

University of Tennessee at Chattanooga

UTC Scholar

---

Honors Theses

Student Research, Creative Works, and  
Publications

---

5-2020

## Helios and Tenasi results for the workshop for integrated propeller prediction

Aaron Crawford

University of Tennessee at Chattanooga, twm449@mocs.utc.edu

Follow this and additional works at: <https://scholar.utc.edu/honors-theses>



Part of the [Aerodynamics and Fluid Mechanics Commons](#)

---

### Recommended Citation

Crawford, Aaron, "Helios and Tenasi results for the workshop for integrated propeller prediction" (2020).  
*Honors Theses*.

This Theses is brought to you for free and open access by the Student Research, Creative Works, and Publications at UTC Scholar. It has been accepted for inclusion in Honors Theses by an authorized administrator of UTC Scholar. For more information, please contact [scholar@utc.edu](mailto:scholar@utc.edu).

# **Helios and Tenasi Results for the Workshop for Integrated Propeller Prediction**

**Aaron Michael Crawford**

Departmental Honors Thesis

The University of Tennessee at Chattanooga

Mechanical Engineering

Examination Date: March 27, 2020

Dr. Kidambi Sreenivas

Associate Professor of Mechanical Engineering

Thesis Director

Dr. Charles Margraves

UC Foundation Associate Professor of

Mechanical Engineering

Department Examiner

# Table of Contents

Nomenclature	Page 3
Abstract	Page 4
Introduction	Page 5
Scope	Page 11
Methodology	Page 13
Results	Page 25
Conclusion	Page 65
Recommendations	Page 66
Acknowledgments	Page 67
References	Page 68

## 1 Nomenclature

ADD – Aviation Development Directorate, NASA Ames Office

AIAA – American Institute of Aeronautics and Astronautics

AOA – Angle of Attack

$c$  – Chord Length

$C_D$  – Coefficient of Drag

CDTC – Coefficient of Drag Thrust Corrected

CFD – Computational Fluid Dynamics

CFL - Courant–Friedrichs–Lewy

$C_L$  – Coefficient of Lift

$C_P$  – Coefficient of Pressure

CSD – Computational Structural Dynamics

$C_T$  – Coefficient of Thrust

DOD – Department of Defense

Helios – Helicopter Overset Simulation

LMMB - Langtry Menter, Medida-Baeder

NASA - National Aviation and Space Administration

RPM – Rotations per Minute

WIPP – Workshop for Integrated Propeller Prediction

## 2 Abstract

Beginning in 2018, the AIAA began the workshop for integrated propeller prediction. This effort was intended to bolster the CFD prediction capabilities of the aerospace community due to a resurgent interest in propeller driven vehicles for electric applications. The test bed for this workshop is the NASA X-57, a model of which was experimentally tested in the Lockheed Martin Low Speed Wind Tunnel in January 2019. Using the 3D scanned geometry provided by the workshop a mesh was created and all priority one simulations were performed using two flow solvers. The flow solvers used were Helios, developed by the ADD office, and Tenasi developed at the UTC SimCenter. The flow solvers were compared as closely as possible with results relating directly to the metrics studied by the experimental tests. This includes the aerodynamic parameters of  $C_D$ ,  $C_L$ ,  $C_T$ . This also includes  $C_P$  plots at 6 span locations for all cases and lastly a wake survey yielding integrated torque and thrust values for the powered propeller case. The results show that while Tenasi functions well as an all-purpose solver, outperforming Helios in some areas, the best predictions of the propellers and their interactions come from the Helios flow solver.

## **3 Introduction**

### **3.1 Workshop**

The American Institute for Aeronautics and Astronautics (AIAA) is a national organization that promotes and showcases research in various fields. It often holds workshops that seek to develop research in an area by bringing together many people and groups to work on the same project. The workshop this research is concerned with is the Workshop for Integrated Propeller Prediction (WIPP). Through this program, using the Maxwell X-57 as a test bed, the goal was to develop the CFD prediction capabilities for integrated propeller models due to a renewed interest in propeller-driven aircraft for green applications (Ginn and Hooker 2019). The workshop provided the testbed as well as the wind tunnel testing that the CFD results were to be compared to. Beginning in 2019, I began work on this workshop at the SimCenter using Tenasi. This work was continued over the summer of 2019 at the ADD office using Helios. The work at UTC and ADD were both continued as the workshop simulations were completed and the methods and solvers used at both offices are compared.

### **3.2 Introduction to CFD**

Computational Fluid Dynamics (CFD) works in conjunction with Experimental Fluid Dynamics (EFD) by simulating test conditions without having a physical model or the equipment to perform the experimental tests. CFD approaches essentially provide best estimates based upon the Navier-Stokes and other equations; however experimental tests measure values directly. CFD processes involve the use of a computer-based 3D model of the test object. A 3D mesh is created around the object so that desired variables can be calculated at each point. For this particular case the geometry provided by the workshop serves as the starting point. With a good understanding of

computational processes, many things can be modeled accurately. However, experimental data is always needed to verify CFD results. Since the 1960s-70s, CFD has been used for increasingly complex flow problems. Since most complex problems cannot be solved analytically, CFD must be relied on to give answers to these problems.

One of the most crucial parts of the CFD process is the flow solver. At its core it solves three fundamental equations: the conservation of mass, conservation of momentum, and conservation of energy. These three equations are solved in most cases over a computational domain consisting of non-overlapping control volumes or elements.

Flow solvers are either steady or unsteady. In this case both options are used to complete the simulations. What distinguishes a steady vs unsteady simulation is time. A steady solution is computed with the time derivatives pushed to zero. This indicates the solution is converged, meaning the solution no longer changes with respect to time. Using a steady solution does involve the assumption that the solution is steady. Often, this is not correct as many real-world fluids problems have unsteady elements to them. Of particular concern, in this case, is unsteady vortices. These often occur at the blunt trailing edge of an airfoil and can cause variation in the convergence of a steady solution. Despite this, steady solutions are computationally efficient and require less computational resources. This is because the solution does not need to progress in time and can utilize convergence acceleration techniques like local time-stepping; wherein the time step is allowed to vary from one control volume to another to drive the time derivative to zero as quickly as possible. Consequently, typical steady solutions do not employ sub iterations. Unsteady solutions are solved by advancing the solution both in time and in space. This typically causes a

solution to be much slower with a higher computational expense. Additionally, it may take some time for motion or flow features to reach a periodic state in time. In particular, propellers and rotor behave this way. Often, cases with spinning bodies like this need a few revolutions to allow the downwash to convect past the geometry and progress past the transient start-up phase.

In addition to this, flow solvers often include the use of turbulence models. The goal of a turbulence model is to provide a numerical solution that describes the phenomena of fluids that are not directly covered or included in the governing equations. Many turbulence models have been created. Some have been created for very particular circumstances while others work well on a wide variety of cases. Beyond turbulence models another model to consider is transition models. For aircraft, these models can be extremely important. These models help predict when and how an incoming laminar flow transitions to turbulent as it passes over a geometry like an airfoil. For all models a more accurate picture of how the flow behaves will provide a more accurate solution, however, models not suited for a problem can create non-physical flow features or cause more inaccuracy.

The addition of turbulence and transition models help to bolster the capabilities and accuracies of a flow solver. Though there are many flow solvers available, the two that will be focused on are Helios and Tenasi. FUN3D is a NASA developed unstructured flow solver that has been around and in use for more than two decades. This solver has been a standard flow solver and used in a variety of applications (Borer et al. 2017). This flow solver works in the overarching Helios framework and is tied to the many other modules available within Helios. Tenasi is the in-house flow solver for the UTC SimCenter. Tenasi is a stand-alone solver belonging to UTC so



simulations using Tenasi are run locally through its specific software setup. This flow solver, like the others, has been used for a variety of problems (Sreenivas et al.).

It is important to recognize the role of HPC or High-Performance Computing in the completion of this project. When considering these large cases often with millions or hundreds of millions of points the computational power of a single desktop computer or a single core would never be able to process the solution fast enough or contain enough memory. Through HPC capabilities, the computational domain can be split into many pieces and sent off to different cores and calculations can proceed in parallel rather than in a sequential manner. This allows a solution to take hours or days as opposed to months or years. For this project the resources granted through the UTC SimCenter and the DOD HPC program were utilized. Resources in excess of 5,000 hours (wall-clock time) and 750,000 core hours were used in the completion of this project.

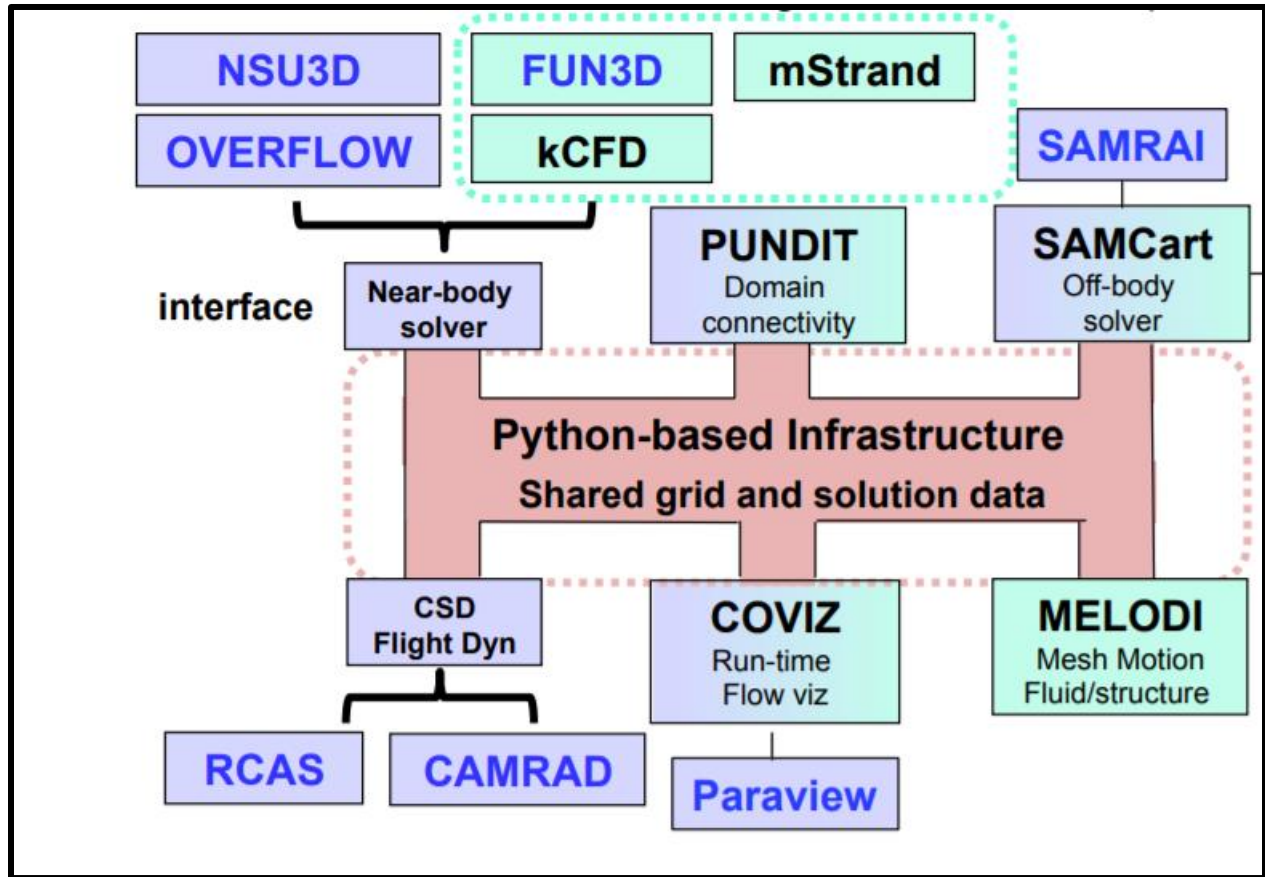
### **3.3 Tenasi**

Work began at the University of Tennessee at Chattanooga in September 2018, generating meshes and running simulations for the proposed test conditions. Grids were updated as more information became available and the wind tunnel tests were completed in January 2019. All the simulations that were carried out used actual propeller geometries with appropriate relative motion. At the University of Tennessee at Chattanooga's SimCenter meshes were generated using Pointwise (Pointwise Version 18.2R2) and run using the in-house flow solver Tenasi. Tenasi is a fully unstructured flow solver developed at the SimCenter. To complete the simulations with accurate relative motion a sliding interface was utilized around the propeller and nose of the geometry. Successful simulations were completed for test conditions shown in Figure 4 below. The

simulations were run on a local cluster using 48 cores. Each case was started under steady-state conditions to setup the general flow characteristics and to get the flow moving in the correct direction. Once this was accomplished, the simulation was continued in a time-accurate manner for at least another five propeller revolutions.

### **3.4 Helios**

In the summer of 2019, this project was continued at the US Army Aviation Development Directorate (ADD) office at the NASA Ames site in Moffett Field, California. Using the tools from the CREATE-AV suite, including Helios, further CFD simulations were completed. Helios is a CFD software package developed by the Aviation Development Directorate (ADD) office at NASA Ames (Wissink et al. 2016). Helios operates primarily on Department of Defense (DOD) supercomputing facilities and requires users to be authorized before using it. This framework was specifically designed with rotorcraft in mind; however, it can be applied to other flow fields too. The package Helios (Figure 1) acts as an overarching structure that ties in many different modules in one Python framework. Helios allows for a few different near-body solvers including FUN3D, OVERFLOW, and mStrand. Helios also contains a Cartesian off-body solver with an automatic mesh refinement named SAMCart. Additionally, Helios has modules for visualization (Coviz), body forces (MELODI), and grid cutting (PUNDIT) (Wissink et al. 2016). All these components work together to create one powerful and unparalleled CFD solver.



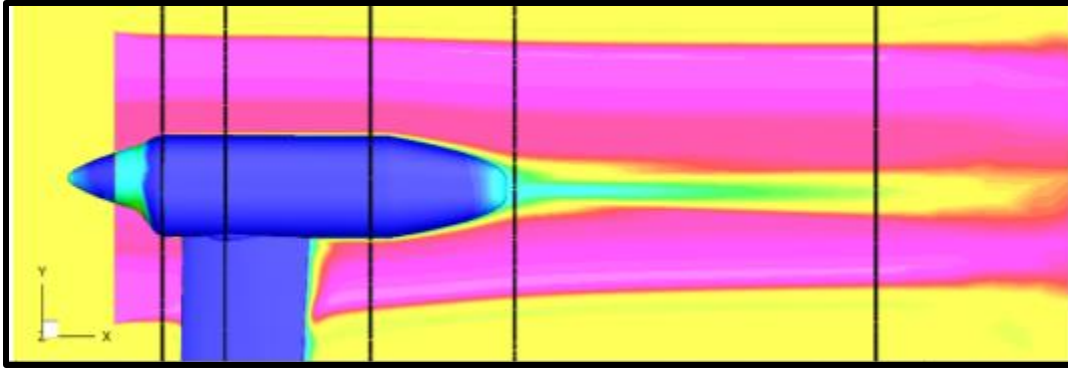
**Figure 1: Helios Infrastructure Reference**

The off-body flow solver chosen for all simulations was the Helios developed SAMCart, a fully cartesian grid with automatic mesh refinement capabilities. Using Helios, one set of run conditions was simulated for several geometry configurations: an isolated propeller and nacelle, an isolated wing, and an integrated wing propeller configuration. The two configurations of particular focus in this project correspond with the priority one configurations: the integrated wing propeller configuration and isolated wing configuration. Two different near body solvers were used. FUN3D (FUN3D Ver. 13.6-717bd48) was chosen because of its long history and extensive capabilities. Additionally, several cases were carried out with mStrand, the ADD developed strand solver (Lakshminarayan, et al. 2017). Though an integrated wing and propeller configuration

mStrand simulation was never completed, both solvers performed well with mStrand showing promising results. However, for this research mStrand was excluded and no priority one simulations were completed due to difficulties as well as time constraints.

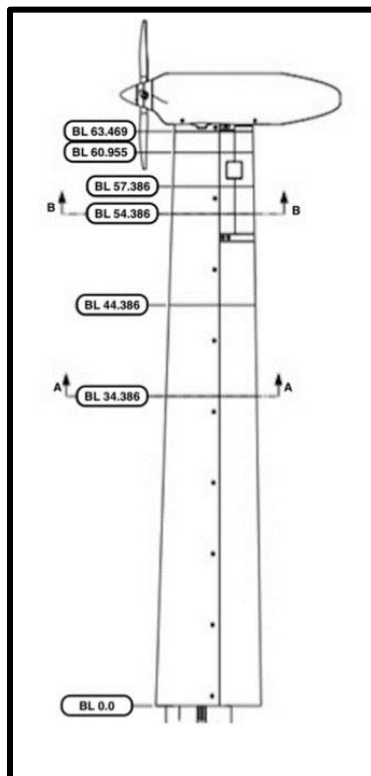
## **4 Scope**

The scope of this project has changed drastically since its first inception. Initially, this thesis was to be about CFD validation of a Particle Image Velocimeter (PIV). This was based on the stipulation that one of these setups would be purchased by the department. Over the summer of 2019 this did not occur, so the scope of this thesis needed to shift. During the summer, I was working closely with mStrand and FUN3D at the ADD offices at Moffett Field. The scope of the thesis shifted to testing these two solvers on the WIPP case. Given the relatively immature nature of mStrand this would serve as a perfect test case. Unfortunately, during the summer and fall semester, significant progress was not made in simulating relative motion with mStrand. These cases would blow up due to extremely high residuals or produce NAN results because of negative pressure (non-physical). After this, the focus of this thesis shifted once again to comparing Helios and Tenasi against the workshop data. With these two solvers selected not much needed to be done to compare them accurately. The simulations selected to perform were all the priority one simulations as shown below (Figure 4). In addition to these simulations the desired results from all runs are the wake data produced by the vertical and horizontal rake for the powered simulations at zero degrees. Data is taken from locations in X at 1.5, 5, 13, 21 and 41 inches (Figure 2). From this data the thrust and torque could be calculated. This data directly mirrors the rake data collected and analyzed during the wind tunnel tests.



**Fig 2. Wake Locations (Vertical lines indicate data extraction locations)**

Additionally, the  $C_p$  plots for each simulation were created at all the span locations shown below (Figure 3). Again, this directly mirrors the experimental tests. Here  $C_p$  is the non-dimensional pressure value on the pressure and suction surfaces of the airfoil. This is found by non-dimensionalizing the surface pressure relative to the freestream pressure.



**Figure 3: Pressure Taps Locations**

Lastly, the  $C_L$  and  $C_D$  for each simulation is calculated. These values represent the non-dimensional lift and drag calculated by non-dimensionalizing with the freestream velocity, density of the air, and the area of the wing. Through this method the performance of the model can be determined regardless of size or speed. The cumulation of all results is shown in section 6.2.

Test Conditions for CFD Simulation						
Priority	Mach	q (psf)	q (psi)	h (ft)	Re/ft (x10 <sup>6</sup> )	ReMAC (x10 <sup>6</sup> )
1	0.08	9.68	0.0672	0	0.567801	0.480265
2	0.11	18.31	0.1272	0	0.780726	0.660364

Angles of Attack for CFD Simulation	
Priority	Angles-of-Attack (degrees)
1	0, 5, 7, 15, 17
2	-7, -5, 0, 2, 5, 7, 9, 11, 13, 15, 17

Propeller Power Setting	
Priority	Prop Thrust Coefficient (CT)
1	0, 0.4
2	0, 0.04, 0.20, 0.40

Note: CT = 0 is Propeller-Off

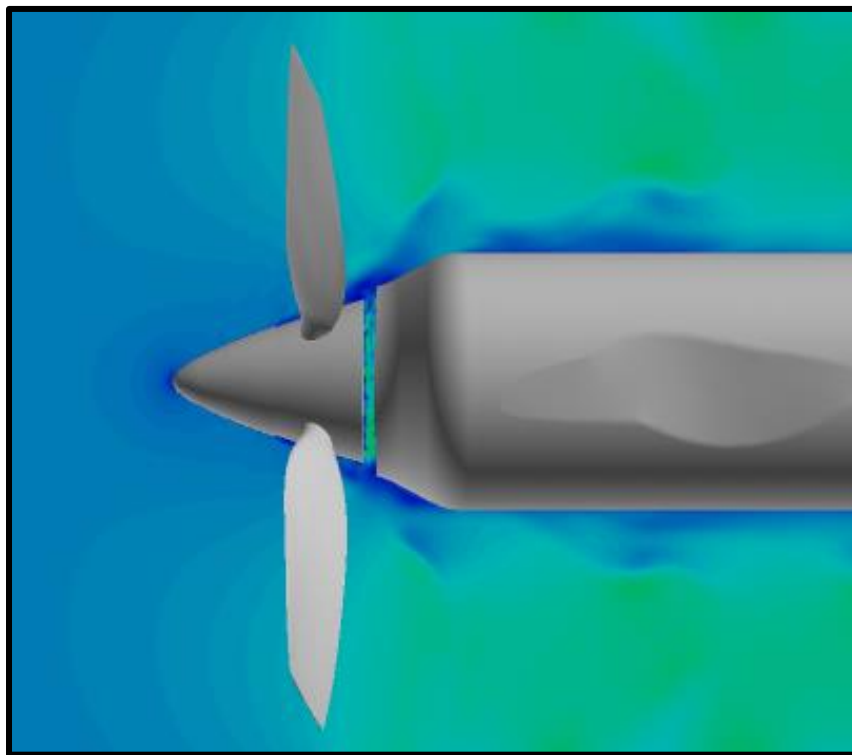
**Figure 4:** Workshop Specified Test Conditions

## 5 Methodology

### 5.1 Mesh

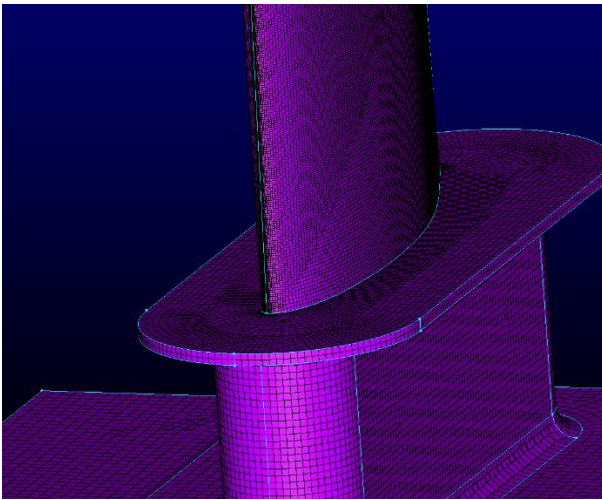
The first steps of this project began in the Fall of 2018 when the project was initiated, and the process of meshing began. At this time only the CAD model had been released by the workshop and it did not have the geometry of the props. Early in the Spring of 2019 an accurate scanned model was released after the wind tunnel tests took place. This model was created by performing a complete 3D scan of the test model. The mesh was created and modified in Pointwise. The mesh initially created for this model was updated many times. The off the wall spacing was based off a  $y^+$  of 1 using the propeller tip speed as the reference velocity. The  $y^+$  value indicates the non-

dimensional normal spacing off the wall. Since surfaces are often modeled with a no-slip condition the velocity gradient very close to the surface becomes steep. The faster the object is moving the steeper the gradient. To properly resolve this boundary layer an appropriate spacing off the wall is necessary. By basing the  $y^+$  on the propeller tip speed, the model location with the highest speed, this causes a bit of excess refinement around the stationary areas particularly the wing. This extra refinement could likely be reduced, however, the difference in computational cost is likely negligible. The most notable meshing details involve changes to the geometry itself. To deal with the relative motion found in the Helios solver a small gap has to be created just behind the spinning nose (Figure 5). This allows for the nose and props to rotate while the rest of the model stays stationary.

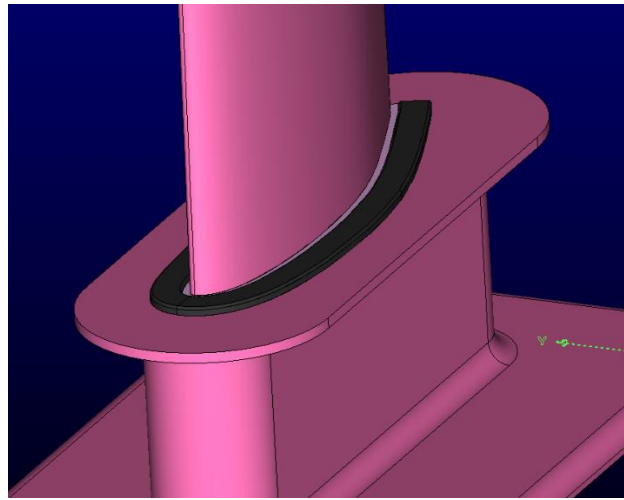


**Fig 5. Gap Between Nose and Nacelle**

Another notable change has been made to the base. The small lip shown on the CAD model (Figure 7) has been removed for ease of meshing (Figure 6). This choice may contribute to some inaccuracies however the overall impact of removing this feature is likely small.



**Fig 6. Final Mesh With No Lip**

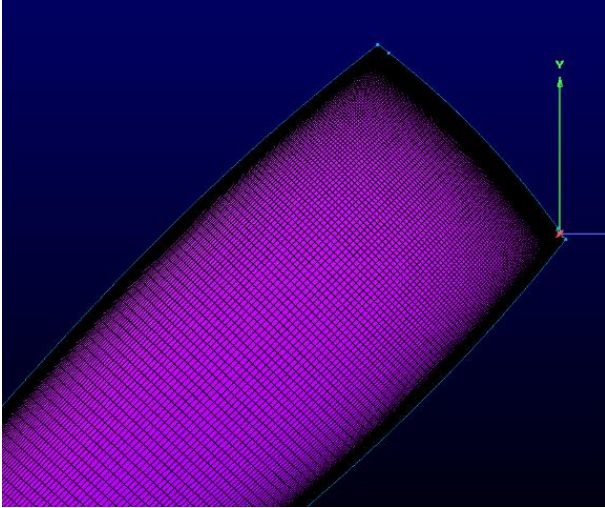


**Fig 7. Workshop Geometry With Lip**

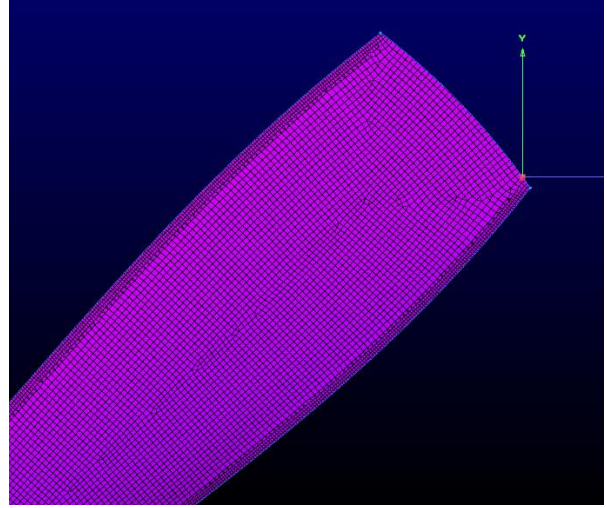
Lastly, the propellers are of importance. The version used for all Helios simulations assumes that all propellers are identical, i.e., only one propeller is meshed and then rotated around the nose. On the other hand, the Tenasi simulations contained four separately meshed propellers. This leaves the Helios model with identical propellers and the Tenasi simulations containing four propellers each with slight differences due to the physical manufacturing differences. It is highly unlikely that these slight differences contribute to any substantial differences in the simulated performance. Additionally, these were meshed using a combination of structured and unstructured meshes. This choice was made due to the ease of meshing as well as allowing for the best refining and mesh characteristics given the incredibly thin blades with incredibly fine leading edges. Due to the inability to create a working Tenasi model a previous version of the mesh was used. This mesh contained significant differences in refinement particularly in regards to the propellers. The



following figure illustrates this (Figure 8 and 9). This difference likely corresponds to the greater inaccuracies found in the Tenasi simulations.



**Fig 8. Helios Propeller**



**Fig 9. Tenasi Propeller**

The following mesh statistics describe the near body and off body point counts both before and after using Helios’s Automatic Mesh Refinement (AMR). Additionally, it shows the point count for each Tenasi simulation including sliding interface and stationary meshes.

**Table 1: Helios Simulations Point Counts (Millions)**

	<b>Point count (off-body)</b>	<b>Point count (near-body)</b>	<b>Point count After AMR (off-body)</b>	<b>Total Total Count</b>
Propeller	20	52	21	73
No Propeller	19	32	20	52

**Table 2: Tenasi Simulations Point Counts (Millions)**

	<b>Stationary</b>	<b>Sliding Interface</b>	<b>Total Point Count</b>
Propeller	26	19	45
No Propeller	27	N/A	27

## 5.2 Tenasi Methodology

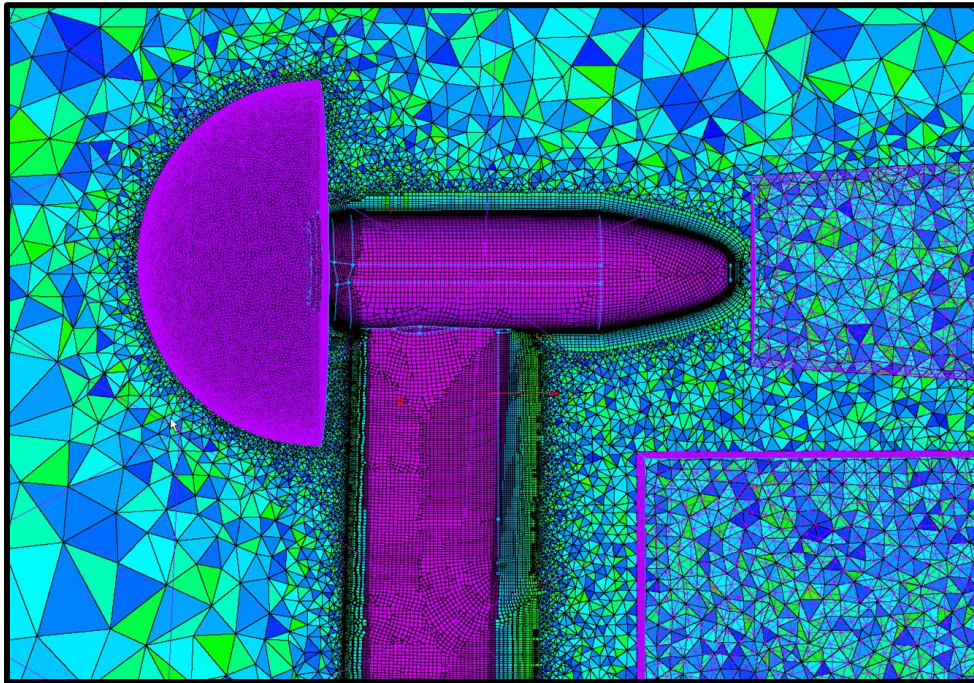
Using the meshes created previously, the simulation case was created at the SimCenter using the modules available in house. The mesh used for the no propeller case was identical to the mesh used for the same Helios case. The propeller mesh used was one created before the to Helios work and contains fewer points as shown above in Fig 8 and 9. The boundary conditions used for the propeller and no propeller cases are as shown below.

**Table 3: Tenasi Boundary Conditions**

<b>Boundary</b>	<b>Condition</b>	<b>Volume Condition</b>
Nacelle Wing and Base	Viscous Adiabatic	Stationary
Nose	Viscous Adiabatic	Stationary/Rotating* (depending on the propeller or no propeller case)
Farfield	Farfield	Stationary
Propellers	Viscous adiabatic	Rotating
Sliding Interface	Sliding and Sliding Partner	Stationary/Rotating

Most notably, the setup for the powered Tenasi cases requires the use of a sliding interface. This method is very important for the simulation of the propellers as it allows for the motion of the propellers relative to the rest of the model. Essentially it works by designating a volume mesh that

will rotate or slide within another stationary mesh. In this case the rotating mesh was created around the nose as well as the propellers.



**Fig 10. Sliding Interface**

As seen in the image the semi-circle volume encloses the rotating portions of the model. The outer surface of this spinning mesh is then duplicated and applied to the stationary mesh similar to a wall though there is nothing inside it (Figure 10).

This identical sliding interface works by allowing the rotating mesh to move at each time step. Information is then passed between each mesh through the sliding interface. Since they are identical surfaces there are always points close-by on the opposite surface for each point to pass information to. For these cases the forces on each component of the model are calculated. As per the workshop the forces on the base are not included in the lift and drag calculations. The initial conditions are those specified by the workshop (Figure 4). The speed of rotation is prescribed at

5960 RPM. Across several workshop wind tunnel tests this is approximately the average with an error of up to 100 rpm. This average is considered good enough for all simulations considering it is at most up to 1.5% off.

The simulation for all spinning cases progresses for 1800-time steps, or 5 revolutions, using a steady solution method. After this period the solution is progressed unsteady. The solution is advanced this way so that the first section of simulation can set up the general flow features and convect the flow downstream while using less computational expense. After this the true unsteady solution does not have to run for as long to achieve convergence. The time step used for all the cases is 0.000027s. One case was simulated with a quarter degree turn per time step, or 0.25 times the original time step. This case is compared to the others below. The solution is considered converged analyzing the force convergence in addition to the residuals. Example plots are shown below in the results. Due to the vortices of the propellers interacting with the nacelle and wing the forces of lift and drag do oscillate and are inherently unsteady. The convergence is thus checked through the occurrence of a periodic oscillation that is steady. Output files were created through averaging every time step for one revolution or 360 steps.

For the no power cases the mesh containing only one volume block and no propellers are used. This case is similar to the propeller case and requires identical boundary conditions and initial conditions except for the exclusion of all rotating related inputs. These cases were run for a minimum of 20000 iterations or until forces converged well. Output files were averaged over the final 500 iterations.

The solver conditions used for Tenasi simulations are detailed as follows. For propeller cases the search tolerance for the sliding interface was set as 0.05 m. This is the distance over which the solver will attempt to find a point across the sliding interface in order to pass information from the stationary mesh to the rotating mesh or vice versa. The CFL for most cases was set at 25. The CFL parameter controls the size of the time step taken for each simulation. Typically lowering the CFL improves the stability of a simulation however, the solution takes longer to progress to an equivalent convergence. For some of the steady no propeller cases the CFL was lowered to 15 due to issues with pressure going negative. For the propeller cases the time step was equivalent to 1 degree of propeller rotation at an RPM of 5960. All unsteady cases had 3 newton iterations and were solved 2<sup>nd</sup> order in time. The turbulence model used for all cases was a standard SST model. This model is a good choice for these simulations because it combines two similar and robust turbulence models, K-Omega and K-Epsilon. These models both include two equations one, K, which seeks to model turbulent kinetic energy and Omega/Epsilon which is a numerical dissipation term. Both these models and SST have been used extensively for external flows and aircraft. (Menter et. al 1994) (Spalart et. al.)

### **5.3 Helios Methodology**

The initial conditions used for all Helios cases are the same as the Tenasi cases. This includes the RPM, and axis of rotation. The initial flow conditions were specified in Helios using an altitude and Mach number of 0 and 0.08 respectively. The boundary conditions are much the same however, the naming convention is slightly different due to the different solvers used. The boundary conditions specified for Helios are designated by the near body or off body solver used.

**Table 4: Helios Boundary Conditions**

<b>Boundary</b>	<b>Condition</b>	<b>Volume Condition</b>	<b>Solver</b>
Nacelle Wing and Base	Wall	Stationary	FUN3D
Nose and Propellers	Farfield	Rotating	FUN3D
Floor	Symmetry	Stationary	FUN3D
Walls	Inviscid Walls	Stationary	SAMCart
Inflow/Outflow	Inflow/Outflow	Stationary	SAMCart

The wall condition specified varies slightly from the Tenasi wall conditions in that it only enforces a zero velocity. This, however, is not a significant difference because although the Tenasi boundary conditions also enforce no heat transfer, heat transfer is not a considered variable at the wall and there is no specified heat from the wall to the flow. The simulation of the walls of the wind tunnel as inviscid walls is assumed to be a good assumption. This boundary condition enforces a normal velocity of zero. This is very similar to what is seen in a wind tunnel since the bulk of the flow moves tangent to the walls. Also, since no forces are being measured on the walls any drag that occurs due to viscous flow is inconsequential.

Two built-in modules inside Helios required some special specifications. The first is PUNDIT. PUNDIT is the built-in whole cutting module that helps create the overlap between near body and off body meshes and cuts the meshes at each time step at the location with motion. Through previous difficulties it has been found that the “meclude” values and “mfringe” values need to be altered. If they are not changed there is difficulty in the gaps between the spinning nose and the nacelle. The default values for these are 4 and 3, and are changed to 2 and 1 respectively. These

values determine the number of layers of overlap that determine where PUNDIT begins to cut the mesh. Additionally, since the model is not closed and has an open body this information must be specified in PUNDIT. In this case, the open body is in the Y+ direction, meaning the body is hollow moving from the bottom samcart bounds in the Y+ direction. This designation ensures no points are placed inside the geometry.

The second module with required inputs is SAMCart. SAMCart determines the outer bounds of the computational space. The bounds specified closely model the specifications of the wind tunnel. The extents of the domain are as follows and are recorded in inches.

**Table 5: Specified SAMCart Bounds (Inches)**

<b>Plane</b>	<b>High</b>	<b>Low</b>
X	225	-75
Y	192	0
Z	140	-140

The boundary conditions specified for each plane can be found above. The spacing is the next major component of the SAMCart specifications. The coarsest global spacing set for all cases is 2 inches. Due to the cartesian nature of the mesh the global spacing must be evenly divide into the bounds specified above. For this case geometry refinement was applied. This setting makes SAMCart automatically refine the off-body mesh near a body. It does this by reducing the spacing until the ratio between the SAMCart cells and the near body cells are as close to 1 as possible. The decrease in SAMCart cell size must occur in halves, because of this the ratio may not reach 1

easily, to be conservative SAMCart will always tend toward smaller cells, or a ratio less than 1. SAMCart’s ability to perform automatic mesh refinement is also used for all cases. This ability refines the around regions of high vorticity as calculated through Q criterion. This ability while very helpful can often cause the mesh point count to explode. For this reason, the region in which SAMCart refines is limited to the following bounds. This region roughly represents the box surrounding the geometry.

**Table 6: SAMCart AMR Region (Inches)**

<b>Plane</b>	<b>High</b>	<b>Low</b>
X	40	-10
Y	90	0
Z	16	-16

The last and third module is Coviz. The stands for Co-Vizualization and is responsible for writing out the visualization files typically in field view format. For all propeller cases the boundary Fieldview files are exported every 144-time steps or 10 degrees. This is necessary so that the  $C_p$  plots on the surface can be time-averaged across one or more propeller revolutions as the downwash influences a minimum of two  $C_p$  lines on the wing. In addition to the boundary, contour files are output for the propeller cases. These files output the surfaces with a Q criterion above a particular threshold, in particular 0.001. These files together help visualize the impact of the propeller over time and how the downwash impacts the nacelle and wing. In addition to these files virtual taps also are used here to collect wake data. During wind tunnel testing a pitot tube rake was used to collect data in the wake of the propeller. By specifying a location corresponding to a



wind tunnel data collection location, rake data can be reproduced computationally. This data includes, pressure, axial and swirl velocity. This data is exported in increments of 5 degrees. This output can then be integrated to predict the thrust and torque of the propeller. No special inputs were specified for the MELODI module.

The flow solver conditions were given as follows. The turbulence model for all cases was specified as Spalart-Allmaras. The Spalart-Allmaras turbulence model has been heavily used in many aerospace simulations over more than 20 years. It benefits over other turbulence models because it is only one equation thus requiring less computational resources. It is also generally stable and has been shown to work well for many external flow problems. All unsteady simulations are solved 4<sup>th</sup> order in time and space. The CFL for these cases is 100. For all cases presented no transition model was used, however, the three transition models available in Helios have been tested against one another and compared using their lift and drag predictions. The three transition models tested are Coder (Coder et. al 2014), Menter (Menter et. al. 2015), and Langtry-Menter, Medida-Baeder (Medida et. al. 2011).

## **5.4 Post Processing and Comparison**

The post-processing typically occurs through the use of FieldView (FieldView Version 19) as well as MATLAB (MATLAB Version R2019b). To obtain accurate lift and drag values the forces on the appropriate components are averaged over the last 10% of simulation or last revolution. These averages are computed using MATLAB. A Fieldview script is used to parse through boundary files and extract the pressure lines needed. MATLAB is then used to averages, nondimensionalize, and combine the  $C_p$  data into one CSV file that can be used with workshop data to produce side

by side plots. Surface pressure data is non-dimensionalized based on the freestream pressure. A Fieldview script is also used to create images and visualize the propeller wake. Lastly, MATLAB is used to process the wake data, produce plots, and integrate over the proper region. The integration is performed using MATLAB's integration method and is integrated only over the regions in the downwash of the propeller as specified by the workshop.

## **6 Results**

Before my senior year, initial work was performed on this project using Helios. The results of this simulation can be seen below.

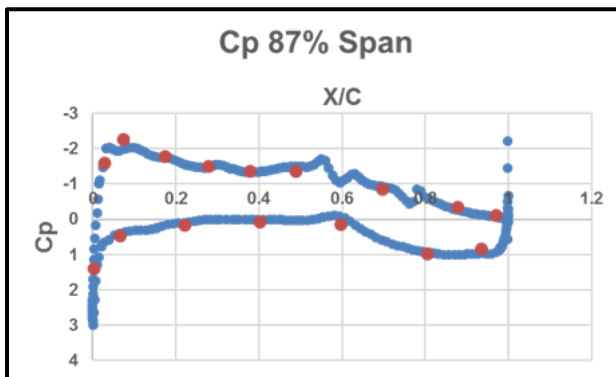
### **6.1 Initial Helios Results**

During summer 2019 work was completed using Helios. These simulations do not reflect any of the required priority one results, however, they do serve as a baseline simulation to show the effectiveness and accuracy of these solvers. Additionally, it served as a great starting place upon which all further priority one simulations would be completed.

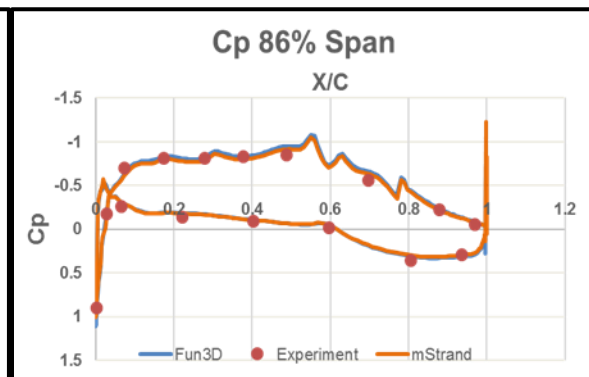
Figure 11 below shows the FUN3D solution from Run 176 at the fifth pressure line, approximately 87% span. This solution was time-averaged over approximately 1.5 revolutions. It is in the downwash of the propeller and it matches the experimental data very well. Figure 5 is a comparison of both the FUN3D and mStrand solution as well as the experimental results for the isolated wing case (No Propellers). These figures show a very close agreement for both FUN3D and mStrand solutions. This gives a positive indication for the  $C_p$  plots of the remaining simulations.

**Table 11: Helios Results M=0.11, CT= 0, 0.4**

Quantity	Experiment	mStrand	Fun3D
Propeller and Nacelle Thrust	N/A	151.16	151.72
Isolated Wing CL	0.5806	0.577	0.6005
Isolated Wing CD	0.0302	0.0403	0.0423
Full Model CL	0.7012	N/A	0.6734
Full Model $C_D$ Thrust Included	-0.3757	N/A	-0.3755
Full Model $C_D$ Thrust Removed	0.0255	N/A	0.0404



**Figure 11:** FUN3D  $C_p$  Plot Pressure Line 5 (Run 176)

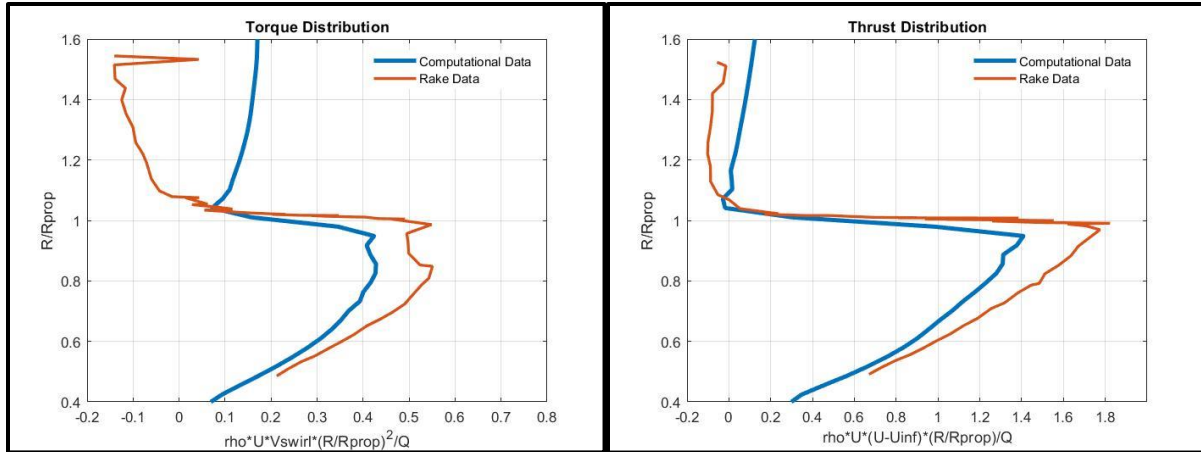


**Figure 12:** FUN3D, mStrand and experiment pressure line 6  $C_p$  plot (Run 03)

Also, torque and swirl data are computed for the Helios FUN3D Run 176. The torque and swirl were calculated by beginning at a Y coordinate of 70 inches and moving up at 0.25-inch increments until approximately two times the propeller radius. The upper integration limit is taken as  $R/R_{prop}=1$  despite having information beyond this location. This does not exactly duplicate the experimental tests, however; the computational results allow for the computational rake to get closer to the nacelle than allowed experimentally. Table 4 details the comparison of computational, experimental and calibrated results. The results in Table 4 show a slight underprediction of the experimental integrated values. For thrust the workshop determined the experimental integration provides an overprediction of thrust. For the case of thrust the calibrated results seem to be more accurate and result in only an 11% error from the computational results. With the torque the accuracy of the experimental integration is in question however, computational results produce a 21% error for this metric. At this time only the 1.5-inch rake location has been examined.

**Table 12: Comparison of integrated quantities (Helios; x = 1.5 in)**

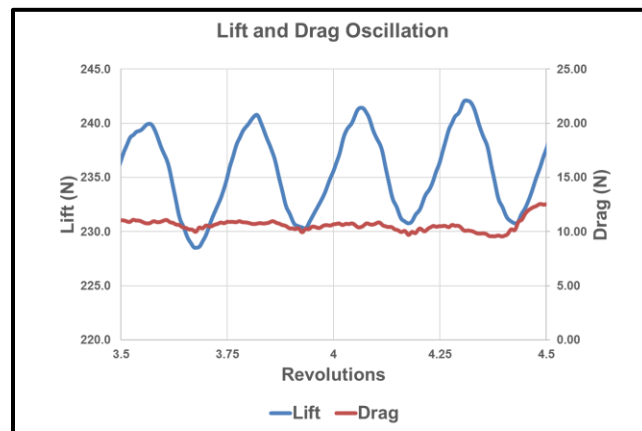
<b>Quantity</b>	<b>Computation</b>	<b>Experiment</b>	<b>Experimental Calibrated Results</b>
Torque (ft*lbs)	6.51	8.39	N/A
Thrust (lbs)	29.89	37.28	33.86



**Fig 13. Torque and Thrust Distributions  $\alpha=1.5$  in (Run 176)**

Both plots show general agreement between computational and experimental results, though while the computations show an under-prediction, the general shape of distributions is well matched.

Above and beyond the experimental results, computationally it is possible to investigate some things that are otherwise difficult to do experimentally. For example, shown below is a plot of the oscillation of lift and drag on the wing for of one propeller revolution. This gives further insight into the effect of the propeller on the wing.



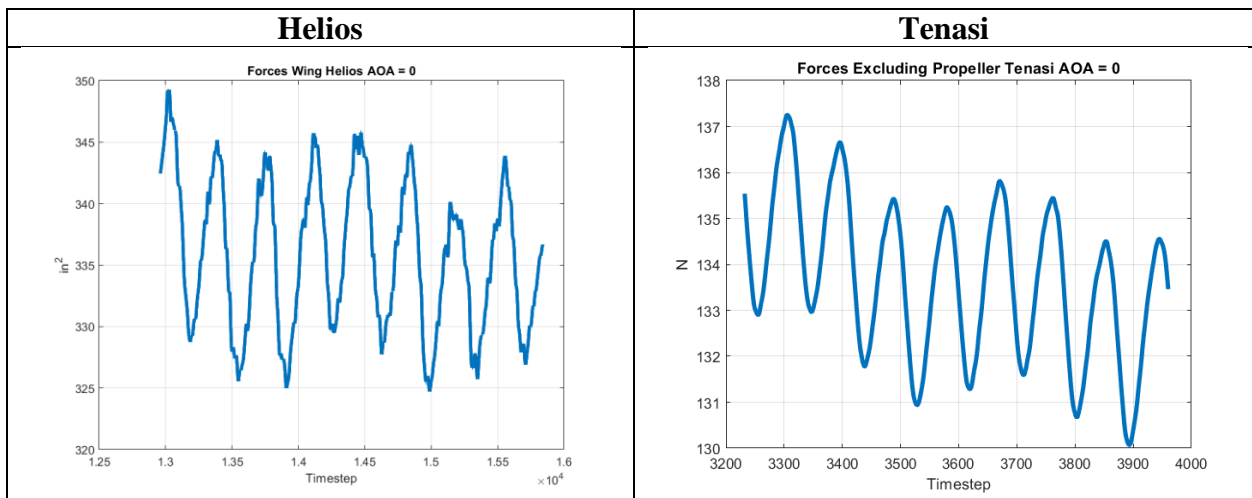
**Fig 14. Lift and Drag Oscillation on the Wing (Run 176)**

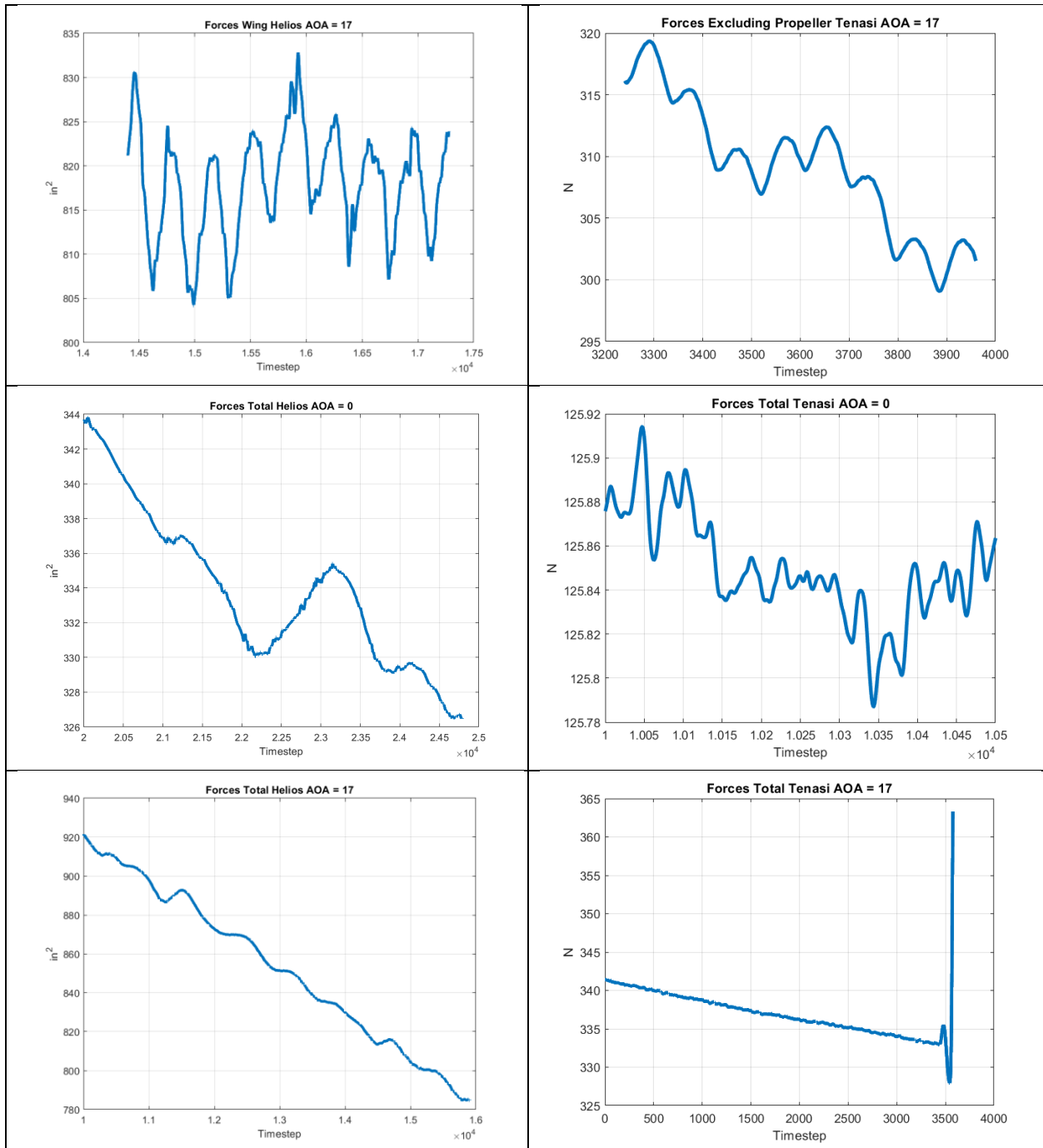
## 6.2 Priority One Results

For each priority one case shown below, the computational results will be compared directly with the wind tunnel test data. Finally, a detailed comparison between Tenasi and Helios will be performed considering the meshing process, the computational resources required, and overall accuracy of the results.

The first component of the results to be examined is the convergence. To have a successful simulation one of the most basic things that must occur is residuals and forces must converge. Typically, residuals converge much easier than forces. Below is a table comparing some of the convergence of the force-based values of each case. The first four plots represent the values calculated for the propeller cases. The last four plots represent the values calculated for the steady-state cases with no propeller. The values displayed in all plots represent the lift component of the forces.

**Table 13: Lift Force Convergence Comparison**





Its important to note the slight differences between the cases shown. Firstly, the units used on the Helios side are inches squared as opposed to Newtons in all Tenasi plots. This is because it has been partially non-dimensionalized. For the cases with the propeller it is easy to note the regular oscillations in all cases. This is a result of the propeller wake that crashes into the wing at regular

intervals. If viewed closely it is noticeable that these oscillations occur at the same frequency, 4 waves per revolution being 360 or 1440 timesteps for Tenasi and Helios simulations respectively. The convergence of this is different though, for example the Helios values at 0 degrees lines up very well but the 17-degree case is a bit more sporadic and nonuniform. This is due to the other flow features at work due to the large angle of attack.

This same trend can be seen in Tenasi cases. The Tenasi case at 17 degrees seems barely converged. Perhaps some more simulation time would improve this. It is also possible that more run time may not improve the force convergence due to inherent unsteadiness or the other components included in the force sum. The steady cases are slightly different with convergence being denoted by a value that is largely unchanging over many iterations. With these cases, the zero-degree simulations converge much better than the 17-degree cases. For example, the steady zero-degree Tenasi case shown above varies less than 2 newtons over 500 iterations. The same case at 17 degrees, however, blows up, having a large increase in lift force before the case dies. The cases in between largely experience force convergence behavior that falls in-between these cases. There is the occasional steady-state simulation that will die unexpectedly at a lower angle of attack. This is likely due to a slight unsteady flow feature that eventually effects a large section of the simulation due to numerical error causing it to die.



**Table 14: Tenasi Case No Prop M=0.08 CT = 0**

<b>Case AOA</b>	<b><math>C_D</math> Experimental</b>	<b><math>C_D</math> Tenasi</b>	<b>Error</b>	<b><math>C_L</math> Experimental</b>	<b><math>C_L</math> Tenasi</b>	<b>Error</b>
<b>0</b>	0.0324	0.0450	38.9%	0.5653	0.5910	4.56%
<b>5</b>	0.0619	0.0793	28.2%	0.9516	0.9617	1.07%
<b>7</b>	0.0751	0.0983	31.0%	1.0271	1.1028	7.37%
<b>15</b>	0.1441	0.2067	43.5%	1.2812	1.5011	17.2%
<b>17</b>	0.1809	0.2381	31.6%	1.3117	1.5681	19.5%

The table above shows the results from the Tenasi cases with no propeller. The trends of the data show the increasingly poor prediction of lift at higher angles of attack. This is typical as flow features such as separation begin to occur, and the flow becomes more difficult to model using a steady method. The  $C_L$  calculated for all cases is an overprediction, however this error gets worse as the angle increases. The drag of these cases fluctuates and is predicted rather poorly. Like lift, drag is always overpredicted. It is possible a transition model implemented into this case could help improve the drag predictions. For such a small wing the location of the transition could play an important role in drag prediction.

**Table 15: Tenasi Propeller Cases M=0.08 CT=0.4**

<b>Case</b>	<b><math>C_D</math> Thrust</b>	<b><math>C_D</math> Thrust</b>	<b>Error</b>	<b><math>C_D</math> Thrust</b>	<b><math>C_D</math> Thrust</b>	<b>Error</b>	<b><math>C_L</math> Exp</b>	<b><math>C_L</math> Tenasi</b>	<b>Error</b>
<b>AOA</b>	<b>Included</b>	<b>Included</b>		<b>Removed</b>	<b>Removed</b>				
	<b>Exp</b>	<b>Tenasi</b>		<b>Exp</b>	<b>Tenasi</b>				
<b>0</b>	-0.3757	-0.3094	17.6%	0.0247	0.04067	64.4%	0.6999	0.6274	10.4%
<b>5</b>	-0.3453	-0.2835	17.9%	0.0528	0.0674	27.7%	1.1367	1.0394	8.56%
<b>7</b>	-0.3276	-0.2671	18.5%	0.0663	0.0844	27.2%	1.2326	1.1776	4.46%
<b>15</b>	-0.2299	-0.1596	30.6%	0.1451	0.1969	35.8%	1.6104	1.5375	4.53%
<b>17</b>	-0.1817	-0.0974	46.4%	0.1875	0.2605	39.0%	1.6416	1.5472	5.75%

The table above shows some different trends than no propeller cases. These cases differ in that lift is now under predicted. This is likely a result of poor resolution of the propeller wake. As the flow speed increases due to the propeller, the sections of the wing in the downwash experience a higher velocity and should experience more lift. If the wake of the propeller is predicted poorly this could cause indirect inaccuracies in the lift. This case, however, performs better with respect to lift than the no propeller cases. The lift error trend, however, is inverted with the worst prediction of lift occurring at the lowest angle of attack. The thrust removed drag is very similar to the cases without a propeller. The same amount of error with no definitive trend. Lastly, the thrust included drag is predicted fairly well but begins to decrease in accuracy with an increase in the angle of attack. This value is always underpredicted indicating that the wake of the propeller is not accurately captured.

**Table 16: Helios Case No Prop M=0.08 CT=0**

<b>Case AOA</b>	<b><math>C_D</math>Experimental</b>	<b><math>C_D</math>Helios</b>	<b>Error</b>	<b><math>C_L</math>Experimental</b>	<b><math>C_L</math>Helios</b>	<b>Error</b>
<b>0</b>	0.0324	0.0285	12.0%	0.5653	0.4941	12.6%
<b>5</b>	0.0619	0.0614	0.755%	0.9516	0.8710	8.46%
<b>7</b>	0.0751	0.0717	4.49%	1.027	0.9522	7.29%
<b>15</b>	0.1441	0.1647	14.3%	1.2812	1.185	7.51%
<b>17</b>	0.1809	0.2003	10.7%	1.3117	1.270	3.21%

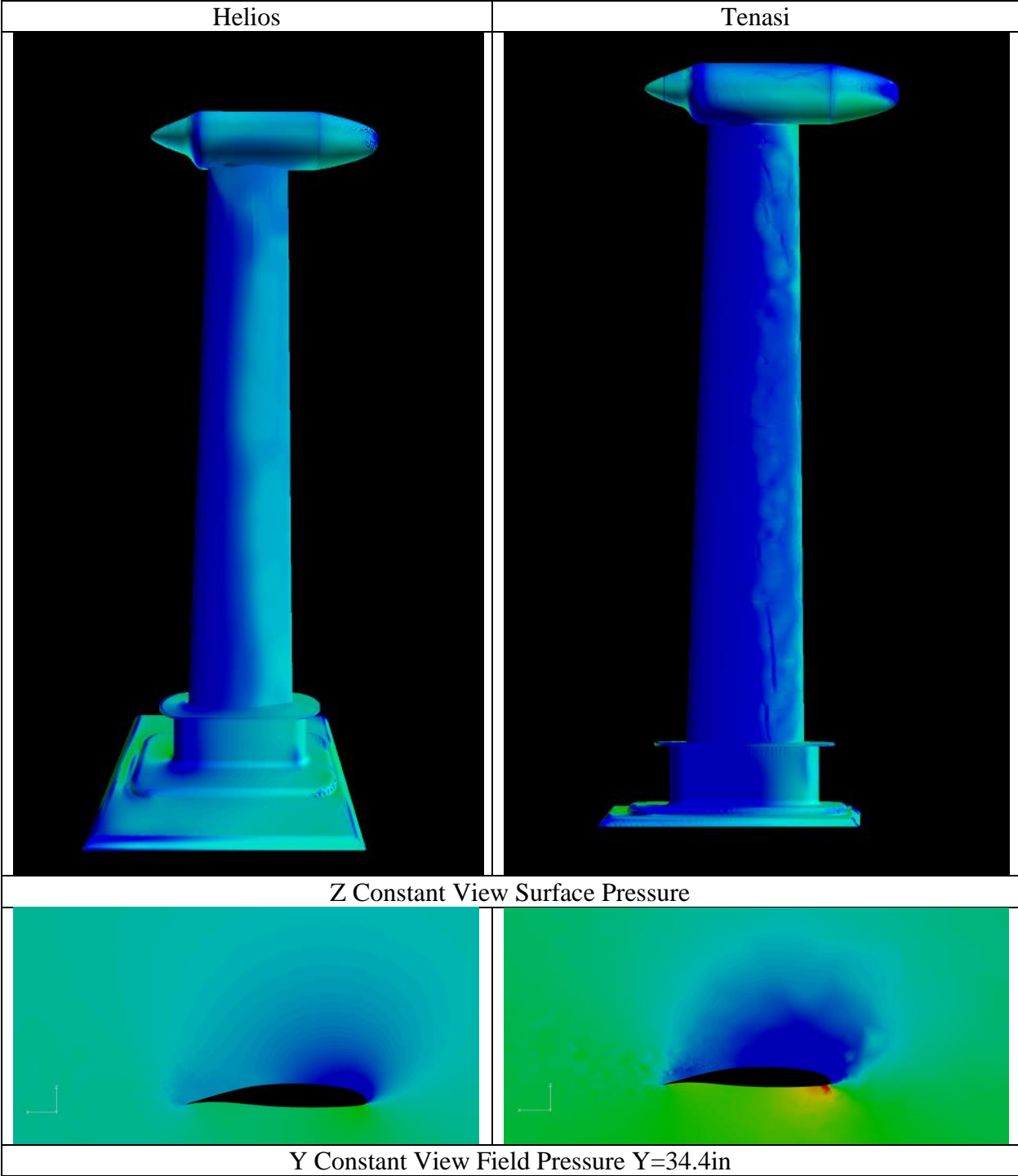
The table above shows the error in lift and drag. The trends in the lift are opposite Tenasi predictions. Lift here is underpredicted while Tenasi overpredicts. Helios improves with the angle of attack while Tenasi degrades. The accuracy of the lift is similar with around 5-10% error with most cases. Drag, however, is predicted better with Helios with error values ranging from 1%-14% compared with 28%-43% with Tenasi cases. Drag performs increasingly well as it transitions from overprediction to under prediction. This could be due to the transition point. It seems to be calculated well at 5 degrees but shifts in either direction as the angle of attack changes.

**Table 17. Helios Cases Propeller M=0.08 CT=0.4**

<b>Case</b>	$C_D$	$C_D$	<b>Error</b>	$C_D$ Thrust	$C_D$ Thrust	<b>Error</b>	$C_L$	$C_L$	<b>Error</b>
<b>AOA</b>	<b>Thrust</b>	<b>Thrust</b>		<b>Removed</b>	<b>Removed</b>		<b>Exp</b>	<b>Helios</b>	
	<b>Included</b>	<b>Included</b>		<b>Exp</b>	<b>Helios</b>				
	<b>Exp</b>	<b>Helios</b>							
<b>0</b>	-0.3757	-0.3747	0.269%	0.0247	0.0440	77.9%	0.6999	0.5336	23.7%
<b>5</b>	-0.3453	-0.3361	2.66%	0.0528	0.0808	53.1%	1.1367	0.9524	16.2%
<b>7</b>	-0.3276	-0.3188	2.67%	0.0663	0.0977	47.3%	1.2326	1.1196	9.18%
<b>15</b>	-0.2299	-0.2196	4.48%	0.1451	0.1873	29.1%	1.6104	1.504	6.61%
<b>17</b>	0.1875	-0.1761	3.01%	0.1817	.2303	22.9%	1.6416	1.4919	9.12%

The table above shows the predictions of lift and drag for the Helios propeller cases. The trend in lift and thrust removed drag mirrors the no propeller case but with worse accuracy. It is not exactly clear why these values are predicted so poorly. Similar to the Tenasi propeller case and Helios no propeller cases lift is still underpredicted. The positives of this simulation are its ability to very accurately predict the thrust included drag. Despite the inaccuracy of the thrust removed drag the thrust of the propellers is the dominant component and is predicted much more accurately than Tenasi.

Visualization can be a powerful tool when troubleshooting and explaining what is occurring in a simulation. In this case a sample visualization has been performed for the surface pressure of the No Propeller cases at 17 degrees. As noted previously Tenasi wildly overpredicts while Helios underpredicts. A visualization helps show why that is the case.

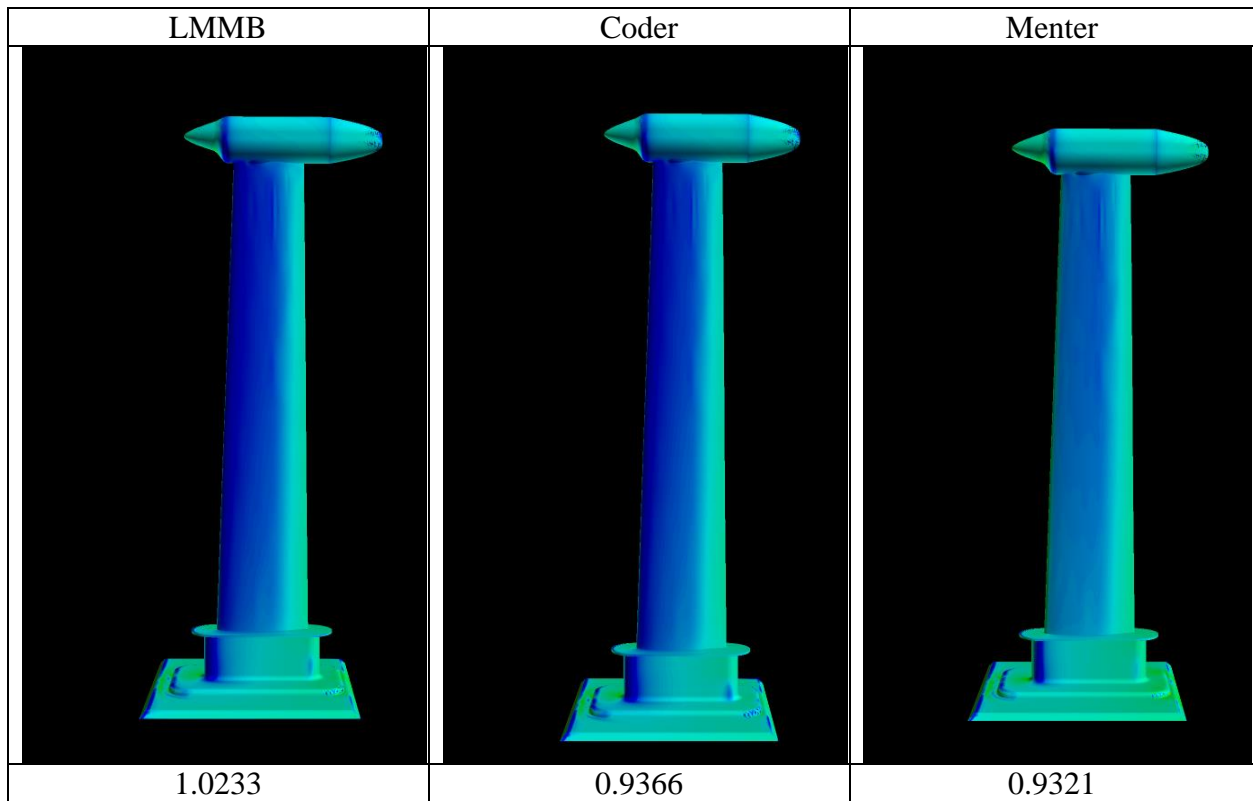


**Fig 15. Tenasi and Helios Visualized Pressure Comparison**

**Table 18. Helios Transition Model Comparison, 7° Angle of Attack M=0.08 CT=0**

<b>Transition Model</b>	$C_D$ Exp.	$C_D$ Comp.	Error	$C_L$ Exp.	$C_L$ Comp.	Error
<b>LMMB</b>	0.0751	0.0817	8.83%	1.027	1.0233	0.370%
<b>Coder</b>		0.0727	3.16%		0.9366	8.81%
<b>Menter</b>		0.0725	3.42%		0.9321	9.25%
<b>No Transition</b>		0.0717	4.49%		0.9522	7.29%

The table above shows the comparison of the three built-in transitions models and their effect on the lift and drag prediction for the no propeller case at a 7° angle of attack. For these cases all were set to run non-time accurately for 20000 steps. The Langtry-Menter, Medida-Baeder, denoted LMMB did not complete these 20000 steps and crashed due to a negative pressure at less than 14000 steps. Averages for lift and drag forces were taken over the last steady period of run time. The two codes that completed the simulation both improve on drag by a slim margin, slightly over 1% compared to the case with no transition model. While this is an improvement, both of these perform worse when it comes to the lift increasing the in accuracy by over 1% in both cases. The LMMB transition model is difficult to judge. Because it did not complete the simulation it is difficult to assess whether taking an average before forces diverged is credible. This brings the reliability of the lift and the drag calculations into question. In particular it seems hard to believe that the lift prediction accuracy is improved to less than 0.5% error, while the drag is predicted with double the error as the case with no transition model applied. Similar to the visualization above a visualization of surface pressure can help to show the differences in the transition model.



**Fig 16. Transition Model Comparison**

It is clear to see by the difference in pressure contours why the LMMB has the highest  $C_L$  and Menter has the lowest. Though the visualizations do not lend any empirical data they do help explain what is happening.

**Table 19. Tenasi Time Step Comparison M=0.08 CT=0.4 7° Angle of Attack**

	$C_D$ Thrust Included Exp	$C_D$ Thrust Included Helios	Error	$C_D$ Thrust Removed Exp	$C_D$ Thrust Removed Helios	Error	$C_L$ Exp	$C_L$ Helios	Error
<b>1°/Time Step</b>	-0.3276	-0.2671	18.5%	0.0663	0.0844	27.2%	1.2326	1.1776	4.46%
<b>0.25°/Time step</b>		-0.2607	20.4%		0.0894	34.7%		1.1569	6.14%

The table above shows the results from a side by side comparison of Tenasi cases with varying time steps. Both cases were simulated at a 7° angle of attack. The first represents the standard simulation at 1° per time step or 0.000027s per time step. The second case is 0.25° per time step or .00000699s per time step. The results are surprising. Both cases were run for 1800 steps steady and transitioned to unsteady for a total of 10 revolutions with lift and drag being averaged over the last revolution. Logically, a decrease in time step should yield an increase in accuracy. Unfortunately, this is not the case. In all categories, there drop in the accuracy with the smaller time step. The most likely cause of this is simply a lack of convergence. The forces may change slightly if the simulations are allowed to progress for more revolutions.

Currently the workshop data is unlabeled so the direct comparison of wake data through plots is impossible. Fortunately, the integration values are available and those will be compared below. The integrated values are taken from the first wake plane X=1.5 inches or 2.65 in behind the propeller plane and X=5 inches. The error is determined based on the predetermined thrust. This value is from the calibrated experimental results performed before the workshop. The integration was performed using the following formulas.



$$\frac{Thrust}{Q_{\infty} R_{prop}^2} = 2\pi \int_0^1 \frac{\rho u(u - u_{\infty})}{Q_{\infty}} \frac{r}{R_{prop}} d \frac{r}{R_{prop}}$$

$$\frac{Torque}{Q_{\infty} R_{prop}^3} = 2\pi \int_0^1 \frac{\rho u v_{swirl}}{Q_{\infty}} \left(\frac{r}{R_{prop}}\right)^2 d \frac{r}{R_{prop}}$$

Figure 17. Thrust and Torque Integration Formulas

Table 20: Thrust Integration Comparison M=0.08 CT=0.4

Thrust (lb)	Thrust Computational Integration	Thrust Rake Integrated	Thrust Predetermined	Error
Helios X=1.5	13.533	20.21	17.77	23.8%
Helios X=5	15.294			13.9%
Tenasi X=1.5	15.232			14.3%
Tenasi X=5	17.0910			3.8%

As shown above Tenasi provides a much better prediction of the Thrust than Helios. A likely cause of inaccuracy in Helios is that the solution was only exported and averaged every 72 steps or 18 degrees where as the Tenasi case was averaged every step or 1 degree. Both solvers are consistent however in underpredicting thrust. This is slightly odd since the workshop integrations over predict thrust. Interestingly across both solvers the second wake plane at X=5 inches provides a much more accurate prediction of thrust. The reason for this is unclear. It is possible that at a location closer to the propellers the flow from the thrust has not fully developed as opposed to a position

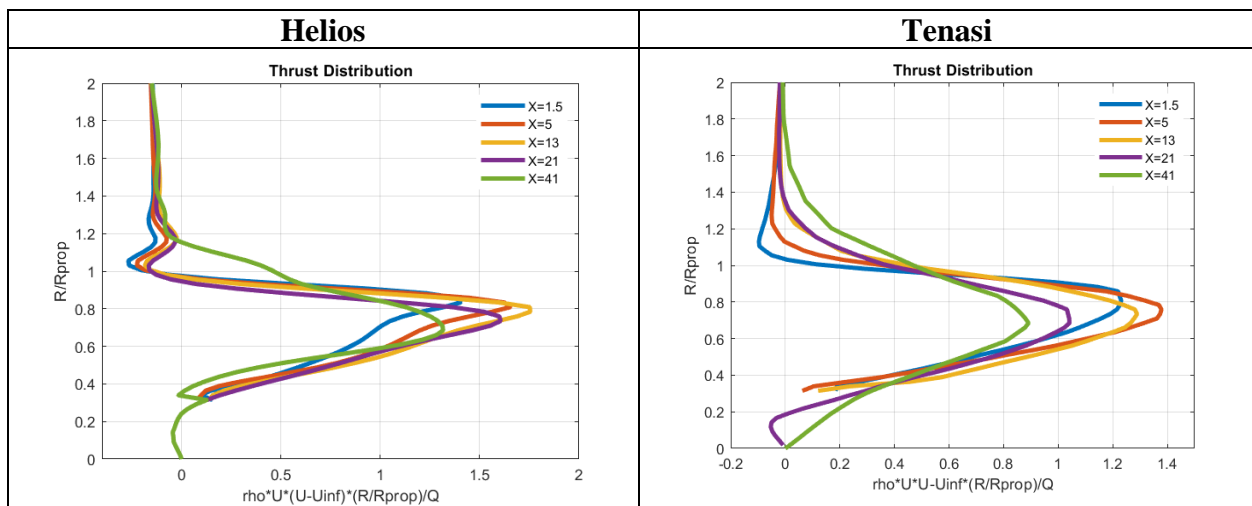
further downstream. Further investigation could be done to find the optimum location for thrust and torque integration.

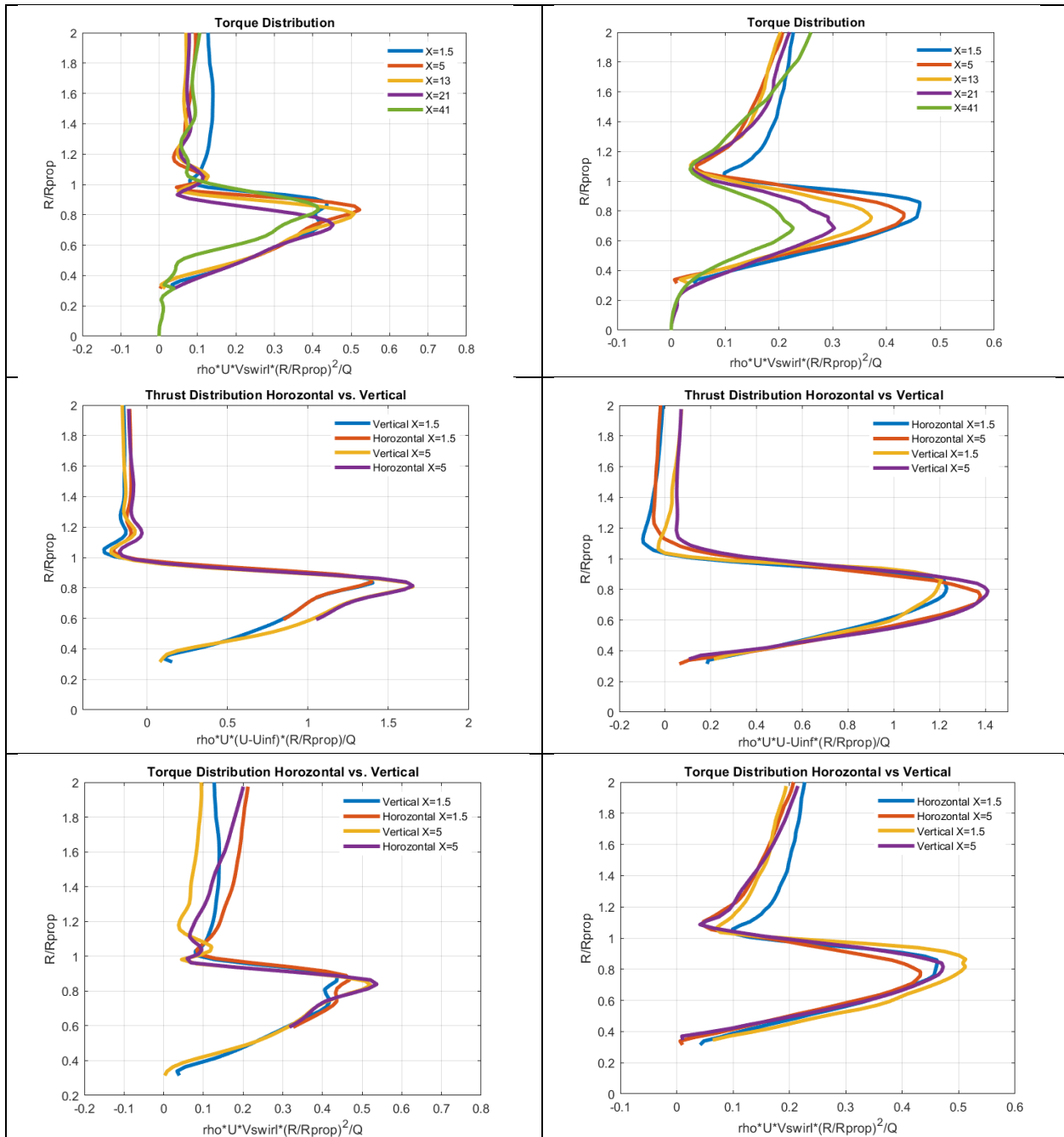
**Table 21: Torque Integration Comparison M=0.08 CT=0.4**

Torque (ft*lb)	Torque Computational Integration	Torque Rake Integration	Error
Helios X=1.5	3.5116	4.74	25.9%
Helios X=5	3.4295		27.7%
Tenasi X=1.5	3.9176		17.4%
Tenasi X=5	3.4707		26.8%

Similar to thrust, Tenasi performs better predicting torque. Again, this is most likely due to the larger period between values averaged. Unlike the previous case torque is not predicted better at the second wake plane. As mentioned, this could be due to the development of the flow.

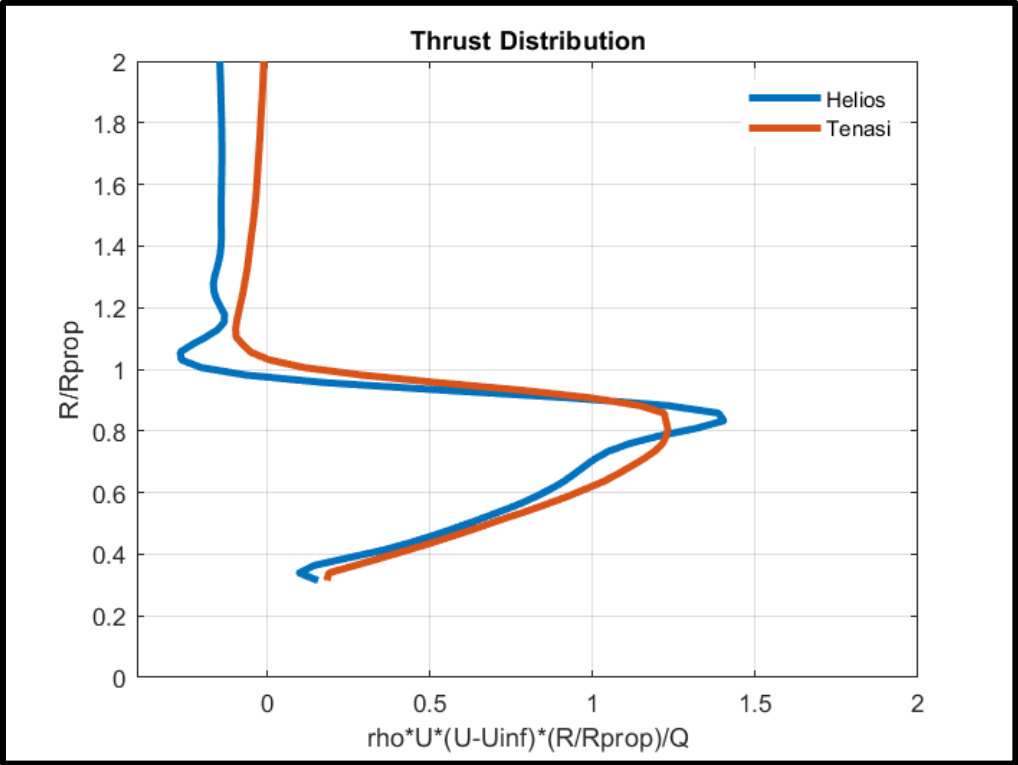
**Table 22: Wake Data Comparison**



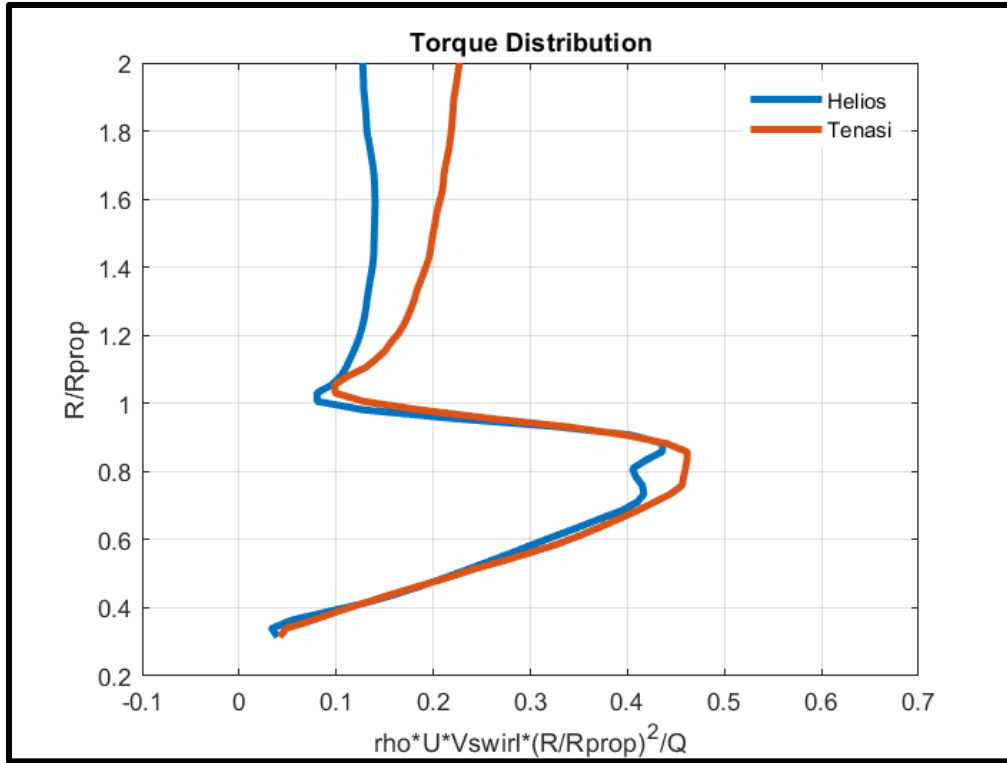


The images below show the direct comparison of the Helios and Tenasi wake data. Though taken using the same spatial discretization the shape of each profile varies slightly. The overall shape is very similar to Tenasi having a little smoother profile. It is also clear to see how the integration of the Helios profile would yield a slightly lower thrust and torque. Additionally, across both solvers

the profiles of the horizontal and vertical rake data are nearly identical indicating there is little bias associated with the positioning of the rake.

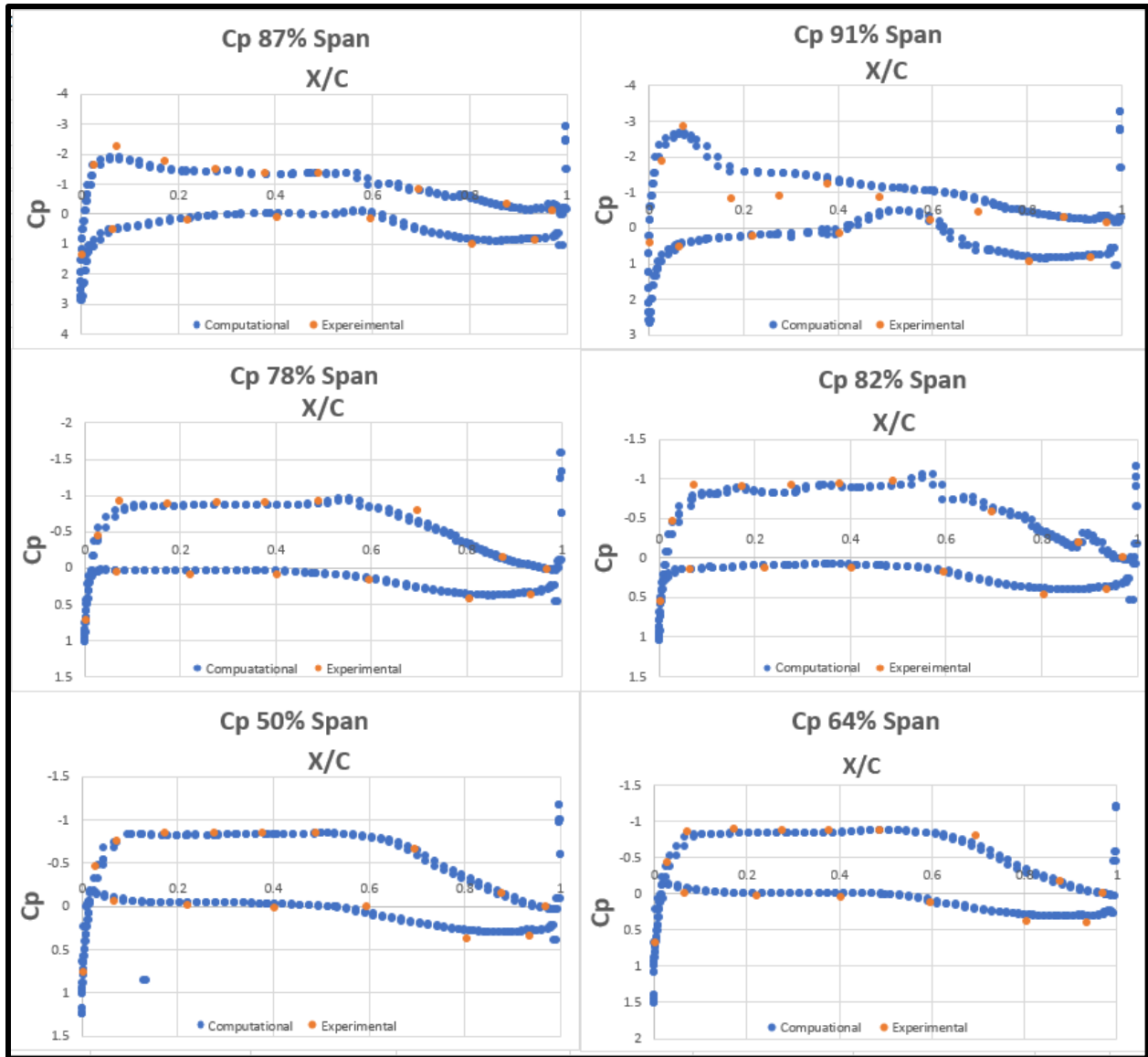


**Fig 18. Helios Tenasi Thrust Comparison X=1.5**



**Fig 19. Helios Tenasi Torque Comparison X=1.5**

The last pieces of data for these simulations are the  $C_p$  plots. Shown below are the  $C_p$  plots for the Tenasi Cases.



**Fig 20. Tenasi CT=0.4 Angle of Attack = 0°**

These  $C_p$  plots accurately predict the  $C_p$  values found experimentally. The slight spikes shown in the 82% Span location are a result of physical inconsistencies in the geometry. The worst predictions come at the 91% span. This location is incredibly close to the nacelle and as a result has the most complicated flow to predict.

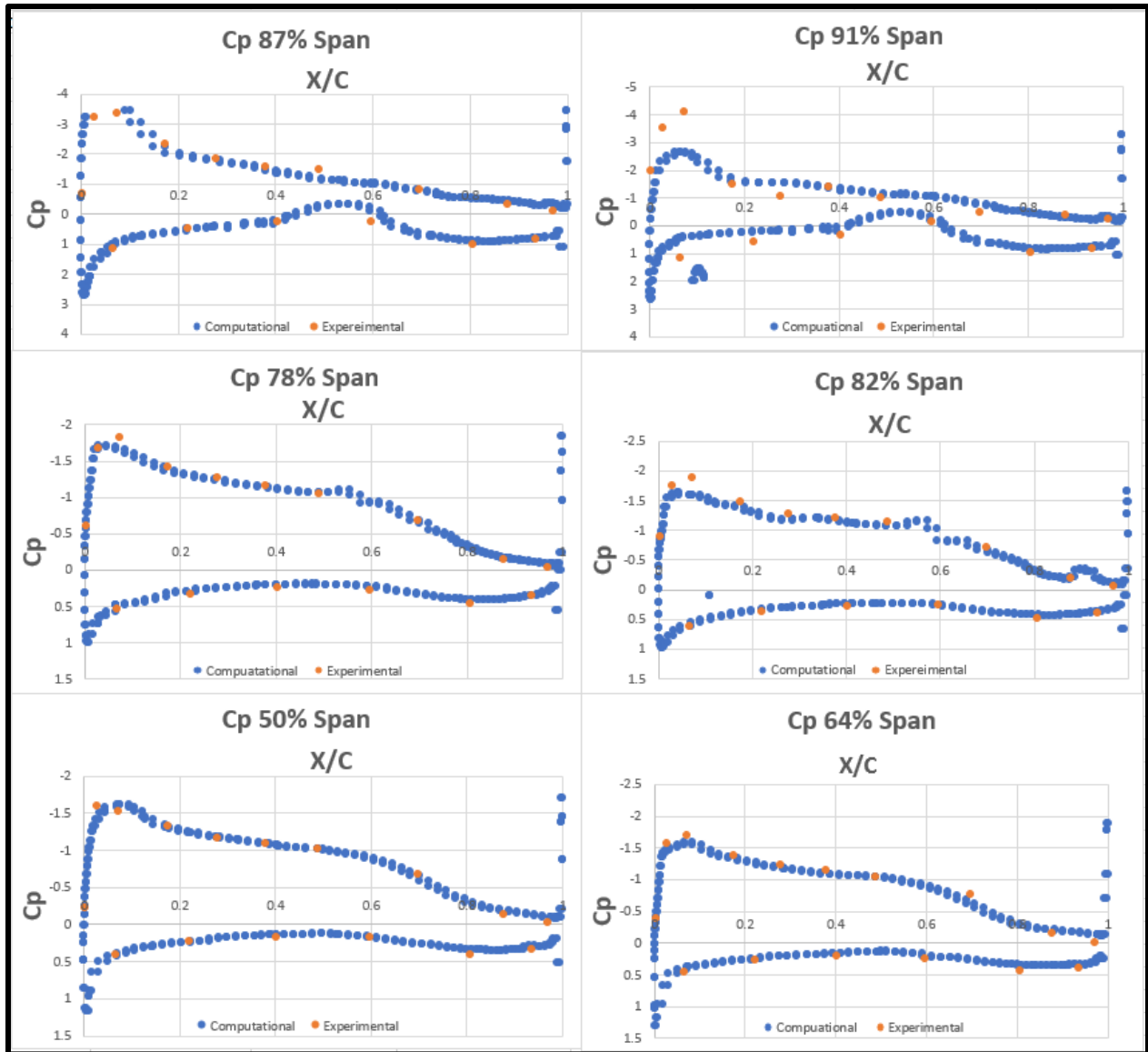


Fig 21. Tenasi  $CT=0.4$  Angle of Attack =  $5^\circ$

The plots shown above are for the angle of attack of 5 degrees. The simulations predict  $C_p$  rather well similar to the previous case. Again the most difficult pressure line is at the 91% span location. This  $C_p$  is even worse underpredicting  $C_p$  at the leading edge by almost 50%.

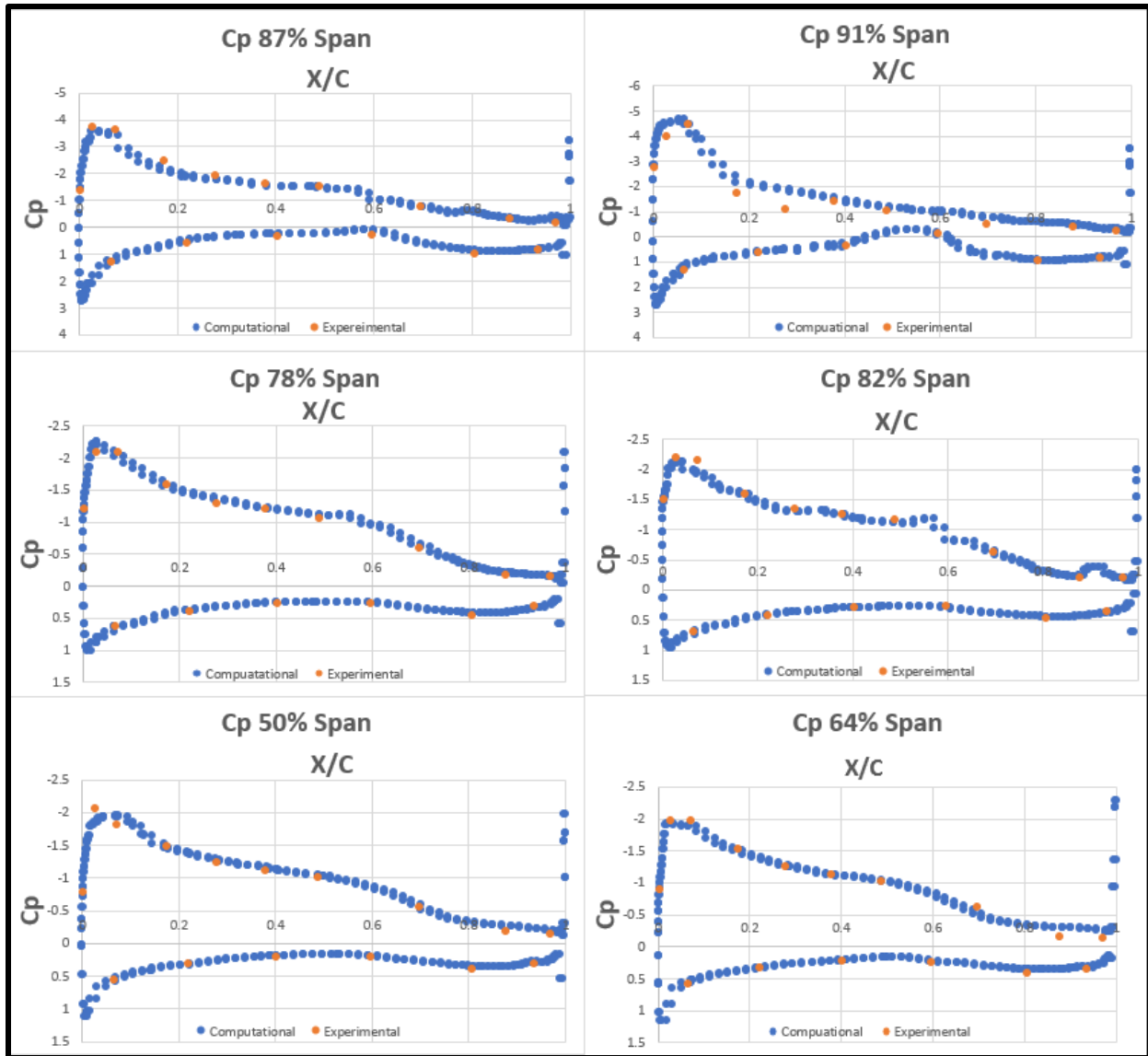
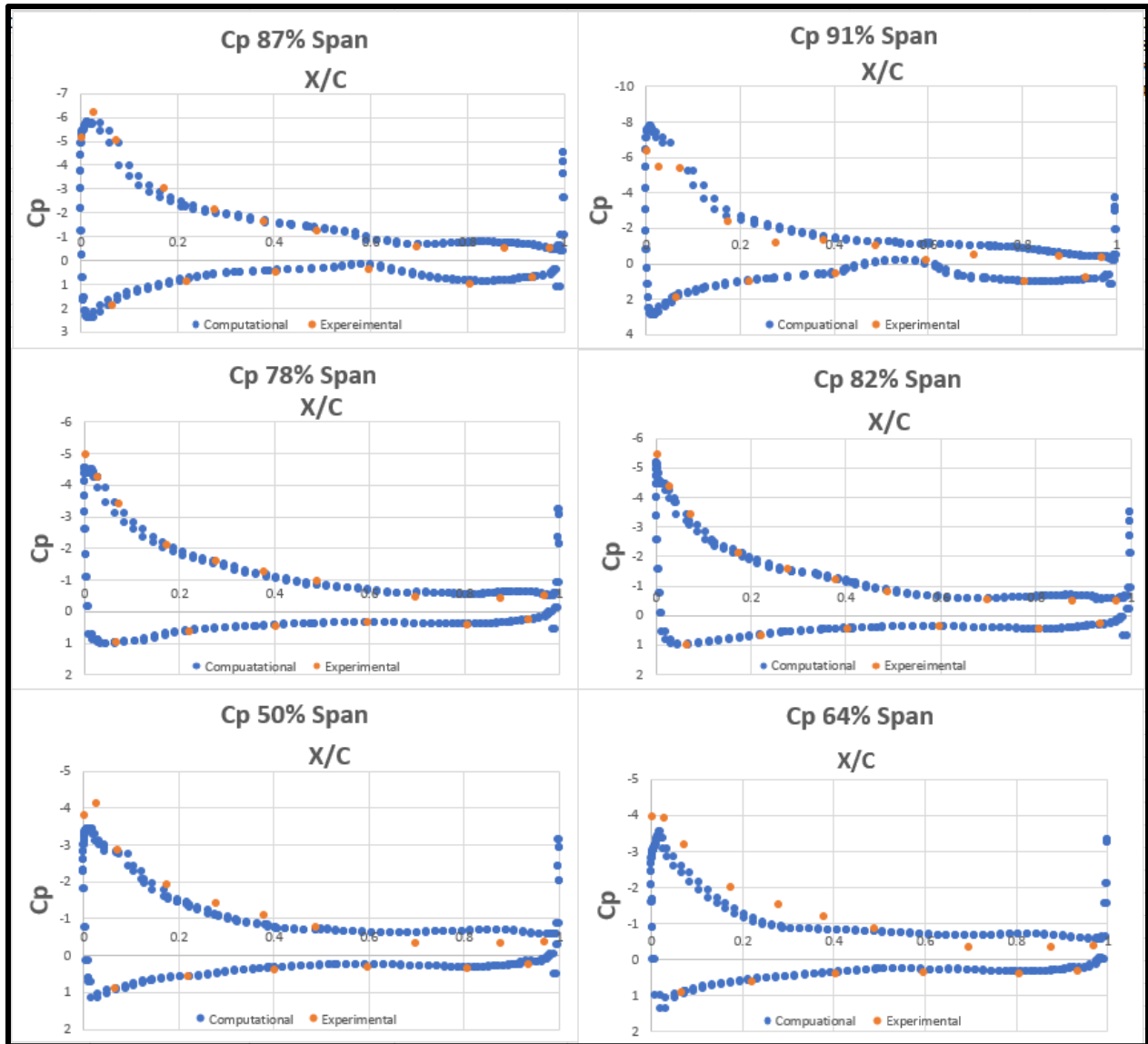


Fig 22. Tenasi CT=0.4 Angle of Attack = 7°

This case improves on the others.  $C_p$  is predicted better at the higher percent span locations. Otherwise it still predicts  $C_p$  well at the lower span locations.



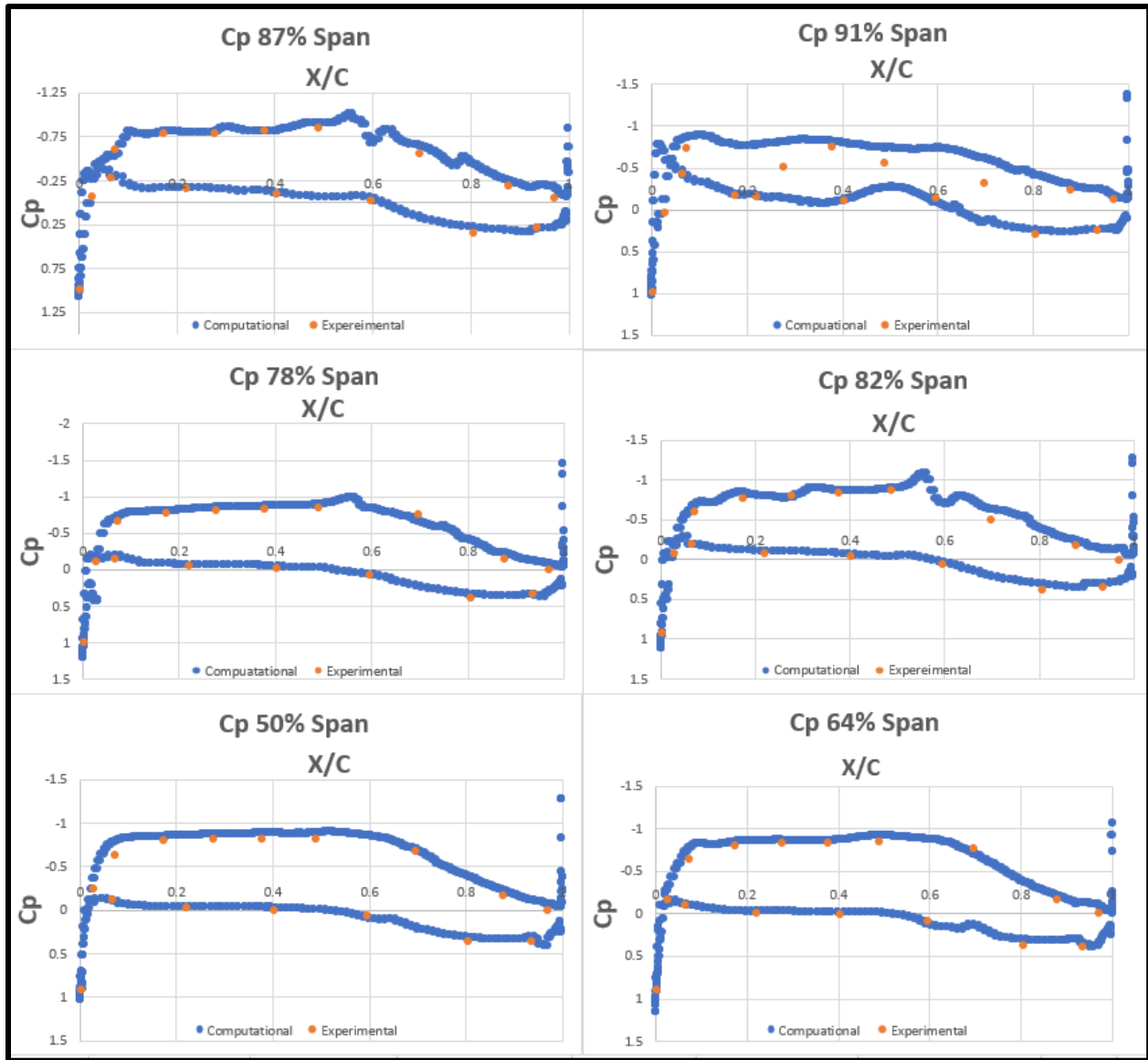


**Fig 23. Tenasi CT=0.4 Angle of Attack = 15°**

At this much higher angle of attack  $C_p$  is not predicted as well. There is a slight under-prediction at the lower percent span locations while there is a slight overprediction at the higher percent span locations. This is likely due to the difficulties with the high angle of attack. With higher angles of attack the  $C_p$  drops drastically on the suction side causing difficulties with the simulation. In extreme cases, absolute pressure can be calculated negative causing the simulation to stop.

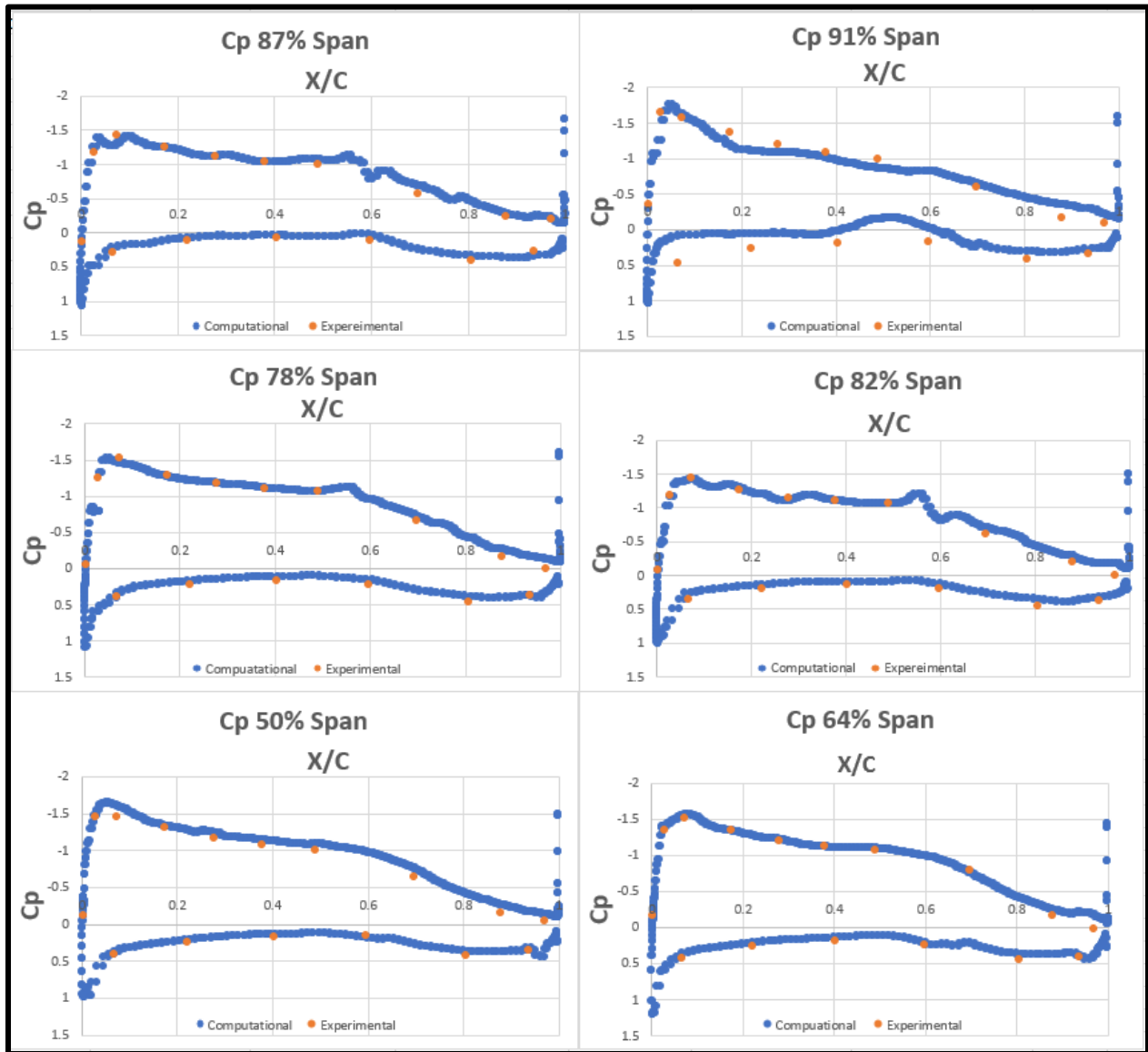


The following 5 images show the Tenasi  $C_p$  plots with no propeller or  $CT=0$



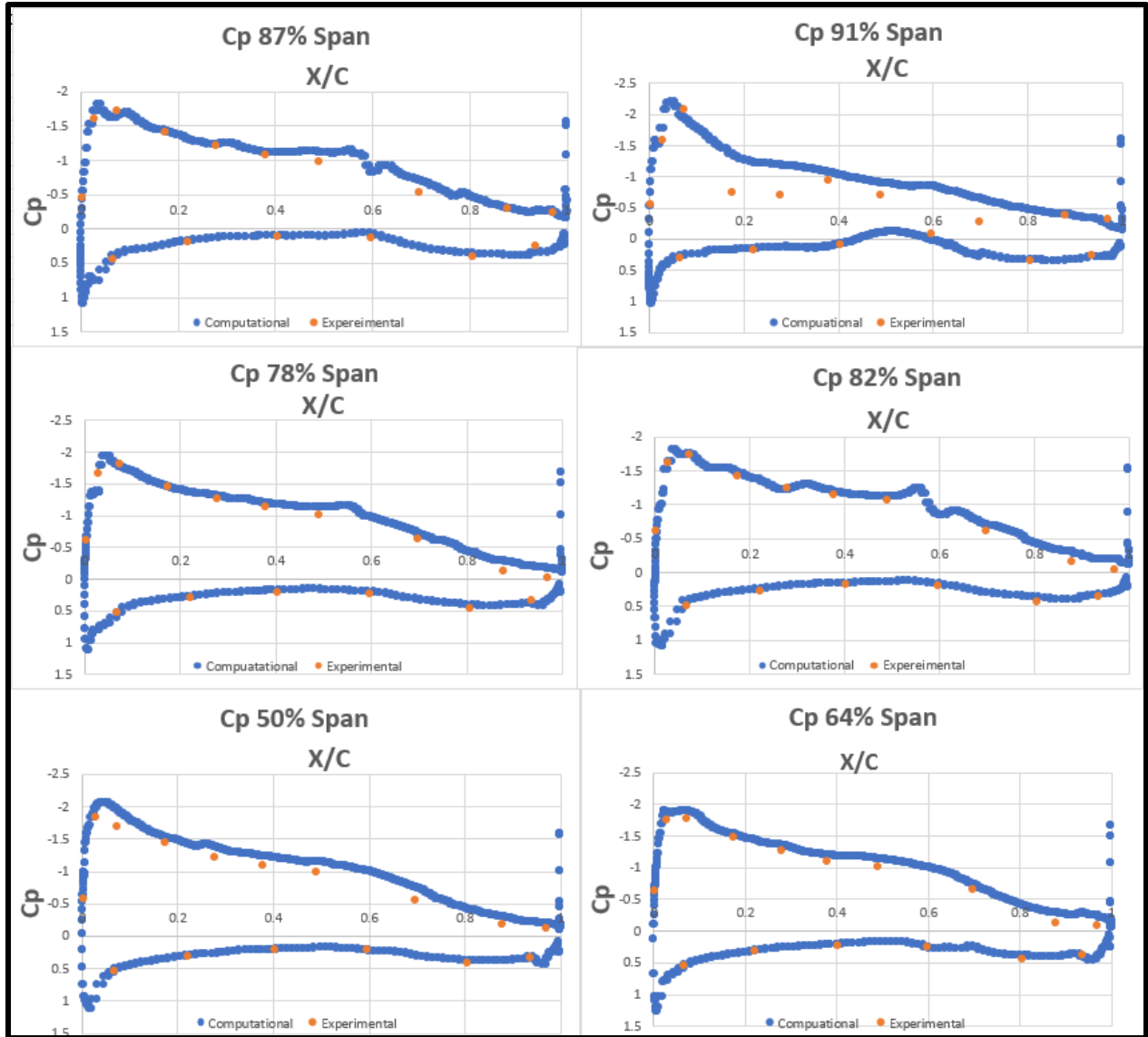
**Fig 25. Tenasi  $CT=0$  Angle of Attack =  $0^\circ$**

The case above shows good agreement between the experimental and computational data. Once again, the worst predicted  $C_p$  location is at the 91% span location. This indicates that some inaccuracy comes as a result of the geometry itself not only because of the propeller wake interaction.



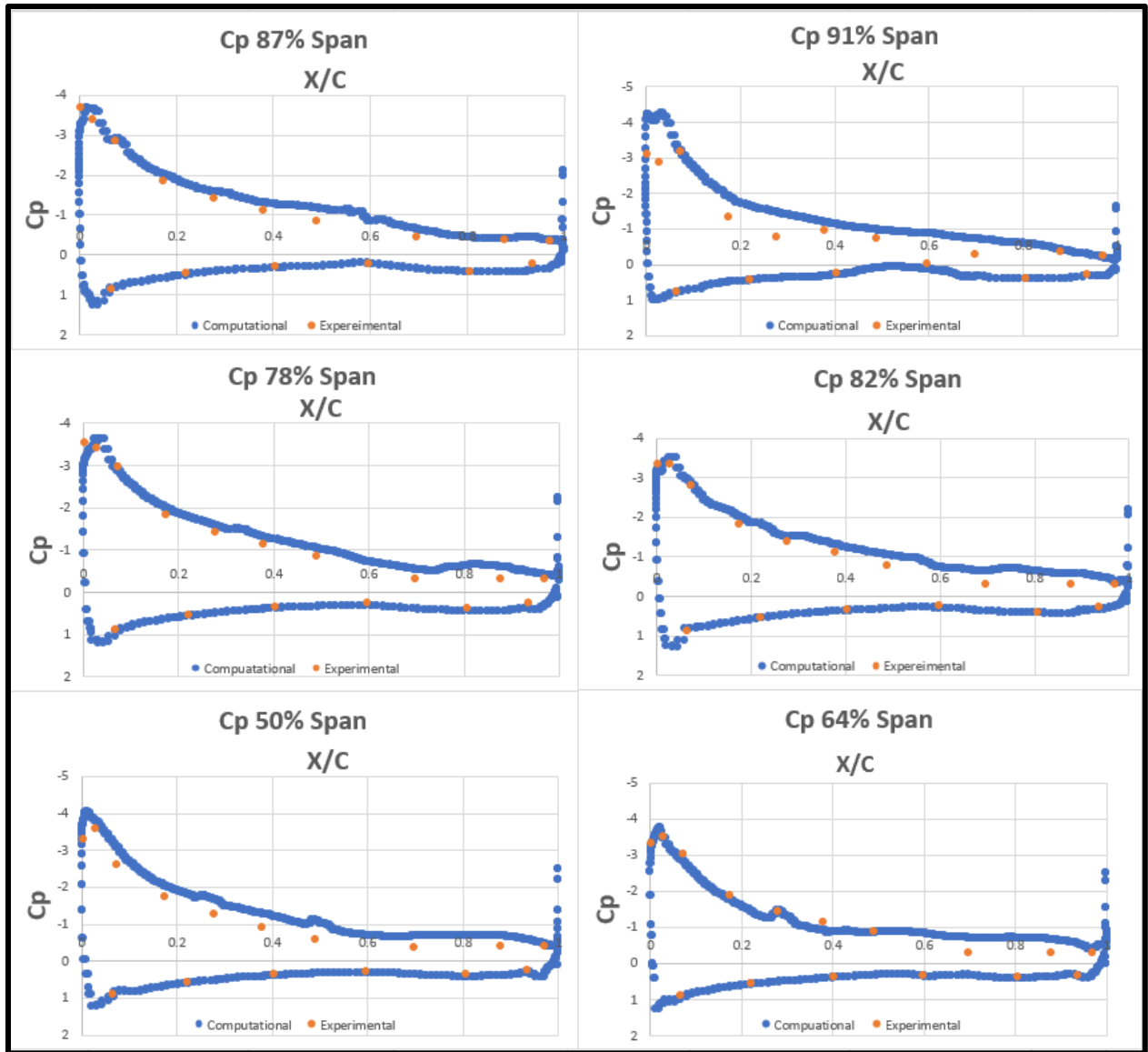
**Fig 26. Tenasi CT=0 Angle of Attack = 5°**

At an angle of attack of 5° the inaccuracies still lie at the highest pressure line. In this case there is inaccuracy both on the suction and pressure side. Until now all pressure sides have been predicted well. This could be the result of some more complicated flow conditions that are not modeled well.



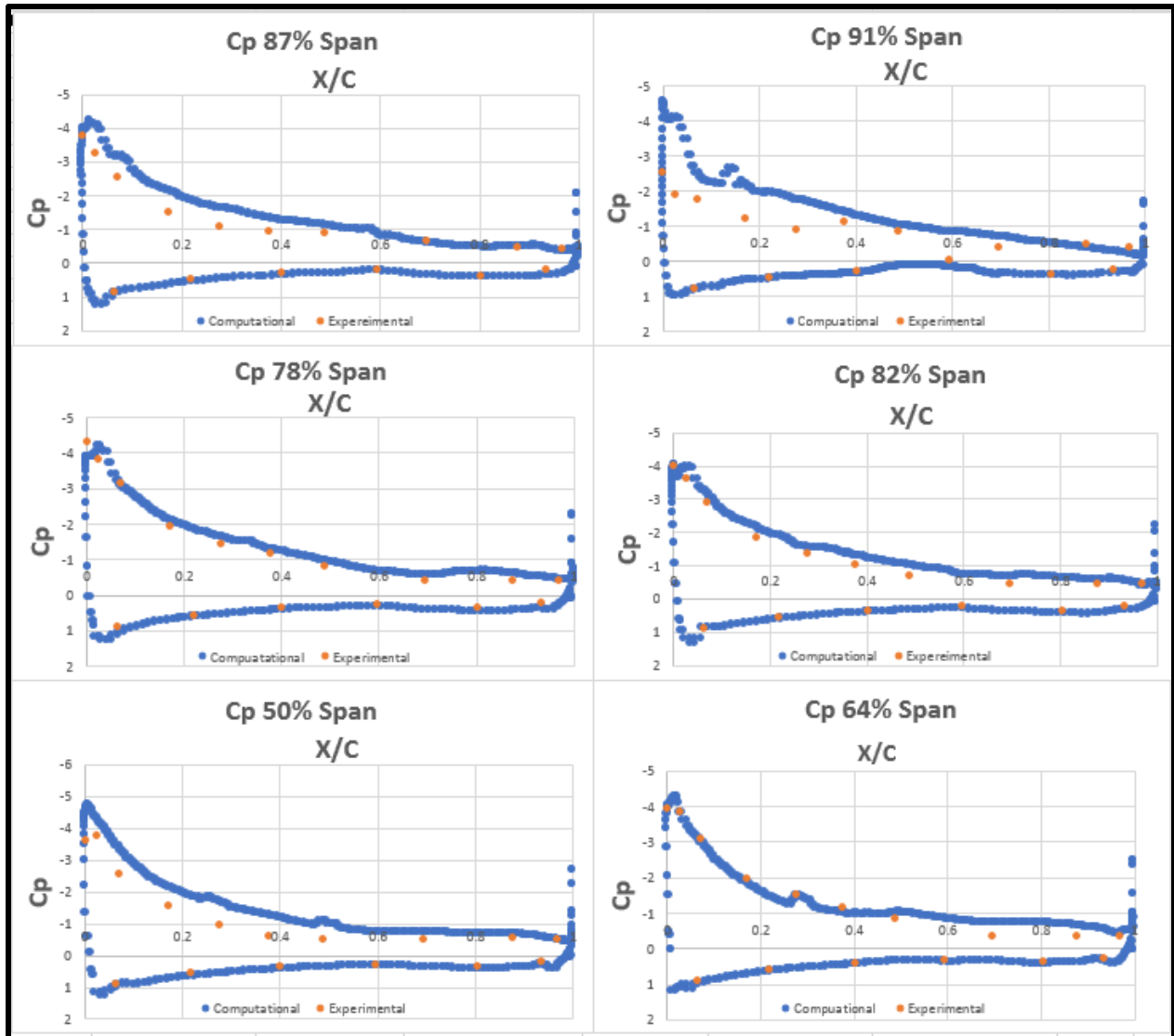
**Fig 27. Tenasi CT=0 Angle of Attack =  $7^\circ$**

Similar to  $5^\circ$ , this case has an increase in accuracy at the same span position, over predicting  $C_p$ . Additionally, the lower  $C_p$  lines seem to be more inaccurate with the steady cases compared to the unsteady cases. This indicates that a steady simulation is not a perfect model for this configuration especially at higher angles of attack.



**Fig 28. Tenasi CT=0 Angle of Attack = 15°**

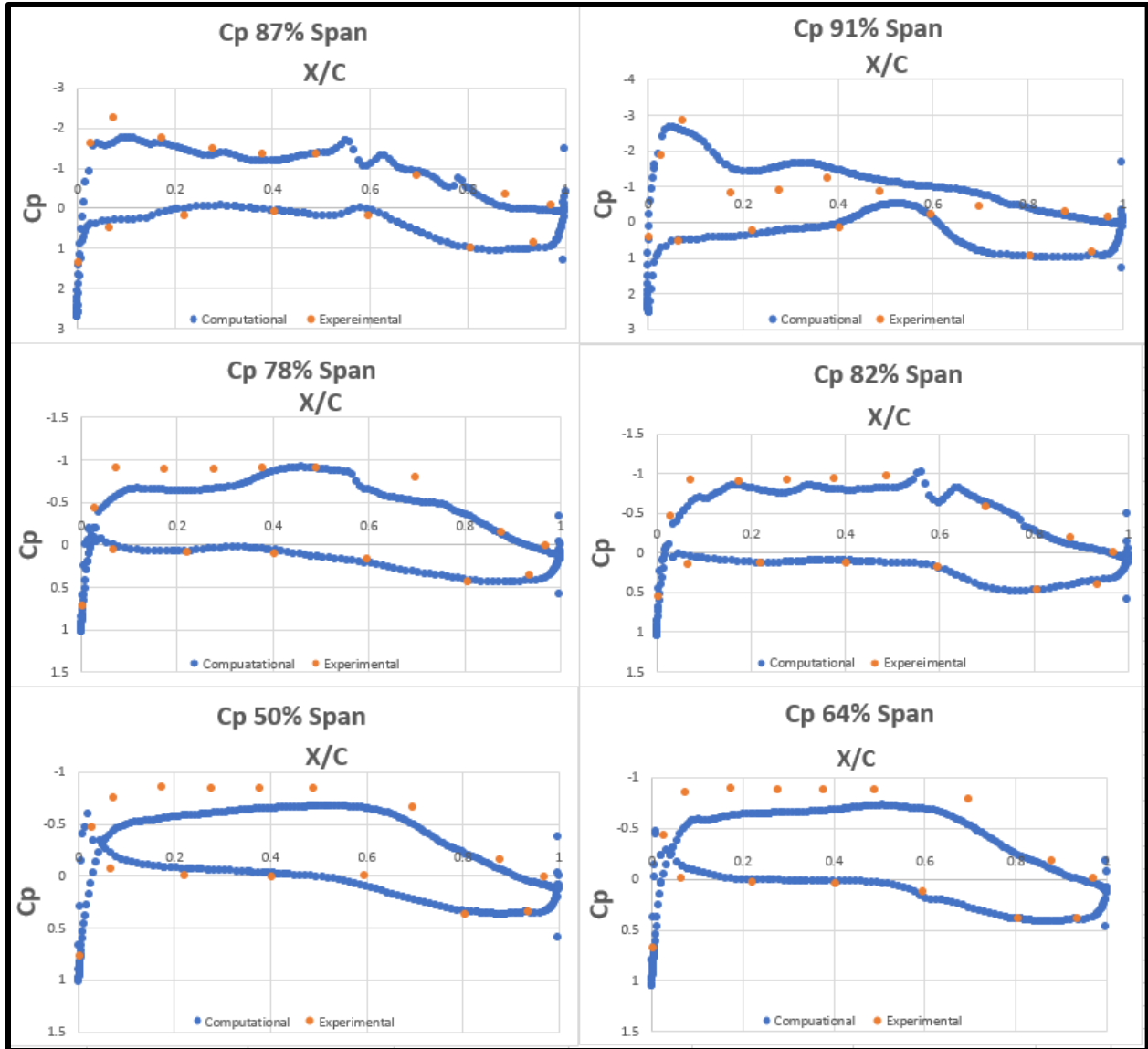
This 15° case continues the trend of slight inaccuracies at lower angles of attack with over-prediction at the higher span locations. This indicates the unsteady nature of the simulation at this high angle of attack.



**Fig 29. Tenasi CT=0 Angle of Attack =  $17^\circ$**

Continuing the same trends as before this angle shows the worst predictions of  $C_p$  for the Tenasi no propeller case.

The following 5 images show the  $C_p$  Plots for the Helios Cases where CT=0.4

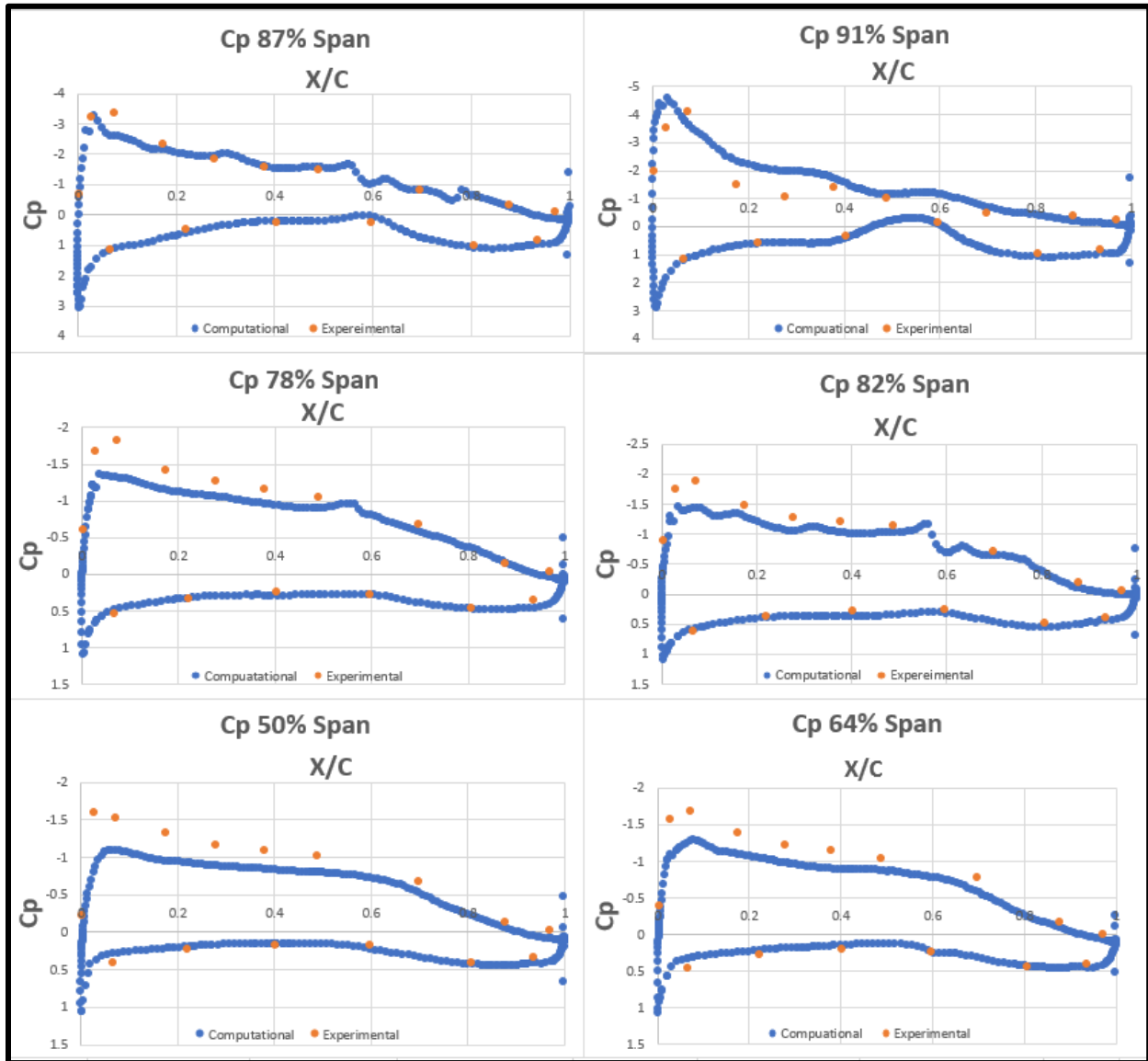


**Fig 30. Helios CT=0.4 Angle of Attack = 0°**

The  $C_p$  plots shown above show the poor prediction of  $C_p$  by the Helios solver. Contrary to the Tenasi cases  $C_p$  is predicted the best at the higher span locations. The poor prediction at the lower span locations is unexpected. It could be influenced by the large intervals from which the average was taken, however the lower span locations should be largely time-invariant since they are not

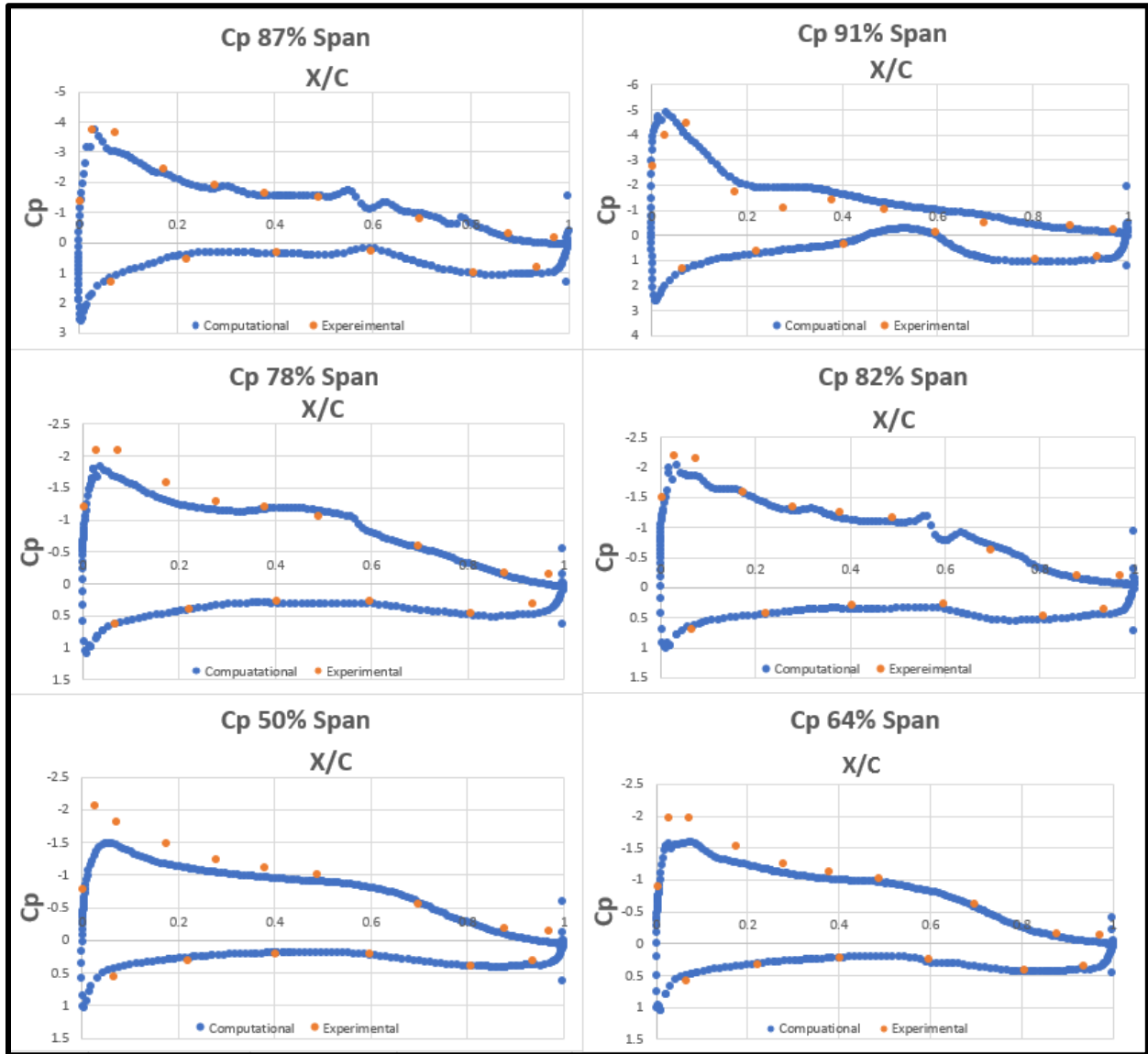


located near the downwash of the propeller and should experience conditions very similar to freestream



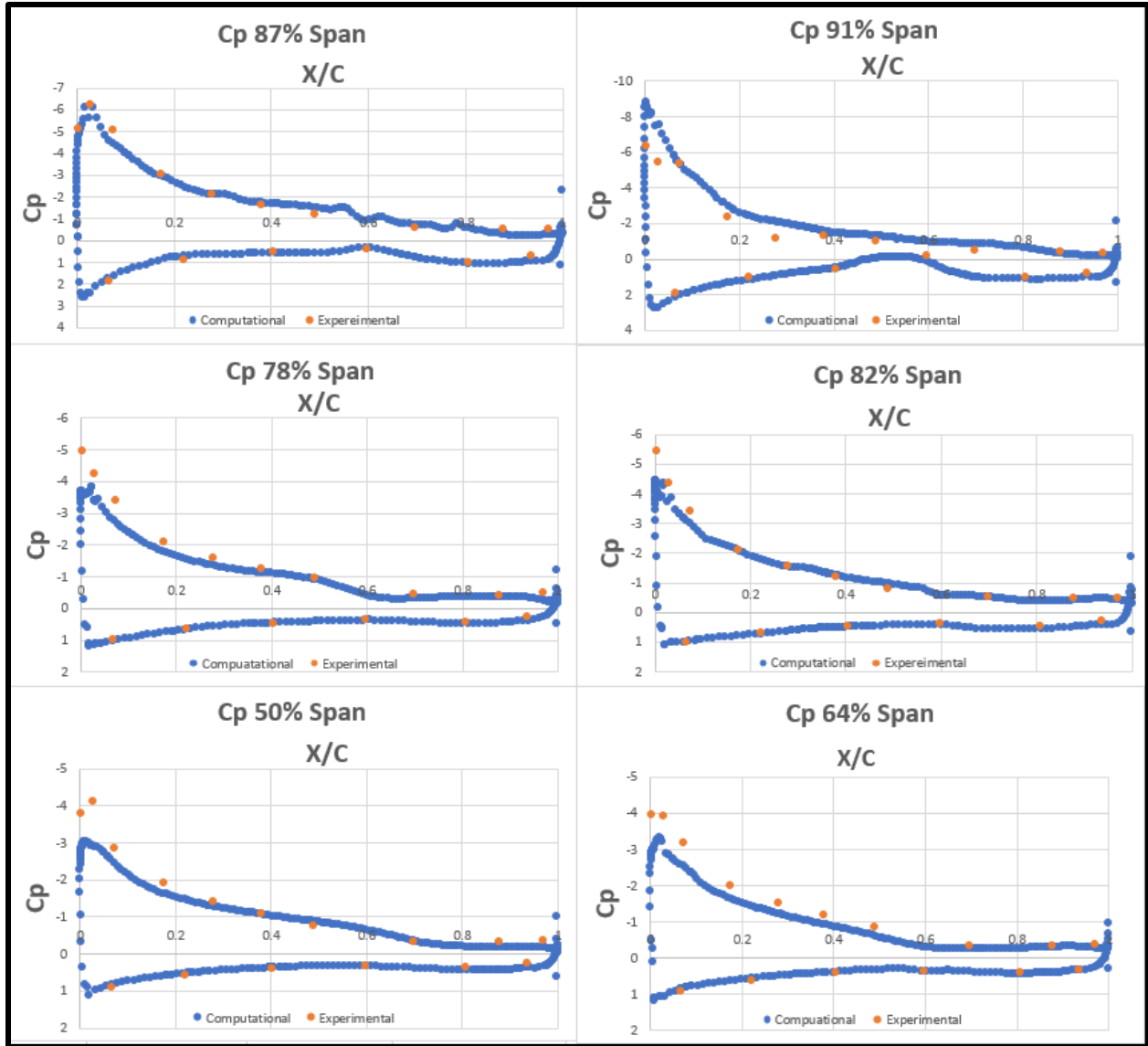
**Fig 31. Helios  $CT=0.4$  Angle of Attack =  $5^\circ$**

This case improves slightly at the lower span locations. Still this inaccuracy fits with the fact that lift is predicted poorly by Helios. Since lift is an integration of the pressure over the wing, it follows that if  $C_p$  is poorly predicted the integration of it would also be poorly predicted.



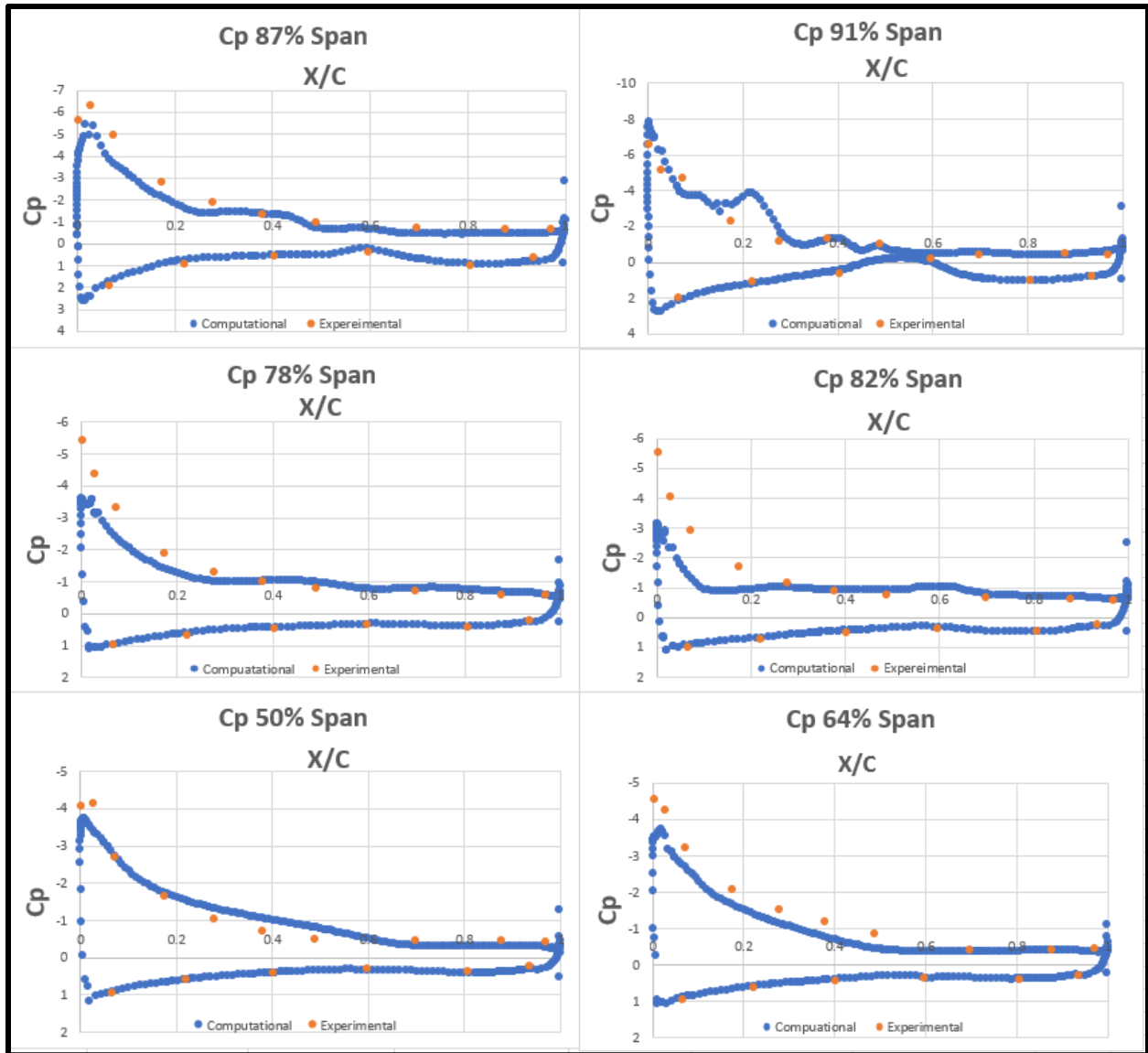
**Fig 32. Helios CT=0.4 Angle of Attack = 7°**

At 7°  $C_p$  is predicted better than the cases before, with the most inaccuracy occurring at the lowest span locations.



**Fig 33. Helios CT=0.4 Angle of Attack = 15°**

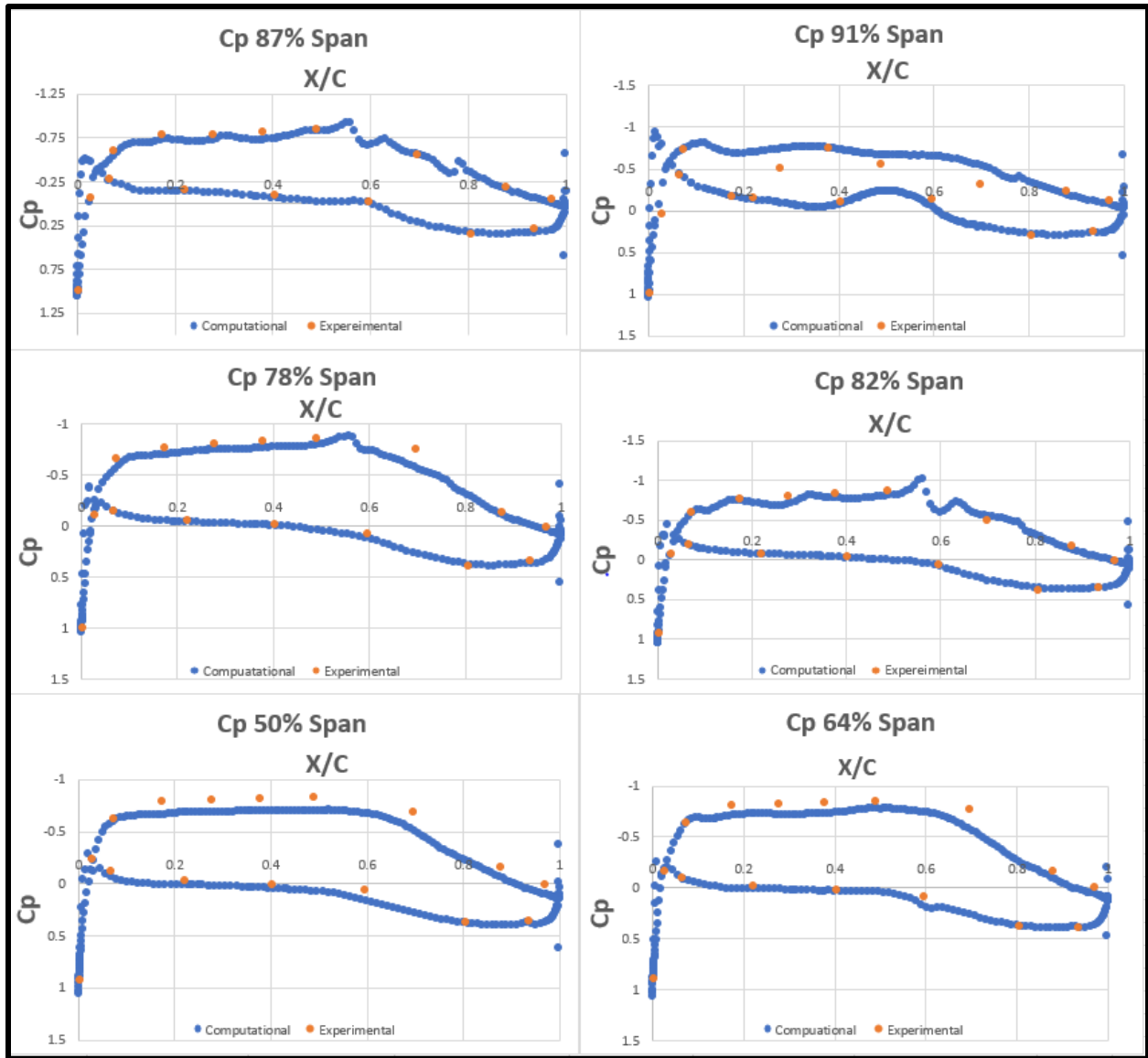
Once again there is a continued improvement at this higher AOA however, the inaccuracies show an underprediction of  $C_p$ . This under prediction fits with the underprediction of lift calculated previously.



**Fig 34. Helios CT=0.4 Angle of Attack = 17°**

This last location does the best at predicting the low span locations. The terrible inaccuracy at the 82% span is unknown. It could be a result of an odd interaction of the propeller or perhaps some localized separation.

The following 5 images show the  $C_p$  plots for the Helios cases,  $CT=0$



**Fig 35. Helios  $CT=0$  Angle of Attack =  $0^\circ$**

The steady case here shows some improvements compared with the unsteady Helios case. This also agrees with the improvement of accuracy of lift between the propeller and no propeller Helios cases. It is however less accurate than the Tenasi cases.

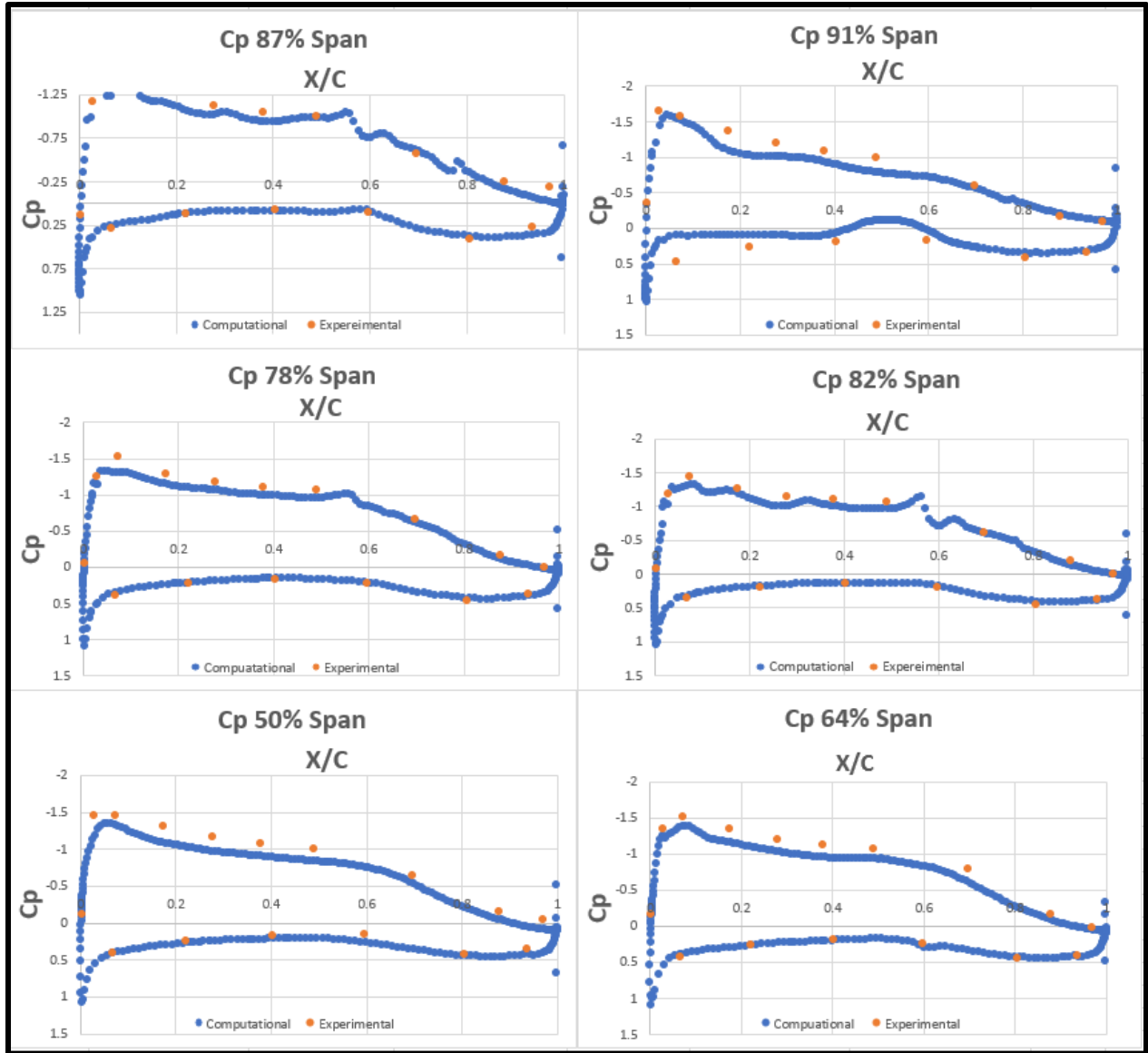
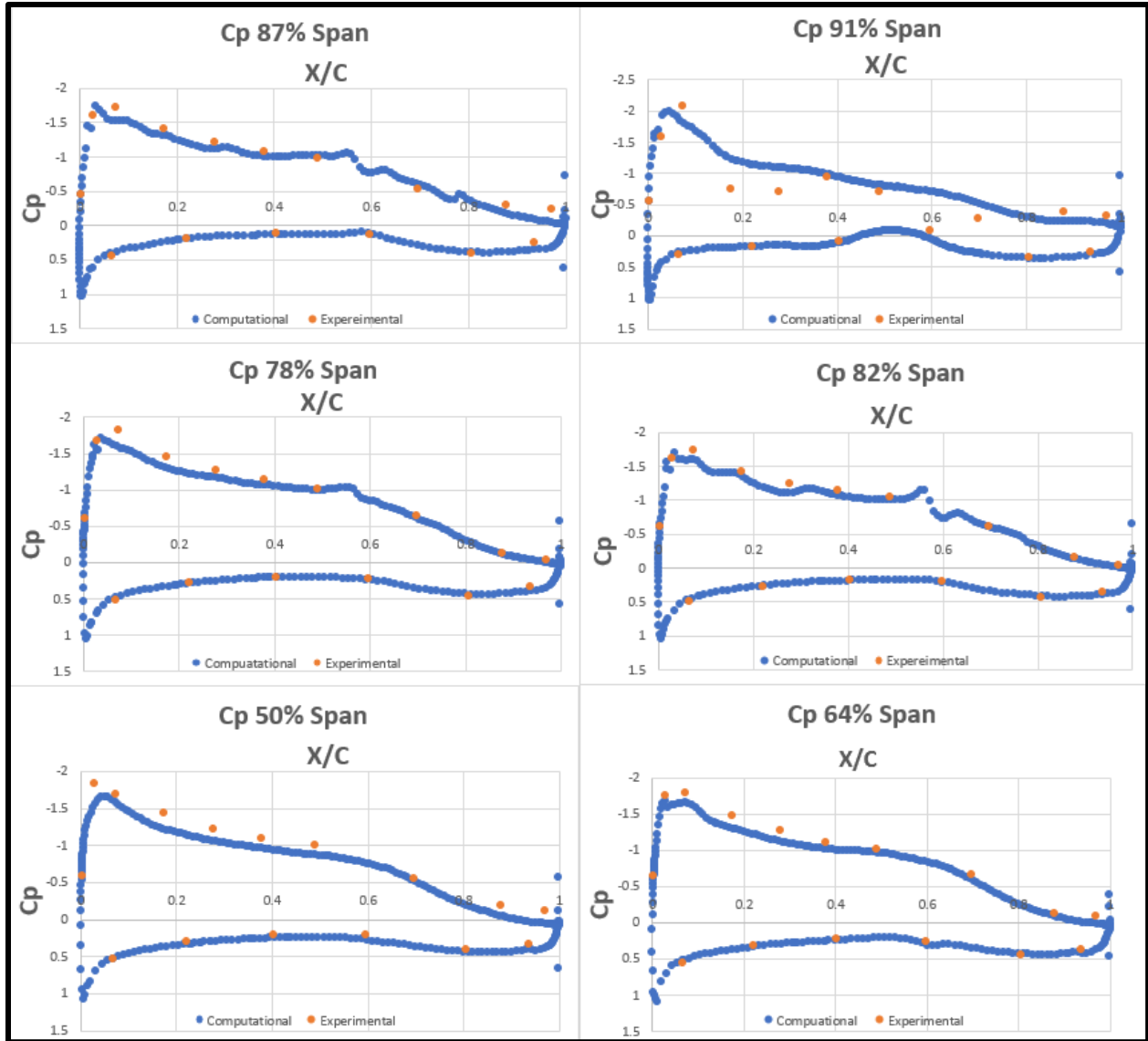


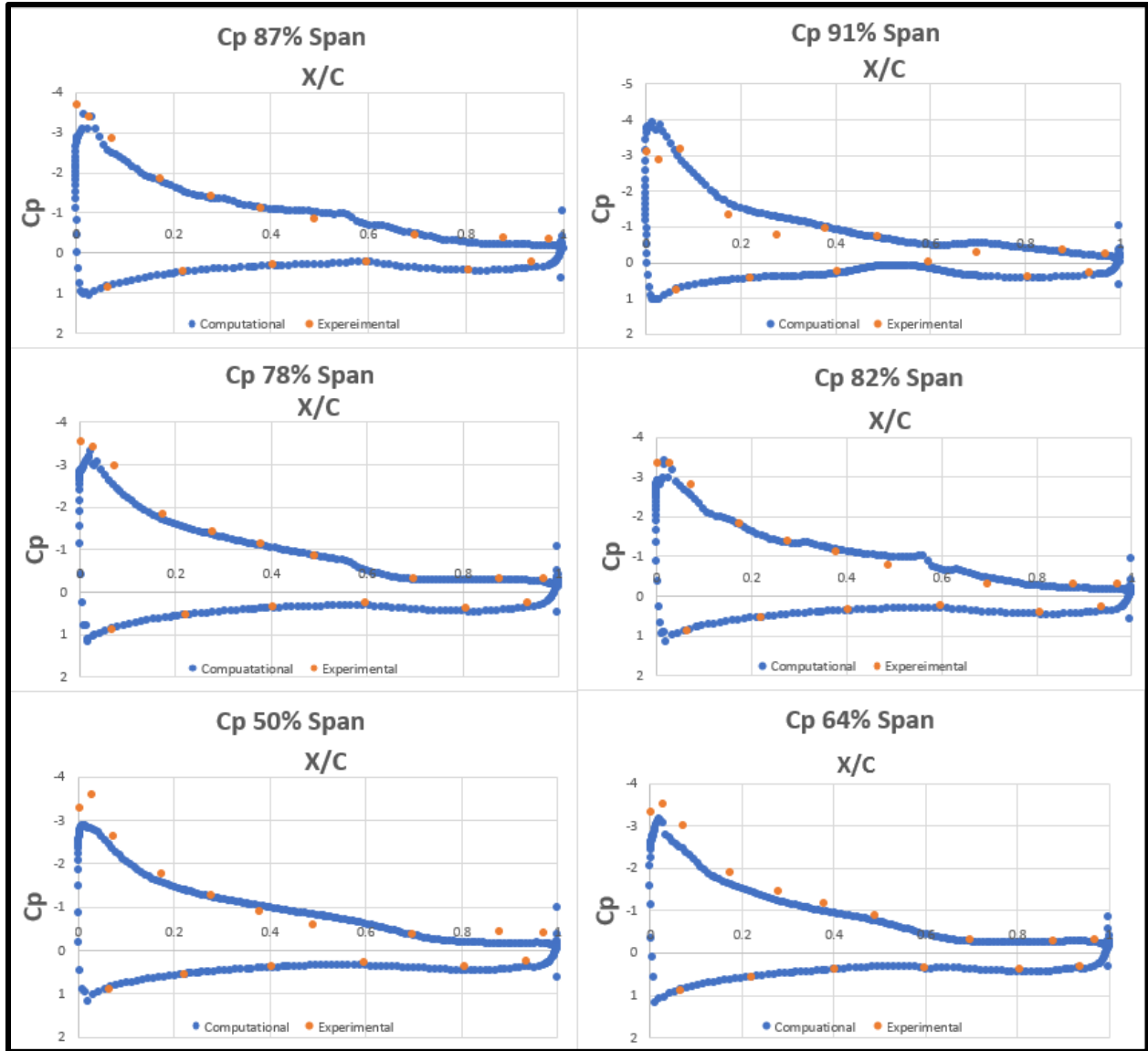
Fig 36. Helios CT=0 Angle of Attack =  $5^\circ$

As before an improvement in  $C_p$  prediction is observed with an increase in the angle of attack. This trend holds for Helios cases while the opposite is true for the Tenasi cases.



**Fig 37. Helios CT=0 Angle of Attack = 7°**

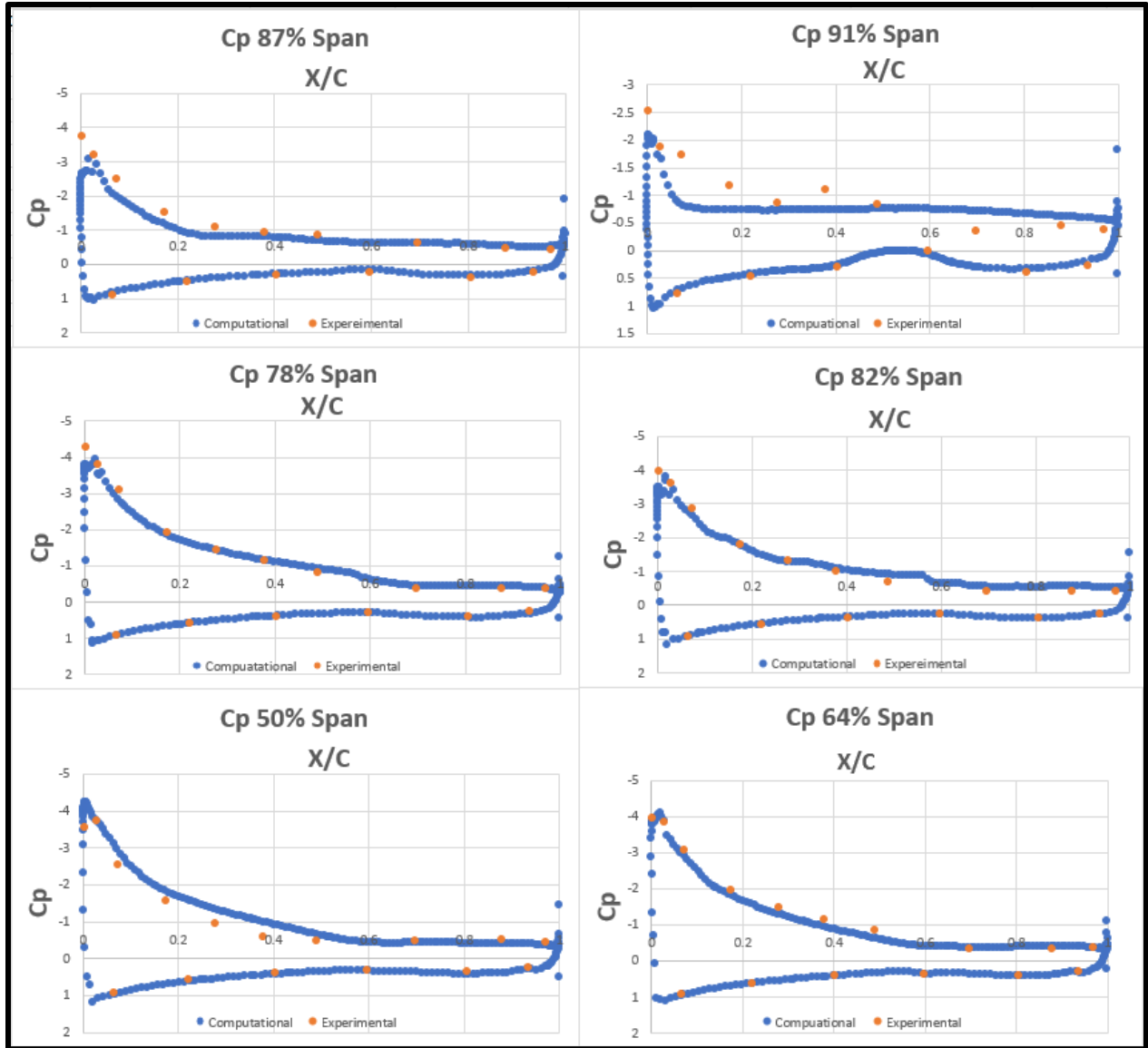
The same trend continues, however, there is a slight inaccuracy at the 91% span location. This is also shared with the Tenasi case.



**Fig 38. Helios CT=0 Angle of Attack =  $15^\circ$**

Once again improvement is observed as angle of attack rises and the under prediction of  $C_p$  validates the underprediction of lift.





**Fig 39. Helios CT=0 Angle of Attack = 17°**

This case is the most accurate in the lift and except for the 91% span location is the most accurate in terms of  $C_p$  as well.

Overall, the  $C_p$  is predicted best by Tenasi, however, as the angle of attack increases Helios performs slightly better. Additionally, because the propeller wake is predicted better the higher

span locations are predicted better using Helios. In both cases the No Propeller cases seem to perform better despite the inability to capture unsteady flow features using a steady simulation. Some direct comparisons are shown below. These represent the most basic and most extreme cases spanning the zero and 17-degree angle of attack cases as well as the 50 and 91 percent span locations. The general trend holds with the poor prediction by Helios at low angles of attack improving as the angle increases. The inverse is true of Tenasi. Also as span increases Helios performs better where Tenasi performs worse.

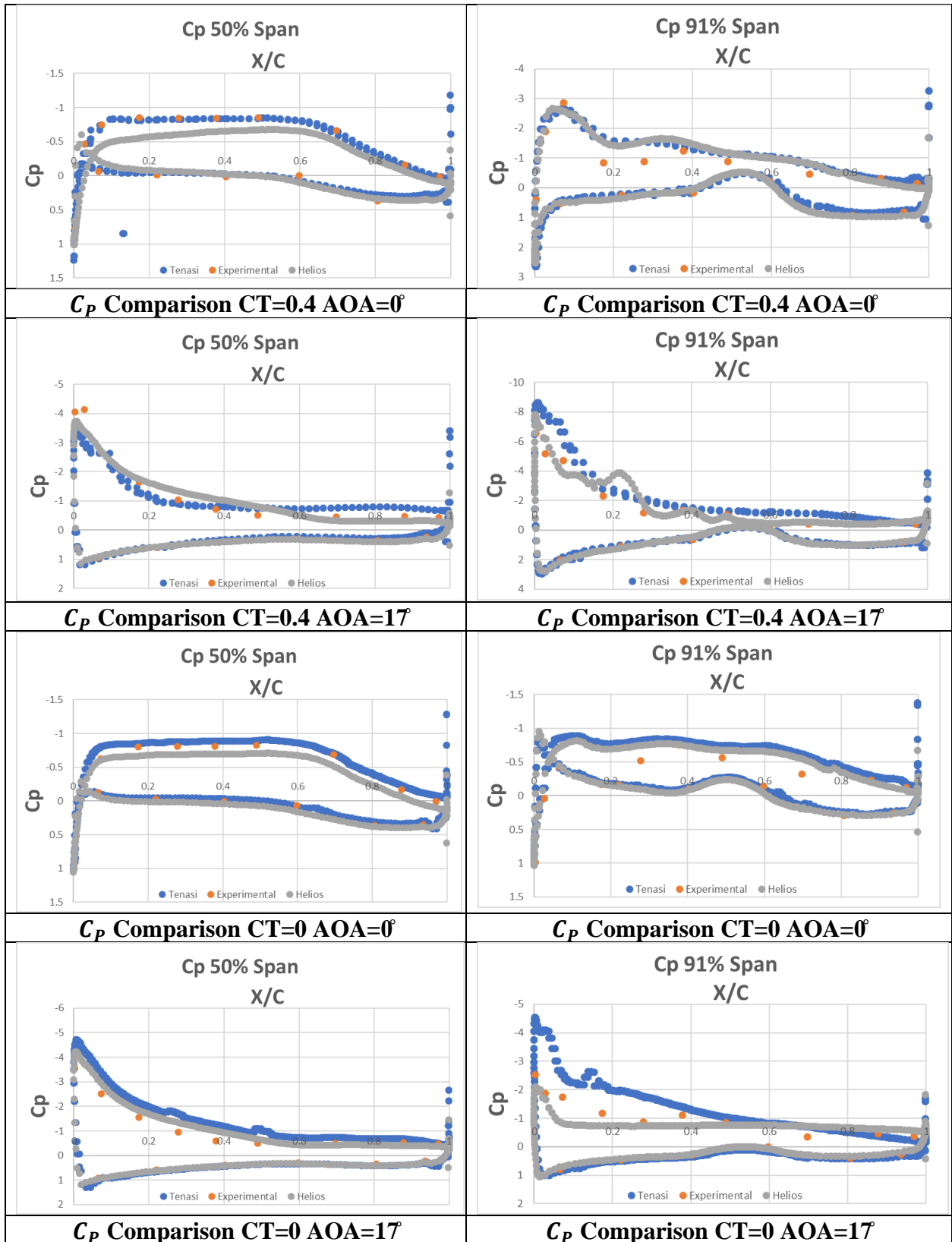


Fig 40. Helios and Tenasi  $C_p$  Comparisons

## **7 Conclusions**

Overall, the simulations of all cases were successful, however, the most accurate simulations came from the unsteady simulations. The cases at the mid-range of angles of attack performed the best. The refinement of the propellers and the addition of AMR improved the prediction of thrust. This is shown in the improvement in accuracy of thrust between the Tenasi and Helios cases. Helios cases all around could likely be improved through a better averaging of values. In the case of both solvers a low Mach number preconditioner could be used. Since the bulk of the flow is relatively low Mach number this could influence the accuracy of the simulations. Helios simulations increase in accuracy as the angle of attack goes up, however, the inverse is true of Tenasi. Tenasi simulation proved to be less expensive computationally, however, the ability to run larger cases faster on Helios using DOD resources could prove beneficial if the scope of this project were to expand beyond simply a propeller and wing. In the meshing and simulation process more difficulties were experienced with Tenasi since a sliding interface had to be created. Helios does not have this problem due to its hole cutting abilities. Tenasi does have the ability to be run locally however and requires no clearance to do so. Ultimately both solvers performed well, each performing well in certain areas. Likely with continued work, more detailed and specific methods can be developed to improve the accuracy of these simulations.

## **8 Recommendations**

There are many aspects of this topic of research that can be continued. If time had permitted the largest goal of this study is to accurately compare both solvers. Because two different meshes were used for both propeller cases, a true comparison would allow for the same mesh to be used in Tenasi. If further research is to be conducted, successfully performing simulations with the same

mesh is of the highest priority. Additionally, the inconsistencies in the non-propeller cases are of some concern. Most likely the best way to perform these is to begin with steady simulation and transition to unsteady. Transition models are of interest because they have shown abilities to improve the accuracy of the simulation. Further simulations performed with transition models may help select one that improves accuracy across all simulations. These simulations should be carried out unsteady as that is typically what they have been designed for. Lastly, further interaction effects could be studied. Particularly concerning the cyclic loading of both propeller and wing forces and the spanwise loading of the wing. These CFD – CSD tied simulations can help better inform the design of the structural components of the aircraft.

## **9 Acknowledgements**

I would like to acknowledge the gracious use of technological, computational, and human resources provided by the SimCenter at the University of Tennessee at Chattanooga, through a grant from the UC Foundation. Additionally, Dr. Kidambi Sreenivas and Dr. Charles Margraves have been invaluable to this process. Images were created by using FieldView as provided by Intelligent Light through its University Partners Program. I would also like to acknowledge computational resources and access to Helios have been provided by the ADD office and DOD HPC program. Without such support this thesis would not be possible. Additional thanks for the support of the ADD office, particularly Roger Strawn, Mark Potsdam, Buvana Jayaraman, and Jared Sagaga.

## References

- Borer, Nicholas et al. *Comparison of Aero-Propulsive Performance Predictions for Distributed Propulsion Configurations*. 2017.
- Coder, J. G. and Maughmer, M. D., "Computational Fluid Dynamics Compatible Transition Modeling Using an Amplification Factor Transport Equation," *AIAA Journal*, Vol. 52, No. 11, 2014, pp. 2506-2512
- FieldView, FieldView Inc. Software Package, Ver. 19, Rutherford, NJ
- FUN3D, NASA. Software Package, Ver. 13.6-717bd48
- Ginn, Starr and Rick Hooker. "Wind Tunnel Testing of Propeller Wingtip Interactions." *AIAA Aviation*, 2018.
- Lakshminarayan, Vinod K. et al. "Development and Validation of a Multi-Strand Solver for Complex Aerodynamic Flows." *Computers & Fluids*, vol. 147, 2017, pp. 41-62, doi:<https://doi.org/10.1016/j.compfluid.2017.02.002>.
- Lim, Joon W. "Fundamental Investigation of Proprotor and Wing Interactions in Tiltrotor Aircraft." 2018.
- MATLAB, MATLAB Inc. Software Package, Ver. R2019b, Asheboro, NC
- Medida, S., and Baeder, J. D., "Application of the Correlation-based  $\gamma\tilde{Re}_{\theta t}$  Transition Model to the Spalart-Allmaras Turbulence Model," 20th AIAA Computational Fluid Dynamics Conference, No. AIAA-2011-3979, Honolulu, HI, 2011.
- Menter, F. R., "Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications," *AIAA Journal*, Vol. 32, No. 8, August 1994, pp. 1598-1605.
- Menter, Florian R., et al. "A One-Equation Local Correlation-Based Transition Model." *Flow, Turbulence and Combustion*, vol. 95, no. 4, Dec. 2015, pp. 583–619. doi:10.1007/s10494-015-9622-4.
- Pointwise, Pointwise, Inc. Software Package, Ver. 18.2R2, Fort Worth, TX
- Spalart, P. R. and Allmaras, S. R., "A One-Equation Turbulence Model for Aerodynamic Flows," *Recherche Aerospaciale*, No. 1, 1994, pp. 5-21
- Sreenivas, Kidambi et al. "Numerical Simulation of the Interaction between Tandem Wind Turbines." *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 157, 2016, pp. 145-157, doi:<https://doi.org/10.1016/j.jweia.2016.09.003>.
- Wissink, Andrew M. et al. "Recent Advancements in the Helios Rotorcraft Simulation Code " 2016.