

Improved Computational and Experimental Validation Using Different Turbulence Models

Jay Marcos* and David D. Marshall†

California Polytechnic State University, San Luis Obispo CA 93407-0352

This paper will explore the methods and techniques necessary to perform a more accurate CFD validation of experimental results. The methods and techniques used will be validated against experimental wind tunnel data of a 2D high lift airfoil with a 3D engine performed by Georgia Tech Research Institute. Preliminary results showed that computational methods over predict the lift and drag coefficient, while still showing very similar trends in C_L and C_D . To further improve the preliminary results and the predictive capabilities, different turbulence models will be investigated. Results of this validation will assist in determining the appropriate turbulence models, boundary conditions, mesh characteristics and other CFD modeling techniques necessary to capture complicated flow physics associated with the coupling of circulation control wings and engine exhaust flows.

Nomenclature

A	= Area	S	= Wing planform area
C	= Chord length	T	= Static temperature
C_L	= Lift coefficient	v^2	= Velocity scale
C_D	= Drag coefficient	V	= Velocity
C_μ	= Slot blowing coefficient	x/C	= x-Location of engine
C_T	= Thrust coefficient	z/D	= z-Location of engine
D	= Engine nozzle diameter	Greek Symbols	
e_{app}	= Approximate relative error	α	= Angle of attack
e_∞	= Extrapolated relative error	ΔV	= Cell volume
f	= Elliptic relaxation factor	ε	= Turbulent dissipation rate
GCI	= Grid convergence index	ϕ	= Key variable
h	= Average mesh size	ω	= Specific dissipation rate
k	= Turbulent kinetic energy	Subscripts	
\dot{m}	= Mass flow rate	∞	= Freestream
n	= Order of solution	CFD	= Computational fluid dynamics
N	= Total number of cells	exp	= Experimental
P	= Static pressure	eng	= Engine conditions
q	= Dynamic pressure	$slot$	= Slot conditions
r	= Grid refinement factor	1	= Coarsest mesh
R	= Gas constant	2	= Intermediate mesh
		3	= Finest mesh

I. Introduction

THE intent of this paper is to show the development and advantages, as well as disadvantages, of using computational fluid dynamics as a validation tool for existing wind tunnel experimental data of a 2D high lift airfoil with a 3D engine performed by Englar et. al. at Georgia Tech Research Institute¹. Figure 1 shows the GTRI model equipped with circulation control wings (CCW) and upper surface blowing (USB) from the engine.

* Student, Aerospace Engineering Dept. Student Member.

† Associate Professor, Aerospace Engineering Dept. Senior Member.

The motivation behind this validation is to develop grid generating and solving techniques that will be applied to future CFD analysis of other aircraft configurations from the NASA Research Announcement (NRA) project. One of the primary purposes of the NASA contract is to develop the predictive capabilities for the design and performance of a 100-passenger N+2 generation cruise efficient, short take-off and landing subsonic aircraft configuration. The main tasks GTRI sought to achieve with the wind tunnel experiments were to gain thorough understanding of geometric and pneumatic interactions between CCW and USB or over the wing (OTW) powered lift as well as perform dual-radius CCW flap experiments with and without blowing.

To validate GTRI's experimental results, this paper focuses on CFD grid generation methods as well as determination of the boundary conditions and turbulence model for the CFD solver. Computed data for coefficient of lift and drag are compared to the reported C_L and C_D for GTRI's Configuration B model. A summary of Configuration B's different geometries is shown in Table 1 and Figure 2.

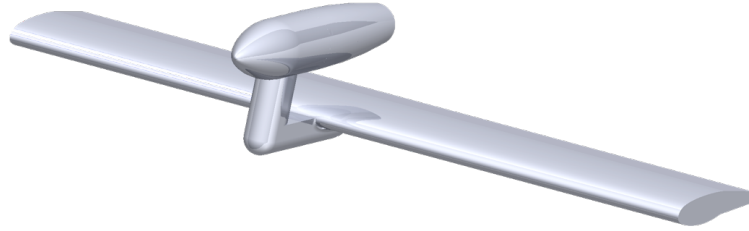


Figure 1. Configuration B equipped with high-lift and upper surface blowing.

Table 1. Summary of Configuration B's geometry.

Config	$\delta_{\text{flap}}, ^\circ$	$\delta_{\text{LE}}, ^\circ$	x/C, eng	x_{te}/D	z/D, eng	$\delta_{\text{hood}}, ^\circ$	$h_{\text{ccw}}, \text{in}$
B	0	0	0.03	5.25	0.37	Off	0.02

Configuration B: $\frac{x}{C}=0.03$, $\frac{x_{\text{te}}^*}{D}=5.25$, $\frac{z}{D}=0.37$

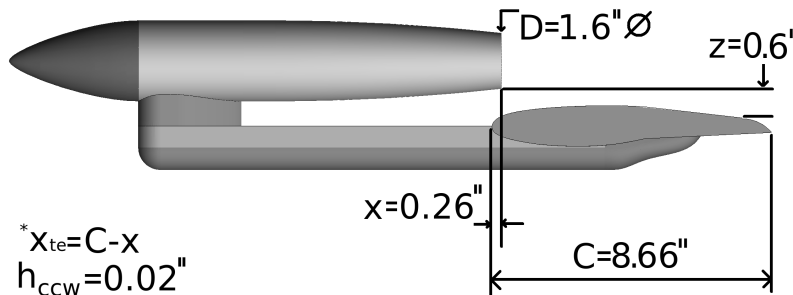


Figure 2. Configuration B with no trailing edge flap deflection and no leading edge flap and engine hood installed.

Ansys Icem 11.0² was utilized to construct a combination of structured and unstructured meshes to accurately discretize the flow domain. With the appropriate boundary conditions and solver settings, Ansys Fluent 6.3³ was then used to solve for the governing equations of momentum, mass, and energy within the generated mesh. This paper also focuses on the different steps in improving the initial data with the application of different turbulence models. The ability of $k-\epsilon$ ⁴, and $k-\omega$ ⁵, Fluent's built in turbulence models, as well as the ν^2-f turbulence models of Storm and Marshall⁶ to validate the experimental data will be presented.

II. Grid Generation

Before the governing equations for mass, momentum, and energy can be solved an appropriate mesh will be generated to discretize the flow domain of the model. The mesh will consist of a surface mesh on the model, a boundary layer mesh to accurately model the viscous forces, and a volume mesh that will cover all parts of the flow domain.

To more accurately match the experimental conditions, the simulation was performed with the presence of wind tunnel walls as opposed to free air to capture wall effects. This meant that the dimensions of the flow domain around the CFD model had to be the same as the dimension of the MTF, 30x43x90 in. Figure 3 shows the CFD model inside the flow domain where the mesh will be constructed. The additional lengths located at the front and behind the “wind tunnel” walls will act as a “*symmetry*” boundary condition to straighten the flow entering the “wind tunnel.”

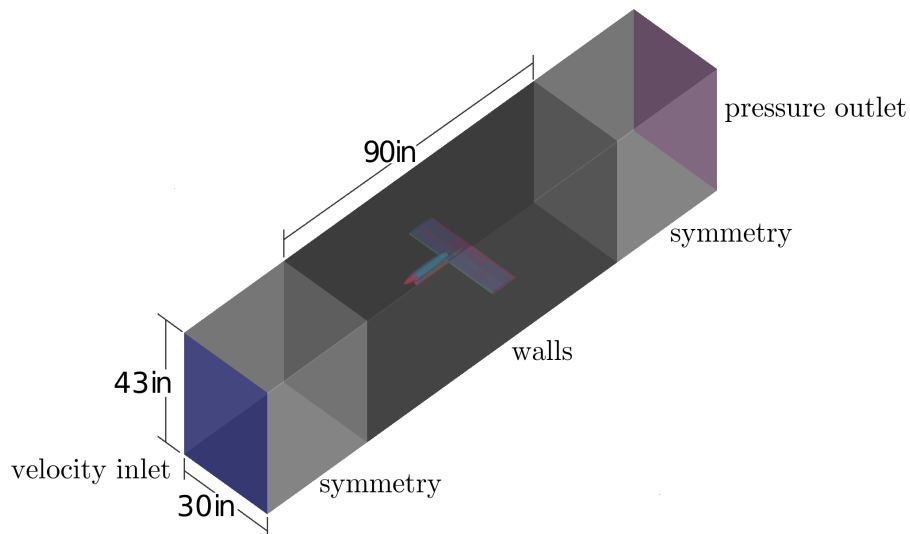


Figure 3. Flow domain representing the wind tunnel walls to capture wall effects.

A.CAD/Geometry

A computer aided drafting model of GTRI's Configuration B must first be generated before meshing can begin. Using Ansys Icem, the model is then broken up into different parts, allowing for a mesh with different cell sizes. This can limit the number of smaller cells to a particular region on the model that will be experiencing high flow property gradients, minimizing the overall cell count. Figure 4 shows a typical configuration broken up into different parts and ready for surface meshing.

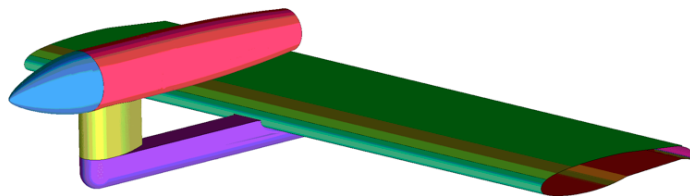


Figure 4. CAD model broken up into different parts

B. Unstructured Surface Mesh

A surface mesh must then be generated to accurately define the shape of the model and also capture the high flow property gradients caused by the blowing of the engine and the CCW slot. The transitioning of smaller cells to larger cells on the surface will ultimately determine the growth rate of the volume mesh. It's essential that surface mesh is carefully generated since accuracy of the solution is dependent on the sizes and growth rate of the cells. Figure 5 shows the smaller cells on the leading edge as it transitions into larger cells towards the upper surface of the wing. Breaking the model into different parts becomes helpful because it allows for cell refinement around a more complex geometry like the leading edge and cell coarsening on top of the wing, a less complex geometry feature. Cell refinement not only helps define the complex geometries of the model, but can also capture the high flow property gradients that are present in places like the trailing edge surface. The blowing of air out of the CCW slot will cause the top surface of the trailing edge to experience high flow property gradients, indicating that smaller cells are necessary to capture these changes. Figure 6 shows the smaller cells on the trailing edge surface to capture the high flow property gradients.

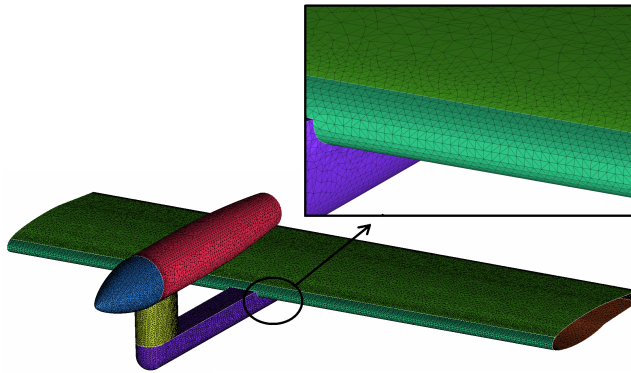


Figure 5. Leading edge surface mesh.

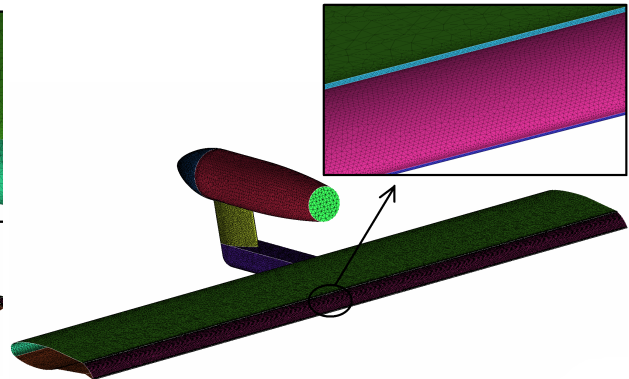


Figure 6. Trailing edge surface mesh.

C. Unstructured Interior Volume Mesh

An unstructured mesh will then be generated in the flow domain near the model which will then transition into a structured mesh for the flow domain far from the model. Constructing a good mesh near the model is important as it presents high flow property gradients from the engine and CCW slot that can cause a lot of unsteadiness in the numerical solution. With Ansys Icem 11.0, the process of generating an unstructured volume mesh is simplified by providing user friendly options that can generate the mesh in a small amount of time. Figure 7 shows the unstructured tetrahedral cells generated by the Octree method. The Octree algorithm in Ansys Icem assures that cell refinement is performed where necessary while constructing coarser cells wherever possible. This method meshes the outer domain first before generating smaller cells to capture the complex geometry of the model.

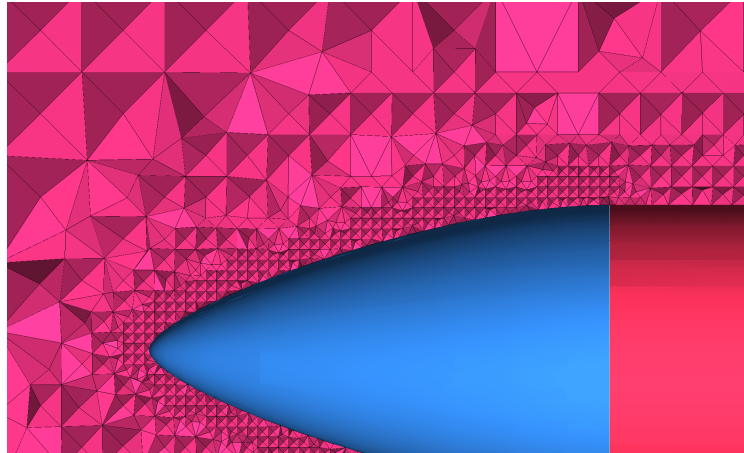


Figure 7. Unstructured interior volume mesh

D. Boundary Layer Mesh

A good quality boundary layer mesh must then be formed in order to capture the near wall flow features. The quality of the boundary layer mesh is highly dependent on several factors: the cell shape in terms of aspect ratio, skewness, and warp angle. The user has the ability to directly control the aspect ratio of the cells by specifying the initial height and growth ratio of the prisms. The length of the base of the prism will be predetermined by the size of the surface mesh, since the prisms are generated from the triangular cells. Therefore, when sizing the surface mesh, it is important to consider the prism layers that will be grown on it. The user also has some ability in improving the skewness of the prisms by using the built in smoothing algorithms of the grid-generation software. Figure 8 shows the boundary layer mesh grown on the surface of the model transitioning into the unstructured volume mesh. To ensure a good volume transition of the prism mesh, each prism layer normally has a growth ratio of 40% and the last prism element has a roughly 1:1 volume ratio to the adjacent tetrahedral element.

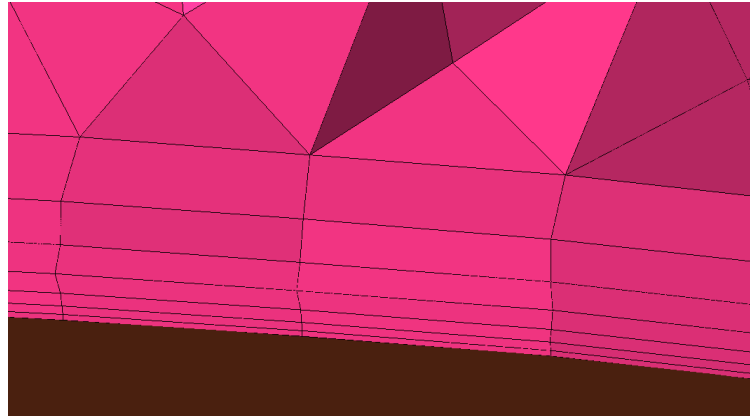


Figure 8. Boundary layer mesh

E. Structured Exterior Volume Mesh

The structured mesh is then constructed using the multi-blocking approach. The flow domain far away from the model is first broken up into smaller blocks. Breaking the blocks into smaller sections allows the user to map each block with different meshing criteria. The user can then refine any block sections to capture flow properties without having to refine other parts of the domain that will not be experiencing high flow property gradients. For example, the flow domain located behind the model will experience the highest flow property gradients because of the exhaust from the engine and the circulation control slot. The user can capture these changes by refining the mesh within the blocks located behind the model without having to refine the mesh within the blocks located in front of the model. Figure 9 shows the blocking scheme used to break the domain up into different blocks for hexahedral volume meshing.

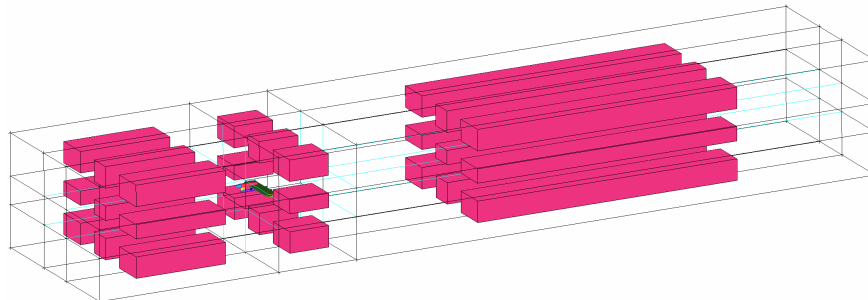


Figure 9. Multi-blocking used to map structured exterior mesh

III. Numerical Simulation

A. Solver Settings and Boundary Conditions

Once a good quality mesh has been generated throughout the entire domain, the mesh file is then loaded into the CFD solver, Fluent 6.3. Before the governing equations can be solved, the appropriate settings must first be enabled throughout Fluent's interface and the correct boundary conditions must be specified to accurately match the conditions in which the experimental evaluations were performed. Table 2 shows a list of solver settings and boundary conditions from the numerical simulation done in the original investigation.

Table 2. Fluent solver settings and boundary conditions.

Fluent 6.3 Solver Settings	
Turbulence Model	<i>Standard k-ε</i>
Solver	<i>Compressible</i>
Density Calculator	<i>Ideal-gas Law</i>
Viscosity Calculator	<i>Sutherland's Law</i>
Engine Outlet	<i>Mass Flow Inlet</i>
CCW Outlet	<i>Mass Flow Inlet</i>
Freestream Conditions	
Pressure, psi	14.7
Velocity, ft/s	68
Mach Number	0.06
Temperature, R	534.65
Engine	
Engine, psi	13.98
Mass Flow Rate, slug/s	<i>Varying</i>
Temperature, R	600.76
TE CCW	
Engine, psi	13.98
Mass Flow Rate, slug/s	<i>Varying</i>
Temperature, R	600.76

B. C_μ and C_T calculation

The given experimental data used to determine the boundary conditions included values for dynamic pressure, q , slot blowing coefficient, C_μ , and thrust coefficient, C_T . A “mass-flow-inlet” boundary condition was set for the slot and the engine in Fluent, which required a mass flow rate, \dot{m} , to be inputted in the settings. Values for \dot{m} were calculated from the reported experimental C_μ and C_T using

$$\dot{m}_{slot} = \sqrt{\frac{C_{\mu,exp} P_\infty A_{slot} q_\infty S}{RT_\infty}} \quad (1)$$

$$\dot{m}_{eng} = \sqrt{\frac{C_{T,exp} P_\infty A_{eng} q_\infty S}{RT_\infty}} \quad (2)$$

where A is the slot area, S is the wing planform area, and R is the gas constant. After specifying a \dot{m} and the rest of the boundary conditions, the simulation was performed. The numerical C_μ was then calculated using the velocity at the exit plane of the slot and engine with the following equation

$$C_{\mu,CFD} = \frac{\dot{m}_{slot} V_{slot}}{q_{\infty} S} \quad (3)$$

$$C_{T,CFD} = \frac{\dot{m}_{eng} V_{eng}}{q_{\infty} S} \quad (4)$$

In Fluent, the velocity at the slot and the engine, V_{slot} and V_{eng} respectively, can be directly obtained from the results. However, in wind tunnel experiments, measuring the actual velocities at the slot and engine can be quite difficult. Traditionally, these velocities are approximated using the known measurement of mass flow rate and freestream density instead of the actual density outside the slot or engine as seen from the following expressions

$$V_{slot,exp} = \frac{\dot{m}_{slot}}{\rho_{\infty} A_{slot}} \quad (5)$$

$$V_{eng,exp} = \frac{\dot{m}_{eng}}{\rho_{\infty} A_{eng}} \quad (6)$$

C. Grid Independence Study Using Richardson's Extrapolation

To perform a grid independence study, Richardson's extrapolation was used to determine the grid convergence index, GCI , and the extrapolated solution if an infinitely large mesh size was used. This method is applicable to a study using three different meshes, which are all proportional in size. Celik et. al.⁷ recommended the following procedure for estimating the discretization error. First the average mesh, h , was determined from the following equation

$$h = \left(\frac{1}{N} \sum_{i=1}^N \Delta V_i \right)^{1/2} \quad (7)$$

where ΔV is the volume of the i^{th} cell and N is the total number of cells. As a measure of refinement from one grid to the next, the grid refinement factor, r , was calculated using

$$r_{ij} = \frac{h_i}{h_j} \quad (8)$$

where i corresponds to the coarser grid and j corresponds to the finer grid. The apparent order, n , of the method was then calculated using the expression

$$n = \frac{1}{\ln r_{23}} \ln \left[\left(\frac{\phi_1 - \phi_2}{\phi_3 - \phi_2} \right) \left(\frac{r_{23}^n - 1}{r_{12}^n - 1} \right) \right] \quad (9)$$

where ϕ is a variable important to the object of the simulation of the study. For this validation, ϕ represented the resulting C_L or C_D from the different meshes. The subscripts 1, 2, and 3 correspond to the mesh used, 1 being the coarsest and 3 being the finest. Richardson's extrapolation was then used to extrapolate the solution for an infinitely large mesh. The extrapolated solution was found from the expression

$$\phi_\infty = \frac{r_{23}^n \phi_3 - \phi_2}{r_{23}^n - 1} \quad (10)$$

The approximate and extrapolated relative errors were then found using

$$e_{app} = \left| \frac{\phi_3 - \phi_2}{\phi_3} \right| \quad (11)$$

$$e_\infty = \left| \frac{\phi_3 - \phi_2}{r_{23}^n \phi_3 - \phi_2} \right| \quad (12)$$

Finally, the grid convergence index was calculated using

$$GCI = \frac{1.25e_{app}}{r_{23}^n - 1} \quad (13)$$

IV. Turbulence Models

In addition to investigating Fluent's built in turbulence models, additional turbulence models as discussed in Ref. 8 will also be studied. The following are the different turbulence models that can improve the solver's ability to predict the effects of circulation controls flows.

A. k - ε Turbulence Model

Three k - ε turbulence models are available in Fluent's viscous settings. The standard, RNG, and realizable k - ε models all have similar forms but the major differences between them are

- the method of calculating turbulent viscosity
- the turbulent Prandtl numbers
- the generations and destruction terms in the ε equation.

For the standard k - ε model, the turbulence kinetic energy, k , and its dissipation rate, ε , are derived from the following transport equations.

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial t}(\rho k u_i) = \frac{\partial}{\partial x_j} \left[\left(\mu + \frac{\mu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right] + G_k + G_b - \rho \varepsilon - Y_M + S_k \quad (14)$$

$$\frac{\partial}{\partial t}(\rho \varepsilon) + \frac{\partial}{\partial x_i}(\rho \varepsilon u_i) = \frac{\partial}{\partial x_j} \left[\left(\mu + \frac{\mu_t}{\sigma_\varepsilon} \right) \frac{\partial \varepsilon}{\partial x_j} \right] + C_{1\varepsilon} \frac{\varepsilon}{k} (G_k + C_{3\varepsilon} G_b) - C_{2\varepsilon} \rho \frac{\varepsilon^2}{k} + S_\varepsilon \quad (15)$$

where G_k is the turbulence kinetic energy from the mean velocity gradients, G_b is the turbulence kinetic energy due to buoyancy, and Y_M is the contribution of the fluctuating dilatation in compressible turbulence to the overall dissipation rate. σ_k and σ_ε are the turbulent Prandtl numbers. The turbulent viscosity, μ_t , is defined as

$$\mu_t = \rho C_\mu \frac{k^2}{\varepsilon} \quad (16)$$

The default values for the constants $C_{1\varepsilon}$, $C_{2\varepsilon}$, C_μ , σ_k , and σ_ε are given as the following

$$C_{1\varepsilon} = 1.44, C_{2\varepsilon} = 1.92, C_\mu = 0.09, \sigma_k = 1.0, \sigma_\varepsilon = 1.3$$

Using the renormalization group (RNG) methods, the RNG k - ε turbulence model is derived from the Navier-Stokes Equations. The derivation yields constants with different values from those in the standard k - ε model. With a similar form to the standard model, the RNG transport equations are

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_i}(\rho k u_i) = \frac{\partial}{\partial x_j} \left[\alpha_k \mu_{eff} \frac{\partial k}{\partial x_j} \right] + G_k + G_b - \rho \varepsilon - Y_M + S_k \quad (17)$$

$$\frac{\partial}{\partial t}(\rho \varepsilon) + \frac{\partial}{\partial x_i}(\rho \varepsilon u_i) = \frac{\partial}{\partial x_j} \left[\alpha_\varepsilon \mu_{eff} \frac{\partial \varepsilon}{\partial x_j} \right] + C_{1\varepsilon} \frac{\varepsilon}{k} (G_k + C_{3\varepsilon} G_b) - C_{2\varepsilon} \rho \frac{\varepsilon^2}{k} - R_\varepsilon + S_\varepsilon \quad (18)$$

The main difference between the standard and the RNG k - ε models is the additional R_ε term in the ε equation defined as

$$R_\varepsilon = \frac{C_\mu \rho \eta^3 (1 - \eta/\eta_0) \varepsilon^2}{1 + \beta \eta^3} \frac{\varepsilon^2}{k} \quad (19)$$

Finally, the realizable k - ε turbulence model satisfies some mathematical constraints on the normal stresses that are consistent with the physics of turbulent flows. The transport equations for the realizable k - ε are

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_i}(\rho k u_i) = \frac{\partial}{\partial x_j} \left[\left(\mu + \frac{\mu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right] + G_k + G_b - \rho \varepsilon - Y_M + S_k \quad (20)$$

$$\frac{\partial}{\partial t}(\rho \varepsilon) + \frac{\partial}{\partial x_i}(\rho \varepsilon u_i) = \frac{\partial}{\partial x_j} \left[\left(\mu + \frac{\mu_t}{\sigma_\varepsilon} \right) \frac{\partial \varepsilon}{\partial x_j} \right] + \rho C_1 S_\varepsilon - \rho C_2 \frac{\varepsilon^2}{k + \sqrt{\nu \varepsilon}} + C_{1\varepsilon} \frac{\varepsilon}{k} C_{3\varepsilon} G_b + S_\varepsilon \quad (21)$$

The main difference between the realizable k - ε and the first two turbulence models is that C_μ is no longer constant. C_μ is defined as

$$C_\mu = \frac{1}{A_0 A_s \frac{k U^x}{\varepsilon}} \quad (22)$$

B. k - ω Turbulence Model

Two k - ω turbulence models are available in Fluent's viscous settings. The standard and SST k - ω models both have similar forms but the major differences between them are

- the modified turbulent viscosity formulation accounting for the transport effects of the principal turbulent shear stress and
- the gradual change from the standard k - ω model in the inner region of the boundary layer to a high Reynolds number version of the k - ε in the outer part of the boundary layer.

The turbulent kinetic energy, k , and the specific dissipation rate, ω , are derived from the following transport equations

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_i}(\rho k u_i) = \frac{\partial}{\partial x_j} \left(\Gamma_k \frac{\partial k}{\partial x_j} \right) + G_k + Y_k + S_k \quad (23)$$

$$\frac{\partial}{\partial t}(\rho \omega) + \frac{\partial}{\partial x_i}(\rho \omega u_i) = \frac{\partial}{\partial x_j} \left(\Gamma_\omega \frac{\partial \omega}{\partial x_j} \right) + G_\omega + Y_\omega + S_\omega \quad (24)$$

where the effective diffusivities, Γ_k and Γ_ω , are defined as

$$\Gamma_k = \mu + \frac{\mu_t}{\sigma_k} \quad (25)$$

$$\Gamma_\omega = \mu + \frac{\mu_t}{\sigma_\omega} \quad (26)$$

The turbulent viscosity, μ_t , is then defined as

$$\mu_t = \alpha^x \frac{\rho k}{\omega} \quad (27)$$

$$\alpha^x = \alpha_\infty^x \left(\frac{\alpha_0^x + \text{Re}_t / \text{Re}_k}{1 + \text{Re}_t / \text{Re}_k} \right) \quad (28)$$

C. Standard v^2 - f Turbulence Model

The standard v^2 - f model⁸ is very similar to the standard k - ε model, but it uses near-wall turbulence and pressure strain effects. Generally, it is a low Reynolds number turbulence model that is also valid for solid walls, eliminating the need to use wall functions. It is known for accurately simulating flows dominated by separation. The v^2 - f turbulence model is generally used in flows in which near-wall turbulence is significant, specifically flows with separation and recirculation. Instead of two transport equations like those of k - ε and k - ω , the v^2 - f solves four different transport equations for turbulence kinetic energy, turbulence dissipation rate, velocity scale, and elliptic relaxation factor. The transport equations for the velocity scale and the elliptic relaxation factor are defined as

$$\frac{\partial \overline{v^2}}{\partial t} = kf - u_j \frac{\partial \overline{v^2}}{\partial x_j} - \frac{\partial \overline{v^2}}{k} \varepsilon + \frac{\partial}{\partial x_j} \left[\left(\mu + \frac{\mu_t}{\sigma_{v^2}} \right) \frac{\partial \overline{v^2}}{\partial x_j} \right] \quad (29)$$

$$f = \frac{C_1 - 1}{T} \left(\frac{2}{3} - \frac{\overline{v^2}}{k} \right) + C_2 \frac{P_k}{\varepsilon} + L^2 \frac{\partial^2 f}{\partial x_j^2} \quad (30)$$

where P_k is the production of turbulent kinetic energy due to the mean flow gradients and is defined as

$$P_k = 2\mu_t S^2 \quad (31)$$

D. Modified ν^2 - f Turbulence Model

The modified ν^2 - f models of Storm and Marshall introduced a nonlinear eddy viscosity model⁹ and a streamline curvature correction to the standard ν^2 - f model¹⁰. The nonlinear eddy viscosity version of the ν^2 - f accounts for the turbulence anisotropy, something that the standard ν^2 - f fails to achieve, by using nonlinear relationships proposed by Petterson Reif⁹, which can improve the ν^2 - f model's predictive capability for turbulent shear flows.

The standard ν^2 - f is also unable to model its insensitivity to streamline curvatures. To sensitize the ν^2 - f 's ability to streamline curvature either the turbulent viscosity coefficient can be sensitized to invariants of strain and vorticity or the production and dissipation terms can be modified.

Validation of these two modified ν^2 - f turbulence models is presented in detail in Ref. 8. Preliminary results show further improvement of the standard ν^2 - f turbulence model in predicting circulation control flows by modifying it with a nonlinear eddy viscosity formulation and streamline curvature correction.

V. Results

The purpose of this section is to validate the CFD simulations through comparison with the experimental data. The results for the k - ε turbulence model will be presented, starting with the effects of variations in C_μ on the lift coefficient, followed by the effects of variations in C_T on the lift and drag coefficients, then the lift curves for two different C_μ with the same C_T . A grid independence study using Richardson's extrapolation will also be presented. Then the validation results for the k - ω turbulence model will then be compared to the results for the k - ε turbulence model.

A. Realizable k- ε Turbulence Model

Figure 10 shows the lift variation as a function of thrust and slot blowing for Configuration B. At higher values of C_μ , Fluent over-predicts C_L even more compared to lower values of C_μ . For a $C_T=0$, the relative error increases from a low of 12%, which occurs at the lowest C_μ , to a max of 20%, which occurs at the highest C_μ . Similarly, for a $C_T=2.1$ the relative error increases from a low of 19% to a max of 23%. Not only does Fluent's over-prediction of C_L increases as C_μ increases, but also as C_T increases.

Although the error in C_L increases as C_μ and C_T increases, the overall trend of increasing C_L with increasing C_μ is still captured. Validation of circulation control airfoils with no engine, $C_T=0$, reported by Jones et. al.¹¹ shows similar results where C_L was over-predicted but the trend of increasing C_L with increasing C_μ was still captured. Turbulence models and CFD grid issues were reported as possible reasons why the CFD code failed to match the experimental results.

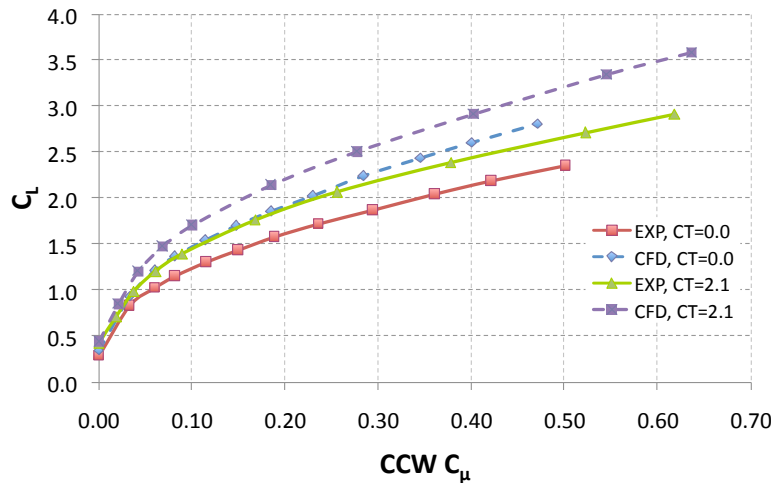


Figure 10. Configuration B. CCW C_μ sweep at $\alpha=5.5^\circ$, $\alpha=0^\circ$

Using Eq. (3), C_μ was calculated from the CFD results and is shown in Table 3. It's interesting to see from Table 3 that at a $C_T=0$ the relative error in C_μ increases as C_μ increases. However, for a $C_T=2.1$, the relative error decreases as C_μ increases. This is linked to the difference between the experimental C_μ and the CFD C_μ . The experimental C_μ is approximated using the freestream density as the "averaged" density at the slot instead of the actual density as shown in Eq. (5). For the case where $C_T=2.1$, the "averaged" density at the slot is getting closer to the freestream density, therefore the CFD C_μ is getting closer to the experimental C_μ .

Table 3. Experimental and numerical slot blowing coefficient.

$C_T = 0.0$				$C_T = 2.1$			
V_{slot}	$C_{\mu\text{CFD}}$	$C_{\mu\text{EXP}}$	% error	V_{slot}	$C_{\mu\text{CFD}}$	$C_{\mu\text{EXP}}$	% error
0.00	0.0000	0.0000	0.00	0	0.0000	0.0000	0.00
55.96	0.0330	0.0329	0.29	47.09	0.0208	0.0185	12.52
75.25	0.0599	0.0600	-0.12	66.70	0.0419	0.0374	12.17
87.47	0.0812	0.0816	-0.45	84.89	0.0681	0.0610	11.72
103.40	0.1140	0.1151	-0.96	102.49	0.0998	0.0898	11.17
117.18	0.1472	0.1494	-1.47	138.53	0.1847	0.1682	9.77
130.89	0.1847	0.1885	-2.01	168.55	0.2771	0.2560	8.28
145.45	0.2295	0.2357	-2.64	201.23	0.4021	0.3779	6.38
161.05	0.2838	0.2940	-3.46	231.88	0.5448	0.5225	4.26
176.78	0.3449	0.3605	-4.30	248.85	0.6357	0.6178	2.90
189.65	0.4000	0.4211	-5.02				
204.65	0.4708	0.5009	-6.02				

Figure 11 shows the lift variation as a function of thrust and blowing. Experimental results show little effect on lift from variation in C_T . Fluent was able to capture the trend in C_L but the relative error in C_L increases with a higher C_μ . For a $C_\mu=0$, the relative error in C_L was about 14% while for a $C_\mu=0.5$ the relative error increases to 25%.

Figure 12 shows the drag variations as functions of thrust and blowing. Fluent captures the thrust recovery very well with very little relative error. The maximum error of about 7% occurs at the highest C_T value. The coefficient of drag presented in this figure is the viscous and pressure stress contributions on the surfaces as well as the thrust components of the engine and slot.

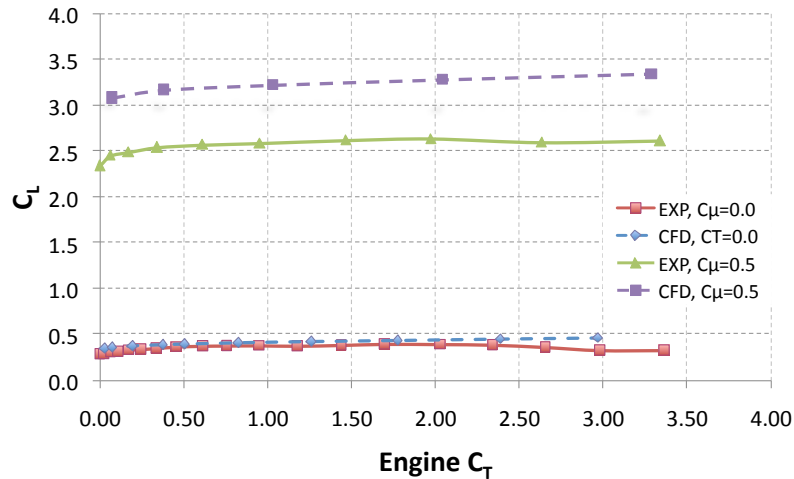


Figure 11. Configuration B lift as a function of C_T at $q=5.5\text{psf}$, $\alpha=0^\circ$.

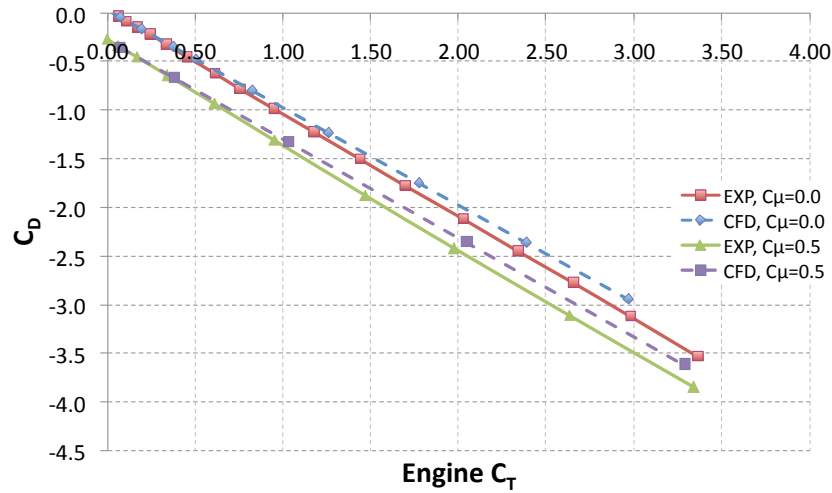


Figure 12. Configuration B lift as a function of C_T at $q=5.5\text{psf}$, $\alpha=0^\circ$.

Table 4 shows the calculated and the experimental thrust coefficients using Eq. (4). From this table, it's apparent that Fluent matches C_T more accurately at higher values of C_T . Even at low values of C_T the highest error is about 12%.

Table 4. Experimental and numerical thrust coefficient.

$C_\mu = 0.0$				$C_\mu = 0.5$			
V_{eng}	$C_{T,CFD}$	$C_{T,EXP}$	% error	V_{eng}	$C_{T,CFD}$	$C_{T,EXP}$	% error
28.98	0.0272	0.0241	0.00	46.59	0.0703	0.0622	0.00
47.21	0.0724	0.0642	12.78	107.58	0.3803	0.3410	11.52
76.94	0.1934	0.1725	12.13	174.66	1.0320	0.9529	8.31
106.66	0.3748	0.3370	11.22	240.60	2.0461	1.9739	3.66
123.27	0.5035	0.4554	10.57	297.15	3.2862	3.3380	-1.55
156.59	0.8243	0.7563	8.99				
191.67	1.2590	1.1775	6.92				
225.03	1.7745	1.6971	4.56				
257.62	2.3859	2.3410	1.92				
283.87	2.9667	2.9809	-0.48				

A grid independence study was also performed to determine a grid size acceptable for this validation. The results of the three different meshes are shown on Table 5. Figures 13 and 14 show the convergence of the different meshes towards the extrapolated solution as given by the Richardson Extrapolation for the lift and drag coefficient. The errors associated with these solutions are also plotted on the grid points. Although the finer mesh yields a C_L much closer to the extrapolated solution, the intermediate mesh was used mostly because of faster residual convergence.

Table 5. Calculation of GCI and discretization error.

	Lift Coefficient	Drag Coefficient
N_1	3.97×10^6	3.97×10^6
N_2	7.11×10^6	7.11×10^6
N_3	8.64×10^6	8.64×10^6
r_{12}	1.21	1.21
r_{23}	1.07	1.07
ϕ_1	1.732	0.0499
ϕ_2	1.708	0.0454
ϕ_3	1.691	0.0444
n	5.59	1.5
ϕ	1.6549	0.0349
e_{app}	0.0095	0.0221
e_{∞}	0.0221	0.275
GCI	0.027	0.269

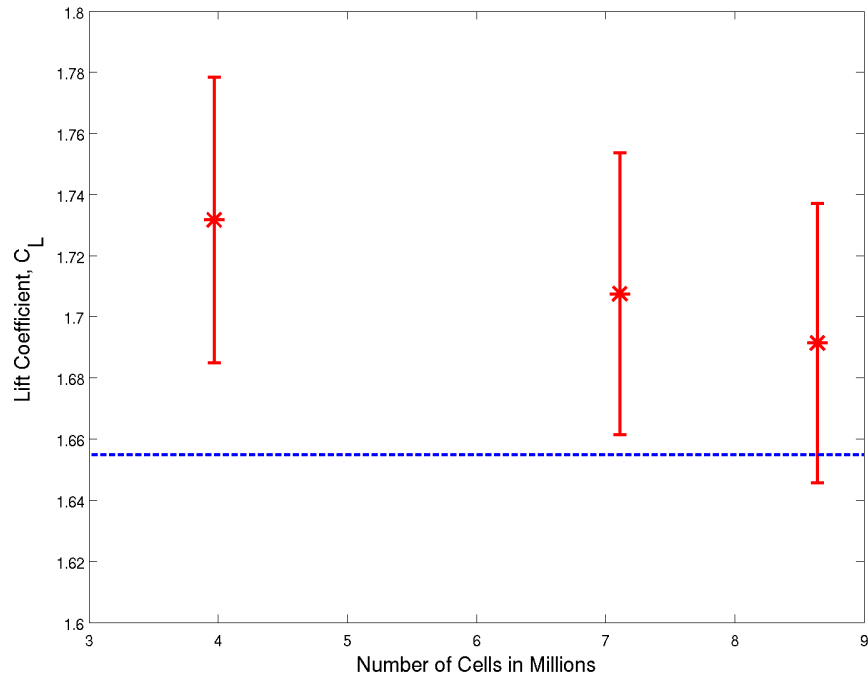


Figure 13. Convergence of lift coefficient towards the extrapolated solution as mesh size increase.

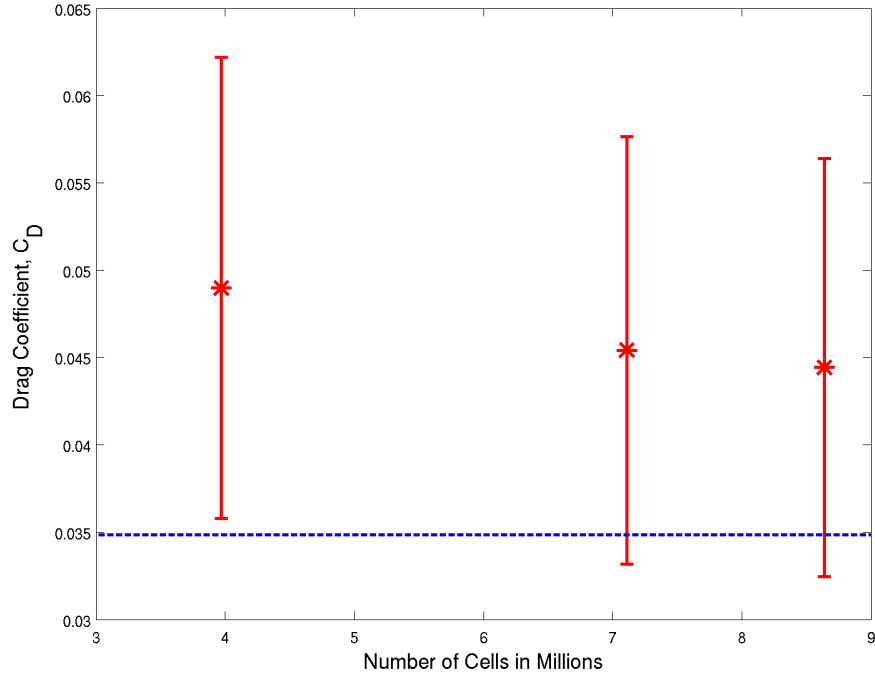


Figure 14. Convergence of drag coefficient towards the extrapolated solution as mesh size increase.

B. SST $k-\omega$ Turbulence Model

Similar analyses were performed using Fluent's $k-\omega$ turbulence model. Figure 15 shows the lift variation as a function of slot blowing coefficient for Configuration B with a thrust coefficient of zero. Using the $k-\omega$ turbulence model, Fluent does a poorer job in approximating the C_L compared to the $k-\epsilon$ turbulence model. Although the trend of increasing C_L with increasing C_μ is still captured, $k-\omega$ over predicts C_L even more at higher values of C_μ . Figure 16 shows a similar analysis but for a thrust coefficient of 2.1. For a $C_T=0$, the relative error increases from 17%, which occurs at the lowest C_μ , to a max of 23%, which occurs at the highest C_μ . The over prediction of C_L is even worse when the over the wing engine is turned on. For a $C_T=2.1$ the relative error increases from 28% to a max of 33%.

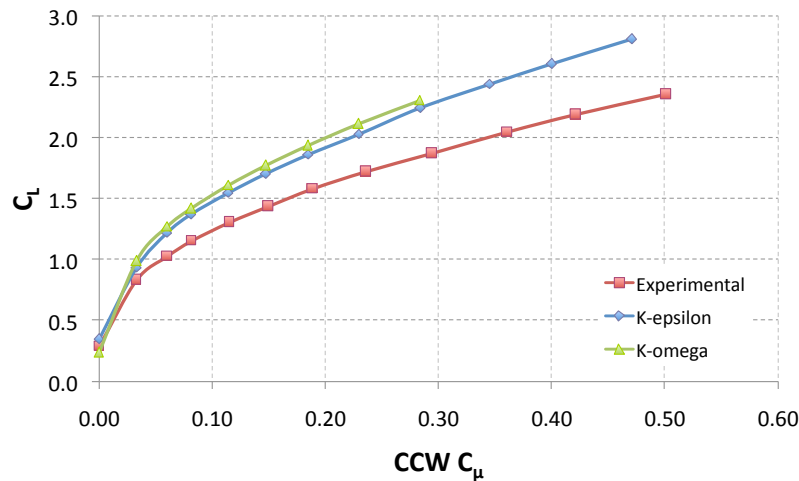


Figure 15. Lift distribution as a function of slot blowing coefficient at $q=5.5\text{psf}$, $\alpha=0^\circ$, and $C_T=0.0$ for Config. B.

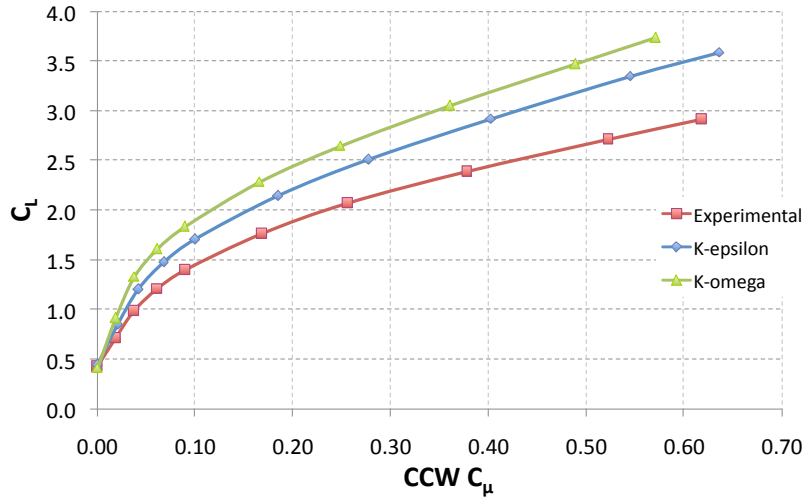


Figure 16. Lift distribution as a function of slot blowing coefficient at $q=5.5\text{psf}$, $\alpha=0^\circ$, and $C_T=2.1$ for Config. B.

Figure 17 shows the lift variation as a function of engine thrust coefficient for Configuration B with a slot blowing coefficients of 0.0 and 0.5. Again, regardless of what turbulence model is used, Fluent's over prediction of C_L is a lot higher when the circulation control is turned on. Increasing C_T has very minimal effect on the lift distribution except for the $k-\omega$ case with $C_\mu=0.5$. As C_T increases, the numerical C_L seems to be diverging quicker from the experimental C_L .

Regardless of what turbulence model is being used, Fluent over predicts C_L for every case and $k-\omega$ seems to perform the worst compared to $k-\epsilon$. However, when analyzing C_D the performance of either turbulence model is almost identical. Figures 18 and 19 show the drag distribution as a function of engine thrust coefficient for Configuration B with slot blowing coefficients of 0.0 and 0.5, respectively. Fluent still over predicts C_D by a maximum of 7% compared to the experimental when either turbulence model is used.

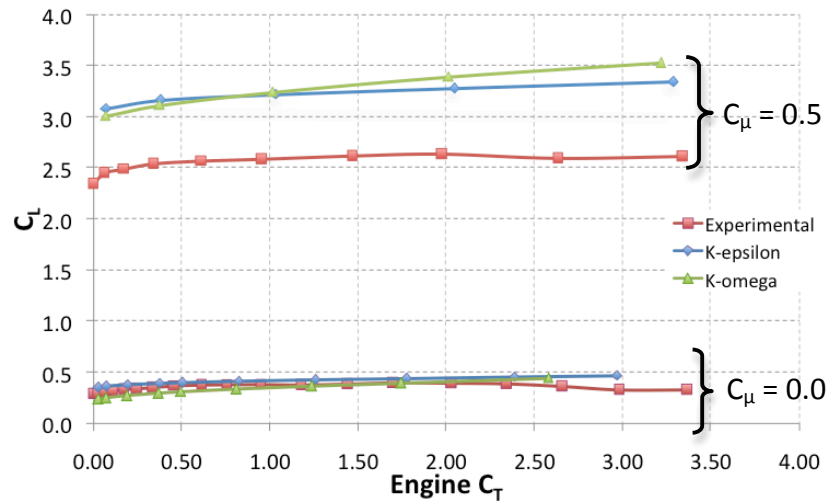


Figure 17. Lift distribution as a function of engine thrust coefficient at $q=5.5\text{psf}$ and $\alpha=0^\circ$ for Config. B.

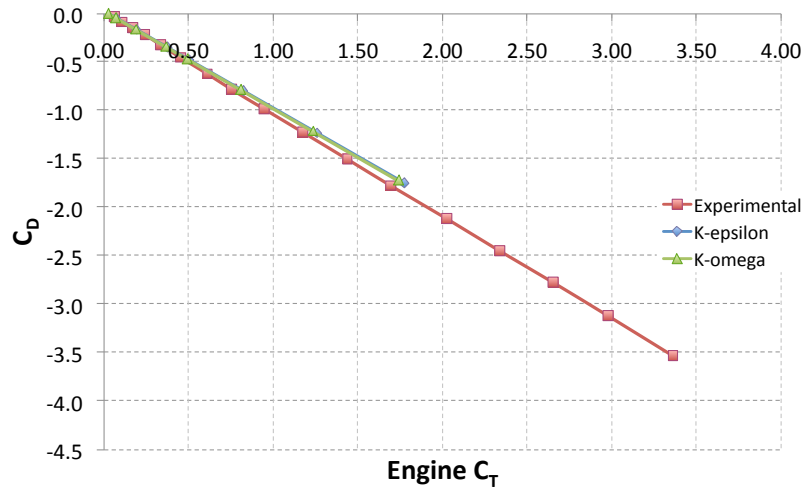


Figure 18. Drag distribution as a function of engine thrust coefficient at $q=5.5\text{psf}$, $\alpha=0^\circ$, and $C_\mu=0.0$ for Config. B.

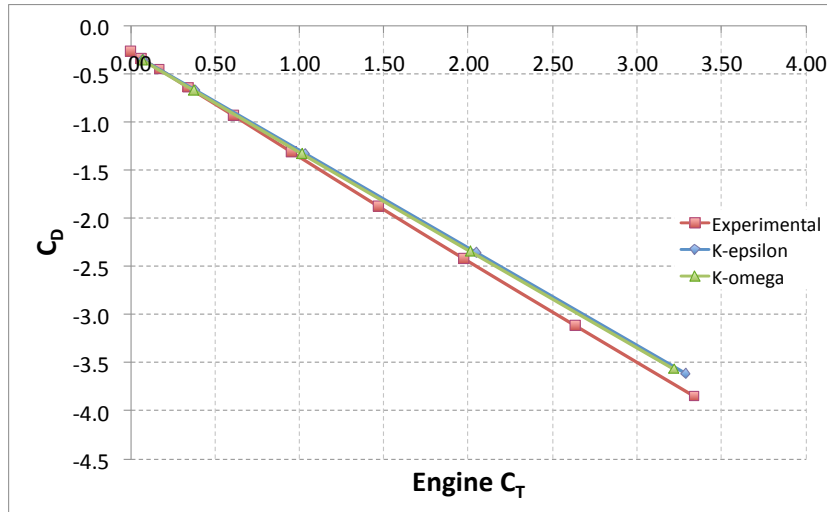


Figure 19. Drag distribution as a function of engine thrust coefficient at $q=5.5\text{psf}$, $\alpha=0^\circ$, and $C_\mu=0.5$ for Config. B.

C. Residual Convergence

An example of the convergence history is shown in Figure 20. The large spikes around 1000 iterations result from the solution being changed from first order to second order upwind. The solution was then iterated enough times until there was insignificant change in the residuals, or until they became fairly horizontal.

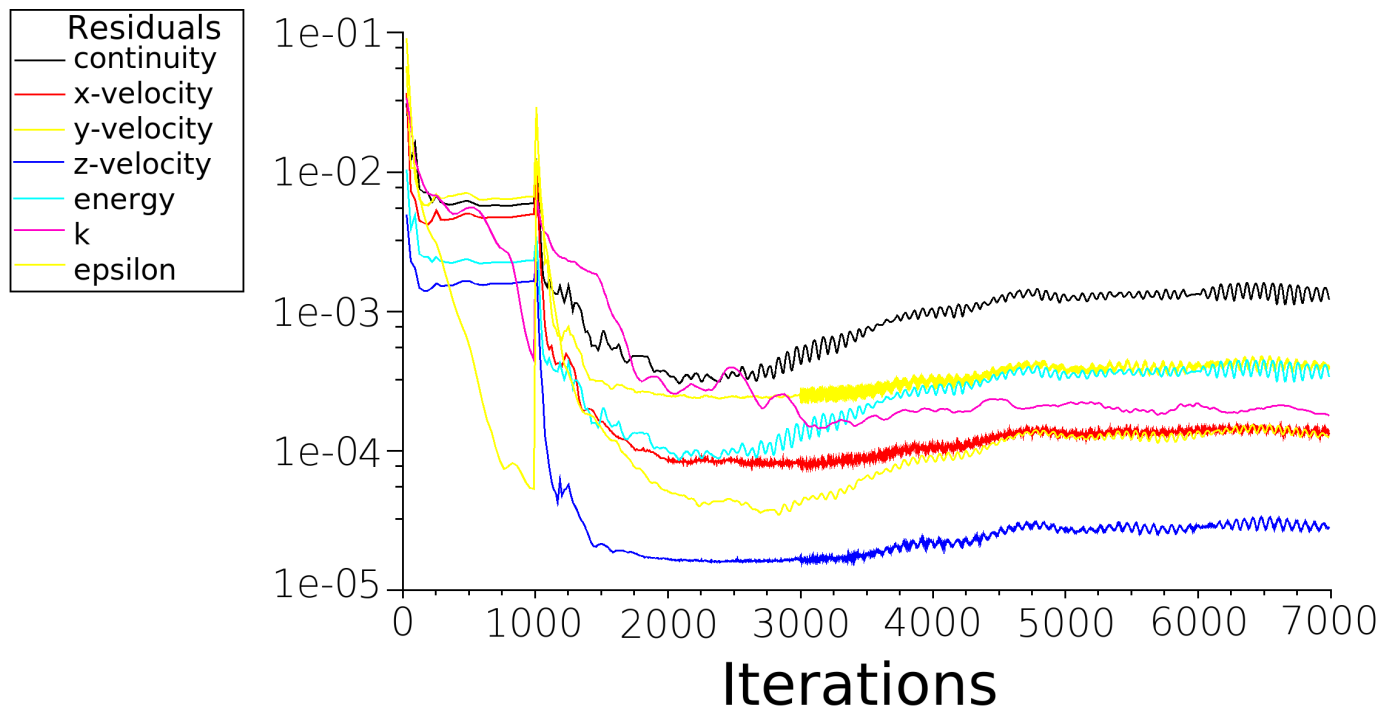


Figure 20. Typical convergence history.

D. Flow Visualization

Figure 21 shows earlier evaluations of similar powered-lift models in the same MTF tunnel presented by Englar et al.¹, which was used as the basis of GTRI's aerodynamic testing. The flow visualization shows the deflection capability of the engine thrust caused by the entrainment of the engine exhaust flow by the CCW flap. Similar results were obtained from CFD analysis of Configuration B. Figure 22 shows similar deflection of the engine thrust caused by the entrainment of the engine exhaust. The engine static thrust deflection observed in Fig. 22 does not exactly match the thrust deflection shown in Fig. 21 because of multiple reasons. It is currently unknown where the engine of the original model was positioned compared to the surface of the wing and the CCW slot. For this reason, x/C and z/D may be different between the two models. Also, Configuration B has no CCW flap deflection. This causes a smaller engine exhaust deflection angle for Configuration B. The main purpose of presenting these two figures is to show that engine thrust deflection is still present in the CFD analysis.

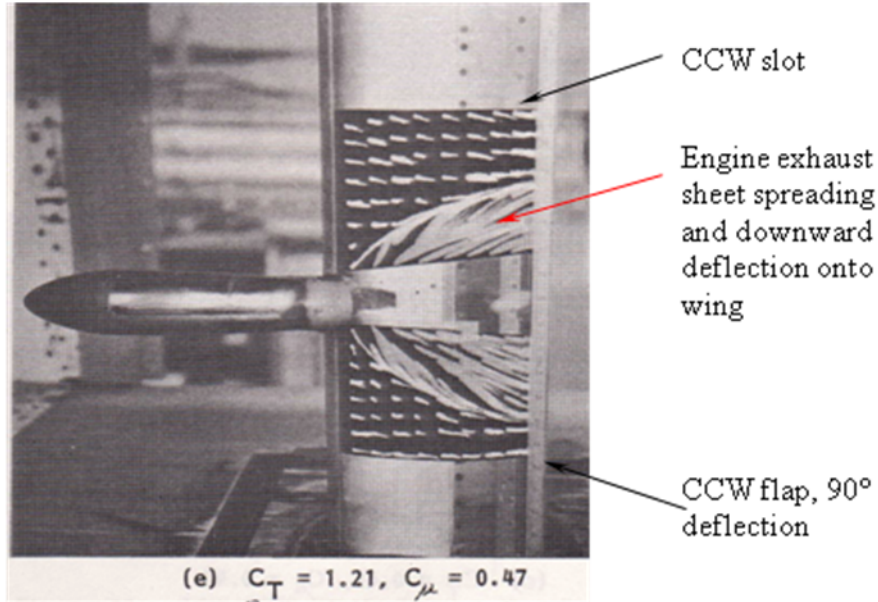


Figure 21. CCW/OTW model with jet flow entrainment shown by flow visualization from Englar et al.¹

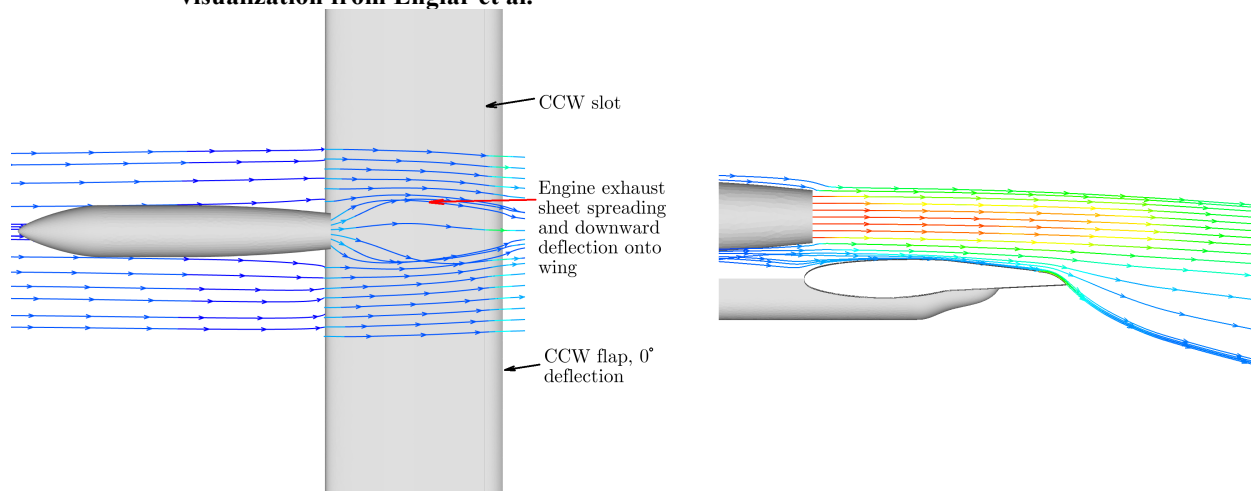


Figure 22. Configuration B with jet flow entrainment shown by streamlines. Streamlines are colored by increasing velocity magnitude from blue to red. $C_T=2.1$, $C_\mu=0.631$

VI. Conclusion

This paper presented CFD methods in validating experimental data of a circulation control airfoil coupled with upper surface blowing from a 3D engine. The motivation behind this validation was to develop grid generating and solving techniques that will be applied to future CFD analysis of other aircraft configurations equipped with over the wing engine and circulation control. Numerous validations have been performed on circulation control airfoils and it's been observed numerous times that current CFD codes over-predict lift and drag coefficient of the airfoil because of the code's inability to match the jet separation and the streamline turning of the jet exhaust. This failure was linked to the available turbulence models and CFD grid issues.

This validation was performed on a single configuration to determine the effects of variations in thrust coefficients and slot blowing coefficients on lift and drag. Thrust coefficient ranged from 0 to 3.36 and slot blowing coefficient ranged from 0 to 0.62. Aerodynamic test results showed that the grid generation method presented in this paper and the use of $k-\epsilon$ turbulence model still over-predicted lift and drag coefficient. These results agreed with the general trend found in using two-equations turbulence models, such as the $k-\epsilon$ and $k-\omega$ turbulence models, in which C_L is over-predicted as C_μ increases. For $k-\epsilon$ and at constant thrust coefficient, lift was over-predicted by approximately 12% at the lowest momentum coefficient but increased to approximately 20% at the highest

momentum coefficient. Similarly, at a constant momentum coefficient, lift was over-predicted by approximately 15% at the lowest thrust coefficient and increases to approximately 25% at the highest thrust coefficient. For $k-\omega$ and at constant thrust coefficient, lift was over predicted by 17% and increased to 23% as the slot blowing coefficient increased. At a constant slot blowing coefficient, lift was over predicted by 28% and increased to 33% as the thrust coefficient increased.

Validating the experimental thrust and slot blowing coefficients was important to ensure that the flow features of the exhaust and jet velocity were accurately captured. Results showed that the relative error in thrust coefficient decreased at higher thrust coefficients. Similarly, the relative error in momentum coefficient decreased at higher momentum coefficients only if the exhaust from the engine was present. At a thrust coefficient of zero, the relative error of the momentum coefficient actually increased at higher momentum coefficients. This was linked to the difference in how the experimental C_μ and the CFD C_μ were calculated. When $C_T \neq 0$, the process in which the CFD C_μ was calculated became more and more similar to the calculation of the experimental C_μ .

Results of this validation strongly agree with other literature that grid issues and turbulence models caused an over prediction of the lift and drag coefficients. Even with the added complexity of an engine, the results still followed the general trend that is seen from two-equation turbulence models. Unfortunately, the development of the modified v^2-f turbulence model did not finish on time for this validation. Future work includes analyzing more turbulence models, including the v^2-f .

Acknowledgments

This work was funded as part of a NASA Research Announcement award under Contract #NNL07AA55C with Craig Hange and Joe Posey as the technical monitors. The authors wish to thank Bob Englar and his staff at Georgia Tech Research Institute for developing the designs and experiments of the powered lift, upper surface blowing configurations that were the basis of this validation. The authors also wish to thank the hard work of Bryan Blessing, Rory Golden, John Pham, Travis Storm, Jon Lichtwardt and Eric Paciano.

References

1. Englar, R. J., Blaylock, G. H. Gaeta, R. J., “*Recent Experimental Development of Circulation Control Airfoils and Pneumatic Powered-Lift Systems*” AIAA Paper 2010-0345, presented at 48th AIAA Aerospace Science Meeting, Orlando, FL, January 4-7, 2010.
2. Ansys ICEM CFD, Software Package, Vers. 11.0.1. <www.ansys.com>
3. Ansys FLUENT, Software Package, Vers. 6.3.26, Fluent Inc. <www.ansys.com>
4. Harlow, F. H., and Nakayama, P. I., “*Transport of Turbulence Energy Decay Rate*,” Los Alamos Sci. Lab., LA-3854, 1968.
5. Wilcox, D. C., “*Reassessment of the Scale-Determining Equation of Advanced Turbulence Models*,” AIAA Journal vol. 26, pp. 1299-1310, 1988.
6. Storm, T., and Marshall, D. D., “*Assessing the v^2-f Turbulence Models for Circulation Control Applications*” presented at 48th AIAA Aerospace Science Meeting, Orlando, FL, January 4-7, 2010.
7. Celik, I. B., Ghia, U., Roache, P. J., and Freitas, C. J., “*Procedure for Estimation and Reporting of Uncertainty Due to Discretization in CFD Applications*,” Journal of Fluids Engineering.
8. Durbin, P. A., “*Near-Wall Turbulence Closure without ‘Damping Functions’*,” Theoretical and Computational Fluid Dynamics 3:1-13, 1991.
9. Pettersson Reif, B. A., Durbin, P. A., and Ooi, A., “*Modeling Rotational Effects in Eddy-Viscosity Closures*,” International Journal of Heat and Fluid Flow, 1999.
10. Duraisamy, K., and Iaccarino, G., “*Curvature Correction and Application of the v^2-f Turbulence Model to tip Vortex Flows*,” Center for Turbulence Research Annual Research Briefs, 2005.
11. Jones, G. S., Lin, J. C., Allan, B. G., Milholen, W. E., Rumsey, C. L., Swanson, R.C., “*Overview of CFD Validation Experiments Circulation Control Applications at NASA*,” 2008 International Powered Lift Conference, 22-24 Jul.

2008, London, United Kingdom.