Department of Aerospace Engineering
Research Project
Date 17/04/2024

DEVELOPING A COMPUTATIONALLY EFFICIENT FLUID AND STRUCTURE MODEL TO PREDICT UPWIND SAILING PERFORMANCE

Matthew R. Bell

Department of Aerospace Engineering, University of Bristol, Queen's Building, University Walk,
Bristol. BS8 1TR. UK.

ABSTRACT

Since the turn of the millennium, high-performance sailing teams have developed sailing simulators, aiming for an improvement in training facilities and rapid prototyping capability. Current simulators use a force-mapping approach, and this paper investigates whether a real-time solution method is a feasible and accurate alternative for upwind sailing, eliminating the need for any CFD. A coupled fluid-structure model was written using Python to communicate between NASTRAN and Athena Vortex Lattice (AVL) and achieves adequate computational speed. This model uses an iterative method where displaced nodal co-ordinates from NASTRAN are interpreted to generate sail camber lines for AVL input, then pressure outputs from AVL are written into the NASTRAN input file and a new sail shape is calculated. Using this iterative method to find a quasi-steady solution, the resulting code can solve an inviscid, incompressible, irrotational VLM combined with a linear FEA model within 0.5s. The simulation takes inputs for all the control lines of an ILCA 7 and for wind variables. Calculated optimal control inputs match conventional sailing advice, but upwind VMG is marginally underestimated at 1.86ms^{-1} in 5ms^{-1} wind speed due to simplified modelling assumptions.

Keywords: Fluid-Structure interaction, Sailing, Simulation, Membrane, Thin aerofoil

INTRODUCTION

As the sport of sailing has grown, so has the importance of understanding how to extract maximum performance from a vessel in varying atmospheric conditions, either through sail and hull design, or through sailing technique. To this end, sailors are now looking for accurate simulations to design, test, and sail models virtually, but how should a sail be modelled?

Sailing boats have two distinct regimes, where the sail aerodynamics are markedly different: upwind sailing (where the boat sails towards where the wind is blowing from) and downwind sailing (sailing in the same direction as the wind). The main difference between upwind and downwind sailing is flow attachment. When sailing upwind and trimmed correctly, the sail is at a low angle of incidence to the airflow, which results in attached flow. However, in the downwind regime, sails have a high angle of incidence, leading to detached flow. Due to the attached flow observed in upwind sailing, a panel method could be used to simulate airflow with a satisfactory level of accuracy.

Since the early 2000s, researchers have used Reynolds-Averaged Navier-Stokes (RANS) simulations, with measured sail shapes modelled as a rigid body for use in a numerical flow simulation [1], but less research has been conducted on the fluid-structure interaction of a flexible mast and membrane structure. Masuyama et al. [2] found the vortex lattice method (VLM) underestimated force coefficients compared to RANS, and both methods underestimated C_L compared to an experimental sail dynamometer.

To enable CFD use, the flying shapes of larger yachts' sails (J80 and Stewart 34) have been measured using the VSPARS method pioneered at the University of Auckland [3]. In 2011, Morris integrated this with a pressure tapping method to measure lift [4]. More recently, unsteady RANS simulations have been compared to experimental measurements on a downwind sail, showing clear agreement between the two methods [5]. These results reinforce the importance of CFD use for the detached flow of downwind sails.

One clear conclusion from the existing research is that an accurate sail model requires both a structural and an aerodynamic model, either through measuring static sail shapes or through the use of FEA software. One example of a linked fluid-structure model is the ARAVANTI method developed by K-Epsilon which uses an inviscid, incompressible, irrotational Vortex Lattice Method (VLM) for the fluid and Finite Element Analysis (FEA) for the structure. Their inviscid VLM model is noticeably faster to solve than the viscous solver [6] but neither can solve real-time problems due to the FEA model, and this precludes its use in a sailing simulator.

Due to the growing interest in sailing simulators such as the ones employed by America's Cup teams [7], a computationally efficient method of predicting forces on a dinghy is required and CFD is simply not suitable for this purpose. Current simulators represent the sail and foils as "force maps, parameterized from CFD" [8] but this paper aims to investigate if an aeroelastic model could be developed which uses a real-time solution method, removing the need for upwind CFD models. For well-funded teams this would reduce virtual prototyping time, and to the average sailor this model could be accessible without requiring any CFD.

In conclusion, there is potential utility in developing an aeroelastic model suitable for use in a computationally efficient sailing simulator and this paper aims to confirm its feasibility and accuracy. Such a model could give racing teams an advantage in rapid prototyping sail design, practising sail trimming, and racing tactics. Inviscid methods exist for simulating airflow over a thin sail structure, but there is currently no simulator which can achieve real-time force predictions based on panel methods and linear structural elements. The aeroelastic model would benefit from modelling all control tensions and unsteady wind effects.

This project aims to create a computationally efficient method for predicting sails' upwind flying shapes and performance with sufficient accuracy to allow potential simulator use.

To improve the utility of such a simulator, the model shall include all the relevant control inputs that control a sail's shape. This simulator shall have scope to include these controls, including sheeting, kicker, cunningham, mast ram, spreaders and stay tensions. A simple sailing dinghy shall be modelled in this way to verify the simulation's accuracy and computational speed.

This paper presents how this model was created, its modelling assumptions and limitations, and draws conclusions about using this method for accurate sail modelling.

Sailing Terms Glossary

Sailing precedes recorded history, such that sailors have their own language. This glossary is included as reference for any terminology used in this paper.

For a visual representation of the calculations below using wind and boat velocity vectors, see Figure 2.

True Wind: the wind vector a stationary observer will experience above the water's surface.

TWS (True Wind Speed): the wind speed a stationary observer will experience.

TWA (True Wind Angle): the angle between the centreline of the boat and the wind direction when the boat is stationary.

Apparent Wind: the wind vector a boat in motion will experience. This is equal to the difference between the true wind vector ($\overrightarrow{TW}$) and the boat velocity vector ($\overrightarrow{V_b}$). This vector can be expressed as a speed **AWS** (Apparent Wind Speed) and an angle **AWA** (Apparent Wind Angle), relative to the boat's centreline.

$$\overrightarrow{AW} = \overrightarrow{TW} - \overrightarrow{V_b} \tag{1}$$

AWS (Apparent Wind Speed): the magnitude of the **Apparent Wind** vector ($\overrightarrow{AW}$).

AWA (Apparent Wind Angle): the angle between the **Apparent Wind** vector ($\overrightarrow{AW}$) and the boat's centreline.

VMG (Velocity Made Good): the speed at which the boat is travelling towards its destination. In this paper, the destination is always directly upwind. When sailing upwind, the sailboat cannot sail directly upwind, so progress is measured by resolving the boat velocity vector.

$$VMG = boat\ speed\ \cdot \cos(True\ Wind\ Angle) \tag{2}$$

Cringle: The name given to a hole in the sail through which a rope is passed (used for control lines and hoisting/reefing lines), usually strengthened by a metal ring or grommet.

Tack: the name for the bottom front corner of the sail

Clew: the aft corner of the sail.

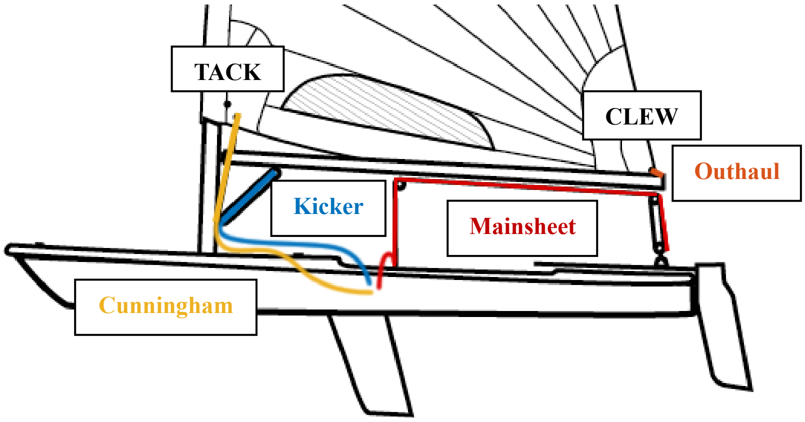

Figure 1: Sail control lines.

Figure 1 shows the control lines modelled in this study. All controls are assumed to be under tension. The free body diagrams below show how the wind vector, boat velocity vector, and mainsheet angle, resolve to give sail AoA and V_∞.

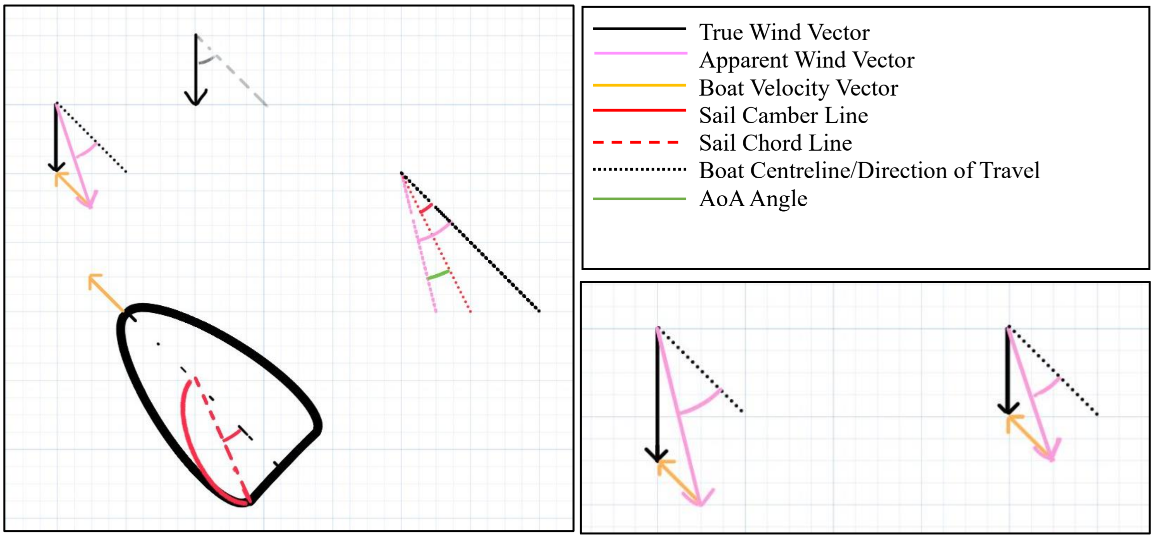

Figure 2 (Left): Showing how to resolve the True Wind vector and Boat Velocity vector to find the Apparent Wind vector. Then how to find the sail AoA using the mainsheet angle and AWA.
Figure 3 (Right): Showing the difference in Apparent Wind vector at the sail root (right) and sail tip (left) due to the faster True Wind Speed.

METHODOLOGY

STRUCTURAL MODEL

In order to build up a robust aeroelastic model of a sail, the sail's membrane structure must first be accurately represented in a finite element (FE) software. This section describes the FE modelling process.

Choice of Modelling Software and Solution Method

Due to its documentation, and ability to communicate with Python using the pyNastran module [9], MSC NASTRAN was chosen as the finite element tool. The ILCA 7 dinghy sail (chosen for this project) is a non-linear structure with zero compressive strength because it is made of woven polyethylene, so a non-linear solution within NASTRAN should be preferred. The non-linearity of a sail only becomes important when modelling sail flapping or buckling behaviour. For this initial project no manoeuvres are modelled, thus the sail would not flap and modelling non-linear behaviour is not necessary. Speed is also preferred in this project as we are evaluating the effectiveness of fast-solving methods for a real-time simulation, and this leads to the choice of a linear solution (Sol 101) because this is both the quickest and manoeuvres are not modelled.

The aerodynamic forces on the sail can be found by interrogating the model's constraints. Total pressure load is the sum of the forces at the constraints minus the forces applied by the controls. Forward speed is then calculated using force equilibrium with hull drag values for similar boats [10]. Hull drag follows a square law:

$$Thrust = Drag = kV_b{}^2 \tag{3}$$

where k is a constant found by substituting measured values for $Drag$ and V_b (for this model, 18lbf of drag at 5kts boat speed from *Higher Performance Sailing*, pg 525 [10]).

The boat is assumed to travel in a straight line along its centreline, and not slip downwind (zero lee slip assumption), and this will increase sail AoA and thus the model's ability to point upwind.

Choice of Sail

The ILCA 7 dinghy (International Laser Class Association, formerly Laser) was chosen as the initial test subject. Due its popularity [11], use in high-level events such as the Olympics, and the large amount of available data regarding its performance, the ILCA 7 is ideal for validating the model's output.

Sail CAD Modelling Challenges

Modelling the ILCA 7 sail (or any sail-shaped membrane) using CAD software is difficult, as the sail shape depends on the sewing pattern used to create it.

The ILCA 7 sail is constructed from planar triangular and quadrilateral sections, sewn together to create a sail which cannot lie flat on one plane. This cannot be accurately modelled in CAD software as a single membrane because internal tensions that arise from joining these planar sections will not be captured.

Technical data available on the ILCA website [12] gave constraints for the sail's three edge lengths, but not regarding the variation of chord shape and internal tensions along the span of the sail.

The initial model was created in CAD to match images and the measurement data from the class regulations, thus it does not accurately represent a sewn membrane, because no data were available regarding the sewing pattern. As a result, the sail is modelled as an isotropic polyester sheet in this initial prototype, although in reality it is a woven polyester fabric. The figure below gives a visual indication of how the model's deformation matches real world flying shapes of the ILCA 7 sail:

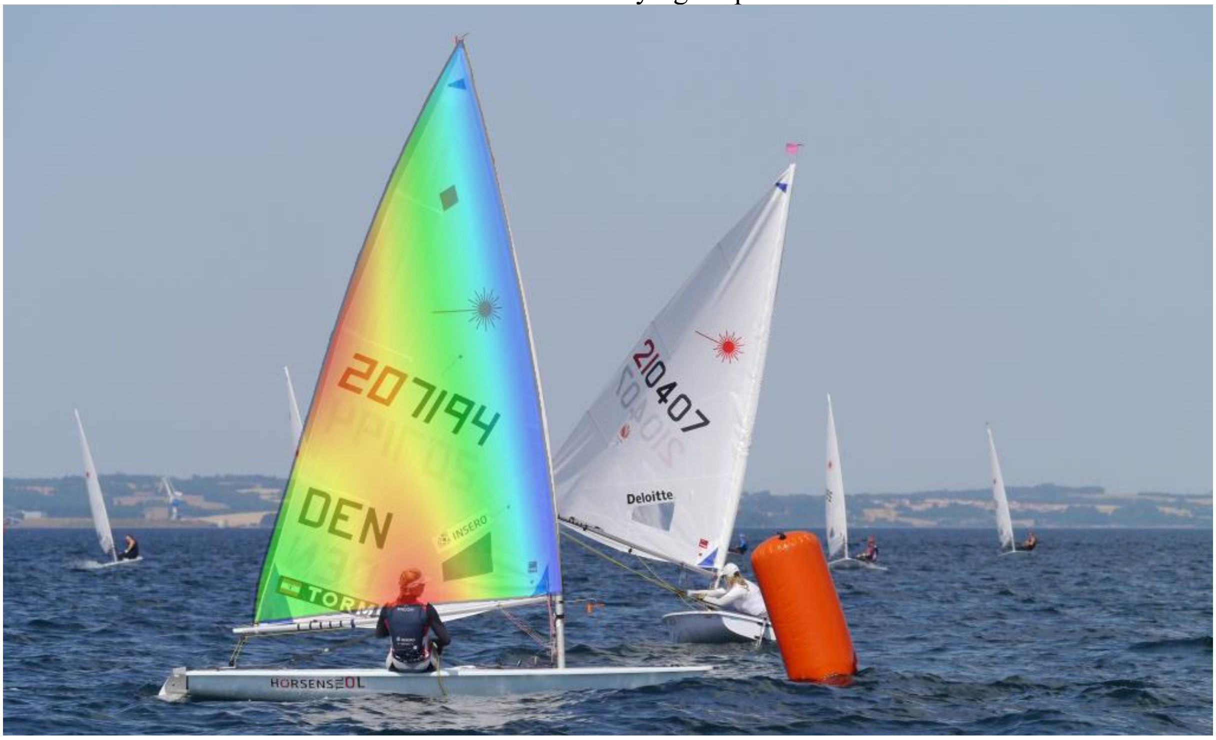

Figure 4: Comparison of model solution shape and real-world sail shape. Image taken from ILCA website [13].

In future study, the sail's finite element model could be constructed the same way the real sail is constructed (as a collection of individual membrane elements) by modelling each individual section.

Structural Meshing and Convergence Study

Having modelled the sail in Autodesk Inventor, the sail CAD model is discretised into CTRIAR and CQUADR elements, made up of 913 nodes and 910 elements. These elements assume plane stress, and zero strain through the element thickness. The nodes on the leading edge are also joined together with idealised CBAR elements, matching mast cross sectional properties found in the ILCA 7 regulations [14], to model the mast bend and stiffness, which affects sail shape. The CBAR element is a one-dimensional bending element assumed to be centred on the shear centre of a prism.

Mesh convergence was assessed by applying a uniform pressure to an unstructured fine mesh in NASTRAN to find the most accurate solution possible with this tool and coarsening the mesh until nodal displacement no longer matched the fine solution (within ±95%). Nodes were equally spaced, and no local refinement was applied to the mesh.

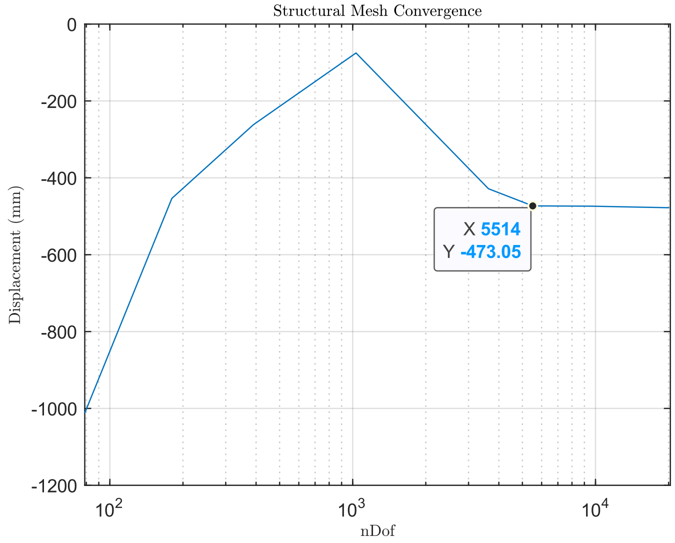

Figure 5: FE model mesh convergence study. 5514 nDof leads to a target element edge length of 75mm.

The spike in this convergence study graph is unusual and is a result of constraining the single node at the Clew vertex, rather than a group of nodes (to distribute the force applied). This is corrected in the final model, shown in Figure 8.

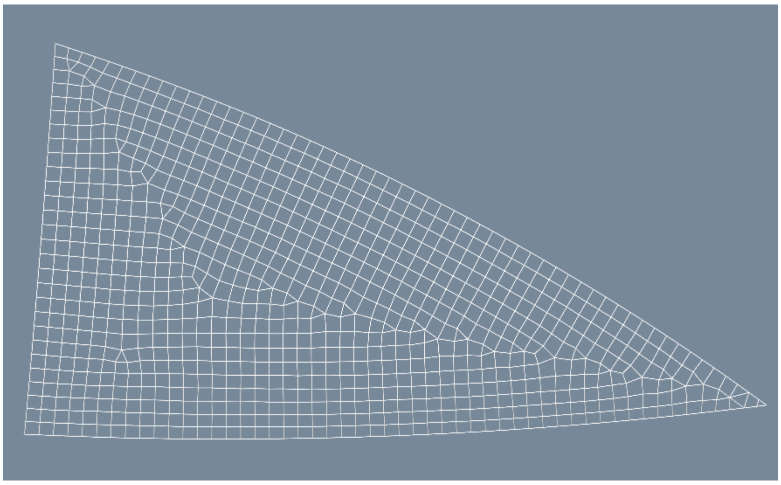

Figure 6: Sail meshing after convergence study.

Finite-Element Control Tensions

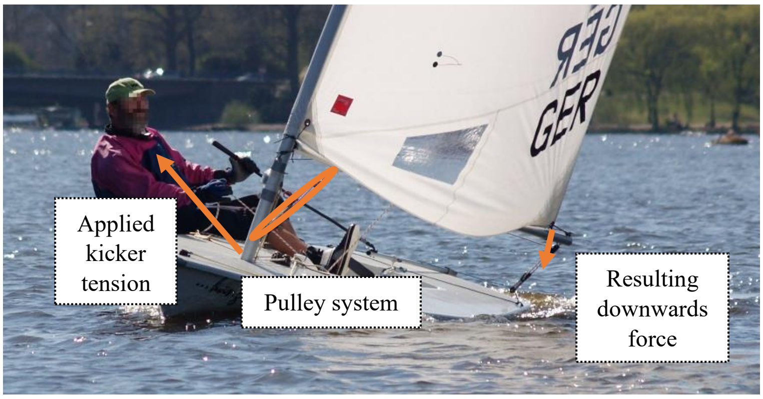

Figure 7: Kicker pulley system and moment arm. [15]

In the final simulator, the user must be able to control the sail by changing the forces applied to the FE model.

The kicker modelled uses a 12:1 pulley system to multiply the tension, is at a 45° angle to the horizontal, and is applied approximately $^1/_5$ of the way along the boom, as in Figure 7. The combination of all these force multiplications means that the downward force applied to the clew is approximately 3.5 times the tension applied by the sailor.

The kicker pulley system is not directly connected to the sail, so the kicker force is transmitted along the boom and goes through the outhaul line. Consequently, this force is applied to the FE model's clew nodes.

Similarly, the cunningham is a 2:1 pulley system and, due to the very small angle of application, is modelled as a purely vertical tension applied to the tack cringle.

The mainsheet is modelled as an angle from the boat centreline, rather than a control tension, to explicitly control angle of attack. The ILCA 7 mainsheet does contribute to kicker force but, to clearly assess each control separately, this captured within the kicker input variable. The outhaul is modelled as a horizontal force towards the back of the boat, thus simplifying the code and making a clear distinction between the vertical (kicker) and horizontal (outhaul) forces applied to the clew nodes.

Three nodes are used to model the control tensions at to the clew, as the forces are distributed by a cringle, and this corrects the mesh convergence issues in Figure 5.

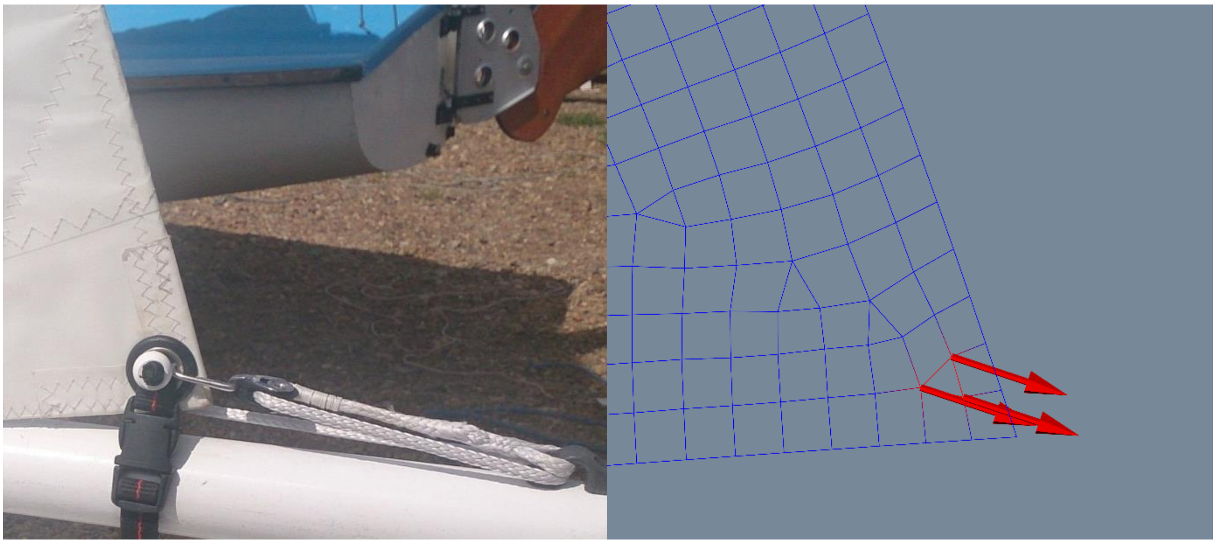

Figure 8: Comparison of clew cringle and applied forces [16]. Note that the applied forces are a combination of kicker and outhaul forces.

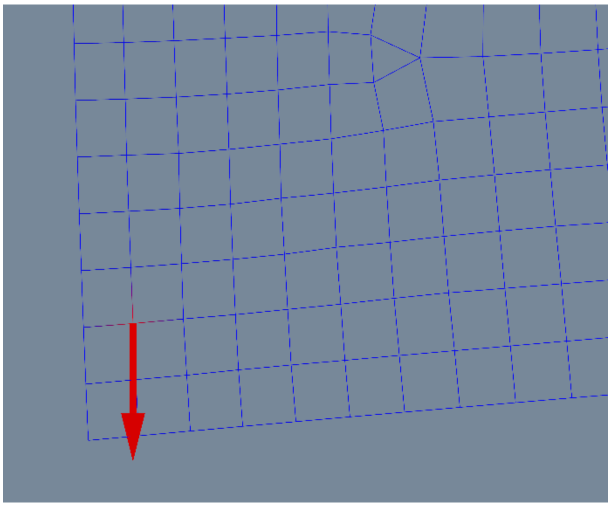

Figure 9: Cunningham forces applied to the tack. Only one node is used due to the coarseness of the mesh in this area.

AERODYNAMIC MODEL

A structural model is only half the picture. While the control tensions are modelled in finite element software, their effect on airflow must also be understood. This section describes how an aerodynamic model was chosen and the limitations imposed on this model as a result.

Choice of Aerodynamic Method

A sail is a thin structure approximated as zero thickness, and the slow wind speeds involved in sailing allow an incompressibility assumption ($M \approx 0$). The high Reynolds number, ($\approx 1 \times 10^7$), resulting from the large sail chord and low velocity, implies complete turbulence which allows for an inviscid flow assumption. These assumptions allow the use of a vortex lattice method.

AVL

Multiple vortex lattice methods could be used, but AVL was chosen for several reasons. AVL (Athena Vortex Lattice) was written by Harold Youngren in 1988 and uses a vortex lattice method to analytically solve simplified Navier-Stokes equations. The AVL documentation states “surfaces and their trailing wakes are represented as single-layer vortex sheets, discretised into horseshoe vortex filaments” [17], thus it matches our assumptions and is ideal for a thin wing. AVL constructs a wing by placing aerofoil section shapes along the span, and linearly interpolating between them. Section shapes are generated from input co-ordinate files, and only the camber line is retained due to the thin wing assumption.

AVL then takes numerous inputs to describe the flow condition and results are saved in a text file, showing differential coefficient of pressure at each control point.

Aerodynamic Panelling and Convergence Study

Confidence in the aerodynamic model is guaranteed by carrying out a convergence study to ensure the model is sufficiently fine to capture all the important flow phenomena. This convergence study is presented below.

Each aerofoil section (in between two defined chord shapes) can have a different number of spanwise panels. For simplicity in modelling, the number of spanwise panels was set to be equal for all sections. This results in a local refinement wherever chord sections are closer together.

Aerodynamic convergence was assessed by taking a sail shape output and calculated flow condition from NASTRAN, applying a fine vortex lattice and adjusting reducing the number of panels (spanwise and chordwise separately) in AVL until the lift distribution no longer matched the fine mesh.

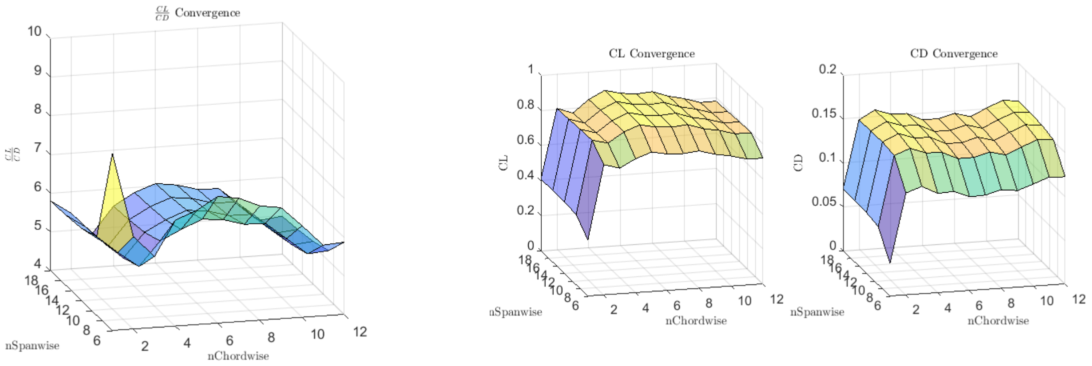

Figure 10: Surface plots of C_L, C_D and $\frac{C_L}{C_D}$ variation with panelling.

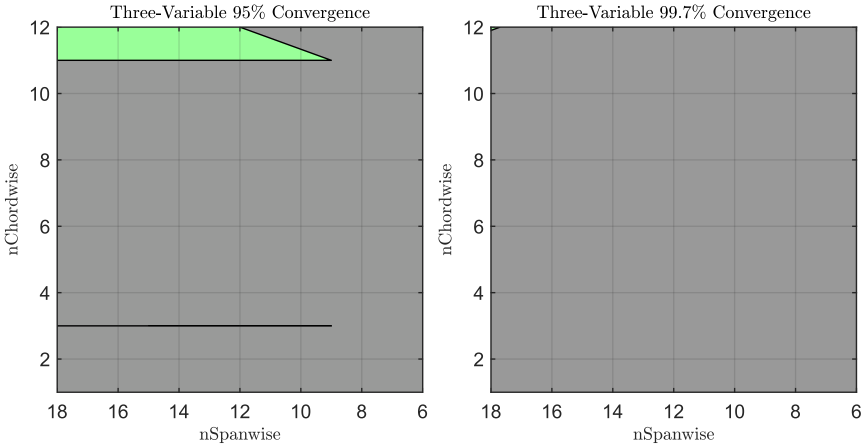

Figure 11: Contour plots to show where the solutions for C_L, C_D and $\frac{C_L}{C_D}$ are all within 95% or 99.7% of the over-fitted case. Green regions show converged panelling, grey is not converged.

It is unknown why there appears to be convergence when nChordwise = 3, so the higher value of nChordwise 11 is used to ensure confidence in the model.

The above results lead to a final panelling of 11 chordwise panels and 9 spanwise panels:

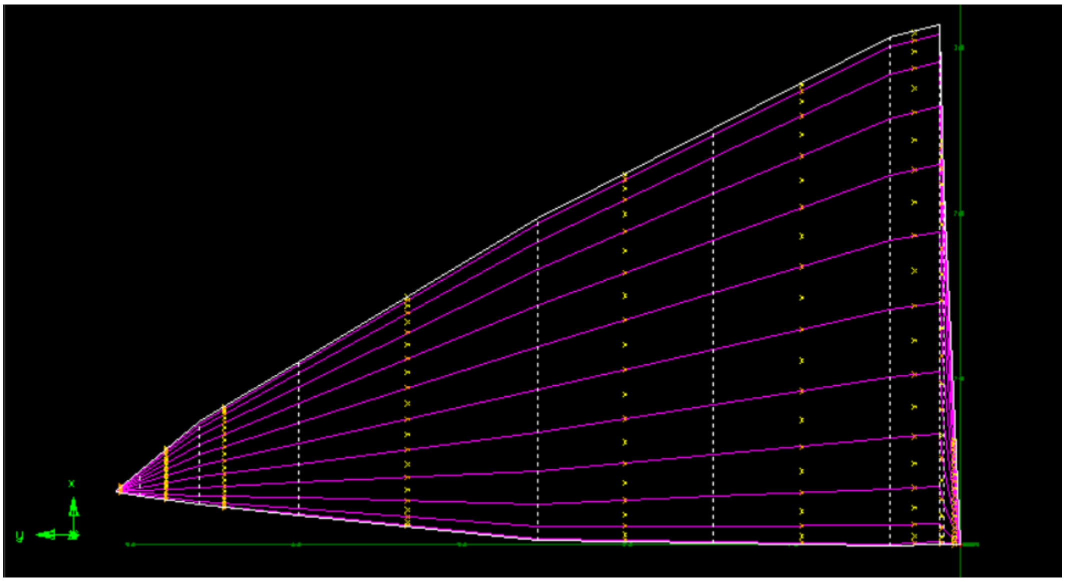

Figure 12: AVL panelling after convergence study. Note the highly skewed panels at the root are inevitable due to the inability to change chordwise panelling distribution for individual sections. Control points are marked in orange.

Wind Shear Layer and Apparent Wind Speed

Wind phenomena near the water's surface affect angles of attack and wind speed. These phenomena cannot be modelled easily in AVL, so the model performs some calculations to ensure these effects are captured.

Due to viscosity, the wind speed near to the ground is much slower than the wind speed higher in the atmosphere, much like the boundary layer over a surface. The following data, from Higher Performance Sailing by Frank Bethwhaite [10], shows how wind speed varies with height above water, for multiple prevailing wind speeds. For code simplicity the wind shear layer is modelled as a linear gradient with wind speed doubling from sail root to sail tip.

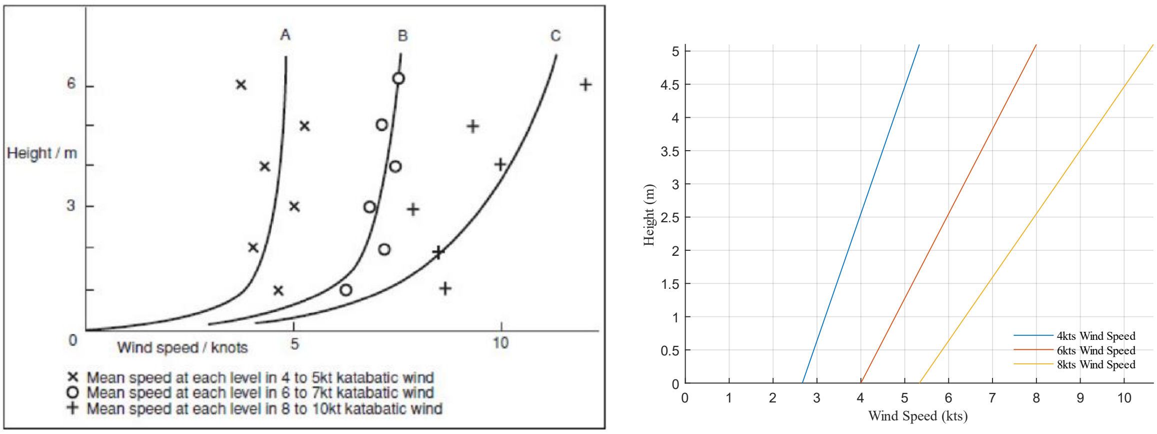

Figure 13: Left: Measured wind shear layer velocity gradients. From "Higher Performance Sailing", pg 277 [10]. Right: Linear approximation used in this model.

Another important wind phenomenon in sailing is that of "Apparent Wind Speed". The forward motion of a sailboat introduces an equal and opposite headwind. The freestream velocity which the sail experiences is then a sum of the headwind vector and the true wind vector. The combination of these two effects leads to a higher wind speed and smaller angle of attack. A twisted sail can generate more forward velocity, as the upper portion of sail (the leech) has a forward-pointing lift vector due to the higher true wind speed. The difference in Apparent Wind at the root and tip is shown in Figure 3.

AVL cannot model a non-uniform freestream velocity, but this must be accounted for to accurately represent the wind shear layer. As a result, the code uses yaw rate in AVL to increase tip wind speed, and the model also introduces a sail twist to model the AoA increase due to wind shear layer.

The aerofoil's angle of attack is calculated with respect to the apparent wind vector. The model assumes a constant TWA with height, so the only change is the TWS, increasing with height. This wind speed difference results in a different angle of attack at the root and at the tip. By separately calculating the AoA for the root and for the tip, the sail shape can be twisted to achieve the same effect in uniform airflow. This is input into AVL as an aerofoil twist. Importantly, this is not a "real" structural twist, but a modelling feature which gives the model the capability to represent an AoA change due to the Wind Shear Layer.

This twist accounts for the change in AoA resulting from the wind shear layer but does not account for the increased wind speed seen at the sail tip. A yaw rate must therefore be implemented in AVL, where the tip moves forward through the air and the root recedes at a constant rate. This leads to faster tip airspeed and lower root airspeed. Due to the assumption that tip wind speed is double the root wind speed, a simple rotational velocity calculation can show that the yaw rate about the sail midpoint (r) is equal to

$$r = \frac{1}{3} \times \frac{2V_\infty}{b} \tag{4}$$

where V_∞ is the freestream velocity and b is the wingspan. Please see the derivation and diagram overleaf:

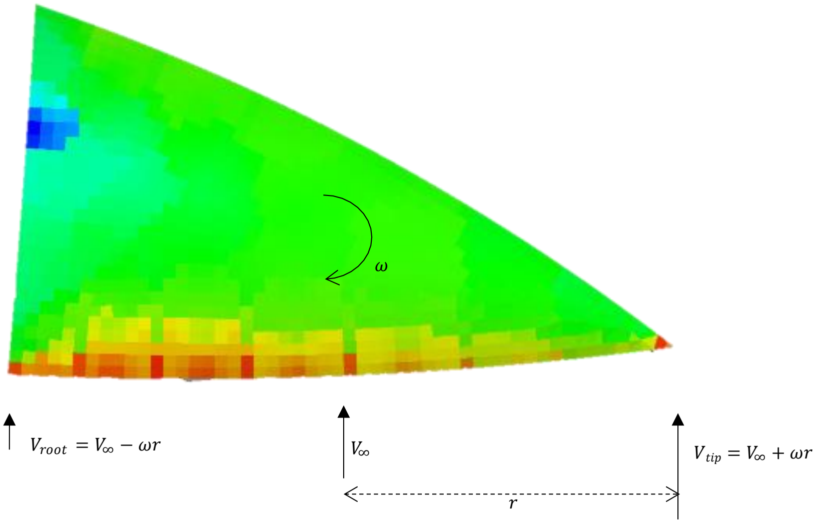
ω
$V_{root} = V_\infty - \omega r$
V_∞
$V_{tip} = V_\infty + \omega r$
r

Figure 14: Use of yaw rate to introduce an approximation to the wind shear layer.

The approximation used in our code is that $V_{tip} = 2V_{root}$. Simple substitution and re-arranging lead to this expression for ω:

$$\omega = \frac{V_\infty}{3r} \tag{5}$$

Where r is half the sail span, or $\frac{b}{2}$, and V_∞ is the apparent wind speed. AVL uses an input format of:

$$Yaw\ Rate\ = \omega \frac{b}{2V_\infty} \tag{6}$$

Combining the two equations gives a simple input value of $Yaw\ Rate\ = \ {}^{1}/_{3}$. The velocity distribution given by this approximation is plotted in Figure 13 and compared to measured wind velocity gradients.

PYTHON CODE

Having established both the structural and aerodynamic models, these two models must communicate with each other in order to model their interactions. A Python code was written to achieve this, and this section explores its algorithm, assumptions, and overall performance.

Algorithm

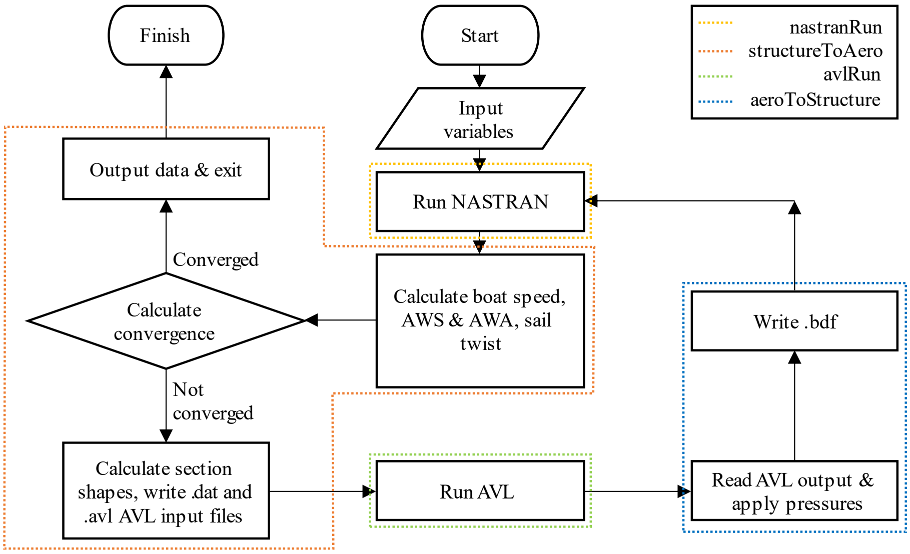

Figure 15: Flowchart of code structure, separated by function.

The code has a modular structure, consisting of four functions, each called by one overhead program. In order:

1. nastranRun calls MSC NASTRAN to execute a .bdf file with some input commands to control memory usage. Nastran writes the solution to a .op2 output file.
2. structureToAero reads the .op2 Nastran output and checks convergence using nodal displacement. It then converts the structure from xyz co-ordinates into an AVL-readable format by generating camber lines (.dat files) and calculates AWA and AWS which is required for AVL inputs. A .avl input file is written for use by AVL.
3. avlRun writes the input string to be used by AVL (controlling inputs for Yaw Rate, V_{∞} and AoA, then opens and runs this string in AVL, which uses the .avl and .dat files written previously.
4. aeroToStructure reads the .txt AVL output to build up a pressure matrix in the same co-ordinate system as Nastran, then calculates the pressure at the centre point of each element in the FE model and writes these to a .bdf file.

The model is built on an iterative framework, where the code will check after each loop if the current solution is different to the previous solution. Once the solution stops changing, a steady state solution has been found and the code stops iterating.

Unfortunately, there is no opportunity to use parallel processing, as each of these processes is dependent on the one before it – this is a feature of the quasi-steady model and not a result of the code. This makes the code highly dependent on processor speed.

Linking the Structure to the Aerodynamics

An initial FEA model of the sail is used to begin the iterative method. The first step is to interpret the sail shape in order to communicate with AVL. The AVL sail shape is generated from deformed node co-ordinates output from NASTRAN using the following method.

Section shapes (aerodynamic chord) are generated by defining a plane and defining line segments between the nodes above and below the plane. Section co-ordinates are then the points where these lines intersect the plane. Section planes and an example of the resulting section camberlines is below. Note the refinement at the sail root, where the sail's chord length is rapidly increasing.

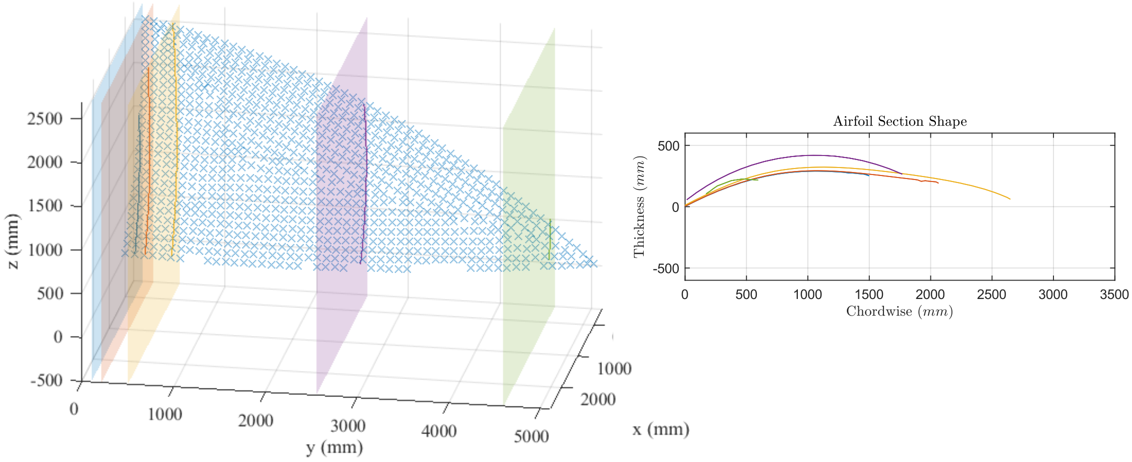

Figure 16: Planes for generating section shapes and the resulting camberlines.

The forward speed of the boat is calculated using its hull drag (Equation 3), which gives the vector for apparent wind speed, and the sail twist is calculated using the wind shear layer approximation. These vectors are then resolved into the frame of reference of the sail, which then gives the angle of attack, yaw rate and freestream velocity inputs for AVL.

Interpreting the Aerodynamic Forces

Pressures are output into a text file from the AVL model. These pressures must be interpreted to apply them to the structural sail model. To convert from dCp (differential pressure coefficient), as given by AVL for a thin surface, to net pressure difference, P, across the sail membrane, dCp is multiplied by the dynamic pressure ($\frac{1}{2}\rho V_\infty{}^2$):

$$P = \frac{1}{2}\rho V_\infty{}^2 \cdot dCp \tag{7}$$

AVL only gives the dCp at each control point. The pressure at the centre of each structural element is therefore a weighted average of the pressure at the two closest control points, and this builds up a pressure distribution across the whole sail. In Figure 17, the dCp would be equal to:

$$dCp\ at\ element\ centre = \frac{dist_a \times dCp_a + dist_b \times dCp_b}{dist_a + dist_b} \tag{8}$$

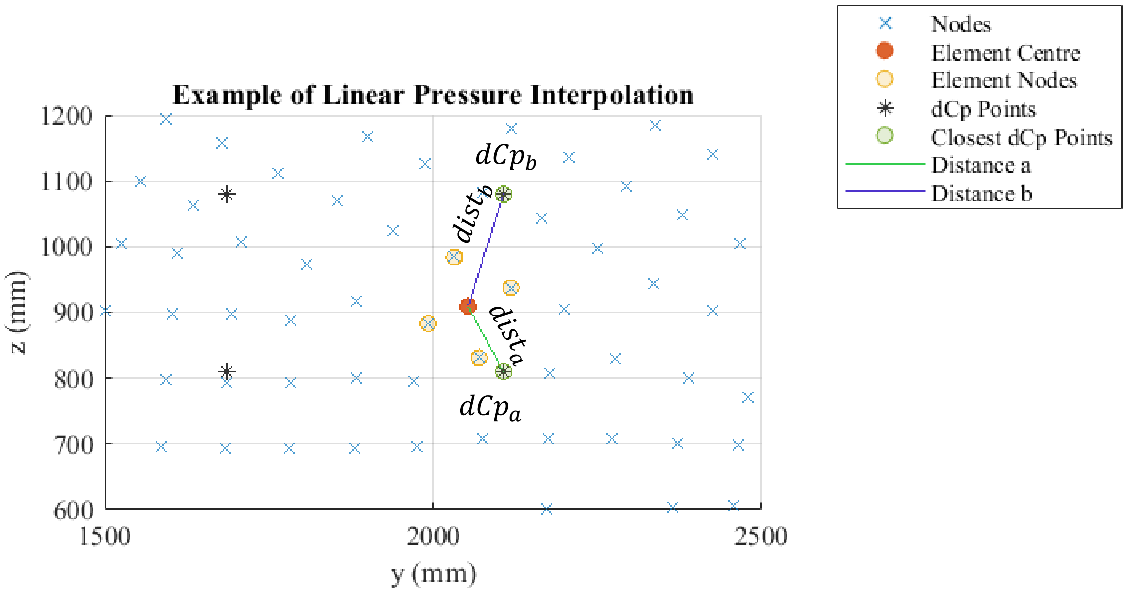

Figure 17: Example of how dCp is linearly interpolated to find the pressure at an element centre.

The use of membrane elements in NASTRAN means the pressure only needs to be applied on one side of the surface, to generate a pressure difference across the membrane. Figure 18 shows the side view of this weighted average method applied to generate a pressure distribution. Due to co-ordinate axes, "negative" pressures point downwind and are thrust generating.

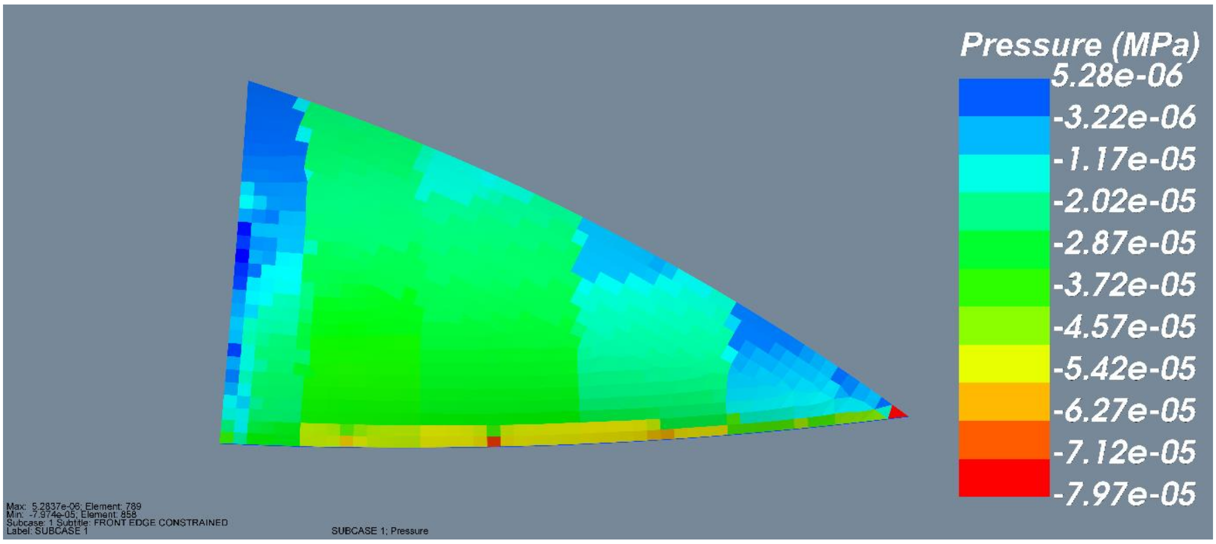

Figure 18: Example pressure distribution from a steady state solution. Note the clear pressure peak at the leading edge. The colour bar shows higher magnitudes as red.

An improvement to this method, with minimal performance impact, would be to use a weighted average of 3 or more points and this would avoid the strips that emerge in the pressure distribution of Figure 18.

Code Performance

If the aim is to use this code in real-time simulations, it must be capable of reaching solutions multiple times each second.

Figure 19 below shows how the code converges to a steady state solution (for the optimum VMG inputs in Figure 24). The code's metric used to assess convergence is the RMS nodal displacement of a sample of 50 nodes across the sail. Displacement is measured with respect to the previous displacement values (such that a displacement of 0mm means a node is in the same position as the previous iteration), and the solution is considered “converged” when sail nodes move less than 10mm from the previous iteration. Figure 24 shows that VMG changes very little once RMS displacement is below 10mm (iteration 3), so this is the chosen convergence criterion.

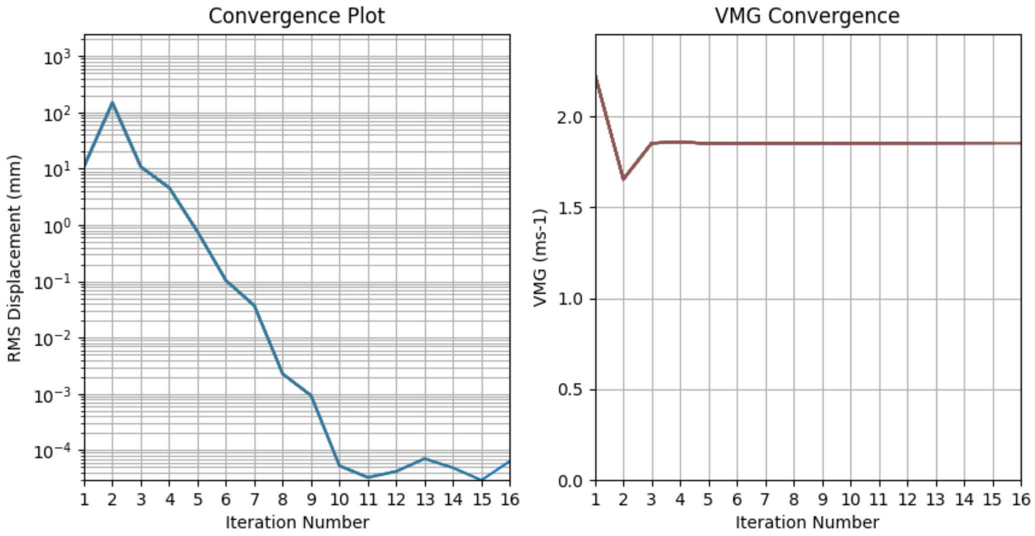

Figure 19: Convergence plots for displacement and Velocity Made Good as the code progresses.

Figure 20 shows the code’s average performance over a test of 1024 iterations. Each function was timed separately, and their average performance is shown below as both a percentage and a time. This was performed on an AMD Ryzen 5 3500U machine with 16GB of 2400MHz memory.

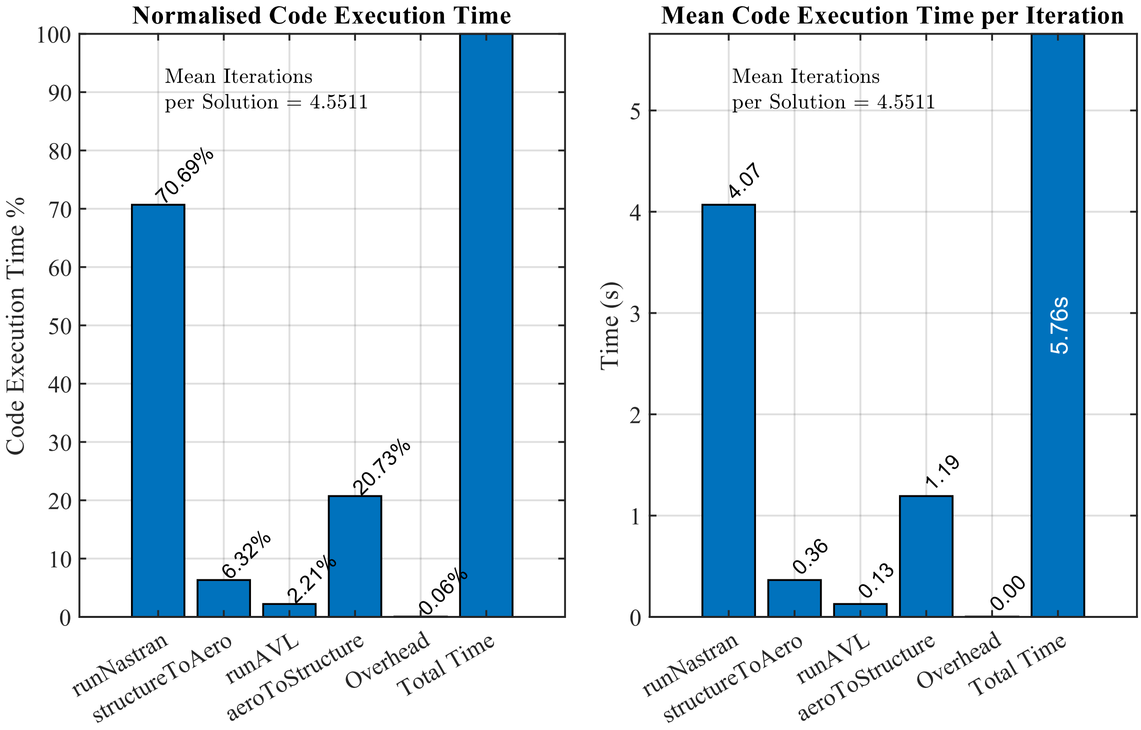

Figure 20: Code speed statistics, separated by function for n = 1024 iterations

The AMD Ryzen 3500U processor uses “Zen” architecture, which achieves 16 FLOPs/cycle for single-point accuracy [18], and runs at approximately 3GHz during these simulations.

$$Total\ FLOPs\ =\ 3GHz \times 4\ cores \times 16\ FLOPs = 192\ GFLOPs \tag{9}$$

Using the same equation, a current high-end workstation processor (AMD Threadripper 3990X) can achieve over 8,000 GFLOPs. This is an oversimplified comparison and underestimates the performance of both processors, but suggests a high-end workstation computer could find a steady-state solution as quickly as 2 times per second, which makes real-time simulation difficult.

VALIDATION OF SOFTWARE OUTPUT

Validity of Solutions

To validate the model's solutions, the boat speed and VMG values will be compared to measured data from real world sailing situations and current simulators in use. Trends can also be compared to conventional sailing advice: if the control adjustments match the expected trends, then the model is behaving as expected. For example, we expect the mainsheet and the kicker to be released as the boat bears away (turns away from the wind) to achieve maximum speed. The results of these validation studies are presented in the following section.

RESULTS

Test 1: Individual Control Variations

This test was designed to understand how each individual control input would affect the sail shape and lift generated. A single base input case was chosen, and inputs for this base case are shown in Table 1:

Table 1: Base Case Input Variables

TWA	45°
TWS	$5ms^{-1}$
Mainsheet	12°
Kicker	1440N
Cunningham	480N
Outhaul	510N

Each of these variables (except for the wind direction and speed) were then varied individually to see how they affect the boat velocity if all other variables are kept the same.

We expect the mainsheet variable to have a clear optimum point, and a high kicker and outhaul tension to generate more lift by reducing structural sail twist at the tip.

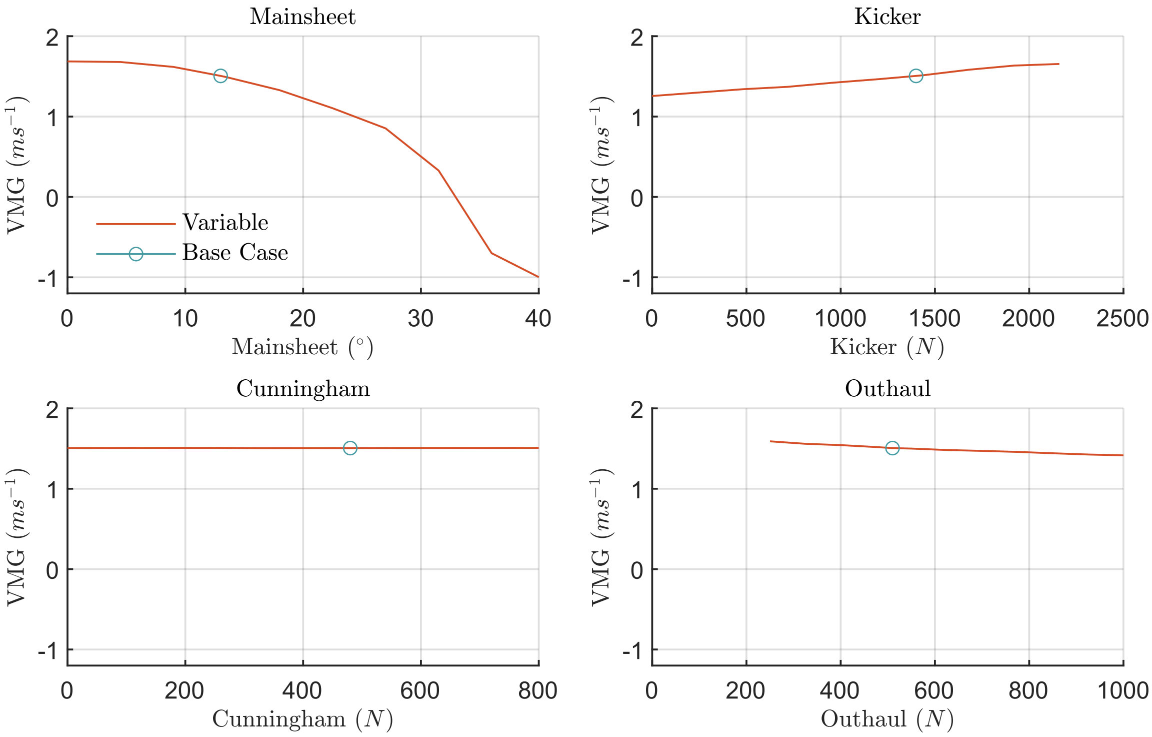

Figure 21: Independent Control Line Effects

These results suggest that changing the cunningham does not significantly affect the lift generated. Kicker should be maximised to maximise lift, at least for this case. One can also see that easing the mainsheet too much leads to unrealistic solutions. Easing the mainsheet too far leads to "negative" values of lift, pushing the boat backwards. This situation is unrealistic because the sail would begin to flap and this behaviour is not modelled. Low outhaul values do not give a converged solutions because outhaul tension is required to supress mechanisms in the FEA model, otherwise the sail is free to "fold" us towards the mast, leading to unrealistic deformations.

Test 2: Control Interactions

To assess how the control lines interact with other inputs, several bi-variate analyses were undertaken. By keeping all variables the same, and varying just two at the same time, the VMG trends for a number of different variable pairs are shown in the figures below. Wind direction was also assessed here, and we should see the wind direction changing how effective the different controls are. Empty points on the graph (white space) indicate un-converged cases, where the model failed to converge to a steady state solution, either due to rigid body modes inherent to membrane models, or due to unrealistic sail shapes which could not be loaded into AVL. All other control variables matched the base case in Table 1.

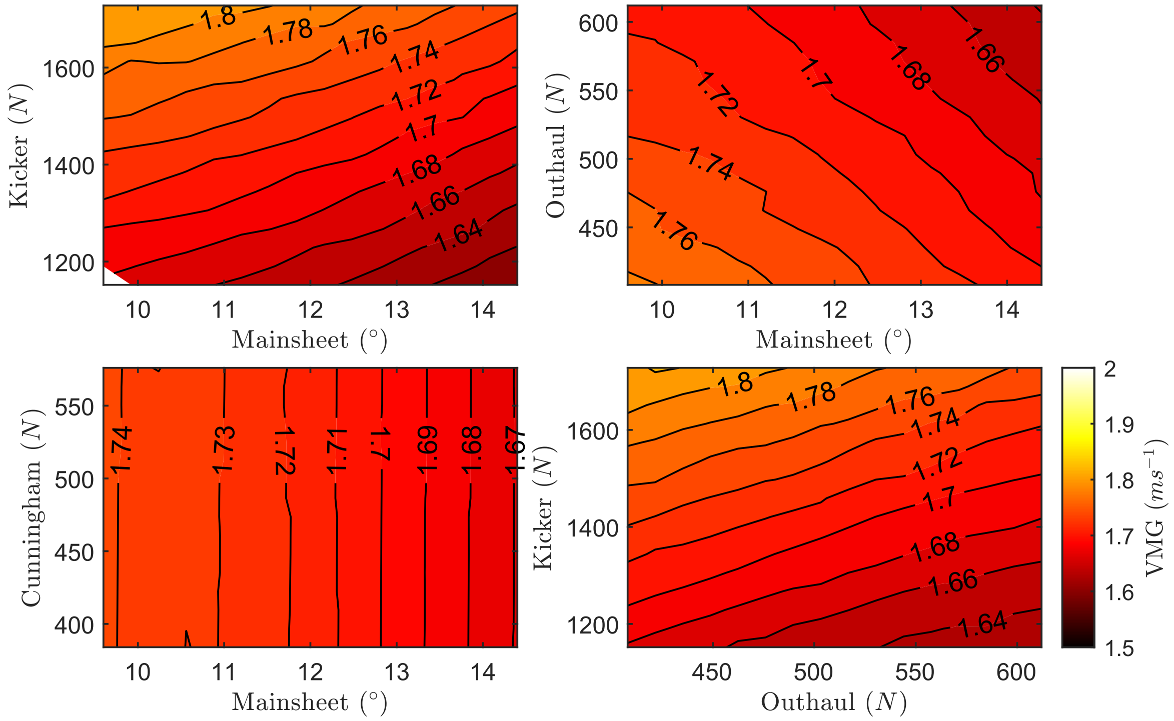

Figure 22: Interactions when varying two input control variables.

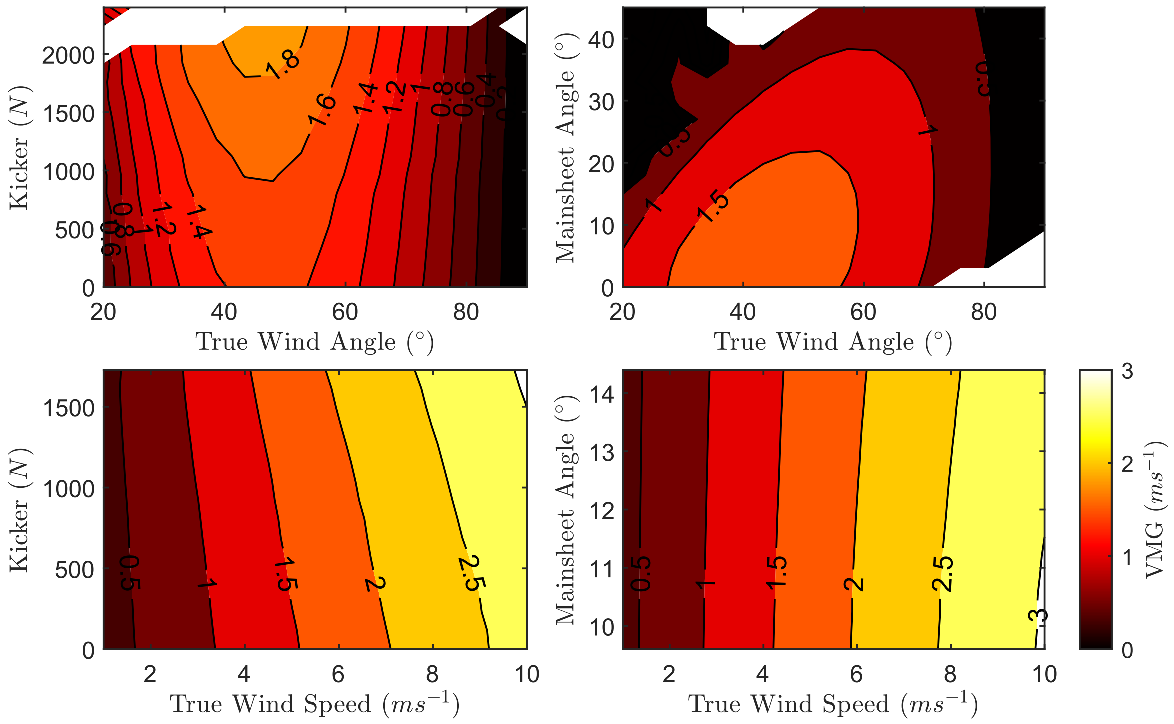

Figure 23: Interactions between wind variables and control inputs.

Some interesting behaviours emerge in these analyses. In Figure 23 top left we can see that, for most cases, a kicker tension greater than 2kN causes a breakdown in solution convergence.
Figure 22 bottom right shows that increasing kicker increases VMG, but outhaul should be minimised. Similarly, one would expect mainsheet to have a big impact on forward speed, and Figure 22 top right agrees while also suggesting outhaul tension is similarly important for the mainsheet angles shown.

Test 3: Optimisation

In order to compare the model's behaviour with conventional sailing advice, the model's speed must first be maximised, to understand if the inputs for maximum performance in the simulation match real world inputs for maximum performance.
A gradient-based optimisation algorithm was used to assess how to maximise this model's upwind performance across the entire regime. For a fixed TWA and TWS (5ms^{-1}), this algorithm finds the local gradient of VMG with respect to the input value ($\frac{VMG}{input\ value}$) for each individual control (mainsheet, kicker, cunningham and outhaul), then increases/decreases the inputs for all four control inputs simultaneously, up the local gradient towards the higher VMG value. If this value is an improvement, the algorithm repeats. If this new input is lower than the previous optimum, the algorithm regresses towards the previous best case until a new optimum is reached. This algorithm then repeats until a local maximum is reached where the $\frac{VMG}{input\ value}$ gradient is zero for all controls. Once a maximum is found for the given TWA, the algorithm begins looking to optimise the sail for a new value of TWA. This builds trends for optimal boat control across the whole upwind regime. TWS is constant throughout.
This algorithm can converge at local maxima, hence the roughness of the curves in Figure 24.

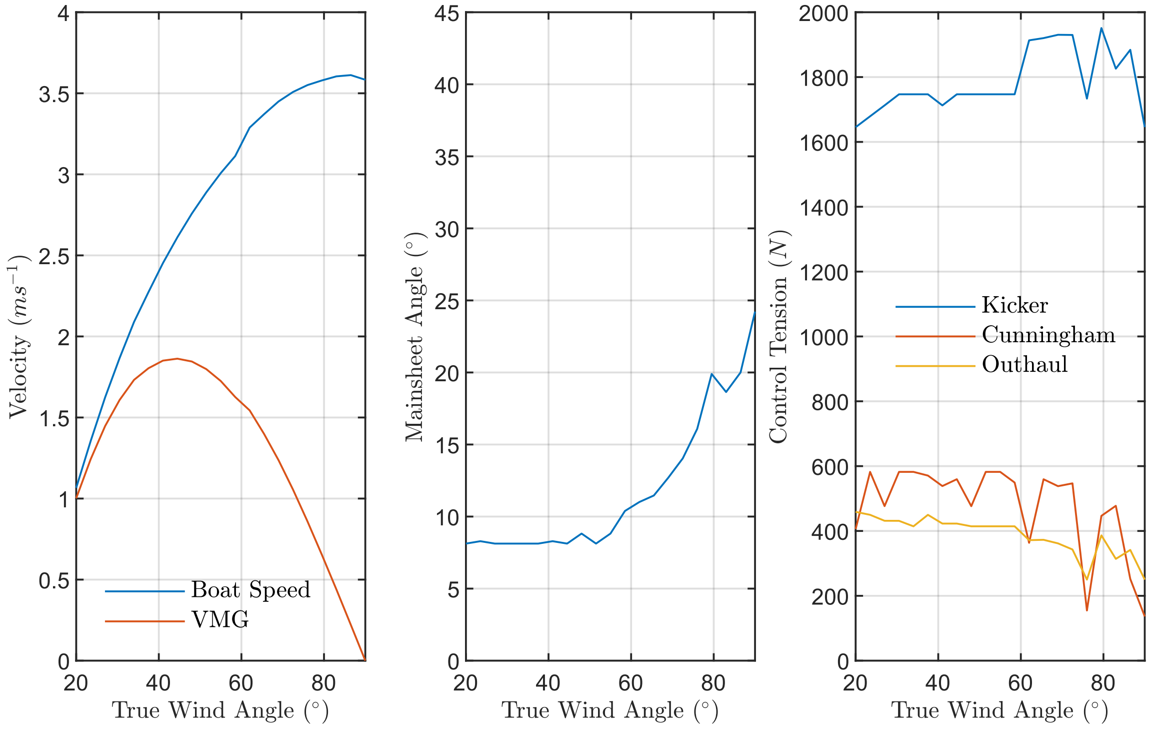

Figure 24: Optimisation results, showing the optimal control inputs for a range of TWA at 5ms^{-1} TWS.

Figure 24 has an anomalous result at 79.5˚ TWA that does not match the trend. Optimal VMG of 1.86ms^{-1} occurs at 44.5˚ TWA, which results in an AWA of 32.6˚.
The resulting sail shape for optimum VMG is shown in Figure 25

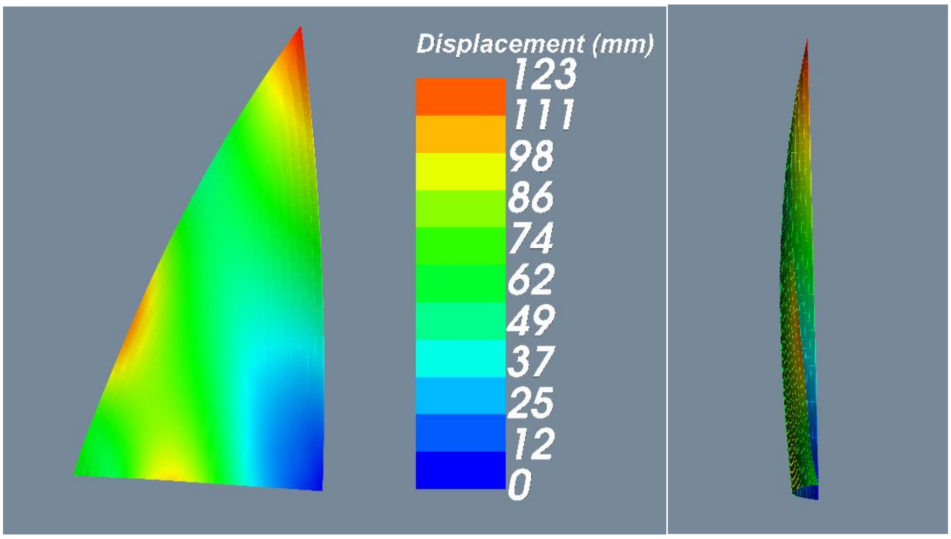

Figure 25: Sail deformation at optimum VMG from Test 3. The optimisation algorithm has exploited high kicker tension to stiffen the sail leech and generate maximum lift higher on the sail.

DISCUSSION

Overall, this model matches real-world trends. Test 1 shows how an un-optimised upwind sail can be improved. The results suggest increasing kicker tension and sheeting in the mainsail, consistent with general sailing knowledge. However, the outhaul trend seems contrary to conventional advice. Due to the inviscid AVL assumption, releasing outhaul to increase sail camber is ideal since stalling is not modelled, but in the real world this could cause stalling or sail flapping if the outhaul is eased.

The model's optimum AWA from Test 3 (32.6°) matches the real-world trend for optimum AWA. According to the research "Optimum runs … lie between 30° and 37° AWA [19]". The general trend for mainsheet is as expected: easing the mainsheet as the boat turns away from the wind. Conventional advice for adjusting kicker is to release the kicker as the boat turns away from the wind, but this is not seen in the model. A possible explanation is due to the inviscid assumption AVL uses. The kicker increases curvature at the sail tip, and this could cause turbulent separation and stall at high AoA, but since viscosity is not modelled the sail does not stall at the tip.

The model marginally underestimates absolute values for VMG at 1.86ms^{-1} when compared to other sailing simulators at 1.88ms^{-1} [20] and allows the boat to point higher into the wind than real world conditions, due to the assumptions of zero lee slip and the inviscid assumption in AVL. The model is expected to overestimate VMG compared to other simulators due to the zero lee slip assumption. Once the code is improved, the VMG is expected to reduce further.

The trends from Test 2 and Test 3 mostly follow conventional sailing advice, with the exception of kicker at high TWA. We expect the kicker to be eased as the boat bears off, but the optimisation algorithm did not capture this behaviour, nor did the bi-variate analysis in Test 2. The inviscid assumption in AVL is the suspected cause of this behaviour as discussed earlier, combined with unrealistic deformations due to high kicker and outhaul tensions. Unconverged regions in Figure 23 are caused by high kicker tension. As the kicker tension increases, the mast bends more up to a point where AVL cannot model the aerofoil. This implies that kicker tension is important in preventing sail flapping and maintaining the linear model's validity.

The Python code is stable across the upwind regime, unless a very high mainsheet angle or kicker tension is input. Unconverged solutions are most often a result of high structural deformations which cannot be modelled in AVL, or high mainsheet angles which violate the linear structural assumption used in NASTRAN. If these conditions are avoided, a solution can be found in 0.5s on high end computers.

CONCLUSION

Conclusions from Comparative Study & Validation

This tool calculates outputs that generally follow trends seen in conventional sailing advice and prior research but underestimates absolute values of boat speed. This could be used for simulation, and could be used for learning how to sail since the model follows general sailing trends, but may not give accurate results to assess different sail designs because viscosity is not modelled. Overall, this model gives sufficient accuracy to be used as an initial test for rapid prototyping and eliminates the need for expensive and time consuming CFD analyses at this initial stage. Accuracy could be improved by modelling lee slip and using a viscous correction to the AVL output (using a tool such as xFoil), and limiting outhaul and kicker tensions to ensure realistic sail shapes are generated.

Due to the code execution time, this model cannot be used for real-time simulation on a low-end machine, but could run near 2Hz in a higher specification computer.

Opportunities for Further Study

The biggest opportunity to speed this solution up is to do away with MSC NASTRAN, because it instantiates the model every loop to change the pressures and re-writes the global stiffness matrix each iteration. A purpose-built FE code could remove the need to re-write the stiffness matrix each loop, and would just apply the forces to find nodal displacement and greatly increase the code's speed.

More complex dinghy designs could be explored, such as a 49er or RS400, which use planing hulls and have more controls available for setup (including spreaders, stays and mast rams) - not to mention two sails - to see if this tool is scalable for other designs.

Because this tool only simulates steady cases, wind gusts and wave interactions are not captured. These are important phenomena in high performance sailing (such as Olympic events), thus modelling unsteady aeroelastic simulations would be a huge improvement. This could potentially be achieved with minor adjustments to the code, and relaxing the requirements for convergence to a quasi-steady solution.

Code Implementation in Industry

Due to the speed of the machine used to compute these initial tests, and the linear nature of the algorithm used, a faster processor would greatly decrease the time needed per solution, as low as 0.5s. This, combined with reducing the FE model solution time, could make this model viable for use in a real-time simulation software.

REFERENCES

1. *Numerical Simulation using RANS-based Tools for America's Cup Design.* **Cowles, Geoff, Parolini, Nicola and Sawley, Mark L.** Annapolis, Maryland, USA : s.n., 2003. SNAME 16th Chesapeake Sailing Yacht Symposium.
2. *Database of sail shapes versus sail performance and validation of numerical calculations for the upwind condition.* **Masuyama, Yutaka, et al.** 1, 2009, Journal of Marine Science Technology, Vol. 14, pp. 137-160.
3. *V-SPARS: A COMBINED SAIL AND RIG SHAPE RECOGNITION SYSTEM.* **Le Pelley, D.J. and Modral, O.** Auckland, New Zealand : s.n., 2008. 3rd High Performance Yacht Design Conference.
4. **Morris, Dale.** *Derivation of Forces on a Sail using Pressure and Shape Measurements at Full-Scale.* Department of Shipping and Marine Technology, CHALMERS UNIVERSITY OF TECHNOLOGY. Göteborg, Sweden : s.n., 2011. Master's Thesis.
5. *Numerical and Experimental Comparison of Spinnaker Aerodynamics Close to Curling.* **Augier, Benoît, et al.** 1, 2021, Journal of Sailing Technology, Vol. 6, pp. 118-132.
6. *SAIL TRIMMING FSI SIMULATION - COMPARISON OF VISCOUS AND INVISCID FLOW MODELS TO OPTIMISE UPWIND SAILS TRIM.* **Sacher, M, et al.** Auckland : s.n., 2015. 5th High Performance Yacht Design Conference.
7. **Ineos Britannia.** Simulating the Cup. [Online] 2023. [Cited: 24 October 2023.] https://www.ineosbritannia.com/en/articles/466_Bold-Protocol-Simulating-the-Cup.html.
8. **Canopy Simulations.** The 35th America's Cup. *Canopy Simulations.* [Online] Canopy Simulations. [Cited: 5 April 2024.] https://canopysimulations.com/sailing/35th-americas-cup/.
9. **Doyle, Steve.** SteveDoyle2/pyNastran. *GitHub.* [Online] 25 March 2024. [Cited: 3 April 2024.] https://github.com/SteveDoyle2/pyNastran.
10. **Bethwaite, Frank.** *Higher Performance Sailing.* s.l. : Bloomsbury Publishing Plc, 2008. 9781408101261.
11. **International Laser Class Association.** About The Boat. *International Laser Class Association.* [Online] [Cited: 3 April 2024.] https://www.laserinternational.org/about-the-laser/the-ilca-formula/.
12. Measurement Diagrams. *International Laser Class Association.* [Online] [Cited: 3 April 2024.] https://www.laserinternational.org/rules-and-technical/laser-class-rules/measurement-diagrams/.
13. Upwind in Lasers. *International Laser Class Association.* [Online] 1 October 2018. [Cited: 3 April 2024.] https://www.laserinternational.org/blog/2018/10/01/upwind-in-lasers/.
14. ILCA Class Rules. *International Laser Class Association.* [Online] 21 August 2023. [Cited: 3 April 2024.] https://www.laserinternational.org/wp-content/uploads/2023/09/ILCA_Class_Rules_2023.pdf.
15. **Andreae, Johann-Nikolaus.** *flickr.* [Online] 14 April 2007. [Cited: 5 April 2024.] https://www.flickr.com/photo_zoom.gne?id=479723212&size=l.
16. **Voodoo 158546.** Outhaul Configuration. *Sailing Forums.* [Online] 16 August 2013. [Cited: 3 April 2024.] https://sailingforums.com/threads/outhaul-configuration.29207/.
17. **Drela, Mark and Youngren, Harold.** AVL 3.40 User Primer. [Online] 22 February 2022. [Cited: 27 March 2024.] https://web.mit.edu/drela/Public/web/avl/avl_doc.txt.
18. **WikiChip.** Floating-Point Operations Per Second (FLOPS) . *WikiChip.* [Online] 25 January 2023. [Cited: 5 April 2024.] https://en.wikichip.org/wiki/flops.
19. **Motta, Dario.** *An experimental investigation of full-scale sail aerodynamics using pressures, shapes and forces.* Department of Mechanical Engineering, The University of Auckland. Auckland, New Zealand : s.n., 2015. PhD Thesis.
20. *Performance profile for ILCA class elite sailors. Differences between men and women.* **Aarón Manzanares, Alberto Encarnación-Martínez, Ion Chicoy-García, Francisco Segado.** 4, 2023, Arch M ed Deporte, Vol. 40, pp. 194-199.