Vortex Lattice Methods

To show that the values from this simulation are trustworthy, some of the parameters will be varied, until changing the parameter has minimal effect on the values of interest. At this point, the mesh will be converged. For all of these tests, α was changed from -5 to 10, the Reynolds number was set to 4.5×10^7 and the Mach number was set to 0.8.

The first parameter that will be changed is N_W, the number of streamwise elements. It will be started at 20 and moved up in increments of 10 until a converged value is found. The results are visible in Figure 1.
50 streamwise elements were decided on as the final number of panels for the minor speed improvement. Increasing the number of elements should increase the accuracy of the results, because more detail on the wing can be resolved. It comes at the cost of longer computational times.

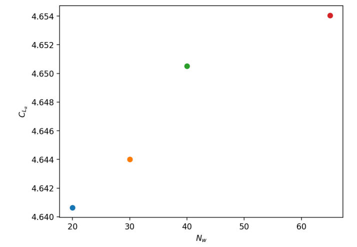


Figure 1: effect of N_w on $\mathrm{C}_{\mathrm{L}_\alpha}$

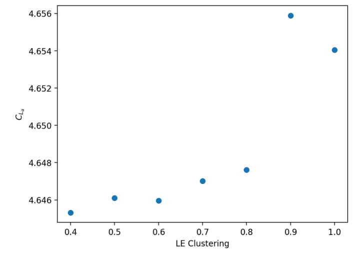


Figure 2: effect of leading edge clustering on $\mathrm{C}_{\mathrm{L}_\alpha}$

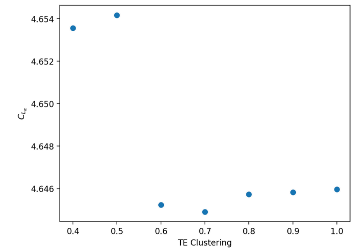


Figure 3: effect of trailing edge clustering on $\mathrm{C}_{\mathrm{L}_\alpha}$

Once a number of spanwise elements was found, leading edge clustering was used. It should be changed, because decreasing the clustering increases the number of panels at the front of the airfoil. The front of the airfoil is where the largest aerodynamic gradients are, and they need to be adequately resolved for good accuracy.
The leading edge clustering started off at 1, and decreased in steps of 0.1.

As is obvious from Figure 2, leading edge clustering has a minimal, but measurable, effect on the lift curve slope. In actual values, this was the variation between 4.654 and 4.645. When rounding the lift curve slope to 3 significant figures, this is no change.
The values of $\mathrm{C}_{\mathrm{L}_\alpha}$ seem to be more stable at lower values of LE clustering, so 0.6 was chosen to proceed with testing trailing edge clustering.

Testing the effects of trailing edge clustering took the same methods as testing leading edge clustering. The same rationale applied here as to the leading edge clustering.
This seems more disordered, visible in Figure 3. However, the only value of TE Clustering that would change anything would be 0.7. It has a value of 4.6449, so is only marginally different to the previous value of 4.65 for the lift curve slope.
To keep everything simple, the chosen value of TE clustering will be 0.6.

Streamwise panels	50
LE Clustering	0.6
TE Clustering	0.6
VLM lift curve slope	4.65

Panel Methods

For panel methods, the same range of tests will be run.

While testing the convergence for panel methods, decreasing the number of CPU cores and turning off hyperthreading increased stability. This obviously came at the cost of increased computational time, but did give the desired results.

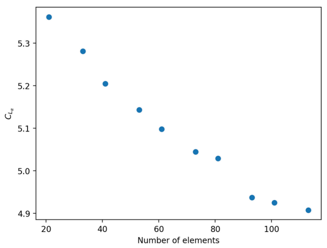


Figure 4: effect of N_w on $\mathrm{C}_{\mathrm{L}_\alpha}$ for PM

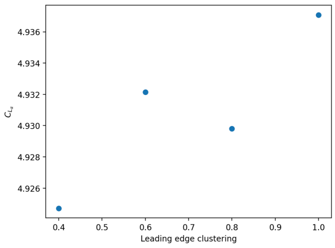


Figure 5: effect of TE Clustering on $\mathrm{C}_{\mathrm{L}_\alpha}$

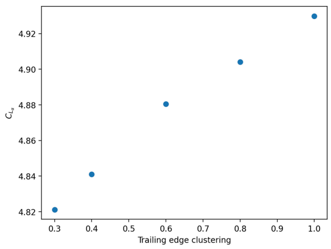


Figure 6: effect of N_w on $\mathrm{C}_{\mathrm{L}_\alpha}$

Clearly, the number of elements decreased the coefficient of lift and only starts to converge around 100 streamwise elements. Obviously, this takes a long time. For this reason, 90 streamwise elements will be used for trailing and leading edge clustering. This lead to a roughly halved computational time compared to 110 elements.

Almost all of the points in Figure 5 give an average of 4.93. Because the lower values of LE clustering are likely more accurate, these will be trusted more. LE clustering will be set at 0.8.

It can be seen that the trailing edge lift slope curve does not converge. The reason that TE Clustering=0.3 is shown instead of 0.2, is that at 0.2, the lift curve slope becomes 0.84. TE clustering is set at 0.6. This is chosen arbitrarily, because none of the values show anything that appears to be converged.
This fact highlights one of the flaws in trusting the smaller values of clustering more. As the elements get smaller, there can be issues with the precision of floating-point numbers in a computer. This is likely what caused the issue at small values of trailing edge clustering.

$$\frac{dC_L}{d\alpha} = \frac{a_0}{1+\frac{a_0}{\pi AR}} \quad (1)$$

Equation 1 gives the lift curve slope for a theoretical wing with aspect ratio AR. This will give an upper limit for what a symmetrical airfoil should give. OpenVSP gives an aspect ratio of 7.48, and putting this in Equation 1 gives a maximum lift curve slope of 4.96.
The lift slope curve of the wing will be less than this in incompressible flow, because the wing does not have an elliptical lift distribution. However, the flow is compressible and as the Mach number increases, so does the lift curve slope. Overall, I think that my results are reasonable, so these will be used.

This lead to an angle of attack of 3° at the design point given. This is not the exact angle of attack that would give the coefficient of lift for level flight, however CFD should not be trusted as accurate without experimental data. In flight tests at any scale, it would be difficult to keep an angle of attack to any number of decimal places, so this is deemed accurate enough. There would also be inaccuracies from rounding the components of velocity in ANSYS, however these aren't a point of concern either.

Value	VLM	PM
angle of attack	3.0	3.0
L/D	24.4	11.3

coefficient of lift	0.245	0.257
coefficient of induced drag	0.00319	0.0130

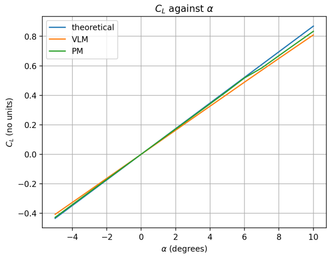


Figure 7: C_{L_α} for different methods

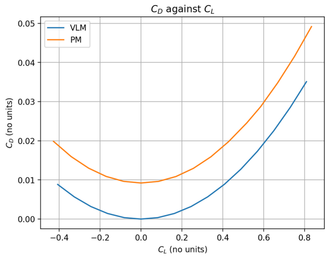


Figure 8: $\mathrm{C_L}$ against $\mathrm{C_D}$ For VLM and PM

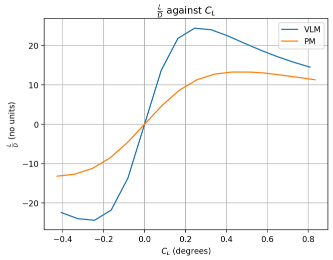


Figure 9: $\mathrm{C_L}$ against $\mathrm{L/D}$ for VLM and PM

The difference in the glide ratio curves in Figure 9 can be explained by Figure 8, there is less drag for the same amount of lift when using VLM. Even with the lower lift curve slope, this leads to an exaggerated glide ratio. Figure 8 suggests that there is some parasitic drag associated with viscous effects, however this could also be because of the fact that the wing has some thickness, whereas VLM has none.

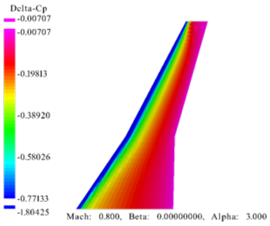


Figure 10: Upper surface at the design point using VLM

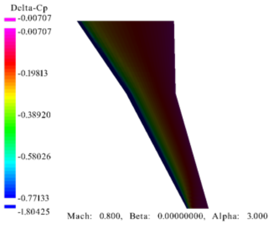


Figure 11: Lower surface at the design point using VLM

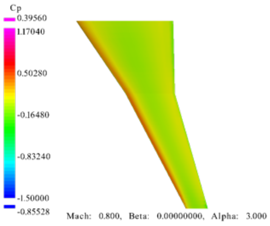


Figure 12: Upper surface at the design point using PM

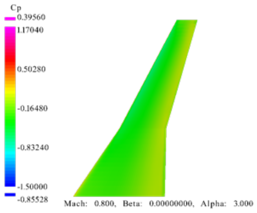


Figure 13: Lower surface at the design point using PM

There are strong gradients on the leading edges of both methods. This shows the need for well-resolved panelling at the leading edge. There are no shockwaves visible on the upper surfaces, because these incompressible solvers cannot resolve them.

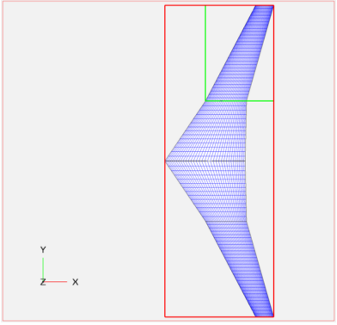


Figure 14: The panelling used for VLM analysis

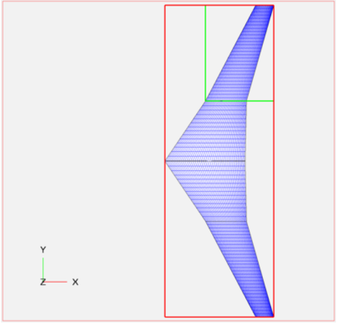


Figure 15:The panelling used for PM analysis

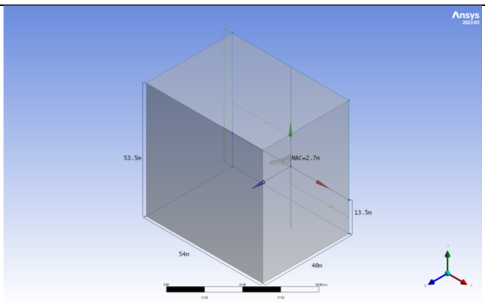


Figure 16: The domain used for Euler CFD analysis

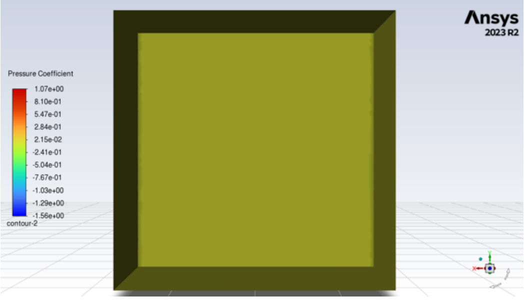
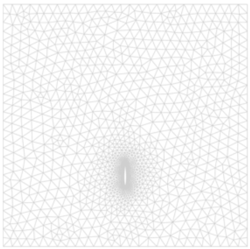
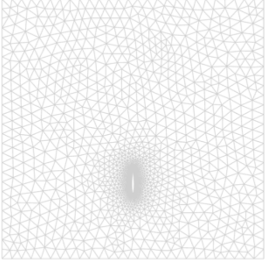
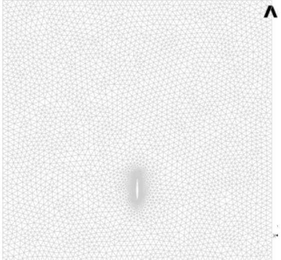
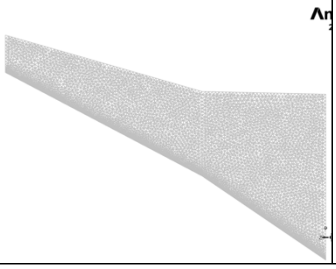
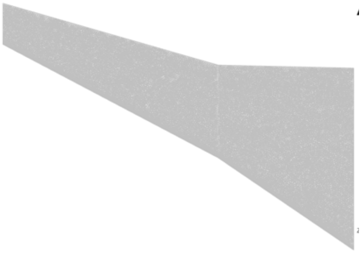
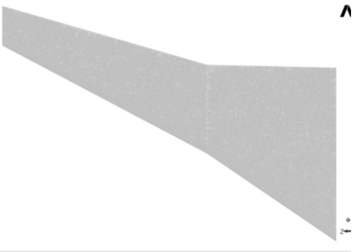
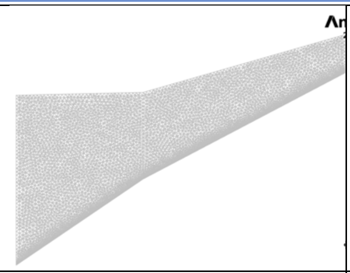
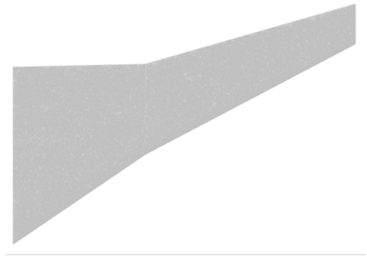
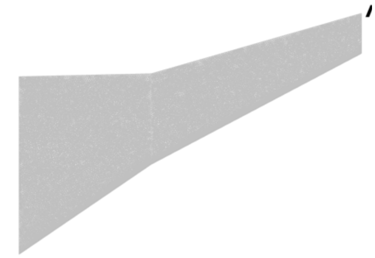


Figure 17: The pressure coefficients on all of the faces of the boundary, excluding the wing sand symmetry plane

This domain is fairly large. Normally, this would be a downside due to the added computational cost, however this was run on a computer that was fast enough for the calculations, and would otherwise be sat idle. There are still costs, such as the energy needed to run a computer for this long, however these will be on the order of £10, which is negligible in comparison to the cost of a University module. This mesh was large enough because none of the walls had significant pressure variations (see Figure 17). This means that they will not have truncated any flow structures of interest.

For the walls in the positive and negative X direction, as well as in the positive Z direction, the pressure far-field boundary condition was used. It is only valid when used with the pressure-based solver, and when using an ideal gas. These conditions are both satisfied here. It allows for the representation of freestream conditions.
In the negative Y direction, there was a pressure inlet. This accurately represents what the wing would experience at altitude. As well as this, the flow is able to slow down if necessary. This means that it will be easy to see if the domain boundaries are too close to the test section.
In the negative Z direction, there was a symmetry boundary condition. This is because the wings are symmetric along this plane. By using the symmetry boundary condition, the fluid could be simulated for only half the wing, but be valid for both sides.
The wing used the wall boundary condition.

	Coarse	Medium	Fine
Symmetry			
Side A			

Side B			
Nodes overall	725166	1252063	1909231
Nodes on wing	41719	98292	122496

Fluent mesh reports

For all of the meshes, the extents were as expected, ±27m in the X direction, -13.5m to 40m in the Y direction, and 0-40m in the Z direction. This is as designed.

For all of the meshes, the total volume was reported at 115555 cubic meters.
For the coarse mesh, the maximum and minimum volumes were 2.7×10^{-7} and 6.7 cubic meters respectively.
For the medium mesh, these were 3.5×10^{-7} and 7.6 meters respectively.
For the fine mesh, these were 4.5×10^{-8} and 6.5 meters.

For face area, the coarse mesh has a minimum of 2.5×10^{-5} and maximum of 7.4 square meters.
This was 6.4×10^{-5} and 7.9 square meters for the medium mesh respectively.
For the fine mesh, this was 2.1×10^{-5} and 7.8 square meters. You can figure out which was the minimum.

For the coarse mesh, one cell had a orthogonal quality of 3.4×10^{-2} and an aspect ratio of 60.
These values were 0.21 and 15 for the medium mesh. This time, they were on different cells.
The fine mesh had values of 0.16 for minimum orthogonal quality, and 25 for aspect ratio, again on different cells.

For each of the meshes, the main solution method used was the coupled solver. These were all pressure based. This was because it allowed for fairly fast convergence. For this, all of the methods used were second-order upwind.
There were some occasions where the solver would be very close to converged (continuity at 1e-5). When this was the case, I briefly changed the solver to SIMPLEC. This caused the residuals to rapidly increased. After a few iterations, the solver would be changed back to coupled. My hope with this was to avoid any local minima that the solver had found. This appears to have worked, as this allowed the solver to converge fairly quickly afterwards. Another solution that I also used was halving the under-relaxation factor when the residuals were around 1e-4. If this didn't happen, the residuals would oscillate, but not decrease that much.

Coarse mesh

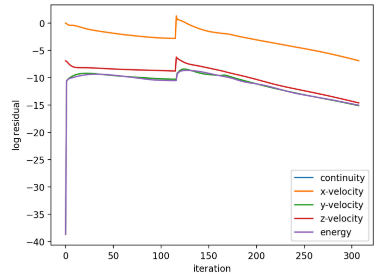


Figure 18: Residuals for the coarse mesh

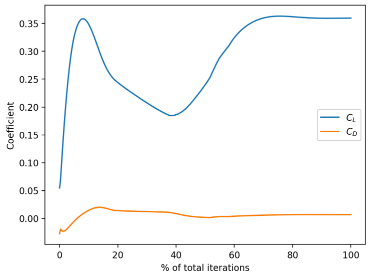


Figure 19: Values of the coefficients converging

In this run, SIMPLEC was left on from previous runs. The effect of this is clear, with the solver failing to find a solution. At around iteration 100, this was noticed and the coupled solver was correctly chosen, and the benefits of this are clear.

Medium mesh

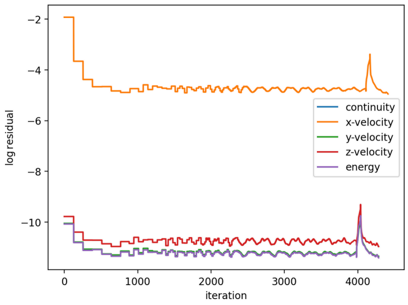


Figure 20: Residuals for the medium mesh

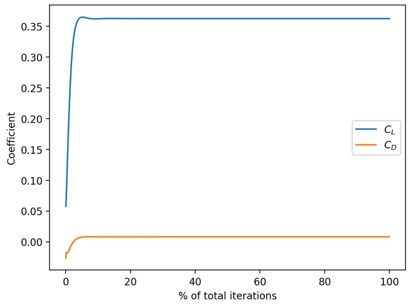


Figure 21: coefficients converging in medium mesh

Clearly, the residual graph has some issues with resolution. This is because there were many times that the solver was killed and restarted. This mesh was run on a different day to the other two, so for around 4000 iterations, I forgot that changing the solver temporarily could fix some of the issues with converging. Oscillatory behaviour is clear in well-resolved portions of the graph, as is the change to SIMPLEC and then the change back to coupled.

Fine mesh

As expected, a good-quality mesh lead to good convergence. The higher cell count meant that problematic cells could be split up into multiple cells, alleviating some of the issues associated with bad orthogonality, skewness and aspect ratio. For this run, the only change that was needed was decreasing the under-relaxation when the residuals became small enough to warrant it.

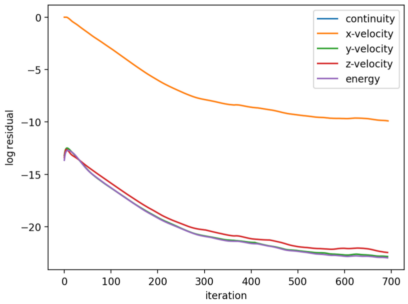


Figure 22: Residuals for the fine mesh

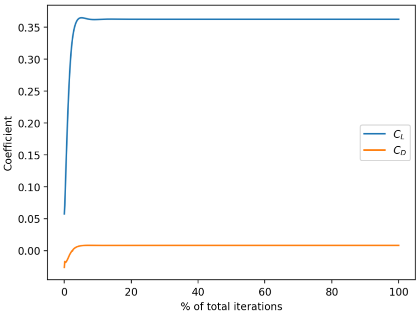


Figure 23: Residuals converging in fine mesh

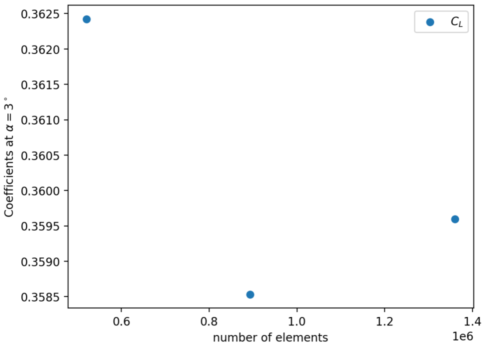


Figure 24: lift plotted against elements

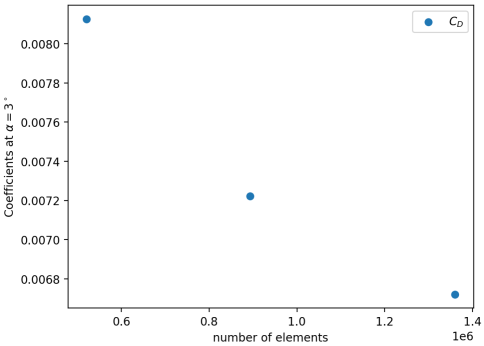


Figure 25: drag plotted against elements

At this scale, it looks like the number of elements has no bearing on the coefficient of lift. However, the scale shows a minute difference. All of these are identical when rounded to 2 significant figures. The drag coefficients have a minute difference between the highest and lowest. These both lead me to believe that the results are independent of the mesh, and any further study would be trying to get data from noise.
I will be using the fine mesh for my final mesh, as this is accurate and runs fast enough to be acceptable.

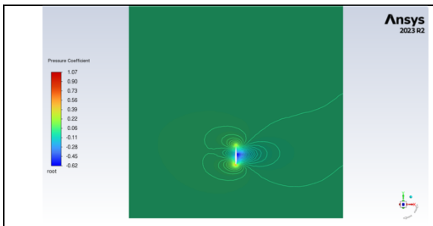


Figure 26: Pressures at the root

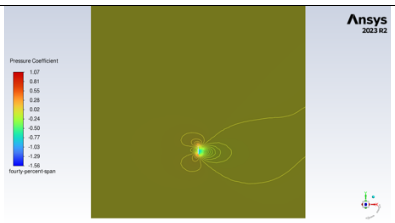


Figure 27: Pressures at 40% of the semispan

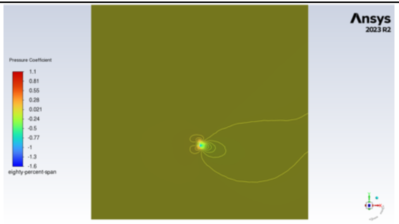


Figure 28: Pressures at 80% of the semispan

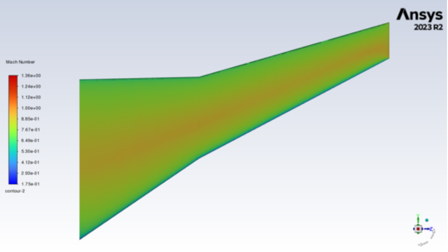


Figure 29: Mach contours on the lower surface

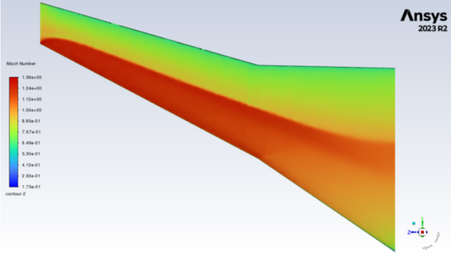


Figure 30: Mach contours on the upper surface

The shock is clearly visible in all of the above figures. This is good, as it indicates that the flow conditions have been set up correctly. There are clearly pressure contours that extend towards the far-field. This is fine, as they do not represent large pressure gradients, so any information lost will be minimal

At $\alpha = 3^{\circ}$	C_L	C_D
Euler CFD	0.360	0.00672
PM	0.260	0.0130
VLM	0.245	0.00319

Clearly, increasing fidelity increases the coefficient of lift. This will partially be because the more computationally expensive the method, the fewer assumptions are made. It is likely that the Euler CFD methods will give the most accurate results. It is unsurprising that Euler CFD has a higher amount of drag than the VLM analysis. This is simply because there is a shockwave visible in the Euler analysis, that is not present in the VLM analysis. This will obviously lead to higher drag. I am unsure why the PM gives such a high value for drag, the code was checked to ensure that the right value of drag was being read from OpenVSP. Verification against other studies would be a good way to check the validity of this.

It is always important to back up CFD results with real-world evidence. This data can be taken as representative of what the final values may be, however results should be checked against other methods, such as thoroughly-reviewed theoretical work. Ideally, real-world flight tests would be used to validate these results. In the real world, there are no assumptions made about the fluid, and the results are the results. However, real world flight tests represent an enormous cost relative to these analysis methods, highlighting why these methods have become so popular.