SAND2017-8267R

Final Report for ALCC Allocation : Predictive Simulations of Complex Flow in Wind Farms

ExaWind Project Team: NREL, Sandia National Laboratories*, UT-Austin
Matt Barone (ALCC PI)

Shreyas Ananthan Matt Churchfield Stefan Domino
Marc Henry de Frahan Robert Knauss Jeremy Melvin
Robert Moser Michael Sprague Stephen Thomas

July 18, 2017

1 Introduction

This report documents work performed using ALCC computing resources granted under a proposal submitted in February 2016, with the resource allocation period spanning the period July 2016 through June 2017. The award allocation was 10.7 million processor-hours at the National Energy Research Scientific Computing Center. The simulations performed were in support of two projects: the Atmosphere to Electrons (A2e) project, supported by the DOE EERE office; and the Exascale Computing Project (ECP), supported by the DOE Office of Science. The project team for both efforts consists of staff scientists and postdocs from Sandia National Laboratories and the National Renewable Energy Laboratory.

At the heart of these projects is the open-source computational-fluid-dynamics (CFD) code, Nalu[1]. Nalu solves the low-Mach-number Navier-Stokes equations using an unstructured-grid discretization. Nalu leverages the open-source Trilinos solver library [1] and the Sierra Toolkit (STK) [2] for parallelization and I/O.

This report documents baseline computational performance of the Nalu code on problems of direct relevance to the wind plant physics application - namely, Large Eddy Simulation (LES) of an atmospheric boundary layer (ABL) flow and wall-modeled LES of a flow past a static wind turbine rotor blade. Parallel performance of Nalu and its constituent solver routines residing in the Trilinos library has been assessed previously under various campaigns. However, both Nalu and Trilinos have been, and remain, in active development and resources

*Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

[1]https://github.com/nalucfd/nalu; http://nalu.readthedocs.io

have not been available previously to rigorously track code performance over time. With the initiation of the ECP, it is important to establish and document baseline code performance on the problems of interest. This will allow the project team to identify and target any deficiencies in performance, as well as highlight any performance bottlenecks as we exercise the code on a greater variety of platforms and at larger scales. The current study is rather modest in scale, examining performance on problem sizes of O(100 million) elements and core counts up to 8k cores. This will be expanded as more computational resources become available to the projects.

2 ABL Simulations

2.1 Problem Setup

The first test case used to generate performace data consists of simulation of a turbulent atmospheric boundary layer under neutral stratification (*i.e.*, isothermal conditions). The domain is a 3km long by 3km wide by 2km high box, with periodic boundary conditions specified at all four lateral boundaries, a rough-wall surface layer boundary condition applied at the ground plane, and an open boundary condition with specified geostrophic wind velocity applied on the top boundary. The solution is initialized with a power law boundary layer velocity profile along with superimposed dilatation-free perturbations applied within the boundary layer to acceleration transition of the flow to a turbulent state. These LES cases are run with a standard Smagorinsky sub-grid scale model. A snapshot of the wind field from one of the simulations is shown in Figure 1. The intent of these simulations was not to perform verification or validation of the LES per se; we only focused on setting up as realistic a case as possible with current physics model capability within the code and observing qualititative solution behavior to ensure the simulations are relevant for our application. However, these cases are expected to also serve as useful V&V cases as we mature the physics model capability.

The mesh consists of a single, structured hexahedral block with uniform spacing in all three coordinate directions. Two meshes are used in this study: a 256x256x256 element mesh, containing a total of 16.7 million elements; and a 512x512x512 element mesh, containing a total of 134.2 million elements. The time step is specified to maintain a maximum Courant number of approximately 0.7 for each simulation; thus, as the mesh is refined, the time step is reduced accordingly. The cases were run for a minimum of 5000 time steps up to a maximum of 50,000 time steps, during which timings were collected. Timings within Nalu are wall-clock times reported for individual elements of the equation system setup and solve operations, for each equation set that is solved. The timings reported here are times per time step, obtained by taking the maximum total wall-clock time for each operation over all MPI ranks, and dividing by the number of time steps. Here, we are assessing MPI-only performance of Nalu; threading strategies are not considered in these baseline simulations.

Most of the ABL simulations were run on the Cori machine at the National Energy Research Scientific Computing Center. Cori is a Cray XC40 computer with two partitions: an Intel Haswell partition (2004 Haswell compute nodes, 32 cores per node), and an Intel Knights Landing partition (9304 KNL nodes, 68 cores per node). The present results are all obtained

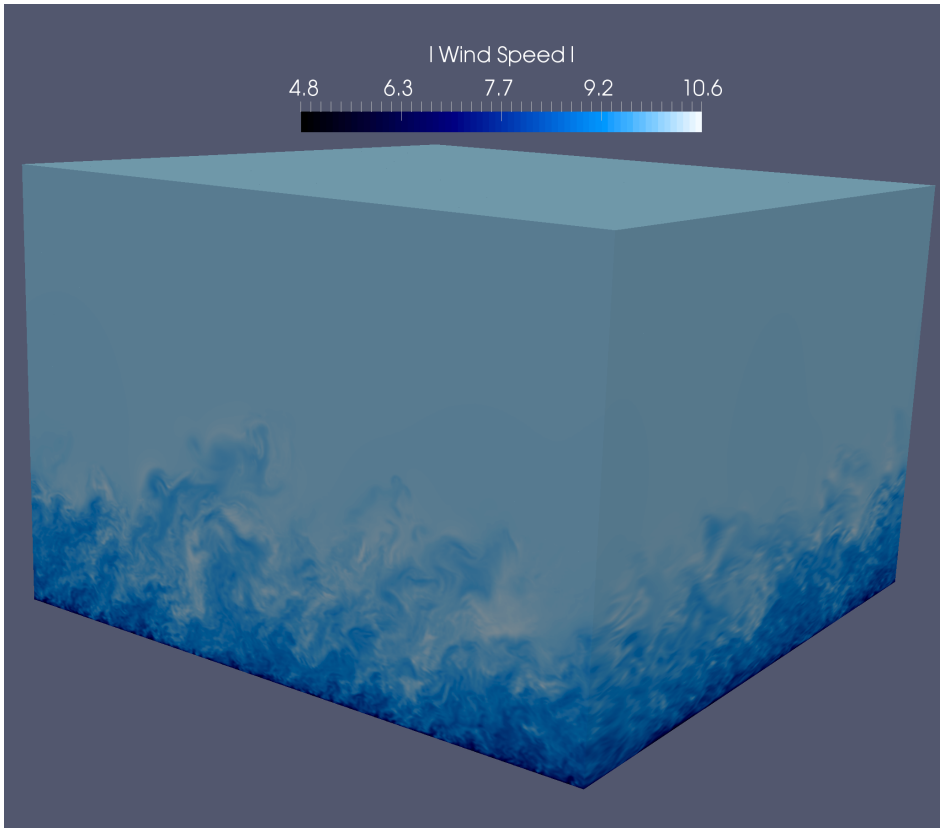

Figure 1: Wind velocity in the driving geostropic direction from LES simulation of the atmospheric boundary layer.

running on the Haswell partion only, with 32 MPI ranks per node. Note that Cori is a relatively new machine, but is very similar to the National Nuclear Security Administration's Trinity machine, on which Nalu has been run at large scale for both acceptance testing as well as during the Trinity Open Science period.

2.2 ABL Simulations on Cori

Results are first presented for strong scaling. We show wall-clock speed-up factors as problem size is kept constant and number of cores (and MPI ranks) is increased. Figure 2 shows strong scaling results for the major momentum equation operations for both the 16.7 million element mesh and the 134.2 million element mesh. In all the plots, the black dashed line represents ideal speed-up, where speed-up is calculated relative to the timing for the job run on the smallest number of cores for a particular problem size. The momentum assembly and solve operations exhibit excellent strong scaling, while the preconditioner setup is only slightly sub-optimal as the number of elements per core reduces to 16K.

Figure 3 gives strong scaling results for the continuity equation system. This involves setup and solution of a Poisson system for pressure. The assembly operation demonstrates ideal scaling or better, while the preconditioner setup and the solve operations suffer a

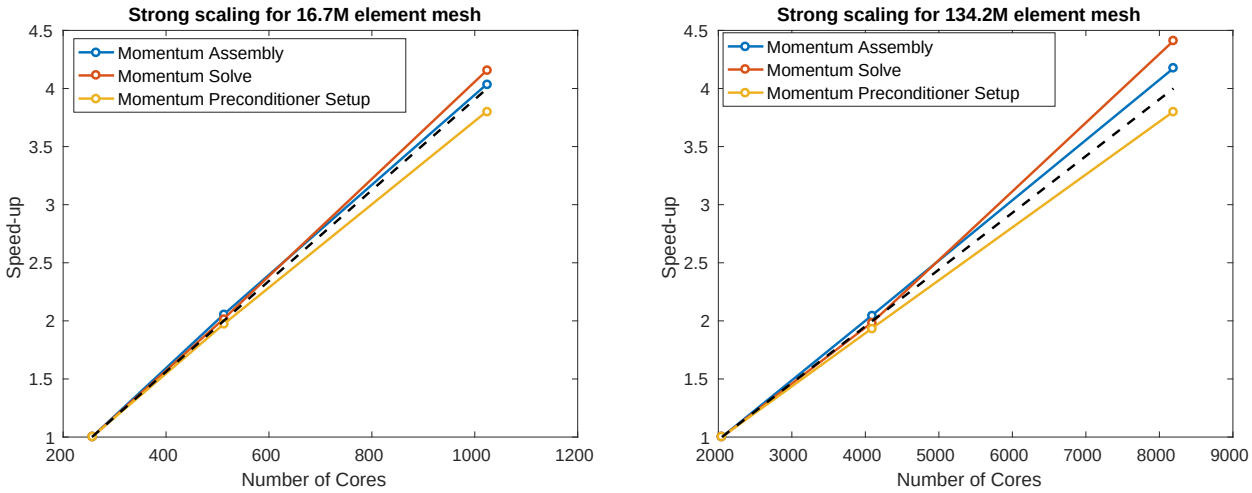

Figure 2: Strong scaling for momentum equation system operations.

degradation in performance as the number of cores increases. Note that the number of linear solver iterations remained nearly constant for each of the simulations. Figure 4 shows strong scaling for the total wall-clock time per time step. While the parallel efficiency begins to deteriorate for 8192 cores, this data point represents an agressive element count per-core of approximately 16,000.

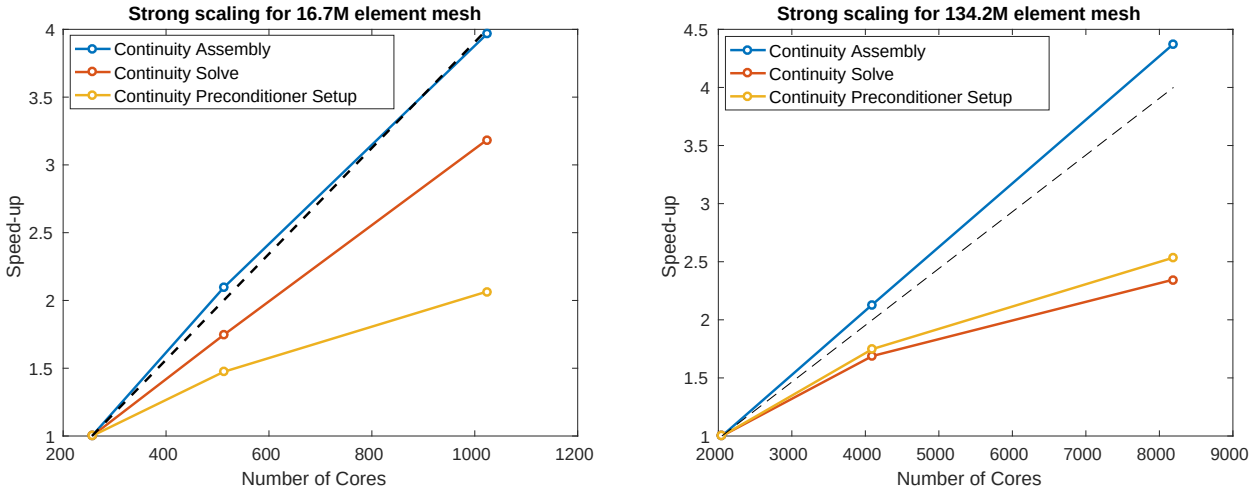

Figure 3: Strong scaling for continuity equation operations (Poisson system) on Cori.

Figure 5 shows weak scaling results for the momentum solve. In weak scaling, the amount of computational work per core is kept constant as the number of cores increases. We plot the time per time step (in seconds) as the number of cores increases, and ideal behavior is that this time remains constant. The conclusions drawn from these results are the same as for the strong scaling study.

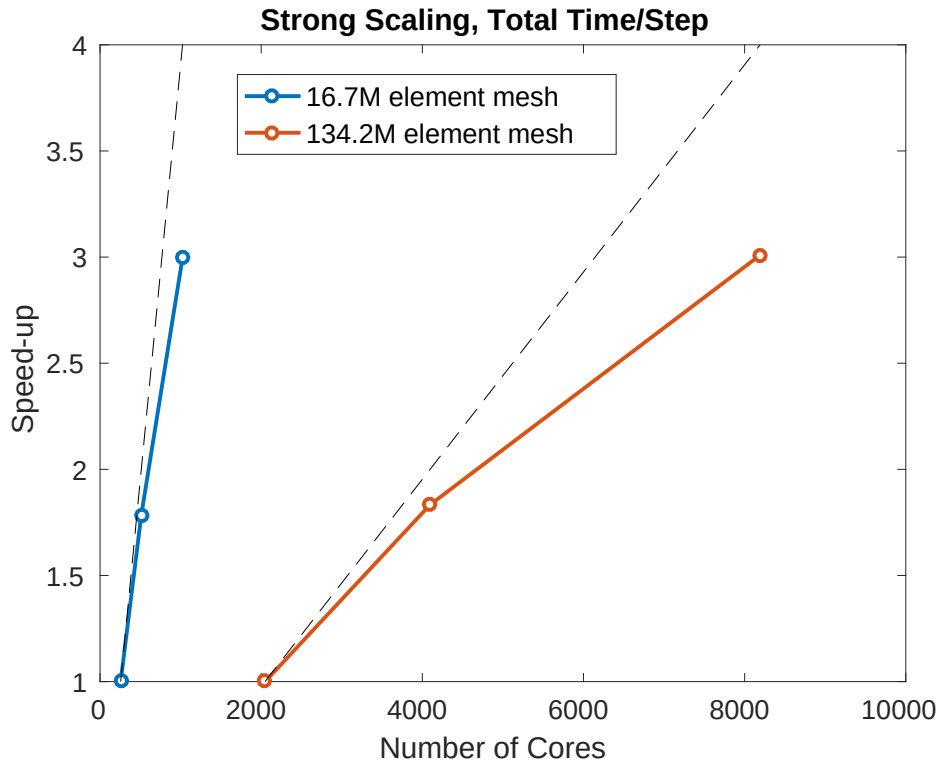

Figure 4: Strong scaling for total wall-clock time per step on Cori.

3 Blade Simulations

We examine here simulations of a finite-length (3.3 ft) NACA0015 blade with 1 ft cord length that was experimentally investigated in a wind tunnel by Mcalister and Takahashi [3]. This system was chosen because experimental data includes details about tip vortex evolution for flat- and rounded-tip configurations in addition to other detailed flow measurements.

This section is organized as follows.

- description of the RANS-LES turbulence model, with testing results,
- description of the numerical methods,
- description of simulation, timing, and scaling results,
- description of preliminary validation through comparison of Nalu results with those published in [3].

3.1 RANS-LES turbulence model

3.1.1 Model description

The hybrid Reynolds-averaged-Navier-Stokes large-eddy-simulation (RANS-LES) turbulence modeling approach chosen for this work is the Shear Stress Transport (SST) – Detached

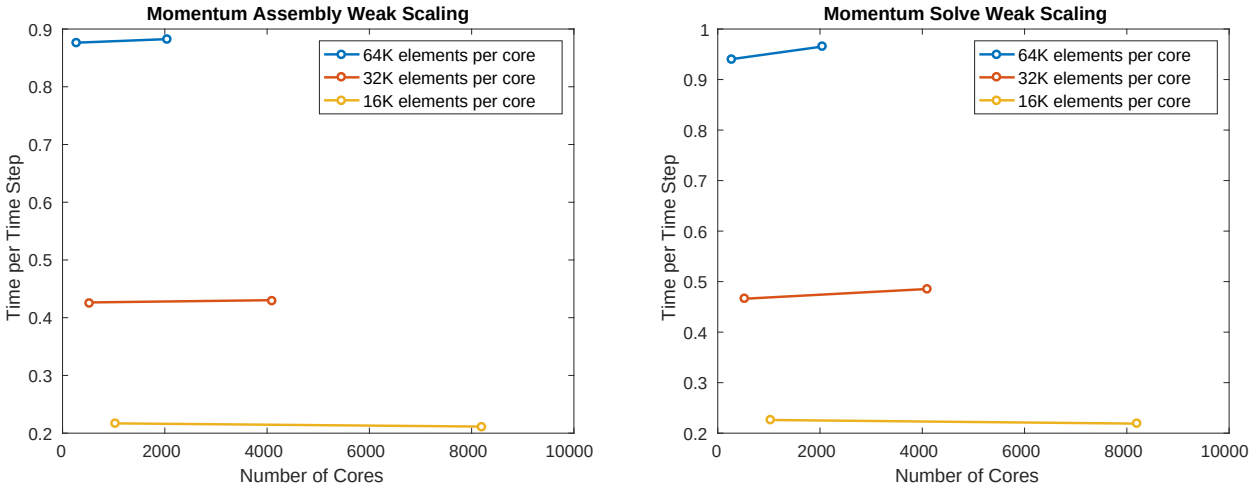

Figure 5: Weak scaling for momentum equation system assemble and solve on Cori.

Eddy Simulation (DES) turbulence model. The 2003 Menter SST model [4] transports the turbulent kinetic energy, k, and the specific dissipation rate, ω. This model combines the advantages of the $k-\epsilon$ and $k-\omega$ models by blending these models to use the former far from the walls and the latter near the walls. The SST-DES model provides a theoretical basis for transient flows by relaxing the solely RANS-based approach. In this model, the SST kinetic-energy equation is modified to incorporate a new minimum dissipation scale. Details about the models and their implementation in Nalu can be found in the Nalu theory documentation (SST, DES[2], and LES[3]). Additionally, we performed large eddy simulations using a nonlinear stabilization operator as a subgrid scale turbulence model. Details about this model can be found in Section 3.2.3.

3.1.2 Automated test suite

Unit tests were developed to exercise the core components of the hybrid RANS-LES turbulence model. This includes tests for the maximum length scale calculation, the turbulent viscosity computed through various turbulence models (SST, LES k_{sgs}, LES WALE, and LES Smagorinsky), and various source terms for the k and ω evolution equations. Following previous testing and verification efforts, the unit tests use manufactured solutions for the necessary input fields to calculate the output field (e.g., turbulent viscosity, source terms, etc.). Field norms are compared to target norms obtained from an independent code. A github pull request[4] describes the unit-test implementation in Nalu.

[2]http://nalu.readthedocs.io/en/latest/source/theory/supportedEquationSet.html
[3]http://nalu.readthedocs.io/en/latest/source/theory/turbulenceModeling.html
[4]https://github.com/NaluCFD/Nalu/pull/155

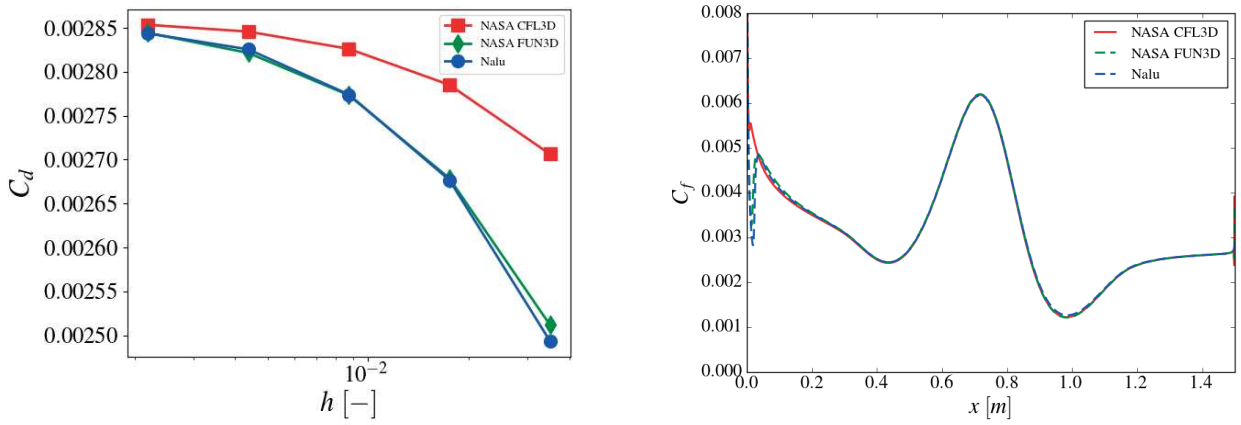

Figure 6: Code-to-code comparison between Nalu and NASA codes for the flat plate drag coefficient vs. element size (left) and the skin friction coefficient distribution for a 2D bump in channel flow at t = 0.4 (1409×641 mesh) (right) using the SST RANS model.

3.1.3 Verification and validation efforts (SST RANS)

Code-to-code verification of the SST RANS model was performed in a comparison of several standard test cases proposed in NASA's Turbulence Modeling Resource[5]. For these simulations, as opposed to those of the McAlister blade described in Section 3.2.7, no wall function was used to model the boundary layer. Figure 6 shows Nalu simulation results of the 2D zero pressure gradient flat plate and the 2D bump-in-channel as well as simulation results from the NASA CFL3D and FUN3D codes. Though both of NASA's codes solve the compressible Navier-Stokes equations and Nalu solves the low-Mach equations, the Mach numbers for these test cases are low enough such that the flow can be considered incompressible. Figure 6(a) shows the drag coefficient, C_d, as a function of grid size. FUN3D is a vertex centered scheme and uses a normal distance at the wall node close to that used in Nalu. This accounts for the very close agreement between Nalu and FUN3D. For the 2D bump-in-channel, Figure 6(b) shows the skin-friction-coefficient streamwise distribution, C_f, at $t = 0.5$, where each model has the same mesh (1409×641).

Following NASA's Turbulence Modeling Resource validation test cases, experimental data of turbulent flow over a backward facing step [5] were used to perform validation of the SST RANS model. Figure 7 shows vertical velocity distributions at two locations in the domain as produced by Nalu and CFL3D along with experimental results. Nalu simulation results compare well with the experimental data and published NASA code results.

3.2 Numerical methods

The Nalu code base, as described in greater detail within the theory manual[6], supports a low-order control volume finite element (CVFEM) and edge-based vertex centered (EBVC) on generalized topological meshes and a high-order CVFEM implementation on hexahedral

[5]https://turbmodels.larc.nasa.gov
[6]http://nalu.readthedocs.io

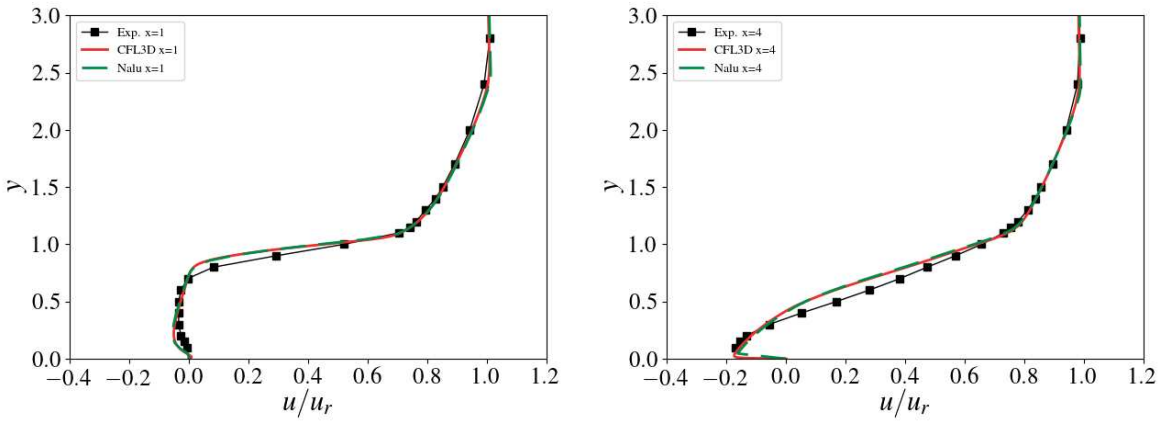

Figure 7: Comparison of velocity profiles from Nalu using the SST RANS model, the NASA CFLD3D code, and experimental data [5] at different downstream locations. $x = 1$ behind the step (left). $x = 4$ behind the step (right).

meshes. This set of vertex-based schemes are naturally described as Petrov-Galerkin schemes because the test function (defined as piecewise-constant) is different than the underlying interpolation scheme. A representative hexahedral and tetrahedral element, with associated dual-mesh definitions, can be found in Figure 8. Quadrature points are either at the subcontrol surfaces for integrated-by-parts terms, e.g., advection and diffusion, or subcontrol volume points for volumetric contributions, e.g., source terms and mass matrix. For the edge-based scheme, subcontrol volume surfaces are assembled to all connected edges whereas subcontrol volumes are assembled to all connected nodes. Therefore, the element-based scheme is transformed to an edge- and node-based data structure.

In all fluids algorithms, an equal-order interpolation scheme is used. This drives the need for pressure stabilization which can be interpreted as adding a fine-scale momentum residual to the continuity equation. In this low-Mach turbulent flow application, an incremental, approximate pressure-projection scheme is used. Splitting errors are controlled by nonlinear iteration which, along with the BDF2 time integrator, demonstrates second-order-in-time accuracy. The low-order schemes are limited to second-order accuracy, whereas a generalized polynomial promotion demonstrates spectral convergence.

An interesting numerical nuance of the CVFEM methodology is that the Laplace (i.e., diffusion) operator, can be non-monotonic for element aspect ratios greater than $\sqrt{3}$ (the classic finite element method diffusion operator loses monotonicity at $\sqrt{2}$). Extreme volume-element volume jumps, especially on tetrahedral meshes, are also problematic. Although the diffusion operator for the edge-based scheme does not suffer monotonicity issues, the usage of nonorthogonal correction terms, can be problematic. These correction terms are a simple consequence of the fact that, on unstructured meshes, the edge-distance vector and edge-area vector are not orthogonal.

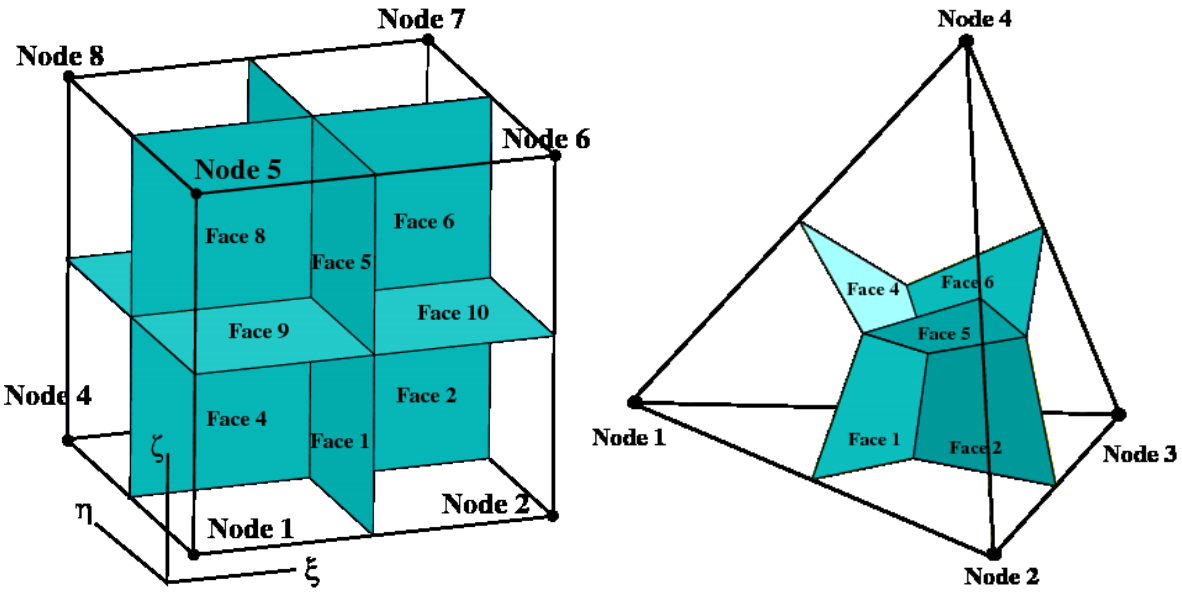

Figure 8: Representative CVFEM dual-mesh definitions. Low-order hexahedral CVFEM element (left) and low-order tetrahedral CVFEM element (right).

3.2.1 Hybrid CVFEM and EBVC

When concentrating on low-order solutions, a compromise between speed and accuracy can be found. Specifically, the continuity equation may be solved using an element-based scheme while the time/advection/diffusion/source PDE set can use an edge-based scheme. This hybrid scheme requires the mass flow rate from the continuity system to be assembled to the edges of the element.

3.2.2 Quadrature for CVFEM

In order to mitigate the non-monotonic issues associated with the diffusion operator on high-aspect ratio meshes, the CVFEM subcontrol surface quadrature points can be moved (or shifted) to the element edge-midpoints. For orthogonal meshes, the scheme reverts to a canonical seven-point stencil on hexahedral meshes. For high aspect ratio meshes, this mitigates the sign flipping of the off diagonal diffusion operator column entries. On tetrahedral meshes, moving quadrature points does not help as the gradient operator is constant.

Frequently, the pressure field is very smooth and shifting quadrature points is not required. However, in certain situations which are associated with high aspect ratio, hybrid meshes, the algebraic multigrid coarsening algorithm can suffer due to this non-monotonic operator. The numerical scheme also supports using a reduced stencil Jacobian, while allowing for the full element-based stencil for the residual.

3.2.3 Advection stabilization

A variety of stabilization approaches are supported in Nalu. The stabilization ranges from Peclet blending of limited second-order upwind methods (MUSCL) to the usage of nonlinear stabilization operators (NSO). For details on the stabilization supported, the reader is referred to the Nalu theory manual[7].

Although usage of upwind-limited stabilization operators remains the norm for hybrid-based simulations, usage of such methods on generalized unstructured meshes provides a numerical scheme that is limited to second-order in space. Therefore, the project is evaluating the usage of NSO for both stabilization and even LES turbulence models. The formal finite element kernel for a NSO approach is as follows,

$$\sum_e \int_\Omega \nu(\mathbf{R}) \frac{\partial w}{\partial x_i} g^{ij} \frac{\partial \phi}{\partial x_j} d\Omega, \tag{1}$$

where $\nu(\mathbf{R})$ is the artificial viscosity which is a function of the PDE fine-scale residual and g^{ij} is the covariant metric tensor. At present, a variety of fine scale residuals are supported. When the form of ν is proportional to the equation set, we view this as an advection stabilization approach. In some cases, ν can be written in terms of a fine scale kinetic energy residual and follows concepts similar to an entropy-viscosity approach where the interpretation is transformed from pure advection stabilization to an implicit LES modeling approach. Using a piecewise-constant test function suitable for CVFEM and EBVC schemes Eq. 1 can be written as,

$$-\sum_e \int_\Gamma \nu(\mathbf{R}) g^{ij} \frac{\partial \phi}{\partial x_j} n_i dS. \tag{2}$$

3.2.4 Linear solvers

The main computational costs incurred during a time-step are due to the iterative solution of the momentum and continuity equations, embedded within an outer nonlinear-system-solve Picard iteration. The momentum equation includes gradient and divergence operators associated with advection and pressure force terms. The resulting matrix is non-symmetric and three-times larger than the continuity equation matrix. The continuity equation involves the Laplacian and the CVFEM discretization can result in a non-symmetric matrix form. Nalu employs a time-split algorithm where the momentum and continuity equations are solved separately instead of together in a large block matrix form. Both the momentum and continuity linear solvers are based on the preconditioned generalized miminum residual (GMRES) Krylov iteration for non-symmetric matrices. The momentum linear system is far less 'stiff' than the continuity equation. Either a stationary relaxation or incomplete ILU factorization is typically employed as the momentum preconditioner. For the more ill-conditioned continuity solver, smoothed-aggregation algebraic multigrid (SA-AMG) is employed as the preconditioner. The symmetric SA-AMG algorithm in the Trilinos/MueLu framework is applied with an implicit restriction operator where the coarse matrix is $A_c = P^T A P$. The SA-AMG coarsening algorithm generates the matrices at each level of a V-cycle hierarchy. The coarsening rate between levels is crucial for convergence and efficiency;

[7] http://nalu.readthedocs.io

typically an order-ten-times decrease in the number of rows between levels is optimal. Thus, a drop tolerance and possibly a distance Laplacian are required when the grid aspect ratio is large. Smoothers are applied at each level of the V-cycle and these may be relaxation schemes such as Gauss-Seidel-L1, Chebyshev polynomial iterations, or incomplete ILU factorizations. The latter take the form of additive-Schwarz sub-domain solves with or without overlap.

3.2.5 Simulation and timing results

Simulations of the [3] NACA0015 experiment were performed to establish baseline Nalu turbulence modeling and solver capabilities.

3.2.6 Experimental setup

The experiments were performed in the NASA Ames 7 ft $\times$ 10 ft Subsonic Wind Tunnel No. 2. The wings tested were semispan NACA0015 wings without twist or taper. These were mounted on a splitter plate extending from floor to ceiling and positioned about 1 ft away from the side wall. Wings with different aspect ratios, chord lengths, angle of attacks, and tip types (square or rounded) were instrumented to study pressure, lift, and drag coefficients at different Reynolds numbers, Re. Importantly, wing-tip vortex characteristics were measured using a laser-velocimeter system and a rapid-scan technique for capturing velocity profiles.

3.2.7 Simulation description

Nalu was used to perform simulations of the $Re = 1.5 \times 10^6$ case with a 12° angle of attack, a 1 ft chord length, 3.3 aspect ratio, and a rounded wing tip. To accurately model the experimental setup, the mesh geometry contains the tunnel walls, splitter plate (wall mount), and the half-span wing. These model components are shown in Figure 9. The full computational domain size is 30 ft in x (downstream direction), 10 ft in y (spanwise direction), and 7 ft in z (vertical direction). Boundary conditions on the splitter plate and the walls are set to no-slip. A wall function, as described in the theory manual[8], is used on the wing to model the boundary layer. The tunnel outlet is an open boundary condition. The tunnel inlet is an inflow boundary condition. The turbulence modeling framework is DES, a hybrid RANS-LES framework, with SST as the RANS model and Smagorinsky as the LES model (see §3.1). For this turbulence model, assuming an eddy viscosity ratio of $\frac{\mu_T}{\mu} = 0.1$ and a free stream turbulence intensity of $I = \frac{u'}{u} = 0.0015$, the inflow turbulent kinetic energy, k, and specific dissipation rate, ω, rate are determined by

$$k = \frac{3}{2}(uI)^2 \quad \text{and} \quad \omega = \frac{\rho k}{\mu}\left(\frac{\mu_T}{\mu}\right)^{-1},$$

respectively. The values for $\frac{\mu_T}{\mu}$ and I are values typically chosen for these types of simulations and inflow conditions.[9]

[8] http://nalu.readthedocs.io/en/latest/source/theory/boundaryConditions.html
[9] https://turbmodels.larc.nasa.gov

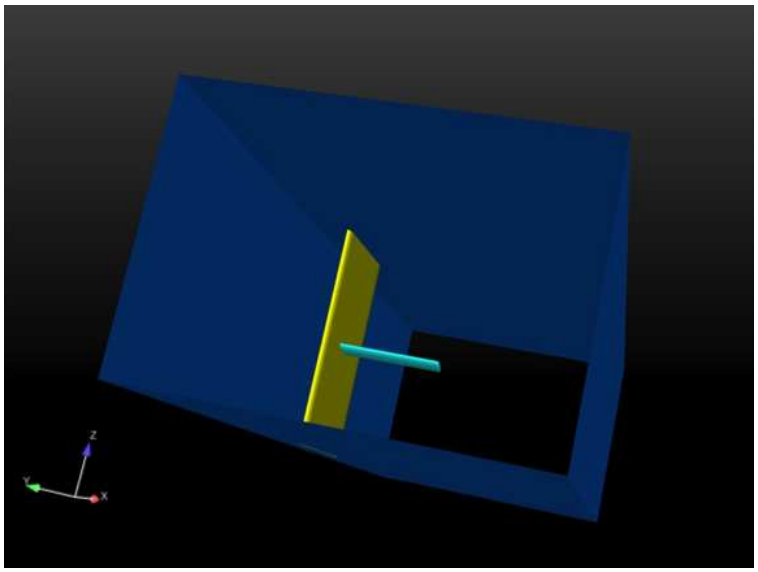

Figure 9: Simulation geometry with wind-tunnel side walls (blue), wing (cyan), and splitter plate (yellow).

3.2.8 Meshes

Using the commercial meshing package Pointwise[10], a series of fully conformal (point matched) unstructured meshes were constructed consisting of mixed finite element types. The wing surface is discretized with quadrilateral and triangle faces, which are elevated away from the surface to create a near-wing layer of hexahedral and wedge elements with faces that are naturally aligned parallel to, and normal to, the wing surface. This near-wing layer transitions to an outer region of the wind tunnel test section domain, which is meshed with tetrahedra. The transition from hexahedral to tetrahedral elements is achieved using pyramid transition elements. A hexahedral mesh block with fine mesh spacing is placed in the tip vortex region, beginning 1.2 chord lengths downstream from the wing tip and extending for 16 chord lengths to the end of the domain. Examples of the surface meshes are shown in Figure 10. This hexahedral mesh block is also surrounded by a layer of transitional pyramid elements separating it from the surrounding tetrahedral elements.

In our mesh-creation process a first coarse mesh was made, consisting of 65 million finite elements and 35 million nodes. We encountered substantial difficulty in converging the Poisson equation for pressure on this mesh. Closer examination of the mesh revealed a prevalence of highly skewed tetrahedral elements near the transition from the near-wing mesh layer and the outer tetrahedral-element region. There were over 8,000 finite elements with an equi-angle skewness metric greater than 0.9 on this mesh. Solution quality was poor where these finite elements were prevalent in areas of high solution gradients. The near-wing mesh region was regenerated, allowing for a larger number of element layers and alleviating the element skewness problem. This regenerated mesh was composed of 68M finite elements and 41M nodes. Note that some skewed elements were still present on the new mesh (about 15 finite elements with equi-angle skewness of more than 0.9); however, these elements were

[10]http://www.pointwise.com

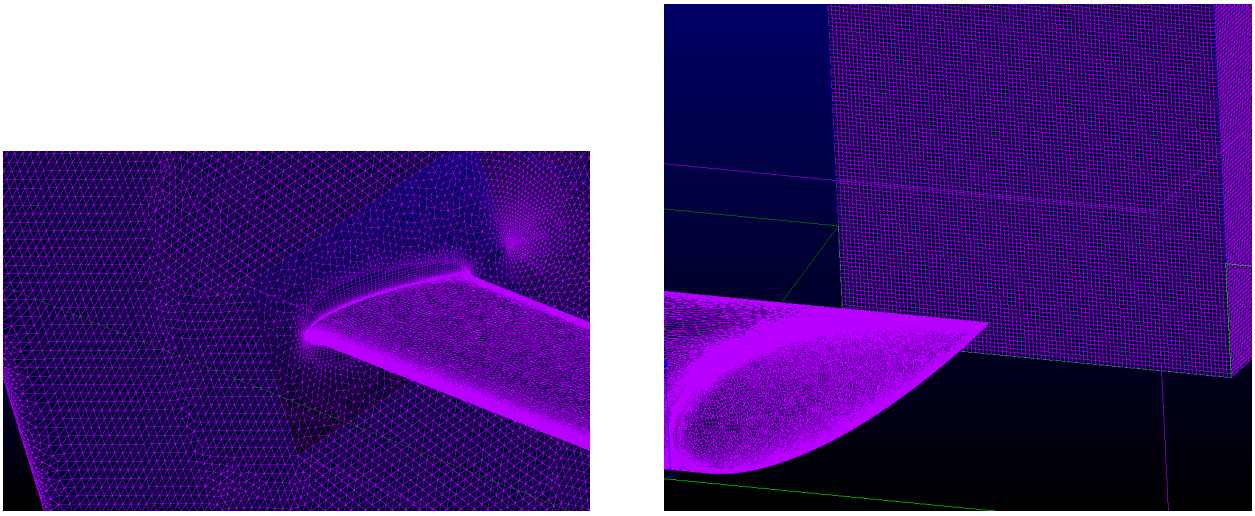

Figure 10: Surface meshes for the [3] wing. Mesh near wing root and splitter plate (left). Mesh on wing near the tip and tip-vortex block (right)

no longer present in a region of high solution gradients. Poisson-system solver performance, as well as solution quality, improved dramatically with this improvement in the mesh.

A finer mesh was constructed by refining the tip vortex hexahedral block by a factor of two in each coordinate direction. This resulted in an increase in finite element count in the tip vortex block by a factor of eight, as well as an increase in element count in the surrounding unstructured mesh blocks adjoining the hexehedral block. The total finite element count of this finer mesh is 299 million, while the node count is 224 million. The finite element count increased by a factor of 4.6 over the coarser mesh, while the node count increased by a factor of 6.4.

These baseline meshes were constructed with a very fine wall-normal mesh spacing at the wing surface, in order to allow the near-wall RANS model to be directly integrated to the surface without use of a wall function. Thus, the maximum finite element aspect ratio is approximately 34,000. In order to examine the impact of element aspect ratio on solver performance, we also constructed a family of meshes with near wall spacing that is one hundred times larger than the original spacing. Thus, the maximum element aspect ratio on this second family of meshes is approximately 340. In simulations on these meshes, a wall function is used to provide an appropriate shear stress boundary condition value on the wing surface. Table 1 summarizes the meshes used in this study.

Mesh Name	Elements (M)	Nodes (M)	Max Elem. AR	Max Elem. VR
grid07conformal03	68.3	40.5	34,672	26.8
grid07conformal03lowAR	63.6	35.6	346.7	32.4
grid07conformal04	298.8	224.3	34,672	26.8
grid07conformal04lowAR	294.31	219.45	346.7	32.4

Table 1: Mesh characteristics. AR = Aspect Ratio. VR = Volume Ratio.

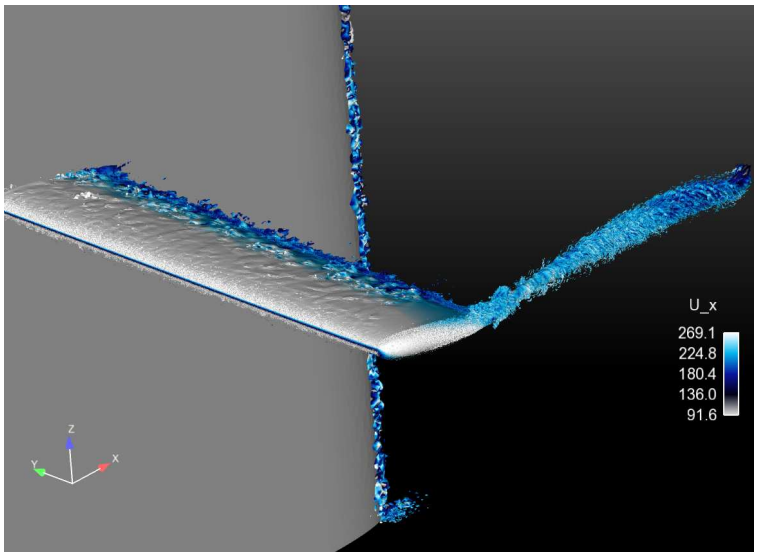

Figure 11: Flow visualization of the tip vortex generated by the McAlister wing using an unstructured mesh with 68M finite elements.

3.2.9 Results

Simulations were run with a variety of turbulence models and linear-solver options. Figure 11 shows representative results for the *grid07conformal03* mesh where the tip vortex is evident.

Linear solver performance

As all of the simulations were run using a Courant-limited timestep of less than 2, the time/advection/diffusion solves using GMRES and symmetric-Gauss-Seidel, e.g., momentum and SST equation sets, demonstrated highly efficient solves. The increased diagonal dominance of the linear system due to the timestep (mass) term provided for linear GMRES iterations of generally less than 5. All simulations using the low-aspect ratio mesh were stable using both the SST-DES and LES models.

The continuity solver also employs the non-symmetric GMRES Krylov iteration, but with an algebraic-multigrid (AMG) preconditioner provided by the Trilinos MueLu framework. The coarsening algorithm is based on smoothed aggregation AMG with a drop-tolerance. The AMG smoother adopted for these simulations was one iteration of additive-Schwarz with an ILU(k) factorization of the sub-domain matrix. The XML file containing these settings is provided in Appendix A. To investigate the effect of mesh resolution on the continuity solve, Figure 12 outlines the GMRES iteration counts for the shifted CVFEM Jacobian with the full CVFEM Laplace operator for both the 68M and 294M SST-DES simulation. The simulation is using additive-Schwarz (0-overlap) ILU(0). Results indicate similar iteration counts for the LES-based models indicating that physics does not drive continuity solver performance (not shown). This finding is expected based on the isothermal, uniform turbulent flow physics in use. Average continuity iteration counts for all of the simulations over 20,000 time steps were generally $\mathcal{O}(15)$. Averages for momentum and SST solves were less than $\mathcal{O}(5)$.

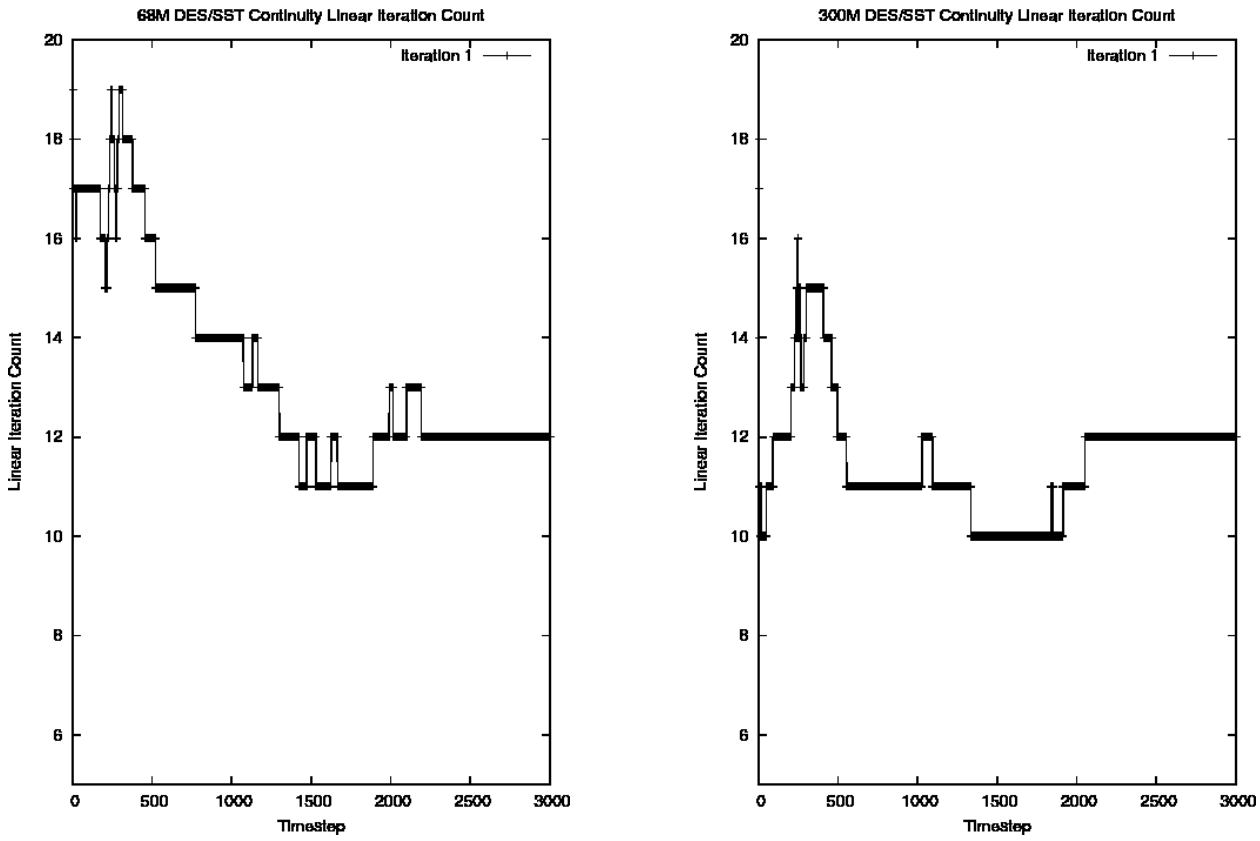

Figure 12: Number of GMRES solver iterations in the continuity solve for each time step. 68M element mesh *grid07conformal03lowAR* (left). 300M element mesh *grid07conformal04lowAR* (right).

Strong scaling

The McAlister-blade Nalu model was employed in a strong scaling performance study on the NERSC Cori[11] supercomputer. The Nalu model was tested with 32-core Intel Haswell nodes under MPI-only parallelism. For testing purposes, the model was integrated 50 time steps past an initial transient to a point where the Courant number of the flow is approaching 1 and the GMRES iteration counts are relatively constant for both momentum and continuity solves. The continuity solver takes up almost half of the measured run-time and the choice of MueLu-AMG smoother affects the GMRES iteration counts and, consequently, the overall run-time of the model. The smoothers compared for the full CVFEM discretization were a novel additive-Schwarz method (ASM) with threshold incomplete factorization ILUT(1) with the maximum fill per row specified, four sweeps of Gauss-Seidel-L1, four sweeps of the Chebyshev iteration (XML parameters provided in Appendix A). The threshold variant of the ILU smoother is required to handle the more complex CVFEM discretisation, for the continuity equation with numerical issues including monotonicity and high mesh aspect ratios. A distance Laplacian is required for the aggregation algorithm when the full CVFEM is employed. The 68M finite element mesh problem was executed with pure MPI on up to 8196 cores. Run times are plotted in Figure 13.

With 1024 cores there are 68K finite elements per core and at 2048 cores with roughly 33K finite elements per core. We observe that the Nalu code continues to scale in the range of 2048 up to 6144 cores, however, with the Gauss-Seidel-L1 and Chebyshev smoothers the run-times begin to trail off beyond 6144 processor cores when there are 11000 finite elements per core. This differs from previous studies based on hybrid edge/element discretizations without the full CVFEM discretisation. Parallel efficiency for these simulations is plotted in Figure 14. In contrast with the Gauss-Seidel-L1 and Chebyshev smoothers, the ASM-ILUT(1) smoother exhibits superior run-times and parallel efficiency remains above the important 0.5 level. The three smoothers are expected to display different scaling properties at high core counts because Chebyshev employs matrix-vector products, whereas Gauss-Seidel-L1 solves are local to the MPI rank sub-domain. The ASM-ILUT(1) smoother is also local to sub-domains but requires only one sweep. Single-processor efficiency of the incomplete ILU factorization smoothers depends directly on the sparse forward and backward solvers in Trilinos. These could be further improved for optimal L1-cache memory access patterns. The full CVFEM scaling study was also carried out on the NREL Peregrine cluster with 24 Haswell cores per node. The results up to 101 nodes are presented in Figure 15. We observe that the timing curves are steeper than Cori and flatten slightly less towards the higher core counts. We attribute the improved scaling on Peregrine to fewer nodes per core. However, we did not push beyond 28K finite elements per core.

In order to assess the performance of Nalu on the McAlister problem at higher core counts, the mesh resolution was increased to 300 million finite elements. The number of MPI ranks was varied on Cori from 1024 up to 16384 cores. where the subdomain size is 18K finite elements at the highest core count. A simple WALE turbulence model was applied for this scaling study. A hybrid edge/element discretization was employed for these simulations. An additive-Schwarz ASM-ILU(0) smoother for the continuity GMRES-AMG solver was found to be very effective as noted above for turbulence modeling. Chebyshev

[11] http://www.nersc.gov/users/computational-systems/cori/

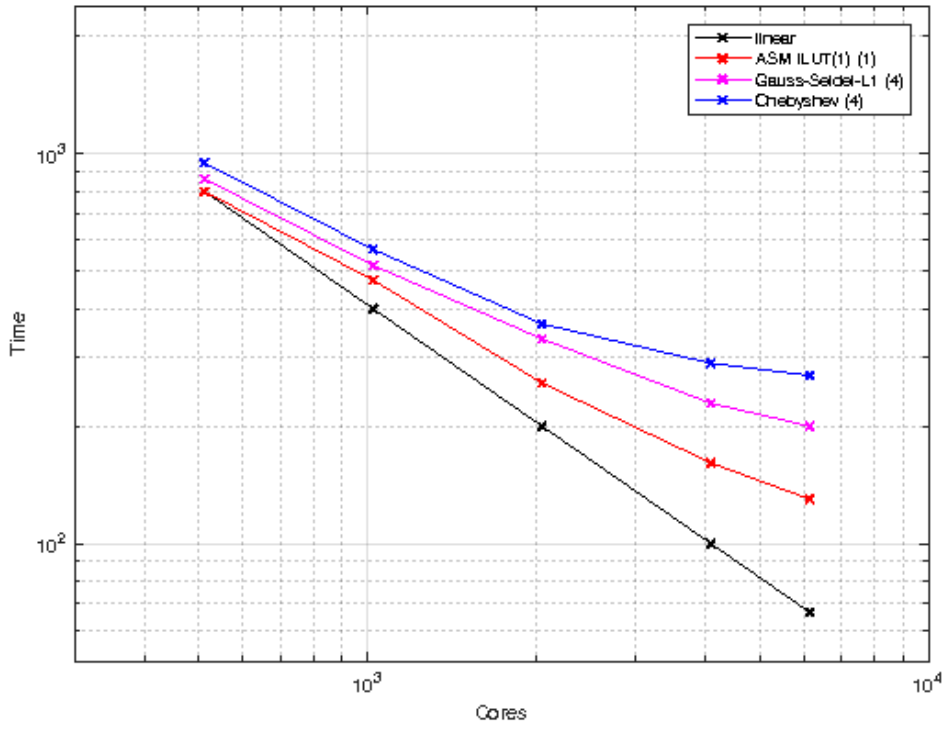

Figure 13: Strong scaling study on NERSC Cori using the McAlister 68M finite element mesh *grid07conformal03lowAR*. The full CVFEM discretisation stencil is specified and WALE turbulence model is applied. Continuity GMRES MueLu-AMG smoothers; additive-Schwarz method (ASM) with threshold incomplete factorization ILUT(1) with max fill per row, Chebyshev iteration with degree (4) polynomial or sweeps and Gauss-Seidel-L1 relaxation with (4) sweeps.

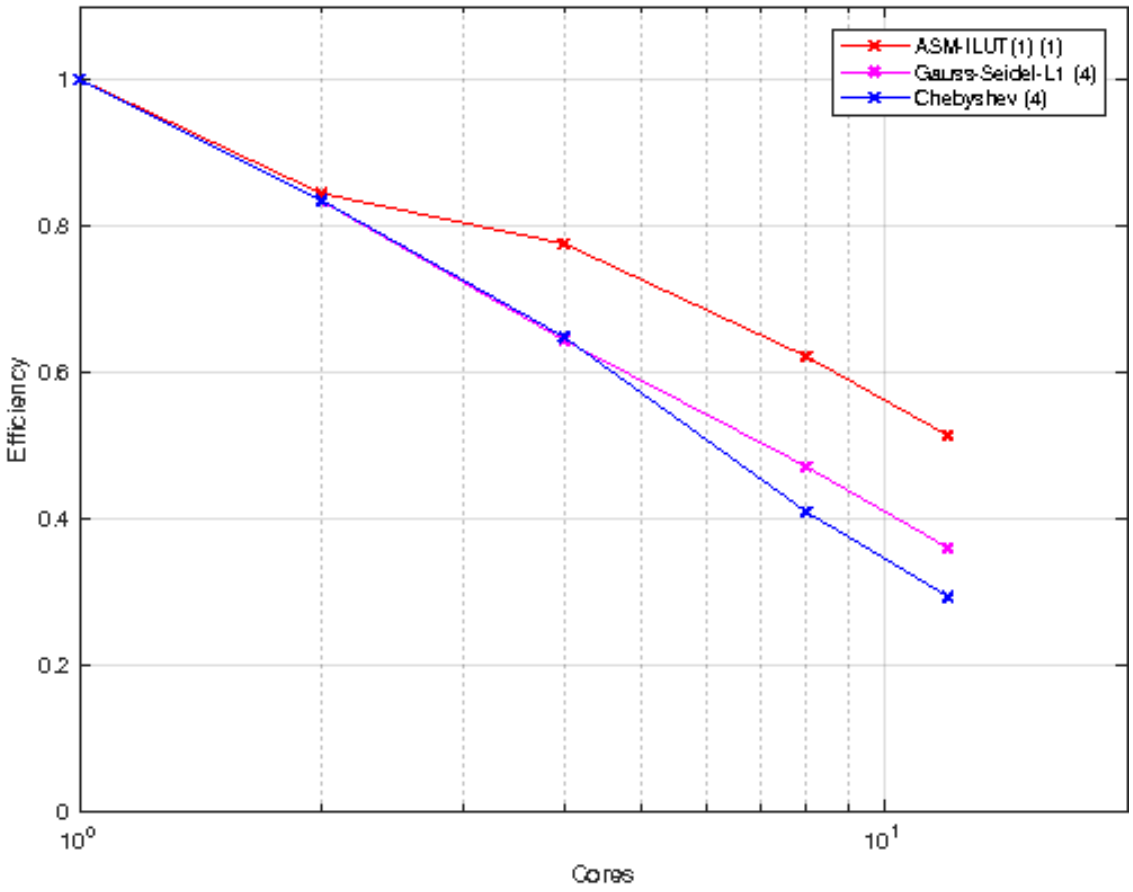

Figure 14: Parallel efficiency on NERSC Cori using the McAlister 68M finite element mesh *grid07conformal03lowAR*. The full CVFEM discretisation stencil is specified and WALE turbulence model is applied. Normalized cores from 1 through 24 are employed on the x-axis. Continuity GMRES MueLu-AMG smoothers; additive-Schwarz method (ASM) with threshold incomplete factorization ILUT(1) with max fill per row, Chebyshev iteration with degree (4) polynomial or sweeps and Gauss-Seidel-L1 relaxation with (4) sweeps.

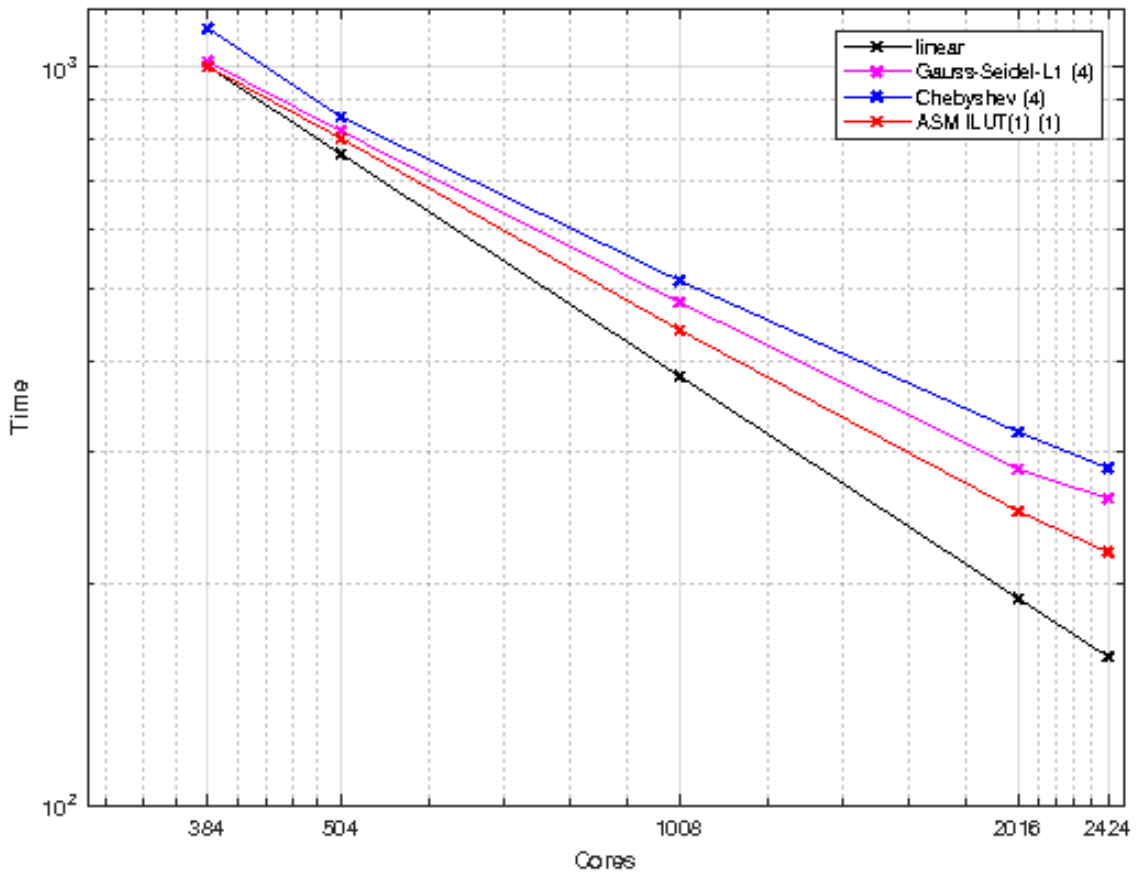

Figure 15: Strong scaling study on NREL Peregrine using the McAlister 68M finite element mesh *grid07conformal03lowAR*. The full CVFEM discretisation stencil is specified and WALE turbulence model is applied. Continuity GMRES MueLu-AMG smoothers; additive-Schwarz method (ASM) with threshold incomplete factorization ILUT(1) with max fill per row, Chebyshev iteration with degree (4) polynomial or sweeps and Gauss-Seidel-L1 relaxation with (4) sweeps.

and Gauss-Seidel-L1 smoothers were also included for comparison. In particular, we found that the continuity GMRES iteration counts remained relatively constant throughout the simulations with increasing core counts and corresponding sub-domain sizes, with a mean count of $\mathcal{O}(15)$. The scaling results are plotted in Figure 16 and the parallel efficiency is plotted in Figure 17. We note the efficiency drops below 0.5 above 8K cores when the number finite elements per core is less than 36K and there are 18K finite elements per core at 16K processors. The higher number of 32 cores per node on Cori versus 24 on Peregrine may contribute to the flattening of the strong scaling curve.

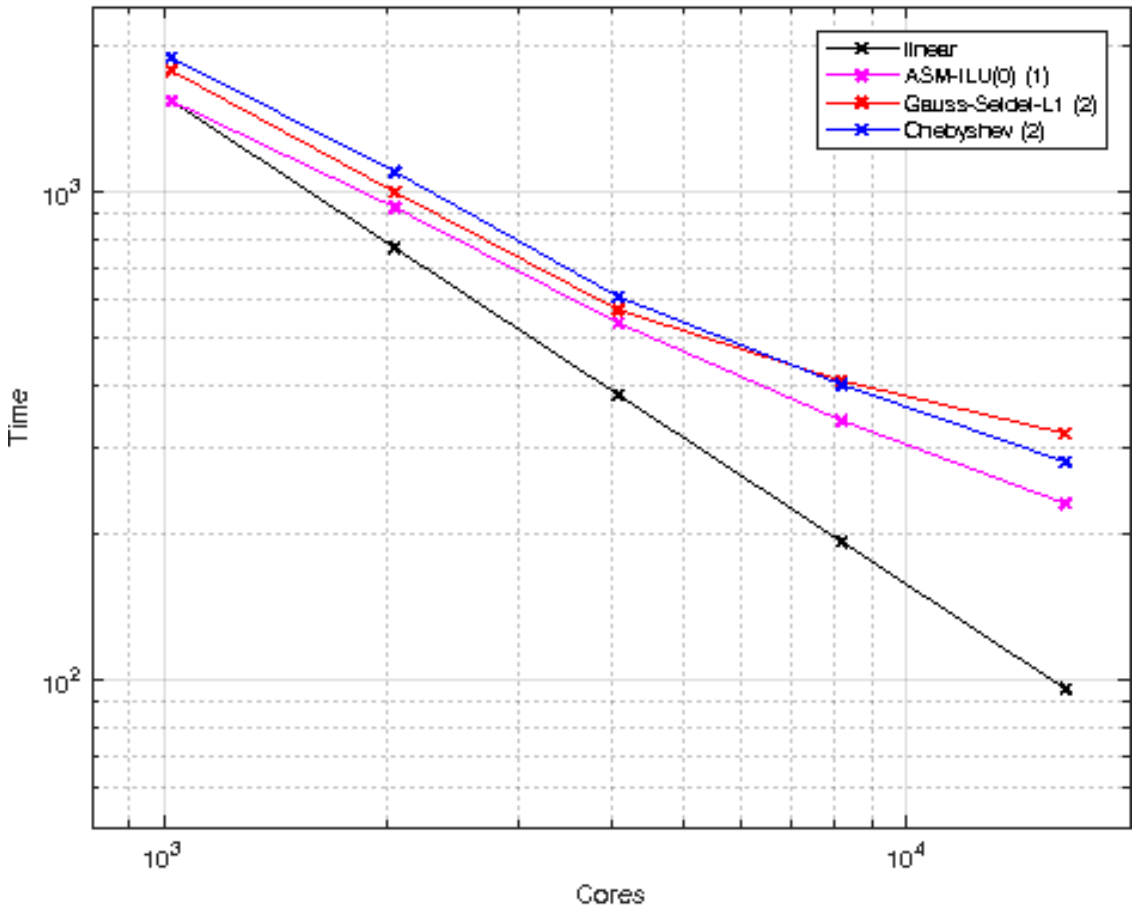

Figure 16: Strong scaling study on NERSC Cori using the McAlister 300M finite element mesh *grid07conformal04*. The hybrid edge momentum/element continuity discretisation is specified and WALE turblulence model is applied. Continuity GMRES MueLu-AMG smoothers; additive-Schwarz method (ASM) with incomplete factorization ILU(0) with 0-levels of fill, Chebyshev iteration with degree (2) polynomial or sweeps and Gauss-Seidel-L1 relaxation with (2) sweeps.

Preliminary validation

A preliminary validation study was performed to evaluate turbulence-model performance and solution dependence on mesh resolution. In the post-processing of tip-vortex simulation

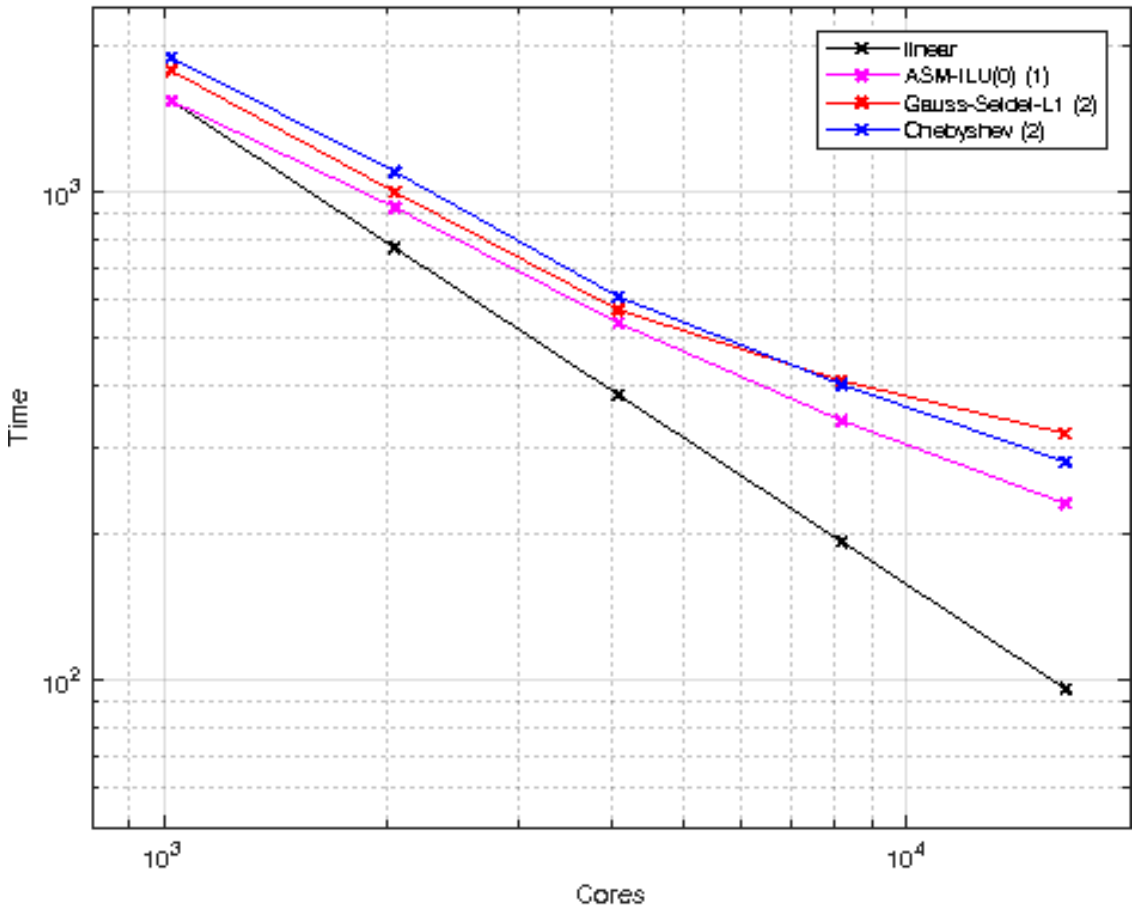

Figure 17: Parallel efficiency on NERSC Cori using the McAlister 300M finite element mesh *grid07conformal04*. The hybrid edge momentum/element continuity discretisation is specified and WALE turbulence model is applied. Normalized cores from 1 through 16 are employed on the x-axis. Continuity GMRES MueLu-AMG smoothers; additive-Schwarz method (ASM) with incomplete factorization ILU(0) with 0-levels of fill, Chebyshev iteration with degree (2) polynomial or sweeps and Gauss-Seidel-L1 relaxation with (2) sweeps.

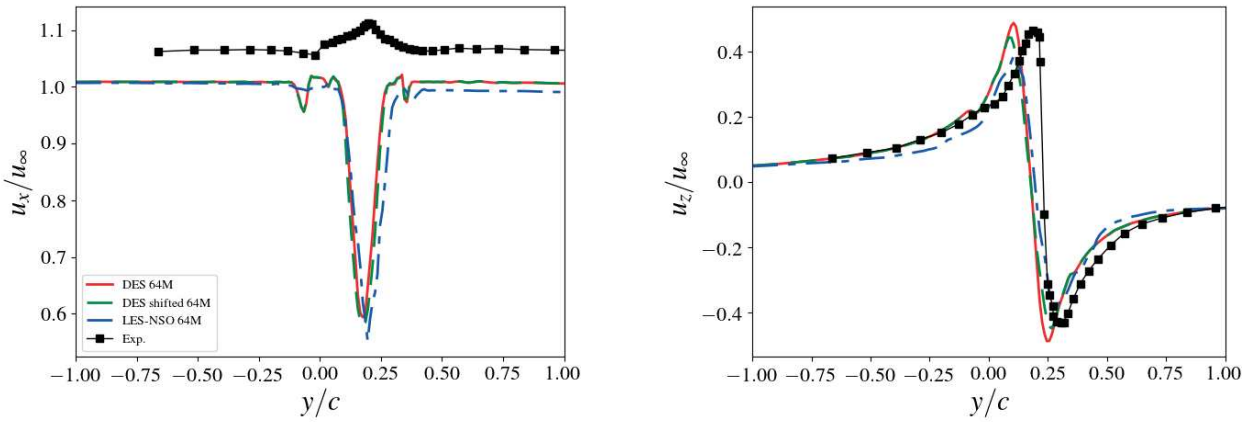

Figure 18: Velocities in the tip-vortex at $x/c = 5$ as function of spanwise direction. Streamwise velocity (left). Vertical velocity (right).

data, field slices in the downstream wake region were taken for the last twenty simulation time steps. These were then temporally averaged to provide the point-wise field averages in each slice. The tip-vortex core was determined by finding the minimum pressure location in each averaged slice. Finally, the downstream and vertical velocities were extracted in the spanwise directions across the vortex core. A similar temporal averaging procedure was followed for post-processing wing surface data. For the *grid07conformal03lowAR* mesh, these simulations typically ran for 30 hours on 2K cores reaching $t = 0.068$s.

Figure 18 compares the experimental data from [3] with three different simulations using the same mesh (*grid07conformal03lowAR*) but different turbulence models: SST-DES (one with shifted quadrature rules and one without, see discussion in Section 3.2.2) and a LES with NSO. The coordinate-system origin used in these figures is at the leading edge of the tip of the wing. At this resolution, there is no significant difference between the three different simulations. All three simulations indicate a large x-direction velocity deficit (instead of an acceleration) in the vortex core. This discrepancy in u_x with the experimental data may be due to an over-dissipative turbulence model in the wake, as described in [6]. Additionally, for the high resolution mesh (*grid07conformal04lowAR*), the resolution was increased in the tip-vortex block but not in the unstructured mesh region between the wing and the start of the tip-vortex block. Therefore, the mesh resolution across the vortex core may also be under resolved. While there is a small difference in the vortex core location, Figure 18 shows that there is good agreement between the vertical velocity distribution and the experimental data at this location.

Additionally, we performed a mesh refinement study using the SST-DES turbulence model and the *grid07conformal03lowAR* and *grid07conformal04lowAR* meshes described in Table 1. Figure 19 shows the downstream and vertical velocity distributions for which both models used the SST-DES turbulence model. There is little difference between the two mesh resolutions for the SST-DES model. Finally, Figure 20 shows the pressure-coefficient distribution at two blade locations, near the wing tip and at the wing midpoint, using both the *grid07conformal03lowAR* and *grid07conformal04lowAR* meshes. All models predict similar wing surface pressure coefficients across most of the wing, though the NSO model is

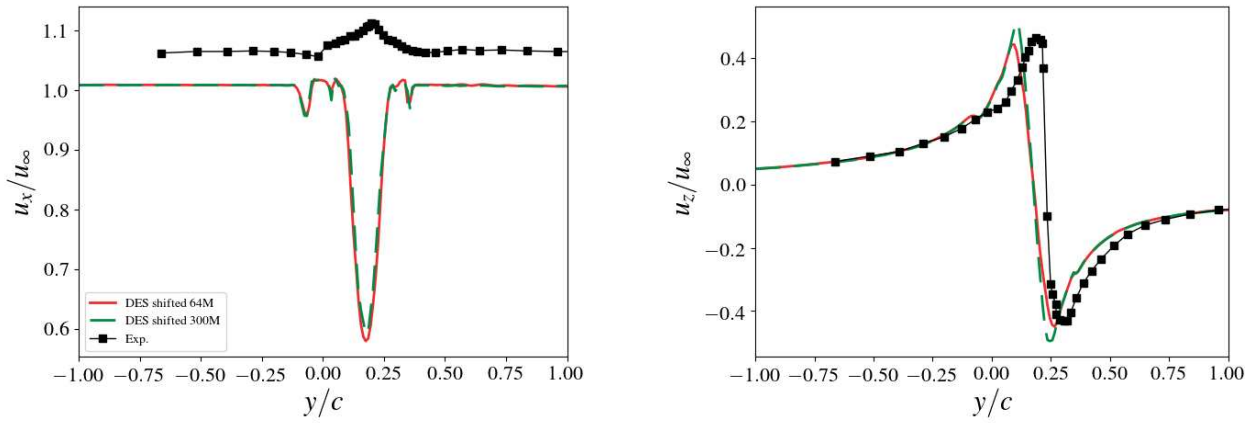

Figure 19: Velocity distributions in the tip-vortex at $x/c = 5$ as function of the spanwise direction. Streamwise velocity (left). Vertical velocity (right).

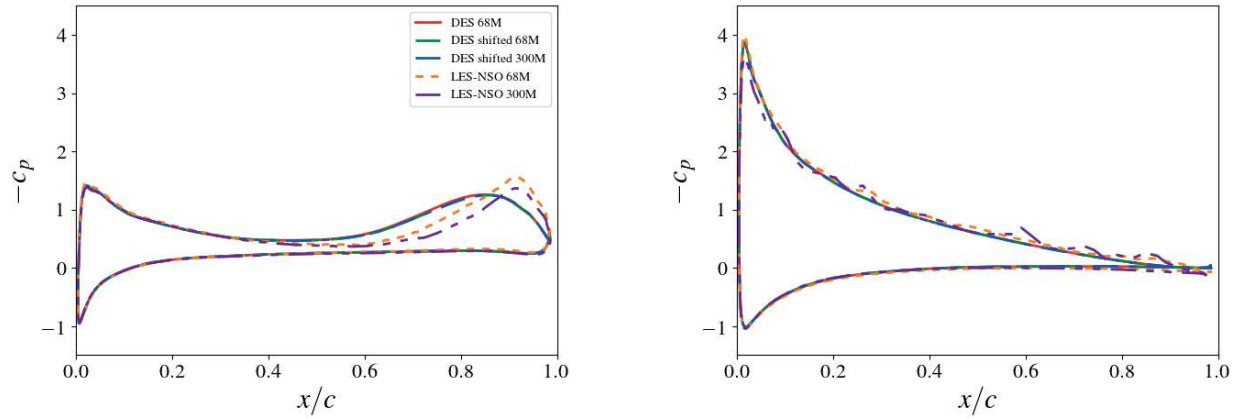

Figure 20: Negative of the wing pressure coefficient, C_p, near the wing tip $y/c = 0.0198$ (left) and near the wing midpoint $y/c = 1.6797$ (right).

slightly different at the trailing edge of the wing tip.

4 Summary

Our ALCC allocaiton was put to good use in demonstrating the ability of the Nalu CFD code to simulate atmospheric flows and complex geometries relevant to wind energy simulations. In particular, we performed a series of neutral atmospheric boundary layers to establish baseline timings and solver scaling performance on Cori. We also successfully simulated a finite-length blade (the McAlister-Takahashi wing), that has extensive validation data from an established wind-tunnel experiment. We established best-performing linear-solver settings under MPI-only communication by exploring multiple solver/preconditioner combinations and we established further baseline timing and scaling results on Haswell processors for mixed-element meshes, against which we can compare future improvements. Importantly,

we are moving to MPI+OpenMP parallelism in the next phase of the ExaWind project, and these timing and scaling data will be important for performance quantification. Finally, we also established baseline validity of our detached-eddy simulation, the improvement of which is a focus of future A2e and ExaWind project elements.

References

[1] M. Heroux, R. Bartlett, V. Howle, R. Hoekstra, J. Hu, T. Kolda, R. Lehoucq, K. Long, R. Pawlowski, E. Phipps, A. Salinger, H. Thornquist, R. Tuminaro, J. Willenbring, and A. Williams. An overview of Trilinos. Technical Report SAND2003-2927, Sandia National Laboratories, 2003.

[2] H. C. Edwards, A. Williams, G. Sjaardema, D. Baur, and W. Cochrant. SIERRA Toolkit computational mesh conceptual model. Technical Report SAND2010-1192, Sandia National Laboratories, 2010.

[3] K. W. McAlister and R. K. Takahashi. NACA 0015 wing pressure and trailing vortex measurements. Technical Report NASA-A-91056, National Aeronautics and Space Administration, AMES Research Center, Moffett Field, CA, 1991.

[4] F. R. Menter, M. Kuntz, and R. Langtry. Ten years of industrial experience with the SST turbulence model. In K. Hanjalic, Y. Nagano, and M. Tummers, editors, *Turbulence, Heat and Mass Transfer 4*, pages 625–632. Begell House, Inc., 2003.

[5] D.M. Driver and H.L. Seegmiller. Features of a reattaching turbulent shear layer in divergent channel flow. *AIAA Journal*, 23, 1985.

[6] Chao-Tsung Hsiao and Laura L. Pauley. Numerical study of the steady-state tip vortex flow over a finite-span hydrofoil. *J. Fluids Eng.*, 120(2):345, 1998.

A MueLU-AMG parameter settings

A.1 MueLu-AMG parameter settings for the Additive-Schwarz ILU(0) smoother

```
<ParameterList name="MueLu">
  <Parameter        name="verbosity"                          type="string"   value="low"/>
  <Parameter        name="coarse: max size"                   type="int"      value="2000"/>
  <Parameter        name="max levels"                         type="int"      value="6"/>
  <Parameter        name="transpose: use implicit"            type="bool"     value="true"/>
  <Parameter        name="smoother: type"                     type="string"   value="schwarz"/>
  <ParameterList    name="smoother: params">
     <Parameter     name="schwarz: num iterations"            type="int"      value="1"/>
     <Parameter     name="schwarz: overlap level"             type="int"      value="0"/>
     <Parameter     name="inner preconditioner name"          type="string"   value="RILUK"/>
     <ParameterList    name="inner preconditioner parameters">
        <Parameter     name="fact: iluk level-of-fill"          type="int"      value="0"/>
```

```
    </ParameterList>
  </ParameterList>
  <Parameter        name="aggregation: type"                type="string"   value="uncoupled"/>
  <Parameter        name="aggregation: drop tol"            type="double"   value="0.005"/>
  <Parameter        name="repartition: enable"              type="bool"     value="true"/>
  <Parameter        name="repartition: min rows per proc"   type="int"      value="1000"/>
  <Parameter        name="repartition: start level"         type="int"      value="1"/>
  <Parameter        name="repartition: max imbalance"       type="double"   value="1.327"/>
  <Parameter        name="repartition: partitioner"         type="string"   value="zoltan2"/>
</ParameterList>
```

A.2 MueLu-AMG parameter settings for the Additive-Schwarz ILUT(1) smoother

```
  <Parameter        name="verbosity"                        type="string"   value="low"/>
  <Parameter        name="coarse: max size"                 type="int"      value="2000"/>
  <Parameter        name="max levels"                       type="int"      value="5"/>
  <Parameter        name="transpose: use implicit"          type="bool"     value="true"/>
  <Parameter        name="smoother: type"                   type="string"   value="schwarz"/>
  <ParameterList    name="smoother: params">
     <Parameter     name="schwarz: num iterations"          type="int"      value="1"/>
     <Parameter     name="schwarz: overlap level"           type="int"      value="0"/>
     <Parameter     name="inner preconditioner name"        type="string"   value="ILUT"/>
     <ParameterList    name="inner preconditioner parameters">
        <Parameter     name="fact: ilut level-of-fill"      type="int"      value="1"/>
        <Parameter     name="fact: drop tolerance"          type="double"   value="10.0"/>
     </ParameterList>
  </ParameterList>
  <Parameter        name="aggregation: type"                type="string"   value="uncoupled"/>
  <Parameter        name="aggregation: drop tol"            type="double"   value="0.005"/>
  <Parameter        name="aggregation: drop scheme"         type="string"   value="distance laplacian"/
  <Parameter        name="repartition: enable"              type="bool"     value="true"/>
  <Parameter        name="repartition: min rows per proc"   type="int"      value="1000"/>
  <Parameter        name="repartition: start level"         type="int"      value="1"/>
  <Parameter        name="repartition: max imbalance"       type="double"   value="1.327"/>
  <Parameter        name="repartition: partitioner"         type="string"   value="zoltan2"/>
</ParameterList>
```

A.3 MueLu-AMG parameter settings for the Gauss-Seidel-L1 smoother

```
<ParameterList name="MueLu">
  <Parameter        name="verbosity"                        type="string"   value="low"/>
  <Parameter        name="coarse: max size"                 type="int"      value="2000"/>
  <Parameter        name="max levels"                       type="int"      value="6"/>
  <Parameter        name="transpose: use implicit"          type="bool"     value="true"/>
  <Parameter        name="smoother: type"                   type="string"   value="RELAXATION"/>
  <ParameterList    name="smoother: params">
     <Parameter name="relaxation: type"                     type="string"   value="Gauss-Seidel"/>
     <Parameter name="relaxation: fix tiny diagonal entries" type="bool"    value="true"/>
     <Parameter name="relaxation: use l1"                   type="bool"     value="true"/>
     <Parameter name="relaxation: sweeps"                   type="int"      value="4"/>
     <Parameter name="relaxation: damping factor"           type="double"   value="1.00"/>
```

```
  </ParameterList>
  <Parameter        name="aggregation: type"                type="string"   value="uncoupled"/>
  <Parameter        name="aggregation: drop tol"            type="double"   value="0.005"/>
  <Parameter        name="aggregation: drop scheme"         type="string"   value="distance laplacian"/
  <Parameter        name="repartition: enable"              type="bool"     value="true"/>
  <Parameter        name="repartition: min rows per proc"   type="int"      value="1000"/>
  <Parameter        name="repartition: start level"         type="int"      value="1"/>
  <Parameter        name="repartition: max imbalance"       type="double"   value="1.327"/>
  <Parameter        name="repartition: partitioner"         type="string"   value="zoltan2"/>
</ParameterList>
```

A.4 MueLu-AMG parameter settings for the Chebyshev smoother

```
<ParameterList name="MueLu">
  <Parameter        name="verbosity"                        type="string"   value="none"/>
  <Parameter        name="coarse: max size"                 type="int"      value="2000"/>
  <Parameter        name="max levels"                       type="int"      value="6"/>
  <Parameter        name="transpose: use implicit"          type="bool"     value="true"/>
  <Parameter        name="smoother: type"                   type="string"   value="CHEBYSHEV"/>
  <ParameterList    name="smoother: params">
     <Parameter name="chebyshev: degree"                    type="int"      value="4"/>
     <Parameter name="chebyshev: ratio eigenvalue"          type="double"   value="20"/>
     <Parameter name="chebyshev: min eigenvalue"            type="double"   value="1.0"/>
     <Parameter name="chebyshev: zero starting solution"    type="bool"     value="true"/>
     <Parameter name="chebyshev: eigenvalue max iterations" type="int"      value="15"/>
  </ParameterList>
  <Parameter        name="aggregation: type"                type="string"   value="uncoupled"/>
  <Parameter        name="aggregation: drop tol"            type="double"   value="0.005"/>
  <Parameter        name="aggregation: drop scheme"         type="string"   value="distance laplacian"/
  <Parameter        name="repartition: enable"              type="bool"     value="true"/>
  <Parameter        name="repartition: min rows per proc"   type="int"      value="1000"/>
  <Parameter        name="repartition: start level"         type="int"      value="2"/>
  <Parameter        name="repartition: max imbalance"       type="double"   value="1.327"/>
  <Parameter        name="repartition: partitioner"         type="string"   value="zoltan2"/>
</ParameterList>
```