SAND2017-10354J

Design-order, non-conformal low-Mach fluid algorithms using a hybrid CVFEM/DG approach

Stefan P. Domino

Sandia National Laboratories Computational Thermal and Fluid Mechanics, 1541

P.O. Box 5800 MS 0828
Albuquerque, New Mexico, 87109-0828

Abstract

A hybrid, design-order sliding mesh algorithm, which uses a control volume finite element method (CVFEM), in conjuction with a discontinuous Galerkin (DG) approach at non-conformal interfaces, is outlined in the context of a low-Mach fluid dynamics equation set. This novel hybrid DG approach is also demonstrated to be compatible with a classic edge-based vertex centered (EBVC) scheme. For the CVFEM, element polynomial, P, promotion is used to extend the low-order P=1 CVFEM method to higher-order, i.e., P=2. An equal-order low-Mach pressure-stabilized methodology, with emphasis on the non-conformal interface boundary condition, is presented. A fully implict matrix solver approach that accounts for the full stencil connectivity across the non-conformal interface is employed. A complete suite of formal verification studies using the method of manufactured solutions (MMS) is performed to verify the order of accuracy of the underlying methodology. The chosen suite of analytical verification cases range from a simple steady diffusion system to a traveling viscous vortex accross mixed-order non-conformal interfaces. Results from all verification studies demonstrate either second- or third-order spatial accuracy and, for transient solutions, second-order temporal accuracy. Significant accuracy gains in manufactured solution error norms are noted even with modest promotion of the underlying polynomial order. The paper also demonstrates the

Email address: spdomin@sandia.gov (Stefan P. Domino)

CVFEM/DG methodology on two production-like simulation cases that include an inner block subjected to solid rotation, i.e., each of the simulations include a sliding mesh, non-conformal interface. The first production case presented is a turbulent flow past a high-rate-of-rotation cube (Re, 4000; RPM, 3600) on like and mixed-order polynomial interfaces. The final simulation case is a full-scale Vestas V27 225 kw wind turbine (tower and nacelle omitted) in which a hybrid topology, low-order mesh is used. Both production simulations provide confidence in the underlying capability and demonstrate the viability of this hybrid method for deployment towards high-fidelity wind energy validation and analysis.

Keywords: control volume finite element, discontinous Galerkin, higher-order, sliding mesh, non-conformal

1. Introduction

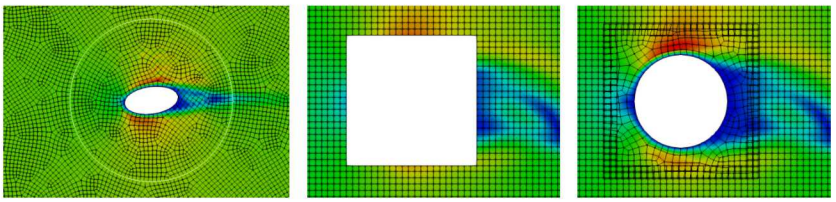
Many applications in computational fluid mechanics involve predicting flow with moving boundaries. Such requirements arise in, turbomachinery, rotorcraft, and, of interest to this work, wind energy applications. Recently, the ExaWind project [1] was funded by the Exascale Computing Project [2] (ECP) to drive high performance computing (HPC) on next generation platforms (NGP) to enable predictive simulations and understanding of the complex flow in wind farms, and to expose pathways to lower cost of wind energy. The ExaWind project goal is to support the simulation of O(100) multi-MW wind turbines sited within a 10 km x 10 km domain with possible complex terrain. It is anticipated that billions of computational elements will be required in addition to exascale computing (defined by 10^{18} calculations per second). Other wind energy use-cases of interest include multi-physics simulations, e.g., turbulent flow with fluid structure interactions, for non-design extreme gust events. The primary goal of this paper is to outline a surface-based, design-order, non-conformal interface boundary approach suitable for the low-Mach large-eddy simulation (LES) application space to support the ExaWind project objectives.

Demonstration of this method on both low- and high-order topologies are of direct interest to the ECP-funded ExaWind project since it is anticipated that higher-order regions of the mesh to capture wake vortex dynamics will be required.

In wind energy applications, the following two discretization technologies have emerged as viable approaches to support such applications: sliding mesh [3] and overset [4]. In sliding mesh applications, a prescribed mesh interface is defined to manage the solid body movement/rotation. Specifically, a set of blocks containing the solid body rotates, thereby, giving rise to a set of interfaces that slide with respect to each other. In an overset method, see [5] for a classic description of the technology, a background mesh is intersected with an overset mesh that represents the solid geometry of interest. Advanced hole cutting algorithms are employed to define a set of doner and receiver/receptor objects that are used to reconstruct fields on each of the overlapping/fringe mesh definions [6].

Figure 1 provides a graphical example of the two methods under consideration within the ExaWind project. In the non-conformal sliding mesh implementation, see (a), a prescribed, shared interface is defined. In this rotating ellipse flow example, the inner mesh block is rotating and creates a dual surface interaction that is non-conformal in nature. In fact, even in static mesh applications the mesh can be non-conformal should the analysis desire disparate resolutions between the meshed blocks. In an overset method, a hole cutting approach is used to mark inactive elements, see (b), in addition to a defined overlap region, see (c).

Although the ExaWind project is exploring both sliding mesh and overset technologies in the context of low-Mach fluids flow, the objective of this particular paper is to deploy a design-order sliding mesh scheme that combines two core discretization methodologies: the control volume finite element method [7] and a discontinuous Galerkin [8] approach at the non-conformal interface. Here, the most rigorous definition of design-order is the demonstration of a numerical method to be $O(\Delta x)^{P+1}$ for discretization basis polynomial orders of order,



(a) Sliding mesh. (b) Overset hole-cut region. (c) Overset overlap region.

Figure 1: Overview of the conceptual differences between non-conformal and overset methodologies. In the sliding mesh technology, (a), a set of surfaces are defined that form a non-conformal interface while in the overset method, a background mesh is intersected with an overset mesh to remove inactive elements, (b); overlap regions exist and the combined mesh is shown in (c).

P. For prescribed sliding mesh revolution use-cases, only a small subset of the mesh connectivity is changing. Therefore, it is anticipated that exploiting the prescribed connectivity change can lead to reductions in both matrix re-initialization and preconditioner setup costs since the reinitializtion procedure can focus on the known matrix rows and columns that are expected to change.

If one defines a success metric as deploying a stable, accurate, *and* design-order low-Mach fully implicit non-conformal method, challenges remain. For example, in the original work of Gartling, [9], a two-dimensional finite element method (FEM) that employed a multi-point constraint algorithm to manage the non-conformal interface was outlined. This procedure followed a very similar approach that has been deployed for a thermal heat conduction tied-contact application space (see [10] for a foundational review of non-conformal algorithms that have been used in thermal applications). Although such constraint approaches are not design-order due to the one-sided discretization stencil, monolithic sliding mesh laminar fluids applications were successfully demonstrated. Recently, sliding mesh algorithms in the context of two-dimensional spectral high-order finite elements has been proposed [11] using a mortar method with curved elements. This method was proven to be both design-order (through rigorous code verification), and viable for a variety of simple canonical flows with sliding mesh,

e.g., laminar flow-past an ellipse and a stirred tank. In this scheme, the spectral higher-order methodology was deployed for a two-dimensional compressible, explicit fluid mechanics equation set. In later works, Zhang, [12], the method was extended from two-dimensioanl to three-dimensional, while remaining in the explicit and accoustically compressible formulation.

In the finite volume discretization application space, extrusion-based approaches are widely used. For example, in McNaughton *et al*. [13], a low-Mach cell-centered finite volume approach used an extruded, extrapolated approach. In Blades [14], a compressible, explicit, edge-based vertex-centered (EBVC) scheme using extrusion concepts was developed. In both of the aforementioned discretization approaches, elements at the non-conformal interface are projected (or extruded) into the opposing domain. The new degree of freedom locations of the fictitious/extruded mesh points are reconstructed in a manner simular to overset methods. In low-order finite volume schemes, the projected nodal gradient is generally used to extrapolate values to the fictitious point based on the distance vector between the closest opposing cell-center and the new fictitious point. Such approaches follow flux reconstruction techniques where inclusion of full matrix Jacobian entries may be problematic as the matrix sensitivities for the projected nodal gradient are generally ignored. Reconstruction techniques can also use an underlying basis interpolation procedure. For vertex-centered finite volume (VCFV) techniques, the element in which the point resides can be used to construct the fictitious values using the underlying elemental basis. In cell-centered approaches, a least-square polynomial reconstruction technique analogous to overset methods [15] can be used. Again, matrix contributions can be difficult, especially if the underlying basis for interpolation is greater than the underlying basis for discretization.

The extrusion-based approach was used in the wind-energy uncertainty quantification (UQ) work of Eldred *et al*. [16] where Hermite polynomial reconstruction was used to retain a design-order method. In this study, a fully implicit low-Mach EBVC scheme was used to perform vertical axis wind turbine (VAWT) simulations in which a large number of idealized gust locations

and wind amplitudes were realized. Unfortunately, extrusion-based approaches for vertex-centered finite volume schemes can contain corner nodes, i.e., sharp features, at the non-conformal interface that are problematic as compared to cell-centered approaches. In fact, in both examples cited above [14, 16], smooth non-conformal surface definitions were in use to avoid the problematic definition of extrusion directions at corners. Such extrusion-based complexities for vertex-based methods are simply because a closed dual mesh is required due to the element-node stencil connectivity. For a cell-centered scheme, the connectivity is defined by element-face pairs. At corners, no extrusion complexity is found in a cell-centered approach whereas mesh quality of the extruded mesh for a vertex-based approach can be poor. Therefore, when using a VCFV scheme, a surface-based approach that avoids an extrusion-based step is greatly desired.

Building upon the work of Carnes [10], a FEM/DG methodology for non-conformal surfaces was prototyped in the context of a monolithic low-Mach fluids algorithm [17]. In later works, [18], the procedure outlined by Domino [17] was extended for use on an explicit, compressible FEM-based formulation. Several explicit, compressible use cases, all with static geometries, were outlined. Therefore, the general mixed-method technique has demonstrated over a range of fluid mechanics applications.

In this paper, the FEM/DG-based methodology is extended to VCFV schemes and is implemented, verified, and tested within the open-source Nalu code base [19]. This code base is actively being developed by a multi-lab, multi-institutional team [1] supported by both the ECP and Atmosphere to Electrons [20] funding portfolios to drive high-fidelity wind farm predictions by resolving the turbine geometry and motion directly. When lower-fidelity wind farm blade modeling is desired, actuator line [21] methods are used.

The Nalu application code leverages Sandia infrastructures including both the unstructured mesh interface, i.e., the Sierra Toolkit [22], and the sparse linear solver package Trilinos [23]. Both of these infrastructures are funded by contributions from the Advanced Simulation and Computing (ASC) program and the Advanced Technology Development and Mitigation (ATDM) program.

Both programs are National Nuclear Security Administration (NNSA) initiatives to support science-based stockpile stewardship. Nalu has been used in a variety of high-fidelity LES studies [24] in addition to more recent structural uncertainty perturbation techniques for LES [25]. Scaling of this code base has been demonstrated on billions of elements using $O(100)$ thousand cores [26].

In section 2, the basic principles of the non-conformal DG method are provided in the context of a variational FEM formulation. Extending the method from FEM-based to VCFV-based methods is also presented in section 2. In section 3, a low-Mach VCFV algorithm using the DG non-conformal interface concept is outlined. Section 4 presents the non-conformal algorithm implementation details from the perspective of required parallel searches and underlying data structures required for matrix assembly. The effect of reduced matrix contributions is outlined for the low-Mach application space. The paper overviews four high quality verification studies in section 5. Finally, a suite production turbulent applications, each of which include a sliding mesh interface, is provided in section 6.

2. Algorithmic Description for Simple Diffusion

The original non-conformal interface has its roots in the Sandia National Laboratories Sierra/Thermal application code, [10], and is now briefly overviewed. After this overview, the method is extended to a vertex-based finite volume approach using a specific definition of the chosen test function.

Consider two domains, Ω^A and Ω^B, which have a common interface, Γ^{AB}, and a set of interfaces not in common, $\Gamma \backslash \Gamma^{AB}$, see Figure 2. Each domain has a set of outwardly pointing normals n_j^A and n_j^B. In this cartoon, the interface is well resolved via a linear description of the interface, although in practice, curved surfaces are frequently present. Also, consider that the solution of the time-dependent heat conduction equation is to be solved in both domains,

$$\rho C_p \frac{\partial T}{\partial t} + \frac{\partial q_j}{\partial x_j} = S, \tag{1}$$

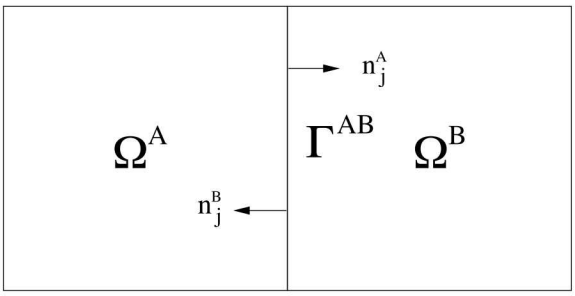


Figure 2: Two-block example with two blocks, Ω^A and Ω^B, and one common surface definition, Γ^{AB}.

where ρ is the density and C_p is the specific heat. A general diffusional flux vector, q_j for a general scalar ϕ can be written as,

$$q_j = -\rho D \frac{\partial \phi}{\partial x_j}. \tag{2}$$

Above, the diffusional flux coefficient, ρD, is interpreted as the thermal conductivity, k, while the scalar ϕ is simply the temperature, T,

$$q_j = -k \frac{\partial T}{\partial x_j}. \tag{3}$$

For generality of the equation set, a source term, S, is also included.

The numerical non-conformal interface description follows the unified approach of Arnold *et al.* [8] in which numerical fluxes for DG interior penalty (IP) scheme are presented. The variational statement for both domain A and B is obtained by multiplying Equation 1 by an arbitrary test function w that is continuous within each domain, however, can be discontinuous across the shared interface, Γ^{AB}. As shown below, the diffusive flux contribution is integrated by parts.

$$\begin{aligned} \int_{\Omega^A} w \left(\rho C_p \frac{\partial T}{\partial t} - S \right) d\Omega \quad &- \quad \int_{\Omega^A} \frac{\partial w}{\partial x_j} q_j d\Omega \\ &+ \quad \int_{\Gamma \setminus \Gamma^{AB}} w q_j n_j d\Gamma \\ &+ \quad \int_{\Gamma^{AB}} w \hat{T}(A,B) d\Gamma \end{aligned}$$

$$+ \quad \int_{\Gamma^{AB}} \frac{\partial w}{\partial x_j} n_j^A \lambda_{IP}^{AB} (T^A - T^B) d\Gamma. \tag{4}$$

$$\begin{aligned} \int_{\Omega_B} w \left(\frac{\partial \rho T}{\partial t} - S \right) d\Omega \quad &- \quad \int_{\Omega_B} \frac{\partial w}{\partial x_j} q_j d\Omega \\ &+ \quad \int_{\Gamma \setminus \Gamma_{BA}} w q_j n_j d\Gamma \\ &+ \quad \int_{\Gamma_{BA}} w \hat{T}(B, A) d\Gamma \\ &+ \quad \int_{\Gamma_{BA}} \frac{\partial w}{\partial x_j} n_j^B \lambda_{IP}^{BA} \left(T^B - T^A \right) \big) \, d\Gamma. \end{aligned} \tag{5}$$

Adding both equations together reverts to the classic DG jump notation [8]. In this paper, the numerical flux notation is consistently followed. The general numerical flux for each block, $\hat{T}(\alpha, \beta)$, is

$$\hat{T}(\alpha, \beta) = \frac{1}{2}(q_j^\alpha n_j^\alpha - q_j^\beta n_j^\beta) + \lambda_T^{\alpha\beta}(T^\alpha - T^\beta). \tag{6}$$

In the above notation, α is the current quadrature location on Γ_α (at the standard gauss point locations) while β is the opposing quadrature location at given projected location residing on Γ_β. The precise definition of these quadrature points will be provided in section 4. Above, the penalty parameters $\lambda_T^{\alpha\beta}$ are computed to be averages of diffusive flux coefficients and respective mesh length scales,

$$\lambda_T^{\alpha\beta} = \frac{1}{2}(\frac{k^\alpha}{L^\alpha} + \frac{k^\beta}{L^\beta}). \tag{7}$$

Ideally, the jump in mesh spacing is desired to be near unity, however, as future verification studies demonstrate, see section 5, demonstrate design-order convergence even with mesh spacing jumps as high as 1.6.

As noted in the Arnold paper, the interior penalty method has been proven to be consistent, stable, and design-order. This description is the basis of the chosen path forward for low-Mach non-conformal interface approaches. Note that the fifth term in each of the above two equations includes a gradient of the test function which adds one layer of elements at the non-conformal interface.

2.1. Extension from FEM to Vertex-based schemes

The class of discretization methods of interest to the ExaWind project fall within vertex-centered finite volume methods, see [27] for a foundational textbook that describes several common discretization approaches. The control volume finite element method, [7], defines a dual mesh constructed within each element. See Figure 3 for both low-order and higher-order node and dual volume rules (shown for the quadralateral topology). Integration points are defined at the subcontrol surfaces (for flux calculations) and subcontrol volumes (for source and time contributions). To recover a low-order edge-based vertex-centered (EBVC) method, dual volume subcontrol area vectors are assembled to the edges of the low-order elements while dual nodal volumes are assembled to the nodes of the elements. For a recent paper on extension of CVFEM to higher-order via polynomial promotion, the reader is referred to [28]. For an example usage of these low- and higher-order schemes, in the context of a LES validation study, see [24]. Finally, for detailed information on the full CVFEM method, the reader is referred to either the formal Nalu theory manual [29] or the online, open-source documentation, [30].

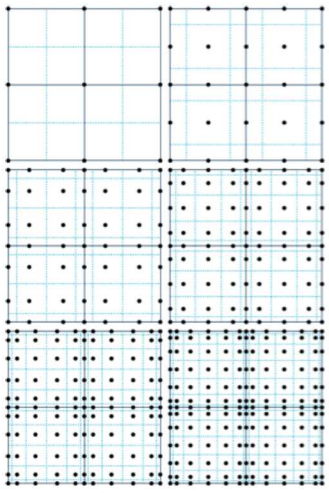


Figure 3: Polynomial promotion for a canonical CVFEM quad element patch from P=1 (upper left) to P=6 (lower right).

In vertex-centered finite volume techniques, the previously defined test function, w, is chosen to be a piecewise-constant (see Swaminanthan, *et al.* [31], and also the work of Martinez [32]). This chosen Heaviside functional form for the test function is unity within the dual volume and zero outside of it. A key property of this function is that the gradient of the piecewise constant test function is a distribution of delta functions on the subcontrol volume surface,

$$\frac{\partial w}{\partial x_j} = -n_j \delta(x_j - x_j^{ip}), \tag{8}$$

where n_j is the outward normal on that boundary and x_j^{ip} is the subcontrol volume surface integration point location. Therefore, the CVFEM and EBVC scheme can be categorized as a Petrov-Galerkin methodology as the test function is different from the underlying interpolation basis. Note that an important aspect to this formulation is that accurate quadrature is required over the dual volume. As there may be multiple dual volume definitions within a higher-order element, the number of required quadrature points for higher-order CVFEM is larger than a like polynomial order in FEM. In this proof-of-concept paper, simulation studies are limited to a polynomial order of two.

Substitution of the above piecewise-constant test function definition, see Equation 8, into Equation 4 and Equation 5 yields the following standard finite-volume form:

$$\int_{\Omega^A} \left(\rho C_p \frac{\partial T}{\partial t} - S \right) d\Omega + \int_{\Gamma \setminus \Gamma^{AB}} q_j n_j d\Gamma + \int_{\Gamma^{AB}} \hat{T}(A,B) d\Gamma = 0, \tag{9}$$

$$\int_{\Omega^B} \left(\rho C_p \frac{\partial T}{\partial t} - S \right) d\Omega + \int_{\Gamma \setminus \Gamma^{BA}} q_j n_j d\Gamma + \int_{\Gamma^{BA}} \hat{T}(B,A) d\Gamma = 0. \tag{10}$$

In the above equation set, the numerical flux, $\hat{T}(\alpha, \beta)$ remains the same form of Equation 6. Note that due to the piecewise-constant test function, the standard IP that includes the $\frac{\partial w}{\partial x_j}$ scaling vanishes.

As with FEM, a CVFEM approach defines an underlying basis that can be used to, for example, interpolate a variable, ϕ, within the element to an

integration point, ip by

$$\phi_{ip} = \sum N_k^{ip} \phi_k. \tag{11}$$

Gradients at integration points can be obtained by taking the derivative of the above equation,

$$\frac{\partial \phi_{ip}}{\partial x_j} = \sum \frac{\partial N_{j,k}^{ip}}{\partial x_j} \phi_k. \tag{12}$$

For the edge-based approach, Equation 11 can be used with the element represented as a two-node edge. This interpolation reverts to a simple weighting, i.e., $\frac{1}{2}$, between the two nodes that define the edge. Interpolation and derivatives at the non-conformal interface, which are required for the DG numerical flux shown in Equation 9 and 10, follow Equation(s) 11 and 12, respectively.

Quadrature point locations at the non-conformal interface for CVFEM, assuming an isoparametric range of $(-1 : +1)$, are $(-\frac{1}{2} : +\frac{1}{2})$. For the edge-based scheme, the natural locations are the Gauss-Labatto locations of $(-1 : +1)$.

3. Equal-order Pressure Stabilized low-Mach Fluids Algorithm

This section describes the low-Mach fluids algorithm. A low-Mach physics set can be characterized as a variable density flow whereby the acoustic pressure waves have been filtered [33]. The equation of state for density is, therefore, written in terms of the constant in space thermodynamic pressure. The low-Mach modified momentum equation includes a pressure gradient contribution that represents a perturbation about the thermodynamic pressure (also known as the motion pressure). For more details on the low-Mach approach, the reader is referred to [33, 34, 29]. The low-Mach method used in this paper can be classified as an equal-order, pressure-stabilized, approximate pressure projection-based scheme [35]. In addition to a brief overview of the numerical method, the transport equations solved for for momentum, continuity, and scalar transport in the presence of the non-conformal DG interface will be provided. For more details of the core algorithm, the reader is again referred to the open-source Nalu

theory manual [29] and a recent description of the extension from a low-order CVFEM to higher-order [28].

Pressure stabilization is simply provided by adding a scaled fine-scale momentum residual to the continuity equation in a manner equivalent to the FEM-based Pressure Stabilized Petrov-Galerkin (PSPG), which is adapted for a vertex-centered finite volume method, [31]. The fine-scale momentum residual is written in terms of a projected momentum residual evaluated at the quadrature points. Algebraic manipulation yields an expression that can be written in terms of a difference between an integration point pressure gradient and projected nodal gradient interpolated to the integration point,

$$\mathbf{R}(u_j) = \left(\frac{\partial p}{\partial x_j} - G_j p \right). \tag{13}$$

The above equation is derived simply by writing a fine-scale momentum equation at the integration points and using the nodal projected momentum residual to reconstruct the individual momentum terms. Although this method of pressure stabilization can be viewed as a Rhie and Chow interpolation procedure, [36], the method follows closely the formal class of stabilization methods known as local projection stabilization, see [37], which leverages other foundational work to extract the positive attributes from a range of discretization/stabilization approaches [38].

The projected nodal gradient, $G_j p$, is given by the following equation set:

$$\int_{\Omega^A} G_j p dV = \int_{\Gamma \setminus \Gamma^{AB}} p n_j d\Gamma + \int_{\Gamma^{AB}} \frac{1}{2} \left(p^A + p^B \right) n_j d\Gamma, \tag{14}$$

$$\int_{\Omega^B} G_j p dV = \int_{\Gamma \setminus \Gamma^{BA}} p n_j d\Gamma + \int_{\Gamma^{BA}} \frac{1}{2} \left(p^B + p^A \right) n_j d\Gamma. \tag{15}$$

The above equation set represents a L_2 minimization between the discontinuous elemental gradient operator and the nodal gradient sampled at the quadrature points. In a low-order method, $G_j p$ can be lumped, whereas in the higher-order method, an additional equation must be solved using a consistent mass matrix quadrature rule, i.e., evaluated at the subcontrol volume quadrature points.

The continuity equation system solved, which is coupled over all blocks in

the mesh, is as follows:

$$\int_{\Omega_A} \frac{\partial \rho}{\partial t} d\Omega + \int_{\Gamma \setminus \Gamma^{AB}} \rho \hat{u}_j n_j d\Gamma \quad + \quad \int_{\Gamma^{AB}} \hat{C}(A,B) d\Gamma = 0, \tag{16}$$

$$\int_{\Omega_B} \frac{\partial \rho}{\partial t} d\Omega + \int_{\Gamma \setminus \Gamma^{BA}} \rho \hat{u}_j n_j d\Gamma \quad + \quad \int_{\Gamma^{BA}} \hat{C}(B,A) d\Gamma = 0, \tag{17}$$

where, $\hat{u_j}$ represents the modified velocity that includes pressure stabilization,

$$\hat{u}_j = u_j - \frac{\Delta t}{\rho} \left(\frac{\partial p}{\partial x_j} - G_j p \right). \tag{18}$$

An integration point momentum field, m_j, is defined to include pressure stabilization,

$$m_j = \rho \hat{u}_j = \rho u_j - \Delta t \left(\frac{\partial p}{\partial x_j} - G_j p \right). \tag{19}$$

This integration point momentum expression can be defined for each non-conformal surface, α,

$$m_j^\alpha = \rho^\alpha u_j^\alpha - \Delta t \left(\frac{\partial p^\alpha}{\partial x_j} - G_j^\alpha p \right). \tag{20}$$

The numerical flux for the continuity equation, $\hat{C}(\alpha, \beta)$, is simply,

$$\hat{C}(\alpha,\beta) = \frac{1}{2}(m_j^\alpha n_j^\alpha - m_j^\beta n_j^\beta) + \lambda_C^{\alpha\beta}(p^\alpha - p^\beta) \tag{21}$$

Note that a mass flow rate can be defined for the α and β pair as a function of the continuity numerical flux,

$$\dot{m}^{\alpha,\beta} = \hat{C}(\alpha,\beta) d\Gamma^\alpha. \tag{22}$$

As with the standard mass flow rate expression, pressure stabilization is retained at the non-conformal interface and has been found to be critical (via numerical experiments) for the stability of the algorithm. Moreover, the mass flow rate, whether in the core interior algorithm or at the non-conformal interface, is not upwinded. The penalty parameter for the continuity equation is simply,

$$\lambda_C^{\alpha\beta} = \frac{\Delta t}{2} \left(\frac{1}{L^\alpha} + \frac{1}{L^\beta} \right) \tag{23}$$

Above, L^α and L^β are defined as a length scale in the direction normal to the surface. The shape function derivative operator is used to obtain the desired length scale, which is shown for L^α), is

$$L^\alpha = \left(\sum \frac{\partial N^\alpha_{j,k}}{\partial x_j} n^\alpha_j \right)^{-1}. \tag{24}$$

The discrete momentum equation system, including the contribution for the non-conformal interface, is

$$\begin{aligned} \int_\Omega^A \left(\frac{\partial \rho u_i}{\partial t} + G^A_j p - S_i \right) d\Omega &+ \int_{\Gamma \setminus \Gamma^{AB}} \left(\rho \hat{u}_j u_i - \tau_{ij} \right) n_j d\Gamma \\ &+ \int_{\Gamma^{AB}} \hat{M}_i(A,B) d\Gamma = 0, \end{aligned} \tag{25}$$

$$\begin{aligned} \int_\Omega^B \left(\frac{\partial \rho u_i}{\partial t} + G^B_j p - S_i \right) d\Omega &+ \int_{\Gamma \setminus \Gamma^{BA}} \left(\rho \hat{u}_j u_i - \tau_{ij} \right) n_j d\Gamma \\ &+ \int_{\Gamma^{BA}} \hat{M}_i(B,A) d\Gamma = 0. \end{aligned} \tag{26}$$

Above, the viscous stress tensor, τ_{ij}, is defined as

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \mu \frac{\partial u_k}{\partial x_k} \delta_{ij}. \tag{27}$$

Note that the pressure gradient, $G_j p$, and general source term, S_i, appear in the volumetric kernel. The numerical flux for momentum, $\hat{M}_i(\alpha, \beta)$, now includes an advection and diffusion contribution,

$$\begin{aligned} \hat{M}_i(\alpha,\beta) d\Gamma &= \frac{1}{2} (\tau^\alpha_{ij} n^\alpha_j - \tau^\beta_{ij} n^\beta_j) d\Gamma^\alpha + \lambda^{\alpha\beta}_i (u^\alpha_i - u^\beta_i) d\Gamma^\alpha \\ &+ \dot{m}^{\alpha\beta} \frac{(u^\alpha_i + u^\beta_i)}{2} + \eta \frac{|\dot{m}^{\alpha\beta}|}{2} (u^\alpha_i - u^\beta_i), \end{aligned} \tag{28}$$

Above, the penalty parameter, $\lambda^{\alpha\beta}_i$, follows that of the standard diffusion form of Equation 7 with the viscosity, μ, serving the role as the diffusional flux coefficient. The penalty parameter for momentum is shared over all components. The parameter η is added for additional upwind stabilization (when $\eta = 1$).

Finally, the set of physics to be modeled may include transport of auxiliary variables such as mixture fraction, Z, (used in variable density reacting flow)

or static enthalpy, h, which is used in non-isothermal applications. A general transport equation system for scalar ϕ can be written as,

$$\int_{\Omega^A}\left(\frac{\partial\rho\phi}{\partial t}-S_\phi\right)d\Omega+\int_{\Gamma\setminus\Gamma^{AB}}\left(\rho\hat{u}_j\phi+q_j\right)n_j d\Gamma+\int_{\Gamma^{AB}}\hat{Q}_i^\phi(A,B)d\Gamma=0, \quad (29)$$

$$\int_{\Omega^B}\left(\frac{\partial\rho\phi}{\partial t}-S_\phi\right)d\Omega+\int_{\Gamma\setminus\Gamma^{BA}}\left(\rho\hat{u}_j\phi-q_j\right)n_j d\Gamma+\int_{\Gamma^{BA}}\hat{Q}_i^\phi(B,A)d\Gamma=0. \quad (30)$$

The diffusional flux vector, q_j, again follows Equation 2 with the appropriate diffusional flux coefficient, $\frac{\mu}{Sc}$ for mixture fraction (Sc is the laminar Schmidt number) and $\frac{\mu}{Pr}$ for enthalpy (Pr is the laminar Prandtl number). The numerical flux for scalar ϕ follows the standard established form in the momentum equation,

$$\begin{aligned}\hat{Q}_\phi(\alpha,\beta)d\Gamma &= \frac{1}{2}(q_j^\alpha n_j^\alpha-q_j^\beta n_j^\beta)d\Gamma^\alpha+\lambda_\phi^{\alpha\beta}(\phi^\alpha-\phi^\beta)d\Gamma^\alpha \\ &+ \dot{m}^{\alpha\beta}\frac{(\phi^\alpha+\phi^\beta)}{2}+\eta\frac{|\dot{m}^{\alpha\beta}|}{2}(\phi^\alpha-\phi^\beta). \quad (31)\end{aligned}$$

Again, the penalty form for $\lambda_\phi^{\alpha\beta}$ follows the previously defined functional form with the appropriate diffusive flux coefficient.

The core algorithm used in this study splits the monolithic momentum and continuity equation set and follows an incremental pressure projection scheme. Specifically, momentum is solved using a provisional pressure gradient. The resulting velocity is used in the continuity equation to define the new pressure field. Finally, the velocity is projected into the space of known divergence based on the continuity constraint. Splitting errors are repaired via non-linear iterations over the full equation set to ensure that the splitting is exact. In this core algorithm, as with a monolithic scheme, only pressure stabilization errors remain. For more information on the core pressure projection scheme, the user is referred to either [35] or [29].

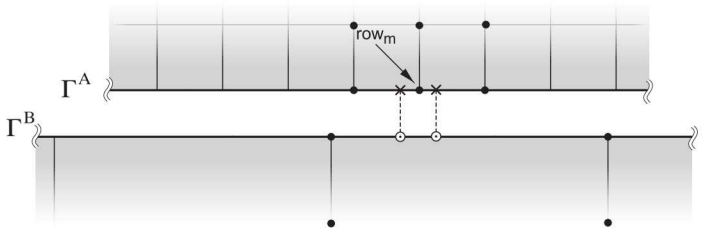
4. Non-conformal Algorithm Implementation Details

The implementation for the calculation of numerical fluxes at the non-conformal interface requires parallel searches to determine the set of opposing

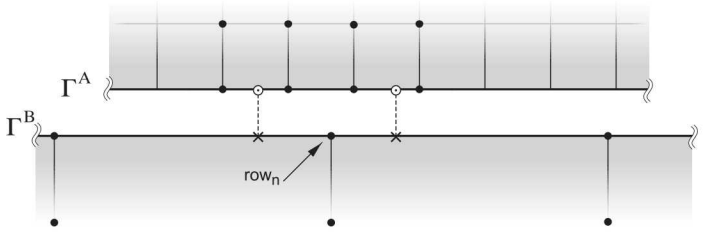
faces to the current set of integration points. Figure 4 graphically demonstrates the procedure in which current integration point values for a set of non-conformal surfaces are projected to the opposing exposed non-conformal face. The first step in the algorithm is to project standard integration points of block A to block B, see (a), while the second step is to project standard integration points from block B to block A, shown in Figure 4 (b). In the present algorithm, the integration point locations at the current surface are defined by either the CVFEM or EBVC exposed surface integration point locations. As currently implemented, the quadrature point locations for each surface pair interaction are not shared. It is only at a perfectly conformal interface that integration points match. Although this may have consequence for conservation statements, the order of accuracy, which will be established in section 5, remains design-order. Finally, the black nodes illustrated in Figure 4 define the complete discretization stencil.

A data structure, which can be iterated to provide flux and matrix contributions at the non-conformal interface, contains the mapping between the current integration points (along with the face in which this integration point resides) and the opposing surface. Opposing faces that are off processor rank are ghosted to the locally owned processor rank for the given current integration point. Volume elements, which are required for gradients, such as diffusion and stabilization, are extracted based on the mesh surface-to-element connectivity. Such relations are provided by the Sierra Toolkit.

As noted, the current integration point isoparametric quadrature points reside at the standard exposed face quadrature locations while the opposing quadrature points are dynamic. The opposing isoparametric coordinates are determined by the fine-scale parallel search in which the best opposing face is determined from the possible set of candidate opposing faces. Given this data structure, which includes the fully required mesh connectivity, numerical fluxes from the DG scheme can be computed. Implicit solves include the full connectivity for the current and opposing non-conformal contributions. As already noted, opposing faces my be off-processor to the locally owned node, thereby



(a) Current and opposite sides are A and B respectively.



(b) Current and opposite sides are B and A respectively.

Figure 4: Description of the numerical flux calculation for the DG algorithm whereby integration point locations are projected to the opposing interface.

requiring parallel ghosting. When mesh motion is active, the process of parallel search and matrix connectivity/initialization is processed at the top of each time step (note that the mesh connectivity is fixed over the nonlinear iteration). At present, the stencil is not artificially increased to allow for less frequent initialization steps.

4.1. Matrix Contributions

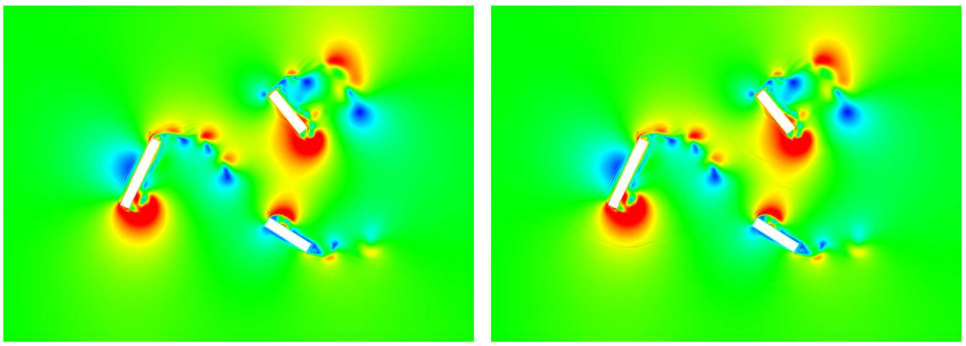
A fully implicit matrix contribution for the complete non-conformal interface discretization stencil, as already illustrated in Figure 4, is desired. In the general use case of interest, wind turbines, the non-conformal mesh interface is sliding. Specifically, rotation of one block may be present, e.g., simulations of a spinning cube, production wind turbine analysis, etc., in an analogous manner as already described in Figure 1, (a). Therefore, a dynamic stencil matrix assembly and solve interface is required. For example, if Ω_A is rotating, then the row entry for a given node on the Γ_A interface, row m from Figure 4 (a), will have column

entries from constant-in-time stencil (Ω_A) and a set of dynamic opposing column entries that reside on Γ_B.

In the current implementation, a parallel search, parallel ghosting and matrix re-initialization is performed to capture the complete illustrated matrix stencil for all current and opposing pairs in Figure 4. Such a connectivity is especially important for elliptic-based systems where the transfer of information across the non-conformal interface is crucial. An alternative to such an approach is to omit the opposing column entries in the Jacobian assembled. Removing of the opposing Jacobian entries will remove the requirement to re-initialize the matrix system (and preconditioners) at the cost of increased iterations to repair this operator split approach.

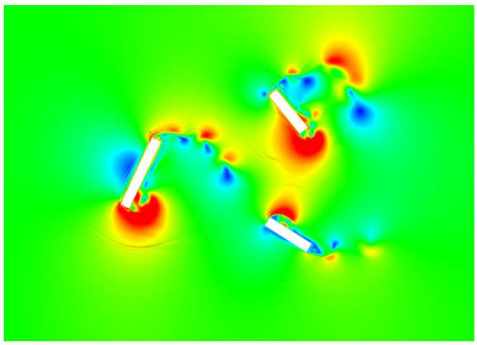
4.2. Reduced Matrix Entries

In this brief section, the effect of removing the opposing surface and opposing element matrix column entries for low-Mach applications is demonstrated. Figure 5, provides a representative low-Mach transient flow past three independently rotating blades (velocity magnitude shadings are shown). Specifically, the comparison in Figure 5 is among a full matrix contribution approach and the effect of removing opposing column entries for the elliptic Poisson solve in favor of increased inner iterations over the pressure Poisson assemble, solve, and update step. By adding extra inner iterations over the continuity solve, the artifical splitting of the domain at the non-conformal interface is repaired to avoid a poor pressure field being used within the velocity projection step. In this study, omitted matrix entries are repaired by inner iteration counts of four, see (b), and two, see (c). When matrix opposing stencil matrix entries are omitted without additional inner iterations, the simulation diverges. When using either four or two inner iterations the simulation remains stable. However, artifacts in the flow field can be seen at the non-conformal interface. Although artifacts in the flow field are somewhat mitigated at four inner iterations, the expense of the Poisson solve increases this particular simulation time by a factor greater than two as compared to the full matrix contribution implementation. As such,



(a) Full matrix entries.

(b) Reduced matrix entires; four inner iterations.



(c) Reduced matrix entries; two inner iterations.

Figure 5: Reduced matrix contributions and possible mitigation strategy through increased inner Poisson assemble and solver micro-iterations. Shown are velocity magnitude shadings. As inner iterations are reduced, artifacts at the sliding mesh interface are noted.

the choice of increased inner iterations as opposed to re-initializing the matrix system at each time step is a trade-space between the cost of re-initilization and the cost of additional inner iterations. In all simulations presented in this paper, the full up-to-date matrix stencil for all primitive variables is constructed and used in the linear solve.

5. Verification Cases

The objective of this section is to demonstrate the order of accuracy for the non-conformal mixed DG interface approach when using vertex-centered

schemes. In all cases, hexahedral mesh interfaces are used. Although the code supports a mixed topology arbitrary interface, e.g, hexahedral, tetrahedral, pyramid and wedges, this series of verification studies is limited to hexahedral topologies. Moreover, a general promotion of low- to higher-order is also supported in the code base. However, for this series of verification studies, only P=1 and P=2 methods are considered. In this verification section, the method of manufactured solutions, [39], is extensively used such that error norms can be directly computed from the analytical solution.

5.1. Steady Laplace/Diffusion MMS

The first objective of this study is to establish the formal order of accuracy for the diffusion operator and the projected nodal gradients. The manufactured solution for the temperature is given by

$$T(x, y, z) = T_o cos(a\pi x) cos(a\pi y) cos(a\pi z), \tag{32}$$

where a and T_o are both unity. For this case, the density, ρ and specific heat, C_p, are also a constant unity. Therefore, the temperature form from Equation 1 reduces to a standard Laplace system. Source terms for the manufactured solution are not included in this paper, however, can be found in the Nalu open-source code base that are exercised nightly with the Nalu regression test suite that resides under the Nalu code base [19].

Figure 6 provides the functional form for the manufactured scalar temperature solution provided in Equation 32. Also shown are the mesh lines for the unstructured hexahedral mesh topology. In this simulation study, two nonconformal blocks are joined at one interface physical location. The ratio of mesh length scales between the two blocks is 1.4. The mesh resolution for the P=1 mesh is uniformly refined to provide three meshes, R_0, R_1, and R_2. The higher-order mesh (P=2) simply promotes the respective P=1 meshes from a hex8 to a hex27 topology. Therefore, the number of elements for the P=1 R_0 mesh and the P=2 R_0 mesh are equal, while the node count will be approximately eight

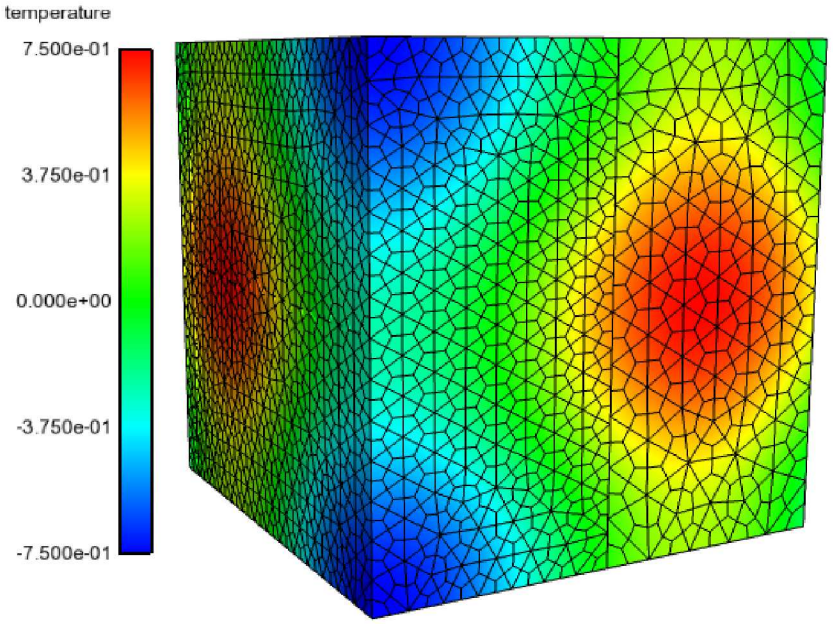
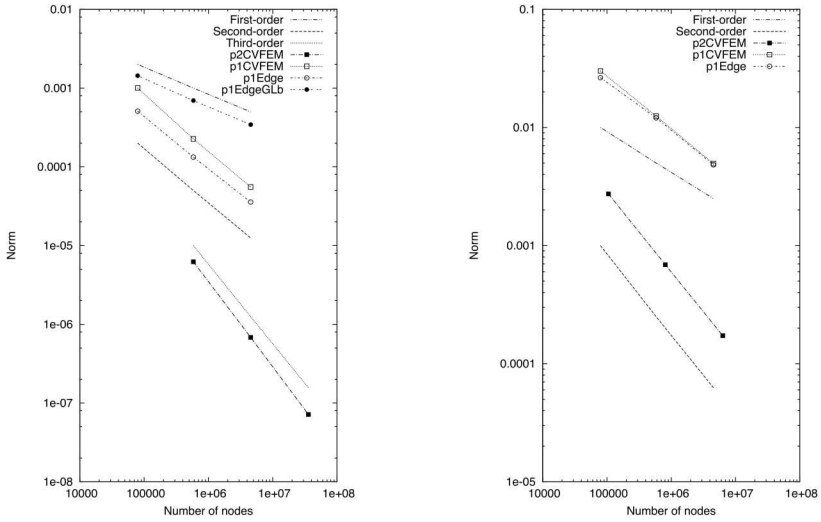


Figure 6: Steady, thermal diffusion manufactured solution; shown are shadings of the temperature field.

times higher. Source terms for the manufactured solutions are added to the equation set and error norms are computed.

In Figure 7, the low-order CVFEM and EBVC methods are shown in addition to the higher-order (P=2) CVFEM solution (L_2 error). Although the Gauss-Lobatto quadrature points are most natural for the EBVC scheme, order of convergence is diminished as compared to usage of the standard CVFEM quadrature points, see Figure 7, (a). A consistent theme of the verification studies is that the error norm for the finest P=1 solution is higher than the first P=2 resolution. Order of convergence for the higher-order scheme is demonstrated to be third-order in space while for P=1, second-order in space is recovered. In this figure, see (b), projected nodal gradient error norms are also shown. Expected order of accuracy for this quantity is order P. In this study, as with all future studies, error norms are constructed based on the discrepency between



(a) Temperature error norms.

(b) Projected nodal gradient error norms.

Figure 7: Steady, passive scalar transport order of accuracy.

the manufactured solution and computed solution at the nodes of the mesh.

5.2. Steady, Passive Scalar Transport MMS

As the secondary goal of the paper is to drive low- and high-order mixed interfaces for production simulations, the choice algorithm for all future cases is CVFEM. With the diffusion operator formally verified, the next case described is a low-Mach fluid mechanics equation set with passive scalar (mixture fraction) transport. The complete manufactured solution is given by

$$u = u_o sin(a\pi x)cos(a\pi y)sin(a\pi z) \quad (33)$$

$$v = -v_o cos(a\pi x)sin(a\pi y)sin(a\pi z) \quad (34)$$

$$w = w_o sin(a\pi x)sin(a\pi y) \quad (35)$$

$$z = z_o cos(b\pi x)cos(b\pi y)cos(b\pi z). \quad (36)$$

$$P = p_o. \quad (37)$$

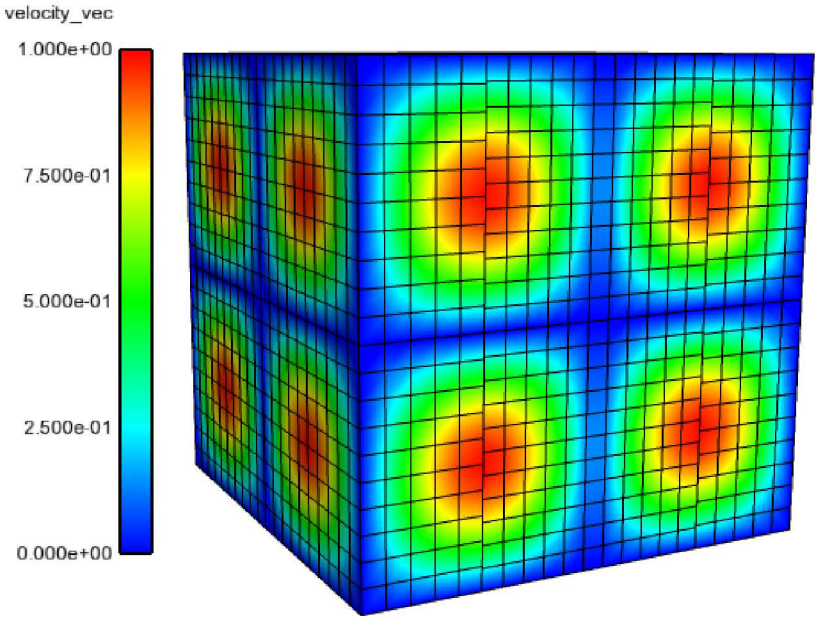
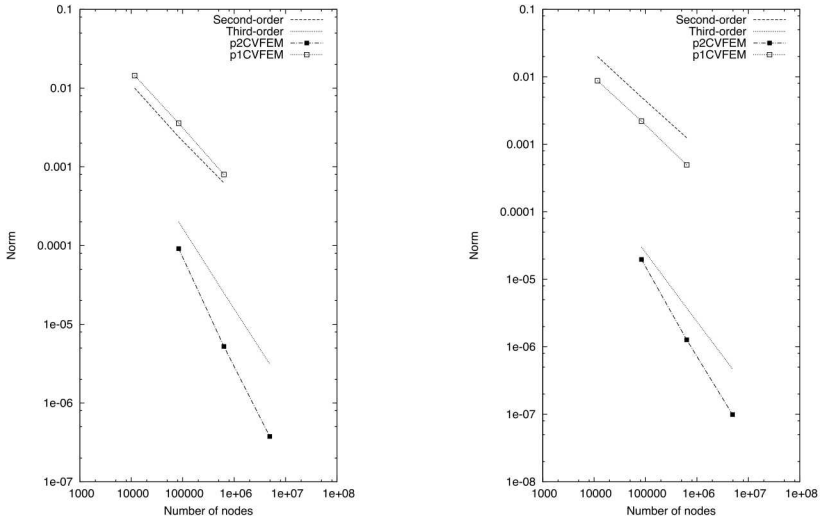


Figure 8: Steady, passive scalar transport MMS solution, shown velocity magnitude shading.

In the above MMS, a and b are 10 and 20, respectively while all reference quantities, e.g., z_o, are unity. The density and viscosity are set such that the coarsest mesh resolution provides a maximum cell Peclet number of five. Figure 8 provides the functional form of the velocity magnitude field for the coarsest R_0 mesh. In this simulation study, three non-conformal blocks are present. Therefore, multiple non-interacting sliding mesh surfaces are supported in the code base. The mesh spacing ratio for the hexahedral interface is 1.1. This MMS uses a constant pressure field with a solenoidal velocity field. Figure 9 outlines the mesh convergence study for the low- and high-order CVFEM solution (L_2 error) for both velocity, (a), and mixture fraction, (b). Results demonstrate design-order for both CVFEM polynomial degree orders tested.



(a) Velocity error norms.

(b) Mixture fraction error norms.

Figure 9: Steady, passive scalar transport order of accuracy.

5.3. Steady, Non-Isothermal, Variable Density MMS

The former construction of a solenoidal velocity field along with a uniform property flow MMS is extended to now include coupled scalar transport (static enthalpy) which provides a variable density MMS (viscosity and specific heat are constant). The solution also includes the effect of a buoyancy source term with an arbitrary gravity vector setting $(-5, 6, 7)^T \frac{m}{s^2}$. The previously described momentum equation is simply augmented by a buoyancy source term, $S_i = (\rho - \rho^o)g_i$. This physics-based MMS was provided as the verification effort to support a recent large-eddy simulation study in which heat flux to a heated cylinder with imposed cross flow was predicted and compared to an experimental data set, see [24]. The complete manufactured solution is

$$u = -u_o cos(a\pi x) sin(a\pi y) sin(a\pi z) \tag{38}$$

$$v = v_o sin(a\pi x) cos(a\pi y) sin(a\pi z) \tag{39}$$

$$w = -w_o sin(a\pi x) sin(a\pi y) cos(a_\pi z) \quad (40)$$

$$h = h_o cos(b\pi x) cos(b\pi y) cos(b\pi z) \quad (41)$$

$$p = -\frac{p_o}{4}(cos(2a\pi x) + cos(2a\pi y) + cos(2a\pi z)). \quad (42)$$

In the above MMS, a and b are 10 and 20, respectively. Again, reference quantities, e.g. p_o, are unity. The density and viscosity are set such that the coarsest mesh resolution provides for a maximum cell Peclet number of six. The laminar Prandtl number is 0.8. Moreover, the universal gas constant, reference pressure, and molecular weight are set such that the reference density, ρ^o, at the reference temperature, T^o, of 300K is 1.0 $\frac{kg}{m^3}$. Temperature is extracted from the transported static enthalpy via the simple relation, $T = \frac{h}{Cp} + T^o$.

Figure 10 provides the functional form of the temperature field on the coarse mesh, R_0. Mesh lines are shown at a two-dimensional slice of the three-dimensional domain. Note that the functional form for the enthalpy manufactured solution is the same as the mixture-fraction field described in section 5.2. In this simulation study, two non-conformal blocks are present. The mesh spacing ratio for the hexahedral interface is 1.6.

Figure 11 outlines the order of convergence for both velocity, (a), and the derived quantity, temperature, (b) (L_1 error shown).

5.4. Viscous Vortex

The final analytical mesh convergence study is that of transient advection of a viscous vortex across a non-conformal interface. In this study, the objective is to evaluate mixed-order interfaces. Specifically, the study compares the following four configurations of P=1 and P=2 interface combinations: $P1P1$, $P1P2$, $P2P1$, and $P2P2$. The centroid of the vortex initial coordinate location is one diameter upstream of the non-conformal interface. The analytical velocity field is given by

$$\begin{aligned} u(x,y,t) &= u^\infty - \frac{\beta}{r^o + \tau^2} e^{(\frac{1}{2}(1-\frac{r^2}{\tau}))} C_y, \\ v(x,y,t) &= \frac{\beta}{r^o + \tau^2} e^{(\frac{1}{2}(1-\frac{r^2}{\tau}))} C_x, \end{aligned} \quad (43)$$

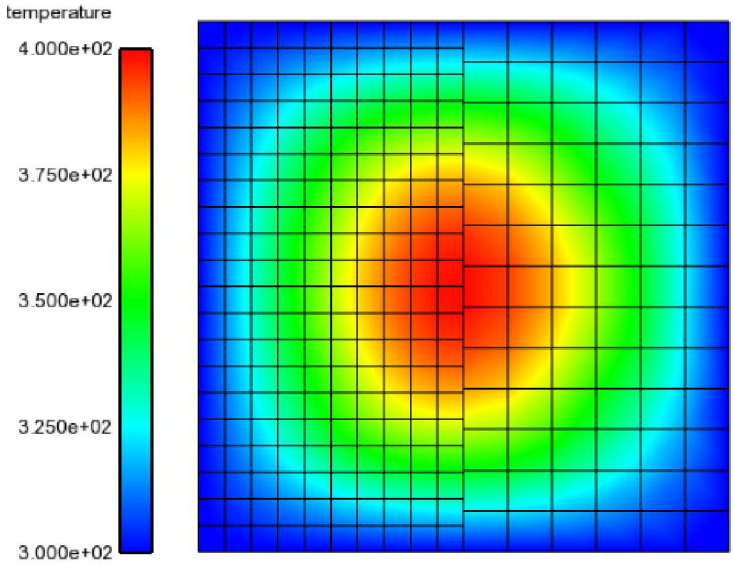
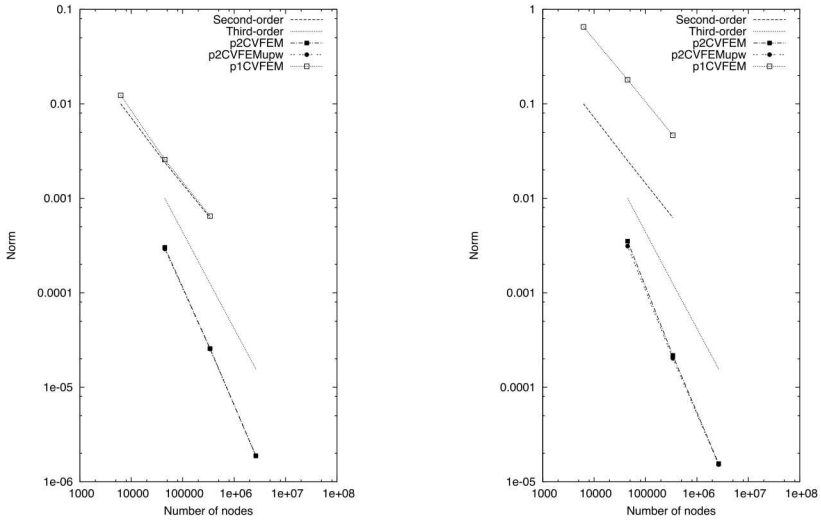


Figure 10: Steady, variable densityMMS solution, shown is a two-dimensional plane of mixture fraction shading.



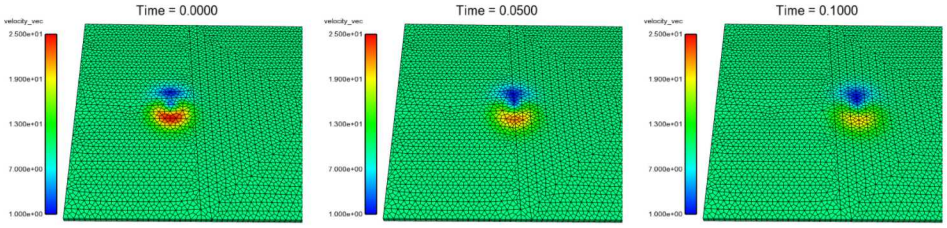
(a) Velocity error norms.

(b) Temperature error norms.

Figure 11: Steady, variable density order of accuracy.

where, u^∞ is the far field x-advection velocity, β is the vortex strength, r^o is the specified vortex radius, r is the local radius, and $\tau = 1 + \frac{2\beta}{r^o Re}t$. In this expression, the pointwise current x-coordinate C_x is a function of the local time, t, via $C_x = x^o + u^\infty t$. The parameters used in this study are as follows: $(u^\infty, \beta, R) = (10, 15, 0.25)$. The viscosity and density are set such that the R_0 mesh provides a maximum cell Peclet number of 18. Although the solution is two-dimensional, a three-dimensional domain is used in this study. Figure 12 provides shadings of the velocity magnitude at three solution times for the R_1 mesh. An unstructured hexahedral topology is again used with top, bottom, front, and back symmetry boundary planes. A symmetry boundary condition is simply defined as a zero-penetration, zero-tangential stress boundary condition. The left-most surface is an inflow boundary while the right-most surface (not shown) is an open boundary.

Error norms are computed after the vortex has moved one diameter past



(a) Time = 0.0 seconds. (b) Time = 0.05 seconds. (c) Time = 0.10 seconds.

Figure 12: Viscous vortex velocity magnitude shadings at three time planes, before, at and after non-conformal interface.

the non-conformal interface. The simulation is terminated at 0.10 seconds and error norms are computed. A second-order BDF2 fully implict time integrator is used. Outer Picard iterations are used to converge within the time step. Studies indicate that four outer iterations result in a reduction of velocity nonlinear residuals of approximately five orders of magnitude. Finally, the pressure field is initialized as, $p(x, y, t^o) = -\frac{1}{2}\rho\beta^2 e^{(1-r^2)}$.

Figure 13 provides velocity error norms at the final solution time of 0.1 seconds for the CVFEM suite of discretizations (L_2 error). Again, the higher-order scheme provide error norms that are much lower than the lower-order solution. Although it is certainly expected that a higher-order solution will be more accurate than a lower-order solution on the same number of nodes, as with all previous studies, a substantial shift in error is noted. This suggests that with a modest increase in the underlying basis, significant error norm reductions can be realized. At present, the higher-order matrix assembly and solve are not optimized for performance. Therefore, this study does not include statements of efficiency for each underlying method (defined by, e.g., simulation time to provide a given error). However, future efforts of interest to the ExaWind project will concentrate on assembly and solver efficiency improvements on next generation platforms (NPG) to drive performance increases for the higher-order method.

An interesting finding to the error norm plots demonsted in Figure 13 is as

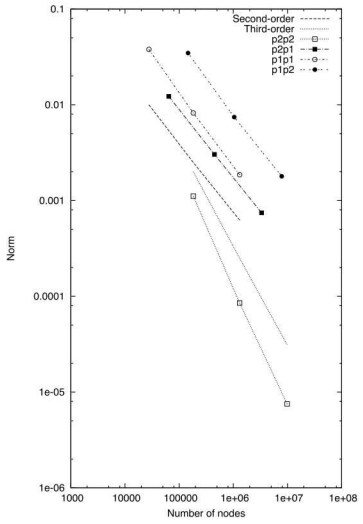


Figure 13: Viscous vortex norms for varying polynomial order combinations at the non-conformal interface.

follows: if a vortex originates within a low-order block, error norms follow the low-order error trends. Moreover, if the vortex begins in the higher-order block and transition to a low-order block, the magnitude of the error is slightly lower than the $P1P1$ simulation while remaining second-order in space. This finding has significance for production runs that require tracking vortices from low- to higher- order blocks or visa-versa.

6. Demonstration Cases

In the final section of this paper, a set of proof-of-concept production-like simulation studies are performed. These simulations serve to establish the ability to run the demonstrated design-order algorithm in practical application areas of interest. Although each case includes the usage of LES, the intent is not to drive solution verification of the chosen flows. Therefore, a formal solution verification process, as outlined in Domino [24], is not followed. In future project efforts, such formal solution verification will be carried out during

the model validation process in which characterization of the numerical error must be performed. The Trilinos solver package is again active using the Tpetra/Balos/ifpack2/Muelu solver stack, [26]. As with previous scaling efforts described in [26], all advection/diffusion systems, i.e., momentum, used a symmetric Gauss-Seidel preconditioned GMRES approach while the pressure Poisson system used the Trilinos/Muelu package [40]. The Muelu preconditioner strategy used a smoothed aggregation algebraic mutigrid (AMG) method that employs a degree two Chebyshev polynomial smoother, direct coarse grid solve, and coarse matrix rebalancing via multidimensional-jagged geometric partitioning. Mesh anisotropy is detected through an auxiliary Euclidean norm-based matrix. For more information on the Muelu package, options, etc., see [40].

6.1. Spinning Cube

Zhang [12], in support of his dissertation, modeled a spinning cube using a spectral higher-order method [11] at modest Reynolds numbers (100 and 1000) with a revolution-per-minute value of $\frac{\pi}{2}$. Q-criterion visualizations for various orders of accuracy were provided. In this paper, the simulation a spinning cube is motivated by this foundational sliding mesh simulation effort, however, the present study uses a higher Reynolds number (4,000) and a much larger revolution rate (3600) in order to tax the non-conformal algorithm. Both efforts are notional in nature as experimental data does not exist to validate the model suite. The quantity of interest provided in this study is both visualized Q-criterion and mean drag coefficient. In this simulation study, a 1 cm^3 cube is placed within a 10x10x23 cm domain. An inner cylindrical mesh of radius 2.5 cm surrounds the cube. The entrance distance from the inflow to the surface of the cube is 7.5 cm. Figure 14 provides a wireframe view of the mesh. Note that the cube is within an inner cylinder solution which allows for generated vortical flow from the spinning cube to pass the non-conformal interface. In Zhang's study, the cylindrical mesh surrounding the spinning cube spanned the complete streamwise domain, i.e., from the inflow boundary to the open boundary condition. All sides of the channel are represented as a symmetry

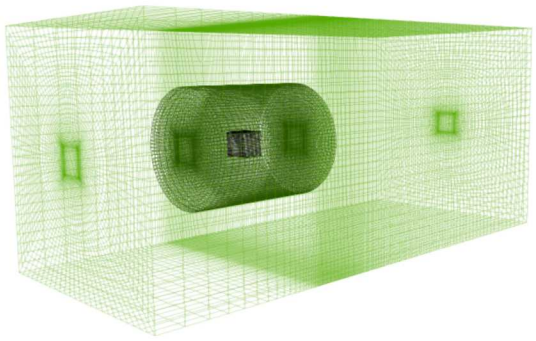


Figure 14: Spinning cube inner mesh with sliding interface represented by the outer cylinder.

boundary condition. For all simulations, a fixed time step was used. The chosen fixed time step for each mesh resolution resulted in a Courant number of approximately three. For the coarsest mesh, a max cell Peclet number of 5000 is noted. A second-order time integrator was again used (BDF2) with multiple nonlinear iterations taken (two per time step).

As this is a turbulent flow, LES model form and low-dissipation numerics must be addressed. The underlying stabilization methodology is based on entropy-viscosity stabilization, [41], with an underlying stabilization kernel following that of Shakib, [42]. The implicit LES model follows that of Guermond *et al*. [43] in that a nonlinear stability operator (NSO) is used for both stabilization and turbulence model. This choice of turbulence model is based on the design-order aspect of the NSO in that as the mesh is refined (either in space or polynomial order) the stabilization/model adjusts according to the activated numerical discretization. For the low-Mach NSO form, a fine-scale turbulent kinetic energy residual was used to drive the modeled viscosity. The NSO kernel for momentum component, i, adapted to the CVFEM form, is

$$-\sum_e \int_\Gamma \nu(\mathbf{R}_{KE}) g^{kj} \frac{\partial u_i}{\partial x_j} n_k dS. \tag{44}$$

Above, the metric tensor, g^{kj} is defined as

$$g^{kj} = \frac{\partial x_k}{\partial \xi_l} \frac{\partial x_j}{\partial \xi_l}. \tag{45}$$

The numerical viscosity, ν, is a function of a fine-scale kinetic energy equation which is approximated as dotting the velocity with the fine-scale momentum residual expression, Equation 13,

$$\mathbf{R}_{KE} = \frac{u_j(\frac{\partial p}{\partial x_j} - G_j p)}{2\rho}. \tag{46}$$

The precise definition of ν is given by,

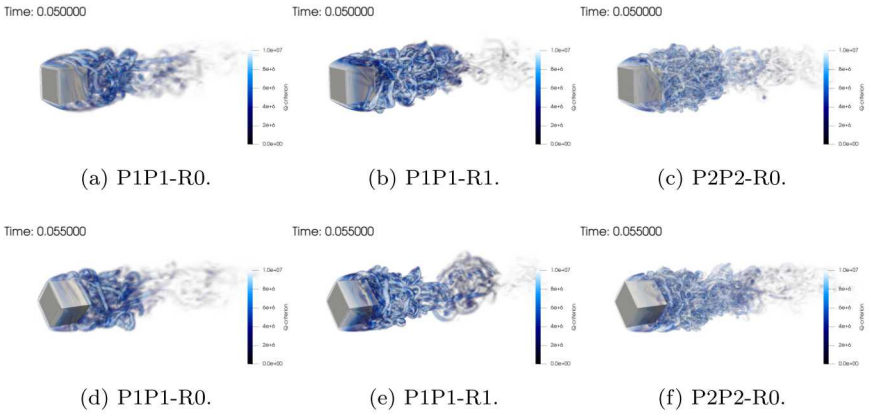
$$\nu = \sqrt{\frac{\mathbf{R}_{KE}\mathbf{R}_{KE}}{\frac{\partial u_i}{\partial x_k} g^{kj} \frac{\partial u_i}{\partial x_j}}}. \tag{47}$$

Table 1 summarizes the meshes used in this study which mirror the viscous vortex mesh order configurations by way of mixed-order mesh interfaces. Also shown in this table is the mean drag coefficient for each mesh simulated. Converged solutions reported herein are defined by a mean drag coefficient to be within 1 percent of the previous time step. Theory and experiments suggests that the drag of a non-spinning cube is order unity, [44], as such, the additional rotation (roll) augments the drag experienced on the cube.

Table 1: Spinning cube simulation number of elements, nodes, and mean drag coefficient, C_D, prediction.

Simulation	**Elements**	**Nodes**	C_D
P1P1-R0	753125	784004	1.2016
P1P2-R0	753125	3943814	1.1972
P2P1-R0	753125	2988444	1.1581
P2P2-R0	753125	6148254	1.1576
P1P1-R1	6025416	6148254	1.1565

Figure 15 provides the volume-rendered Q-critereon at 0.05 (top) and 0.055 (bottom) seconds for a select combination of polynomial and mesh resolutions. This time dispartity (0.005 seconds) represents the time for the cube to rotate forty-five degrees. As this set of volume rendering only offers qualitative commentary, one can easily see the effect of mesh and polynomial refinement on the ability to capture the vortical structures created by the spinning cube.



(a) P1P1-R0. (b) P1P1-R1. (c) P2P2-R0.

(d) P1P1-R0. (e) P1P1-R1. (f) P2P2-R0.

Figure 15: Spinning cube at 0.05 (a-c) and 0.055 (d-f) seconds for various mesh and polynomial orders; volume-rendered Q-criterion.

Qualitative results suggest that the polynomial promotion approach provides for a well-resolved turbulent flow field. The clustering of mean drag coefficients represented in Table 1 suggests a trend of convergence for either the refined low-order or baseline higher-order simulation. As would be expected, the polynomial order for the mesh block surrounding the rotating cylinder is less critical for this surface-based quantity of interest.

6.2. Vestas V27 blades

The final simulation study represents a full-scale Vestas V27 (225 kw) wind turbine. In this preliminary effort, only the airfoils are meshed. The baseline hybrid mesh is comprised of approximately 150 million low-order hexahedral, tetrahedral, pyramid, and wedge topologies. Hexahedral elements form the non-conformal interface. The tip speed ratio is approximately 6.8 (inflow velocity 10 $\frac{m}{s}$, RPM = 5.0725, rotor diamter = 27 m). The max cell Peclet number in the flow is approximately 1.6 million. The Wale Adapting Local Viscosity (WALE) LES model is used, see [45], with wall functions (max normalized wall units, Y^+ 150). It is noted that this configuration is solely for numerical testing of the algorithm. More formal validation exercises are planned. Figure 16 outlines the inner thin cylindrical mesh description that rotates with respect the outer

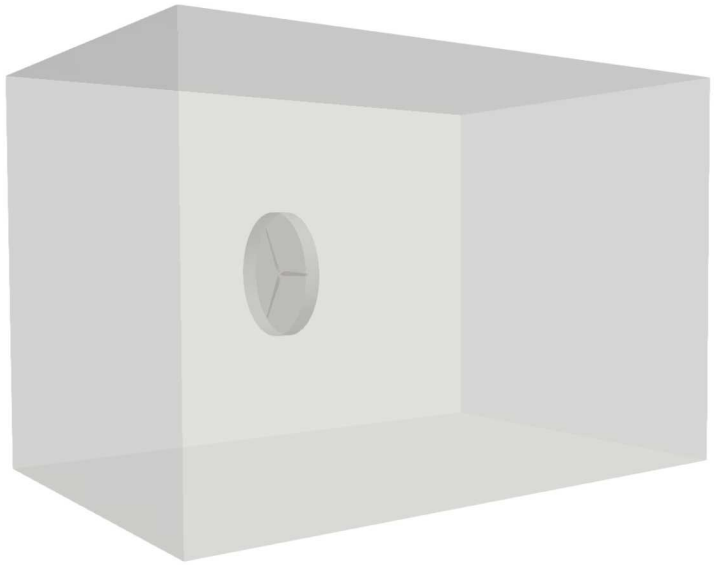


Figure 16: V27 mesh description; shown is the inner cylinder containing the full-scale three wind turbing blades.

domain. As with the previous spinning cube simulation, sharp corners at the cylindrical inner rotating mesh exist.

In Figure 17, the volume rendered Q-criterion is shown from the windward (a) and leeward (b) directions. The advection of the vortical flow through the hybrid mesh proceeds without numerical artifacts. As the flow structure advects through the downstream, low-order tetrahedral mesh, dissipation is noted. This study represents a non-design operating configuration for revolution rate and tip speed ratio. As such, separation is noted. The simulation was run on 1024 cores using the Sandia National Laboratory institutional cluster, Skybridge. This platform is a capacity-based Cray-based platform (approximately 30,000 core) that has a 2.6 GHz Sandy Bridge Intel-based processor with infiniband interconnect. Simulation timing results for matrix assembly, re-initialization, and pre-conditioner setup costs for the 150 million element run are shown in Table 2. Timing breakdowns are shown for both the 3x3 momentum and 1x1 continuity equation system. As expected, the continuity solve consumes more than 50% of

(a) Winward view of the volume rendered Q-criterion.

(b) Leeward view of the volume rendered Q-criterion.

Figure 17: Vestas V27 rotor simulation at 2.389763 seconds.

the simulation time (average linear iterations over the study was approximately 30) while parallel search and the non-conformal algorithm consumed $< 1\%$ of the simulation time. At this scale, the continuity preconditioner setup cost is extremely efficient.

Table 2: Timing breakdown (matrix initialization, matrix assemble, matrix solve, and preconditioner setup) for the 150 million element hybrid V27 simulation; shown are percentages of total simulation time for the equation systems of momentum (uvw) and pressure (p).

System	**Init.**	**Assembly**	**Solve**	**P. Setup**
uvw	7.72 %	9.86 %	10.08 %	<1 %
p	3.48 %	3.03 %	54.01 %	<1 %

7. Conclusions

A novel second- and third-order spatially accurate CVFEM/DG scheme with non-conformal mesh interfaces has been demonstrated using a suite of formal MMS cases ranging from simple diffusion to the full low-Mach fluid equation set. The methodology is motivated from a stable and accurate DG method applied

to the CVFEM non-conformal interface. The hybrid technique has been demonstrated on both non-conformal surfaces with both mesh spacing and polynomial order discontinuities. For simulations that required an inner cylinder rotating block, corner effects were mitigated by deploying a surface-based approach that does not rely on mesh extrusion. Verification solutions indicate design order even with length scale jumps as high as 1.6. When mixing polynomial orders, verification demonstrates order of convergence to revert towards the lower-order polynomial errors. Two simulations of production interest (spinning cube and a full V27 set of blades) have also been provided to establish the ability to deploy the underlying numerical method for high-fidelity wind energy applications that include high Reynolds number turbulent flow with rotating inner solid blades. Results from the spinning cube and Vestas wind turbine indicate no appreciable artifacts at the non-conformal surface even when rotation rates and simulation Reynolds numbers are high. Therefore, confidence in the underlying numerical approach has been established such that the project can move forward towards more formalized LES validation studies including multiple turbines.

Acknowledgments

Sandia National Laboratories is a multimission 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. This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of two U.S. Department of Energy organizations (Office of Science and the National Nuclear Security Administration) responsible for the planning and preparation of a capable exascale ecosystem, including software, applications, hardware, advanced system engineering, and early testbed platforms, in support of the nations exascale computing imperative.

The author would like to thank the Sandia National Laboratories (SNL)

staff Dr. Matt Barone, for the production meshes, and Dr. Robert Knauss for discussions on the viscous vortex verification effort. Dr Tyler Voskuilen, also of SNL, is acknowledged for his idea to bloat the matrix stencil for reduced re-initializtion steps.

References

[1] Exawind project, available from: https://exawind.org/ [last accessed September 2017] (2017).

[2] Exascale Computing Project, available from: https://exascaleproject.org/ [last accessed September 2017] (2017).

[3] D. Digraskar, Simulations of flow over wind turbines, Master's thesis, University of Massachusetts (2010).

[4] A. Kirby, M. Brazell, Z. Yang, R. Rajib, B. Ahrabi, D. Mavriplis, J. Sitaraman, M. Stoellinger, Wind farm simulations using an overset hp-adaptive approach with blade-resolved turbine models, no. AIAA 2017-3958 in 23rd Aerospace Sciences Meeting and Exhibit, Aerospace Sciences Meeting, Denver, CO, Jun. 5–7, 2017, AIAA, Washington, D.C., 2017.

[5] G. Chesshire, W. Henshaw, Composite overlapping meshes for the solution of partial differentail equations, J. Comp. Phys 90 (1990) 1–64.

[6] I. Chie, R. Meakin, On automating domain connectivity for overset grids, no. AIAA 95-054 in 33rd Aerospace Sciences Meeting and Exhibit, Aerospace Sciences Meeting, Reno, NV, Jan. 9–12, 1995, AIAA, Washington, D.C., 1995.

[7] G. Schneider, M. Raw, A skewed, positive influence coefficient upwinding procedure for control-volume-based finite-element convection-diffusion computation, Num. Heat Trans. 9 (1986) 1–26.

[8] D. Arnold, F. Brezzi, B. Cochburn, D. Marini, Unified analysis of the discontinous Galerkin method for eliptic problems, SIAM J. Numer. Anal. 39 (2002) 1749–1779.

[9] D. Gartling, Multipoint constraint methods for moving body and non-contiguous mesh simulations, Int. J. Numer. Meth. Engr. 47 (2004) 471–489.

[10] B. Carnes, K. Copps, Thermal contact algorithms in Sierra mechanics; mathematical background, numerical verification, and evaluation of performance, Tech. Rep. SAND2008-2607, Sandia National Laboratories SAND Series (2008).

[11] B. Zhang, C. Liang, A simple, efficient, and high-order accurate curved sliding-mesh interface approach to spectral difference method on coupled rotating and stationary domains, J. Comp. Phy. 295 (2015) 147–160.

[12] B. Zhang, A high-order computational framework for simulating flows around rotating and moving objects, Ph.D. thesis, George Washington University (2017).

[13] J. McNaughton, I. Afgan, D. Apsley, S. Rolfo, T. Stallard, P. Stansby, A simple sliding-mesh interface procedure and its application to the CFD simulation of a tidal-stream turbine, Int. J. Numer. Meth. Engr. 74 (2014) 250–269.

[14] E. Blades, D. Marcum, A sliding interface method for unsteady unstructured flow simulations, Int. J. Numer. Meth. Engr. 53 (2007) 507–529.

[15] J. Le Gouez, Higher-order overset interpolation via weighted least-square polynomial reconstruciton for finite volume cfd, no. AIAA 2016-2051 in 54th Aerospace Sciences Meeting and Exhibit, Aerospace Sciences Meeting, San Diego, CA, Jan. 4–8, 2016, AIAA, Washington, D.C., 2016.

[16] M. Edred, L. Ng, M. Barone, S. Domino, Multifidelity uncertainty quantification using spectral stochastic discrepancy models., in: R. Ghanem,

D. Higdon, H. Owhadi (Eds.), Handbook of Uncertainty Quantification, Springer International Publishing, Switzerland, 2016, pp. 1–45.

[17] S. Domino, Towards verification of sliding mesh algorithms for complex applications using MMS, in: P. Moin, J. Larsson, N. Mansour (Eds.), Studying Turbulence Using Numerical Simulation Databases - XIII, Stanford Center for Turbulence Research, 2010, pp. 167–182.

[18] J. Lopez, G. Rios Rodriguez, A sliding-mesh strategy based on the mixing of continopus and discontinuous finite element methods for compressible flow problems, available from: https://www.amcaonline.org.ar/ [last accessed September 2017] (2013).

[19] The NaluCFD github project, available from: https://github.com/NaluCFD [last accessed September 2017] (2017).

[20] Department of Energy Wind Technologies Office, Atmosphere to Electrons, available from: https://energy.gov/eere/wind/atmosphere-electrons/ [last accessed September 2017] (2017).

[21] M. Churchfield, S. Lee, P. Moriarty, L. Martinez, S. Leonardi, G. Vijayakumar, J. Brasseur, A large-eddy simulation of wind-plant aerodynamics (AIAA 2017-3958).

[22] H. C. Edwards, A. Williams, G. Sjaardema, D. Baur, W. Cochran, Sierra toolkit computational mesh computational model, Tech. Rep. SAND2010-1192, Sandia National Laboratories SAND Series (2010).

[23] 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, A. Williams, An overview of Trilinos, Tech. Rep. SAND2003-2927, Sandia National Laboratories SAND Series (2003).

[24] S. Domino, Exploring model-form uncertainties in large-eddy simulation, in: P. Moin, J. Urzay (Eds.), Studying Turbulence Using Numerical Sim-

ulation Databases - XVI, Stanford Center for Turbulence Research, 2016, pp. 435–444.

[25] L. Jofre, S. Domino, G. Iaccarino, A framework for characterizing structural uncertainty in large-eddy simulation closures, Flow Turbul. Combust. 99 (2017) 1–23.

[26] P. Lin, M. Bettencourt, S. Domino, T. Fisher, M. Hoemmen, J. Hu, E. Phipps, A. Prokopenko, S. Rajamanickam, C. Siefert, S. Kennon, Towards extreme-scale simulations for low-mach fluids with second-generation Trilinos, Parallel Processing Letters 24 (4). doi:10.1142/S0129626414420055.

[27] J. Ferziger, M. Peric, in: R. Ghanem, D. Higdon, H. Owhadi (Eds.), Computational methods for fluid dynamics, Third edition., Springer International Publishing, Switzerland, 2002.

[28] S. Domino, A comparison between low-order and higher-order low-Mach discretization approaches, in: M. P., J. Urzay (Eds.), Studying Turbulence Using Numerical Simulation Databases - XV, Stanford Center for Turbulence Research, 2014, pp. 387–396.

[29] S. Domino, Sierra low-Mach Module Nalu: Theory manual, Tech. Rep. SAND2015-3107W, Sandia National Laboratories SAND Series (2014).

[30] Online NaluCFD documentation, available from: http://nalu.readthedocs.io/en/latest/ [last accessed September 2017] (2017).

[31] C. Swaminanthan, V. Voller, V. Patankar, A streamline upwing control volume finite element methods for modeling fluids flow and heat transfer problems, Int. J. Numer. Meth. Engr. 13 (1993) 169–184.

[32] M. Martinez, Comparison of galerkin and control volume finite element for advection-diffusion problems, Int. J. Numer. Meth. Engr. 50 (2004) 347–376.

[33] Paolucci, S., On the Filtering of Sound Waves from the Navier-Stokes Equations, Tech. Rep. SAND82-8257, Sandia National Laboratories SAND Series (1982).

[34] C. L. Merkle, Y. H. Choi, Computation of compressible flows at very low-Mach numbers, no. AIAA 86-0351 in 24th Aerospace Sciences Meeting and Exhibit, Aerospace Sciences Meeting, Reno, NV, Jan. 6–9, 1986, AIAA, Washington, D.C., 1986.

[35] S. Domino, Toward verification of formal time accuracy for a family of approximate projection methods using the method of manufactured solutions, in: P. Moin, N. Mansour (Eds.), Studying Turbulence Using Numerical Simulation Databases - XI, Stanford Center for Turbulence Research, 2006, pp. 163–177.

[36] C. Rhie, W. Chow, Numerical study of the turbulent flow past an airfoil with trailing edge separation, AIAA J. 21 (1983) 1525–1532.

[37] M. Braack, G. Lube, Finite elements with local projection stabilization for incompressible flow problems, J. Comp. Math. 27 (2009) 116–147.

[38] R. Codina, Pressure stability in fractional step finite element methods for incompressible flows, J. Comp. Phys. 170 (2001) 112–140.

[39] P. Roache, Code verification by the method of manufactured solutions, J. Fluids Engr. 124 (2002) 4–10.

[40] Online Muelu documentation, available from: https://trilinos.org/packages/muelu// [last accessed September 2017] (2017).

[41] J. Guermond, R. Pasquetti, B. Popov, Entropy viscosity method for nonlinear conservation laws, J. Comp. Phy. 230(11) (2011) 4248–4267.

[42] F. Shakib, T. Hughes, Z. Johan, A new finite element formulation for computational fluid dynamics: X. the compressible Euler and Navier-Stokes equations, Comp. Meth. Appl. Mech. Engr. 89 (1991) 141–219.

[43] J. Guermond, A. Larios, T. Thompson, Validation of an entropy-viscosity model for large eddy simulation.), in: J. Frohlich (Ed.), Direct and Large-Eddy Simulation IX, Springer International Publishing, Switzerland, 2015.

[44] J. Holmes, E. English, Aerodynamic forces and moments on cubes and flat plates, with applications to wind-borne debris (LSU-R-04-002).

[45] F. Nicoud, F. Ducros, Subgrid-scale stress modelling based on the square of the velocity gradient tensor, Flow Turb. Comb. 62 (1999) 183–200.