REPORT DOCUMENTATION PAGE

1

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)
29-08-2014

2. REPORT TYPE
Conference Proceeding

3. DATES COVERED (From - To)
-

4. TITLE AND SUBTITLE
Turbulent Flow Validation in the Helios Strand Solver

5a. CONTRACT NUMBER
W911NF-12-1-0008

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
622307

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

6. AUTHORS
Andrew M. Wissink, Aaron J. Katz, Nicholas K. Burgess

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
Utah State University
1415 Old Main Hill - Room 64

Logan, UT 84322 -1415

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
60390-EG.27

12. DISTRIBUTION AVAILIBILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT
This paper will present a validation of the turbulence modeling approaches used in the Helios code. He- lios is a dual-mesh overset unstructured/Cartesian RANS solver developed by the US Army for high-fidelity simulation of rotorcraft aeromechanics. The paper will present validations of standard 3D flowfields using the current near-body unstructured code NSU3D as well as the new strand mesh solver that is intended to be adopted over the next few years.

15. SUBJECT TERMS
turbulence modeling, Helios, strand grids

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Aaron Katz
UU	UU	UU	UU		19b. TELEPHONE NUMBER 435-797-7021

Report Title

Turbulent Flow Validation in the Helios Strand Solver

ABSTRACT

This paper will present a validation of the turbulence modeling approaches used in the Helios code. He- lios is a dual-mesh overset unstructured/Cartesian RANS solver developed by the US Army for high-fidelity simulation of rotorcraft aeromechanics. The paper will present validations of standard 3D flowfields using the current near-body unstructured code NSU3D as well as the new strand mesh solver that is intended to be adopted over the next few years.

Conference Name: AIAA SciTech 2014

Conference Date: January 07, 2014

Turbulent Flow Validation in the Helios Strand Solver

Andrew M. Wissink[*]
Aaron J. Katz[†]
Nicholas K. Burgess[‡]

This paper will present a validation of the turbulence modeling approaches used in the Helios code. Helios is a dual-mesh overset unstructured/Cartesian RANS solver developed by the US Army for high-fidelity simulation of rotorcraft aeromechanics. The paper will present validations of standard 3D flowfields using the current near-body unstructured code NSU3D as well as the new strand mesh solver that is intended to be adopted over the next few years.

I. Introduction

Rotorcraft computations are challenging due to their dynamic and multidisciplinary nature. Rotor blades experience all the complex aerodynamic flow conditions that complicate traditional fixed-wing CFD predictions - viscous effects, compressibility, turbulence, and stall - except that rotors may experience all these conditions within a single rotor revolution. Moreover, rotorcraft flowfields demand extremely accurate resolution of the wake vortices over relatively long distances because of the importance of blade-vortex interactions and fuselage effects. On the structural dynamics side, blades experience considerable aero-elastic effects with highly unsteady control loads transmitted through a structurally complex hub connection. The strong coupling between the aerodynamic and structural dynamics simulations must also be coupled with vehicle flight dynamics and controls software to achieve a trimmed vehicle state. See the reviews by Strawn et al.[1] and Datta and Johnson[2] for further details on the complexities of rotorcraft modeling.

The Helios software was introduced as a rotary-wing product of the of the CREATE-AV (air vehicles) program,[3] sponsored by the Department of Defense High Performance Computing (HPC) Modernization Office, specifically to provide a high-fidelity analysis capability to the DoD for the acquisition of new rotary-wing aircraft. The dual-mesh paradigm (Fig. 1) that is the basis of the CFD aerodynamics solution procedure consists of unstructured meshes in the near-body to capture viscous flow around complex geometry, and block structured Cartesian grids in the off-body to resolve the wake through a combination of high-order algorithms and adaptive mesh refinement (AMR). An overset procedure facilitates data exchange between the two mesh types as well as enables relative motion between the mesh systems - i.e. the near-body unstructured rotor meshes rotate and deform inside the stationary adaptive Cartesian off-body grids system. Rotor motion, deformation, flight controls and trim operations are provided by an external comprehensive analysis package. Coordination of the different codes is managed through a lightweight and flexible Python-based infrastructure.

The latest version of the software, Helios v3 Rainier,[4] was released to government and industry users in 2012. It is capable of simulating isolated or coupled rotors and fuselages, multiple rotors, and CFD/CSD coupled cases. The both the current near-body unstructured[5] and off-body Cartesian[6] solvers have Detached Eddy Simulation (DES) turbulence modeling capability. The paper will present validations of Helios on the turbulent flow test cases as defined by the AIAA Fluid Dynamics Technical committee.

A new near-body solver[7] is currently under development that operates on automatically-constructed Strand[8] meshes. Validations with this solver will also be presented.

II. DES Turbulence Modeling

Earlier versions of Helios utilized the RANS solver in NSU3D for the near-body with the high-order inviscid Euler solver in SAMARC in the off-body. This paradigm is considered sufficient as long as the flow is attached but when it

[*] U.S. Army Aerodynamics Development Directorate (AMRDEC), Moffett Field CA

[†] Mechanical and Aerospace Engineering, Utah State University, Logan UT

[‡] Science & Tech. Corp, NASA Ames Research Center, Moffett Field CA

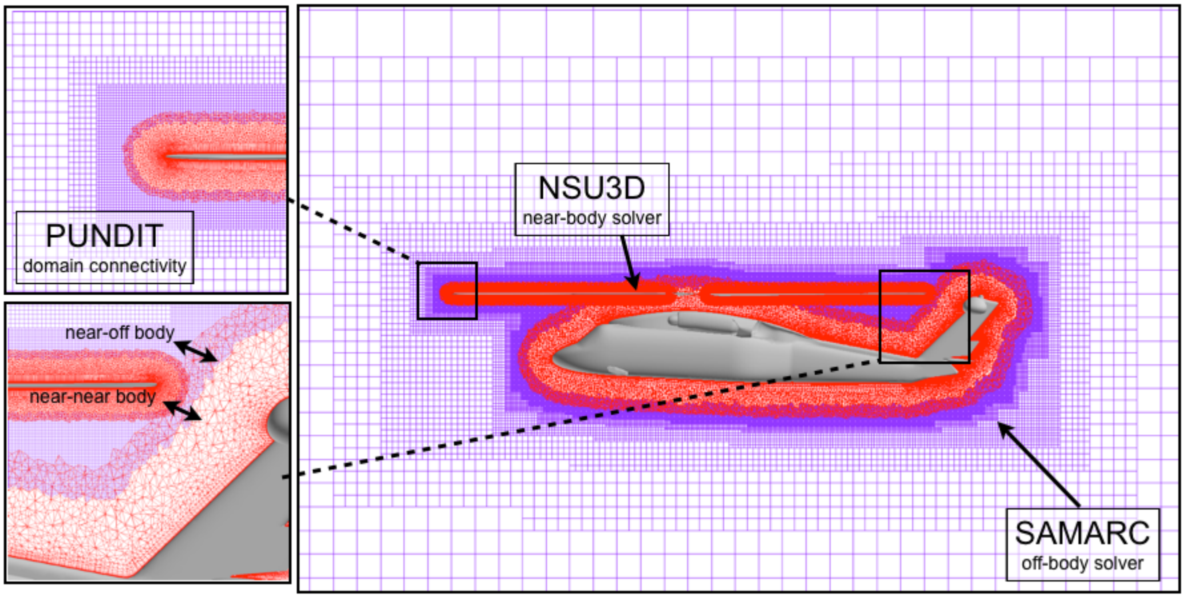


Figure 1. Overset near/off-body gridding paradigm used in `Helios`. Unstructured or curvilinear grids to capture geometric features and boundary layer near body surface, adaptive block-structured Cartesian grids to capture far-field flow features.

becomes separated, which occurs on the rotor in dynamic stall or on the backward side of fuselage protrusions, RANS tends to overpredict the lift and drag. In Rainier the solvers have been augmented in two ways. First, the Detached Eddy Simulation (DES) has been enabled in NSU3D in order to better model the wakes of separated flows. Second, laminar viscous terms have been enabled in the off-body. Further details of these two updates along with validation results are given in the following sub-sections.

II.A. Near-body DES/DDES Formulation

Standard one and two-equation turbulence models used in RANS do a steady average over all turbulent length scales to predict the turbulent flow characteristics. When the turbulence is generated under time-dependent conditions, such as separated flows, a sub-grid scale Large Eddy Simulation (LES) formulation is a better approximation. LES resolves the turbulent length scales to the degree allowed by the computational mesh, then switches to a subgrid scale model for the turbulent length scales that cannot be resolved by the mesh. Since LES is expensive if applied to the very fine viscous grid scales near the wall, a good compromise is the hybrid DES model, which applies standard RANS near the wall and switches to subgrid-scale LES depending on grid density. That is, in regions where the grid cannot support the turbulence length scale, generally near wall boundaries, are solved using traditional RANS. Regions where the grid is length scale is sufficient to resolve the turbulent length scale are solved using LES.

Implementation of DES involves adding the LES model and changing the length scale used in the RANS Spalart-Allmaras turbulence model. The length scale can be written as:

$$\widetilde{d} = d - f_d \max\left(0, d - C_{DES}\Delta\right) \tag{1}$$

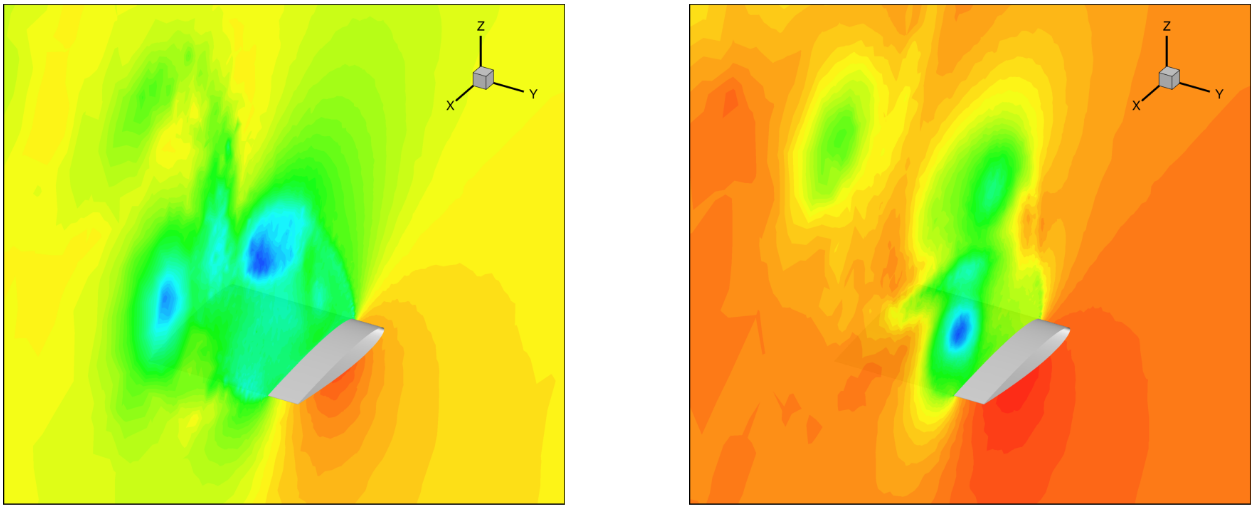
where d is the distance from the wall, $C_{DES} = 0.65$, and Δ a measure of the local grid spacing.[9] In NSU3D, Δ is set to the maximum edge length touching a given vertex on unstructured mesh.[10] If f_d is set to 1, then equation (1) becomes:

$$\widetilde{d} = \min\left(d, C_{DES}\Delta\right) \tag{2}$$

The length scale in this original formulation is grid dependent causing incorrect behavior to be observed for some cases with ambiguous grid densities. The Delayed-DES (DDES) model[11] was introduced to be less sensitive to the grid. DDES computes f_d by:

$$f_d = 1 - \tanh\left(\left[8r_d\right]^3\right) \tag{3}$$

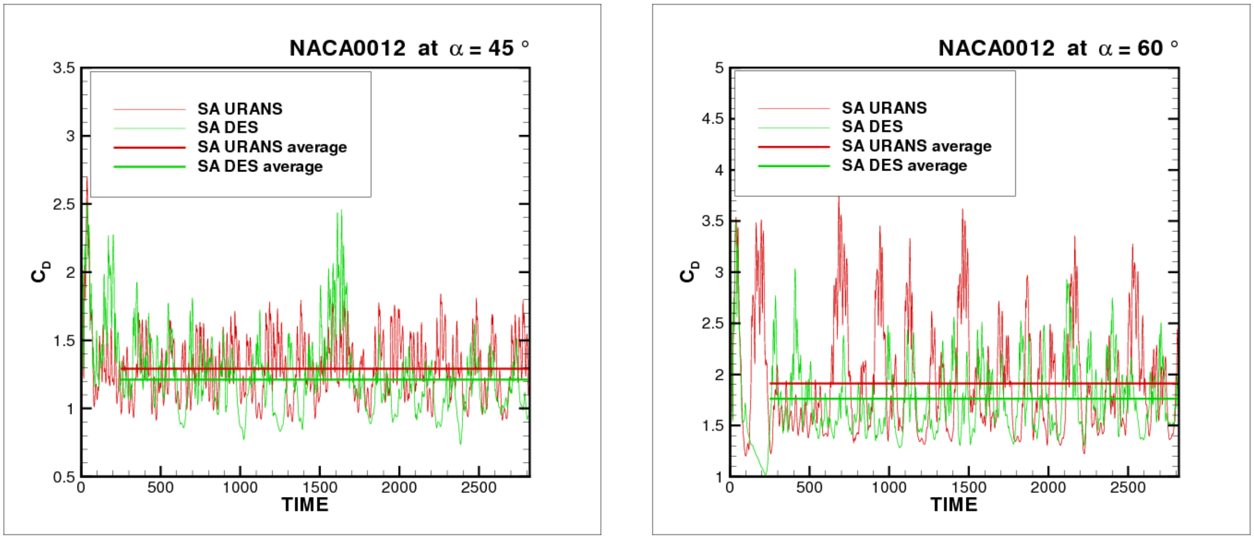
where $r_d = \frac{\nu_t + \nu}{\sqrt{U_{i,j}U_{i,j}}\kappa^2 d^2}$. ν_t and ν are the kinematic eddy viscosity and the molecular viscosity, respectively. $U_{i,j}$ are the velocity gradients. κ is the Karman constant. Either the DES or DDES options can be invoked in NSU3D.



(a) $\alpha = 45^o$ (b) $\alpha = 60^o$

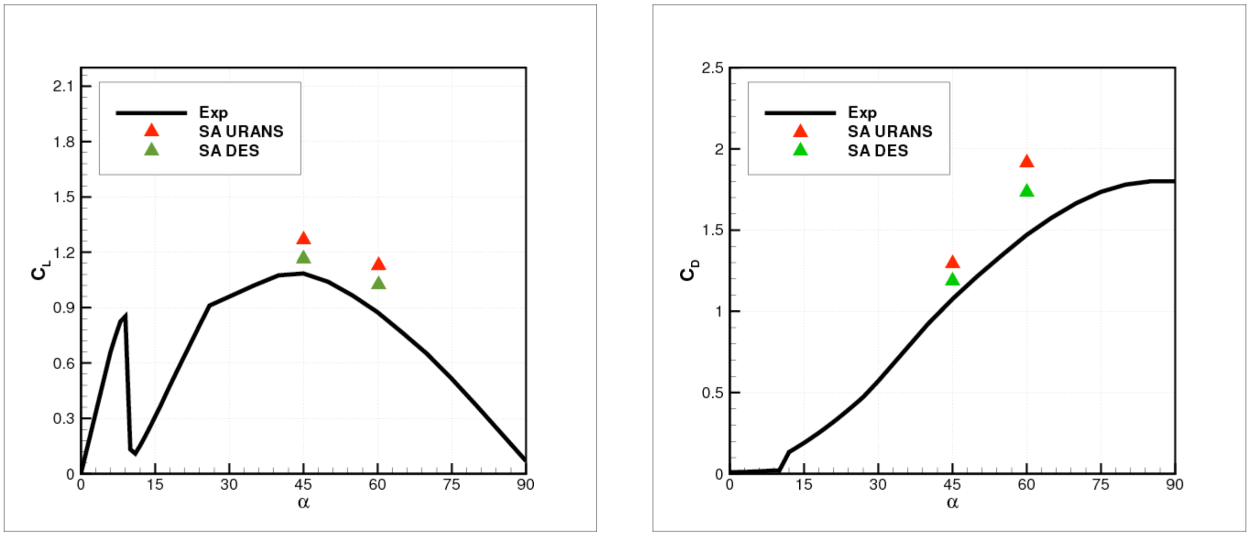
Figure 2. Stalled NACA 0012 wing at $M_\infty = 0.15$, computed density contours with DES turbulence model.

The DES capability is evaluated for calculations about a NACA 0012 airfoil at $M_\infty = 0.15, Re = 10^5$ at 45^o and 60^o angle of attack (AOA). The airfoil is fully stalled at these high angles of attack and experiences time-dependent shedding behavior. Figure 2(a) and (b) shows the density contour solution at the $y = 0.8$ spanwise location for the 45 and 60 degrees AOA cases. Figure 3 shows the computed drag coefficient, both the time dependent shedding behavior and the average over the simulation. Figure 4 shows the computed average C_L and C_D with DES and traditional RANS compared to the experimentally measured quantities.[12] With DES turned on the computed results more closely approximate the measured experimental values.



(a) $\alpha = 45^o$ (b) $\alpha = 60^o$

Figure 3. Time dependent and average computed drag coefficients, RANS vs. DES.



(a) C_L

(b) C_D

Figure 4. Measured and computed average lift and drag coefficients using RANS and DES.

II.B. DES Off-Body

The conservative form of the compressible Reynolds Averaged Navier-Stokes (RANS) equations describing the conservation of mass, momentum and total energy in three dimensions are given as:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \left(\vec{\mathbf{F}}_c(\mathbf{u}) - \vec{\mathbf{F}}_v(\mathbf{u}, \nabla \mathbf{u}) \right) = \mathbf{S}(\mathbf{u}, \nabla \mathbf{u}) \tag{4}$$

subject to the appropriate boundary and initial conditions within a domain Ω. In this work the RANS equations are coupled to the one equation turbulence model of Spalart and Allmaras (SA model)[18] with a modified production term due to Johnson and Allmaras, given in reference.[?]

II.B.1. *Spalart Allmaras Turbulence model*

In this work two forms of the SA turbulence model are employed: one corresponding to the traditional RANS version (SA-RANS) and the other corresponding to the DES version (SA-DES). Begining with the RANS version of the turbulence model which is given by:

$$\frac{\partial \rho \tilde{\nu}}{\partial t} + \nabla \cdot (\rho \tilde{\nu} \vec{v}) = \mathcal{P}(\mathbf{u}) + \frac{1}{\sigma} \left[\nabla \cdot ((\mu + \rho \tilde{\nu}) \nabla \tilde{\nu}) + c_{b_2} \rho \nabla \tilde{\nu} \cdot \nabla \tilde{\nu} \right] - \mathcal{D}(\mathbf{u}) \tag{5}$$

The production term $\mathcal{P}(\tilde{\nu})$is given as:

$$\mathcal{P}(\mathbf{u}) = c_{b_1} \tilde{S} \rho \tilde{\nu} \tag{6}$$

where $\tilde{S}$ is given according to

$$\tilde{S} = \begin{cases} S + \bar{S} & \bar{S} \geq -c_{v_2} S \\ S + \frac{S\left(c_{v_2}^2 S + c_{v_3} \bar{S}\right)}{\left(c_{v_3} - 2c_{v_2}\right) S - \bar{S}} & \bar{S} \leq -c_{v_2} S \end{cases} \tag{7}$$

$$\begin{aligned} S &= \sqrt{\vec{\omega} \cdot \vec{\omega}} - C_{rot} \min\left(0, \sqrt{\vec{\omega} \cdot \vec{\omega}} - S_{ij} S_{ji}\right) \\ \bar{S} &= \frac{\tilde{\nu}^2 f_{v_2}}{\kappa^2 d^2} \end{aligned} \tag{8}$$

where $\vec{\omega}$ is the vorticity vector and S_{ij} is the strain rate tensor (specified below). The wall distance function is given as:

$$d = \min\left(d_w, 0.65 \Delta_{xyz}\right) \tag{9}$$

where Δ_{xyz} is the grid size and d_w is the distance to the closest wall. In RANS mode d is simply set to $d = d_w$. Since the off-body solver does not directly contain any walls $d_w = \infty$. The constants and functions f_{v_1}, f_{v_2} are the same as those in reference.[18] The additional constant c_{v_3} is given as $c_{v_3} = .9$. The destruction term is unmodified from the original version in reference.[18]

$$\mathcal{D}(\mathbf{u}) = c_{w_1}\rho f_w \left(\frac{\tilde{\nu}}{d}\right)^2 \tag{10}$$

The above represents a conservative version of the turbulence model given in reference.[18]

II.B.2. SA-RANS System

The state vector and flux vectors including those of the SA model equation for three-dimensional flow are explicitly given as:

$$\mathbf{u} = \left\{\begin{array}{c} \rho \\ \rho u \\ \rho v \\ \rho w \\ E_t \\ \rho\tilde{\nu} \end{array}\right\}, \quad \mathbf{F_c}^x = \left\{\begin{array}{c} \rho u \\ \rho u^2 + P \\ \rho v u \\ \rho w u \\ u(E_t + P) \\ \rho\tilde{\nu}u \end{array}\right\}, \quad \mathbf{F_c}^y = \left\{\begin{array}{c} \rho v \\ \rho u v \\ \rho v^2 + P \\ \rho w v \\ v(E_t + P) \\ \rho\tilde{\nu}v \end{array}\right\}, \quad \mathbf{F_c}^z = \left\{\begin{array}{c} \rho w \\ \rho u w \\ \rho v w \\ \rho w^2 + P \\ w(E_t + P) \\ \rho\tilde{\nu}w \end{array}\right\},$$

$$\mathbf{F_v}^x = \left\{\begin{array}{c} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u\tau_{xx} + v\tau_{xy} + w\tau xz + c_p\left(\frac{\mu}{Pr} + \frac{\mu_T}{Pr_T}\right)\frac{\partial T}{\partial x} \\ \frac{1}{\sigma}(\mu + \rho\tilde{\nu})\frac{\partial\tilde{\nu}}{\partial x} \end{array}\right\}, \quad \mathbf{F_v}^y = \left\{\begin{array}{c} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ u\tau_{yx} + v\tau_{yy} + w\tau_{yz} + c_p\left(\frac{\mu}{Pr} + \frac{\mu_T}{Pr_T}\right)\frac{\partial T}{\partial y} \\ \frac{1}{\sigma}(\mu + \rho\tilde{\nu})\frac{\partial\tilde{\nu}}{\partial y} \end{array}\right\}, \tag{11}$$

$$\mathbf{F_v}^z = \left\{\begin{array}{c} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ u\tau_{zx} + v\tau_{zy} + w\tau_{zz} + c_p\left(\frac{\mu}{Pr} + \frac{\mu_T}{Pr_T}\right)\frac{\partial T}{\partial z} \\ \frac{1}{\sigma}(\mu + \rho\tilde{\nu})\frac{\partial\tilde{\nu}}{\partial z} \end{array}\right\}, \quad \mathbf{S} = \left\{\begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ c_{b_1}\tilde{S}\rho\tilde{\nu} + \frac{1}{\sigma}\left[c_{b_2}\rho\nabla\tilde{\nu}\cdot\nabla\tilde{\nu}\right] - c_{w_1}\rho f_w\left(\frac{\tilde{\nu}}{d}\right)^2 \end{array}\right\}$$

where ρ is the fluid density, ($\vec{u} = (u, v, w)$) are the Cartesian velocity components, P is the fluid pressure, E_t is the total energy, c_p is the specific heat at constant pressure, T is the fluid temperature, P_r and P_{r_T} are the Prandtl and turbulent Prandtl numbers respectively and τ_{ij} is the total viscous stress tensor including the Boussinesq approximated Reynolds stresses. Assuming a Newtonian fluid and using the Boussinesq approximation for the Reynolds stresses, the viscous stress tensor takes the form (with $x_i = x, y, z; \quad i = 1, 2, 3$):

$$\begin{aligned} &\tau_{ij} = 2(\mu + \mu_T)S_{ij} \\ &S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij} \\ &\text{for } i = 1, 2, 3 \text{ and } j = 1, 2, 3 \end{aligned} \tag{12}$$

where μ is the fluid viscosity obtained via Sutherland's law and μ_T is a turbulent eddy viscosity, which is given by:

$$\begin{aligned} &\mu_T = \left\{\begin{array}{cc} \rho\tilde{\nu}f_{v_1} & \tilde{\nu} \geq 0 \\ 0 & \tilde{\nu} < 0 \end{array}\right. \\ &f_{v_1} = \frac{\left(\frac{\rho\tilde{\nu}}{\mu}\right)^3}{\left(\frac{\rho\tilde{\nu}}{\mu}\right)^3 + c_{v_1}^3} \\ &c_{v_1} = 7.1 \end{aligned} \tag{13}$$

It should be understood that all quantities in the above equations are the Reynolds Averaged quantities (the usual $\bar{()}$ notation is omitted for simplicity). The pressure is obtained from the ideal gas equation of state given as:

$$P = (\gamma - 1)\left[E_t - \frac{1}{2}\rho\left(u^2 + v^2 + w^2\right)\right] \tag{14}$$

where $\gamma = 1.4$ is the ratio of specific heats.

II.B.3. Spatial Discretization

The discretization of the turbulence model equation is a critical components of a CFD solver. The turbulence model has a significant impact on the flow solution and therefore it must be discretized with a method that is both sufficiently accurate and robust. If the discretization is too dissipative then insufficient levels of eddy viscosity are produced.[?, ?] However, recent work[?, ?, ?, ?] has demonstrated that discretizing the turbulence model to higher than first-order accuracy requires significant adjustment to both the continuous definition of the turbulence model and/or the discretization, convective flux function and source term treatments in order to obtain a robust solution. Furthermore, reference[?] has demonstrated that the turbulence model discrete solution is very sensitive to the choice of convective discretization. Therefore, for explicit clarity, the discretization of the convection and diffusion terms of the turbulence model are detailed in this section. The discretization used in this work is less than optimal, since the form of the model employed is only advantageous when using an asymptotically first-order accurate discretization. However, the discretization is robust and is an example of what can be considered common practice for turbulence model discretizations.[?, ?, ?, 18]

II.B.4. Convection Terms

In order to simplify the notation, the discretization is derived in one dimension, with the understanding that in a three dimensional setting the method is applied to each direction independently. Since the current version of SAMARC employes a high-order finite-difference discretization for the convection terms of the RANS equations, the turbulence model will also be discretized using a finite-difference discretization. In one spatial dimension the convection term of the SA turbulence model equation is given as:

$$\frac{\partial \rho\tilde{\nu}u}{\partial x} \tag{15}$$

which is discretized using an asymptotically first-order upwind finite-difference method. Consider that the domain has been gridded and the flow solution is defined as a grid function such that $u \approx \{u_i\}$. The the first-order upwind finite-difference of the flux derivative at the grid node i is given as:

$$\frac{\partial E^{\rho\tilde{\nu}}}{\partial x} \approx \frac{(\rho\tilde{\nu}u)_{i+1/2} - (\rho\tilde{\nu}u)_{i-1/2}}{\Delta x} \tag{16}$$

This is essentially a decoupled flux formulation where the turbulence model is upwinded based solely on the direction of the convective velocity and up winding due to acoustic propagation are ignored.[?] In order to maintain a first-order accurate discretization and also maintain pure upwind approach for a constant velocity field the flux at the half node $i + 1/2$ is given as

$$(\rho\tilde{\nu}u)_{i+1/2} = \frac{u_i}{2}\left((\rho\tilde{\nu})_i + (\rho\tilde{\nu})_{i+1} + sign(u_i)\left((\rho\tilde{\nu})_i - (\rho\tilde{\nu})_{i+1}\right)\right) \tag{17}$$

and the flux at the half node $i - 1/2$ is given as:

$$(\rho\tilde{\nu}u)_{i-1/2} = \frac{u_i}{2}\left((\rho\tilde{\nu})_{i-1} + (\rho\tilde{\nu})_i + sign(u_i)\left((\rho\tilde{\nu})_{i-1} - (\rho\tilde{\nu})_i\right)\right) \tag{18}$$

In essence this method convects the turbulence model variable $\rho\tilde{\nu}$ into the control volume surrounding the node i in an upwind fashion at velocity u_i. Note that this approach defines the velocity at the half nodes using a piecewise constant extrapolation form node i.

II.B.5. Diffusion Terms

One of the principal concerns with discretizing the SA turbulence model equation is the diffusion term treatment. The diffusion term consists of two parts given as:

$$\frac{1}{\sigma}\left[\underbrace{\nabla\cdot\left((\mu + \rho\tilde{\nu})\nabla\tilde{\nu}\right)}_{1} + \underbrace{c_{b2}\rho\nabla\tilde{\nu}\cdot\nabla\tilde{\nu}}_{2}\right] \tag{19}$$

While the term labeled 1 in Eq. (19) can discretized without difficultly there is some debate as to how to discretize the second term. If one were to be general and rigorous one treat this as a source term since it cannot be case in a divergence form. However, since a first-order upwind formulation is employed for the convective fluxes one can re-write the entire diffusion term as:

$$\frac{1}{\sigma}\left[\nabla\cdot\left(\left(\mu+\rho\tilde{\nu}\right)\nabla\tilde{\nu}\right)+c_{b_2}\rho\nabla\tilde{\nu}\cdot\nabla\tilde{\nu}\right]=\frac{1}{\sigma}\left[\nabla\cdot\left(\left(\mu+\rho\tilde{\nu}\left(1+c_{b_2}\right)\right)\nabla\tilde{\nu}\right)-c_{b_2}\rho\tilde{\nu}\nabla^2\tilde{\nu}\right] \tag{20}$$

which is the so-called non-conservative diffusion term.[18] Since the solution is asymptotically first-order accurate one can regard the $\rho\tilde{\nu}$ in the second term as constant on the stencil and discretize the laplacian of $\tilde{\nu}$ using a second-order accurate central difference as:

$$c_{b_2}\rho\tilde{\nu}\nabla^2\tilde{\nu}\approx c_{b_2}\left(\rho\tilde{\nu}\right)_i\left(\frac{\tilde{\nu}_{i+1}-2\tilde{\nu}_i+\tilde{\nu}_{i-1}}{\Delta x^2}\right) \tag{21}$$

The first diffusion term is discretized using a generalization of the central difference of the second derivative:

$$\begin{aligned}&\nabla\cdot\left(\left(\mu+\rho\tilde{\nu}\left(1+c_{b_2}\right)\right)\nabla\tilde{\nu}\right)\approx\frac{1}{\Delta x^2}\left(\frac{\eta_{i+1}}{2}\left(\tilde{\nu}_{i+1}-\tilde{\nu}_i\right)+\frac{\eta_i}{2}\left(\tilde{\nu}_{i+1}-2\tilde{\nu}_i+\tilde{\nu}_{i-1}\right)+\frac{\eta_{i-1}}{2}\left(\tilde{\nu}_{i-1}-\tilde{\nu}_i\right)\right)\\&\eta_i=\mu_i+\rho\tilde{\nu}_i\left(1+c_{b_2}\right)\end{aligned} \tag{22}$$

Adding these two pieces together and grouping the terms by the stencil point gives the following final formula for the discretization of the entire diffusion term as:

$$\begin{aligned}&\frac{1}{\sigma}\left[\nabla\cdot\left(\left(\mu+\rho\tilde{\nu}\left(1+c_{b_2}\right)\right)\nabla\tilde{\nu}\right)-c_{b_2}\rho\tilde{\nu}\nabla^2\tilde{\nu}\right]\approx\frac{1}{\Delta x^2}\frac{1}{\sigma}\left(\tilde{\nu}_{i+1}D_{i+1}-\tilde{\nu}_iD_i+\tilde{\nu}_{i-1}D_{i-1}\right)\\&D_{i-1}=\frac{\eta_i+\eta_{i-1}}{2}-c_{b_2}\rho\tilde{\nu}_i\\&D_i=\frac{\eta_{i+1}}{2}+\eta_i+\frac{\eta_{i-1}}{2}-2c_{b_2}\rho\tilde{\nu}_i\\&D_{i+1}=\frac{\eta_{i+1}+\eta i}{2}-c_{b_2}\rho\tilde{\nu}_i\end{aligned} \tag{23}$$

This formula is formally second-order accurate, but since the convection terms are first-order accurate the second-order accuracy is not critical.

II.B.6. Source Terms

The source term discretization is trivial compared to the convection and diffusion terms. The source terms are simply evaluated point-wise for each node in the grid. However, the vorticity and strain terms required for the production term require gradients of the velocity field. These gradients are obtained via a central difference of the first derivative:

$$\left.\frac{\partial u}{\partial x}\right|_i\approx\frac{u_{i+1}-u_{i-1}}{2\Delta x} \tag{24}$$

from which the vorticity and strain rate magnitude are derived using the standard formulas:

$$\begin{aligned}&\sqrt{\vec{\omega}\cdot\vec{\omega}}=\sqrt{\left(\frac{\partial w}{\partial y}-\frac{\partial v}{\partial z}\right)^2+\left(\frac{\partial u}{\partial z}-\frac{\partial w}{\partial x}\right)^2+\left(\frac{\partial v}{\partial x}-\frac{\partial u}{\partial y}\right)^2}\\&S_{ij}S_{ji}=\sqrt{S_{11}^2+2S_{12}^2+2S_{13}^2+S_{22}^2+2S_{23}^2+S_{33}^2}\\&S_{ij}=\frac{1}{2}\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)\end{aligned} \tag{25}$$

where all the derivatives are evaluated at a grid node according to the formula in Eq. (24).

II.B.7. Explicit Time-stepping

The SAMARC solver was originally developed to solve the Euler equations in the off-body region of the computational domain, and consequently employs explicit time-stepping for temporal discretization. The explicit nature of SAMARC has become deeply embedding within the design of the software and it is beyond the scope of this work to change this fundamental assumption. Therefore in order to ensure the robustness of a turbulent version of SAMARC, the time-step must be redefined to account for the additional complexity of the turbulence model source and diffusion terms.

In order to develop the time-step definition for the turbulent version of SAMARC a linear convection diffusion model equation is employed to establish the stable explicit time-step. Furthermore, the equally spaced Cartesian nature of the SAMARC grids allows one to do analysis in one spatial dimension. Multi-dimensional results come summing the results in three dimensions. The linear convection diffusion model equation is give as:

$$\frac{\partial u}{\partial t} + \frac{\partial (au)}{\partial x} - \nu \frac{\partial^2 u}{\partial x^2} = Su \tag{26}$$

Eq. (26) is discretized using a first order upwind scheme for the convection term and a second order central difference for the diffusion term. The time derivative is discretized using a first order forward Euler method. The fully discretized Eq. (26) is

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + a\frac{u_i^n - u_{i-1}^n}{\Delta x} - \nu \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} = Su_i^n \tag{27}$$

the equation for new time level value of u_i^{n+1} is

$$u_i^{n+1} = u_i^n - \Delta t a \frac{u_i^n - u_{i-1}^n}{\Delta x} + \Delta t \nu \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} + \Delta t S u_i^n \tag{28}$$

The stability of of this scheme is analyzed by employing VonNeumann analysis on Eq. (28). VonNeumann analysis is valid for period boundary conditions and an equally spaced mesh containing N points covering a domain of length L. One starts by introducing the finite Fourier Transform of the discrete solution

$$\begin{aligned} u_i^n &= \sum_{k=-\frac{N}{2}}^{\frac{N}{2}} \hat{u}^n e^{\hat{i}\frac{2\pi k x_i}{L}} \\ \hat{i} &= \sqrt{-1} \end{aligned} \tag{29}$$

Assuming a periodic solution and boundary conditions one can write the coordinate $x_i = i \cdot \Delta x$. Furthermore, to keep the book keeping simply one can define $\frac{2\pi k \Delta x}{L} = \xi$. It is sufficient to consider an arbitrary Fourier mode for the stability analysis. Substituting a single Fourier mode for the solution into Eq. (28) yields

$$\hat{u}^{n+1} e^{\hat{i}\xi(i)} = \hat{u}^n e^{\hat{i}\xi(i-1)} - \Delta t a \frac{\hat{u}^n e^{\hat{i}\xi(i)} - \hat{u}^n e^{\hat{i}\xi(i-1)}}{\Delta x} + \Delta t \nu \frac{\hat{u}^n e^{\hat{i}\xi(i+1)} - 2\hat{u}^n e^{\hat{i}\xi(i)} + \hat{u}^n e^{\hat{i}\xi(i-1)}}{\Delta x^2} + \Delta t S \hat{u}^n e^{\hat{i}\xi(i)} \tag{30}$$

Simplification and division by $\hat{u}^n$ gives

$$g = \frac{\hat{u}^{n+1}}{\hat{u}^n} = 1 - \frac{\Delta t a}{\Delta x}\left(1 - e^{\hat{i}\xi(-1)}\right) - \frac{\Delta t \nu}{\Delta x^2}\left(e^{\hat{i}\xi(1)} - 2 + e^{\hat{i}\xi(-1)}\right) + \Delta t S \tag{31}$$

Defining the Cell Reynolds number ($Re_{\Delta x}$) and CFL number (σ) as

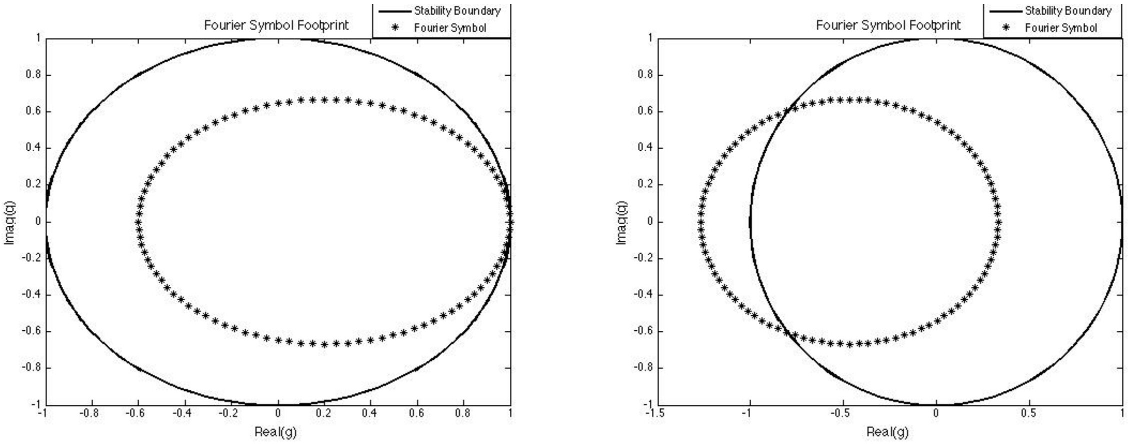
$$\begin{aligned} Re_{\Delta x} &= \frac{a\Delta x}{\nu} \\ \sigma &= \frac{\Delta t a}{\Delta x} \end{aligned} \tag{32}$$

results in

$$g = 1 - \sigma\left(1 - e^{\hat{i}\xi(-1)}\right) - \frac{\sigma}{Re_{\Delta x}}\left(e^{\hat{i}\xi(1)} - 2 + e^{\hat{i}\xi(-1)}\right) + \Delta t S \tag{33}$$

Figure 5(a) shows an example Fourier footprint using $CFL = .75$ and $Re_{\Delta x} = 10$ with zero source term, one should note that all modes are stable. Figure 5(b) shows the Fourier symbol footprint with a non zero source term, demonstrating the onset of instability as a result of adding the source to the equation. The magnitude of the Fourier symbol is

$$\begin{aligned} |g|^2 &= [(1 - \sigma - 2\alpha + S\Delta t) + (\sigma + 2\alpha)\cos(\xi)]^2 + \sigma^2 \sin^2(\xi) \\ \alpha &= \frac{\sigma}{Re_{\Delta x}} \end{aligned} \tag{34}$$



(a) Fourier symbol footprint, $S = 0$

(b) Fourier symbol footprint, $S = -1$

Figure 5. Example Fourier symbol footprint plots using a $CFL = .75$ and $Re_{\Delta x} = 10$, a zero and non zero source term.

which has extrema at $\xi = \pm\pi$. Evaluting Eq. (34) at the $\xi = \pm\pi$ gives

$$max\left(|g|^2\right) = (-1 + 4\alpha - S\Delta t + 2\sigma)^2 \tag{35}$$

The criteria for stability is therefore

$$\begin{aligned} |g|^2 &\leq 1 \\ -1 + 4\alpha - S\Delta t + 2\sigma &\leq \pm 1 \\ \Delta t &\leq \frac{2}{4\frac{\nu}{\Delta x^2} + 2\frac{a}{\Delta x} - S} \end{aligned} \tag{36}$$

This analysis holds for linear problems and before applying it to determine the time-step for SAMARC one must properly interpret and apply the results, in light of the non-linearity of the Navier-Stokes equation. Firstly, the value of a is taken as a_∞, which is the freesream speed of sound. Coincidentally this is the estimate of the maximum eigenvalue currently used in the inviscid version of SAMRC. Secondly, ν which is the viscosity in the analysis should represent the maximum possible diffusion coefficient for the SA turbulence model.

$$\nu = \mu + \max_{\vec{x}}(\rho\tilde{\nu}) \tag{37}$$

Finally, the SA turbulence model has non-linear source terms that my take on positive or negative values. The analysis assumed that S was a positive number therefore the value of S for the Navier-Stokes equations with SA turbulence model is defined as the minmum of zero and the linearization of the SA turbulence model source term with respect to $\rho\tilde{\nu}$.

$$S = \min\left(0, \min_{\vec{x}} \frac{\partial(\mathcal{P}(\mathbf{u}) - \mathcal{D}(\mathbf{u}))}{\partial \mathbf{u}}\right) \tag{38}$$

which will ensure that source term contributions only make the time-step smaller. Putting all this together and accounting for the fact that there are 3 spatial derivative terms gives the final time-step restriction that is used in SAMARC for turbulent computations

$$\Delta t \leq \frac{2}{3\left[4\frac{\mu + \max_{\vec{x}}(\rho\tilde{\nu})}{\Delta x^2} + 2\frac{a_\infty}{\Delta x}\right] - \min\left(0, \min_{\vec{x}}\left(\frac{\partial(\mathcal{P}(\mathbf{u}) - \mathcal{D}(\mathbf{u}))}{\partial \mathbf{u}}\right)\right)} \tag{39}$$

III. Strand Mesh Solver

The overset dual-mesh "strand"–Cartesian approach has been proposed and studied in earlier works[?, 8, 13–17] as a viable means to support automatic viscous mesh generation and adaptation. In the strand paradigm, a body-fitted near-body mesh is constructed by a set of straight line segments grown directly from the surface, each with the same point

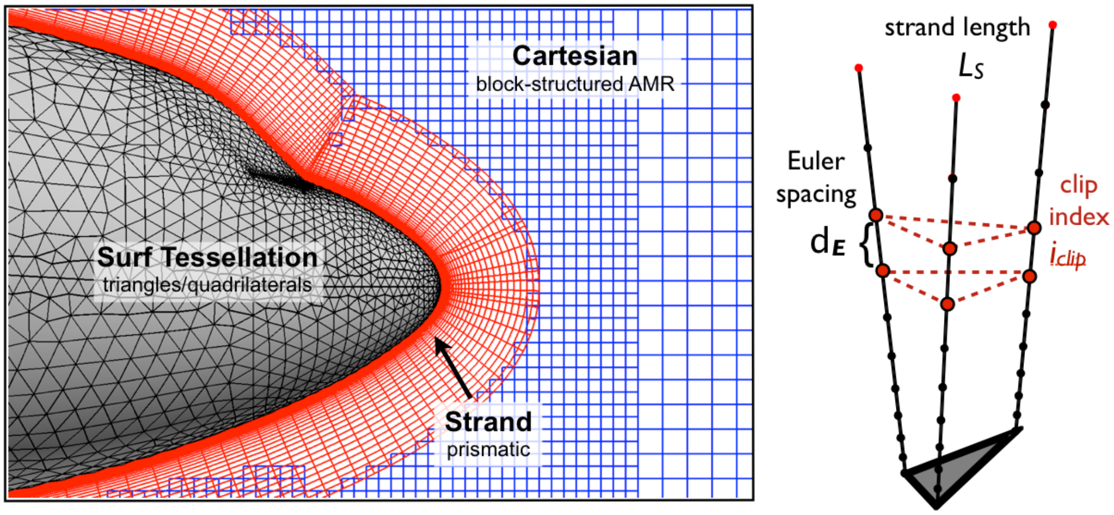


Figure 6. Strand-Cartesian grid system.

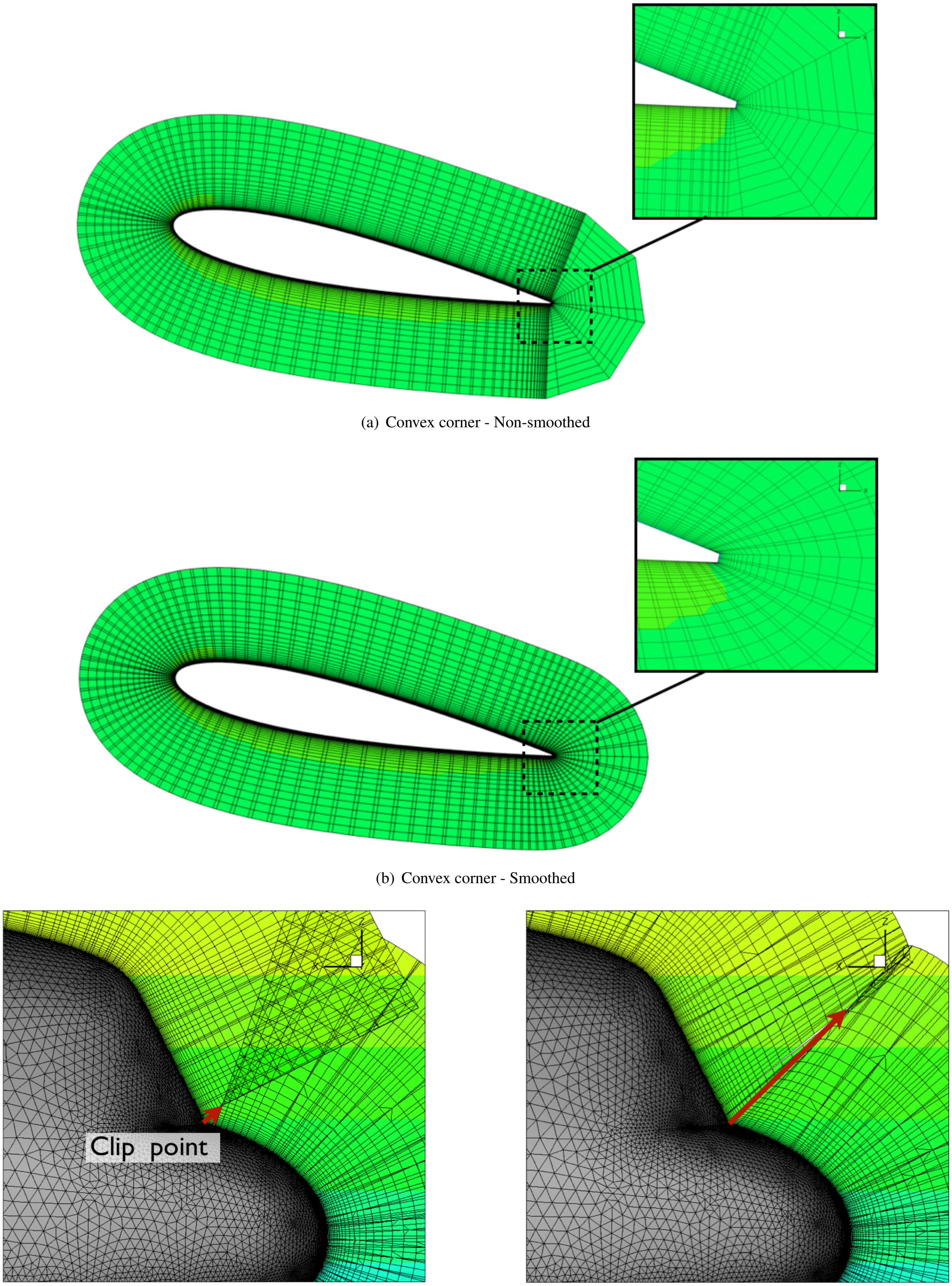
distribution in the normal direction, forming a thin layer of mostly prismatic elements around the body. Once outside the viscous boundary layer, strands transition to isotropic block structured Cartesian grids. The two grid systems intersect through overlapping chimera overset procedures. The procedure is similar in concept to standard prismatic unstructured grid generation techniques, in which prismatic cells are grown at the surface in the viscous boundary layer with tetrahedra elsewhere, except in the strand approach Cartesian grids are used in place of tetrahedra for the Euler solution.

In addition to streamlined and automatic meshing capability, the strand-Cartesian approach presents three other important advantages. First, both strand and Cartesian meshes may be represented with extremely low memory descriptions, enabling the entire global mesh description to fit on each processor in a parallel environment. This allows for significant gains in efficiency and scalability of domain connectivity, effectively eliminating inter-processor search routines. The savings become even more significant in the case of moving body simulations for which domain connectivity must be re-established at each unsteady time-step. Second, both strand and Cartesian meshes possess at least some grid structure, facilitating efficient implementations of high-order accurate discretizations and solution methods. These methods include high-order finite differencing, line-implicit solvers, and directional multigrid coarsening. Third, both the strand and Cartesian grids easily permit use of Adaptive Mesh Refinement (AMR). Because all strands use the same normal point distribution, adaptation is entirely surface-based. This avoids cell quality and edge swapping complexities that have traditionally plagued volume-based unstructured AMR. AMR on Cartesian grids has been known for years to be very effective because the logical data structure naturally facilitates a hierarchical mesh representation and Cartesian cells do not suffer cell quality issues with frequent and persistent adaptation, as can occur with tetrahedral elements.

III.A. Strand Mesh Generation

The starting point for the mesh generation is a tessellated surface composed of either triangles, quadrilaterals, or a mix of both. Strands consist of straight line segments of equal length, number of points, and point distribution, grown from surface vertices. The strands initially are grown normal to the surface and then smoothed to provide coverage in convex corners (Fig. 7a & b) and to push crossing strands away from the surface in concave corners (Fig. 7c & d). The desired degree of smoothing in the mesh may be adjusted at runtime through an input parameter.

Spacing along each strand ranges from viscous at the root to transitional, or “Euler” spacing δ_E at the tip. The set of strands produce a prism stack associated with each surface triangle. Any negative volumes associated with crossing strands are clipped by associating an integer “clip index” i_{clip} with each surface triangle. The clip index may also be used to clip elements that protrude an outer mold line for two strand meshes that lie in close proximity to one another. The user can supply the desired strand length L_S through input or allow the strand length to be computed automatically. If the length L_S is computed automatically, the algorithm seeks to make the transitional spacing δ_E at the strand end equal to the surface tessellation spacing. This ensures the transition cells are roughly isotropic at the strand ends so



(a) Convex corner - Non-smoothed

(b) Convex corner - Smoothed

(c) Concave corner - Non-smoothed

(d) Concave corner - Smoothed

Figure 7. Strand direction vector smoothing.

they transition nicely to Cartesian off-body meshes. The strand length L_S, Euler spacing δ_E, and clip index i_{clip}, are all pictured graphically in Figure 6.

Once a near-body strand mesh is available, an adaptive Cartesian off-body mesh is automatically constructed. The output of the near-body strand mesh generation is a set of prism stacks each with a designated clip index i_{clip} and the normal spacing of the prism element at this clip index, or Euler spacing δ_E, provides the basis for the initial Cartesian grid generation. The location of the clip index elements (x,y,z) and the Euler spacing δ_E are provided to the Cartesian grid generator. Block structured Cartesian grids are built in a hierarchical fashion, the coarsest level defines the physical extent of the computational domain and new levels are constructed from coarsest to finest. Each finer level is formed by selecting cells on the coarser level and then clustering the marked cells together to form block regions that will constitute the new finer level. The Cartesian grid is initially refined to match the Euler spacing δ_E elements in the strand mesh, and Cartesian grids are then adapted throughout the simulation to capture time-dependent solution features such as vorticity. Further details of the off-body mesh generation procedure have been presented in previous work.[17]

IV. RANS-SA Solver

In this work we solve the Reynolds-averaged Navier-Stokes (RANS) equations in three dimensions. Turbulence closure is accomplished with the Spalart-Allmaras (SA) model.[18] The RANS-SA equations may be expressed as

$$\frac{\partial Q}{\partial t} + \frac{\partial F_j}{\partial x_j} - \frac{\partial F_j^v}{\partial x_j} = S, \tag{40}$$

where the conserved variables, Q, inviscid fluxes, F_j, viscous fluxes, F_j^v, and source term, S, are defined as

$$Q = \begin{pmatrix} \rho \\ \rho u_i \\ \rho e \\ \rho\tilde{\nu} \end{pmatrix}, \quad F_j = \begin{pmatrix} \rho u_j \\ \rho u_i u_j + p\delta_{ij} \\ \rho h u_j \\ \rho\tilde{\nu} u_j \end{pmatrix}, \quad F_j^v = \begin{pmatrix} 0 \\ \sigma_{ij} \\ \sigma_{ij} u_i - q_j \\ \frac{\eta}{\sigma}\frac{\partial\tilde{\nu}}{\partial x_j} \end{pmatrix}, \quad S = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \mathcal{P} - \mathcal{D} + C_{b2}\rho\frac{\partial\tilde{\nu}}{\partial x_k}\frac{\partial\tilde{\nu}}{\partial x_k} \end{pmatrix}. \tag{41}$$

Here, ρ is the density, u_i is the Cartesian velocity vector, e is the total energy per unit mass, $\tilde{\nu}$ is the turbulence working variable, p is the pressure, h is the total enthalpy per unit mass, σ_{ij} is the deviatoric stress tensor, q_j is the heat flux vector, and η/σ is the turbulent diffusion coefficient. The turbulent source term consists of a production term, $\mathcal{P}$, and a destruction term $\mathcal{D}$. The stress tensor is defined as

$$\sigma_{ij} = 2(\mu + \mu_T)s_{ij}, \tag{42}$$

where μ is the dynamic viscosity, μ_T is the turbulent viscosity, and s_{ij} is the rate of strain tensor, defined as

$$s_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}. \tag{43}$$

The heat flux vector is obtained with Fourier's Law,

$$q_j = -C_p\left(\frac{\mu}{Pr} + \frac{\mu_T}{Pr_T}\right)\frac{\partial T}{\partial x_j}, \tag{44}$$

where C_p is the specific heat, Pr is the Prandtl number, Pr_T is the turbulent Prandtl number, and T is the temperature. The ideal gas equation of state, $p = \rho RT$ is used to close the equations.

IV.A. Discretization and Solution Methods

The strand grid spatial discretization is based on a cell-centered approach where the primary unknowns are located at the centroid of the prisms formed by adjacent strands. The solver accommodates both quadrilateral and triangular prisms depending on the surface topology. However, control volumes are composed entirely of triangular facets by triangulating any non-planar quadrilateral faces. This is important for second-order accuracy on general prismatic grids with no assumption of underlying smoothness.[19] Linear reconstruction is employed to obtain second-order accuracy through first obtaining consistent nodal values of the conserved variables from surrounding cell-center values. A projection method is used to obtain these nodal values via least squares interpolation in a regression plane through the three-dimensional stencil of cells surrounding a strand.[19] Once the nodal values have been obtained, a Green-Gauss surface integration procedure is performed to obtain cell gradients in each control volume.

Inviscid fluxes rely on a reconstruction upwind formula for the numerical flux based on the approximate Riemann solver of Roe,[20]

$$\hat{\mathcal{F}} = \frac{1}{2}\left(\mathcal{F}(Q_R) + \mathcal{F}(Q_L)\right) - \frac{1}{2}\left|A(Q_R, Q_L)\right|\left(Q_R - Q_L\right), \tag{45}$$

where $\mathcal{F} = F_j n_j$ is the directed flux at a face with normal n_j, and $A = \partial\mathcal{F}/\partial Q$ is the directed flux Jacobian. The viscous terms are computed using values of Q and ∇Q determined at each face,

$$\mathcal{F}^v = \mathcal{F}^v(Q_f, \nabla Q_f), \tag{46}$$

where f refers to the face reconstructed values. These face values are easily obtained once nodal values have been reconstructed using the projection method described above. This method is similar to the node averaging schemes outlined by Diskin, et al.[21] Both the inviscid and viscous discretization methods described herein have been verified to be second-order accurate for arbitrary prismatic meshes under a variety of conditions using the method of manufactured solutions.[19]

The result of the spatial discretization of the viscous and inviscid fluxes is a coupled set of non-linear equations. In this work, we adopt a pseudo-time framework to march the steady or unsteady discretized equations to steady-state,

$$V\frac{\partial Q}{\partial \tau_k} + R(Q) = 0. \tag{47}$$

Here, V is the cell volume, and τ_k is the pseudo-time variable. The residual, $R(Q)$, contains the inviscid and viscous flux balances at each cell based on the cell-center discretization schemes described above. In order to reach a pseudo-steady state using an implicit scheme, the residual is linearized, leading to the following linear system to be solved at each pseudo-time step:

$$\left[\frac{V}{\Delta\tau_k}I + \frac{\partial R^k}{\partial Q}\right]\left(Q^{k+1} - Q^k\right) = -R(Q^k). \tag{48}$$

Here, $\partial R^k/\partial Q$ is the Jacobian of the residual. The linear system in Equation 47 in general is large and sparse, rendering direct inversion impractical. Iterative line Gauss-Seidel (GS) methods are employed to solve this system, where contributions along strands are collected to form a tridiagonal system. To facilitate the line GS iterations and to increase robustness, we introduce an additional "linear time" variable, τ_l,

$$V\frac{\partial Q}{\partial \tau_l} + \left[\frac{V}{\Delta\tau_k}I + \frac{\partial R^k}{\partial Q}\right]\left(Q^{k+1} - Q^k\right) = -R(Q^k). \tag{49}$$

The linear time is introduced to improve the diagonal dominance of the line GS procedure in order to increase robustness. Rearranging Equation 49 in terms of solution updates in linear time results in

$$\left[\left(\frac{1}{\Delta\tau_k} + \frac{1}{\Delta\tau_l}\right)VI + \frac{\partial R^k}{\partial Q}\right]\left(Q^{l+1} - Q^l\right) = -R(Q^k) - \left[\frac{V}{\Delta\tau_k}I + \frac{\partial R^k}{\partial Q}\right]\left(Q^l - Q^k\right). \tag{50}$$

Upon convergence of the linear iterations in l, the linear system of Equation 48 is satisfied. At that point, the next pseudo-time step in k proceeds. When the pseudo-time iterations converge, then the residual equation $R(Q)$ is satisfied for a given physical time station. All Jacobian terms in this work are first order and retain only nearest neighbor contributions. Further details of the implicit solution method may be found in previous work.[16]

IV.B. Turbulence Model

The standard SA model is used when the turbulent working variable is positive. Details of the positive model, including the well-known definitions of the production and destruction terms, may be found in the original work by Spalart and Allmaras.[18] Modifications to the model to accommodate negative values of the turbulence working variable have been suggested recently by Allmaras[22] and are employed in this work. In the case of negative values of $\tilde{\nu}$, the following turbulence equation replaces the standard model:

$$\frac{\partial\tilde{\nu}}{\partial t} + u_j\frac{\partial\tilde{\nu}}{\partial x_j} = C_{b1}(1 - C_{t3})\Omega\tilde{\nu} + C_{w1}\left(\frac{\tilde{\nu}}{d}\right)^2 + \frac{1}{\sigma}\left[\frac{\partial}{\partial x_j}\left((\nu + \tilde{\nu}f_n)\frac{\partial\tilde{\nu}}{\partial x_j}\right) + C_{b2}\frac{\partial\tilde{\nu}}{\partial x_k}\frac{\partial\tilde{\nu}}{\partial x_k}\right], \tag{51}$$

where,

$$f_n = \frac{C_{n1} + \chi^3}{C_{n1} - \chi^3}, \quad (C_{n1} = 16).$$

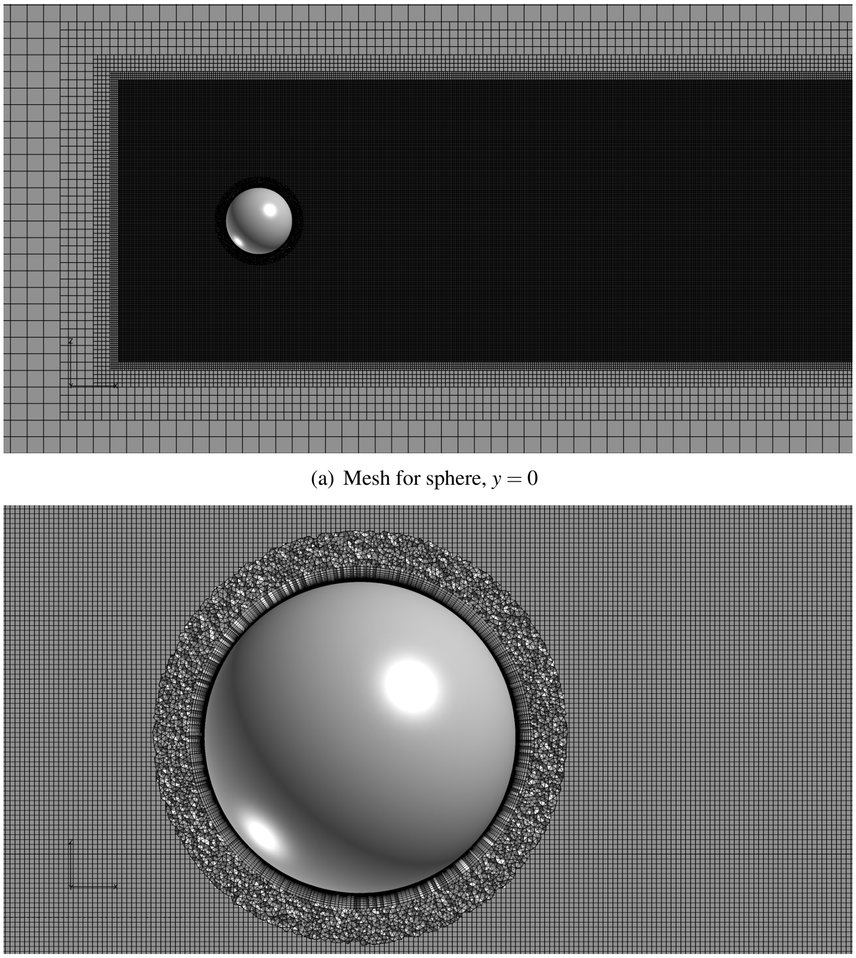
Here, Ω is the vorticity magnitude, d is the distance to the nearest wall, and $\chi = \tilde{\nu}/\nu$ is the ratio of the turbulent working variable to the kinematic viscosity of the fluid. All other constants in Equation 51 take the values found in the standard model.

V. Results

V.A. Helios Validation

V.A.1. Flow over a sphere

The first application example is the flow over a sphere at high reynolds number. This is a canonical fluid mechanics problem and provides a suitable initial test of the turbulence model in SAMARC. The flow conditions for this case are free-stream Mach number $M_\infty = .3$, angle of attack $\alpha = 0.0$, side slip angle $\beta = 0.0$ and a Reynolds number based on sphere diameter of $Re = 6.76e^6$. The near-body mesh contains $1,440,699$ grid points and the off-body mesh has $139,485,538$ grid points. A slice of the mesh at the $y = 0$ plane is shown in Figure 8(a). This test case is time-accurate with a time-step of $7.0e^{-7}$ seconds, a total of $50,000$ time-steps are simulated.

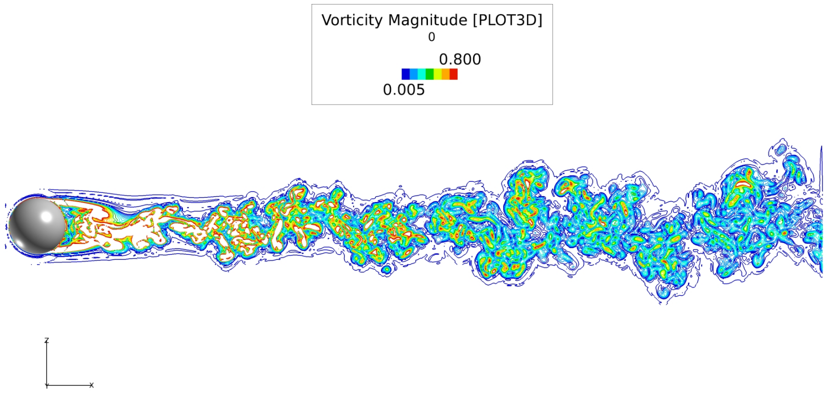


(a) Mesh for sphere, $y = 0$

(b) Close-up of near-body mesh , $y = 0$ plane

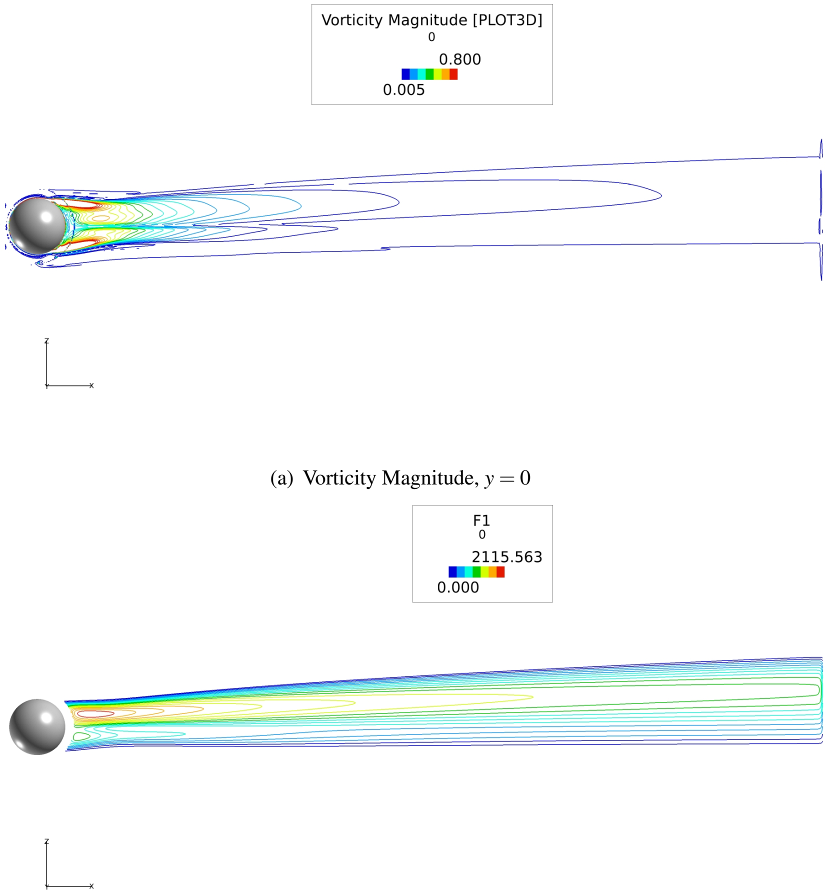
Figure 8. Mesh used for the computation of the flow over a sphere.

This work is focused on evaluating the various physical modeling options in SAMARC. Therefore the physical model used in the near-body mesh is always viscous and employs either the SA-RANS or SA-DDES turbulence models. The overset dual-mesh capability of Helios allows for various combinations of physical models in the near-body (NB) and off-body (OB) meshes. The following combinations of physical models are employed for this test case: SA-RANS in the near-body with inviscid off-body, SA-RANS near-body with SA-RANS off-body, and SA-DDES near-body with SA-DES off-body. Figure 9(a) depicts the vorticity contours at the final time-step on the



(a) Vorticity Contours , $y = 0$ plane

Figure 9. Vorticity for the flow over a sphere with an inviscid solver in the off-body region.



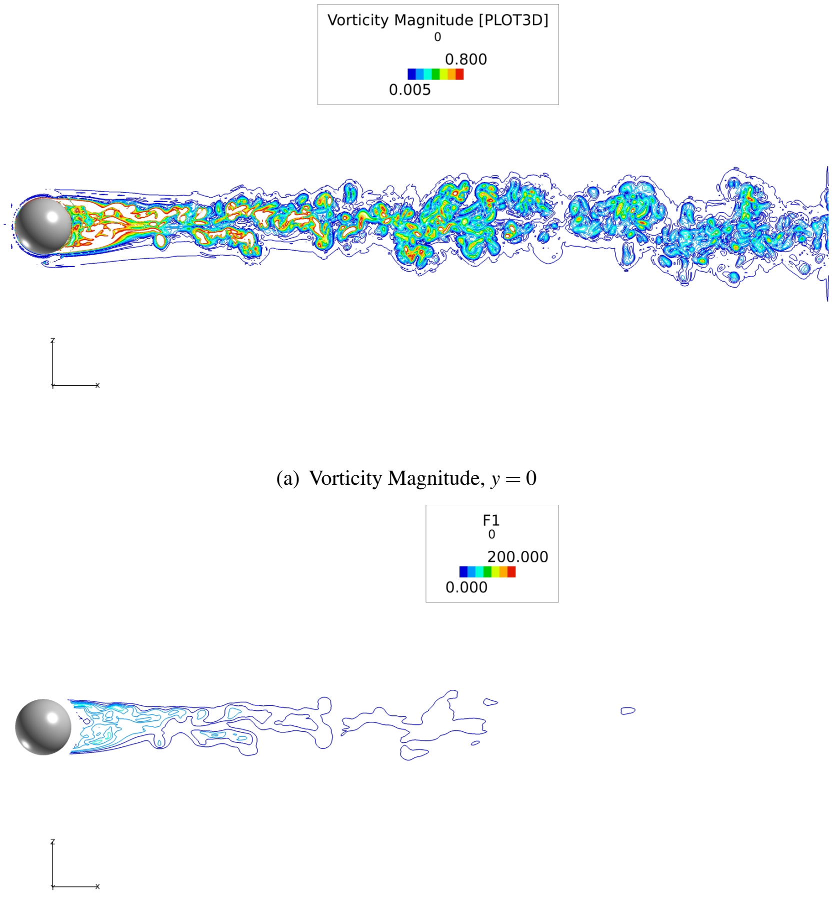
(a) Vorticity Magnitude, $y = 0$

(b) Eddy Viscosity, $y = 0$ plane

Figure 10. Vorticity for the flow over a sphere with a SA-RANS solver in the off-body region.

$y = 0$ plane for using the inviscid off-body and RANS near-body physical model combination. The inviscid version of the off-body solver generates a highly unsteady wake solution, which represents the base-line Helios solution for this problem.

Figure 10(a) shows the vorticity contours on the $y = 0$ plane using SA-RANS near-body and SA-RANS off-body combination. Employing the SA-RANS turbulence model in the off-body region has changed the wake structure from a length scale rich and obviously unsteady wake to one that more closely resembles that of a steady wake where all scales have been modeled. Figure 10(b) shows the turbulent eddy viscosity for this case, which is highest near the body and quickly decreases in the downstream region. The eddy viscosity distribution confirms that extensive turbulence modeling is occurring in this region.



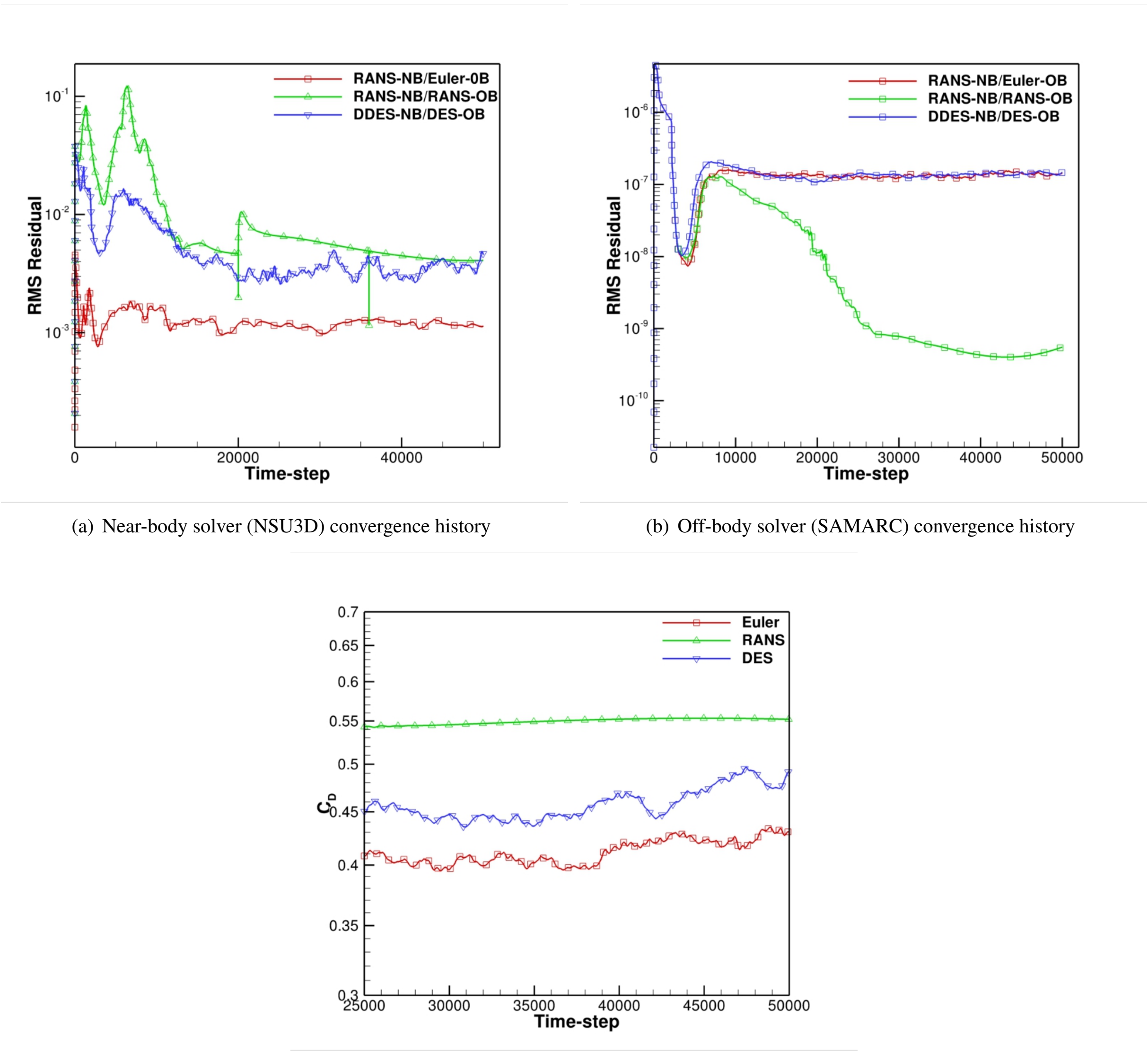
(a) Vorticity Magnitude, $y = 0$

(b) Eddy Viscosity, $y = 0$ plane

Figure 11. Vorticity for the flow over a sphere with a SA-DES turbulence model in the off-body region and a DDES turbulence model in the near-body region.

Figure 11(a) depicts the vorticity contours on the $y = 0$ plane using SA-DDES in the near-body mesh and SA-DES in the off-body mesh. Employing SA-DES, which is partially a large eddy simulation (LES) method, in the off-body region results in a wake which is again unsteady and contains a wide variety of captured length scales. Examination of Figure 11(b), which depicts the eddy viscosity using SA-DES in the off-body, confirms that the turbulence model is active in the wake region. Comparing Figure 9(a) to Figure 11(a) qualitatively demonstrates that using an inviscid solver in the off-body region is appropriate provided that there is sufficient grid resolution to resolve at least some of the turbulent structures. However, despite the low values of eddy viscosity in the wake region obtained using SA-DES there are still some noticeable qualitative difference between using SA-DES and inviscid off-body solvers. In particular the wake computed using the SA-DES turbulence model shows fewer small scale turbulent structures as well as a narrower overall wake.

The convergence histories of the near- and off-body solvers are plotted in figures Figure 12(a) and Figure 12(b) respectively. These figures show that solution transients are decayed after $25,000$ time-steps for both the near-body and off-body solvers. Figure 12(c) depicts the computed drag coefficient over the last $25,000$ time-steps of the time



(a) Near-body solver (NSU3D) convergence history

(b) Off-body solver (SAMARC) convergence history

(c) Near-body solver (NSU3D) C_D convergence

Figure 12. Vorticity for the flow over a sphere with a SA-DES turbulence model in the off-body region and a DDES turbulence model in the near-body region.

history, demonstrating that the last $25,000$ time-steps are suitable for obtaining an average drag coefficient value. Therefore, the following average values of drag are computed by sampling the last $25,000$ time-steps of the solver.

Table 1. Computed drag coefficients for the flow over a sphere with flow conditions $M_\infty = .3$, $\alpha = 0^o$, and $Re = 6,760,000$

	Euler-OB	SA-RANS-OB	SA-DES-OB
C_D	.411	.550	.457

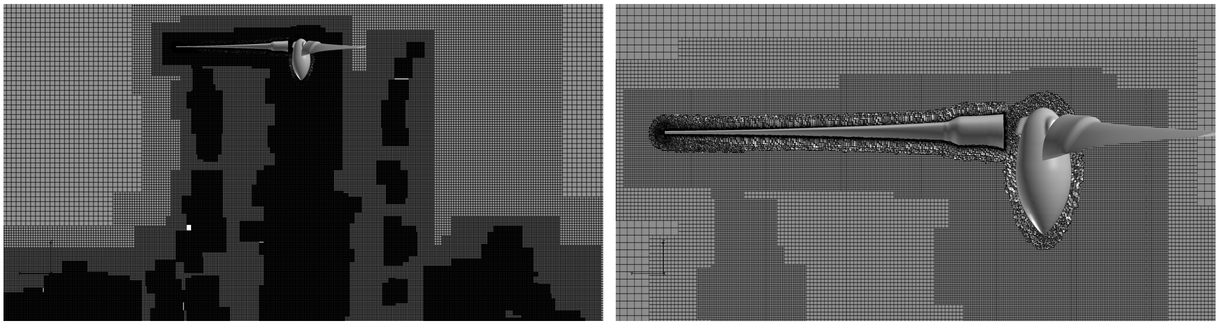
Table 1 contains the average computed drag coefficients for the flow over a sphere. The computed drag coefficient obtained utilizing SA-RANS in the off-body mesh is significantly higher compared with those computed using either of the inviscid or the SA-DES solvers in the off-body mesh. Additionally, it is interesting to note that using SA-DES in the off-body mesh causes the drag coefficient to increase by 11.2% over the drag coefficient computed using the inviscid solver in the off-body mesh.

This test case has demonstrated that the effects of including a turbulence model in the off-body mesh can be significant to both qualitative flow features as well as relevant simulation outputs such as drag coefficient.

V.A.2. Tilt Rotor Aeroacoustics Model (TRAM) Rotor

Helios is primarily used for rotorcraft aeromechanics simulations. In order to examine the effects of including turbulence modeling in SAMARC on a revenant rotorcraft aeromechanics problem, this work considers flow though the Tilt Rotor Aeroacoustics Model (TRAM) rotor. This case generates strong tip-vortices that must be adequately resolved and not artificially dissipated by the off-body solver. Similarly to the flow over a sphere, three models for turbulence in the off-body CFD solver are compared: inviscid, SA-RANS, and SA-DES. Qualitative analysis of the wake structures is discussed to examine where in the domain and how the SA-RANS and SA-DES turbulence models are altering flow physics . Additionally, effects of various off-body turbulence models on the figure of merit (FM) are examined as a quantitative basis of comparison.

The flow conditions of the TRAM rotor are hover conditions with a tip Mach number $M_{tip} = 0.625$ and collective pitch $\theta^0 = 14^o$ on the rotor blades. The tip Reynolds number for this case is $Re_{tip} = 2.1e^6$. The time-step is set such that the rotor rotates $.25^o$ degrees per time-step. The near-body mesh used to compute this flow contains $9,273,348$ grid points. For this test case off-body AMR is employed every 10 time-steps after one revolution of the rotor has been completed. For all test cases the off-body solver is initialized with a uniform cartesian mesh that is $16 \times 16 \times 24$. The off-body mesh is then adapted to the geometry and flow physics with up to nine levels of isotropic mesh refinement. Examples of the types of meshes used in these computations are given in Figure 13(a) and Figure 13(b). Using identical inputs three variants of this test case are considered: near-body SA-RANS with off-body inviscid, near-body SA-RANS with off-body SA-RANS, and near-body SA-DDES with off-body SA-DES.

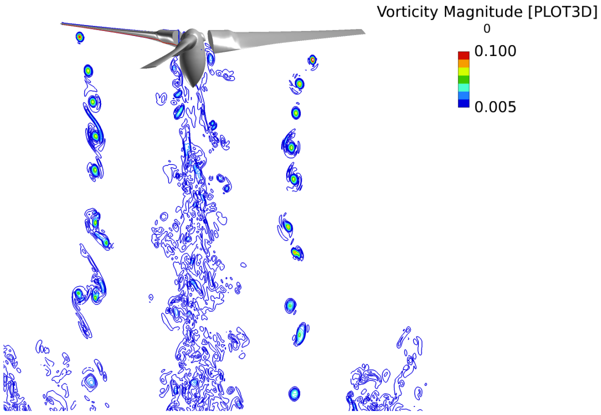

(a) Computational Mesh at $x = 0$ plane

(b) Computational Mesh at $x = 0$ plane

Figure 13. Flow through the TRAM rotor in hover, using the SA turbulence model in the off-body mesh. Vorticity depicted after 8 revolutions of the rotor and using adaptive mesh refinement in the off-body region

.

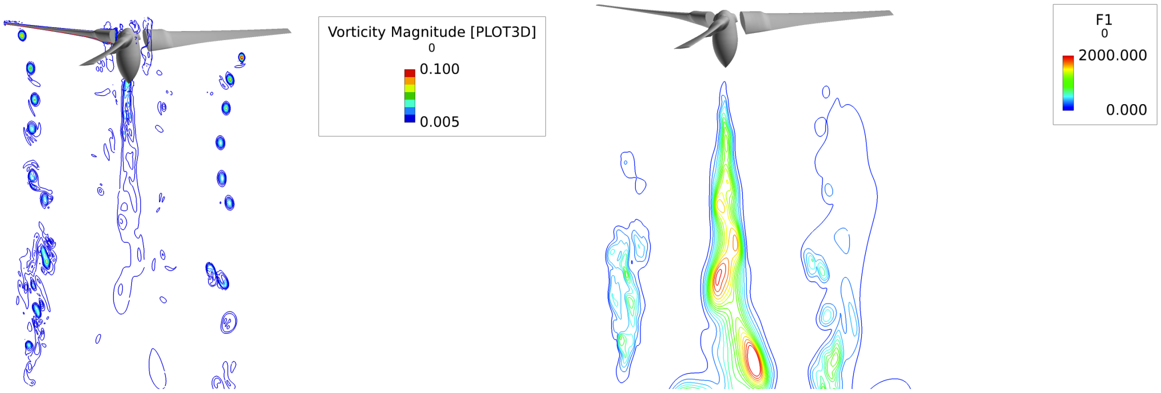
Figure 14(a) depicts the vorticity magnitude contours on the $x = 0$ plane, generated by using an inviscid solver in the off-body region. This figure shows that the tip vortices are clearly resolved and maintained by using an inviscid solver in the off-body region. Furthermore, there is little evidence of strong multi-scale flow structures near the tip vortices. However, the wake generated by the inboard section of the rotor blades as well as by the hub display strong multi-scale flow physics and qualitatively resemble turbulence. Figure 15(a) depicts the vorticity magnitude contours on the $x = 0$ plane, generated using the SA-RANS turbulence model in the off-body region. One should immediate notice from Figure 15(a) that the tip vortices are also clearly resolved despite the use of a SA-RANS turbulence model. In fact comparing Figure 14(a) to Figure 15(a) shows the the tip vortex structures are sufficiently captured using either off-body model. Figure 15(b), which depicts the eddy viscosity for this case, shows why the tip vortices are not dissipated by solver as the eddy viscosity is low near the tip vortices.. However, examination of the wake emanating from the hub shows that the eddy viscosity if relatively large in this region and therefore the wake structure is significantly different from the wake structure shown in Figure 14(a). The hub wake structure in Figure 15(a) contains almost no small scale flow structures as the turbulence model is modeling all length scales as can be observed in Figure 15(b). Figure 16(a) depicts the vorticity magnitude contours on the $x = 0$ plane, generated using the SA-DES turbulence model in the off-body region. Similarly to the previous two results, Figure 15(a) shows that the tip vortices are clearly resolved and comparable to the inviscid flow solution in the off-body mesh. The SA-DES model is not dissipating the vortices because as with the SA-RANS model, the eddy viscosity is small near the tip vortices as shown in Figure 16(b). As with the previous two off-body flow solutions the predominant difference between the SA-DES, SA-RANS, and inviscid off-body solutions is the structure of the wake emanating from the hub. Examining Figure 16(a) and Figure 16(b) shows that the modeling of turbulence is relatively active in the hub wake when using SA-DES in the off-body solution. Furthermore, comparing Figure 14(a), Figure 15(a) and Figure 16(a) shows that



(a) Vorticity Contours at $x = 0$ plane

Figure 14. Vorticity magnitude contours for flow through the TRAM rotor in hover, with an inviscid solver in the off-body mesh after 8 revolutions of the rotor and using adaptive mesh refinement in the off-body region

.

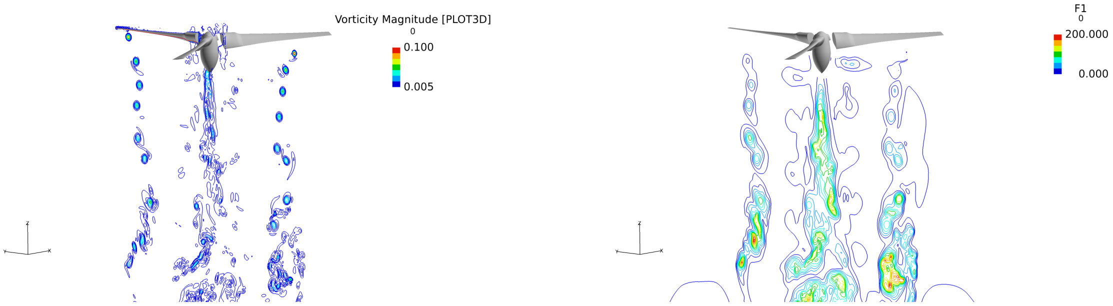


(a) Vorticity Contours at $x = 0$ plane

(b) μ_T at $x = 0$ plane

Figure 15. Voritcity and eddy viscosity contours on the $x = 0$ plane for flow through the TRAM rotor in hover, using the SA turbulence model in the off-body mesh. Solution is depicted after 8 revolutions of the rotor and using adaptive mesh refinement in the off-body region

.



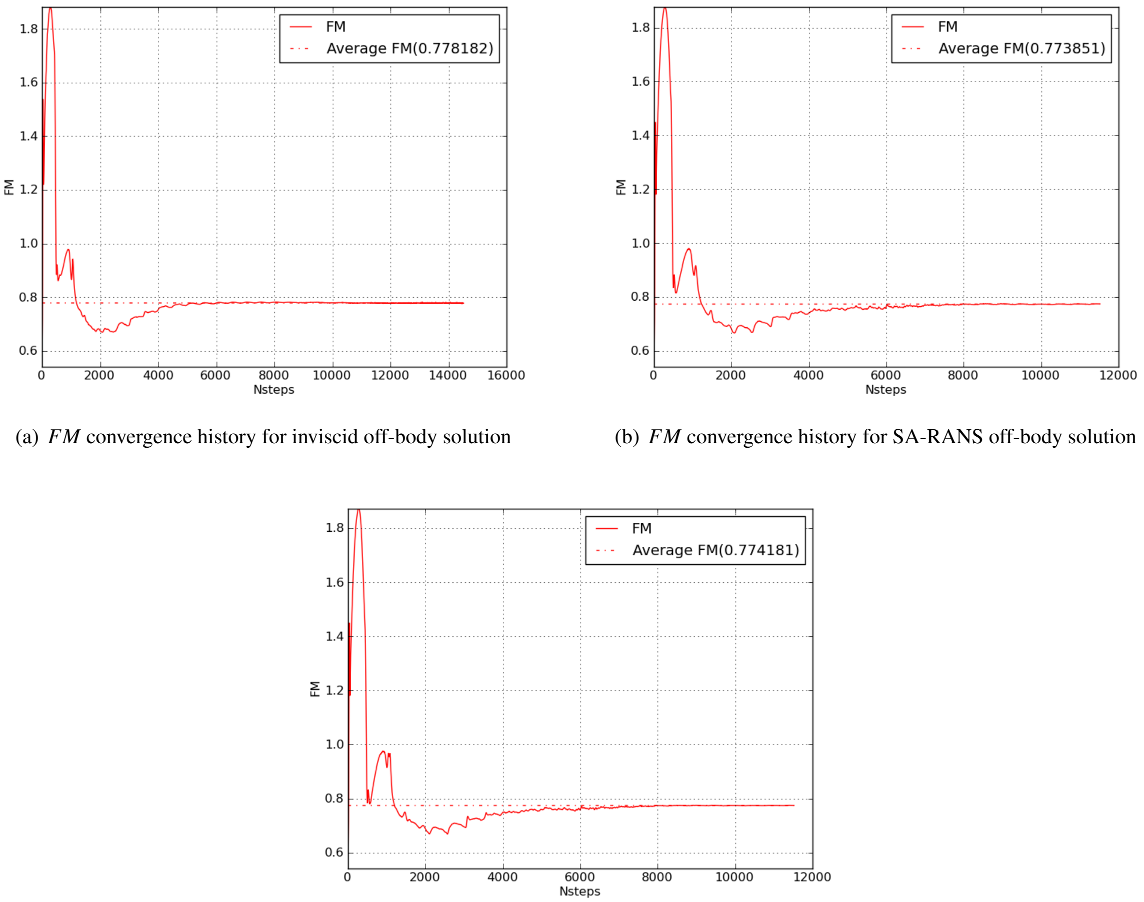
(a) Vorticity Contours at $x = 0$ plane

(b) μ_T at $x = 0$ plane

Figure 16. Flow through the TRAM rotor in hover, using the SA-DES turbulence model in the off-body mesh. Vorticity depicted after 8 revolutions of the rotor and using adaptive mesh refinement in the off-body region

.

using SA-DES in the off-body solver results in a hub wake that is somewhere between the fully modeled SA-RANS solution and the captured only inviscid solution, which is the intended and expected result.



(a) FM convergence history for inviscid off-body solution

(b) FM convergence history for SA-RANS off-body solution

(c) FM convergence history for RANS-DES off-body solution

Figure 17. Figure of merit (FM) convergence history for the flow through the TRAM rotor using inviscid, SA-RANS, and SA-DES in the off-body region.

Finally, the computed figure of merit (FM) obtained using all three off-body solution options is compared. Figure 17(a) through Figure 17(c) shows the convergence histories of the FM's and indicate that the figure of merit has become converged after 8 revolutions of the rotor are completed. Table 2 summarizes the average FM computed using the three off-body solutions as well as the experimental value. Table 2 shows that at the spatial and temporal resolution considered in this work the computed figures of merit are approximately equivalent as the differences between all three off-body solver options are very small ($O(.5\%)$). Therefore, this work has validated the assumption that an inviscid off-body solver is an appropriate option for engineering type calculations.

Table 2. Computed and expire mental FM's for the flow through the TRAM rotor.

	Invsicd	SA-RANS-OB	SA-DES-OB	Expriement
FM	.778	.773	.774	.779

While the addition of turbulence modeling to the SAMARC off-body solver has not improved the computed FM, it is still a valuable addition SAMARC. In particular has some sort of sub grid scale model is beneficial for examine various flow features that can occur in complex rotorcraft configurations.

V.B. Strand Turbulence Model Validation

The S-A implementation in the strand solver is validated for flow over a flat plate at $M = 0.2$ and $Re = 5 \times 10^6$, based on a plate of length unity. This case was taken from the NASA Langley turbulence modeling resource has been made for this case.[23]

The grid used for the flat plate validation is a 137×97 grid shown in Figure 18(a). The plate leading edge begins at $x = 0$ and extends for a length of two. A short inviscid wall entry way beginning at $x = -0.33$ is provided to allow for proper inflow conditions. Stagnation temperature and pressure are specified at the inflow, and static pressure is specified at the outflow. The turbulent viscosity field for this case is shown in Figure 18(b), which has been scaled by a factor of 40 vertically to facilitate visualization. Streamwise velocity and turbulent viscosity profiles are shown in Figure 19(a) and 19(b) for two locations downstream on the plate, and are over plotted with FUN3D and CFL3D results. Note that good agreement is obtained, even for this 137×97 grid which is 16 times more coarse than the FUN3D and CFL3D results in the figures. The computed drag coefficient, which is entirely due to skin friction for this case, is shown in Table 3, along with FUN3D and CFL3D results for the same grid. The drag coefficient falls within the range predicted by the established codes.

The next validation case is 2D flow over a NACA 0012 airfoil at $M = 0.15$ and $Re = 6 \times 10^6$ at various angles of attack. The strand grid shown in Figure 20 consists of a smoothed NACA 0012 airfoil containing 320 surface nodes and 64 cells along each strand, yielding 20,480 total strand cells. Figures 21 and 22 show lift coefficient versus angle of attack and drag coefficient versus lift coefficient, respectively, along with the corresponding experimental data of Ladson.[24] The lift data is matched reasonably well for all angles of attack, although a slight over-prediction of lift is observed for the high-α $\alpha = 15^o$ case. Drag results are shown in Figure 22 with the $\alpha = 15^o$ results summarized in Table 4. FUN3D and CFL3D results using a much finer grid (513 surface nodes instead of 320) are included for comparison. In general the strand solver results fall well within the range of these well established codes.

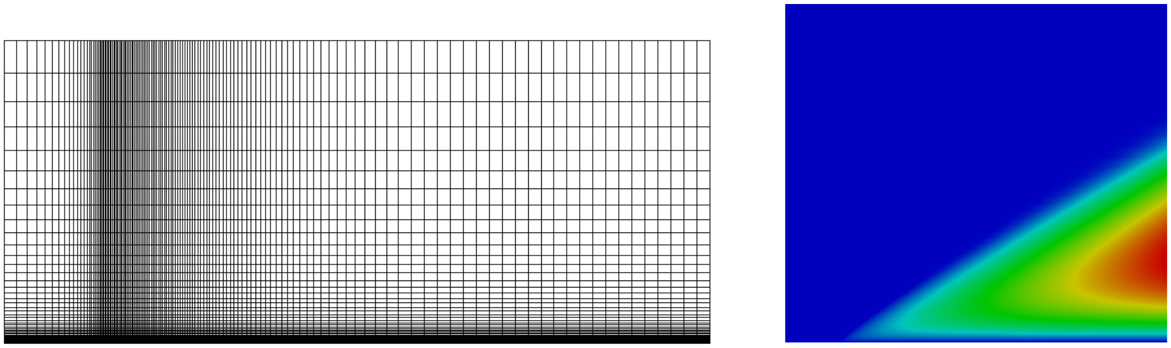
VI. Conclusions & Final Paper

This paper will present a summary of turbulent flow validation in Helios v3. It also includes a validation with the new strand solver that is eventually intended to become the new production solver in Helios.

Sample calculations are shown in this abstract, the final paper will present results from applications as defined by the AIAA Fluid Dynamics Technical Committee.

VII. Acknowledgments

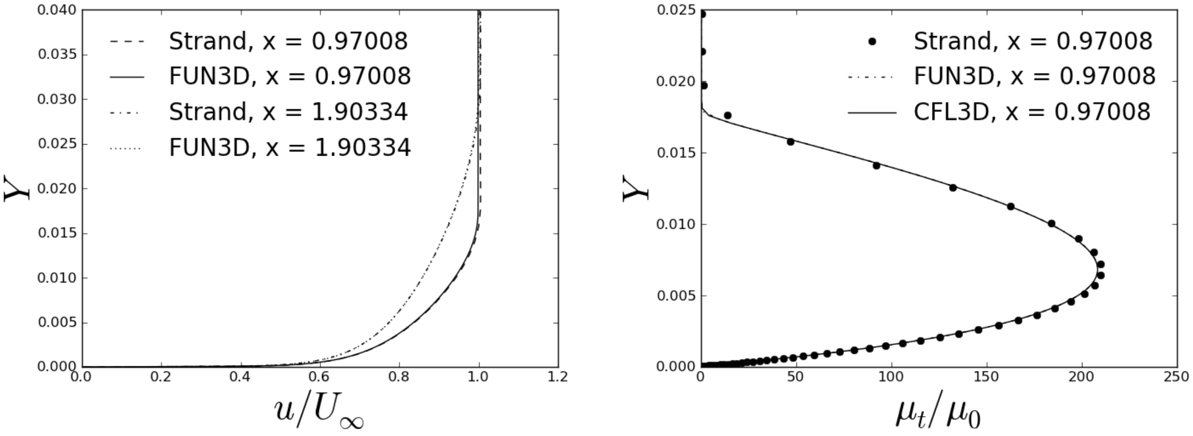
Material presented in this paper is a product of the CREATE-AV (Air Vehicles) Element of the Computational Research and Engineering for Acquisition Tools and Environments (CREATE) Program sponsored by the U.S. Department of Defense HPC Modernization Program Office. This work was conducted at the High Performance Computing Institute for Advanced Rotorcraft Modeling and Simulation (HIARMS). The authors gratefully acknowledge the contributions by Dr. Venkateswaran Sankaran of the U.S. Air Force Research Lab (AFRL) at Edwards Air Force Base, and by Dr. Hossein Saberi and Dr. Wei-Bin Chen of Advanced Rotorcraft Technology, Inc. Compute resources were provided by the DoD High Performance Computing and Modernization Office (HPCMO) DOD Shared Resource Centers (DSRC).



(a) 137x97 grid

(b) Contours of μ_t/μ_0

Figure 18. Grid and turbulent viscosity contours for flow over a flat plate at $M = 0.2$ and $Re = 5 \times 10^6$.



(a) x-velocity profiles

(b) turbulent viscosity profile

Figure 19. Comparison of streamwise velocity and turbulent viscosity profiles for flow over a flat plate at $M = 0.2$ and $Re = 5 \times 10^6$.

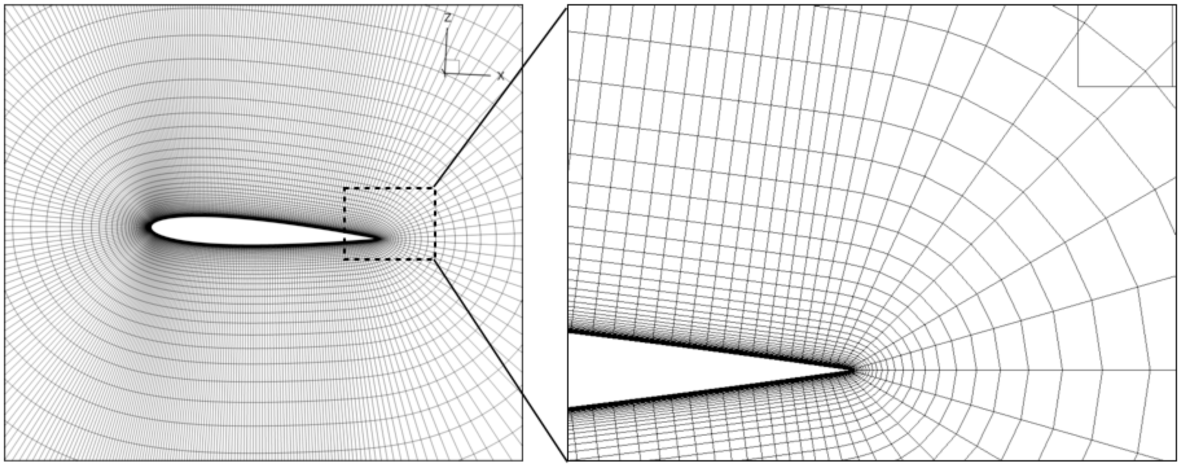


Figure 20. 137×97 **NACA 0012 strand grid**

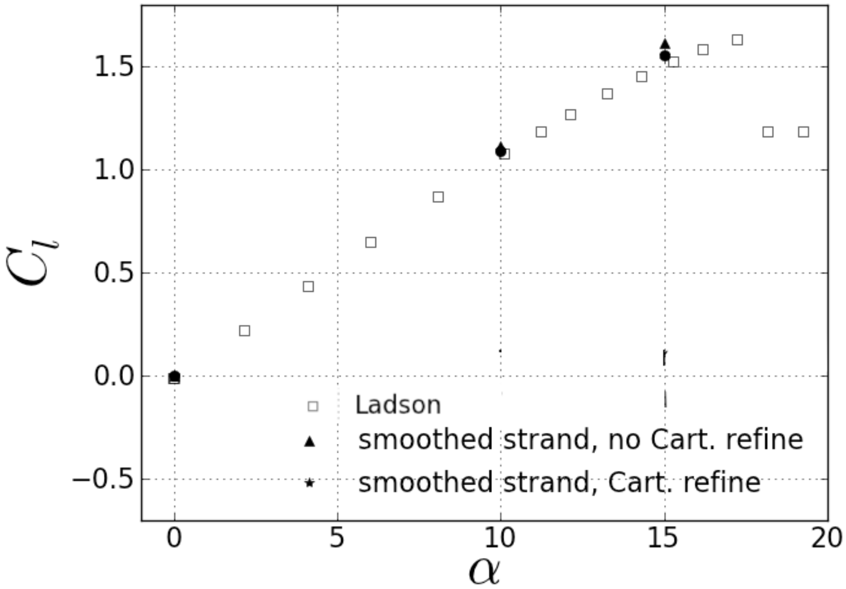


Figure 21. C_l **vs.** α **compared to experiment for flow over NACA 0012 airfoil at** $M = 0.15$ **and** $Re = 6 \times 10^6$**.**

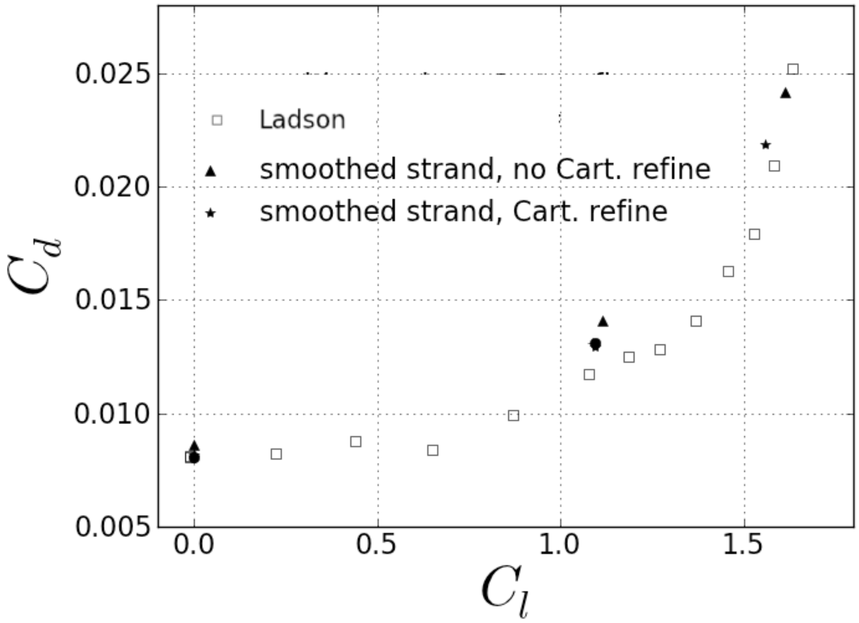


Figure 22. C_l **vs.** C_d **compared to experiment for flow over NACA 0012 airfoil at** $M = 0.15$ **and** $Re = 6 \times 10^6$**.**

References

[1] Strawn, R.C., F.X. Caradonna, E.P.N. Duque, “30 years of Rotorcraft Computational Fluid Dynamics Research and Development,” *Journal of the American Helicopter Society*, Vol. 51, (1), Jan 2006, pp. 5–21.

[2] Datta, A., and W. Johnson, “An Assessment of the State-of-the-art in Multidisciplinary Aero-mechanical Analysis,” American Helicopter Society Technical Specialists Meeting, San Francisco, January 22-24, 2008.

[3] Post, D.E., “A new DoD initiative: the Computational Research and Engineering Acquisition Tools and Environments (CREATE) program,” *Journal of Physics*, Conference Series 125, 2008.

[4] Wissink, A.M., B. Jayaraman, A. Datta, J. Sitaraman, M. Potsdam, S. Kamkar, D. Mavriplis, Z. Yang, R. Jain, J. Lim, R. Strawn, “Capability Enhancements in Version 3 of the Helios High-Fidelity Rotorcraft Simulation Code,” AIAA-2012-0713, 50th AIAA Aerospace Sciences Meeting, Nashville TN, Jan 2012.

[5] Mavriplis, D. J., and V. Venkatakrishnan, “A Unified Multigrid Solver for the Navier-Stokes Equations on Mixed Element Meshes,” *International Journal for Computational Fluid Dynamics*, Vol. 8, 1997, pp. 247-263.

[6] Burgess, N., and A. Wissink, “Effects of Turbulence Modeling for a Dual Mesh CFD Solver,” 21st AIAA Computational Fluid Dynamics Conference, June 2013,San Diego, CA.

[7] Wissink, A.M, N.K. Burgess, A.J. Katz, J. Sitaraman, and R. Haimes, “Progress in Automatic Viscous Mesh Generation From CAD Using Strand/Cartesian Meshes,” 21st AIAA Computational Fluid Dynamics Conference, June 2013, San Diego, CA.

[8] Meakin, R., A. M. Wissink, W. M. Chan, S. A. Pandya, and J. Sitaraman, “On Strand Grids for Complex Flows,” AIAA-2007-3834, 18th AIAA Computational Fluid Dynamics Conference, Miami FL, June 2007.

[9] S Spalart, P. R., W.H. Jou, M. Stretlets, and S.R. Allmaras, ”Comments on the Feasibility of LES for Wings and on the Hybrid RANS/LES Approach”, Proceedings of the First AFOSR International Conference on DNS/LES Aug 4-8, 1997, Ruston, LA. In: Advances in DNS/LES, C. Liu and Z. Liu Eds., Greyden Press, Columbus, OH, USA (1997).

[10] D. J. Mavriplis, J. Pelaez and O. Kandil, “Large Eddy and Detached Eddy Simulations using an Unstructured Multigrid Solver,” Proc. of the 3rd AFOSR Int. Conf. on DNS/LES, Arlington, TX, Aug 2001, C. L. Liu, L. Sakell, and T. Beutner Eds (2001).

[11] Spalart, P. R., S. Deck, M.L. Shur, K.D. Squires, M. Strelets, and A. Travin, ”A New Version of Detached-Eddy Simulation, Resistant to Ambiguous Grid Densities,” Theoretical and Computational Fluid Dynamics, Vol. 20, No. 3, pp.181-195., 2006.

[12] Sheldahl, R. E., and P.C. Klimas,”Aerodynamic Characteristics of Seven Airfoil Sections Through 180 Degrees Angle of Attack for Use in Aerodynamic Analysis of Vertical Axis Wind Turbines,” SAND80-2114, Sandia National Laboratories, Albuquerque NM, Mar 1981.

[13] Wissink, A.M., A.J. Katz, W.M. Chan, R.L. Meakin, “Validation of the Strand Grid Approach,” AIAA-2009-3792, 19th AIAA Computational Fluid Dynamics Conference, San Antonio TX, June 2009.

[14] Katz, A., A.M. Wissink, V. Sankaran, R.L. Meakin, W.M. Chan, “Application of Strand Meshes to Complex Aerodynamic Flowfields,” *Journal of Computational Physics*, Vol. 230, No. 17, July 2011.

[15] Katz, A., A. Wissink, V. Sankaran, “Convergence Acceleration Techniques for Coupled Adaptive Cartesian-Strand Grid Solutions,” AIAA-2011-772, 49th AIAA Aerospace Sciences Meeting, Orlando FL, Jan 2011.

[16] Katz, A.J., A.M. Wissink, “Efficient Solution Methods for Strand Grid Applications,” 30th AIAA Applied Aerodynamics Conference, New Orleans LA, June 2012.

[17] Wissink, A.M., B. Jayaraman, A. Datta, J. Sitaraman, M. Potsdam, S. Kamkar, D. Mavriplis, Z. Yang, R. Jain, J. Lim, R. Strawn, “Capability Enhancements in Version 3 of the Helios High-Fidelity Rotorcraft Simulation Code,” AIAA-2012-0713, 50th AIAA Aerospace Sciences Meeting, Nashville TN, Jan 2012.

[18] Spalart, P.R., and S.R. Allmaras, “A one-equation turbulence model for Aerodynamic flows,” *Le Recherche Aerospátiale*, Vol. 1, pp. 5–21, 1994.

[19] Katz, A., and V. Sankaran, “Discretization Methodology for High Aspect Ratio Prismatic Grids,” AIAA-2011-3378, 20th AIAA Computational Fluid Dynamics Conference, Honolulu HI, June 2011.

[20] Roe, P.L., “Approximate Riemann Solvers, Parameter vectors, and Difference Schemes,” *J. Comput. Phys.*, Vol. 43, pp. 357–372, 1981.

[21] Diskin, B., J. Thomas, E. Nielsen, and H. Nishikawa, “Comparison of Node-Centered and Cell-Centered Unstructured Finite-Volume Discretizations. Part 1: Viscous Fluxes,” AIAA-2009-0597, 47th AIAA Aerospace Sciences Meeting, Orlando FL, Jan 2009.

[22] Allmaras, S.R., Forrester, J.T., and Spalart, P.R. ”Modifications and Clarifications for the Implementation of the Spalart-Allmaras Turbulence Model”, ICCFD-1902, 7th International Conference on Computational Fluid Dynamics (ICCFD7), Big Island, Hawaii, July 2012.

[23] National Aeronautics and Space Adiminstration. ”Turbulence Modeling Resource”, NASA Langley, 2012. http://turbmodels.larc.nasa.gov/.

[24] Ladson, C., “Effects of Independent Variation of Mach and Reynolds Numbers on the Low-Speed Aerodynamic Characteristics of the NACA 0012 Airfoil Section,” *NASA TM 4074*, Oct 1988.

[25] Karypis, G., and V. Kumar, “A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs,” *SIAM Journal on Scientific Computing*, Vol. 20, No. 1, pp. 359–392, 1999.

[26] Sitaraman, J., B. Roget, and A. Wissink, “OSCAR - An Overset Grid Assembler for Overlapping Strand/Cartesian Mesh Systems,” 11th Symposium on Overset Composite Grids and Solution Technology, Dayton OH, Oct 2012. http://2012.oversetgridsymposium.org/.

	C_d
Strand	2.82287E-3
FUN3D (quads)	2.84005E-3
FUN3D (triangles)	2.80289E-3
CFL3D	2.86621E-3

Table 3. Comparison of computed drag coefficients for flow over a flat plate at $M = 0.2$ and $Re = 5 \times 10^6$.

[27] Burgess, N. K , ”An Adaptive Discontinuous Galerkin Solver for Aerodynamic Flows”, Ph.D. thesis, University of Wyoming, Novmeber 2011.

[28] Burgess, N.K. and Mavriplis, D.J. ”Robust Computation of Turbulent flows using a Discontinuous Galerkin Method”, AIAA-2012-457, 50th AIAA Aerospace Scieneces Meeting, Nashville TN, Jan 2012.

[29] Burgess, N.K. and Mavriplis, D.J. ”High-order Discontinuous Galerkin Methods for Turbulent High-lift Flows”, ICCFD-4202, 7th International Conference on Computational Fluid Dynamics (ICCFD7), Big Island, Hawaii, July 2012.

[30] Johnson, F. T., Kamenetsky, D. S., Melvin, R. G., Venkatakrishnan, V., Wigton, L. B., Young, D. P., Allmaras, S. R., Bussoletti, J. E., Hilmes, C. L., “Observations Regarding Algorithms Required for Robust CFD Codes”, to appear in *Modern Trends in Computational Aerodynamics – a Special Thematic Issue of Mathematical Modeling of Natural Phenomena*, June 2011.

[31] Allmaras, S. R., Bussoletti, J. E., Hilmes, C. L., Johnson, F. T., Melvin, R. G., Tinoco, E. N., Venkatakrishnan, V., Young, D. P., “Algorithm Issues and Challenges Associated with the Development of Robust CFD Codes”, *Variational Analysis and Aerospace Engineering, Springer Optimization and Its Applications*, 2009, Volume 33, pp. 1–19;*note: authors in alphabetical order; F. T. Johnson primary author*

[32] Venkatakrishnan, V., Allmaras, S. R., Kamenetskii, D., Johnson, F. T., “Higher Order Solutions for the Compressible Navier-Stokes Equations”, AIAA Paper 2003-3987, June 2003.

	C_l	C_d
Smoothed strand (320 points)	1.5578	0.02189
FUN3D (513 points)	1.5547	0.02159
CFL3D (513 points)	1.5461	0.02124

Table 4. Comparison of computed lift and drag coefficients for flow over a NACA 0012 airfoil at $M = 0.15$, $\alpha = 15^o$ $Re = 6 \times 10^6$. Note that the FUN3D and CFL3D results use a much finer grid.