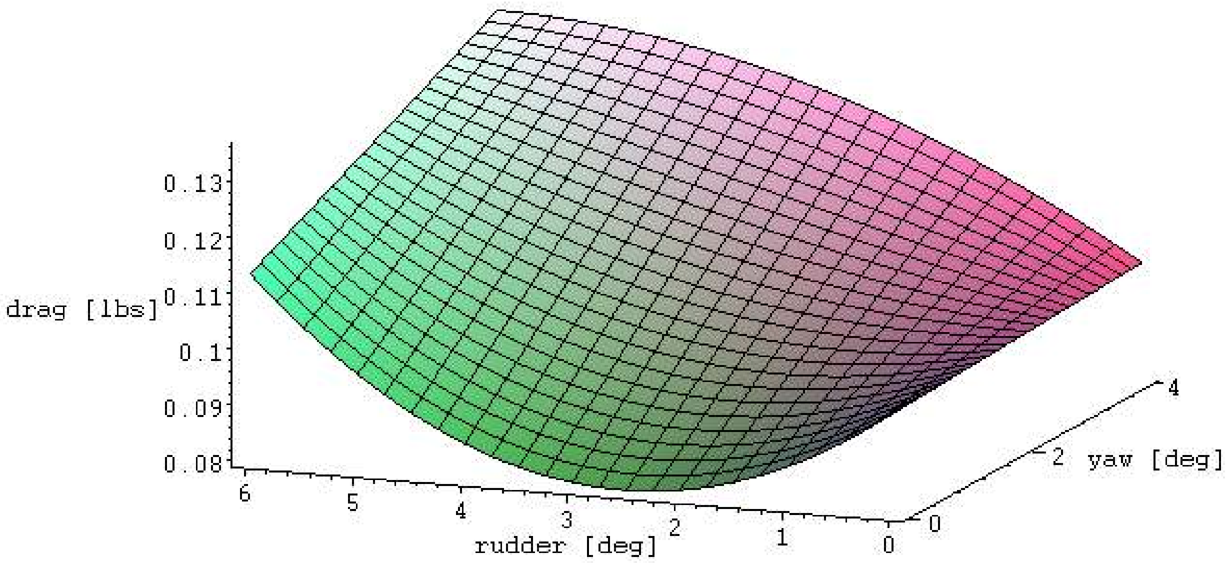


Figure 46 – Model drag from tank data defined by rudder angle and yaw angle

Now that an interpolation method was found, the tank data was compared to PCSail. For each sailing condition, an interpolated hydrodynamic tank condition was found. Each condition

[12] See Appendix C for the MAPLE script

had a specified heel angle and boat speed which could be matched as a tank condition. Each PCSail condition also included the aerodynamic forces acting on the sails. Therefore, the aerodynamic drive and side force were matched with the hydrodynamic data found in the tank.

Unfortunately, there were two problems with this comparison method. First, the tank data either over or under-predicted the PCSail forces by up to 20% for each condition. This percentage of error was unacceptable in making comparisons between tank and CFD prediction methods. The source of this error was due to the simplified algorithms in the hydrodynamic code of PCSail. The predicted velocity and heel parametrically determined from PCSail for a sailing condition was not hydrodynamically the same as the similar condition measured in either CFD or the tank.

Secondly, yaw and rudder angle could not be solved using an interpolative comparison to the PCSail data. Since the forces found in the tank included drag, fwd-lift, and aft-lift, there were too many variables to solve for both yaw and rudder angles.

The solution was to create a new, customized Velocity Prediction Program which was focused only on the Mk II Navy 44 STC.

NAVY 44 VELOCITY PREDICTION PROGRAM

Having the hydrodynamic data acquired from multiple sources, there were two components from a velocity prediction program which were missing: aerodynamic data and a solver. Because of its ability to show internal results and generate plots useful in evaluating the code, Excel was chosen as the platform to handle the code. Acquired data from tank testing, FKS, and SPLASH were stored in a worksheet, and most of the code was written in Visual Basic as well as cell functions in the spreadsheet.[13]

Aerodynamic Theory

The aerodynamic data was calculated using the Hazen method, as in PCSail. As wind passes over a sail from a certain direction, lift and drag are created as with a wing. The lift is perpendicular to the angle of attack of the wind, while the drag is parallel to the angle of attack of the wind. Lift and drag are calculated in terms of coefficients (Eqn (28)). [5]

$$F_{aero} = \tfrac{1}{2} C \rho_{air} A V^2 \tag{28}$$

There is no difference between this calculation of aerodynamic force and of hydrodynamic force. Designation changes are that the density of the fluid (ρ_{air}) is multiplied by the nominal area of the sail (A) and the velocity of the surrounding air (V).

The coefficient of lift (C_L) generated by a sail is assumed a function only of apparent wind angle. There are different coefficients of lift for each sail: mainsail, jib, and spinnaker. [5]

The coefficient of drag (C_D) is the summation of three variables: parasitic drag, induced drag, and windage. Parasitic drag (C_{DP}) is the friction associated with wind passing over a sail. Like lift, it is assumed also to be only a function of apparent wind angle. [5]

Induced drag (C_{DI}) is the result of vortices created as the airflow is sucked around the windward portion of the sail to the leeward side (Eqn (29)). [9] Induced drag is a result of the pressure differential on the sail and the end condition at the top of the sail. Based on simple wing theory, it is a function of the square of the lift coefficient for a sail. Furthermore, the aspect ratio (AR) of a sail plays an important role in sail efficiency. A high aspect ratio sail will lessen the induced drag by effectively delaying the amount of vortices which can be created. [7]

[13] See Appendix D for the Mk II Navy 44 VPP

$$C_{DI} = C_L^2 \left(\frac{1}{\pi AR} + 0.005 \right) \tag{29}$$

The aspect ratio used in calculating induced drag is not of any particular sail, but of the entire sailing vessel. For upwind sailing, the height of the deck above the water (FBD) is assumed to play an effect in the induced drag of the vessel. Therefore, there are two equations for the calculation of aspect ratio. Close-hauled, aspect ratio is a function of the freeboard (FBD), the effective height of the mast, and the nominal sail area (A_N). [9]

$$AR_{upwind} = \frac{\left(1.1(EHM + FBD)\right)^2}{A_N} \tag{30}$$

For any other sailing condition besides upwind (true wind angle > 45 degrees), the freeboard term is simply removed from the aspect ratio equation.

Windage (C_{DO}) is a function of the characteristics of the vessel, and is a crude method of determining the aerodynamic drag of the rigging and hull (Eqn (31)). [9]

$$C_{DO} = 1.13 \frac{\left(B_{MAX} \cdot FBD\right) + \left(EHM \cdot MD\right)}{A_N} \tag{31}$$

Windage is determined from the maximum beam of a vessel (B_{MAX}), the average freeboard, the effective height of the mast (EHM), the diameter of the mast (MD), and the nominal area of the sails (A_N). For the Mk II Navy 44, all of these variables were listed in the IMS certificate from David Pedrick.

Compiled from experimental sources, the IMS recently updated and published its coefficients of lift and parasitic drag which are used in their VPP. Each sail has its own coefficients. However, interaction between the headsail and the mainsail is neglected. For the mainsail, curve fit functions were created to turn the experimental data into a usable form for the VPP code. [2]

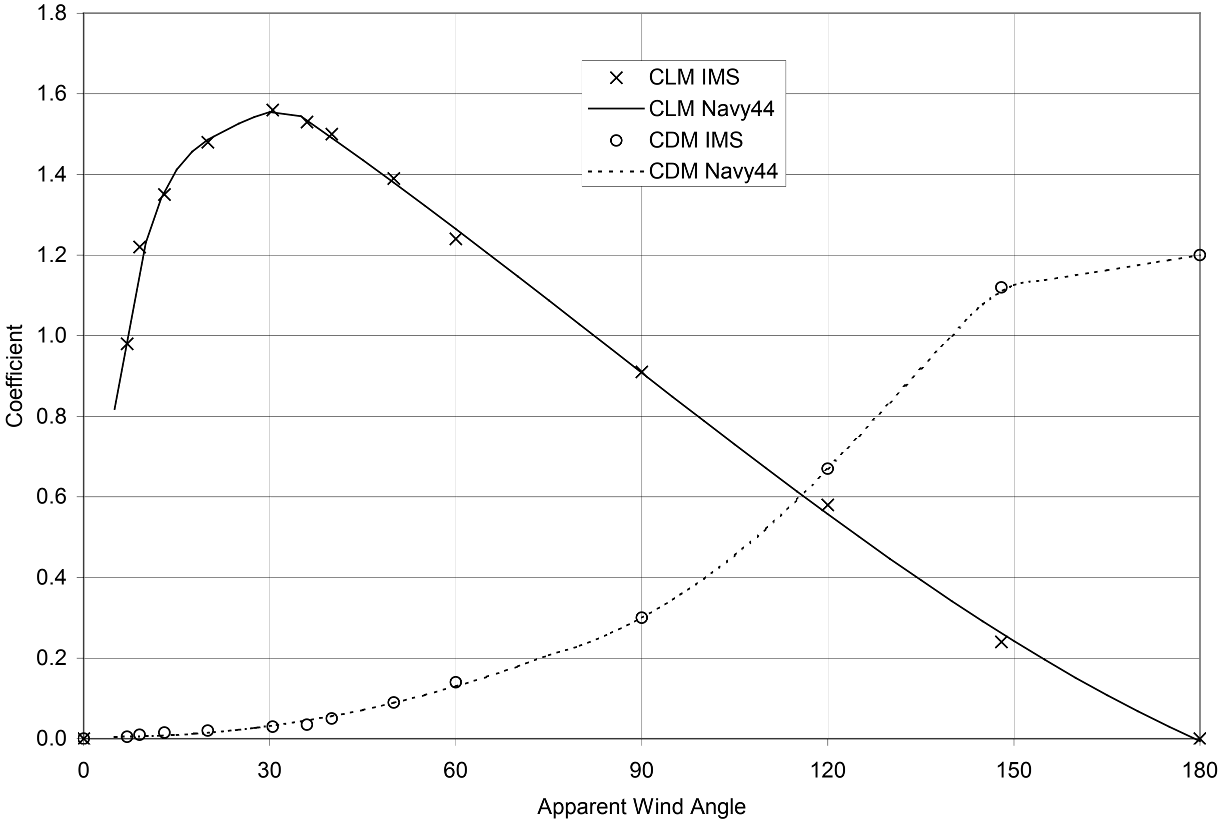


Figure 47 – Mainsail lift and parasitic drag coefficients

Figure (47) shows how both lift and parasitic drag vary as a function of apparent wind angle. Interestingly, high values of drag at wind angles aft of 90 degrees actually increase the speed of the vessel. At these wind angles, the sails act more to catch wind rather than to act as a wing.

The coefficients of lift and parasitic drag are presented in Figure (48). The values of lift and drag for the jib end at an apparent wind angle of 100 degrees. After this angle, the use of the jib would be of limited efficacy. Assuming that the rig could not be torn down for any wind speed, the use of a spinnaker at these high apparent wind angles would always produce faster boat speeds. If a jib was to be used at high wind angles, it would be “poled-out” and used on the opposite side of the boat. Experimental data of this effect was not used. Furthermore, a spinnaker would still be more efficient.[2]

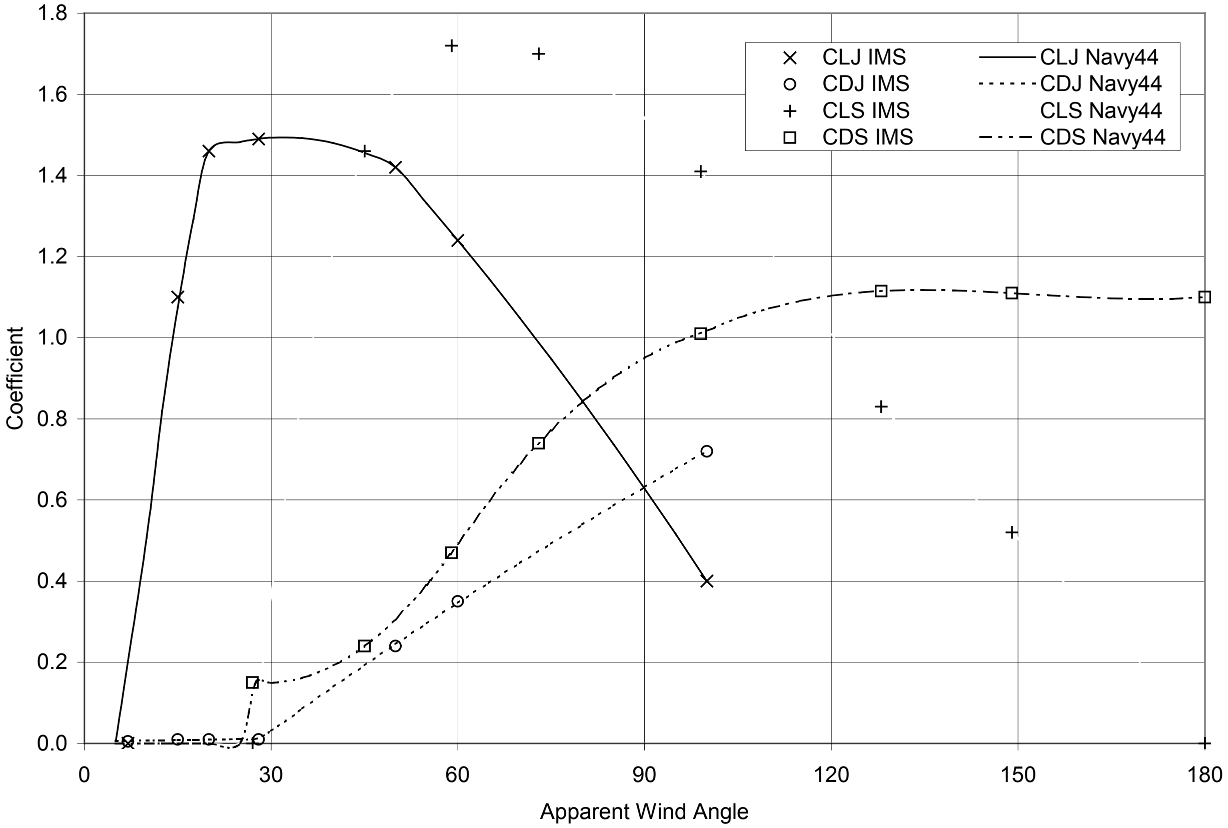


Figure 48 – Headsail lift and parasitic drag coefficients

The aerodynamic lift and total drag coefficients are recalculated to a new axis. Since these coefficients are still in the apparent wind axis, lift and total drag are recalculated as the variables drive and heel force. Furthermore, drag and heel force are calculated not being towards the bow, but in the direction of motion and in the plane of heel (Φ), not the waterplane with the inclusion of yaw angle (Figure (49)).

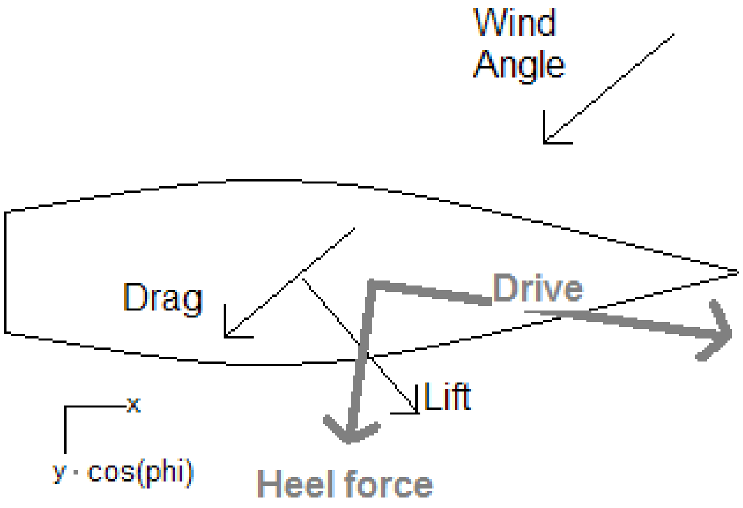


Figure 49 – Sail forces in the heeled xy-plane

The side force on the boat is the lateral force on the boat in the xy-plane. This is calculated as the heel force multiplied by the cosine of the heel angle. [5]

To calculate the total aerodynamic forces, a few more corrections were made. First, based on the Hazen model of aerodynamics, the sails could be trimmed to prevent overpowering the boat. Overpowering occurs when too much sail area is exposed, or when the exposed sail area creates too much lift. The typical effect of overpowering is that the vessel heels over too far. If heel becomes too great, deck-edge immersion occurs causing significant drag increases and a reduced maximum velocity. Furthermore, neither CFD or tank tests went so far as deck-edge immersion, so the boundary of the experimental data would be reached.

To incorporate sail trim, two variables are used in the calculation. The first variable, called "flat" acts as if the camber of the sail was being reduced. The effect of this is that coefficient of lift of the sail is reduced proportionally.[5]

The second variable, called "reef" acts as if the exposed sail area was reduced. The reefing variable affects three components. First, lift is reduced proportionally as for the flat variable. Secondly, the parasitic drag is reduced by a factor of reef squared. Finally, as if truly reefing a sail, the vertical center of effort of the sail is reduced. [5]

Another correction to calculate the actual sail forces is in the velocity variable (Eqn (28)). From the solver portion of a VPP, wind speed and direction are given as true wind speed and true wind angle. However, the aerodynamic functions for drag and lift are dependent on apparent wind speed and apparent wind angle. Furthermore, the velocity of the wind (V_{EFF}) at the height of the center of effort of the sails over the water (CE_z) is not the same as the true wind speed (V_{TWS}). Because of friction with the water, at low altitudes, the true wind speed is reduced. True wind speed can be found using Milgram's logarithmic function (Eqn (32)). [9]

$$V_{EFF}\left(CE_z\right) = \mathrm{V}_{TWS}(0.621154 + 0.10857(\ln(\mathrm{CE}_Z))) \tag{32}$$

Velocity has to be corrected two more times (Figure (50)). First, the true wind velocity in the y-direction must be factored by cosine of heel in order to find the velocity in the heeled plane.

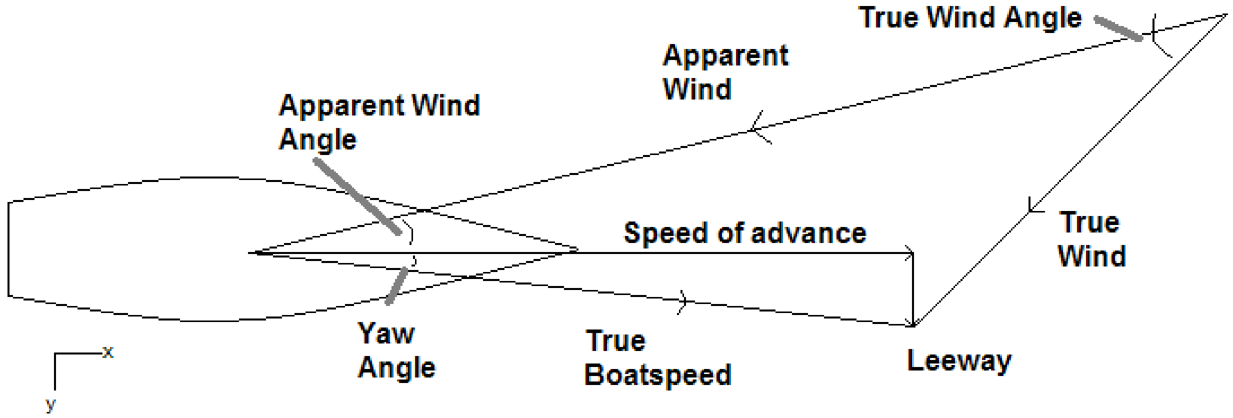


Figure 50 – Velocity diagram in the xy-plane

Finally, the vectors from boat speed and wind speed are added to create an apparent wind speed. The boat speed vector includes both speed of advance and leeway angle. Using the apparent wind speed in Eqn(28), the total aerodynamic heelforce and drive are calculated. [5]

The center of effort of each sail is calculated using approximations. For the mainsail, the vertical center of effort (CE_{mz}) is calculated using the vertical length of the sail (P), the height of the boom-above-deck (BAD), the height of the freeboard at the mast (FBD), and the heel angle (Φ). The 0.39 value is an approximation that the center of effort on the sail is 39 percent from the bottom of the sail assuming the sail is semi-triangular in shape. [9]

$$CE_{mz} = (0.39P \cdot reef + BAD + FBD)\cos(\Phi) \tag{33}$$

The center of effort of the jib (CE_{jz}) uses the same calculation as that for the mainsail, except the sail is assumed to start at the deck, not at a certain height above the deck (BAD). Furthermore, the vertical length of the jib (I) is used instead of the luff of the main (P). [9]

$$CE_{jz} = (0.39I \cdot reef + FBD)\cos(\Phi) \tag{34}$$

The center of effort of the spinnaker (CE_{sz}) uses a calculation also similar to the mainsail. Because of the shape of the spinnaker, the center of effort is assumed to be 50 percent from the bottom of the sail. However, the bottom of the spinnaker normally begins at the bottom of the mainsail (BAD+FBD). The value EHM is the effective height of the mast and extends from the deck to the top of the mast where the spinnaker begins. [9]

$$CE_{sz} = (0.5EHM \cdot reef + BAD + FBD)\cos(\Phi) \tag{35}$$

Furthermore, the longitudinal centers of effort of each sail can be found as well. For upwind sailing, the angle of attack on the mainsail is typically around 15 degrees while the jib is at 12.5 degrees. As the wind moves aft, these angles increase to 27 degrees for both sails when the apparent wind is at 90 degrees. Finally, both sails have angles of attack approximated at 90 degrees when the wind is fully aft of the ship. For a triangular three dimensional foil under typical sailing conditions, the longitudinal center of effort can be found from Figure (51). [8]

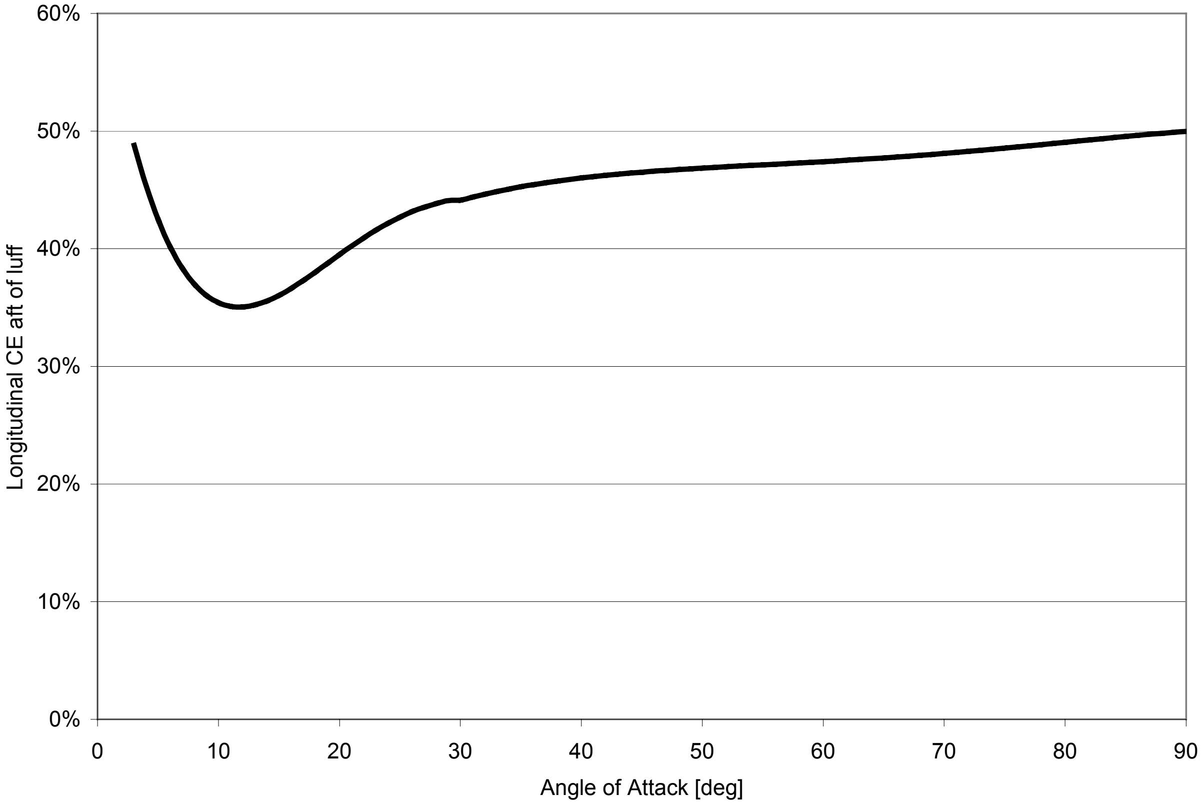


Figure 51 – Center of effort for a three-dimensional triangular foil with moderate camber

Heel force and drive, however, are located on the vertical center of effort of the sails (CE_Z) above the datum on the z-axis. At this center of effort, two more variables, the heeling moment and the trimming momentum can be calculated using CE_Z as the moment arm. The rolling moment is the heel force, renewed at an axis perpendicular to the length of the boat, multiplied by the arm CE_Z. Heeling moment calculates the moment which heels the boat. Trimming moment is renewed at an axis parallel with the length of the boat. Trimming moment calculates the moment which pushes the bow of the boat down.

By multiplying the longitudinal center of effort moment arm of each sail by the xy-planar forces from each sail, the yawing moment due to aerodynamics can be calculated.

Hydrodynamic and Aerodynamic Solution - Tank Data

Since both the aerodynamic and hydrodynamic forces could now be calculated, a solution had to be found to equate all forces in order to maintain equilibrium. From the tank data, the drag data was scaled using the Froude hypothesis to extrapolate to full-scale resistance. The resistance could be equated to the drive variable.

$$FroudeExtrapolation(Drag) = Drive \tag{36}$$

Furthermore, the total lift measured in the tank was the summation of the lift measured from the forward force block and the lift at the aft force block. This summation multiplied by the cubed scaling factor (λ) could be equated to the aerodynamic sideforce.

$$(LiftAft + LiftFwd) \cdot \lambda^3 = Sideforce \tag{37}$$

After using FASTSHIP to find the hydrostatic righting moment of the ship, it was coded as an automatic visual basic macro (RM(Φ)) which provided the static righting moment as a function of heel. However, to find the hydrodynamic righting moment, three approximations were made. First, it was assumed that the data from the forward lift block closely approximated the force generated by the keel. Secondly, the data from the aft lift block approximated the force generated by the rudder. Third, it was assumed that the lift of the keel acted on a point at ½ the draft of the keel, and the lift of the rudder acted on a point at ½ the draft of the rudder. These dynamic functions of lift acted against the hydrostatic righting moment of the vessel. Therefore, in summation, the righting moment could be equated to the heeling moment.

$$RM(\Phi) - Liftfwd \cdot \lambda^4 \cdot \tfrac{1}{2} \cdot Draft_{keel} - Liftaft \cdot \lambda^4 \cdot \tfrac{1}{2} \cdot Draft_{rud} = Heelingmoment \tag{38}$$

Finally, the yawing moment from the tank data was the force from the aft force block multiplied by a moment arm from the pin location to the heave post. This moment could be equated to the aerodynamic yawing moment.

$$LiftAft \cdot Arm \cdot \lambda^4 = Yawingmoment \tag{39}$$

Hydrodynamic and Aerodynamic Solution - SPLASH Data

In general, the SPLASH solution was identical to the tank solution. Both solutions used the same aerodynamic equations. One advantage of the SPLASH data was that scaling did not have to occur. However, in order to gain better correlation to the tank data, the same viscous stripping method was used. The viscous component of SPLASH's prediction was subtracted from its total resistance calculation. The viscous prediction using the ITTC friction equation and the form factor from the tank were used instead. The equations used were:

$$Resistance_{SPLASH} = Drive \tag{40}$$

$$Lift_{SPLASH} = Sideforce \tag{41}$$

$$RM(\Phi) - Lift_{SPLASH} \cdot \tfrac{1}{2} \cdot Draft_{keel} = Heelingmoment \tag{42}$$

$$Yawingmoment_{SPLASH} = Yawingmoment \tag{43}$$

VPP Predictions

Now that four equations were made to satisfy equilibrium, a prediction code was necessary to find the optimum speed.

Normally, a VPP will predict conditions which create the greatest boat speed for a given wind angle and direction. Some modification of this was necessary for the acquired data set. First, the Navy 44 VPP included yaw angle as a variable. If solving an upwind condition based upon greatest boat speed alone, the VPP would predict extremely high yaw angles (which would be caused by large amounts of sail area which create high boat speeds). However, the high yaw angle would not mean the fastest upwind course, or highest VMG. Instead, the solver was made to determine the greatest VMG, not boat speed for wind angles where a sailor would be trying to sail upwind or downwind. For close reaching through broad reaching wind angles, a sailor would be content with the maximum boat speed since VMG is not the goal. Because of this, from 30 to 50 degrees and from 150 to 180 degrees true wind angle, the solver was set to solve for maximum VMG, not boat speed. This was set as the variable "maxspeed."

In solving for maxspeed, only two variables were constant: true wind speed and true wind angle. Initially, a method of predicting the optimum condition was used similar to PCSail's method. This involved the use of the SOLVER function in Excel. By holding the solver to the requirements that the four equilibrium equations must not exceed 0.05% error, one variable, maxspeed, could be maximized. Items changed were boat speed, yaw angle, heel angle, rudder angle, reef, and flat.

Ultimately the SOLVER method was discontinued from use in the code. The SOLVER code made predictions which were unreasonable, but which were supported by the data. The problem was that in assembling the data matrix, allowances were not made on which headsail was theoretically flown. In tank testing, the trim weights were moved which simulated the trimming moment as a result of either the jib or the spinnaker. In Figure (19), these conditions can be shown as clockwise patterns for each wind condition. The jib is flown first at the lowest speed (which is not zero), indicating a close-hauled VMG condition. The jib is flown until the third following point where it was predicted the spinnaker would be flown instead. The spinnaker continues use along each clockwise pattern until a heel angle of zero degrees which is assumed to be a downwind VMG course.

The solver would often predict heel angles for upwind conditions at downwind heel angles and vice versa. Furthermore, the SOLVER was very sensitive to any dips in drag for a given heel angle with respect to constant velocities. The SOLVER would often predict the same heel angle for a large range of speeds. Finally, with four sets of equations as well as large matrices of data used for interpolation, the SOLVER function on Excel was very slow.

A new method was created which used circular functions in Excel. All four equilibrium conditions produced a set of four simultaneous equations. By breaking the equations into their hydrodynamic and aerodynamic components, a matrix could be formed which represented the equations. The four tank data equilibrium equations in matrix form are shown in Eqn (44).

$$\left[\begin{array}{cccc|c} resistance & 0 & 0 & 0 & Drive \\ 0 & -Liftfwd \cdot \lambda^4 \cdot \tfrac{1}{2} Draft_{keel} & -Liftaft \cdot \lambda^4 \cdot \tfrac{1}{2} Draft_{rud} & RM(\Phi) & HeelMoment \\ 0 & Liftfwd \cdot \lambda^3 & Liftaft \cdot \lambda^3 & 0 & Sideforce \\ 0 & 0 & Liftaft \cdot Arm \cdot \lambda^3 & 0 & YawMoment \end{array}\right] \quad (44)$$

By taking the matrix multiplication of the inverse hydrodynamic matrix with the aerodynamic matrix, a solution matrix can be formed. This solution matrix represents the factors by which resistance, righting moment, fwd-lift, and aft-lift should change in order to come to an equilibrium. The SPLASH data matrix was identical to Eqn (44) with corrections made using

Eqns (40-43). The solution for the SPLASH data represented the factors by which resistance, righting moment, lift, and yawing moment should change.

Four simplifications were made in the iterative approach. First, resistance was assumed to be a function mostly of velocity. Secondly, righting moment was assumed a function of heel angle. For the tank data, Fwd-lift was assumed a function of yaw, and aft-lift was assumed a function of rudder angle. For the SPLASH data, lift was assumed a function of yaw, and yawing moment a function of rudder angle. By modifying these four variables by small amounts of the solution matrix, the spreadsheet eventually settled on a solution which was in equilibrium. This simple finite difference method was written directly into the spreadsheet cells themselves. Each condition took at most 5 seconds to solve.

Consideration in the finite difference solver had to be taken in the movement of the variables. If an initial degree of freedom (heel, yaw, rudder angle, or velocity) was terribly unrealistic for wind speed and wind angle, the iterations had the ability to self-destruct and some components would go to infinity. This would stop all calculations. Self-destruction would occur if the solution matrix steps were not small enough to fall within the limits of the data. Furthermore, if the matrix steps were too large, the program would end up iterating between two conditions forever, never falling into the set maximum error of 0.05%. The problem with setting the steps too small, however, was that the program could take hours to run a complete set of conditions. A solution was found in that the step size would decrease per number of iterations. If the code were to self-destruct, it would automatically reset and start again at a new initial positions with a smaller step size than before.

The next step was to determine how the sails should be set to maximize maxspeed. By starting at a predetermined condition of reef and flat, reef and flat would be changed to test for greater conditions of maxspeed. For wind speeds greater than 18 knots, it was found that reef and flat should be initially set to 0.8 for both variables. At mid-ranged wind speeds, the initial reef and flat was at 0.9. For wind speeds less than 12 knots, reef and flat were both set at their maximum values of 1.0. Finally, the spinnaker was always flown at true wind angles greater than 90 degrees.

After many trials with the new VPP, it was found that the rudder angle was not being correctly predicted. This was due fully to the longitudinal aerodynamic center of effort calculation. While this calculation approximated the center of effort for moderate wind speeds and close-hauled angles, the approximation broke down as reef and flat changed the shape and and camber of the sail. Furthermore, sailors have other controls to balance the yaw of the boat besides rudder angle. Mast rake and other minor sail controls can significantly change the aerodynamic yawing moment of the boat without much loss of boat speed. The solution for this VPP was that this fourth equilibrium condition was taken out of the solver. Instead, the rudder position was set at 0 degrees while reef and flat iterated. Once a final sail solution was found, the rudder angle would iterate between 0 and 6 degrees in order to maximize maxspeed. However, the yaw-moment calculation still remained in the code. The Yaw-moment imbalance has plagued researchers and has not yet been adequately solved.[14] A partial solution was reached and provided the preferred mast rake position which is valuable information for mast tuning.

[14] Discussion with Naval Architect Bill Cook, *Oracle* Performance Manager, in March 2003

PERFORMANCE PREDICTION COMPARISON

Polar diagrams are plots in which boat speed is a function of true wind angle for a given wind speed. For sailors, polar diagrams help to tune a boat and to find the quickest course to a mark. For designers, polars help provide comparisons between boats and between different configurations of the same boat. Ultimately, sets of polars were made which compared the IMS VPP, PCSail, tank testing, and SPLASH predictions. Although an FKS VPP was attempted, its results were of limited use since lift circulation was not calculated in the CFD code. With four different methods at four different wind speeds, it became difficult to interpret all predictions on the same plot. Therefore, the results were investigated at each wind speed.

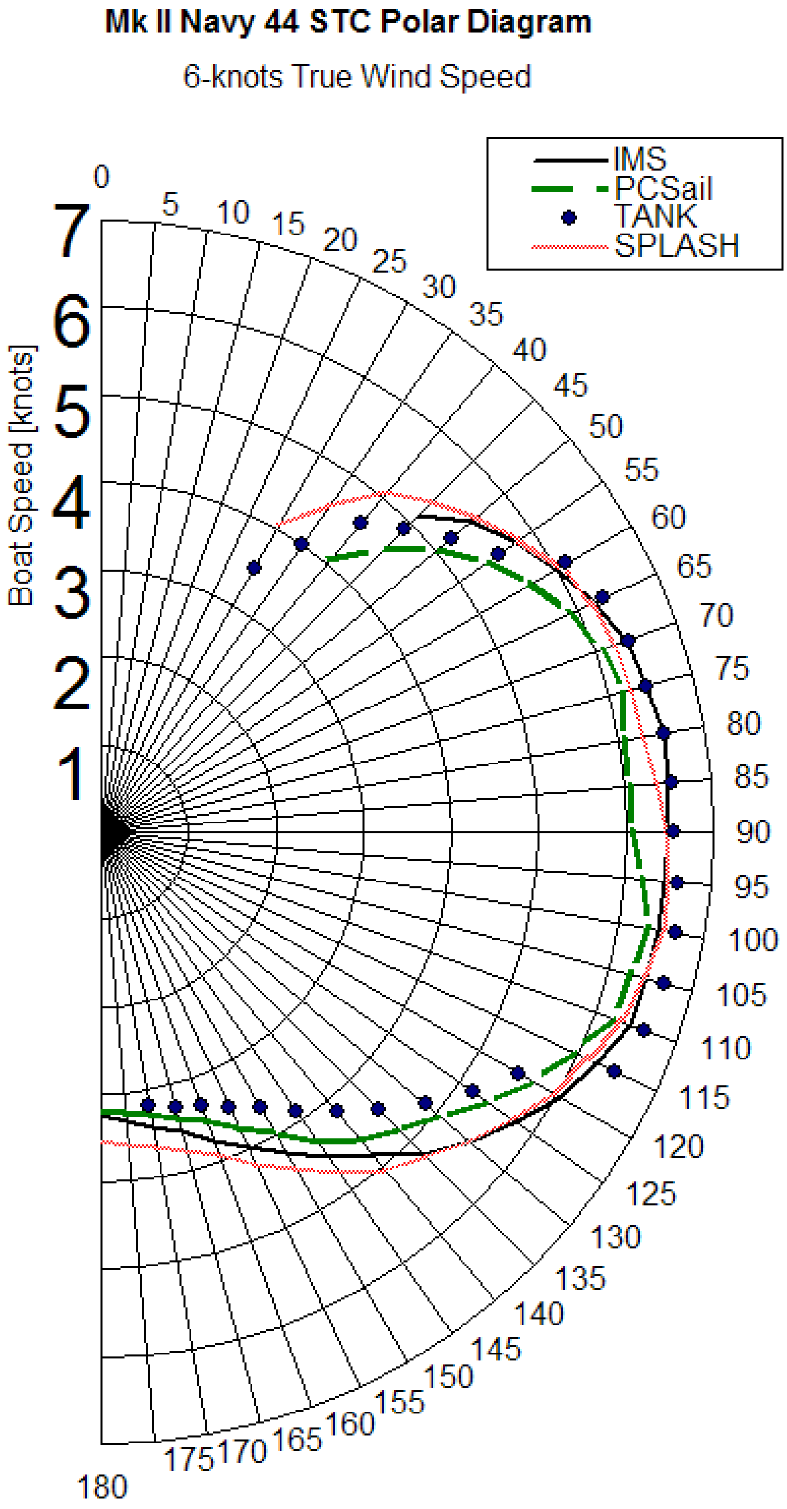


Figure 52 – Polar diagram for 6 knots true wind speed

The 6-knot wind speed polar diagram showed the most deviation between each prediction method for a given wind speed. Still, the average deviation was within 0.19 knots. In the 6-knot condition, the IMS VPP was assumed to have the greatest accuracy because of its complex aerodynamic algorithm. Using the IMS VPP as the baseline, noticeably lower speed predictions were made in tank testing particularly at deep reaching angles. This was considered error due to scale effects at slow model speeds and sub-turbulent Reynolds numbers.

SPLASH, however, showed results which were very consistent with the IMS predictions with an average deviation of less than 0.08 knots. Even though the aerodynamic algorithm was shared by the tank VPP and the SPLASH VPP, the interaction of the hydrodynamics and aerodynamics in the solution lead to a chance of decreased accuracy. For instance, if tank testing miscalculated resistance due to scale effects, the accuracy of the aerodynamic iteration would also be reduced.

PCSail showed consistently lower velocity predictions for this windspeed. Because PCSail used a much smaller parametric database than the IMS VPP, these predictions deviated both due to a lack of sufficiently populated hydrodynamic data as well as a less complex aerodynamic functions. Again, the iteration of the hydrodynamic and aerodynamic models lead to errors in this low wind condition. In a practical sense however, the lower accuracy of the prediction methods may not be too significant for the future uses of the Mk II Navy 44. To meet Naval Academy Sailing schedules, the engine is usually turned on when boat speeds fall below five knots. Furthermore, races are rarely started until wind speeds top six-knots.

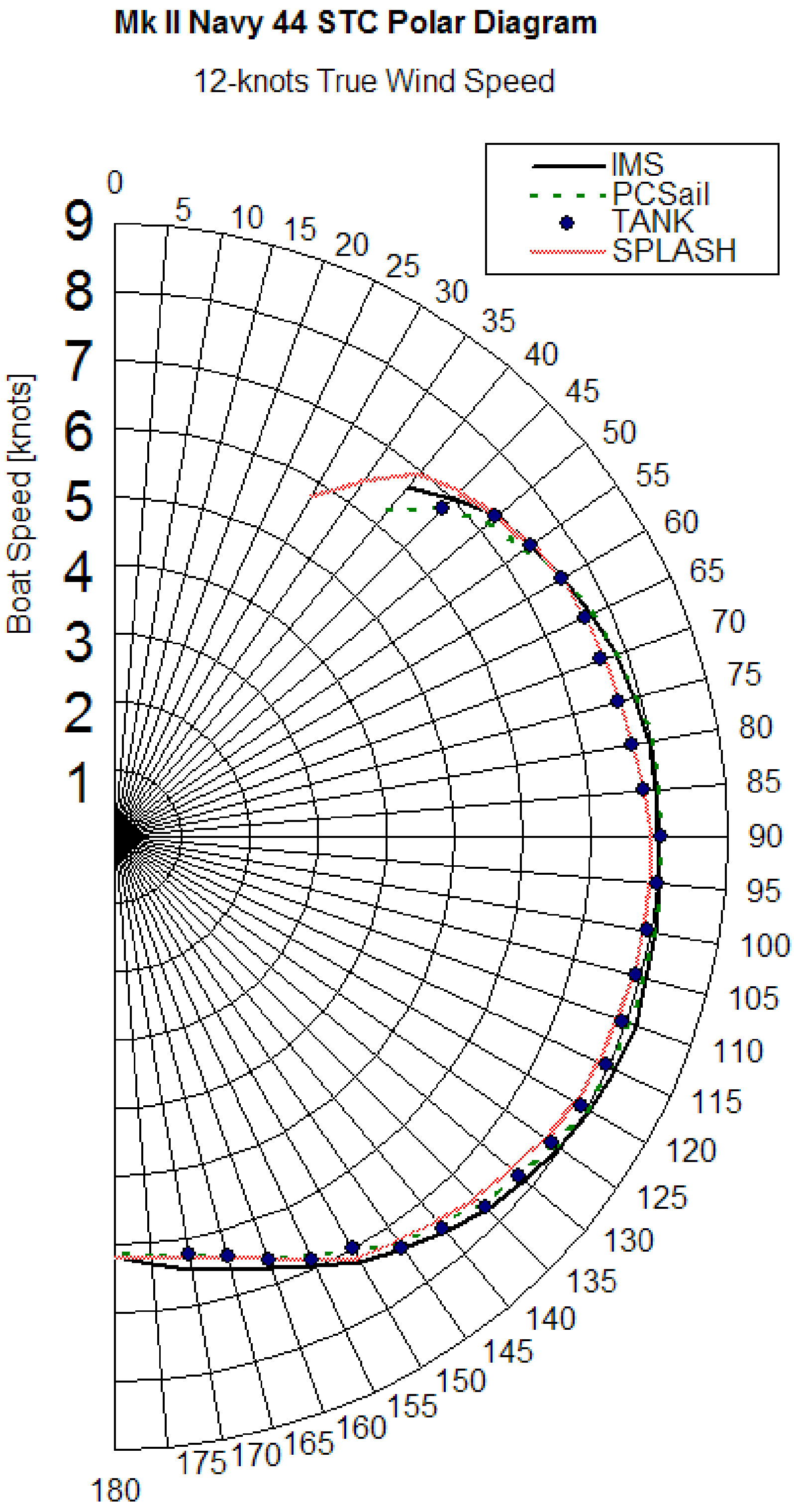


Figure 53 – Polar diagram for 12 knots true wind speed

The 12-knot wind speed data showed excellent correlation between all four methods. The average deviation between all of the methods was less than 0.08 knots.

The results at this wind speed were significant in that 12-knots true wind speed is the normal wind speed which the Mk II Navy 44 STC will see in operation. In both the Chesapeake Bay and near the East Coast in the Atlantic Ocean, the Navy 44 will often find itself sailing in 12-knots of wind.

The 12-knot wind condition showed the least deviation of the experimental data because the tank testing data was out of the scale effects region, and the SPLASH data was unhindered by any discontinuities in the free-surface continuum.

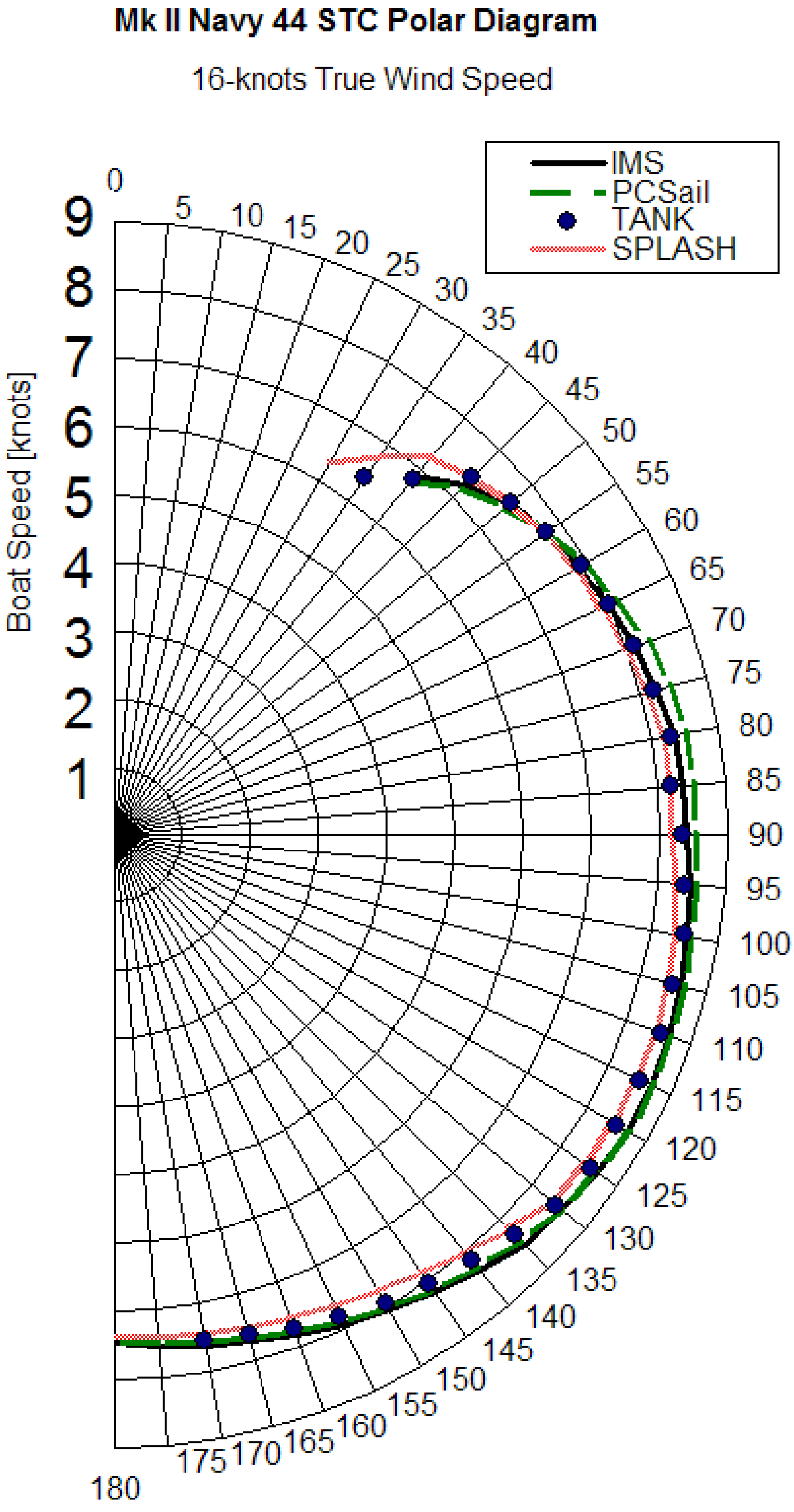


Figure 54 – Polar diagram for 16 knots true wind speed

At 16-knots of wind speed, all methods predicted very similar results upwind (45-60 degrees). However, as the boat turned downwind, SPLASH predicted less boat speed. At the higher speed range, SPLASH predicted a greater resistance than tank testing because of its different method of viscous calculations. The tank data mostly fell between the IMS and the SPLASH predictions.

Furthermore, both SPLASH and tank testing predicted velocities an average of 0.14 knots lower than the IMS predictions. Since the SPLASH and tank VPPs solved the yaw balance, this error was found to be in the IMS VPP since it could not calculate the specific effects of the lifting surfaces (such as vortices and wake shed from of the appendages).

Overall, at 16-knots of wind speed, the average deviation for the methods was less than 0.10 knots.

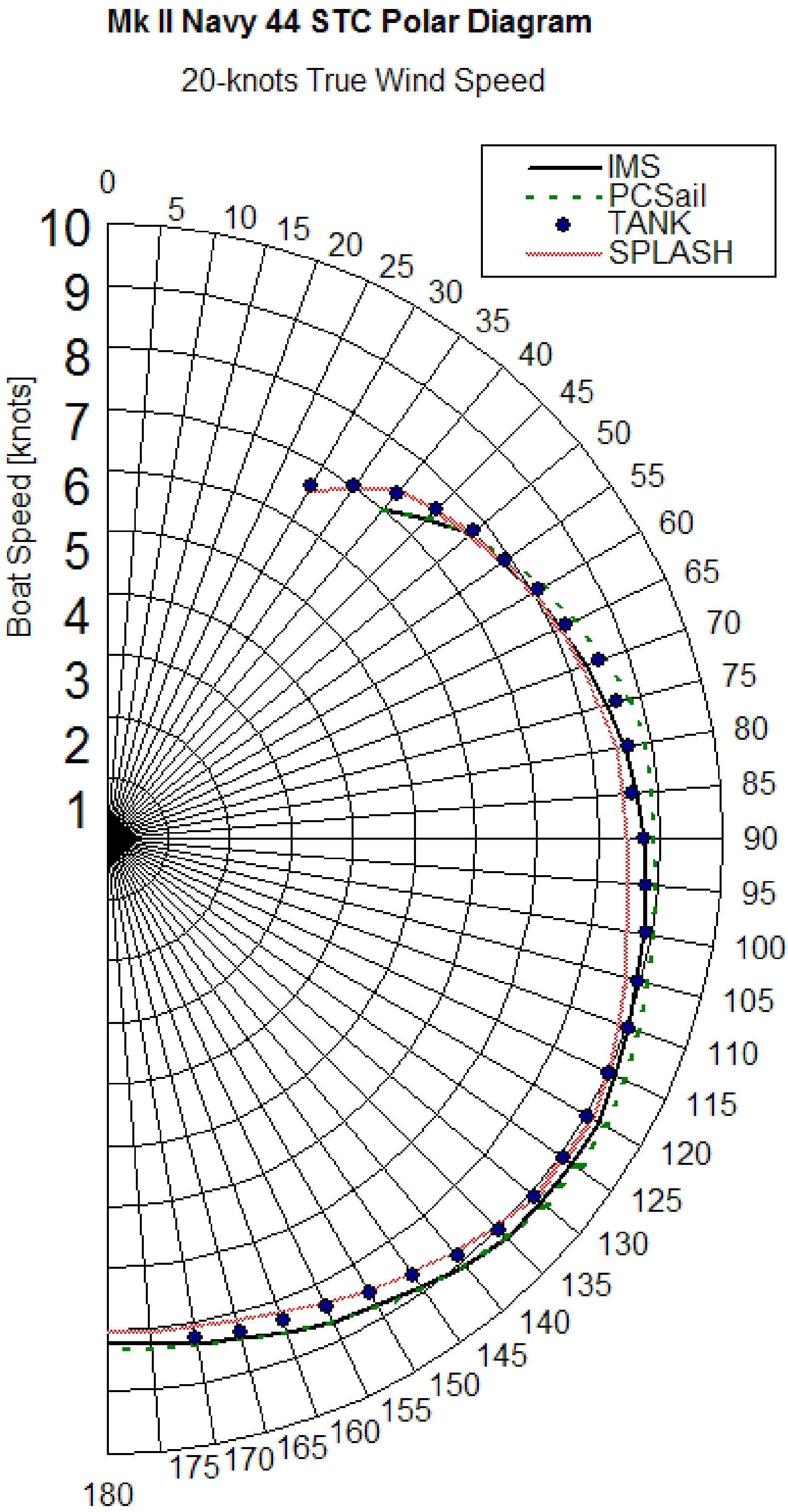


Figure 55 – Polar diagram for 20 knots true wind speed

At 20-knots wind speed, the tank and SPLASH VPPs continued to predict boat speeds generally lower than the IMS VPP. The lower boat speed of SPLASH as compared to tank testing near beam-reaching angles (85-105 degrees) again reflects the differences in the viscous calculations used by both methods.

While PCSail showed minimal deviation with the IMS prediction at downwind courses, PCSail overestimated boat speed in close-reaching conditions (65-85 degrees). Again for PCSail, this error was due to not having as robust a hydrodynamic dataset compared to the IMS VPP.

PERFORMANCE PREDICTION CONCLUSIONS

Ultimately, writing the new customized Mk II Navy 44 VPP was the only way to compare different methods of performance prediction. The end result was to say that the customized VPP provided very good correlation to both the PCSail code and IMS code in terms of relative performance between wind angles and wind speeds. The customized VPP also made rudder and yaw angle predictions which neither the PCSail or IMS VPP calculated. Therefore, the customized VPP made a more complete performance prediction which also provided more input to sailors in how to sail the Mk II Navy 44.

In comparing the hydrodynamic predictions, tank testing was the easiest method used to evaluate the performance of the boat. Tank testing had the benefit of many years of previous research to help fine tune the performance prediction process. However, tank testing required many hours spent in the tow tank. The hours spent testing many conditions meant that most conditions were tested only once or twice. There was no possible analysis for evaluating the precision of each run.

Both CFD codes had the advantage of having 100% precision in their tests. FKS, however, proved that having 100% precision was meaningless if the accuracy was insufficient. FKS showed itself to be an excellent tool for predicting the upright performance of vessels. Once the meshed hull was built, a full range of upright tests took twenty minutes. This compared very favorably to a week's worth of tank testing.

Because of the far-field calculations used, FKS was shown to be very sensitive to inviscid effects and assumptions from slender-ship theory. This resulted in the large humps and hollows in the coefficient of resistance (Figure (34)). FKS was not written to calculate the effects of lifting surfaces, therefore only upright conditions were possible. In the end, FKS was shown to be an excellent numerical approximation of a closed-form inviscid resistance calculation.

SPLASH showed impressive accuracy compared to tank testing and the IMS predictions. The main advantage of SPLASH was that the hull was free to move and pivot in any direction based on aerodynamic and hydrodynamic forces. This allowed for an improved mesh over the hull and therefore more accurate results.

The disadvantage to SPLASH was from a user's perspective. Being such a complex code, SPLASH was difficult to make operational. However, once working, SPLASH was very fast to run. Whereas the full sailing matrix took weeks to perform in tank testing, a similar matrix took nineteen hours to calculate with SPLASH.

Based on all the research, the performance prediction process for the Mk II Navy 44 STC was completed by making a final set of polars.

By taking input from each method, a best-fit spline was constructed for each wind speed. In some areas, the IMS VPP was deemed the most accurate (especially at 6-knots wind speed because of its complex aerodynamic code). However, for higher boat speeds, tank testing and SPLASH conditions were always incorporated because of their highly accurate hydrodynamic data (Figure (56)).

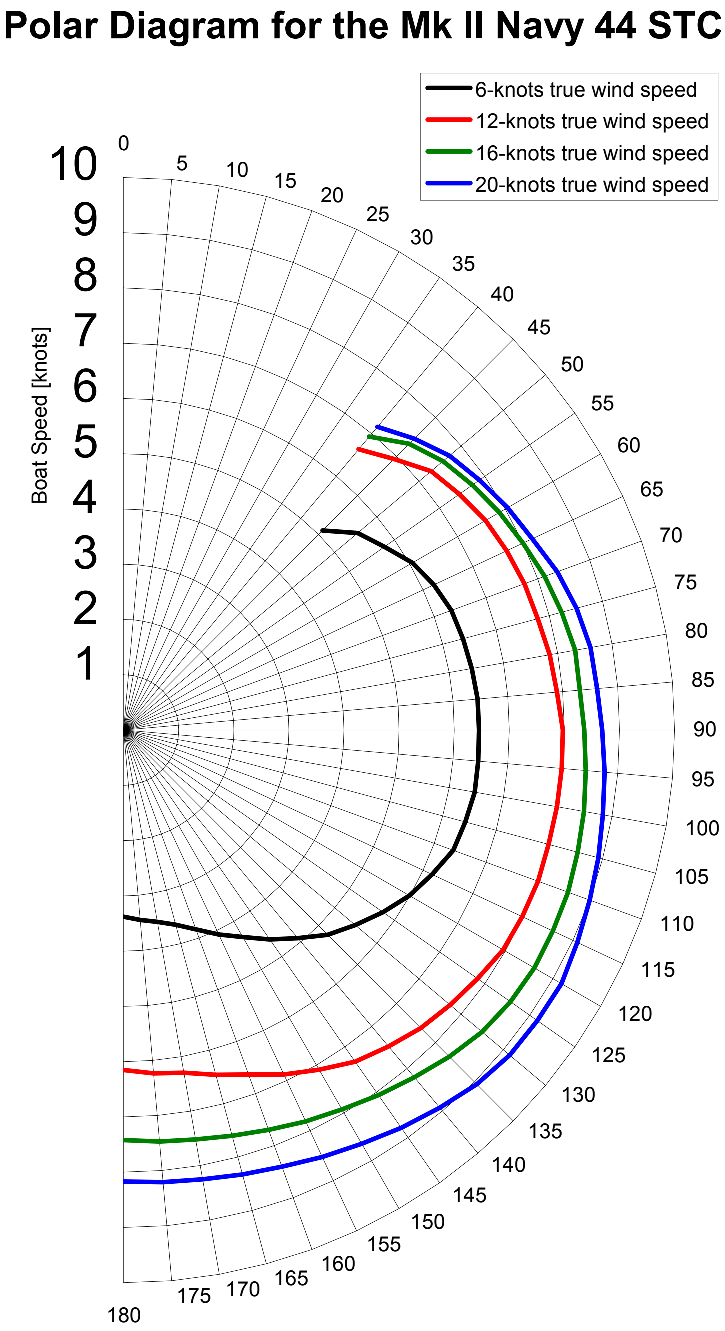


Figure 56 – Polar Diagram for Mk II Navy 44 STC

Given the inherent errors present in scale effects and flow simulation, the use of tank testing to setup and normalize CFD codes would produce the prediction with the least amount of error. For sail craft, once a hydrodynamic prediction is made, its VPP should be used alongside predictions from other sources for a full estimate of the performance of the vessel.

Once a prototype Mk II Navy 44 STC is built, full-scale trials could validate some of the results, although the setup and testing for full-scale tow trials would be difficult. In the short term, another option to add to the data set would be to conduct tank testing of a model twice or even three times the size of the model used in this study. While taking more time to run the tests, this would improve the accuracy of the tank testing data and would reduce the scale effects present in the results.

A further option would be to conduct CFD tests using a viscous code. While these codes take substantially longer to run, the flow modeling on the ship would be improved.

Finally, although not a prediction, an accurate GPS along with months of sailing on the prototype would record the optimum sailing speeds and angles. This technique would validate the predictions made in the design phase.

RUDDER REDESIGN

Since SPLASH had shown its accuracy in the previous section of performance prediction, it was used to help perform a rudder redesign. Rudder experimentation is an area in yacht design with limited development. Indeed, besides wind tunnel testing with very large appendages for IACC yachts, most sailboat rudders are designed with little experimental data. While most of the hull design of the Mk II Navy 44 STC is nearly finalized, the rudder design is not close to completion. At this stage of the design of the Navy 44, designer David Pedrick has simply added a generic rudder found in other typical sailboats. By completing a systematic rudder design for the Mk II Navy 44 STC, the knowledge of rudder development can be improved.

There were three comparisons made in the rudder redesign:

1) Tank testing versus SPLASH
2) Various planform shapes
3) Rudder location and depth

The tank testing comparison evaluated two rudders in SPLASH and in the tank. The first rudder, named the "Pedrick" rudder, was the rudder which was supplied by naval architect David Pedrick with the original lines of the Mk II Navy 44. Bill Beaver of the USNA Laboratory Support Staff built the second rudder, named the "Beaver" rudder. Bill Beaver's design was of an elliptical rudder far larger than the Pedrick rudder.

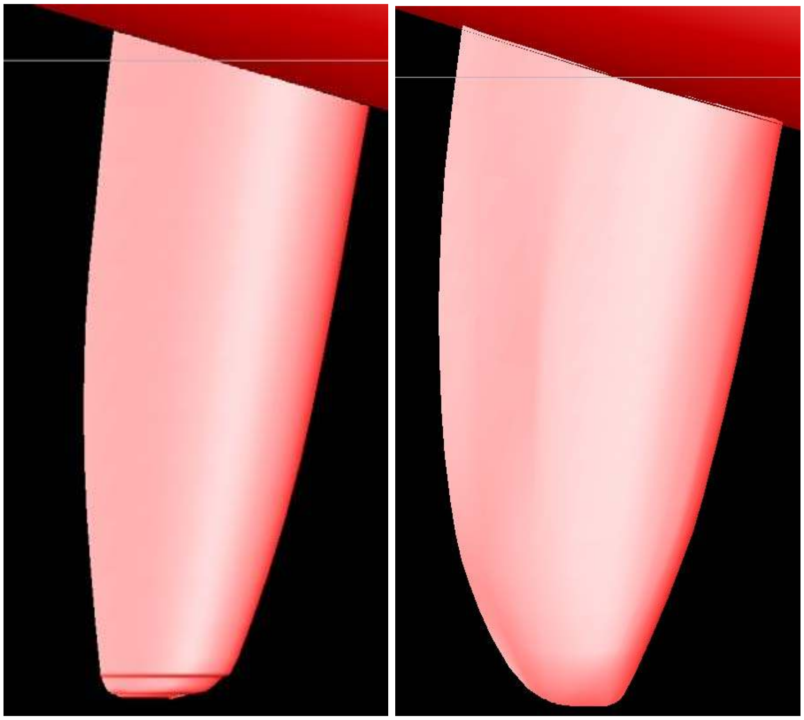


Figure 57 – Planforms of Pedrick (left) and Beaver (right) rudders

The Beaver rudder was positioned at the same rudder stock location as the Pedrick rudder.

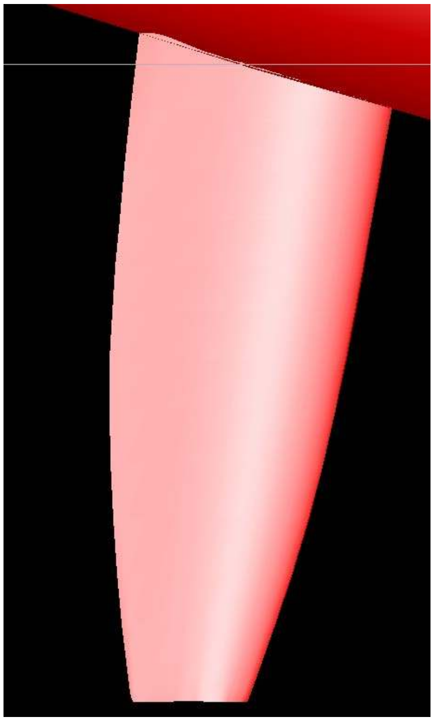


Figure 58 – Planform of the Baseline rudder

The Planform testing evaluated SPLASH's capabilities at detecting changes in the planform of a rudder. There were six rudders used for comparison. The first rudder was the Pedrick rudder. The second rudder was named the "Baseline" rudder as it was very similar to the Pedrick rudder with slight modifications to the tip. The tip of the Baseline rudder was faired and the base of the tip was made flat instead of rounded as in the Pedrick rudder. The Baseline rudder acted as the starting point for the rest of the modified rudders of these series. The Baseline rudder maintained the same rudder shaft location as the Pedrick rudder. Furthermore, for the entire series of rudders, the section type of each rudder was kept the same as the Pedrick rudder.

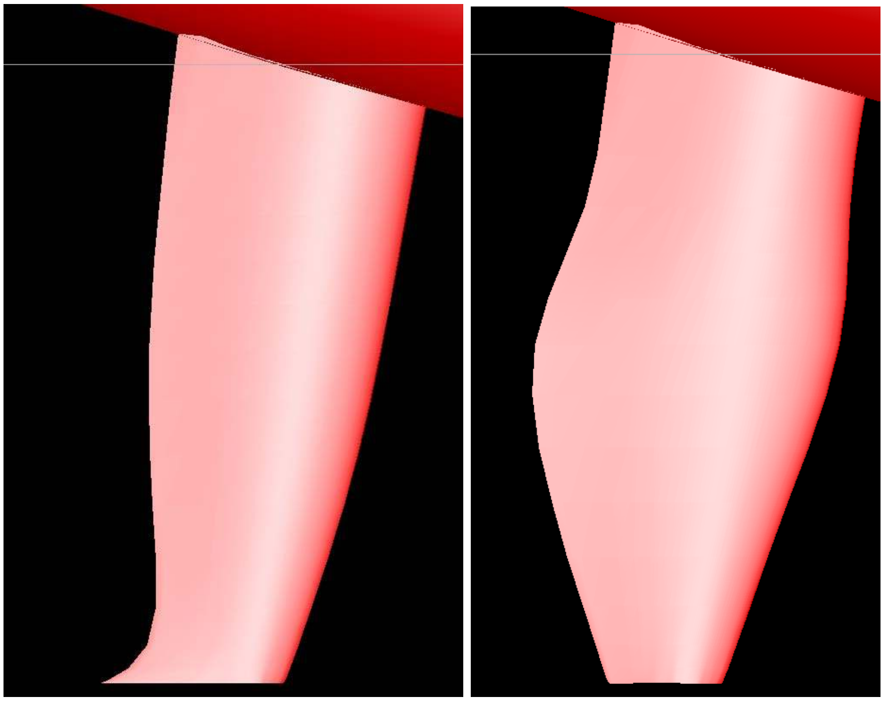


Figure 59 – Planforms of the Tip (left) and Bulge (right) rudders

There were four modifications of the Baseline rudder. The "Tip" rudder was the Baseline rudder with the tip extended aft to investigate fluid flow at the end condition. The "Bulge" rudder was the Baseline rudder modified so that the mid portion of the rudder was 1.3 times the chord of both the root and tip. The Bulge rudder was constructed to investigate the effects of fluid flow at the center of the span.

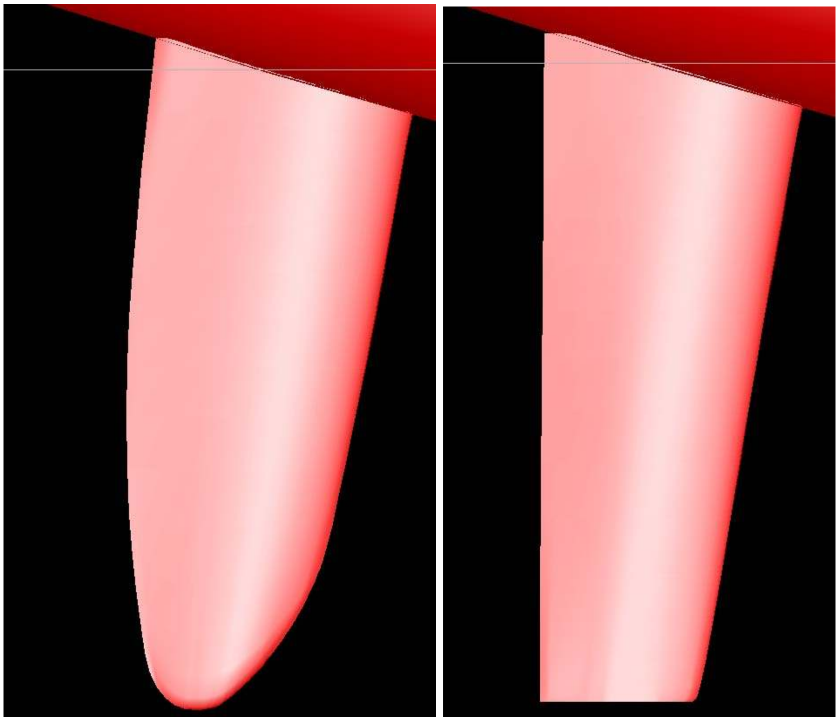


Figure 60 – Planforms of the Elliptical and Zoid rudders

The "Elliptical" rudder was the Baseline rudder which was faired into a semi-ellipse. Elliptical planforms have a tradition of use as wings since they reduce induced drag. Furthermore, elliptical rudders maintain a near constant coefficient of lift across the span. This creates even stalls across the entire rudder. The "Zoid" rudder was a trapezoid rudder of the same dimensions as the elliptical rudder. Trapezoidal planforms for wings have similar lift and drag characteristics as elliptical rudders but are easier to construct.

The location and depth testing included four rudders. These tests were conducted to investigate effects of aspect ratio as well as location of the rudder relative to the keel. Often, the vortices from the keel can act against the lift-producing effects of the rudder. However, if a rudder is positioned too far aft, it can be exposed above the free surface and lose its efficiency. Again, the Baseline rudder was used as the baseline planform for the depth testing as well as the baseline location. The "Maxdepth" rudder was a scaled version of the baseline rudder where the tip of the rudder came to 0.85 of the draft of the keel. This is the maximum draft the rudder could be with respect to good design practice. If any longer, rudder damage due to grounding would become more probable. The "Halffwd" rudder was a movement of the Baseline rudder to a position one half of a span length forward of its normal position. The "Onefwd" rudder was a movement of the Baseline rudder to a position one full span length forward of its normal position. Both the Halffwd and Onefwd rudders had their depths held at the same depth of the Maxdepth rudder. Figure (61) shows the Baseline rudder interposed in the pictures of the Halffwd and Onefwd rudders.

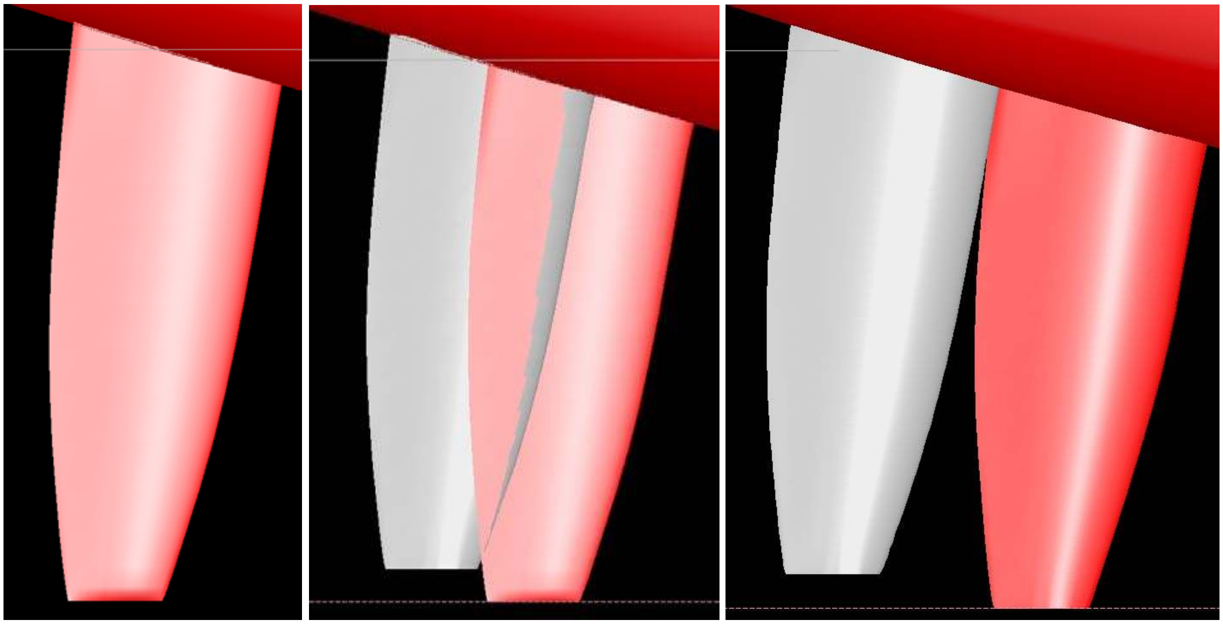


Figure 61 – Rudder planforms for Maxdepth, Halffwd, and Onefwd, respectively

Tank Comparison

The two rudders tested both in SPLASH and in the tank were the Pedrick and Beaver rudders. For both methods, three yaw angles were tested of 0, 4, and 6 degrees, with four rudder angles of 0, 2, 4, and 8 degrees. All tests were conducted with zero heel.

Since actual rudders were being evaluated, not just shapes of rudders, the requirement was made that the better rudder had two qualities: a higher lift to drag ratio and a higher yaw-moment to drag ratio. The higher lift to drag ratio would provide faster sailing while on a constant course since the lift of the rudder acts against some of the aerodynamic sideforce. The higher yaw-moment to drag ratio would provide for faster boat-handling and turning. This is important for inshore races where maneuvering is very important. Likewise, as a training vessel, the ability to turn the Mk II Navy 44 STC would add to the safety of the crew.

In the original assumptions about appendages, yaw was assumed to have a linear affect on both lift and drag, while rudder angle was assumed to have a quadratic affect. If plotting Lift versus Drag in which both yaw and rudder change, the effect would be cubic. Figure (62) shows the Pedrick and Beaver rudders tested both in SPLASH and in the tank. A general cubic fit of the data was observed, validating these assumptions.

SPLASH and tank testing analysis showed the same trend for both rudders. The Pedrick rudder produced more lift for less drag at low lift conditions, but the opposite was true at high lift conditions. This can be further described as that the Pedrick rudder was more efficient at low angles of attack while the Beaver rudder was more efficient at high angles of attack.

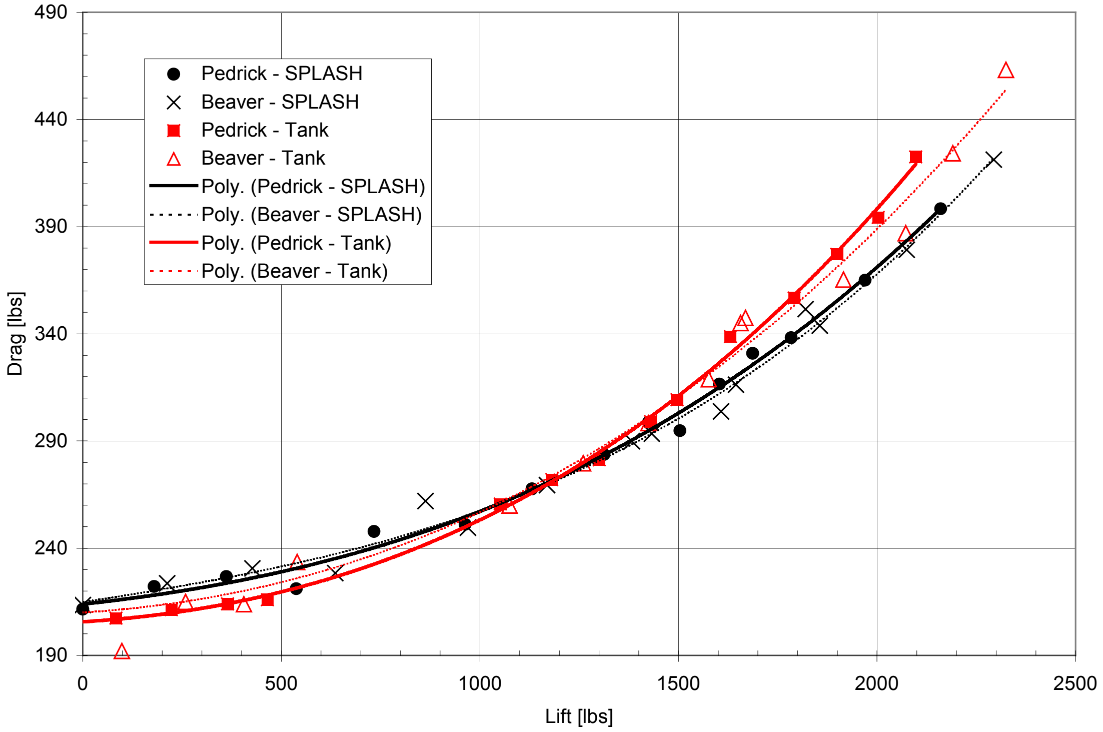
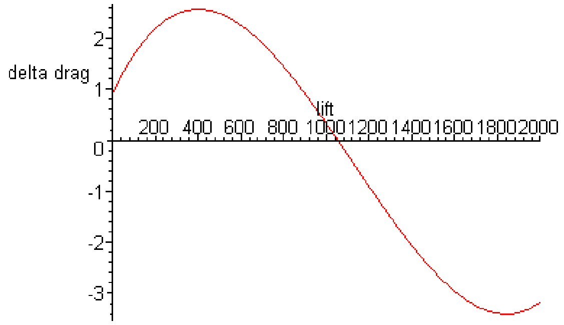


Figure 62 – Lift versus drag comparing the Pedrick and Beaver rudders

This data was applied numerically through the cubic regression. Two new curves (Figures (63), (64)) were made of the drag of the Pedrick rudder subtracted from the drag of the Beaver rudder over a range of lift for both experimental methods.



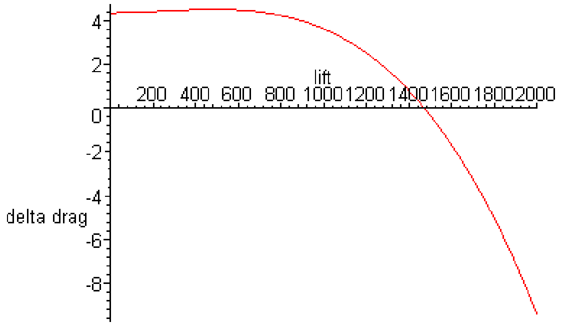


Figure 63 – Drag delta of Beaver to Pedrick in SPLASH

Figure 64 – Drag delta of Beaver to Pedrick in tank

Tank testing predicted a greater advantage of drag in low lift conditions for the Pedrick rudder than in SPLASH. However, some error in the low lift conditions of the Beaver rudder were evident from the data. Furthermore, the inherent scale effect errors described earlier are amplified when using small appendages leading to the conclusion that SPLASH produces more reliable lift and drag predictions especially for small angles of attack. Both methods still predicted a cross-over point where the Beaver rudder gains advantage over the Pedrick rudder in the low lift condition.

In plotting Yaw-moment versus Drag, the information desired was mainly how the rudder angle affects the yawing moment. Therefore, the effect of rudder on the boat for an averaged yaw was quadratic.

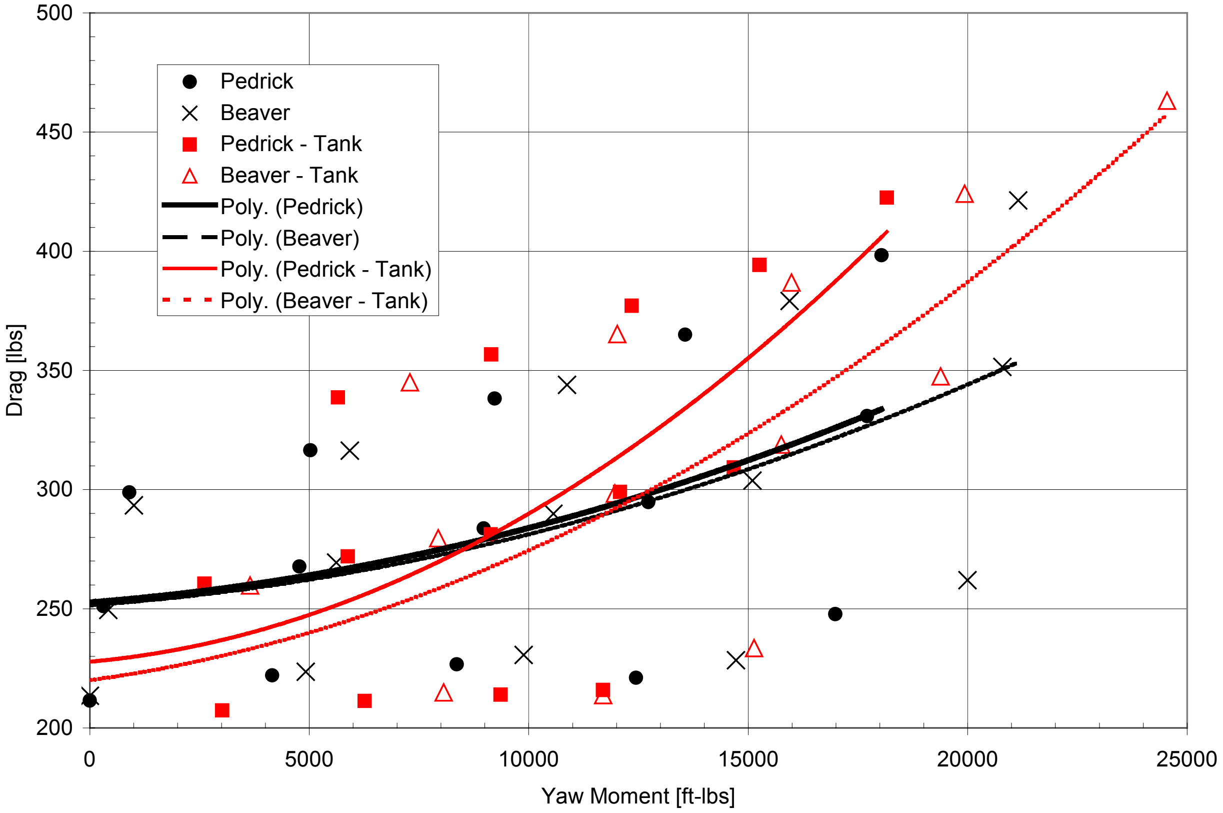


Figure 65 – Yaw-moment versus drag comparing the Pedrick and Beaver rudders

Figure (65) shows that the Beaver rudder had less drag for the same amount of yaw moment. SPLASH and tank testing showed an increase of the Beaver rudder's advantage for high angles of attack of the rudder. Therefore, the Beaver rudder would have reduced drag for a given yaw-moment against the Pedrick rudder, leading to faster maneuvers. Furthermore, upwind sailing would be faster with the Beaver rudder since the yaw moment of the rudder is applied against the aerodynamic yaw-moment. This would allow for more power to be produced by the mainsail, leading to more aerodynamic drive, and therefore more boat speed. However, a note of caution is that all rudder tests took place at a position of zero degrees heel. To fully quantify the sailing performance of the Beaver rudder a full sailing spectrum of heel and yaw in SPLASH should take place.

Planform Comparison

As presented in the tank comparison section, tank testing, while providing general trends in appendage performance, does not accurately predict minute changes for small models. Therefore, SPLASH was used exclusively to determine the general characteristics of different planforms of rudders. Furthermore, appendage testing in CFD required roughly one half hour to build each rudder in a computer-aided design program, and forty-five minutes to complete the testing matrix in SPLASH. For model design, at least two days were required to build an appendage followed by four to five hours of tank testing.

The object of the planform comparison was to ascertain different qualities of the shape of each rudder. The comparison was of the effectiveness of a rudder type, not just the best rudder for a certain boat. Therefore, a desired lift or yaw moment was not a necessary test. The rudder size differences were negligible as long as they were all within limits of not creating excess wavemaking drag. Instead of lift, drag, or yaw moment, nondimensionalized coefficients of the respective terms were used. This allowed for the evaluation of different sizes easily.

Figure (66) shows the results of the efficiency of a rudder with respect to lift. SPLASH was able to differentiate between all of the shapes even though the average performance of each rudder (shown by the cubic fit) were almost identical.

Figure (67) shows the results of the efficiency of a rudder with respect to yaw moment. There was more visible difference in performance with respect to turning the boat.

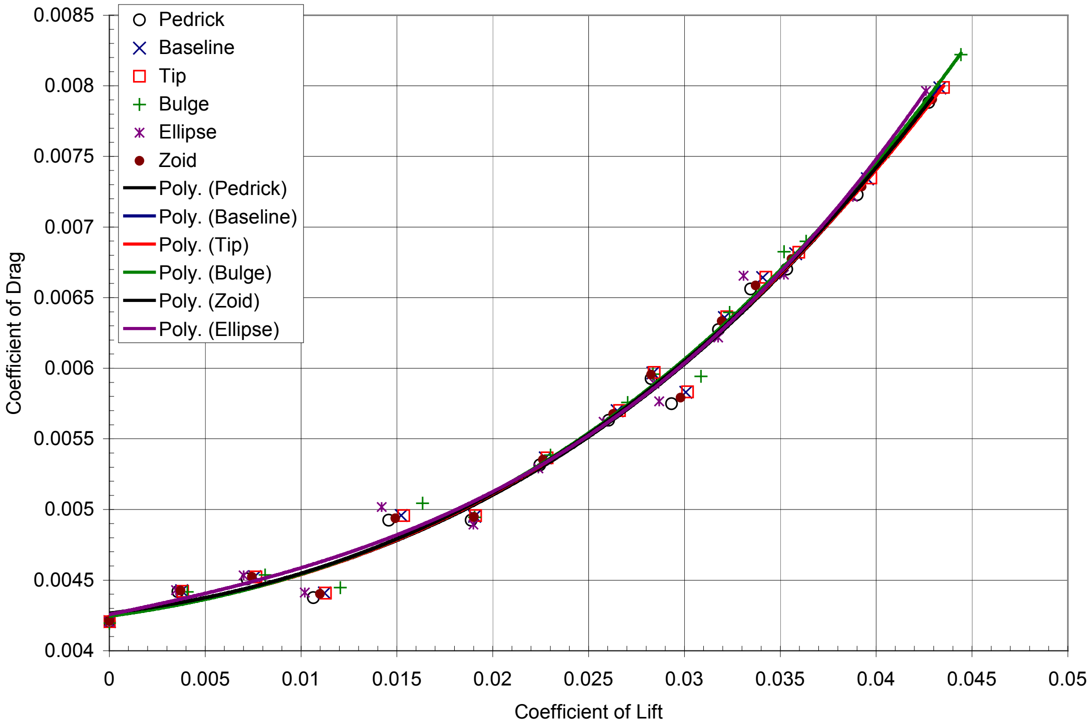


Figure 66 – Lift versus drag for different planforms

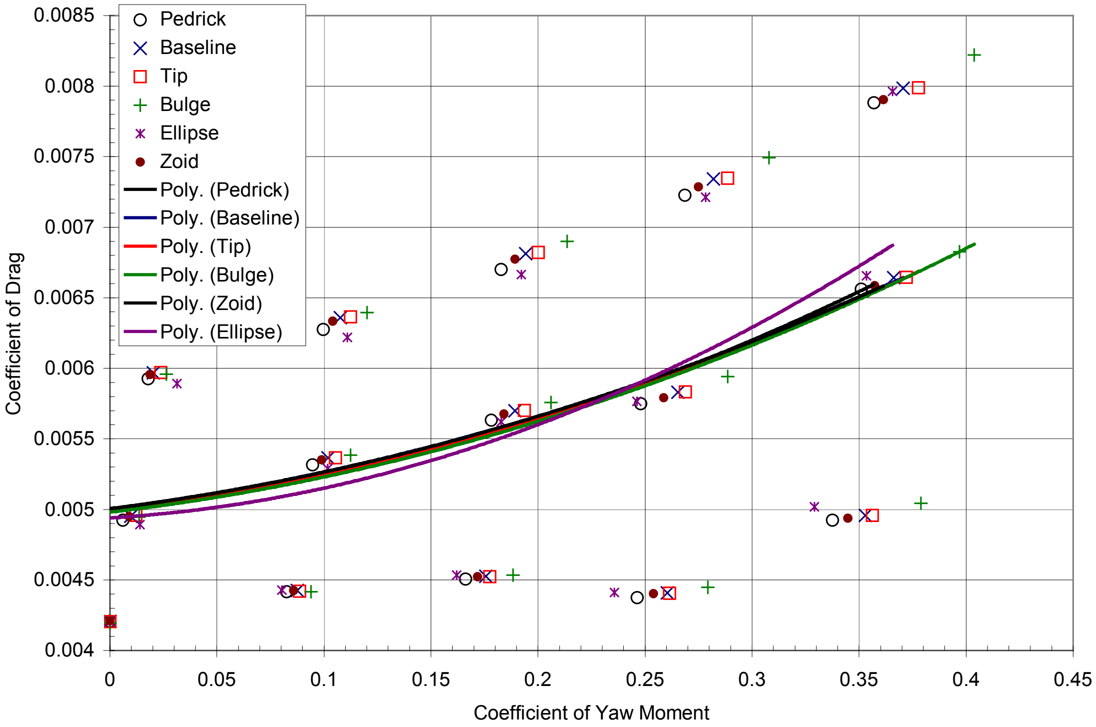


Figure 67 – Yaw moment versus drag for different planforms

Improved resolution was found for both Figures (66) and (67) by creating plots of the difference in drag coefficient between the Baseline rudder and the other rudders.

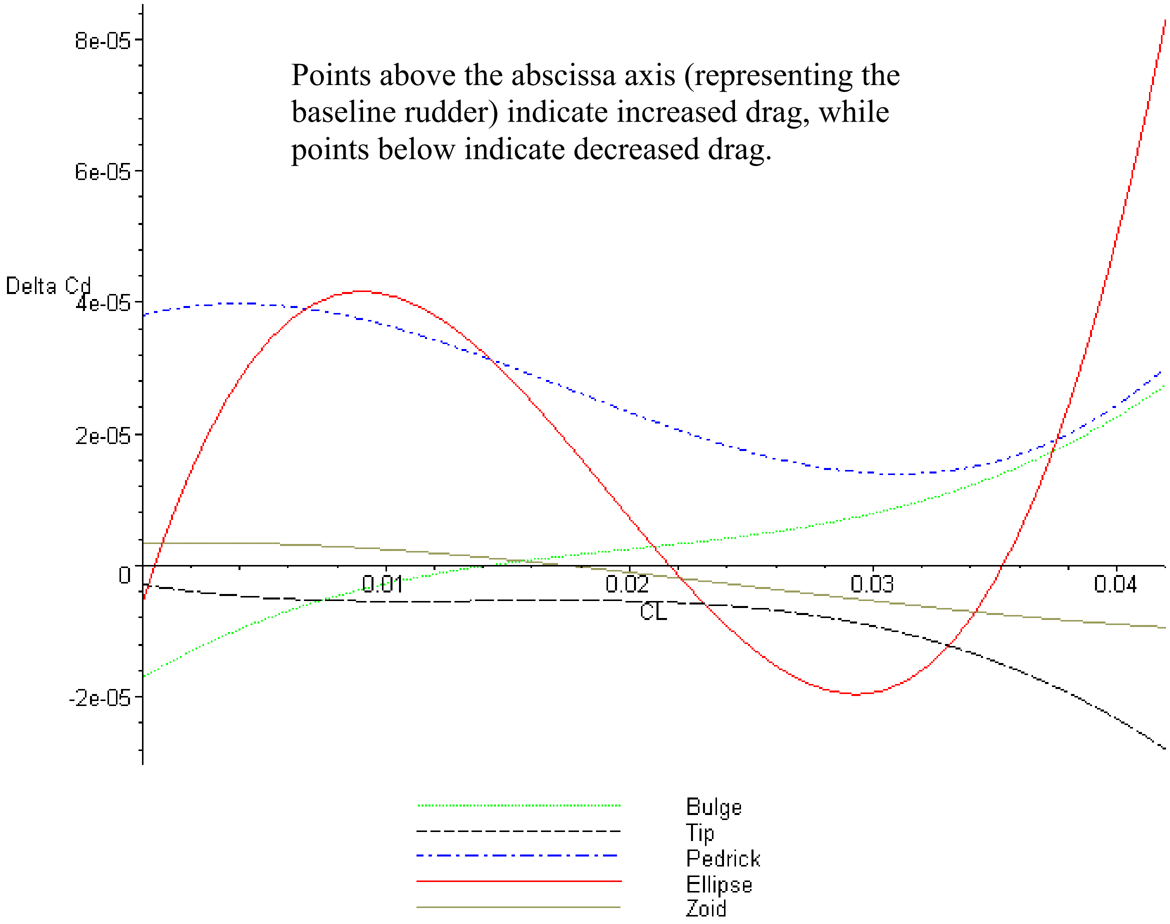


Figure 68 –Drag delta with respect to lift of planforms to the Baseline rudder

Figure (68) indicated that adding a tip to the trailing edge of the rudder decreased the drag. The tested geometry of the Pedrick rudder, however, clearly indicated decreased performance from the Baseline rudder. The Bulge rudder showed advantage for small amounts of lift while the opposite was true for the Zoid rudder. From these results, sailing performance would be improved downwind with the Bulge rudder where little lift is needed, but would be improved upwind using the trapezoidal planform in high-lift conditions. No general trend could be made of the elliptical rudder, except that in only a very limited window was it better than the Baseline rudder. The Zoid rudder was theorized to have the same characteristics of the elliptical rudder. However, the elliptical rudder showed a very wide range of performance while the Zoid rudder stayed consistently close to the baseline.

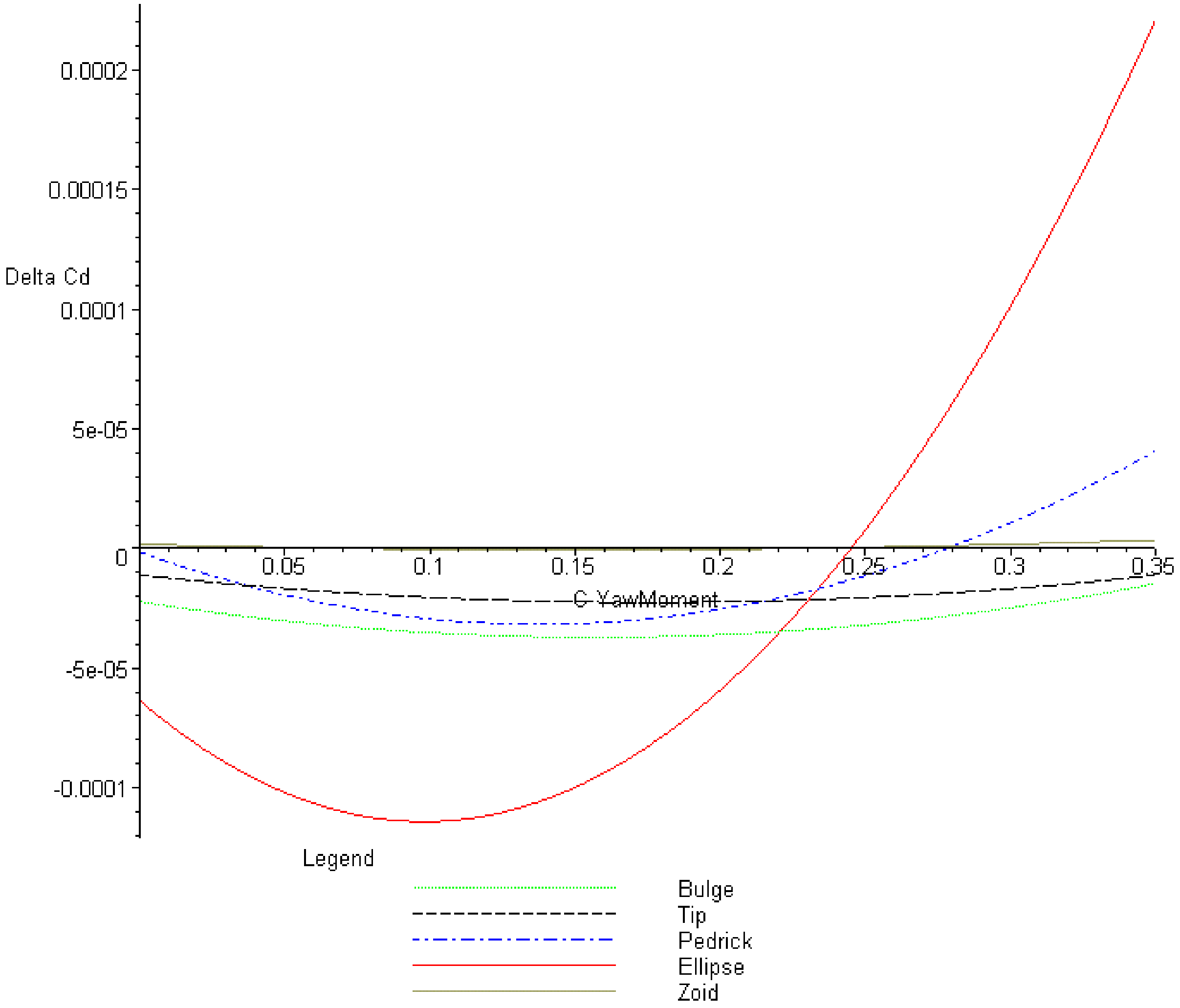


Figure 69 – Drag delta with respect to yaw moment of planforms to the Baseline rudder

Barely visible along the ordinate axis in Figure (69) was the Zoid rudder, indicating that this rudder was almost identical to the Baseline rudder with respect to the efficiency in yawing the boat. Of increasing efficiency in yawing the boat from the Baseline there were two rudders. The Tip rudder was the next most efficient followed by the Bulge rudder. This shows that increasing the longitudinal distribution of surface area increases turning performance. The Pedrick rudder showed more efficiency in turning the boat than the Baseline rudder for small amounts of yaw moment (corresponding to small rudder angles). However, as rudder angle increased, the Pedrick rudder lost efficiency when compared to the Baseline. Finally, the Elliptical rudder showed a large increase in efficiency very early, but also lost its efficiency as rudder angle increased, much like the Pedrick rudder. The yaw moment calculations for the Elliptical rudder and the Pedrick rudder indicate an error in their programmed geometries because drag rose considerably for both rudders at high yaw moments. In either case, high yaw moments will accentuate flow patterns around the leading edge of the rudder and in the induced drag vortex formed at the tip. However, a numerical integration of forces does not provide a method for analyzing the actual cause of the problem.

Explanations for the differences in performance could only be found in plots of the actual flow patterns across each rudder. For the maximum lift condition (6 degrees yaw and 8 degrees rudder), where the differences between rudders was the most extreme, SPLASH plots were constructed of both the back and face surfaces of each rudder. The face surface is the side of the rudder which contains the static pressure point and typically has positive pressure on all parts. The back surface is the side of the rudder which generates most of the suction on the rudder and contains mostly negative pressures. For each plot, the pressure coefficient as well as velocity vectors were placed on top of the rudder. A yellow color indicates zero pressure, while red indicates positive pressure. Green indicates some negative pressure while blue indicates a large amount of negative pressure.

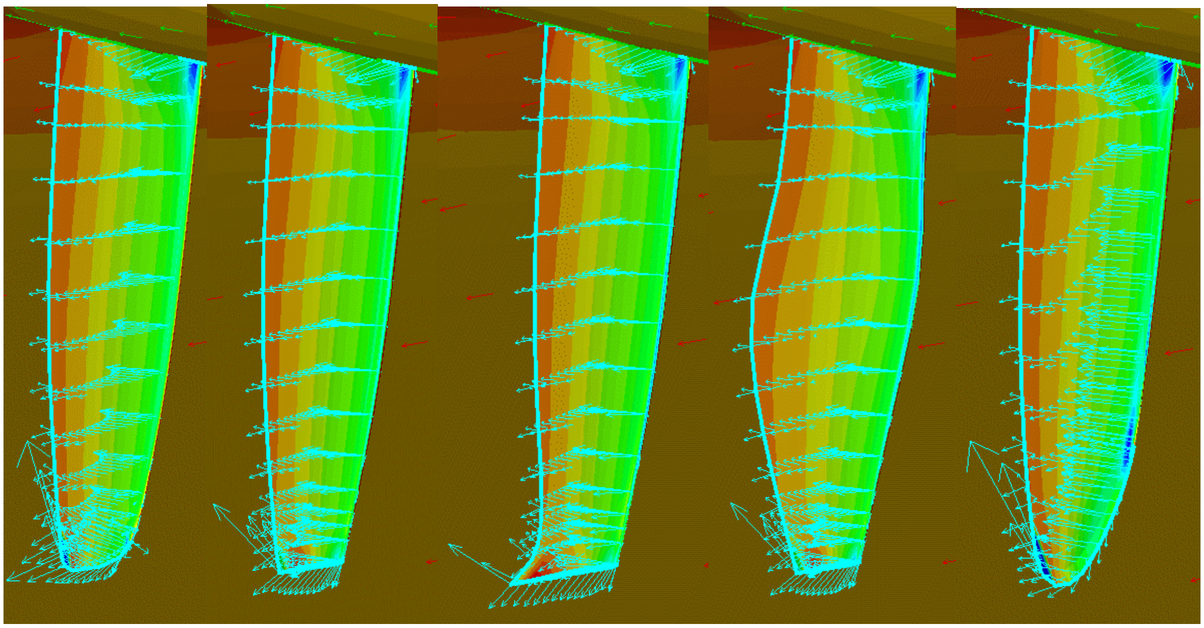


Figure 70 – Back surface of Pedrick, Baseline, Tip, Bulge, and Elliptical Rudders (direction of flow is left)

The vector field helped to indicate flow variations between the rudders. All rudders showed signs of induced drag vortices at the tip. However, the Tip rudder had a much smaller vector which was also at a distance from the rest of the lifting surface. In the same manner, the Bulge rudder had the main portion of its lifting surface positioned at a distance away from the vortex.

However, the Pedrick rudder showed the vortex vector moving almost vertically up the span. The Elliptical rudder showed the same circumstance. The effect of having a vortex at a distance away from the section increased the local coefficient of lift. Furthermore, the vector fields show a flow reversal at the tip of the Pedrick rudder. Because of the flow reversal, the tip geometry of the Pedrick rudder was seen to have an adverse affect on the efficiency of the rudder by causing a much stronger induced drag vortex. A closer look at the coded grid geometry of the Pedrick rudder showed discontinuous surface gradients at the tip, which proved to be the error.

The contour plot of the coefficient of pressure also indicated the effect of the flow variations. All rudders showed large low pressure near the leading edge at the root. The large low pressure signaled separated flow and indicated that the flow across the rudder which was next to the hull suffered from a negative pressure gradient. However, both the Elliptical rudder and the Pedrick rudder showed separation at the trailing edge near the tip. This loss of pressure was also due to a negative pressure gradient as a function of the section curve at the tip. As a design consideration, this negative pressure gradient should not occur at the tip of the rudder and was due to improper geometry. Interestingly, the tip of the Tip rudder showed very high pressure as a result of the strong vortices which it extended. Finally, along the leading edge of the Elliptical rudder, separation was seen at ¾ of the span from the root. This separation indicated that the leading edge of the rudder at this point was too thick to allow for a normal, favorable pressure gradient.

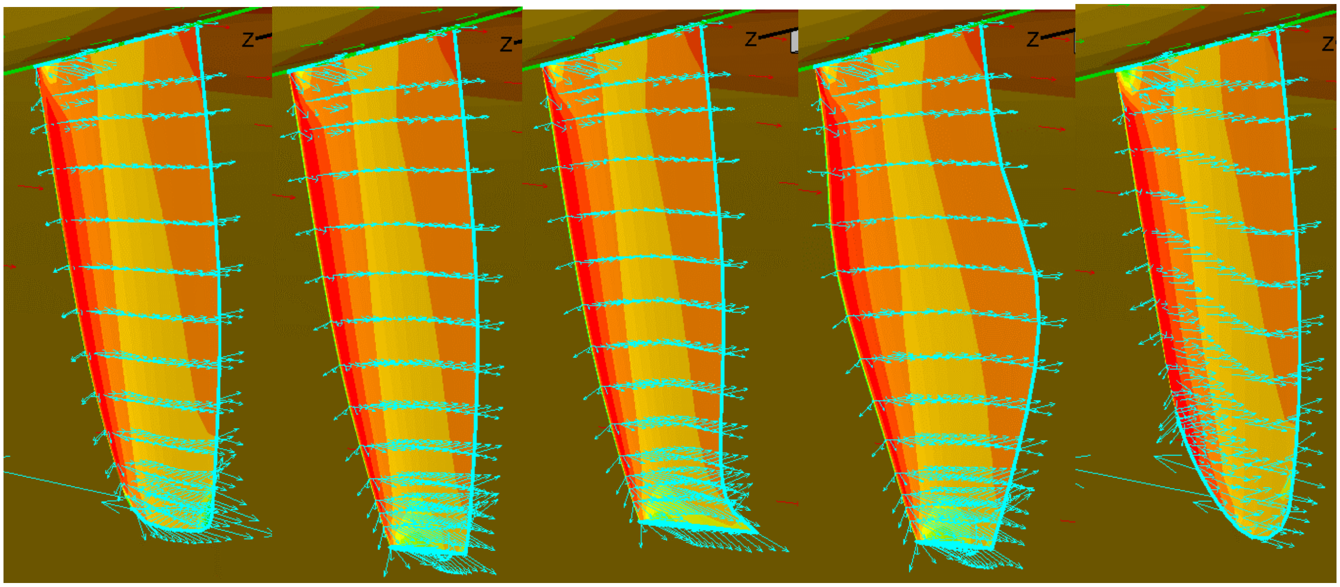


Figure 71 – Face surface of Pedrick, Baseline, Tip, Bulge, and Elliptical Rudders (direction of flow is right)

The face surface had limited use in qualifying the flow effects across the rudder since the flow across the face should not suffer from separated flow or any other negative flow characteristics. However, the Elliptical rudder showed vortices which had an unusual bend across the face. This too indicated the poor construction of the tip geometry of the Elliptical rudder.

Location and Depth Comparison

The location and depth comparison experimented with moving and extending the rudder to different positions. The Baseline rudder was again used to standardize the results and typical lift and yaw moment plots were constructed.

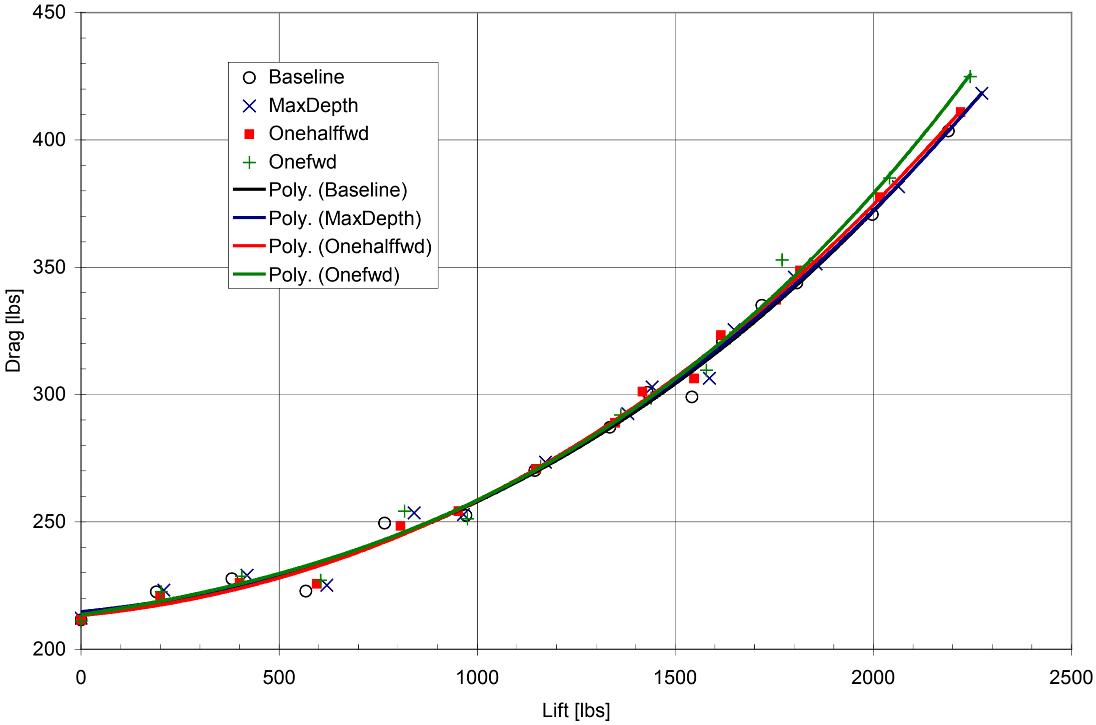


Figure 72 – Lift versus drag for different rudder locations

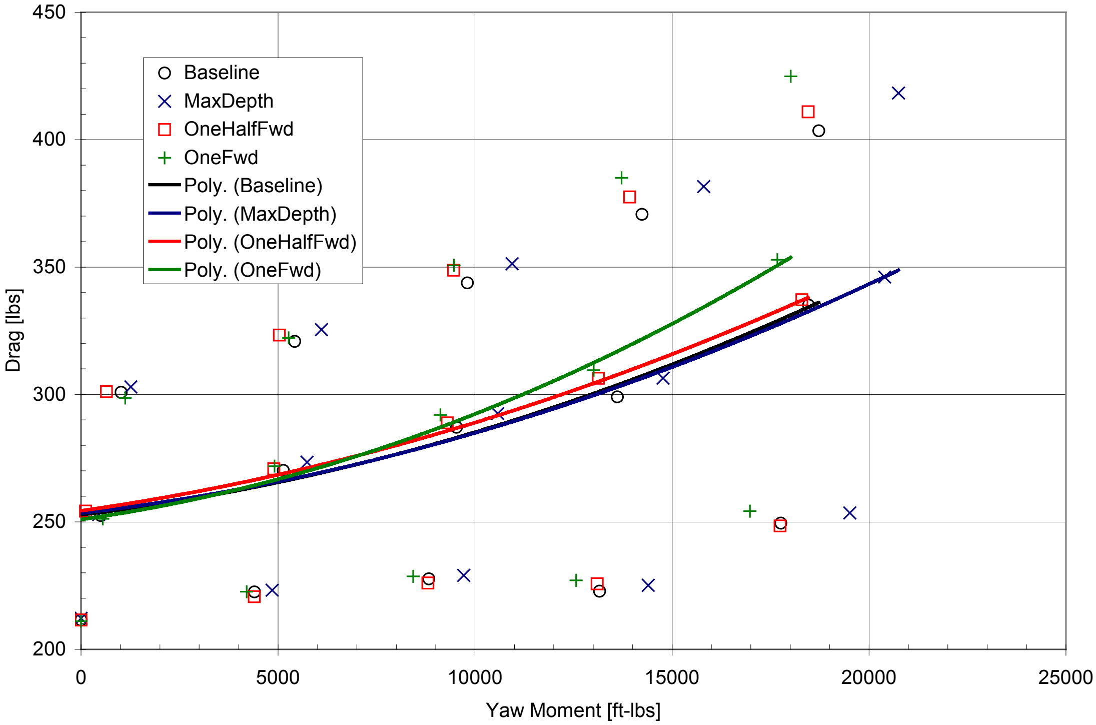


Figure 73 – Yaw moment versus drag for different rudder locations

The lift plot resolution was improved to show interesting results in Figure (72).

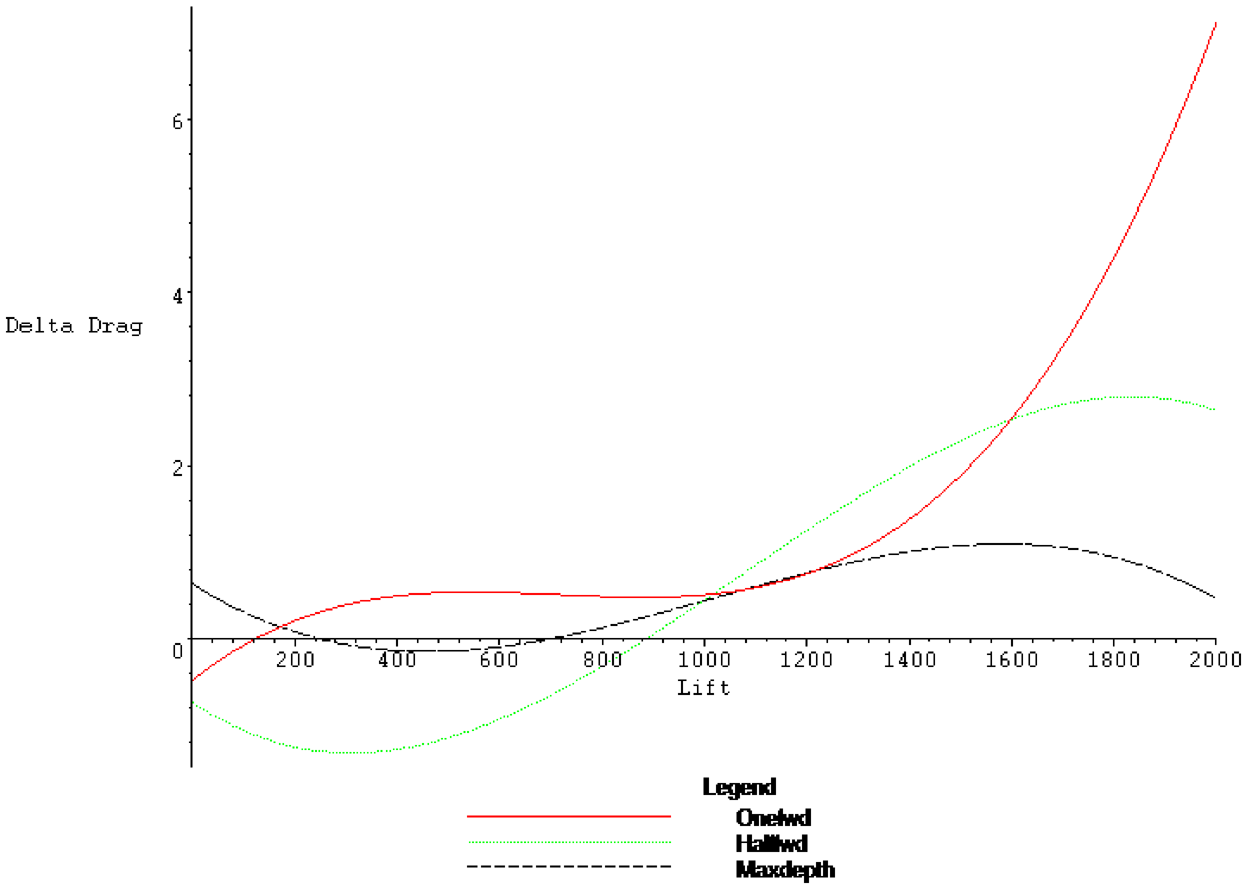


Figure 74 –Drag delta with respect to lift of different locations to the Baseline

Figure (74) shows that the Baseline rudder had good placement compared to the Onefwd, Halffwd, and Maxdepth rudders. The Onefwd as well as the Maxdepth rudders clearly increased drag. The Halffwd rudder showed improvement on the Baseline rudder for small amounts of lift, but became less efficient for larger lift forces.

All of the rudders suffered from flow patterns originating at the keel. As the keel produced more lift, larger vortices formed and disrupted the flow past the rudder. The forward moving rudders were most affected at these high lift forces. The Maxdepth rudder suffered from this effect as well, as the rudder tip was deep enough to by influenced by the keel’s vortex. Furthermore, the Maxdepth rudder did not show any advantages to maximizing the rudder’s aspect ratio. The additional wetted surface area also increased drag and would hinder performance downwind.

Finally, the Halffwd rudder showed a noticeable improvement in performance at conditions of low lift. At these positions, the Halffwd rudder was more submerged than the Baseline rudder which resulted in increased static pressure on the rudder as well as less wavemaking drag. However, these advantages only lasted to a point where the keel vortex gained enough strength to disrupt the flow over the Halffwd rudder.

Without need for increased resolution, Figure (73) shows the efficiency of each rudder in regards to yawing the boat. Because of a decreased moment arm, both the Onefwd and Halffwd rudders had increased drag in turning the boat. However, the Maxdepth rudder was just slightly more efficient than the Baseline rudder due to its larger and more submerged area.

Rudder Redesign Conclusions

There were three advantages in using SPLASH for the rudder redesign versus tank testing. First, SPLASH combined with the FASTSHIP CAD program took less than two hours to complete each comparison whereas tank testing took two days. Secondly, SPLASH did not suffer from scale effects as did tank testing with a small rudder. Finally, the visualization of the flow over the rudder helped to analyze each design.

The planform tests showed that two very important elements of rudder design evaluation are having a quality construction of the geometry, and secondly, locating the rudder vortex away from the main lift generating surface. Having a flattened tip of the Baseline could be considered an advantage from the rounded tip of the Pedrick rudder. However, the Pedrick rudder's geometry had minor programming problems, and a reliable comparison of its tip shape cannot be made. The location and depth test showed that moving the rudder away from the keel vortex also improves efficiency.

In regards to the Mk II Navy 44's rudder, the research produced two recommendations. First, the rudder should be designed to minimize the induced-drag vortices' interference with the lifting surfaces of the rudder. Secondly, the rudder should remain at the same position, and its depth should not be increased.

Therefore, the design of the rudder for the Mk II Navy 44 STC should be either similar to the Tip or Bulge rudder, or even a combination of the two. These rudders provided the most efficiency in regards to both lift and yawing moment. Full-scale testing using different rudders however would be the only method of verifying these results.

RECOMMENDED PERFORMANCE-BASED DESIGN APPROACH

From this study of performance prediction methods, a refined design process was created for the performance-based design of sailboats. In a full ship design, the performance prediction phase would only be one step in the complete design spiral. Using the tools of tow tank testing, a simple parametric VPP (PCSail), a simple CFD code (FKS), and a complex CFD code (SPLASH), an accurate method of predicting performance can be made through iteration in the design.

1. From the design requirements, obtain the initial principal dimensions of the hull, sail plan, and center of gravity.
2. Use PCSail to find the target boat speed and hull form coefficients.
3. Develop preliminary hull lines from the hull form coefficients.
4. Run upright conditions in FKS for a range of Froude numbers from 0.05 to 0.7 to get the effective horsepower (EHP). From the EHP, make an initial engine selection. Also, determine the tow tank drag and lift force block sizes from force estimates. Refine the hull lines based on the predicted wave profile from FKS.
5. Construct a scale model with a basic keel and rudder.
6. Run upright tank tests with and without turbulence stimulators for the likely speeds predicted by PCSail, emphasizing speeds corresponding to a model Reynolds number of less than 1,000,000.
7. Determine the form factor using the sand strip correction factor.
8. Run tank tests at 6, 16, and 24 knots of true wind speed at true wind angles of 40, 90, and 120 degrees using heel and speed estimates from PCSail. Include yaw angle tests of 0 and 5 degrees and rudder angles of 0, 3, and 6 degrees. Record drag, lift, and yaw moment.
9. Standardize SPLASH's trimming moment estimations with the tank testing data.
10. Run SPLASH sailing predictions until the hull lines provide the desired performance
    A. Utilize SPLASH for the full sailing matrix at wind speeds of 6, 12, 18, and 24 knots and wind angles of 40, 60, 90, 120, and 170 degrees. Include yaw angles of 0, 3, and 5 degrees and rudder angles of 0, 3, and 6 degrees.

    B. Export updated lines to a hydrostatic program to obtain the righting moment.

C. Set the SPLASH data into the custom-VPP and run the full polar prediction

11. Evaluate high pressure and large vortex regions on the appendages in SPLASH and create modified appendages.

12. Run SPLASH appendage predictions until satisfied the upwind and downwind performance.

    A. Utilize SPLASH for testing upwind and downwind conditions at yaw angles of 0, 3, and 5 degrees with rudder angles of 0, 3, and 6 degrees.

    B. Run VPP predictions from updated SPLASH appendage data.

    C. Reevaluate the high pressure and large vortex regions on the appendages. Modify the appendages.

13. Obtain the final EHP and final engine selection from the SPLASH upright predictions.

14. Run VPP predictions from finalized SPLASH sailing data until satisfied by the predicted aerodynamic performance.

    A. Modify the Mainsail area, Jib Area, and Spinnaker areas to maximize boat speed.

    B. Assemble matrix of mast rake and trim predictions.

15. Provide the resistance data, rig and trim predictions, and righting moment curve to a sailmaker for evaluation in an aerodynamic CFD program.

16. Generate the final hull lines and assemble the predicted aerodynamic forces for the structural and construction design phase.

By using this performance-based design process in the design spiral, the predicted performance of the boat is maximized by varying the hull, the appendages, and the sails. Through utilizing the strengths of each component of the performance design process, the predicted error is minimized. Furthermore, because the experimental elements of the process have been shown to correlate well to other prediction methods, validation tests of the components are not required.

Ultimately, while performance prediction remains a process of extrapolation to the full-scale, the methods used in prediction will continue to change as theories improve and technology evolves.

REFERENCES

1. Bertin, John J. Engineering Fluid Mechanics. Englewood Cliffs, NJ: Prentice-Hall, 1984.

2. Claughton, Andrew. “Developents in the IMS VPP Formulations.” 14th Chesapeake Sailing Yacht Symposium (1999): 1.

3. Fox, Robert and Alan McDonald. Introduction to Fluid Mechanics. New York, NY: John Wiley & Sons, 1973.

4. Franzini, Joseph B. and E. John Finnemore. "Fluid Mechanics with Engineering Applications, 9th Ed." Boston: WCB/McGraw-Hill, 1997.

5. Larsson, Lars and Rolf E. Eliasson. Principles of Yacht Design. 2nd Ed. Camden, ME: International Marine, 2000.

6. Lloyed, A R J M. Seakeeping: Ship Behaviour in Rough Weather. Gosport, Hampshire, UK: A R J M Lloyd, 1998.

7. Marchaj, C. A. Aero-hydrodynamics of Sailing. Camden, ME: International Marine, 1988.

8. Marchaj, C. A. Sail Performance: Techniques to Maximize Sail Power. Camden, ME: International Marine, 1990.

9. Martin, David. "PCSAIL, A Velocity Prediction Program for a Home Computer." 15th Chesapeake Sailing Yacht Symposium (2001): 100.

10. Newman, J. N. Marine Hydrodynamics. Cambridge, MA: MIT Press, 1986.

11. Percival, Scott, Dane Hendrix, and Francis Noblesse. “Hydrodynamic Optimization of Ship Hull Forms.” Applied Ocean Research. 2001. Vol. 23: 337.

12. Rosen, Bruce, Joseph Laiosa, and Warren Davis. "CFD Design Studies for America's Cup 2000." 18th Applied Aerodynamics Conference. 2000: 2.

13. Tupper, Eric. "Introduction to Naval Architecture." Jersey City, NJ: The Society of Naval Architects and Marine Engineers, 2000.

14. Van Manen, J.D. and P. Van Oossanen. "Resistance." Principles of Naval Architecture Ed. Edward V. Lewis Jersey City, NJ: Society of Naval Architects and Marine Engineers, 1988.

APPENDIX A – NAVY 44S PERFORMANCE PREDICTION COMPARISON

Comparison to other boats is an important part of performance prediction. Since the Mk II Navy 44 is a replacement of the Mk I Navy 44, the comparison of the two designs is very relevant.

The basic performance prediction measure for any vessel is the upright resistance results. The Mk I Navy 44 was tested using the same procedure, setup and analysis as the Mk II.

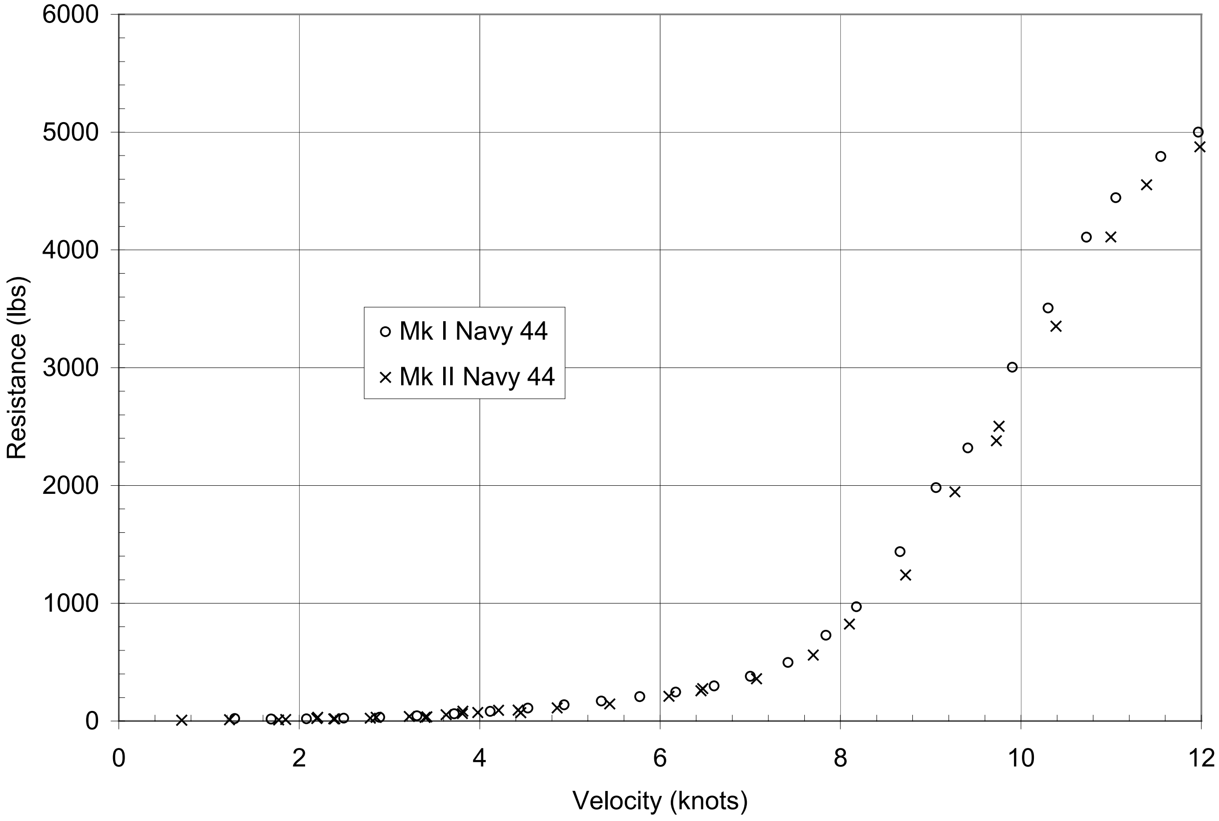


Figure A1 –Upright testing of Mk I Navy 44 and Mk II Navy 44

Figure (A1) shows that the Mk II Navy 44 has reduced resistance for speeds over seven knots. This improvement is due mostly to the improved stern shape of the Mk II Navy 44. The waterline near the stern of the Mk II Navy 44 is beamier than the Mk I Navy 44, and does not include a bustle. At these higher velocities, boats suffer from an effect called “squatting.” The squatting of a boat is the stern sinking into the water, making it more difficult to enter the semi-planning condition of high speeds. When the waterplane area of the stern is increased, the effect of squatting is reduced, and the boat can more easily increase its speed.

Sailing performance is the second measure of performance prediction. Although tank testing sailing tests were completed with the Mk I Navy 44, the data was not populated sufficiently to make a full prediction using the customized VPP. Therefore, the IMS VPP prediction was used exclusively to compare the sailing performance of the two designs.

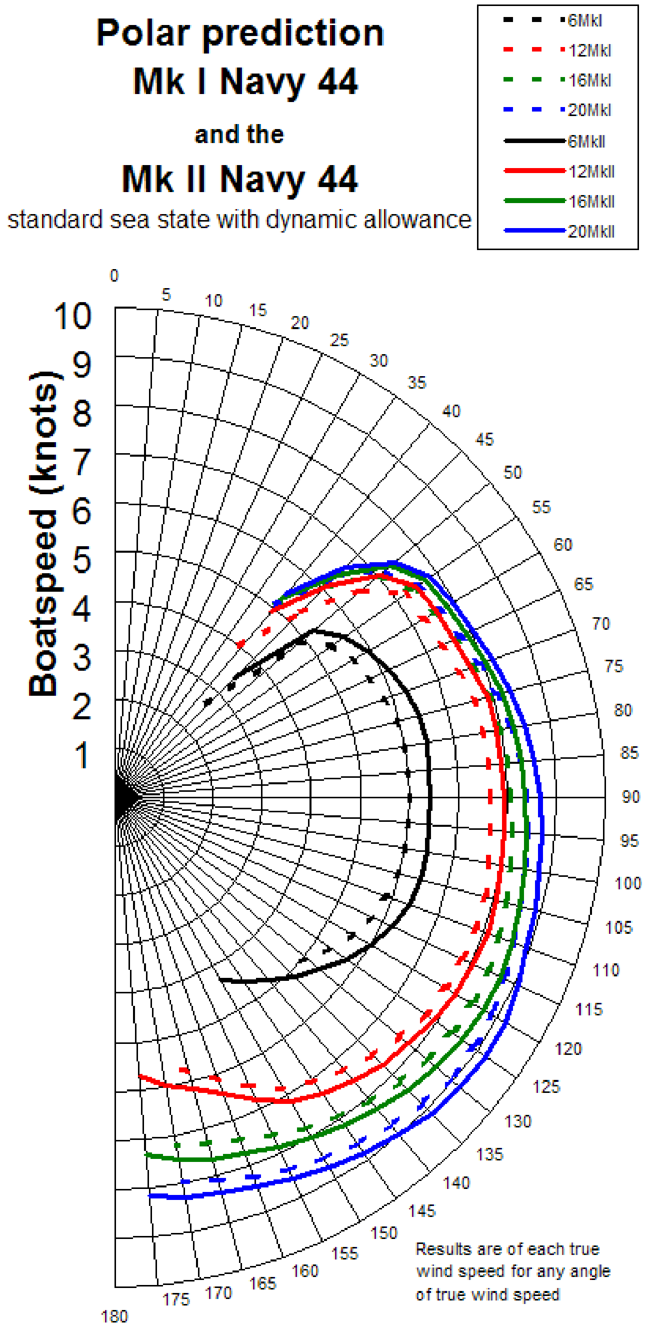


Figure A2 –Polars of Mk I Navy 44 and Mk II Navy 44

Figure (A2) shows the IMS VPP Polars of the predicted speeds of the Navy 44s. Unlike the earlier polars these include the added resistance due to waves for the appropriate sea states. The polars showed two important design changes between the boats.

First, the Mk II Navy 44 showed an average speed improvement of 0.3 knots. Not only was this a result of improved stern waterplane area, but was also a function of a longer waterline and a smoothed underwater profile. The longer waterline of the Mk II Navy 44 acted to decrease the Froude number of the hull, which decreased the wave making resistance. The smoothed underwater profile improved the flow near the stern by reducing flow separation.

Finally, the upwind sailing angle of the Mk II Navy 44 was improved by almost 3 degrees. This increased the ability of the Mk II Navy 44 to sail quickly around a racecourse or to reach a destination when sailing into the wind at sea. The VPP did not include rudder design characteristics however. This would favor the newer design as an earlier midshipman project identified the Mk I rudder as having significantly more drag than the newer design.[15] From these studies, it appears that the newer design should perform better in most if not all conditions.

[15] Ref: Miller, Paul H. "Student Research Projects for the New Navy 44 Sail Training Craft." 16th Chesapeake Sailing Yacht Symposium. (2003): 135.

APPENDIX B – FKS CODES

Grid Setups

The first scripts generated the code to be run in gridgen. The first code, “mkglf.exec” generated another code to be run in gridgen. An input of heel, yaw, rudder, and pitch would create the necessary gridgen script. A file named “mkglf.data” was called in this file. This is a TCL/CL script previously generated which would be added to the script at runtime. An example of the output of this script is the last code included in this section.

```
==> mkglf.exec <==
#!/bin/sh
name=H$1Y$2R$3.glf
namegrd=H$1Y$2R$3.grd

echo $name
touch $name
echo "# Gridgen Journal File V1 (Gridgen 14.02 REL 1)" > $name
echo "gg::varSet name $namegrd" >> $name
echo "gg::varSet heelalpha -$1" >> $name
echo "gg::varSet yawalpha $2" >> $name
echo "gg::varSet rudderalpha -$3" >> $name
echo "gg::varSet pitchalpha 0" >> $name
cat $name mkglf.data > mkglf.tmp
mv mkglf.tmp $name
```

The second script is a setup of the sailing condition which were run in FKS. This script uses the previous “mkglf.exec” script to generate the files for use in Gridgen.

```
==> sail2.sh <==
#!/bin/sh
./mkglf.exec      0     0     0     0
./mkglf.exec      2.5   0     0     0
./mkglf.exec      5     0     0     0
./mkglf.exec      7.5   0     0     0
./mkglf.exec      10    0     0     0
./mkglf.exec      12.5  0     0     0
./mkglf.exec      15    0     0     0
./mkglf.exec      17.5  0     0     0
./mkglf.exec      20    0     0     0
./mkglf.exec      22.5  0     0     0
./mkglf.exec      25    0     0     0
./mkglf.exec      27.5  0     0     0
./mkglf.exec      0     0     3     0
./mkglf.exec      2.5   0     3     0
./mkglf.exec      5     0     3     0
./mkglf.exec      7.5   0     3     0
./mkglf.exec      10    0     3     0
./mkglf.exec      12.5  0     3     0
./mkglf.exec      15    0     3     0
./mkglf.exec      17.5  0     3     0
```

```
./mkglf.exec      20    0     3     0
./mkglf.exec      22.5  0     3     0
./mkglf.exec      25    0     3     0
./mkglf.exec      27.5  0     3     0
./mkglf.exec      0     0     6     0
./mkglf.exec      2.5   0     6     0
./mkglf.exec      5     0     6     0
./mkglf.exec      7.5   0     6     0
./mkglf.exec      10    0     6     0
./mkglf.exec      12.5  0     6     0
./mkglf.exec      15    0     6     0
./mkglf.exec      17.5  0     6     0
./mkglf.exec      20    0     6     0
./mkglf.exec      22.5  0     6     0
./mkglf.exec      25    0     6     0
./mkglf.exec      27.5  0     6     0
./mkglf.exec      0     2     0     0
./mkglf.exec      2.5   2     0     0
./mkglf.exec      5     2     0     0
./mkglf.exec      7.5   2     0     0
./mkglf.exec      10    2     0     0
./mkglf.exec      12.5  2     0     0
./mkglf.exec      15    2     0     0
./mkglf.exec      17.5  2     0     0
./mkglf.exec      20    2     0     0
./mkglf.exec      22.5  2     0     0
./mkglf.exec      25    2     0     0
./mkglf.exec      27.5  2     0     0
./mkglf.exec      0     2     3     0
./mkglf.exec      2.5   2     3     0
./mkglf.exec      5     2     3     0
./mkglf.exec      7.5   2     3     0
./mkglf.exec      10    2     3     0
./mkglf.exec      12.5  2     3     0
./mkglf.exec      15    2     3     0
./mkglf.exec      17.5  2     3     0
./mkglf.exec      20    2     3     0
./mkglf.exec      22.5  2     3     0
./mkglf.exec      25    2     3     0
./mkglf.exec      27.5  2     3     0
./mkglf.exec      0     2     6     0
./mkglf.exec      2.5   2     6     0
./mkglf.exec      5     2     6     0
./mkglf.exec      7.5   2     6     0
./mkglf.exec      10    2     6     0
./mkglf.exec      12.5  2     6     0
./mkglf.exec      15    2     6     0
./mkglf.exec      17.5  2     6     0
./mkglf.exec      20    2     6     0
./mkglf.exec      22.5  2     6     0
./mkglf.exec      25    2     6     0
./mkglf.exec      27.5  2     6     0
./mkglf.exec      0     4     0     0
./mkglf.exec      2.5   4     0     0
./mkglf.exec      5     4     0     0
./mkglf.exec      7.5   4     0     0
```

```
./mkglf.exec      10    4     0     0
./mkglf.exec      12.5  4     0     0
./mkglf.exec      15    4     0     0
./mkglf.exec      17.5  4     0     0
./mkglf.exec      20    4     0     0
./mkglf.exec      22.5  4     0     0
./mkglf.exec      25    4     0     0
./mkglf.exec      27.5  4     0     0
./mkglf.exec      0     4     3     0
./mkglf.exec      2.5   4     3     0
./mkglf.exec      5     4     3     0
./mkglf.exec      7.5   4     3     0
./mkglf.exec      10    4     3     0
./mkglf.exec      12.5  4     3     0
./mkglf.exec      15    4     3     0
./mkglf.exec      17.5  4     3     0
./mkglf.exec      20    4     3     0
./mkglf.exec      22.5  4     3     0
./mkglf.exec      25    4     3     0
./mkglf.exec      27.5  4     3     0
./mkglf.exec      0     4     6     0
./mkglf.exec      2.5   4     6     0
./mkglf.exec      5     4     6     0
./mkglf.exec      7.5   4     6     0
./mkglf.exec      10    4     6     0
./mkglf.exec      12.5  4     6     0
./mkglf.exec      15    4     6     0
./mkglf.exec      17.5  4     6     0
./mkglf.exec      20    4     6     0
./mkglf.exec      22.5  4     6     0
./mkglf.exec      25    4     6     0
./mkglf.exec      27.5  4     6     0
```

FKS testing

The first step in running FKS was to create the control files necessary for input. This script generated the required file based on the name of the grid to be used and the Froude number to be set. The script also ran FKS and created the appropriate output files.

```
==> f2.exec <==
#!/bin/bash
rm -f fks.in
rm error.bc
cat << EOF > fks.in
      BASIC INPUT FILE FOR CODE FKS

1) Enter name of input file that defines the ship hull surface
grid.in

2) Enter Froude number
EOF
echo $2 >> fks.in
cat <<EOF >> fks.in
```

```
3) Set kwave = 1 or 0 to compute or ignore wave component
       1

4) Set klocal = 0,1,2,3 or 4 to ignore,compute,inf fluid,double body, or inf
speed local component
       1

5) Set kdrag = 1 or 0 to compute or ignore wave drag
       1

6) Set kext equal to -1 , 0 , 1 , or 2
   kext = -1 : compute flow along ship waterline
   kext = 0  : compute flow at ship hull surface
   kext = 1  : compute flow at free-surface
   kext = 2  : compute flow within flow domain
   0

7) If kext > 0 : enter name of input file defining flow field points
   ( any string of characters is acceptable if kext = 0 or -1 )
EOF
echo wavcutxyz_$1_$2.dat >> fks.in
cat <<EOF >> fks.in

8) Enter nrowref = number of longitudinal rows of patch ref points
    2

9) For i between 1 & nrowref , enter
   yref(i) , zref(i) , xrefmin(i) , xrefmax(i) :
    0.0    ,  0.0    ,   0.0    ,   1
    0.0    , -0.2   ,   0.0   ,   1

10) Enter name of basic output file summarizing main io information
EOF
echo fkssum_$1_$2.dat >> fks.in
cat <<EOF >> fks.in

11) If kdrag = 1 : enter name of output file to plot spectrum function
    ( any string of characters is acceptable if kdrag = 0 )
EOF
echo spec_$1_$2.dat >> fks.in
cat <<EOF >> fks.in

12) If kext > -1 : enter name of output file to PLOT velocity and
                   pressure computed at flow field points
    ( any string of characters is acceptable if kext = -1 )
EOF
echo plotuvw_$1_$2.dat >> fks.in
cat <<EOF >> fks.in

13) If kext = -1 or if kext = 0 with nseg > 0 : enter name of output
    file to PLOT wave profile ( any string of characters is
    acceptable if kext > 0 or if kext = 0 with nseg = 0 )
EOF
echo wavprof_$1_$2.dat >> fks.in
cat <<EOF >> fks.in
```

```
14) If kext = 0 : enter name of output file giving velocity @ centroids
    of hull panels & waterline segments FOR INPUT TO FKNK & FKX
    ( any string of characters is acceptable if kext not equal to 0 )
EOF
echo uvwpan_$1_$2.dat >> fks.in
cat <<EOF >> fks.in

15) Enter name of output file that defines patches/panels/segments
EOF
echo pnseg_$1_$2.dat >> fks.in
fks.exec
rm -f pan*.dat
```

Another script file was made for use only in sailing conditions. For multiple hulls, multiple pre-processings had to be made using the program “g2fks.” This program pre-processed each condition and then ran the condition using the above code, “f2.exec.”

```
==> f.exec <==
#!/bin/bash
g2fks.exec <<EOF
2
$1.grd
0
0
0
0
EOF
hull=$1
speed=$2
./f2.exec $hull $speed
```

The last two codes show examples of how a set of upright tests and sailing tests would be performed.

```
==> upright.sh <==
#!/bin/bash
./f2.exec H0Y0R0 .10
./f2.exec H0Y0R0 .12
./f2.exec H0Y0R0 .14
./f2.exec H0Y0R0 .16
./f2.exec H0Y0R0 .18
./f2.exec H0Y0R0 .20
./f2.exec H0Y0R0 .22
./f2.exec H0Y0R0 .24
./f2.exec H0Y0R0 .26
./f2.exec H0Y0R0 .28
./f2.exec H0Y0R0 .30
./f2.exec H0Y0R0 .32
./f2.exec H0Y0R0 .34
./f2.exec H0Y0R0 .36
```

```
./f2.exec H0Y0R0 .38
./f2.exec H0Y0R0 .40
./f2.exec H0Y0R0 .42

==> sail2run.sh <==
./f.exec H0Y0R0 0.1
./f.exec H2.5Y0R0 0.1
./f.exec H5Y0R0 0.1
./f.exec H7.5Y0R0 0.1
./f.exec H10Y0R0 0.1
./f.exec H12.5Y0R0 0.1
./f.exec H15Y0R0 0.1
./f.exec H2.5Y0R3 0.1
./f.exec H5Y0R3 0.1
./f.exec H7.5Y0R3 0.1
./f.exec H15Y4R3 0.2
./f.exec H17.5Y4R3 0.2
./f.exec H20Y4R3 0.2
./f.exec H22.5Y4R3 0.2
./f.exec H25Y4R3 0.2
./f.exec H27.5Y4R3 0.2
./f.exec H0Y4R6 0.2
./f.exec H2.5Y2R0 0.3
./f.exec H5Y2R0 0.3
./f.exec H7.5Y2R0 0.3
./f.exec H10Y2R0 0.3
./f.exec H12.5Y2R0 0.3
./f.exec H15Y2R0 0.3
./f.exec H0Y4R0 0.3
./f.exec H2.5Y4R0 0.3
./f.exec H5Y4R0 0.3
./f.exec H7.5Y4R0 0.3
./f.exec H10Y4R0 0.3
./f.exec H15Y0R0 0.4
etc
```

Gridgen Script

The last code is the TCL/CL script which was used to import an iges surface of the Navy 44 into Gridgen and get out a nondimensional plot3d grid. This code has already been generated by the script "mkglf.exec"

```
gg::varSet name H0Y0R0.grd
gg::varSet heelalpha -0
gg::varSet yawalpha 0
gg::varSet rudderalpha -0
gg::varSet pitchalpha 0
package require PWI_Glyph 1.4
gg::memClear
gg::aswSet GENERIC -dim 3
gg::defReset
gg::tolReset
gg::updatePolicy DISPLAY_AND_INPUT
```

```
#Get the Mk II Navy 44 database file
gg::dbImport "/home/c03/m036330/dimenNavy44mv.dba" -type DBA
set _DB(53) [gg::dbGetByName -- {Navy44a-F2-2}]

NOTE: use set _DB(#) to all iges surface names.
The rest of the set command have been omitted.

gg::dbTransformBegin [list \
    $_DB(53) \
    $_DB(55) \

NOTE: list all DB names required for transformation.
The rest of the DBs have been omitted.

  ] -maintain_linkage
  gg::xformTranslate ".1 .1 .1"
  gg::xformTranslate "-.1 -.1 -.1"
gg::dbTransformEnd
set _DB(54) [gg::dbGetByName -- {Navy44a-F3-3}]
set _DB(68) [gg::dbGetByName -- {Navy44a-F3-3-copy1}]
set _DB(4) [gg::dbGetByName -- {Navy44a-E9-9}]
set _DB(5) [gg::dbGetByName -- {Navy44a-E10-10}]
set _DB(6) [gg::dbGetByName -- {Navy44a-E11-11}]
set _DB(7) [gg::dbGetByName -- {Navy44a-E12-12}]
set _DB(43) [gg::dbGetByName -- {Navy44a-F3-21}]
set _DB(69) [gg::dbGetByName -- {Navy44a-F3-21-copy1}]
set _DB(70) [gg::dbGetByName -- {Navy44a-E12-12-copy1}]
set _DB(71) [gg::dbGetByName -- {Navy44a-E11-11-copy1}]
set _DB(72) [gg::dbGetByName -- {Navy44a-E10-10-copy1}]
set _DB(73) [gg::dbGetByName -- {Navy44a-E9-9-copy1}]
# apply a rudder angle
set _ggTemp_(1) $rudderalpha
gg::dbTransformBegin [list \
    $_DB(54) \
    $_DB(68) \
    $_DB(4) \
    $_DB(5) \
    $_DB(6) \
    $_DB(7) \
    $_DB(43) \
    $_DB(69) \
    $_DB(70) \
    $_DB(71) \
    $_DB(72) \
    $_DB(73) \
  ] -maintain_linkage
  gg::xformRotate ".908104 0 0" ".908104 0 -10" $_ggTemp_(1)
gg::dbTransformEnd
unset _ggTemp_(1)
# apply a heel angle
set _ggTemp_(1) $heelalpha
gg::dbTransformBegin [list \
    $_DB(53) \
    $_DB(54) \
    $_DB(55) \
NOTE: list all DB names required for transformation.
The rest of the DBs have been omitted.

  ] -maintain_linkage
  gg::xformRotate [list 0 0 0] [list 0.500024353 0 0] $_ggTemp_(1)
gg::dbTransformEnd
unset _ggTemp_(1)
# apply a trim angle
set _ggTemp_(1) $pitchalpha
gg::dbTransformBegin [list \
    $_DB(53) \
    $_DB(54) \
    $_DB(55) \

NOTE: list all DB names required for transformation.
The rest of the DBs have been omitted.

  ] -maintain_linkage
  gg::xformRotate ".5 0 0" ".5 10 0" $_ggTemp_(1)
gg::dbTransformEnd
unset _ggTemp_(1)
# apply a yaw angle
set _ggTemp_(1) $yawalpha
gg::dbTransformBegin [list \
    $_DB(53) \
    $_DB(54) \
    $_DB(55) \

NOTE: list all DB names required for transformation.
The rest of the DBs have been omitted.

  ] -maintain_linkage
  gg::xformRotate ".5 0 0" ".5 0 10" $_ggTemp_(1)
gg::dbTransformEnd
unset _ggTemp_(1)

set _DB(142) [gg::dbPlane -z [lindex "0 0 0" 2]]

# Intersect the body with a z = 0 waterplane
gg::dbIntersect $_DB(142) [list \
    $_DB(53) \
    $_DB(63) \
    $_DB(86) \
    $_DB(103) \
  ] -los [list 0 0 1]
set _DB(144) [gg::dbGetByName -- {Navy44a-F1-211-isect1}]
set _DB(143) [gg::dbGetByName -- {Navy44a-F2-3-isect1}]
gg::conBegin
  gg::segBegin -type DB_CUBIC
    gg::segAddControlPt -db [list 1 0 $_DB(144)]
    gg::segAddControlPt -db [list 0.0031643302 0 $_DB(144)]
    gg::segAddControlPt -db [list 0.1243136332 0 $_DB(143)]
    gg::segAddControlPt -db [list 1 0 $_DB(143)]
  gg::segEnd
set _CN(1) [gg::conEnd]
set _DB(146) [gg::dbGetByName -- {Navy44a-F1-211-copy1-isect1}]
set _DB(145) [gg::dbGetByName -- {Navy44a-F2-3-copy1-isect1}]
gg::conBegin
  gg::segBegin -type DB_CUBIC
    gg::segAddControlPt -db [list 1 0 $_DB(146)]
    gg::segAddControlPt -db [list 0.006846838817 0 $_DB(146)]
    gg::segAddControlPt -db [list 0.1173754558 0 $_DB(145)]
    gg::segAddControlPt -db [list 1 0 $_DB(145)]
  gg::segEnd
set _CN(2) [gg::conEnd]
set _ggTemp_(1) [gg::conGetPt $_CN(1) -db -arc 1]
unset _ggTemp_(1)
set _ggTemp_(1) [gg::conGetPt $_CN(1) -db -arc 1]
unset _ggTemp_(1)
gg::conBegin
  gg::segBegin -type DB_CUBIC
    gg::segAddControlPt -db [list 1 0 $_DB(143)]
    gg::segAddControlPt -db [list .51 0 $_DB(3)]
    gg::segAddControlPt -db [list .60 0 $_DB(3)]
    gg::segAddControlPt -db [list .99 0 $_DB(3)]
    gg::segAddControlPt -db [list 1 0 $_DB(41)]
    gg::segAddControlPt -db [list .7316 0 $_DB(41)]
    gg::segAddControlPt -db [list 0 0 $_DB(36)]
  gg::segEnd
set _CN(3) [gg::conEnd]

gg::conBegin
  gg::segBegin -type DB_CUBIC
    gg::segAddControlPt -db [list 1 0 $_DB(144)]
    gg::segAddControlPt -db [list 0.2865071407 0 $_DB(41)]
    gg::segAddControlPt -db [list 0.4740180149 0 $_DB(41)]
    gg::segAddControlPt -db [list 1 0 $_DB(36)]
  gg::segEnd
set _CN(4) [gg::conEnd]
gg::conOnDBEnt [list \
    $_DB(36) \
    $_DB(82) \
  ]
gg::conDim $_CN(3) 20
set _CN(5) [lindex [gg::conGetAll] 4]
gg::conDim $_CN(5) 25
gg::conDim $_CN(4) 20
gg::conDim $_CN(1) 25
gg::domBegin -type STRUCTURED
  gg::edgeBegin
    gg::edgeAddCon $_CN(3)
  gg::edgeEnd
  gg::edgeBegin
    gg::edgeAddCon $_CN(5)
  gg::edgeEnd
  gg::edgeBegin
    gg::edgeAddCon $_CN(4)
  gg::edgeEnd
  gg::edgeBegin
    gg::edgeAddCon $_CN(1)
  gg::edgeEnd
set _DM(1) [gg::domEnd]
```

```
gg::conDim $_CN(2) 25
set _CN(6) [lindex [gg::conGetAll] 5]
gg::conDim $_CN(6) 25
gg::domBegin -type STRUCTURED
  gg::edgeBegin
    gg::edgeAddCon $_CN(2)
  gg::edgeEnd
  gg::edgeBegin
    gg::edgeAddCon $_CN(4)
  gg::edgeEnd
  gg::edgeBegin
    gg::edgeAddCon $_CN(6)
  gg::edgeEnd
  gg::edgeBegin
    gg::edgeAddCon $_CN(3)
  gg::edgeEnd
set _DM(2) [gg::domEnd]
set _ggTemp_(1) [list $_DM(1) $_DM(2)]
gg::domEllSolverBegin $_ggTemp_(1)
  gg::domTFISolverRun $_ggTemp_(1)
  gg::domEllSolverStep -iterations 4 -nodisplay
gg::domEllSolverEnd
unset _ggTemp_(1)
gg::conOnDBEnt $_DB(35)
gg::conOnDBEnt $_DB(8)
gg::conOnDBEnt $_DB(37)
gg::conOnDBEnt $_DB(83)
gg::conOnDBEnt $_DB(78)
gg::conOnDBEnt $_DB(81)
set _CN(12) [lindex [gg::conGetAll] 11]
gg::conDim $_CN(12) 25
set _CN(11) [lindex [gg::conGetAll] 10]
gg::conDim $_CN(11) 25
set _CN(10) [lindex [gg::conGetAll] 9]
gg::conDim $_CN(10) 25
gg::domBegin -type STRUCTURED
  gg::edgeBegin
    gg::edgeAddCon $_CN(6)
  gg::edgeEnd
  gg::edgeBegin
    gg::edgeAddCon $_CN(12)
  gg::edgeEnd
  gg::edgeBegin
    gg::edgeAddCon $_CN(11)
  gg::edgeEnd
  gg::edgeBegin
    gg::edgeAddCon $_CN(10)
  gg::edgeEnd
set _DM(3) [gg::domEnd]
set _CN(7) [lindex [gg::conGetAll] 6]
gg::conDim $_CN(7) 25
set _CN(8) [lindex [gg::conGetAll] 7]
gg::conDim $_CN(8) 25
set _CN(9) [lindex [gg::conGetAll] 8]
gg::conDim $_CN(9) 25
gg::domBegin -type STRUCTURED
```

```
  gg::edgeBegin
    gg::edgeAddCon $_CN(7)
  gg::edgeEnd
  gg::edgeBegin
    gg::edgeAddCon $_CN(8)
  gg::edgeEnd
  gg::edgeBegin
    gg::edgeAddCon $_CN(9)
  gg::edgeEnd
  gg::edgeBegin
    gg::edgeAddCon $_CN(5)
  gg::edgeEnd
set _DM(4) [gg::domEnd]
gg::dispViewDoms FALSE


gg::dbIntersect $_DB(142) [ list \
    $_DB(54) \
    $_DB(68) \
  ] -los [list 0 0 1]
gg::dispViewCenter [gg::dbUVToXYZ [gg::conGetPt $_CN(1) -db -arc 0]]
set _DB(147) [gg::dbGetByName -- {Navy44a-F3-21-isect1}]
set _DB(148) [gg::dbGetByName -- {Navy44a-F3-21-copy1-isect1}]
gg::conBegin
  gg::segBegin -type DB_CUBIC
    gg::segAddControlPt -db [list 0 0 $_DB(4)]
    gg::segAddControlPt -db [list 0.66 0 $_DB(4)]
    gg::segAddControlPt -db [list 1 0 $_DB(147)]
  gg::segEnd
set _CN(13) [gg::conEnd]
gg::conBegin
  gg::segBegin -type DB_CUBIC
    gg::segAddControlPt -db [list 0 0 $_DB(73)]
    gg::segAddControlPt -db [list 0.66 0 $_DB(73)]
    gg::segAddControlPt -db [list 1 0 $_DB(148)]
  gg::segEnd
set _CN(14) [gg::conEnd]
gg::conOnDBEnt $_DB(70)
gg::conOnDBEnt $_DB(71)
gg::conBegin
  gg::segBegin -type DB_CUBIC
    gg::segAddControlPt -db [list 1 0 $_DB(72)]
    gg::segAddControlPt -db [list 0.1 0 $_DB(72)]
    gg::segAddControlPt -db [list 1 0 $_DB(147)]
  gg::segEnd
set _CN(17) [gg::conEnd]
gg::conOnDBEnt $_DB(7)
gg::conOnDBEnt $_DB(6)
gg::conBegin
  gg::segBegin -type DB_CUBIC
    gg::segAddControlPt -db [list 1 0 $_DB(5)]
    gg::segAddControlPt -db [list 0.1 0 $_DB(5)]
    gg::segAddControlPt -db [list 1 0 $_DB(148)]
  gg::segEnd
set _CN(20) [gg::conEnd]
gg::conDim $_CN(14) 10
```

```
gg::conDim $_CN(20) 15
set _CN(19) [lindex [gg::conGetAll] 15]
gg::conDim $_CN(19) 10
set _CN(18) [lindex [gg::conGetAll] 14]
gg::conDim $_CN(18) 15
gg::domBegin -type STRUCTURED
  gg::edgeBegin
    gg::edgeAddCon $_CN(14)
  gg::edgeEnd
  gg::edgeBegin
    gg::edgeAddCon $_CN(20)
  gg::edgeEnd
  gg::edgeBegin
    gg::edgeAddCon $_CN(19)
  gg::edgeEnd
  gg::edgeBegin
    gg::edgeAddCon $_CN(18)
  gg::edgeEnd
set _DM(5) [gg::domEnd]
set _CN(15) [lindex [gg::conGetAll] 11]
gg::conDim $_CN(15) 15
set _CN(16) [lindex [gg::conGetAll] 12]
gg::conDim $_CN(16) 10
gg::conDim $_CN(17) 15
gg::conDim $_CN(13) 10
gg::domBegin -type STRUCTURED
  gg::edgeBegin
    gg::edgeAddCon $_CN(15)
  gg::edgeEnd
  gg::edgeBegin
    gg::edgeAddCon $_CN(16)
  gg::edgeEnd
  gg::edgeBegin
    gg::edgeAddCon $_CN(17)
  gg::edgeEnd
  gg::edgeBegin
    gg::edgeAddCon $_CN(13)
  gg::edgeEnd
set _DM(6) [gg::domEnd]
gg::dispViewDoms TRUE
gg::dispViewReset
set _ggTemp_(1) $name
set _ggTemp_(2) [file join "~" $_ggTemp_(1)]
unset _ggTemp_(1)
# Output as Plot3d format in local directory
gg::domExport ALL $_ggTemp_(2) \
   -style PLOT3D -form ASCII -precision SINGLE
unset _ggTemp_(2)
```

APPENDIX C - DATA ANALYSIS CODES

MATLAB Script

This MATLAB script was a generic script which would input a data file and tranform a sprarse grid into a dense, square grid. This particular script was used on the tank testing data. Other scripts were modified from this script to analyze FKS data or SPLASH data.

```
==> gridsparsedata.m <==
for yc = 0:1
for rc = 0:2
for wc = 3:5
yaw=yc*4;
rudder=rc*3;
whichone=wc;    %whichone refers to drag, lift or yaw in columns 3,4,or 5
D=load('data.txt');  %data is loaded from the file data.txt
O=load('initzero.txt');
start=1+yaw*5+rudder*40/3;
if whichone==3
      type='Drag';
      AZ=-37.5;
elseif whichone==4
      type='Forward Lift';
      AZ=60;
elseif whichone==5
      type='Aft Lift';
      AZ=-170;
end
%heel is in column 1, which is the y value
%speed is in column 2, which is the x value
x=[O(:,2);D(start:start+19,2)]; y=[O(:,1);D(start:start+19,1)];
z=[O(:,whichone);D(start:start+19,whichone)];
xm = max(x); ym=max(y);
x=x/xm; y=y/ym;
ti = 0:.05:1;
[XI,YI] = meshgrid(ti,ti);
ZI = griddata(x,y,z,XI,YI,'cubic');
YI = YI *ym;
y= y*ym;
XI = XI *xm;
x = x*xm;
%mesh(XI,YI,ZI), hold
%plot3(x,y,z,'o'), hold off
%ylabel('Heel Angle [degrees]'),xlabel('Speed [ft/s]'),zlabel([type '
[lbs]']);
%title([type ' with ' int2str(yaw) ' degrees of yaw and ' int2str(rudder) '
degrees of rudder']);
%view(AZ,45)
% saveas(1,[type '' int2str(yaw) '' int2str(rudder) '.bmp'],'bmp')
% save type XI YI ZI -ASCII -TABS
%true velocity is in column 8
%true heel is in column 9
```

```
w = interp2(XI,YI,ZI,D(start:start+19,8),D(start:start+19,9),'cubic')
%name = [type '' int2str(yaw) '' int2str(rudder) '.interp']
%save(name,'w','-ASCII','-TABS')
if whichone==3
      w1 = w
elseif whichone==4
      w2 = w
      ZIf = ZI
      zf = z
elseif whichone==5
      w3 = w
      ZIa = ZI
      za = z
end
end
ZIn = ZIf + ZIa
zn = zf + za
mesh(XI,YI,ZIn), hold
contour3(XI,YI,ZIn,15)
plot3(x,y,zn,'o'), hold off
ylabel('Heel Angle [degrees]'),xlabel('Speed [ft/s]'),zlabel(['Side Force
[lbs]']);
title(['Side force with ' int2str(yaw) ' degrees of yaw and ' int2str(rudder)
' degrees of rudder']);
view(AZ,45)
saveas(1,['SF' int2str(yaw) '' int2str(rudder) '.bmp'],'bmp')
ww = [w1,w2,w3]
name = ['i' int2str(yaw) '' int2str(rudder) '.interp']
save(name,'ww','-ASCII','-TABS')
end
end
```

Maple 6 Analysis

Below is the script used to develop lift and drag for any yaw angle and rudder angle.

```
> restart;
> L:=(AH5,AH3)-
>(alpha1*AH5+alpha2)*AH3^2+(beta1*AH5+beta2)*AH3+(gamma1*AH5+gam
ma2);
```

$$L := (AH5, AH3) \rightarrow (\alpha 1\ AH5 + \alpha 2)\ AH3^2 + (\beta 1\ AH5 + \beta 2)\ AH3 + \gamma 1\ AH5 + \gamma 2$$

```
> L(0,0)=AG2; assign(%);
```

$$\gamma 2 = AG2$$

```
> L(4,0)=AG3; gamma1=solve(%,gamma1); assign(%);
```

$$4\ \gamma 1 + AG2 = AG3$$

$$\gamma 1 = -\frac{1}{4} AG2 + \frac{1}{4} AG3$$

```
> L(0,3)=AG4; alpha2=solve(%,alpha2); assign(%);
```

$$9\ \alpha 2 + 3\ \beta 2 + AG2 = AG4$$

$$\alpha 2 = -\frac{1}{3}\ \beta 2 - \frac{1}{9}\ AG2 + \frac{1}{9}\ AG4$$

```
> L(4,3)=AG5; alpha1=solve(%,alpha1); assign(%);
```

$$36\ \alpha 1 - AG2 + AG4 + 12\ \beta 1 + AG3 = AG5$$

$$\alpha 1 = -\frac{1}{36}\ AG3 + \frac{1}{36}\ AG2 - \frac{1}{36}\ AG4 - \frac{1}{3}\ \beta 1 + \frac{1}{36}\ AG5$$

```
> L(0,6)=AG6; beta2=solve(%,beta2); assign(%);
```

$$-6\ \beta 2 - 3\ AG2 + 4\ AG4 = AG6$$

$$\beta 2 = -\frac{1}{2}\ AG2 + \frac{2}{3}\ AG4 - \frac{1}{6}\ AG6$$

```
> L(4,6)=AG7; beta1=solve(%,beta1); assign(%);
```

$$-3\ AG3 + 3\ AG2 - 4\ AG4 - 24\ \beta 1 + 4\ AG5 + AG6 = AG7$$

$$\beta 1 = -\frac{1}{8}\ AG3 + \frac{1}{8}\ AG2 - \frac{1}{6}\ AG4 + \frac{1}{6}\ AG5 + \frac{1}{24}\ AG6 - \frac{1}{24}\ AG7$$

```
> L(AH5,AH3);
```

$$\left(\left(\frac{1}{72}\ AG3 - \frac{1}{72}\ AG2 + \frac{1}{36}\ AG4 - \frac{1}{36}\ AG5 - \frac{1}{72}\ AG6 + \frac{1}{72}\ AG7\right) AH5 + \frac{1}{18}\ AG2 - \frac{1}{9}\ AG4 + \frac{1}{18}\ AG6\right) AH3^2 + \left(\left(-\frac{1}{8}\ AG3 + \frac{1}{8}\ AG2 - \frac{1}{6}\ AG4 + \frac{1}{6}\ AG5 + \frac{1}{24}\ AG6 - \frac{1}{24}\ AG7\right) AH5 - \frac{1}{2}\ AG2 + \frac{2}{3}\ AG4 - \frac{1}{6}\ AG6\right) AH3 + \left(-\frac{1}{4}\ AG2 + \frac{1}{4}\ AG3\right) AH5 + AG2$$

```
> R:=(AH5,AH3)-
>subs(AG6=AD6,subs(AG5=AD5,subs(AG4=AD4,subs(AG3=AD3,subs(AG2=AD
2,subs(AG7=AD7,L(AH5,AH3)))))));
```

$$R := (AH5, AH3) \rightarrow \mathrm{subs}(AG6 = AD6, \mathrm{subs}(AG5 = AD5, \mathrm{subs}(AG4 = AD4, \mathrm{subs}(AG3 = AD3, \mathrm{subs}(AG2 = AD2, \mathrm{subs}(AG7 = AD7, \mathrm{L}(AH5, AH3)))))))$$

```
> R(AH5,AH3):=R(AH5,AH3);
```

$$\mathrm{R}(AH5, AH3) := \left(\left(\frac{1}{72}\ AD3 - \frac{1}{72}\ AD2 + \frac{1}{36}\ AD4 - \frac{1}{36}\ AD5 - \frac{1}{72}\ AD6 + \frac{1}{72}\ AD7\right) AH5 + \frac{1}{18}\ AD2 - \frac{1}{9}\ AD4 + \frac{1}{18}\ AD6\right) AH3^2 + \left(\left(-\frac{1}{8}\ AD3 + \frac{1}{8}\ AD2 - \frac{1}{6}\ AD4 + \frac{1}{6}\ AD5 + \frac{1}{24}\ AD6 - \frac{1}{24}\ AD7\right) AH5 - \frac{1}{2}\ AD2 + \frac{2}{3}\ AD4 - \frac{1}{6}\ AD6\right) AH3 + \left(-\frac{1}{4}\ AD2 + \frac{1}{4}\ AD3\right) AH5 + AD2$$

```
> xeqnL:=solve(L(AH5,AH3)=AY2,AH5);
```

$$
\begin{aligned}
&xeqnL := 4\,(-AH3^2\,AG2 + 2\,AH3^2\,AG4 - AH3^2\,AG6 - 18\,AG2 + 9\,AH3\,AG2 \\
&\quad - 12\,AH3\,AG4 + 3\,AH3\,AG6 + 18\,AY2)\,\big/\,(-9\,AH3\,AG3 + 9\,AH3\,AG2 \\
&\quad - 12\,AH3\,AG4 + 12\,AH3\,AG5 + 3\,AH3\,AG6 - 3\,AH3\,AG7 - 18\,AG2 + 18\,AG3 \\
&\quad + AH3^2\,AG3 - AH3^2\,AG2 + 2\,AH3^2\,AG4 - 2\,AH3^2\,AG5 - AH3^2\,AG6 + AH3^2\,AG7)
\end{aligned}
$$

```
> Rfy:=subs(AH5=xeqnL,R(AH5,AH3));
```

$$
\begin{aligned}
&Rfy := \Bigg(4\,\Bigg(\Bigg(\frac{1}{72}\,AD3 - \frac{1}{72}\,AD2 + \frac{1}{36}\,AD4 - \frac{1}{36}\,AD5 - \frac{1}{72}\,AD6 + \frac{1}{72}\,AD7\Bigg)\,(-AH3^2\,AG2 \\
&\quad + 2\,AH3^2\,AG4 - AH3^2\,AG6 - 18\,AG2 + 9\,AH3\,AG2 - 12\,AH3\,AG4 + 3\,AH3\,AG6 \\
&\quad + 18\,AY2)\Bigg)\,\Big/\,(-9\,AH3\,AG3 + 9\,AH3\,AG2 - 12\,AH3\,AG4 + 12\,AH3\,AG5 \\
&\quad + 3\,AH3\,AG6 - 3\,AH3\,AG7 - 18\,AG2 + 18\,AG3 + AH3^2\,AG3 - AH3^2\,AG2 \\
&\quad + 2\,AH3^2\,AG4 - 2\,AH3^2\,AG5 - AH3^2\,AG6 + AH3^2\,AG7) + \frac{1}{18}\,AD2 - \frac{1}{9}\,AD4 \\
&\quad + \frac{1}{18}\,AD6\Bigg)\,AH3^2 + \Bigg(4\,\Bigg(\Bigg(-\frac{1}{8}\,AD3 + \frac{1}{8}\,AD2 - \frac{1}{6}\,AD4 + \frac{1}{6}\,AD5 + \frac{1}{24}\,AD6 - \frac{1}{24}\,AD7\Bigg)\,(\\
&\quad -AH3^2\,AG2 + 2\,AH3^2\,AG4 - AH3^2\,AG6 - 18\,AG2 + 9\,AH3\,AG2 - 12\,AH3\,AG4 \\
&\quad + 3\,AH3\,AG6 + 18\,AY2)\Bigg)\,\Big/\,(-9\,AH3\,AG3 + 9\,AH3\,AG2 - 12\,AH3\,AG4 \\
&\quad + 12\,AH3\,AG5 + 3\,AH3\,AG6 - 3\,AH3\,AG7 - 18\,AG2 + 18\,AG3 + AH3^2\,AG3 \\
&\quad - AH3^2\,AG2 + 2\,AH3^2\,AG4 - 2\,AH3^2\,AG5 - AH3^2\,AG6 + AH3^2\,AG7) - \frac{1}{2}\,AD2 \\
&\quad + \frac{2}{3}\,AD4 - \frac{1}{6}\,AD6\Bigg)\,AH3 + 4\,\Bigg(-\frac{1}{4}\,AD2 + \frac{1}{4}\,AD3\Bigg)\,(-AH3^2\,AG2 + 2\,AH3^2\,AG4 \\
&\quad - AH3^2\,AG6 - 18\,AG2 + 9\,AH3\,AG2 - 12\,AH3\,AG4 + 3\,AH3\,AG6 + 18\,AY2)\,\big/\,(\\
&\quad -9\,AH3\,AG3 + 9\,AH3\,AG2 - 12\,AH3\,AG4 + 12\,AH3\,AG5 + 3\,AH3\,AG6 \\
&\quad - 3\,AH3\,AG7 - 18\,AG2 + 18\,AG3 + AH3^2\,AG3 - AH3^2\,AG2 + 2\,AH3^2\,AG4 \\
&\quad - 2\,AH3^2\,AG5 - AH3^2\,AG6 + AH3^2\,AG7) + AD2
\end{aligned}
$$

```
> isolate(Rfy=res,AH3);
```

$$
\begin{aligned}
&(-54\,AD7\,AH3\,AG2 + 9\,AD7\,AH3^2\,AG6 - 36\,AD7\,AH3^2\,AG4 + 45\,AD7\,AH3^2\,AG2 \\
&\quad + 54\,AD2\,AH3\,AG6 + 117\,AD3\,AH3^2\,AG2 - 324\,AD3\,AH3\,AG2 \\
&\quad + 45\,AD3\,AH3^2\,AG6 - 144\,AD3\,AH3^2\,AG4 - 54\,AD3\,AH3\,AG6 \\
&\quad - 18\,AD2\,AH3^2\,AG2 + 36\,AD2\,AH3^2\,AG4 + 54\,AH3\,AD7\,AY2 + 162\,AD2\,AH3\,AG2 \\
&\quad - 18\,AD2\,AH3^2\,AG6 - 54\,AH3\,AD6\,AY2 - 216\,AH3\,AD5\,AY2 + 216\,AH3\,AD4\,AY2 \\
&\quad - 162\,AH3\,AD2\,AY2 + 162\,AH3\,AD3\,AY2 + 216\,AD5\,AH3\,AG2 \\
&\quad - 36\,AD5\,AH3^2\,AG6 + 144\,AD5\,AH3^2\,AG4 - 144\,AD5\,AH3^2\,AG2 \\
&\quad - 216\,AD2\,AH3\,AG4 - 9\,AD6\,AH3^2\,AG7 + 36\,AD6\,AH3^2\,AG5 - 45\,AD6\,AH3^2\,AG3
\end{aligned}
$$

$$\begin{aligned}
&+ 54\, AD6\, AH3\, AG3 + 36\, AD4\, AH3^2\, AG7 - 144\, AD4\, AH3^2\, AG5\\
&+ 144\, AD4\, AH3^2\, AG3 - 216\, AD4\, AH3\, AG3 - 27\, AD2\, AH3^2\, AG7\\
&+ 108\, AD2\, AH3^2\, AG5 - 99\, AD2\, AH3^2\, AG3 + 162\, AD2\, AH3\, AG3\\
&+ 6\, AH3^3\, AD6\, AG7 - 18\, AH3^3\, AD6\, AG5 + 12\, AH3^3\, AD6\, AG3 + 2\, AH3^4\, AD4\, AG7\\
&- 4\, AH3^4\, AD4\, AG5 + 2\, AH3^4\, AD4\, AG3 - 18\, AH3^3\, AD4\, AG7 + 48\, AH3^3\, AD4\, AG5\\
&- 30\, AH3^3\, AD4\, AG3 - AH3^4\, AD2\, AG7 + 2\, AH3^4\, AD2\, AG5 - AH3^4\, AD2\, AG3\\
&+ 12\, AH3^3\, AD2\, AG7 - 30\, AH3^3\, AD2\, AG5 + 18\, AH3^3\, AD2\, AG3\\
&+ 18\, AH3^2\, AD6\, AY2 + 36\, AH3^2\, AD5\, AY2 - 36\, AH3^2\, AD4\, AY2 + 18\, AH3^2\, AD2\, AY2\\
&- 18\, AH3^2\, AD3\, AY2 - 12\, AH3^3\, AD3\, AG6 - 18\, AH3^3\, AD3\, AG2 + AH3^4\, AD3\, AG6\\
&- 2\, AH3^4\, AD3\, AG4 + AH3^4\, AD3\, AG2 - 18\, AH3^2\, AD7\, AY2 - 6\, AH3^3\, AD7\, AG6\\
&+ 18\, AH3^3\, AD7\, AG4 - 12\, AH3^3\, AD7\, AG2 + AH3^4\, AD7\, AG6 - 2\, AH3^4\, AD7\, AG4\\
&+ AH3^4\, AD7\, AG2 + 18\, AH3^3\, AD5\, AG6 + 30\, AH3^3\, AD3\, AG4 - 48\, AH3^3\, AD5\, AG4\\
&+ 30\, AH3^3\, AD5\, AG2 - 2\, AH3^4\, AD5\, AG6 + 4\, AH3^4\, AD5\, AG4 - 2\, AH3^4\, AD5\, AG2\\
&- AH3^4\, AD6\, AG7 + 2\, AH3^4\, AD6\, AG5 - AH3^4\, AD6\, AG3 + 216\, AD3\, AH3\, AG4\\
&+ 324\, AD3\, AG2 - 324\, AD3\, AY2 - 324\, AD2\, AG2 + 324\, AD2\, AY2\,) \bigg/ (-9\, AH3\, AG3\\
&+ 9\, AH3\, AG2 - 12\, AH3\, AG4 + 12\, AH3\, AG5 + 3\, AH3\, AG6 - 3\, AH3\, AG7 - 18\, AG2\\
&+ 18\, AG3 + AH3^2\, AG3 - AH3^2\, AG2 + 2\, AH3^2\, AG4 - 2\, AH3^2\, AG5 - AH3^2\, AG6\\
&+ AH3^2\, AG7\,) = -18\, res + 18\, AD2
\end{aligned}$$

Before data analysis began, Maple was also used to determine the rudder angle from the reading of the caliper in tank testing.

```
> restart:
> e1:=(C3)^2=A^2+C^2-2*A*C*cos(beta);
```

$$e1 := C3^2 = A^2 + C^2 - 2\, A\, C \cos(\beta)$$

```
> betarad:=solve(e1,beta);
```

$$betarad := \pi - \arccos\left(\frac{1}{2}\,\frac{C3^2 - A^2 - C^2}{A\, C}\right)$$

```
> A:=10.31; C:=10;
```

$$A := 10.31$$

$$C := 10$$

```
> betarad;
```

$$\pi - \arccos(.004849660524\;\; C3^2 - 1.000466052\,)$$

```
> beta:=evalf(betarad*180/Pi);
```

$$\beta := 180.0000000\; - 57.29577950\;\; \arccos(.004849660524\;\; C3^2 - 1.000466052\,)$$

```
> beta;
```

$$180.0000000 - 57.29577950\ \arccos(.004849660524\ C3^{2} - 1.000466052\ )$$

```
> solve(beta=ang,C3);
```

$$\frac{10}{1212415131}\sqrt{3032450448741582030 + 3031037827500000000\ \cos\left(-\frac{360000000}{114591559} + \frac{2000}{11459}\right.}$$

$$, -\frac{10}{1212415131}\sqrt{3032450448741582030 + 3031037827500000000\ \cos\left(-\frac{360000000}{114591559} +\right.}$$

```
> e2:=angle=180-57.2957795*arccos(0.004849660524*(C3+2)^2-
1.000466052)-11.16683841;
```

$$e2 := angle =$$

$$168.8331616 - 57.2957795\ \arccos(.004849660524\ (C3 + 2\ )^{2} - 1.000466052\ )$$

```
> b:=solve(e2,C3)[1];
```

$$b := -2 + \frac{10}{1212415131}\sqrt{3032450448741582030 + 3031037827500000000\ \cos\left(\frac{200000}{1145915}\right.}$$

```
> evalf(b,4);
```

$$-2. + .8248\ 10^{-8}\sqrt{.3032\ 10^{19} + .3031\ 10^{19}\cos(.01745329252\ angle - 2.946694557\ )}$$

APPENDIX D – VELOCITY PREDICTION PROGRAM

The first visual basic module was included in the Mk II Navy 44 VPP. This module interpolated through the data, and also solved for the righting moment.

```
Attribute VB_Name = "Module1"
Function rarm(heel)
rarm = heel ^ 3 * 0.01831 + heel ^ 2 * 5.249 + heel * -2355.7
End Function
Function shull(speed, rud, yaw, heel)
Application.Volatile
Const d00 = 1010
Const d03 = d00 + 24
Const d06 = d03 + 24
Const d40 = d06 + 24
Const d43 = d40 + 24
Const d46 = d43 + 24

D1 = interp(d00, speed, heel)
D2 = interp(d03, speed, heel)
D3 = interp(d06, speed, heel)
D4 = interp(d40, speed, heel)
D5 = interp(d03, speed, heel)
D6 = interp(d46, speed, heel)
shull = 1 / 72 * rud ^ 2 * yaw * D4 - 1 / 72 * rud ^ 2 * yaw * D1 + 1 / 36 *
rud ^ 2 * yaw * D2 - 1 / 36 * rud ^ 2 * yaw * D5 - 1 / 72 * rud ^ 2 * yaw *
D3 + 1 / 72 * rud ^ 2 * yaw * D6 + 1 / 18 * rud ^ 2 * D1 - 1 / 9 * rud ^ 2 *
D2 + 1 / 18 * rud ^ 2 * D3 - 1 / 8 * rud * yaw * D4 + 1 / 8 * rud * yaw * D1
- 1 / 6 * rud * yaw * D2 + 1 / 6 * rud * yaw * D5 + 1 / 24 * rud * yaw * D3 -
1 / 24 * rud * yaw * D6 - 1 / 2 * rud * D1 + 2 / 3 * rud * D2 - 1 / 6 * rud *
D3 - 1 / 4 * yaw * D1 + 1 / 4 * yaw * D4 + D1
End Function


Function liftk(speed, rud, yaw, heel)
Application.Volatile
Const d00 = 290
Const d03 = d00 + 24
Const d06 = d03 + 24
Const d40 = d06 + 24
Const d43 = d40 + 24
Const d46 = d43 + 24

D1 = interp(d00, speed, heel)
D2 = interp(d03, speed, heel)
D3 = interp(d06, speed, heel)
D4 = interp(d40, speed, heel)
D5 = interp(d43, speed, heel)
D6 = interp(d46, speed, heel)
liftk = 1 / 72 * rud ^ 2 * yaw * D4 - 1 / 72 * rud ^ 2 * yaw * D1 + 1 / 36 *
rud ^ 2 * yaw * D2 - 1 / 36 * rud ^ 2 * yaw * D5 - 1 / 72 * rud ^ 2 * yaw *
D3 + 1 / 72 * rud ^ 2 * yaw * D6 + 1 / 18 * rud ^ 2 * D1 - 1 / 9 * rud ^ 2 *
D2 + 1 / 18 * rud ^ 2 * D3 - 1 / 8 * rud * yaw * D4 + 1 / 8 * rud * yaw * D1
- 1 / 6 * rud * yaw * D2 + 1 / 6 * rud * yaw * D5 + 1 / 24 * rud * yaw * D3 -
```

```
1 / 24 * rud * yaw * D6 - 1 / 2 * rud * D1 + 2 / 3 * rud * D2 - 1 / 6 * rud *
D3 - 1 / 4 * yaw * D1 + 1 / 4 * yaw * D4 + D1
End Function

Function liftr(speed, rud, yaw, heel)
Application.Volatile
Const d00 = 1586
Const d03 = d00 + 24
Const d06 = d03 + 24
Const d40 = d06 + 24
Const d43 = d40 + 24
Const d46 = d43 + 24

D1 = interp(d00, speed, heel)
D2 = interp(d03, speed, heel)
D3 = interp(d06, speed, heel)
D4 = interp(d40, speed, heel)
D5 = interp(d43, speed, heel)
D6 = interp(d46, speed, heel)
liftr = 1 / 72 * rud ^ 2 * yaw * D4 - 1 / 72 * rud ^ 2 * yaw * D1 + 1 / 36 *
rud ^ 2 * yaw * D2 - 1 / 36 * rud ^ 2 * yaw * D5 - 1 / 72 * rud ^ 2 * yaw *
D3 + 1 / 72 * rud ^ 2 * yaw * D6 + 1 / 18 * rud ^ 2 * D1 - 1 / 9 * rud ^ 2 *
D2 + 1 / 18 * rud ^ 2 * D3 - 1 / 8 * rud * yaw * D4 + 1 / 8 * rud * yaw * D1
- 1 / 6 * rud * yaw * D2 + 1 / 6 * rud * yaw * D5 + 1 / 24 * rud * yaw * D3 -
1 / 24 * rud * yaw * D6 - 1 / 2 * rud * D1 + 2 / 3 * rud * D2 - 1 / 6 * rud *
D3 - 1 / 4 * yaw * D1 + 1 / 4 * yaw * D4 + D1
End Function

Function drag(speed, rud, yaw, heel)
Application.Volatile
Const d00 = 1154
Const d03 = d00 + 24
Const d06 = d03 + 24
Const d40 = d06 + 24
Const d43 = d40 + 24
Const d46 = d43 + 24

D1 = interp(d00, speed, heel)
D2 = interp(d03, speed, heel)
D3 = interp(d06, speed, heel)
D4 = interp(d40, speed, heel)
D5 = interp(d43, speed, heel)
D6 = interp(d46, speed, heel)
drag = 1 / 72 * rud ^ 2 * yaw * D4 - 1 / 72 * rud ^ 2 * yaw * D1 + 1 / 36 *
rud ^ 2 * yaw * D2 - 1 / 36 * rud ^ 2 * yaw * D5 - 1 / 72 * rud ^ 2 * yaw *
D3 + 1 / 72 * rud ^ 2 * yaw * D6 + 1 / 18 * rud ^ 2 * D1 - 1 / 9 * rud ^ 2 *
D2 + 1 / 18 * rud ^ 2 * D3 - 1 / 8 * rud * yaw * D4 + 1 / 8 * rud * yaw * D1
- 1 / 6 * rud * yaw * D2 + 1 / 6 * rud * yaw * D5 + 1 / 24 * rud * yaw * D3 -
1 / 24 * rud * yaw * D6 - 1 / 2 * rud * D1 + 2 / 3 * rud * D2 - 1 / 6 * rud *
D3 - 1 / 4 * yaw * D1 + 1 / 4 * yaw * D4 + D1
If drag < 0 Then
    drag = 0
End If
End Function

Function dragw(speed, rud, yaw, heel)
```

```
Application.Volatile
Const d00 = 1442
Const d03 = d00 + 24
Const d06 = d03 + 24
Const d40 = d06 + 24
Const d43 = d40 + 24
Const d46 = d43 + 24

D1 = interp(d00, speed, heel)
D2 = interp(d03, speed, heel)
D3 = interp(d06, speed, heel)
D4 = interp(d40, speed, heel)
D5 = interp(d43, speed, heel)
D6 = interp(d46, speed, heel)
dragw = 1 / 72 * rud ^ 2 * yaw * D4 - 1 / 72 * rud ^ 2 * yaw * D1 + 1 / 36 *
rud ^ 2 * yaw * D2 - 1 / 36 * rud ^ 2 * yaw * D5 - 1 / 72 * rud ^ 2 * yaw *
D3 + 1 / 72 * rud ^ 2 * yaw * D6 + 1 / 18 * rud ^ 2 * D1 - 1 / 9 * rud ^ 2 *
D2 + 1 / 18 * rud ^ 2 * D3 - 1 / 8 * rud * yaw * D4 + 1 / 8 * rud * yaw * D1
- 1 / 6 * rud * yaw * D2 + 1 / 6 * rud * yaw * D5 + 1 / 24 * rud * yaw * D3 -
1 / 24 * rud * yaw * D6 - 1 / 2 * rud * D1 + 2 / 3 * rud * D2 - 1 / 6 * rud *
D3 - 1 / 4 * yaw * D1 + 1 / 4 * yaw * D4 + D1
End Function

Function interp(cond, speed, heel)
Dim i As Integer
Dim j As Integer
Dim heels As Variant
Dim heelrat As Double
Dim speeds As Variant
Dim speedrat As Double
Dim drags As Variant
Dim s1 As Double
Dim s2 As Double
Dim l1 As String: Dim l2 As String: Dim l3 As String

l1 = "A" & Trim(Str(cond + 1))
l2 = "B" & Trim(Str(cond))
l3 = "B" & Trim(Str(cond + 1))

'If speed < heel ^ 3 * 0.0002854 - 0.01427 * heel ^ 2 + 0.2817 * heel + 1
Then
'    speed = heel ^ 3 * 0.0002854 - 0.01427 * heel ^ 2 + 0.2817 * heel + 1
'    Sheets("Calc").Range("A1").Value = 0
'Else
'    Sheets("Calc").Range("A1").Value = 1
'End If

heels = Sheets("DATA").Range(l1).Resize(21, 1)
If heel > heels(21, 1) Then
    i = 21
    heelrat = 1
    ElseIf heel <= 0 Then
    i = 2
    heelrat = 0
Else
    i = 1
    Do While heel > heels(i, 1)
        i = i + 1
    Loop
    heelrat = (heel - heels(i - 1, 1)) / (heels(i, 1) - heels(i - 1, 1))
End If
speeds = Sheets("DATA").Range(l2).Resize(1, 26)
If speed > speeds(1, 26) Then
    j = 26
    speedrat = 1
ElseIf speed <= 0 Then
    j = 2
    speedrat = 0
Else
    j = 1
    Do While speed > speeds(1, j)
        j = j + 1
    Loop

        speedrat = (speed - speeds(1, j - 1)) / (speeds(1, j) - speeds(1, j -
1))
End If
drags = Sheets("DATA").Range(l3).Resize(i, j)
s1 = -(drags(i - 1, j - 1) - drags(i, j - 1)) * heelrat + drags(i - 1, j - 1)
s2 = -(drags(i - 1, j) - drags(i, j)) * heelrat + drags(i - 1, j)
interp = -(s1 - s2) * speedrat + s1
End Function

Function interptwo(cond, speed, heel)
Dim i As Integer
Dim j As Integer
Dim heels As Variant
Dim heelrat As Double
Dim speeds As Variant
Dim speedrat As Double
Dim drags As Variant
Dim s1 As Double
Dim s2 As Double
Dim l1 As String: Dim l2 As String: Dim l3 As String

l1 = "AC" & Trim(Str(cond + 1))
l2 = "AD" & Trim(Str(cond))
l3 = "AD" & Trim(Str(cond + 1))

heels = Sheets("DATA").Range(l1).Resize(21, 1)

If heel > heels(21, 1) Then
    i = 21
    heelrat = 1
    ElseIf heel <= 0 Then
    i = 2
    heelrat = 0
Else
    i = 1
    Do While heel > heels(i, 1)
        i = i + 1
    Loop
    heelrat = (heel - heels(i - 1, 1)) / (heels(i, 1) - heels(i - 1, 1))
End If
speeds = Sheets("DATA").Range(l2).Resize(1, 21)
If speed > speeds(1, 21) Then
    j = 21
    speedrat = 1
ElseIf speed <= 0 Then
    j = 2
    speedrat = 0
Else
    j = 1
    Do While speed > speeds(1, j)
        j = j + 1
    Loop

        speedrat = (speed - speeds(1, j - 1)) / (speeds(1, j) - speeds(1, j -
1))
End If
drags = Sheets("DATA").Range(l3).Resize(i, j)
s1 = -(drags(i - 1, j - 1) - drags(i, j - 1)) * heelrat + drags(i - 1, j - 1)
s2 = -(drags(i - 1, j) - drags(i, j)) * heelrat + drags(i - 1, j)
interptwo = -(s1 - s2) * speedrat + s1
End Function

Function LinInt(cond, speed, heel, starty, startx, lengths, lengthh)
Dim i As Integer
Dim j As Integer
Dim heels As Variant
Dim heelrat As Double
Dim speeds As Variant
Dim speedrat As Double
Dim drags As Variant
Dim s1 As Double
Dim s2 As Double
Dim l1 As String: Dim l2 As String: Dim l3 As String

l1 = "starty" & Trim(Str(cond + 1))
l2 = "startx" & Trim(Str(cond))
l3 = "startx" & Trim(Str(cond + 1))

heels = Sheets("DATA").Range(l1).Resize(lengthh, 1)

If heel > heels(lengthh, 1) Then
    i = lengthh
    heelrat = 1
    ElseIf heel <= 0 Then
    i = 2
    heelrat = 0
Else
    i = 1
    Do While heel > heels(i, 1)
        i = i + 1
    Loop
    heelrat = (heel - heels(i - 1, 1)) / (heels(i, 1) - heels(i - 1, 1))
End If
speeds = Sheets("DATA").Range(l2).Resize(1, lengths)
```

```
If speed > speeds(1, lengths) Then
    j = lengths
    speedrat = 1
ElseIf speed <= 0 Then
    j = 2
    speedrat = 0
Else
    j = 1
    Do While speed > speeds(1, j)
        j = j + 1
    Loop

        speedrat = (speed - speeds(1, j - 1)) / (speeds(1, j) - speeds(1, j -
1))
End If
drags = Sheets("DATA").Range(l3).Resize(i, j)
s1 = -(drags(i - 1, j - 1) - drags(i, j - 1)) * heelrat + drags(i - 1, j - 1)
s2 = -(drags(i - 1, j) - drags(i, j)) * heelrat + drags(i - 1, j)
LinInt = -(s1 - s2) * speedrat + s1
End Function
```

The second module solved for the sail force coefficients.

```
Attribute VB_Name = "Module2"
Function clmain(beta)
If beta < 36 Then
clmain = beta ^ 4 * -0.0000033616 + beta ^ 3 * 0.00033299 + beta ^ 2 * -
0.01251 + beta * 0.2182
Else
clmain = 0.000000186041 * beta ^ 3 - 0.0000469267 * beta ^ 2 - 0.0080303 *
beta + 1.875128288
End If
End Function

Function cljib(beta)
If beta < 7 Then
cljib = 0
ElseIf beta < 20 Then
cljib = 1.401482098 * Log(beta) - 2.720302423
ElseIf beta < 50 Then
cljib = -0.000231061 * beta ^ 2 + 0.014840909 * beta + 1.255606061
ElseIf beta < 101 Then
cljib = -0.00006 * beta ^ 2 - 0.0114 * beta + 2.14
Else
cljib = 0
End If
End Function
Function cdmain(beta)
If beta < 80 Then
cdmain = 0.0000406693 * beta ^ 2 - 0.000370797 * beta + 0.00593405
ElseIf beta < 148 Then
cdmain = -0.0000000219664 * beta ^ 4 + 0.00000858857 * beta ^ 3 - 0.001114913
* beta ^ 2 + 0.064256605 * beta - 1.272158531
```

```
Else
cdmain = 0.0025 * beta + 0.75
End If
End Function
Function cdjib(beta)
If beta < 28 Then
cdjib = 0.000225322 * beta + 0.004806867
Else
cdjib = -0.0000164176 * beta ^ 2 + 0.011980652 * beta - 0.313550059
End If
End Function
Function clspin(beta)
If beta < 27 Then
clspin = 0
ElseIf beta < 59 Then
clspin = 0.0000269582 * beta ^ 3 - 0.005485895 * beta ^ 2 + 0.369098301 *
beta - 6.497055512
ElseIf beta < 128 Then
clspin = 0.00000119265 * beta ^ 3 - 0.000518634 * beta ^ 2 + 0.051387093 *
beta + 0.248580029
Else
clspin = -0.00000177111 * beta ^ 3 + 0.000770699 * beta ^ 2 - 0.126128629 *
beta + 8.061621883
End If
End Function

Function cdspin(beta)
If beta < 27 Then
cdspin = 0
ElseIf beta < 59 Then
cdspin = 0.000357143 * beta ^ 2 - 0.020714286 * beta + 0.448928571
ElseIf beta < 99 Then
cdspin = -0.000222527 * beta ^ 2 + 0.048659341 * beta - 1.626282967
Else
cdspin = 0.000000932728 * beta ^ 3 - 0.000427881 * beta ^ 2 + 0.064506767 *
beta - 2.087528817
End If
End Function
```

The final module contained the actual VPP solver itself.

```
Attribute VB_Name = "Module3"
Public rs(200, 17) As Double
Public bs(200, 17) As Double
Public cTWA As Range: 'cTWA = "B4"
Public cSails As Range: ' cSails = "M4"
Public cTWS As Range: 'cTWS = "C4"
Public cHeel As Range: 'cHeel = "L4"
Sub Runflatreef()
Dim bs(700, 17) As Double
Dim flat As Double
Dim reef As Double

Range("J33").Formula = 0 'Set rudder matrix to zero
Sheets("Calc").Select   'goto calc sheet
```

```
Set cellrange = Range(Cells(50, 1), Cells(750, 18))
counter = 1

For reef = 0.5 To 1.0001 Step 0.02
    Range("N5").Value = reef
For flat = 0.5 To 1.0001 Step 0.02
    Range("N4").Value = flat

Iterate
If Range("P17").Value = True Then
Else
    Call record(bs, counter)
    counter = counter + 1
    cellrange.Value = bs
End If

Next flat
Next reef

End Sub



Sub IterateWS()
Sheets("Calc").Select

For WA = 30 To 171 Step 10
Range("B4").Select 'TWA
ActiveCell.FormulaR1C1 = WA
If WA < 46 Then
    Range("F40").Formula = "=ABS(L26)" 'Max this
ElseIf WA < 165 Then
    Range("F40").Formula = "=ABS(M26)" 'Max this
Else
    Range("F40").Formula = "=ABS(L26)" 'Max this
End If
If WA < 89 Then
    Range("M4").Value = "MJ" 'Sail Comb
Else
    Range("M4").Value = "MS" 'Sail Comb
End If
Range("L4").Value = 1.5
Range("C4").Select 'TWS
'For WS = 6 To 20 Step 1
WS = 6
    ActiveCell.FormulaR1C1 = WS
    ActiveCell.Offset(0, 1).Range("A1").Select
    Call Iteratemid
    ActiveCell.Offset(36, -3).Range("A1:Q1").Select
    ActiveCell.Offset(0, 6).Range("A1").Activate
    Selection.Copy
    Sheets("Results").Select
    Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:= _
        False, Transpose:=False
    ActiveCell.Offset(1, 0).Range("A1").Select
    Sheets("Calc").Select
    ActiveCell.Offset(-36, -4).Range("A1").Select
    Application.CutCopyMode = False

WS = 12
    ActiveCell.FormulaR1C1 = WS
    ActiveCell.Offset(0, 1).Range("A1").Select
    Call Iteratemid
    ActiveCell.Offset(36, -3).Range("A1:Q1").Select
    ActiveCell.Offset(0, 6).Range("A1").Activate
    Selection.Copy
    Sheets("Results").Select
    Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:= _
        False, Transpose:=False
    ActiveCell.Offset(1, 0).Range("A1").Select
    Sheets("Calc").Select
    ActiveCell.Offset(-36, -4).Range("A1").Select
    Application.CutCopyMode = False

WS = 16
    ActiveCell.FormulaR1C1 = WS
    ActiveCell.Offset(0, 1).Range("A1").Select
    Call Iteratemid
    ActiveCell.Offset(36, -3).Range("A1:Q1").Select
    ActiveCell.Offset(0, 6).Range("A1").Activate
    Selection.Copy
    Sheets("Results").Select
    Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:= _
        False, Transpose:=False
    ActiveCell.Offset(1, 0).Range("A1").Select
    Sheets("Calc").Select
    ActiveCell.Offset(-36, -4).Range("A1").Select
    Application.CutCopyMode = False

WS = 20
    ActiveCell.FormulaR1C1 = WS
    ActiveCell.Offset(0, 1).Range("A1").Select
    Call Iteratemid
    ActiveCell.Offset(36, -3).Range("A1:Q1").Select
    ActiveCell.Offset(0, 6).Range("A1").Activate
    Selection.Copy
    Sheets("Results").Select
    Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:= _
        False, Transpose:=False
    ActiveCell.Offset(1, 0).Range("A1").Select
    Sheets("Calc").Select
    ActiveCell.Offset(-36, -4).Range("A1").Select
    Application.CutCopyMode = False

'Next WS

Next WA
End Sub

Sub Iteratemid()
Dim bs(400, 17) As Double
Dim rs(400, 17) As Double
Dim flat As Double
Dim reef As Double
Dim counter As Integer
Dim cellrange As Range
Dim rudder As Double
Sheets("Calc").Select   'goto calc sheet
Set cellrange = Range(Cells(50, 1), Cells(450, 18))
counter = 1
Range("J33").Formula = 0 'Set rudder matrix to zero

Range("P14").Value = 1   'Turn calc off
rudder = 0: Range("K4").Value = rudder 'Get rudder angle

If Range("C4").Value < 13 Then 'Get windspeed
    flat = 1: Range("N4").Value = flat 'flat
    reef = 1: Range("N5").Value = reef  'reef
ElseIf Range("C4").Value < 17 Then 'Get windspeed
    flat = 0.94: Range("N4").Value = flat 'flat
    reef = 0.94: Range("N5").Value = reef  'reef
Else
    flat = 0.9: Range("N4").Value = flat 'set initial flat
    reef = 0.9: Range("N5").Value = reef 'get initial reef
End If

Calculate
Range("P14").Value = 0  'Turn calc on
movement = True
rfd = 0.02

If flat + reef = 2 Then
    movement = False
End If

Iterate
Call record(bs, counter)
counter = counter + 1
stopit = 0

While movement = True
    movement = False
    vvg = already(bs, flat, reef, cellrange)
    If vvg = 0 Then
        Iterate
        Call record(bs, counter)
        counter = counter + 1
        first = Range("F40").Value
    Else
        first = vvg
        If counter > 3 Then
            stopit = 1
        End If
    End If

    flat = flat + rfd: Range("N4").Value = flat
    If flat < 1 Then
    vvg = already(bs, flat, reef, cellrange)
```

```
If vvg = 0 Then
    Iterate
    Call record(bs, counter)
    counter = counter + 1
    flatup = Range("F40").Value
Else
    flatup = vvg
End If
flat = flat - rfd: Range("N4").Value = flat
Else: flatup = 0
End If

If reef < 1 Then
reef = reef + rfd: Range("N5").Value = reef
vvg = already(bs, flat, reef, cellrange)
If vvg = 0 Then
    Iterate
    Call record(bs, counter)
    counter = counter + 1
    reefup = Range("F40").Value
Else
    reefup = vvg
End If
reef = reef - rfd: Range("N5").Value = reef
Else: reefup = 0
End If

flat = flat - rfd: Range("N4").Value = flat
vvg = already(bs, flat, reef, cellrange)
If vvg = 0 Then
    Iterate
    Call record(bs, counter)
    counter = counter + 1
    flatdn = Range("F40").Value
Else
    flatdn = vvg
End If
flat = flat + rfd: Range("N4").Value = flat

reef = reef - rfd: Range("N5").Value = reef
vvg = already(bs, flat, reef, cellrange)
If vvg = 0 Then
    Iterate
    Call record(bs, counter)
    counter = counter + 1
    reefdn = Range("F40").Value
Else
    reefdn = vvg
End If
reef = reef + rfd: Range("N5").Value = reef

If flatup > first Then
    flat = flat + rfd: Range("N4").Value = flat: movement = True
End If
If reefup > first Then
    reef = reef + rfd: Range("N5").Value = reef: movement = True
    End If
    If flatdn > first Then
        flat = flat - rfd: Range("N4").Value = flat: movement = True
    End If
    If reefdn > first Then
        reef = reef - rfd: Range("N5").Value = reef: movement = True
    End If

    If flat > 1 Then
        flat = 1: Range("N4").Value = flat
        movement = False
    End If
    If reef > 1 Then
        reef = 1: Range("N5").Value = reef
        movement = False
    End If
    If flat < 0.5 Then movement = False
    If reef < 0.5 Then movement = False

    If counter > 40 Then movement = False
    If stopit = 1 Then movement = False
    'flat = flat + 0.02: Range("N4").Value = flat
    'reef = reef + 0.02: Range("N5").Value = reef
    'If counter > 3 Then
    'If Range("C4").Value > 16 Then
    '    If Range("F40").Value < bs(counter - 3, 6) Then 'Max this variable
    '        reef = reef - 0.02: Range("N5").Value = reef
    '        flat = flat - 0.02: Range("N4").Value = flat
    '    End If
    'End If
    'End If
    'If flat < 0.7 Then reef = 1.02
    'If reef < 0.7 Then flat = 1.02
    cellrange.Value = bs
Wend
    If flat = 1.02 Then
        flat = 1: Range("N4").Value = flat
    End If
    If reef = 1.02 Then
        reef = 1: Range("N5").Value = reef
    End If
    Iterate
    first = Range("F40").Value
    If Range("P17").Value = False Then
        Call record(bs, counter)
        counter = counter + 1
    End If
cellrange.Value = bs
where = MaxL(bs)
flat = bs(where, 11): Range("N4").Value = flat
reef = bs(where, 12): Range("N5").Value = reef
Range("F4").Value = bs(where, 5) 'TBS
Range("L4").Value = bs(where, 8) 'Heel
Range("J4").Value = bs(where, 9) 'Yaw
Range("K4").Value = bs(where, 10) 'Rudder
Range("O4").Value = bs(where, 13) 'Rake
```

```
Reset

dorudder:

Dim counterr As Integer
Dim ruddelta As Double
Dim error2 As Double
rudder = 0: Range("K4").Value = rudder
ruddelta = 2
counterr = 1
Range("J33").Formula = "=Abs(I21)*2" 'Set rudder matrix to not zero
While rudder < 6
    Iterate
    If Range("P15").Value = True Then 'Can it change wrt yaw?
        Range("K4").Value = rudder - ruddelta
        rudder = 7
        GoTo skip
    End If
    Call record(rs, counterr)
    rudder = rudder + ruddelta: Range("K4").Value = rudder
    counterr = counterr + 1
skip:
    'cellrange.Value = rs
Wend
ruddelta = 1
where = MaxL(rs)
flat = rs(where, 11): Range("N4").Value = flat
reef = rs(where, 12): Range("N5").Value = reef
Range("F4").Value = rs(where, 5) 'TBS
Range("L4").Value = rs(where, 8) 'Heel
Range("J4").Value = rs(where, 9) 'Yaw
Range("K4").Value = rs(where, 10) 'Rudder
Range("O4").Value = rs(where, 13) 'Rake
rudder = Range("K4").Value
Reset
Iterate
Call record(rs, counterr)
counterr = counterr + 1
If rudder < 3 Then
    ruddelta = 0.2
Else
    ruddelta = -0.2
End If

While Abs(ruddelta) > 0.05
    Iterate
    If Range("P15").Value = True Then 'Can it change wrt yaw?
        Range("K4").Value = rudder - ruddelta
        ruddelta = 0
        GoTo skip2
    End If
    Call record(rs, counterr)

    error2 = rs(counterr - 1, 6) - rs(counterr - 2, 6)

    If error2 < 0 Then
```

```
        ruddelta = -ruddelta * 1 / 6
    End If

    rudder = rudder + ruddelta: Range("K4").Value = rudder
    counterr = counterr + 1

    If rudder < 0 Then
        ruddelta = 0: rudder = 0: Range("K4").Value = rudder
    End If

    If rudder > 6 Then
        ruddelta = 0: rudder = 6: Range("K4").Value = rudder
    End If

    'cellrange.Value = rs
skip2:
Wend
cellrange.Value = rs
where = MaxL(rs)
flat = rs(where, 11): Range("N4").Value = flat
reef = rs(where, 12): Range("N5").Value = reef
Range("F4").Value = rs(where, 5) 'TBS
Range("L4").Value = rs(where, 8) 'Heel
Range("J4").Value = rs(where, 9) 'Yaw
Range("K4").Value = rs(where, 10) 'Rudder
Range("O4").Value = rs(where, 13) 'Rake
Reset

Iterate


cellrange.Value = bs
Range("J33").Formula = "0"
Exit Sub

RemoveError:
    'flat = 0.5: Range("N4").Value = flat
    reef = reef - 0.01: Range("N5").Value = reef
    Range("P14").Value = 1
    Calculate
    Range("P14").Value = 0
    Calculate


End Sub
Sub Iterate()
Dim i1 As Long
Dim i2 As Integer
i1 = Application.MaxIterations
Application.MaxIterations = 10
i2 = 0
Range("P16").Value = 6
Cagain:
    Calculate
    i2 = i2 + 1
    If Int(i2 / 40) = i2 / 40 Then
```

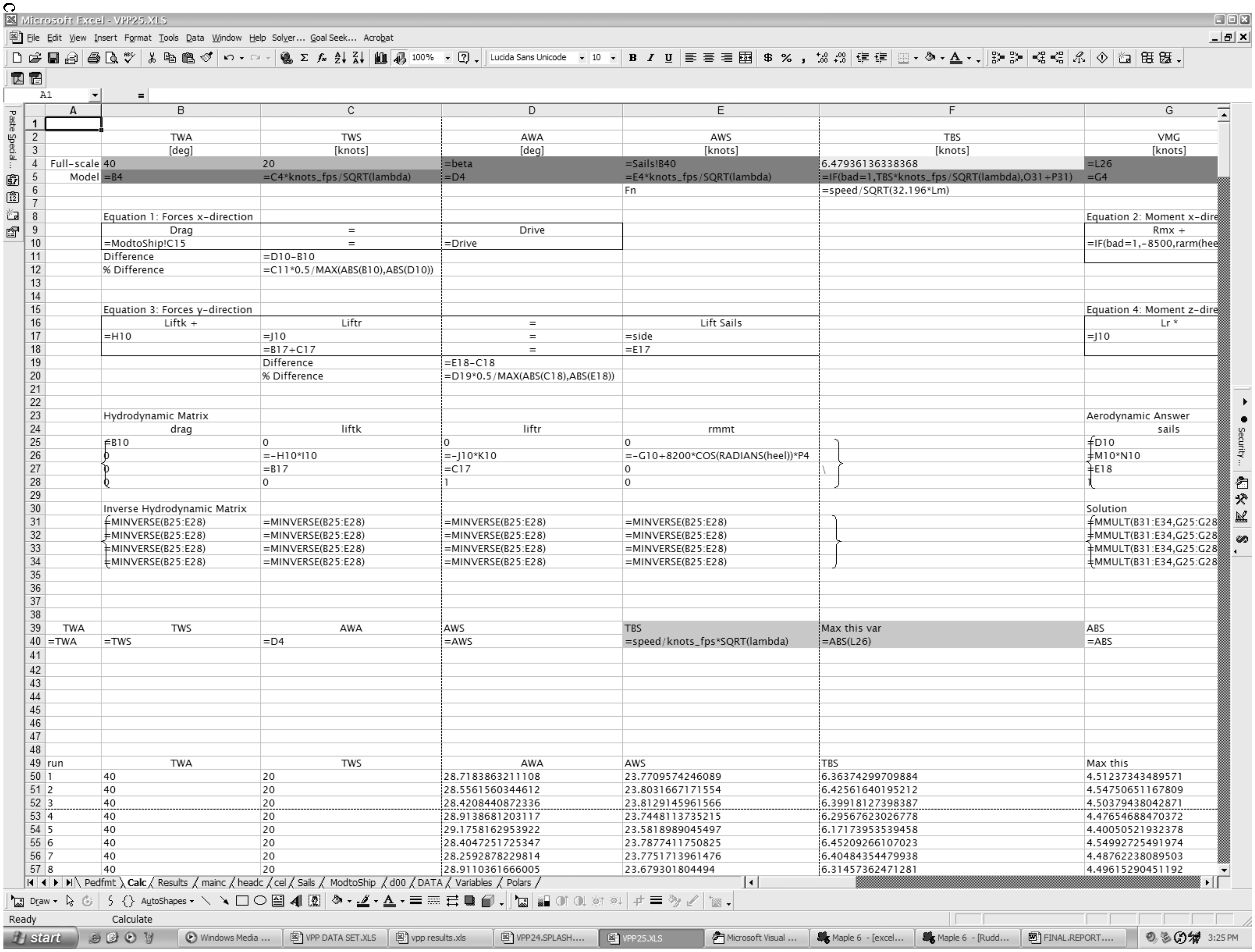
```
        Range("P16").Value = (i2) ^ 0.5 * 2
    End If
    If Range("P18").Value = True Then
        ResetCalc
        Range("P16").Value = Range("P16").Value * 2
    End If
    If Range("L35").Value = False Then
        GoTo Cagain
    End If
Application.MaxIterations = 1
End Sub

Sub ResetCalc()
Dim i1 As Long
Dim heel As Double
i1 = Application.MaxIterations
Application.MaxIterations = 2
heel = Range("L4").Value
Range("L4").Value = Range("L4").Value - 1
Calculate
Range("L4").Value = 3
Range("P14").Value = 1
Calculate

Range("P14").Value = 0
Calculate
Range("L4").Value = heel
Range("P14").Value = 1
Calculate
Range("P14").Value = 0
Calculate
Application.MaxIterations = i1
End Sub

Function MaxL(speeds) As Double
bubba = 0
For i = 0 To UBound(speeds, 1)
    If speeds(i, 6) > bubba Then
        bubba = speeds(i, 6)
        MaxL = i
    End If
Next i
'MsgBox (speeds(MaxL, 6))
End Function

Sub record(bs, counter)
bs(counter - 1, 0) = counter
    bs(counter - 1, 1) = Range("A40").Value 'TWA
    bs(counter - 1, 2) = Range("B40").Value 'TWS
    bs(counter - 1, 3) = Range("C40").Value 'AWA
    bs(counter - 1, 4) = Range("D40").Value 'AWS
    bs(counter - 1, 5) = Range("E40").Value 'TBS
    bs(counter - 1, 6) = Range("F40").Value 'VMG
    bs(counter - 1, 7) = Range("G40").Value 'ABS
    bs(counter - 1, 8) = Range("H40").Value 'Heel
    bs(counter - 1, 9) = Range("I40").Value 'Yaw
    bs(counter - 1, 10) = Range("J40").Value 'Rudder
    bs(counter - 1, 11) = Range("K40").Value 'Flat
    bs(counter - 1, 12) = Range("L40").Value 'Reef
    bs(counter - 1, 13) = Range("M40").Value 'Rake
    bs(counter - 1, 14) = Range("N40").Value 'Heelmom
    bs(counter - 1, 15) = Range("O40").Value 'Sideforc
    bs(counter - 1, 16) = Range("P40").Value 'Drive
    bs(counter - 1, 17) = Range("J34").Value 'Error

End Sub

Function already(bs, flat, reef, cellrange)
already = 0
cellrange.Value = bs
For i = 0 To UBound(bs, 1)
    If Abs(bs(i, 11) - flat) < 0.001 Then
        If Abs(bs(i, 12) - reef) < 0.001 Then
            already = bs(i, 6)
        End If
    End If
Next i
End Function
```

Velocity Prediction Program Spreadsheet

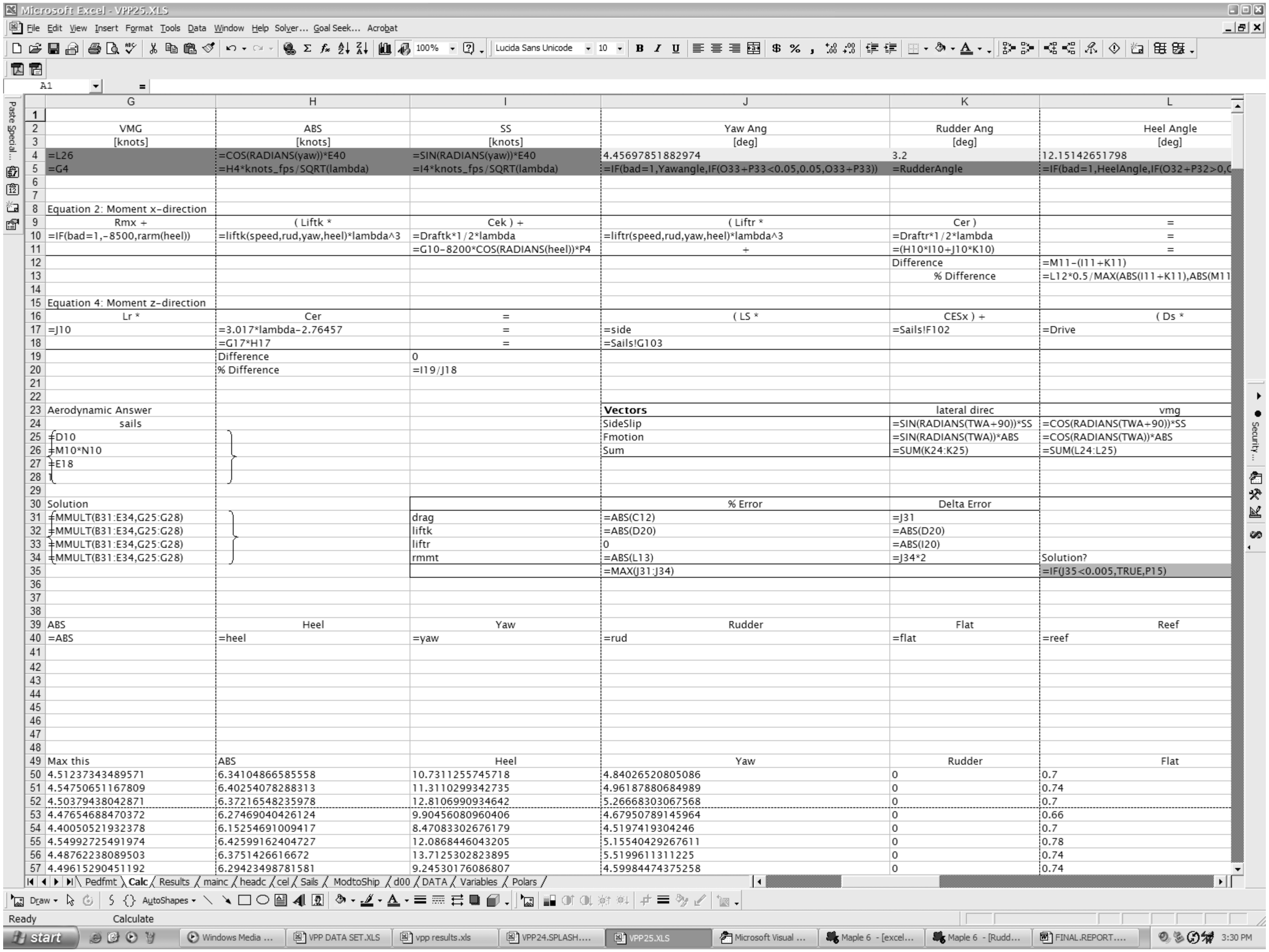
The next few pages contain screenshots of the contents of the actual cells in the tank data spreadsheet.

From the “calc” sheet which was the main engine were these cells:

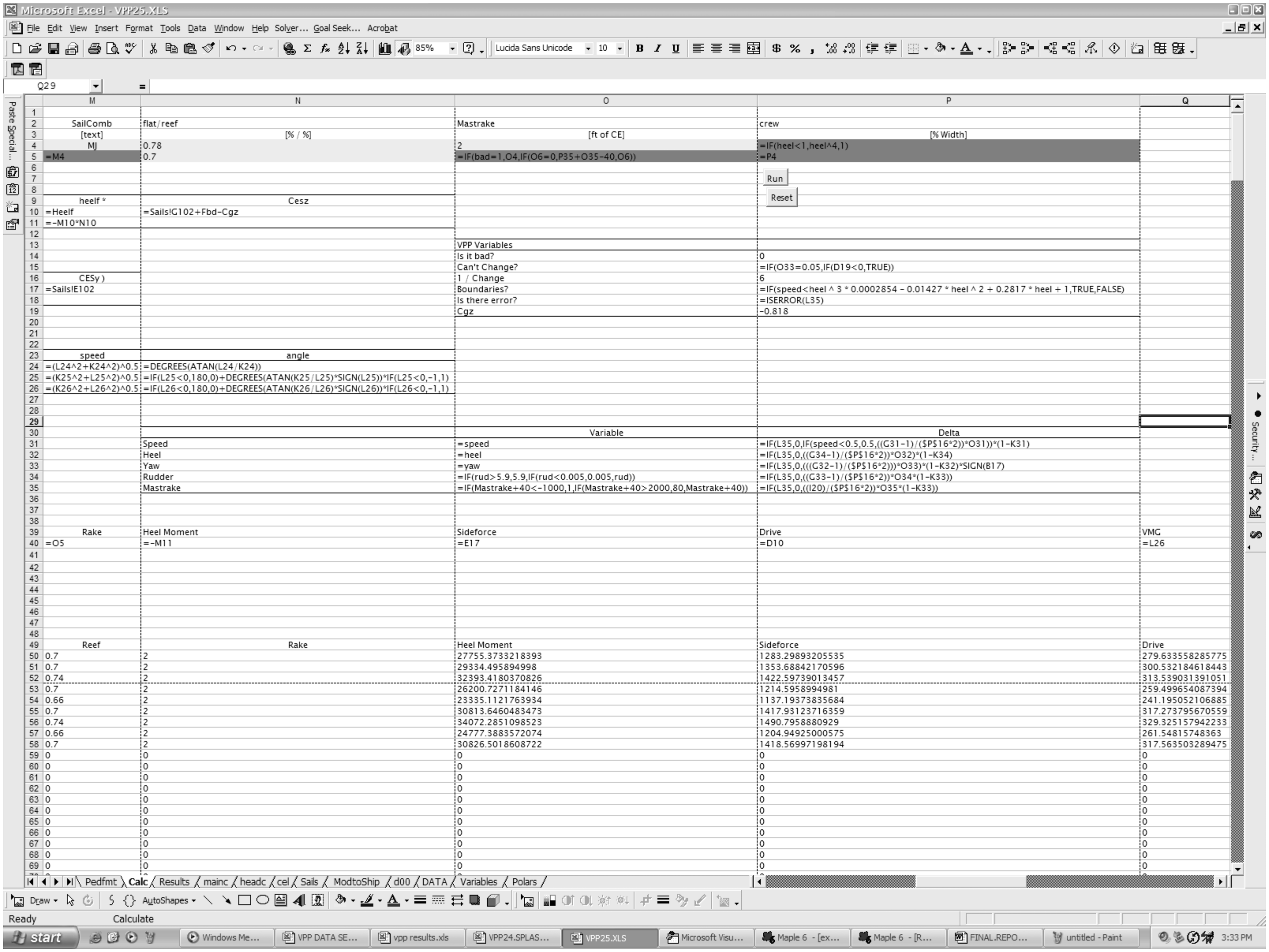
c



	A	B	C	D	E	F	G
1							
2		TWA	TWS	AWA	AWS	TBS	VMG
3		[deg]	[knots]	[deg]	[knots]	[knots]	[knots]
4	Full-scale	40	20	=beta	=Sails!B40	6.47936136338368	=L26
5	Model	=B4	=C4*knots_fps/SQRT(lambda)	=D4	=E4*knots_fps/SQRT(lambda)	=IF(bad=1,TBS*knots_fps/SQRT(lambda),O31+P31)	=G4
6					Fn	=speed/SQRT(32.196*Lm)	
7							
8		Equation 1: Forces x-direction					Equation 2: Moment x-dire
9		Drag	=	Drive			Rmx +
10		=ModtoShip!C15	=	=Drive			=IF(bad=1,-8500,rarm(hee
11		Difference	=D10-B10				
12		% Difference	=C11*0.5/MAX(ABS(B10),ABS(D10))				
13							
14							
15		Equation 3: Forces y-direction					Equation 4: Moment z-dire
16		Liftk +	Liftr	=	Lift Sails		Lr *
17		=H10	=J10	=	=side		=J10
18			=B17+C17	=	=E17		
19			Difference	=E18-C18			
20			% Difference	=D19*0.5/MAX(ABS(C18),ABS(E18))			
21							
22							
23		Hydrodynamic Matrix					Aerodynamic Answer
24		drag	liftk	liftr	rmmt		sails
25		=B10	0	0	0		=D10
26		0	=-H10*I10	=-J10*K10	=-G10+8200*COS(RADIANS(heel))*P4		=M10*N10
27		0	=B17	=C17	0	\	=E18
28		0	0	1	0		
29							
30		Inverse Hydrodynamic Matrix					Solution
31		=MINVERSE(B25:E28)	=MINVERSE(B25:E28)	=MINVERSE(B25:E28)	=MINVERSE(B25:E28)		=MMULT(B31:E34,G25:G28
32		=MINVERSE(B25:E28)	=MINVERSE(B25:E28)	=MINVERSE(B25:E28)	=MINVERSE(B25:E28)		=MMULT(B31:E34,G25:G28
33		=MINVERSE(B25:E28)	=MINVERSE(B25:E28)	=MINVERSE(B25:E28)	=MINVERSE(B25:E28)		=MMULT(B31:E34,G25:G28
34		=MINVERSE(B25:E28)	=MINVERSE(B25:E28)	=MINVERSE(B25:E28)	=MINVERSE(B25:E28)		=MMULT(B31:E34,G25:G28
35							
36							
37							
38							
39	TWA	TWS	AWA	AWS	TBS	Max this var	ABS
40	=TWA	=TWS	=D4	=AWS	=speed/knots_fps*SQRT(lambda)	=ABS(L26)	=ABS
41							
42							
43							
44							
45							
46							
47							
48							
49	run	TWA	TWS	AWA	AWS	TBS	Max this
50	1	40	20	28.7183863211108	23.7709574246089	6.36374299709884	4.51237343489571
51	2	40	20	28.5561560344612	23.8031667171554	6.42561640195212	4.54750651167809
52	3	40	20	28.4208440872336	23.8129145961566	6.39918127398387	4.50379438042871
53	4	40	20	28.9138681203117	23.7448113735215	6.29567623026778	4.47654688470372
54	5	40	20	29.1758162953922	23.5818989045497	6.17173953539458	4.40050521932378
55	6	40	20	28.4047251725347	23.7877411750825	6.45209266107023	4.54992725491974
56	7	40	20	28.2592878229814	23.7751713961476	6.40484354479938	4.48762238089503
57	8	40	20	28.9110361666005	23.679301804494	6.31457362471281	4.49615290451192

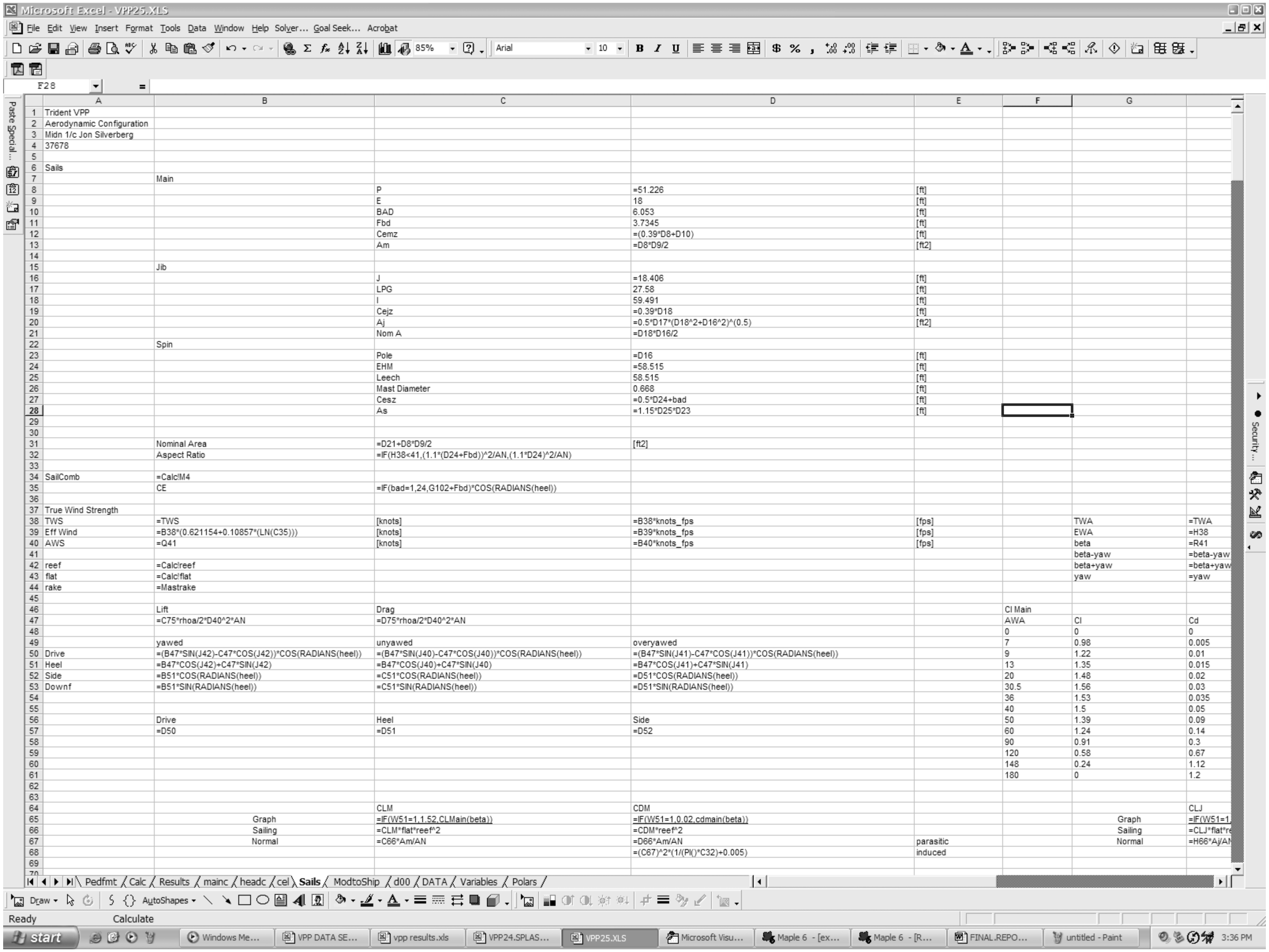


	G	H	I	J	K	L
1						
2	VMG	ABS	SS	Yaw Ang	Rudder Ang	Heel Angle
3	[knots]	[knots]	[knots]	[deg]	[deg]	[deg]
4	=L26	=COS(RADIANS(yaw))*E40	=SIN(RADIANS(yaw))*E40	4.45697851882974	3.2	12.15142651798
5	=G4	=H4*knots_fps/SQRT(lambda)	=I4*knots_fps/SQRT(lambda)	=IF(bad=1,Yawangle,IF(O33+P33<0.05,0.05,O33+P33))	=RudderAngle	=IF(bad=1,HeelAngle,IF(O32+P32>0,C
6						
7						
8	Equation 2: Moment x-direction					
9	Rmx +	(Liftk *	Cek) +	(Liftr *	Cer)	=
10	=IF(bad=1,-8500,rarm(heel))	=liftk(speed,rud,yaw,heel)*lambda^3	=Draftk*1/2*lambda	=liftr(speed,rud,yaw,heel)*lambda^3	=Draftr*1/2*lambda	=
11			=G10-8200*COS(RADIANS(heel))*P4	+	=(H10*I10+J10*K10)	=
12					Difference	=M11-(I11+K11)
13					% Difference	=L12*0.5/MAX(ABS(I11+K11),ABS(M11
14						
15	Equation 4: Moment z-direction					
16	Lr *	Cer	=	(LS *	CESx) +	(Ds *
17	=J10	=3.017*lambda-2.76457	=	=side	=Sails!F102	=Drive
18		=G17*H17	=	=Sails!G103		
19		Difference	0			
20		% Difference	=I19/J18			
21						
22						
23	Aerodynamic Answer			**Vectors**	lateral direc	vmg
24	sails			SideSlip	=SIN(RADIANS(TWA+90))*SS	=COS(RADIANS(TWA+90))*SS
25	{=D10			Fmotion	=SIN(RADIANS(TWA))*ABS	=COS(RADIANS(TWA))*ABS
26	{=M10*N10			Sum	=SUM(K24:K25)	=SUM(L24:L25)
27	{=E18					
28	{					
29						
30	Solution			% Error	Delta Error	
31	{=MMULT(B31:E34,G25:G28)		drag	=ABS(C12)	=J31	
32	{=MMULT(B31:E34,G25:G28)		liftk	=ABS(D20)	=ABS(D20)	
33	{=MMULT(B31:E34,G25:G28)		liftr	0	=ABS(I20)	
34	{=MMULT(B31:E34,G25:G28)		rmmt	=ABS(L13)	=J34*2	Solution?
35				=MAX(J31:J34)		=IF(J35<0.005,TRUE,P15)
36						
37						
38						
39	ABS	Heel	Yaw	Rudder	Flat	Reef
40	=ABS	=heel	=yaw	=rud	=flat	=reef
41						
42						
43						
44						
45						
46						
47						
48						
49	Max this	ABS	Heel	Yaw	Rudder	Flat
50	4.51237343489571	6.34104866585558	10.7311255745718	4.84026520805086	0	0.7
51	4.54750651167809	6.40254078288313	11.3110299342735	4.96187880684989	0	0.74
52	4.50379438042871	6.37216548235978	12.8106990934642	5.26668303067568	0	0.7
53	4.47654688470372	6.27469040426124	9.90456080960406	4.67950789145964	0	0.66
54	4.40050521932378	6.15254691009417	8.47083302676179	4.5197419304246	0	0.7
55	4.54992725491974	6.42599162404727	12.0868446043205	5.15540429267611	0	0.78
56	4.48762238089503	6.3751426616672	13.7125302823895	5.5199611311225	0	0.74
57	4.49615290451192	6.29423498781581	9.24530176086807	4.59984474375258	0	0.74

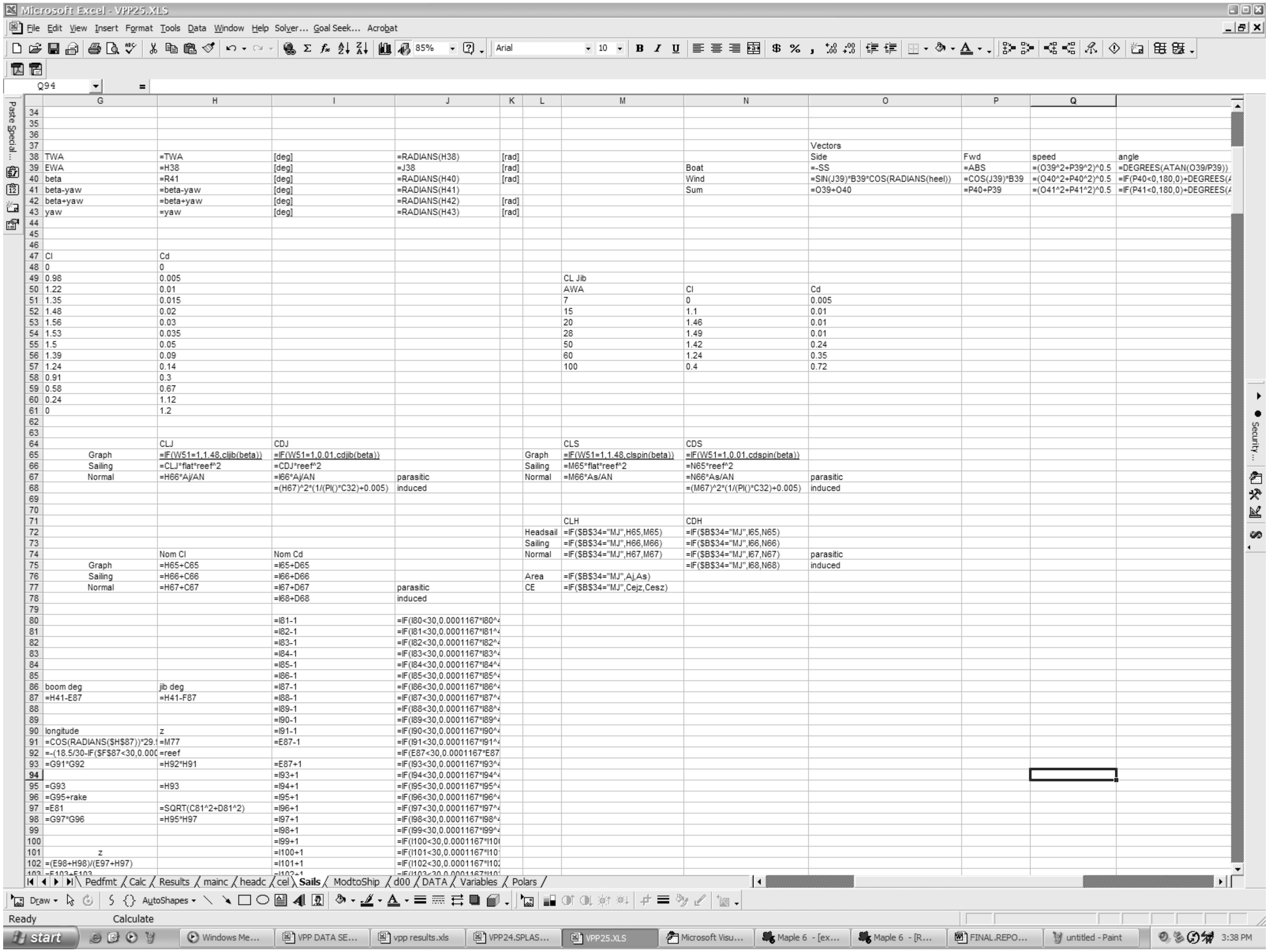


	M	N	O	P	Q
1					
2	SailComb	flat/reef	Mastrake	crew	
3	[text]	[% / %]	[ft of CE]	[% Width]	
4	MJ	0.78	2	=IF(heel<1,heel^4,1)	
5	=M4	0.7	=IF(bad=1,O4,IF(O6=0,P35+O35-40,O6))	=P4	
6					
7				Run	
8				Reset	
9	heelf *	Cesz			
10	=Heelf	=Sails!G102+Fbd-Cgz			
11	=-M10*N10				
12					
13			VPP Variables		
14			Is it bad?	0	
15			Can't Change?	=IF(O33=0.05,IF(D19<0,TRUE))	
16	CESy)		1 / Change	6	
17	=Sails!E102		Boundaries?	=IF(speed<heel ^ 3 * 0.0002854 - 0.01427 * heel ^ 2 + 0.2817 * heel + 1,TRUE,FALSE)	
18			Is there error?	=ISERROR(L35)	
19			Cgz	-0.818	
20					
21					
22					
23	speed	angle			
24	=(L24^2+K24^2)^0.5	=DEGREES(ATAN(L24/K24))			
25	=(K25^2+L25^2)^0.5	=IF(L25<0,180,0)+DEGREES(ATAN(K25/L25)*SIGN(L25))*IF(L25<0,-1,1)			
26	=(K26^2+L26^2)^0.5	=IF(L26<0,180,0)+DEGREES(ATAN(K26/L26)*SIGN(L26))*IF(L26<0,-1,1)			
27					
28					
29					
30			Variable	Delta	
31		Speed	=speed	=IF(L35,0,IF(speed<0.5,0.5,((G31-1)/(P16*2))*O31))*(1-K31)	
32		Heel	=heel	=IF(L35,0,((G34-1)/(P16*2))*O32)*(1-K34)	
33		Yaw	=yaw	=IF(L35,0,(((G32-1)/(P16*2)))*O33)*(1-K32)*SIGN(B17)	
34		Rudder	=IF(rud>5.9,5.9,IF(rud<0.005,0.005,rud))	=IF(L35,0,((G33-1)/(P16*2))*O34*(1-K33))	
35		Mastrake	=IF(Mastrake+40<-1000,1,IF(Mastrake+40>2000,80,Mastrake+40))	=IF(L35,0,((I20)/(P16*2))*O35*(1-K33))	
36					
37					
38					
39	Rake	Heel Moment	Sideforce	Drive	VMG
40	=O5	=-M11	=E17	=D10	=L26
41					
42					
43					
44					
45					
46					
47					
48					
49	Reef	Rake	Heel Moment	Sideforce	Drive
50	0.7	2	27755.3733218393	1283.29893205535	279.633558285775
51	0.7	2	29334.495894998	1353.68842170596	300.532184618443
52	0.74	2	32393.4180370826	1422.59739013457	313.539031391051
53	0.7	2	26200.7271184146	1214.5958994981	259.499654087394
54	0.66	2	23335.1121763934	1137.19373835684	241.195052106885
55	0.7	2	30813.6460483473	1417.93123716359	317.273795670559
56	0.74	2	34072.2851098523	1490.7958880929	329.325157942233
57	0.66	2	24777.3883572074	1204.94925000575	261.54815748363
58	0.7	2	30826.5018608722	1418.56997198194	317.563503289475
59	0	0	0	0	0
60	0	0	0	0	0
61	0	0	0	0	0
62	0	0	0	0	0
63	0	0	0	0	0
64	0	0	0	0	0
65	0	0	0	0	0
66	0	0	0	0	0
67	0	0	0	0	0
68	0	0	0	0	0
69	0	0	0	0	0

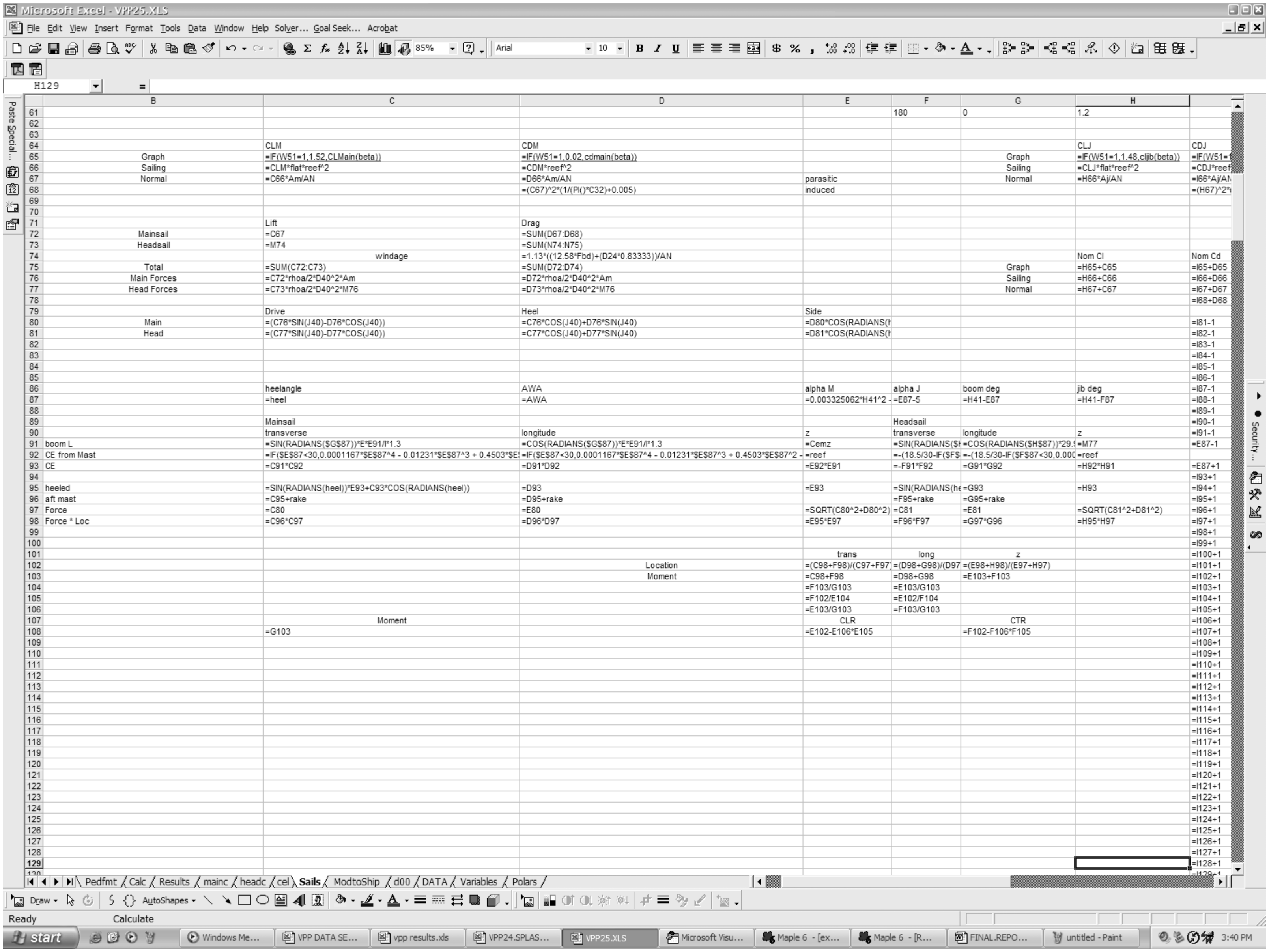
from the sail calculations:



	A	B	C	D	E	F	G	
1	Trident VPP							
2	Aerodynamic Configuration							
3	Midn 1/c Jon Silverberg							
4	37678							
5								
6	Sails							
7		Main						
8			P	=51.226	[ft]			
9			E	18	[ft]			
10			BAD	6.053	[ft]			
11			Fbd	3.7345	[ft]			
12			Cemz	=(0.39*D8+D10)	[ft]			
13			Am	=D8*D9/2	[ft2]			
14								
15		Jib						
16			J	=18.406	[ft]			
17			LPG	27.58	[ft]			
18			I	59.491	[ft]			
19			Cejz	=0.39*D18	[ft]			
20			Aj	=0.5*D17*(D18^2+D16^2)^(0.5)	[ft2]			
21			Nom A	=D18*D16/2				
22		Spin						
23			Pole	=D16	[ft]			
24			EHM	=58.515	[ft]			
25			Leech	58.515	[ft]			
26			Mast Diameter	0.668	[ft]			
27			Cesz	=0.5*D24+bad	[ft]			
28			As	=1.15*D25*D23	[ft]			
29								
30								
31		Nominal Area	=D21+D8*D9/2	[ft2]				
32		Aspect Ratio	=IF(H38<41,(1.1*(D24+Fbd))^2/AN,(1.1*D24)^2/AN)					
33								
34	SailComb	=Calc!M4						
35		CE	=IF(bad=1,24,G102+Fbd)*COS(RADIANS(heel))					
36								
37	True Wind Strength							
38	TWS	=TWS	[knots]	=B38*knots_fps	[fps]		TWA	=TWA
39	Eff Wind	=B38*(0.621154+0.10857*(LN(C35)))	[knots]	=B39*knots_fps	[fps]		EWA	=H38
40	AWS	=Q41	[knots]	=B40*knots_fps	[fps]		beta	=R41
41							beta-yaw	=beta-yaw
42	reef	=Calc!reef					beta+yaw	=beta+yaw
43	flat	=Calc!flat					yaw	=yaw
44	rake	=Mastrake						
45								
46		Lift	Drag			Cl Main		
47		=C75*rhoa/2*D40^2*AN	=D75*rhoa/2*D40^2*AN			AWA	Cl	Cd
48						0	0	0
49		yawed	unyawed	overyawed		7	0.98	0.005
50	Drive	=(B47*SIN(J42)-C47*COS(J42))*COS(RADIANS(heel))	=(B47*SIN(J40)-C47*COS(J40))*COS(RADIANS(heel))	=(B47*SIN(J41)-C47*COS(J41))*COS(RADIANS(heel))		9	1.22	0.01
51	Heel	=B47*COS(J42)+C47*SIN(J42)	=B47*COS(J40)+C47*SIN(J40)	=B47*COS(J41)+C47*SIN(J41)		13	1.35	0.015
52	Side	=B51*COS(RADIANS(heel))	=C51*COS(RADIANS(heel))	=D51*COS(RADIANS(heel))		20	1.48	0.02
53	Downf	=B51*SIN(RADIANS(heel))	=C51*SIN(RADIANS(heel))	=D51*SIN(RADIANS(heel))		30.5	1.56	0.03
54						36	1.53	0.035
55						40	1.5	0.05
56		Drive	Heel	Side		50	1.39	0.09
57		=D50	=D51	=D52		60	1.24	0.14
58						90	0.91	0.3
59						120	0.58	0.67
60						148	0.24	1.12
61						180	0	1.2
62								
63								
64			CLM	CDM				CLJ
65		Graph	=IF(W51=1,1.52,CLMain(beta))	=IF(W51=1,0.02,cdmain(beta))			Graph	=IF(W51=1,
66		Sailing	=CLM*flat*reef^2	=CDM*reef^2			Sailing	=CLJ*flat*re
67		Normal	=C66*Am/AN	=D66*Am/AN	parasitic		Normal	=H66*Aj/AN
68				=(C67)^2*(1/(PI()*C32)+0.005)	induced			
69								



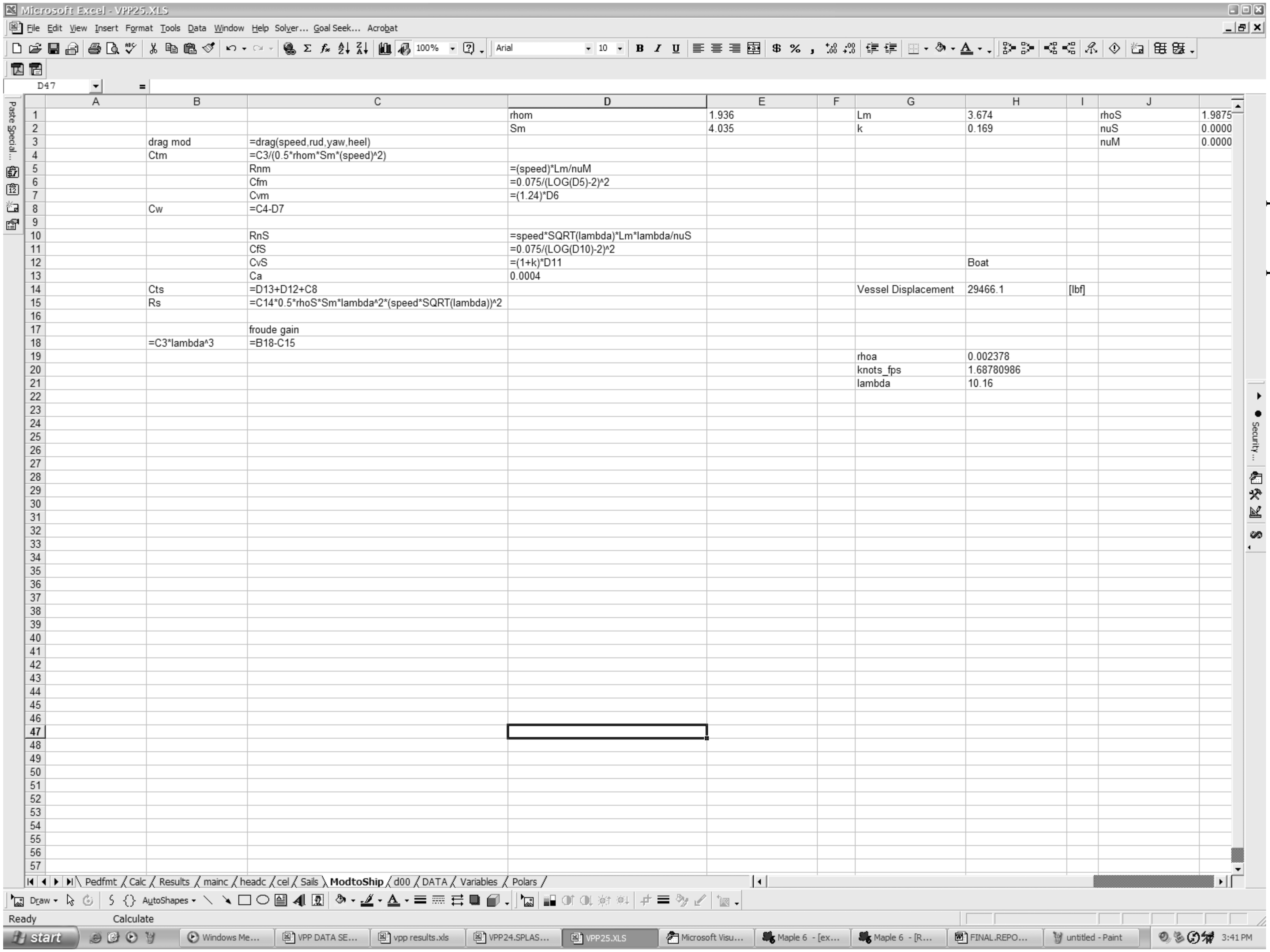
	G	H	I	J	K	L	M	N	O	P	Q	
34												
35												
36												
37									Vectors			
38	TWA	=TWA	[deg]	=RADIANS(H38)	[rad]				Side	Fwd	speed	angle
39	EWA	=H38	[deg]	=J38	[rad]			Boat	=-SS	=ABS	=(O39^2+P39^2)^0.5	=DEGREES(ATAN(O39/P39))
40	beta	=R41	[deg]	=RADIANS(H40)	[rad]			Wind	=SIN(J39)*B39*COS(RADIANS(heel))	=COS(J39)*B39	=(O40^2+P40^2)^0.5	=IF(P40<0,180,0)+DEGREES(
41	beta-yaw	=beta-yaw	[deg]	=RADIANS(H41)				Sum	=O39+O40	=P40+P39	=(O41^2+P41^2)^0.5	=IF(P41<0,180,0)+DEGREES(
42	beta+yaw	=beta+yaw	[deg]	=RADIANS(H42)	[rad]							
43	yaw	=yaw	[deg]	=RADIANS(H43)	[rad]							
44												
45												
46												
47	Cl	Cd										
48	0	0										
49	0.98	0.005					CL Jib					
50	1.22	0.01					AWA	Cl	Cd			
51	1.35	0.015					7	0	0.005			
52	1.48	0.02					15	1.1	0.01			
53	1.56	0.03					20	1.46	0.01			
54	1.53	0.035					28	1.49	0.01			
55	1.5	0.05					50	1.42	0.24			
56	1.39	0.09					60	1.24	0.35			
57	1.24	0.14					100	0.4	0.72			
58	0.91	0.3										
59	0.58	0.67										
60	0.24	1.12										
61	0	1.2										
62												
63												
64		CLJ	CDJ				CLS	CDS				
65	Graph	=IF(W51=1,1.48,cljib(beta))	=IF(W51=1,0.01,cdjib(beta))			Graph	=IF(W51=1,1.48,clspin(beta))	=IF(W51=1,0.01,cdspin(beta))				
66	Sailing	=CLJ*flat*reef^2	=CDJ*reef^2			Sailing	=M65*flat*reef^2	=N65*reef^2				
67	Normal	=H66*Aj/AN	=I66*Aj/AN	parasitic		Normal	=M66*As/AN	=N66*As/AN	parasitic			
68			=(H67)^2*(1/(PI()*C32)+0.005)	induced				=(M67)^2*(1/(PI()*C32)+0.005)	induced			
69												
70												
71							CLH	CDH				
72						Headsail	=IF(B34="MJ",H65,M65)	=IF(B34="MJ",I65,N65)				
73						Sailing	=IF(B34="MJ",H66,M66)	=IF(B34="MJ",I66,N66)				
74		Nom Cl	Nom Cd			Normal	=IF(B34="MJ",H67,M67)	=IF(B34="MJ",I67,N67)	parasitic			
75	Graph	=H65+C65	=I65+D65					=IF(B34="MJ",I68,N68)	induced			
76	Sailing	=H66+C66	=I66+D66			Area	=IF(B34="MJ",Aj,As)					
77	Normal	=H67+C67	=I67+D67	parasitic		CE	=IF(B34="MJ",Cejz,Cesz)					
78			=I68+D68	induced								
79												
80			=I81-1	=IF(I80<30,0.0001167*I80^4								
81			=I82-1	=IF(I81<30,0.0001167*I81^4								
82			=I83-1	=IF(I82<30,0.0001167*I82^4								
83			=I84-1	=IF(I83<30,0.0001167*I83^4								
84			=I85-1	=IF(I84<30,0.0001167*I84^4								
85			=I86-1	=IF(I85<30,0.0001167*I85^4								
86	boom deg	jib deg	=I87-1	=IF(I86<30,0.0001167*I86^4								
87	=H41-E87	=H41-F87	=I88-1	=IF(I87<30,0.0001167*I87^4								
88			=I89-1	=IF(I88<30,0.0001167*I88^4								
89			=I90-1	=IF(I89<30,0.0001167*I89^4								
90	longitude	z	=I91-1	=IF(I90<30,0.0001167*I90^4								
91	=COS(RADIANS(H87))*29.!	=M77	=E87-1	=IF(I91<30,0.0001167*I91^4								
92	=-(18.5/30-IF(F87<30,0.000	=reef		=IF(E87<30,0.0001167*E87								
93	=G91*G92	=H92*H91	=E87+1	=IF(I93<30,0.0001167*I93^4								
94			=I93+1	=IF(I94<30,0.0001167*I94^4								
95	=G93	=H93	=I94+1	=IF(I95<30,0.0001167*I95^4								
96	=G95+rake		=I95+1	=IF(I96<30,0.0001167*I96^4								
97	=E81	=SQRT(C81^2+D81^2)	=I96+1	=IF(I97<30,0.0001167*I97^4								
98	=G97*G96	=H95*H97	=I97+1	=IF(I98<30,0.0001167*I98^4								
99			=I98+1	=IF(I99<30,0.0001167*I99^4								
100			=I99+1	=IF(I100<30,0.0001167*I10								
101	z		=I100+1	=IF(I101<30,0.0001167*I10								
102	=(E98+H98)/(E97+H97)		=I101+1	=IF(I102<30,0.0001167*I10								
103	=E103+F103		=I102+1	=IF(I103<30,0.0001167*I10								


Microsoft Excel - VPP25.XLS
File Edit View Insert Format Tools Data Window Help Solver... Goal Seek... Acrobat
85%
Arial
10
Paste Special...
Security...

H129 =

	B	C	D	E	F	G	H	
61					180	0	1.2	
62								
63								
64		CLM	CDM				CLJ	CDJ
65	Graph	=IF(W51=1,1.52,CLMain(beta))	=IF(W51=1,0.02,cdmain(beta))			Graph	=IF(W51=1,1.48,cljib(beta))	=IF(W51=1
66	Sailing	=CLM*flat*reef^2	=CDM*reef^2			Sailing	=CLJ*flat*reef^2	=CDJ*reef
67	Normal	=C66*Am/AN	=D66*Am/AN	parasitic		Normal	=H66*Aj/AN	=I66*Aj/AN
68			=(C67)^2*(1/(PI()*C32)+0.005)	induced				=(H67)^2*
69								
70								
71		Lift	Drag					
72	Mainsail	=C67	=SUM(D67:D68)					
73	Headsail	=M74	=SUM(N74:N75)					
74		windage	=1.13*((12.58*Fbd)+(D24*0.83333))/AN				Nom Cl	Nom Cd
75	Total	=SUM(C72:C73)	=SUM(D72:D74)			Graph	=H65+C65	=I65+D65
76	Main Forces	=C72*rhoa/2*D40^2*Am	=D72*rhoa/2*D40^2*Am			Sailing	=H66+C66	=I66+D66
77	Head Forces	=C73*rhoa/2*D40^2*M76	=D73*rhoa/2*D40^2*M76			Normal	=H67+C67	=I67+D67
78								=I68+D68
79		Drive	Heel	Side				
80	Main	=(C76*SIN(J40)-D76*COS(J40))	=C76*COS(J40)+D76*SIN(J40)	=D80*COS(RADIANS(h				=I81-1
81	Head	=(C77*SIN(J40)-D77*COS(J40))	=C77*COS(J40)+D77*SIN(J40)	=D81*COS(RADIANS(h				=I82-1
82								=I83-1
83								=I84-1
84								=I85-1
85								=I86-1
86		heelangle	AWA	alpha M	alpha J	boom deg	jib deg	=I87-1
87		=heel	=AWA	=0.003325062*H41^2 -	=E87-5	=H41-E87	=H41-F87	=I88-1
88								=I89-1
89		Mainsail			Headsail			=I90-1
90		transverse	longitude	z	transverse	longitude	z	=I91-1
91	boom L	=SIN(RADIANS(G87))*E*E91/I*1.3	=COS(RADIANS(G87))*E*E91/I*1.3	=Cemz	=SIN(RADIANS($H	=COS(RADIANS(H87))*29.5	=M77	=E87-1
92	CE from Mast	=IF(E87<30,0.0001167*E87^4 - 0.01231*E87^3 + 0.4503*E	=IF(E87<30,0.0001167*E87^4 - 0.01231*E87^3 + 0.4503*E87^2 -	=reef	=-(18.5/30-IF(F	=-(18.5/30-IF(F87<30,0.000	=reef	
93	CE	=C91*C92	=D91*D92	=E92*E91	=-F91*F92	=G91*G92	=H92*H91	=E87+1
94								=I93+1
95	heeled	=SIN(RADIANS(heel))*E93+C93*COS(RADIANS(heel))	=D93	=E93	=SIN(RADIANS(he	=G93	=H93	=I94+1
96	aft mast	=C95+rake	=D95+rake		=F95+rake	=G95+rake		=I95+1
97	Force	=C80	=E80	=SQRT(C80^2+D80^2)	=C81	=E81	=SQRT(C81^2+D81^2)	=I96+1
98	Force * Loc	=C96*C97	=D96*D97	=E95*E97	=F96*F97	=G97*G96	=H95*H97	=I97+1
99								=I98+1
100								=I99+1
101				trans	long	z		=I100+1
102			Location	=(C98+F98)/(C97+F97	=(D98+G98)/(D97	=(E98+H98)/(E97+H97)		=I101+1
103			Moment	=C98+F98	=D98+G98	=E103+F103		=I102+1
104				=F103/G103	=E103/G103			=I103+1
105				=F102/E104	=E102/F104			=I104+1
106				=E103/G103	=F103/G103			=I105+1
107		Moment		CLR		CTR		=I106+1
108		=G103		=E102-E106*E105		=F102-F106*F105		=I107+1
109								=I108+1
110								=I109+1
111								=I110+1
112								=I111+1
113								=I112+1
114								=I113+1
115								=I114+1
116								=I115+1
117								=I116+1
118								=I117+1
119								=I118+1
120								=I119+1
121								=I120+1
122								=I121+1
123								=I122+1
124								=I123+1
125								=I124+1
126								=I125+1
127								=I126+1
128								=I127+1
129								=I128+1

for the model to ship extrapolation:



	A	B	C	D	E	F	G	H	I	J	
1				rhom	1.936		Lm	3.674		rhoS	1.9875
2				Sm	4.035		k	0.169		nuS	0.0000
3		drag mod	=drag(speed,rud,yaw,heel)							nuM	0.0000
4		Ctm	=C3/(0.5*rhom*Sm*(speed)^2)								
5			Rnm	=(speed)*Lm/nuM							
6			Cfm	=0.075/(LOG(D5)-2)^2							
7			Cvm	=(1.24)*D6							
8		Cw	=C4-D7								
9											
10			RnS	=speed*SQRT(lambda)*Lm*lambda/nuS							
11			CfS	=0.075/(LOG(D10)-2)^2							
12			CvS	=(1+k)*D11				Boat			
13			Ca	0.0004							
14		Cts	=D13+D12+C8				Vessel Displacement	29466.1	[lbf]		
15		Rs	=C14*0.5*rhoS*Sm*lambda^2*(speed*SQRT(lambda))^2								
16											
17			froude gain								
18		=C3*lambda^3	=B18-C15								
19							rhoa	0.002378			
20							knots_fps	1.68780986			
21							lambda	10.16			