1 An Overset Mesh Approach for 3D Mixed Element 2 High-Order Discretizations

3 Michael J. Brazell^{a,*}, Jayanarayanan Sitaraman^b, Dimitri J. Mavriplis^a

4 ^a*Department of Mechanical Engineering, University of Wyoming, Laramie, WY*
5 *82071-3295, USA*

6 ^b*Parallel Geometric Algorithms LLC, 814 S Mary Ave., Sunnyvale, CA 94087, USA*

7 Abstract

A parallel high-order Discontinuous Galerkin (DG) method is used to solve the compressible Navier-Stokes equations in an overset mesh framework. The DG solver has many capabilities including: *hp*-adaption, curved cells, support for hybrid, mixed-element meshes, and moving meshes. Combining these capabilities with overset grids allows the DG solver to be used in problems with bodies in relative motion and in a near-body off-body solver strategy. The overset implementation is constructed to preserve the design accuracy of the baseline DG discretization. Multiple simulations are carried out to validate the accuracy and performance of the overset DG solver. These simulations demonstrate the capability of the high-order DG solver to handle complex geometry and large scale parallel simulations in an overset framework.

8 *Keywords:* Overset, High-order, Discontinuous Galerkin, TIOGA

9 1. Introduction

10 High-order methods are attractive because they provide higher accuracy
11 with fewer degrees of freedom. Discontinuous Galerkin (DG) methods [1–22]
12 have received particular attention; these methods combine the ideas of finite-
13 element and finite-volume methods allowing for high-order approximations
14 and geometric flexibility.

*Corresponding author

Email address: mbrazell@uwyo.edu (Michael J. Brazell)

15 The goal of this work is to devise an accurate, efficient and robust three-
 16 dimensional high-order method based on DG discretizations [23] for simu-
 17 lating a wide variety of compressible flows in an overset grid framework.
 18 Although the DG solver can handle unstructured meshes there are some sit-
 19 uations where even an unstructured solver could benefit from an overset grid
 20 framework. Bodies in relative motion such as helicopters or wind turbines are
 21 difficult to simulate with a single grid. Typically these simulations require
 22 mesh movement or re-meshing every time step. Overset grids solve this issue
 23 by allowing multiple grids to move relative to each other. Another advantage
 24 to overset grids is the ability to combine a near-body unstructured DG solver
 25 with an efficient cartesian mesh off-body solver, this can greatly increase the
 26 overall efficiency and capabilities of the solver. This has been demonstrated
 27 with success in [24], which combines a second-order accurate, unstructured,
 28 finite-volume solver [25] with an AMR finite difference solver [26, 27]. Also,
 29 previous work by our group combined a hexahedral DG solver with a second-
 30 order, finite-volume, unstructured solver [28, 29]. These approaches have two
 31 drawbacks: firstly they depend on low-order methods for the unstructured
 32 solvers and secondly many fringe cells are needed in the overlap region of the
 33 grids to maintain accuracy.

34 To overcome these issues high-order methods with compact stencils can
 35 be used. Recently, DG has been used in an overset framework to solve the
 36 two-dimensional, Euler equations on abutting and overset grids [30]. Flux Re-
 37 construction (FR) has been used to solve the two-dimensional, compressible
 38 Navier-Stokes equations on sliding meshes [31]. Streamline/upwind Petrov-
 39 Galerkin (SUPG) has been used to solve the two-dimensional, compressible,
 40 Navier-Stokes equations along with the Reynolds Averaged Navier-Stokes
 41 (RANS) equations on steady and unsteady moving mesh problems [32]. Spec-
 42 tral Element Method (SEM) has been used to solve the three-dimensional,
 43 incompressible Navier-Stokes equations on moving, overlapping, hexahedral
 44 grids [33]. This has demonstrated the feasibility and high level of accuracy
 45 that can be obtained using compact-stencil, high-order methods (DG, FR,
 46 SUPG, and SEM) in an overset framework.

47 In this paper, our previous work has been extended to the new DG solver
 48 using a newly developed high-order overset mesh framework called TIOGA
 49 (Topology Independent Overset Grid Assembler). The high-order, parallel
 50 DG solver [23] is capable of solving the compressible Navier-Stokes equations
 51 and the RANS equations closed by the Spalart-Allmaras turbulence model
 52 (negative-SA variant) [34, 35]. It also can handle hybrid, mixed-element

53 meshes (tetrahedra, pyramids, prisms, and hexahedra), moving meshes, curved
 54 elements, and incorporates both p -enrichment and h -refinement capabilities
 55 using non-conforming elements (hanging nodes). The novelty in this
 56 approach is the capability of using a 3D high-order method with mixed-
 57 elements, moving mesh, and turbulent flow in an overset framework [36].
 58 This gives the ability to solve complicated relative motion problems at high-
 59 order and be combined with an efficient off-body solver.

60 In the following sections, the governing equations are described, fol-
 61 lowed by the DG discretization and its implementation for three-dimensional
 62 problems. The solution methodology is described next which discusses the
 63 strongly coupled, implicit, non-linear solver within the overset framework.
 64 Lastly, a variety of simulations are used to demonstrate the order of ac-
 65 curacy, temporal accuracy, implicit solver performance, and moving mesh
 66 accuracy of the overset solver. This is followed by two large scale simulations
 67 demonstrating the use of mixed element meshes and turbulent flow over a
 68 wing.

69 2. Governing Equations

The Navier-Stokes equations govern the dynamics of compressible fluids and are given as:

$$\frac{\partial U_k}{\partial t} + \frac{\partial F_{ki}}{\partial x_i} = 0 \quad (1)$$

where they represent the conservation of mass, momentum, and energy. The solution vector U and flux F are defined as:

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{bmatrix}, \quad F = \begin{bmatrix} \rho u & \rho v & \rho w \\ \rho u^2 + P - \tau_{11} & \rho uv - \tau_{12} & \rho uw - \tau_{13} \\ \rho uv - \tau_{21} & \rho v^2 + P - \tau_{22} & \rho vw - \tau_{23} \\ \rho uw - \tau_{31} & \rho vw - \tau_{32} & \rho w^2 + P - \tau_{33} \\ \rho uH - \tau_{1j}u_j + q_1 & \rho vH - \tau_{2j}u_j + q_2 & \rho wH - \tau_{3j}u_j + q_3 \end{bmatrix} \quad (2)$$

where ρ is the density, u, v, w are the velocity components in each spatial coordinate direction, P is the pressure, E is total internal energy, $H = E + P/\rho$ is the total enthalpy, τ is the viscous stress tensor, and q is the heat flux. The viscosity is a function of the temperature given by the Sutherland's formula. These equations are closed using the ideal gas equation of state:

$$\rho E = \frac{P}{\gamma - 1} + \frac{1}{2}\rho(u^2 + v^2 + w^2)$$

70 where $\gamma = 1.4$ is the ratio of specific heats. In all of the following, Einstein
 71 notation is used where the subscripts of i and j represent spatial dimensions
 72 and have a range of 1 to 3 and the index k varies over the number of variables.
 73 The arbitrary Eulerian-Lagrangian (ALE) formulation of the compressible
 74 Navier-Stokes equations is formed by adding the ALE flux terms:

$$F_{ALE} = - \begin{bmatrix} \rho \dot{x}_1 & \rho \dot{x}_2 & \rho \dot{x}_3 \\ \rho u \dot{x}_1 & \rho u \dot{x}_2 & \rho u \dot{x}_3 \\ \rho v \dot{x}_1 & \rho v \dot{x}_2 & \rho v \dot{x}_3 \\ \rho w \dot{x}_1 & \rho w \dot{x}_2 & \rho w \dot{x}_3 \\ \rho E \dot{x}_1 & \rho E \dot{x}_2 & \rho E \dot{x}_3 \end{bmatrix} \quad (3)$$

75 to the static flux F (defined previously in equation (3)), where $\dot{x}_i$ are the
 76 mesh movement terms in each coordinate direction. Currently only rigid,
 77 analytic mesh movements are implemented.

78 3. DG Formulation

In this section the DG finite-element formulation used to solve the Navier-Stokes equations is described. To derive the weak form, equation (1) is first multiplied by a test function ϕ and integrated over the domain Ω to give:

$$\int_{\Omega} \phi_m \left(\frac{\partial U_k}{\partial t} + \frac{\partial F_{ki}}{\partial x_i} \right) d\Omega = 0.$$

Next, integration by parts is performed and the residual R_{km} is defined as:

$$R_{km} = \int_{\Omega} \left(\phi_m \frac{\partial U_k}{\partial t} - \frac{\partial \phi_m}{\partial x_i} F_{ki} \right) d\Omega + \int_{\Gamma} \phi_m F_{ki}^* n_i d\Gamma = 0$$

where ϕ are the basis functions and the solution is approximated as $U_k = \phi_m a_{km}$, where the repeated index m is summed over the number of basis functions. The basis is formed by using piecewise discontinuous shape functions. Each element's shape function is represented as continuous polynomials within the cell and zero outside the cell. The basis in this work is hierarchical and is formed by combining a $p = 1$ Lagrangian basis, using the vertex of each element, and a Jacobi polynomial as a kernel for creating higher degree terms. An example of a shape function for a segment is:

$$\phi_m(\xi) = \begin{cases} 1 & m = 0 \\ \frac{1-\xi}{2}, \frac{1+\xi}{2} & m = 1 \\ \left(\frac{1-\xi}{2}\right) \left(\frac{1+\xi}{2}\right) P_{m-2}^{\alpha,\beta}(\xi) & m > 1 \end{cases}$$

79 where α and β are parameters that give Jacobi polynomials their orthogo-
80 nality properties with respect to the weight $(1 - \xi)^\alpha(1 + \xi)^\beta$. Similar to the
81 segment a basis is created for tetrahedral, pyramidal, prismatic, and hexa-
82 hedral element types. Also, the basis is used to represent both the solution
83 and geometrical mapping. The polynomial degree can be chosen indepen-
84 dently for each, for example the solution can be represented by a $p = 1$ basis
85 and the geometrical mapping can be represented by a $p = 2$ basis.

86 The residual consists of volume and face integrals which are approximated
87 using numerical quadrature. For hexahedral elements and quadrilateral ele-
88 ments tensor products of Gaussian quadrature points are used. For the other
89 element types a set of efficient numerical quadrature rules are used [37]. The
90 quadrature rules are chosen to exactly integrate polynomials within the vol-
91 ume integrals to $2p$ accuracy and the face integrals to $2p + 1$ accuracy. The
92 integrals over faces Γ require special treatment for the fluxes in these terms.
93 The advective fluxes are calculated using a Riemann solver. Implemented
94 Riemann solvers include: Lax-Friedrichs [38], Roe [39], and artificially up-
95 stream flux vector splitting scheme (AUFS) [40]. In this work, only the Lax-
96 Friedrichs flux is used. The diffusive fluxes are handled using a symmetric
97 interior penalty (SIP) method [41–43].

98 4. Solution Method

To solve the non-linear set of equations, a damped Newton’s method is
used which has the form:

$$\mathbf{J}^n \Delta a^n = \left[\frac{\mathbf{M}}{\Delta t} + \left(\frac{\partial R}{\partial a} \right)^n \right] \Delta a^n = -R^n \quad (4)$$

where n is the non-linear iteration, $\mathbf{J}$ is the Jacobian matrix, $\mathbf{M}$ is a mass
matrix and Δt is an element-wise time step which is used to modify the
solution of the linear update problem for non-linear continuation purposes
[20, 44, 45]. The mass matrix $\mathbf{M}$, due to the discontinuous basis, only appears
on the block diagonals of the Jacobian matrix. A local time step Δt is set
on every element using

$$\Delta t = \frac{CFL}{h^{-1}(\sqrt{u^2 + v^2 + w^2} + c)}$$

99 where h is a mesh size and c is the speed of sound. The CFL number is
100 not based on an explicit stability limit, but rather is used to control the
101 convergence characteristic of the implicit scheme.

Newton’s method creates a linear system that must be solved to get the update to the coefficients a_{km} as:

$$a^{n+1} = a^n + \sigma \Delta a^n.$$

where σ is an under-relaxation factor which is controlled by specifying a maximum allowable percentage change in pressure and further improved using a line search [45]. To solve the linear system in equation (4), a flexible-GMRES [46] (FGMRES) method is used. To further improve convergence of FGMRES a preconditioner can be applied to the system of equations. In this work a right preconditioner $\mathbf{P}$ is used which takes the form:

$$\mathbf{A}\mathbf{P}^{-1}z = b, \quad z = \mathbf{P}x.$$

102 Preconditioners that have been implemented include Jacobi relaxation, Gauss-
103 Seidel relaxation, line-implicit Jacobi, and ILU(0). In this work, only the
104 ILU(0) and the line-implicit Jacobi preconditioner are used.

105 5. Overset Mesh Assembly

106 The overset grid connectivities are handled through the TIOGA inter-
107 face (Topology Independent Overset Grid Assembler). The TIOGA overset
108 grid assembler relies on an efficient parallel implementation of Alternating
109 Digital Trees (ADT) for point-cell inclusion tests. Multiple grids are loaded
110 in parallel and TIOGA computes the cell donor-receptor information. The
111 receptor cells receive solution data from the donor cells through interpola-
112 tion as shown in Figure 1. In order to interface TIOGA with a high-order
113 method, several callback functions need to be supplied. A callback function
114 is a routine in the flow solver that TIOGA has access to; this allows TIOGA
115 to override internal routines or add functionality. For example, a high-order
116 method needs high-order interpolation, this can be implemented by replacing
117 linear interpolation primitives inside TIOGA with high-order ones through
118 callback functions. This allows high-order interpolation which is required to
119 maintain overall high-order solution accuracy.

120 There are four callback functions that are needed to interface TIOGA
121 with a high-order method. The first callback function is a function which
122 generates a list of receptor points. Typically for finite volume or finite dif-
123 ference discretizations these are the nodes or cell centers in a mesh, but for
124 a high-order DG discretization additional points are needed within each cell.

Figure 1: Receptor and donor cell overset connectivity.

Callback functions	Inputs	Outputs
<code>createReceptorNodes</code>	<code>receptorCellIndex</code>	<code>receptorNodes</code>
<code>donorInclusionTest</code>	<code>donorCellIndex</code> <code>physCoord</code>	<code>passFlag</code> <code>natCoord</code>
<code>interpolationWeights</code>	<code>donorCellIndex</code> <code>natCoord</code>	<code>donorSolnIndex</code> ϕ^d
<code>solutionToCoefficients</code>	<code>receptorCellIndex</code> q	<code>receptorSolnIndex</code> a

Table 1: Callback function inputs and outputs.

125 The second function is a high-order donor cell inclusion test which is nec-
 126 essary for curved elements. Once donor cells are located the third function
 127 returns the weights for the high-order interpolation on the donor cells. The
 128 last callback function converts the interpolated solution at the receptor points
 129 back into solution coefficients or modal coefficients on the receptor cell. An
 130 overview of the inputs and outputs of the four callback functions are given
 131 in Table 1 and each callback function is described in more detail below.

The first callback function creates a list of receptor points interior to
 the cell. The choice of points depends on the implementation of the last
 callback function, which will be discussed subsequently. To find the donor
 cells a callback function for a high-order curved cell inclusion test is used.

TIOGA uses an ADT to search for donor cells and the callback function is used to determine if the receptor point is located inside a donor cell. For straight sided elements TIOGA can handle this process using straight forward geometric inclusion tests. However for a curved element, a specialized callback function is needed. To test if a point is inside an element, the physical coordinates (x, y, z) are converted to natural coordinates (r, s, t) in the mapped space of the standard isoparametric element, as shown in Figure 2. The natural coordinates are found using the following equations:

$$\sum_{m=1}^{n_{map}} \psi_m(r, s, t) b_{1m} = x, \quad \sum_{m=1}^{n_{map}} \psi_m(r, s, t) b_{2m} = y, \quad \sum_{m=1}^{n_{map}} \psi_m(r, s, t) b_{3m} = z,$$

where b are the physical coordinate mapping coefficients, ψ are the mapping basis functions, and n_{map} are the number of mapping coefficients for the element. This gives three non-linear equations with three unknowns. To solve this problem a Newton-Rhapson method is used which can be written as:

$$\begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial s} & \frac{\partial x}{\partial t} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial s} & \frac{\partial y}{\partial t} \\ \frac{\partial z}{\partial r} & \frac{\partial z}{\partial s} & \frac{\partial z}{\partial t} \end{bmatrix} \begin{bmatrix} \delta r \\ \delta s \\ \delta t \end{bmatrix} = - \begin{bmatrix} \sum_{m=1}^{n_{map}} \psi_m(r, s, t) b_{1m} - x \\ \sum_{m=1}^{n_{map}} \psi_m(r, s, t) b_{2m} - y \\ \sum_{m=1}^{n_{map}} \psi_m(r, s, t) b_{3m} - z \end{bmatrix}$$

132 Once the natural coordinates are found it is trivial to test if the point lies
 133 inside the element. For example, a tetrahedra has four faces that are each
 134 associated with a plane that can be tested for inclusion. If any of the following
 135 inequalities in Table 2 are true then the point lies inside the element. In this
 136 table, the parameter ϵ is a small number, if this number approaches zero then
 137 strict inclusion is required. However, if this number is larger, the inclusion
 138 test will be less restrictive and extrapolation (instead of interpolation) is
 139 allowed.

Once a donor cell is found the third callback function creates the interpolation weights. Evaluating the solution basis functions at the natural coordinates found from the inclusion test creates the weights for high-order interpolation on the donor cell. The solution q is passed back to the receptor cell by:

$$q(r, s, t) = \sum_{m=1}^{n_{mode}} \phi_m^d(r, s, t) a_m$$

140 where $\phi_m^d(r, s, t)$ are the donor cell basis functions (where superscript d rep-
 141 resents donor) evaluated at the receptor node locations (r, s, t) , n_{mode} are

Figure 2: Curved tetrahedron in physical coordinates (left) and natural coordinates (right)

tetrahedron	pyramid	prism	hexahedron
$-r - 1 < \epsilon$	$-r - 1 < \epsilon$	$-r - 1 < \epsilon$	$ r - 1 < \epsilon$
$-s - 1 < \epsilon$	$-s - 1 < \epsilon$	$ s - 1 < \epsilon$	$ s - 1 < \epsilon$
$-t - 1 < \epsilon$	$-t - 1 < \epsilon$	$-t - 1 < \epsilon$	$ t - 1 < \epsilon$
$r + s + t < \epsilon$	$r + t < \epsilon$	$r + t < \epsilon$	
	$s + t < \epsilon$		

Table 2: Inclusion test inequalities for each element type.

142 the number of solution basis functions, and a_m are the solution modal coefficients on the donor cell. TIOGA transfers, in parallel, all of the interpolated
143 solutions from the donor cells to the receptor nodes.
144

The fourth callback function converts these nodal solution values back into coefficients on the receptor cells. To convert nodal solution values back into coefficients we investigate two approaches for reconstructing solution coefficient data: the first uses a mass matrix approach and the second a Vandermode matrix approach. One of these two methods is necessary because it is the transfer of solution coefficients rather than the actual solution values that are needed in a modal finite-element method. Both the mass matrix and the Vandermode matrix methods require solution values from the donor cells at specific points on the receptor cell. Figure 3 shows the locations of the receptor nodes for each method. For the mass matrix method the interpolation points correspond to the quadrature points used in the volume integral

Figure 3: Locations of receptor nodes on a quadrilateral element for mass matrix method (left) and Vandermode matrix method (right).

evaluation of the residual, while for the Vandermode matrix method the interpolation points consist of equidistant points. The mass matrix approach starts off with a Galerkin projection:

$$\sum_{n=1}^{n_{mode}} \phi_n^r(\xi_{1k}, \xi_{2k}, \xi_{3k}) a_n = q(\xi_{1k}, \xi_{2k}, \xi_{3k})$$

where ϕ_n^r are the solution receptor cell basis functions (where superscript r represents receptor), n_{mode} is the number of solution receptor cell basis functions, ξ_{ik} are the quadrature points (where the indices $i = 1, 2, 3$ and k runs from one to the number of quadrature points n_{qp}), q are the solution values at the quadrature points, and a_n are the receptor solution coefficients to be solved for. To solve for the coefficients both sides are multiplied by the receptor cell's basis functions and integrated over the cell:

$$\sum_{n=1}^{n_{mode}} \int_{\Omega} \phi_m^r \phi_n^r a_n d\Omega = \int_{\Omega} \phi_m^r q d\Omega.$$

This creates an element mass matrix defined as:

$$M_{mn} = \int_{\Omega} \phi_m^r \phi_n^r d\Omega$$

and the right hand side becomes:

$$f_m = \int_{\Omega} \phi_m^r q d\Omega = \sum_{k=1}^{n_{qp}} \phi_m^r(\xi_{1k}, \xi_{2k}, \xi_{3k}) q(\xi_{1k}, \xi_{2k}, \xi_{3k}) w_k$$

where w_k are the quadrature weights. To solve for the coefficients the right hand side is integrated and the mass matrix is inverted to give:

$$a = \mathbf{M}^{-1}f.$$

145 In the actual implementation the mass matrix is LU factorized ahead of time
146 and only a forward/backward solve is needed to solve for a .

The second approach is the Vandermode method. Again, this starts with a projection operator:

$$\sum_{n=1}^{n_{mode}} \phi_n^r(\zeta_{1m}, \zeta_{2m}, \zeta_{3m}) a_n = q(\zeta_{1m}, \zeta_{2m}, \zeta_{3m})$$

except now the Vandermode matrix:

$$V_{mn} = \phi_n^r(\zeta_{1m}, \zeta_{2m}, \zeta_{3m})$$

is constructed by evaluating the bases at the points ζ_{im} , where the number of points are equal to the number of basis functions, making the Vandermode matrix square. The solution coefficients are solved for by inverting the Vandermode matrix:

$$a = \mathbf{V}^{-1}q$$

147 for the receptor solution coefficients. For efficiency the Vandermode matrix
148 is constructed for every element type and polynomial degree and factorized
149 ahead of time so that only forward/backward solves are needed.

150 There are two downsides to the mass matrix method. The first is that
151 even if the mass matrix is diagonal the right hand side still requires integra-
152 tion with respect to the basis. The second is that, for tetrahedra, pyramids,
153 and prisms, the number of quadrature points is greater than the number
154 of basis modes and thus more receptor points must be created in each cell
155 and more data must be transferred at each interpolation phase. Also, for
156 curved elements more quadrature points are needed making the mass ma-
157 trix approach even more costly. Therefore, the mass matrix method requires
158 more computational work and more communication than the Vandermode
159 matrix method. However, a benefit of the mass matrix method is that it
160 gives a more optimal approximation to the interpolated values compared to
161 the Vandermode approach using equidistant points. The Vandermode matrix
162 at equidistant points can suffer from interpolation errors at high polynomial

degrees. As a side note: for the Vandermode matrix method the points do not have to be equidistant and more optimal interpolation points do exist, the only requirement for this method is that the number of points equals the number of modes in the basis and each point is unique. However, optimization of the points in the Vandermode matrix method has not been attempted in this work.

6. Implicit overset solver

The implicit overset solver requires tight coupling between grids to achieve full Newton convergence. This is accomplished by communicating data using TIOGA both during non-linear iterations and linear iterations. The DG method only requires communication through faces [30]. However, we will show that if cell-based communication is used, tight coupling of the implicit-solver is trivial to implement. Furthermore, it will be shown that cell-based communication is not as costly as typically argued [22].

First, we will justify the cost of a cell-based communication data structure. For cell-based communication only the solution coefficients are passed across processors from real to ghost cells. A hexahedral element contains $(p+1)^3$ modes whereas a tetrahedral element contains $(p+1)(p+2)(p+3)/6$ modes, where p is the polynomial degree of the basis. A hexahedral element contains the most number of modes and a tetrahedral element contains the least number of modes of the four element types used in this discretization. The total number of modes for any three-dimensional element is $\sim p^3$. To communicate through a face only the values at the quadrature points are needed. The number of quadrature points on a face scales as $\sim p^2$. However, communicating through a face requires both the solution value and the derivatives of the solution (only for viscous terms). By comparing the ratio of cell communication to face communication on a hexahedral element:

$$\frac{(p+1)^3}{4(p+1)^2} = \frac{p+1}{4}$$

where the numerator is the total number of volume modal coefficients, the denominator is the total number of face quadrature points, and the 4 comes from the solution and the 3 solution gradients. For this integration rule the ratio becomes unity at $p = 3$. Some might argue that a face integration requires an extra quadrature point in each direction due to the higher degree

polynomial basis of the test function ($\int \phi F^*$ for the face compared to $\int \frac{\partial \phi}{\partial x} F$ for the volume). Therefore the denominator is modified to become:

$$\frac{(p+1)^3}{4(p+2)^2}$$

where the crossover point is now between $p = 4$ and $p = 5$. Lately, practitioners of DG have further increased the number of quadrature points by as much as a factor of two in each spatial direction [47]. This is to prevent aliasing due to under-integration of the inviscid terms in Navier-Stokes along with the high-order curvature of cells. If doubling of quadrature points is taken into consideration then the ratio becomes:

$$\frac{(p+1)^3}{4(2(p+1))^2} = \frac{p+1}{16}$$

177 making the crossover point $p = 15$.

178 For a tetrahedral element all of the faces are triangles. Analytic formulas
179 are not presented because the number of quadrature points is dependent on
180 the chosen quadrature rule. Using a set of quadrature rules that uses the
181 minimum number of quadrature points for a given polynomial degree, the
182 cross over point is near $p = 11$ for a standard integration. If however, double
183 the quadrature points are used then the crossover point is further increased.
184 Choosing between cell and face communication will always depend on the the
185 polynomial degree. Since we want to run simulations in the range of $p = 1$
186 to $p = 4$ we choose cell communication over face communication.

187 During the non-linear solution update either cell or face communication
188 can be used. This is because the residual and its linearization (Jacobian ma-
189 trix) can be created using either ghost cell coefficients or face values (along
190 with derivatives). To perform a non-linear update Newton's method is used,
191 which requires the solution to a linear system. The linear solver and its
192 matrix preconditioners require matrix-vector products. These matrix-vector
193 products operate on the Jacobian matrix which requires a vector of coef-
194 ficients. If cell-based communication is used in TIOGA this step becomes
195 trivial. The same communication framework that passes solution coefficients
196 can also be used to pass any cell-based vector of coefficients. Algorithm 1
197 outlines the procedure to perform an implicit solver in an overset framework.
198 In this algorithm Jacobi's method is used to solve the linear system and calls
199 to the routine *tiogaDataUpdate()* are needed to communicate the solution
200 update Δa . In the actual implementation: FGMRES is the linear solver and
201 line-implicit Jacobi or ILU(0) are used as preconditioners.

Algorithm 1 Implicit overset solver

```
for  $n = 1 \rightarrow$  non-linear iterations do
  call tiogaDataUpdate( $a^n$ )
  call  $[\mathbf{D}, \mathbf{J}, R] = \text{getResidualJacobian}(a^n)$ 
   $\Delta a = 0$ 
  for  $i = 1 \rightarrow$  linear iterations do
     $\Delta a \leftarrow \Delta a - \mathbf{D}^{-1}(\mathbf{J}\Delta a - R)$ 
    call tiogaDataUpdate( $\Delta a$ )
  end for
  call  $\sigma = \text{underRelaxationFactor}(a, \Delta a)$ 
   $a^{n+1} = a^n + \sigma \Delta a$ 
end for
```

202 7. Fringe removal

203 A high-order, finite-difference method uses a large stencil which requires
204 many fringe points. The fringe points constitute a set of points that have
205 solutions from other grids interpolated onto them. Sometimes it may be
206 difficult to obtain enough fringe points to complete the stencil and maintain
207 accuracy. A benefit to using a DG discretization is that only a nearest
208 neighbor stencil is required. Since DG is a cell-based scheme we will discuss
209 the use of fringe cells instead of fringe points. TIOGA was originally designed
210 for finite difference discretizations and may return too many fringe cells.
211 To minimize communication between grids these extra fringe cells can be
212 removed. An example of an overset grid with too many fringe cells is shown
213 in Figure 4.

214 To simplify the fringe removal discussion we use the notation IBLANK
215 which is an array of integers used to identify cell type in the overset frame-
216 work. In this notation a fringe cell has IBLANK value of: $\text{IBLANK} = -1$,
217 a hole cell: $\text{IBLANK} = 0$, and a field cell: $\text{IBLANK} = 1$. Hole cells occur
218 when an outer mesh overlaps with an inner mesh with an enclosed wall-
219 boundary surface (such as the cylinder in Figure 4). TIOGA tags any cell
220 that intersects or is contained within the wall-boundary manifold as a hole
221 cell, which are subsequently removed from the discretization. The minimum
222 requirement for ensuring complete overset connectivity is that the hole cells
223 need to be completely surrounded by fringe cells. This in turn means that the
224 field cells only interact with fringe cells. If a field cell neighbors a hole cell
225 then the overset connectivity fails. It is apparent that the set of hole-cells

Figure 4: Overset grid for cylinder and background.

provided by TIOGA can be expanded further, i.e. extraneous fringe cells can be converted to hole cells, while still meeting the minimum requirement for complete overset connectivity.

To remove extraneous fringe cells we create a second IBLANK cell array. This array is initialized to 0, which is the value that corresponds to a hole located at each cell. Next we loop through faces and check the left and right cell IBLANK values from the original IBLANK array. If a field cell is next to a field or fringe cell then we set the new IBLANK array value to the original IBLANK value. After this process, the new IBLANK array now only contains the field and fringe cells. Figure 5 shows the background mesh IBLANK values from TIOGA and the IBLANK values after the fringe cells are removed. As shown in the figure there is an inner layer of fringe cells that are converted to hole cells. Figure 6 shows the cylinder near body mesh before and after the fringe removal. An outer layer of fringe cells are converted to hole cells. Even if there is not a hole in the domain this will force TIOGA to ignore these cells.

8. Results

In this section six simulations are discussed. The first is a mesh resolution study to verify that the order of accuracy of the DG method is maintained on an overset grid. The second is an unsteady simulation to determine the accuracy of the mass matrix versus the Vandermode matrix methods. The third is a simulation of flow over a cylinder to test the implicit non-linear solver performance on overset grids. The fourth is a pitching airfoil simulation

Figure 5: Initial cell IBLANK (top) and cell IBLANK after fringe removal (bottom) for background mesh.

249 demonstrating overset capability on moving mesh problems. The last two
 250 simulations are large scale simulations using the implicit solver to solve an
 251 unsteady viscous flow and a steady-state Reynolds-Averaged Navier-Stokes
 252 (RANS) simulation.

253 8.1. Verification using Ringleb flow

254 Ringleb flow is a two dimensional exact solution to the Euler equations
 255 [48]. This exact solution can be used to verify the order of accuracy of the
 256 overset implementation. Figure 7 shows contours of pressure on an overset
 257 grid. The overset grid in this case uses hexahedra for the background grid
 258 and prisms for the interior grid, extending into the third dimension by one
 259 cell. This figure shows that smooth contours of pressure are retained through
 260 the overlap regions.

To test the order of accuracy of the overset implementation the L_2 error

Figure 6: Initial cell IBLANK (top) and cell IBLANK after fringe removal (bottom) for cylinder mesh.

norm is used:

$$L_2 = \frac{[\sum_{i=1}^{n_{cell}} \int_{\Omega} (u - u_e)^2 d\Omega]^{1/2}}{\sum_{i=1}^{n_{cell}} V_i}.$$

261 This measures the error between the finite element solution u and the exact
 262 solution u_e , where V is the volume of each cell. Figure 8 shows the L_2 error
 263 for four sets of grids and polynomial degrees ranging from first order ($p = 0$)
 264 to fifth order ($p = 4$). The Figure also contains single grid simulations for
 265 polynomial degrees ranging from $p = 0$ to $p = 2$. The overset grid simulations
 266 match closely in value and slope with the single grid simulations. For a finite
 267 element method error should decrease as Δh^{p+1} , where Δh is some measure
 268 of the cell size. The slopes are reported in Figure 8 and are close to the value
 269 of $p + 1$ demonstrating that the overset grid implementation maintains the
 270 correct order of accuracy.

271 The order of accuracy is maintained because the interpolation accuracy

Figure 7: Pressure contours for Ringleb flow on an overset grid.

272 matches the design accuracy of the finite element method. Figure 9 shows
 273 what happens if the interpolation accuracy is reduced from $p = 2$ to $p = 1$.
 274 The mixed approach has a slope that has deteriorated from 2.82 to 1.75 as
 275 the error becomes dominated by the interpolation procedure in this case.

276 8.2. Isentropic vortex

277 In this section the accuracy of the Vandermode matrix method and the
 278 mass matrix method for an overset mesh configuration are compared with
 279 each other and to a single grid implementation. An explicit unsteady simu-
 280 lation is carried out using a fourth-order Runge-Kutta time stepping scheme
 281 (RK4) and an isentropic vortex is advected through the domain. The vortex
 282 problem was chosen because the exact solution in space and time is known.
 283 The overset grid consists of two hexahedral grids (one element thick) where
 284 the finer inner grid is rotated. The grids are shown in Figure 10 along with
 285 contours of pressure after the vortex exits the overset region.

Figure 8: L_2 error for Ringleb flow on an overset grid for various grid sizes and polynomial degrees

Figure 11 shows the L_2 error for a $p = 0$ simulation. For this case the Vandermode and mass matrix strategies give identical results. This is because both methods use a single point in the center of the cell. Compared to the single grid result there is increased error in the grid overlap fringe region and decreased error in the inner refined overset mesh region. After the vortex exits the overset region the overset approach outperforms the single grid simulation.

Figure 12 shows the L_2 error for a $p = 1$ simulation. For this case the mass matrix method is more accurate than the Vandermode matrix method. Again, there is increased error in the grid overlap fringe region and decreased error in the inner refined overset mesh region. The overset simulation generally outperforms the single grid simulation. The overall error has dropped by around half compared to the $p = 0$ simulation.

Figure 13 shows the L_2 error for a $p = 2$ simulation. For this case the Vandermode matrix is more accurate than the mass matrix method. There is increased error in the grid overlap fringe region and decreased error in the inner refined overset mesh region. However for this case the single grid simulation outperforms the overset grid simulation. The overall error has dropped by around an order of magnitude compared to the $p = 1$ simulation.

Figure 14 shows the L_2 error for a $p = 3$ simulation. For this case the Vandermode matrix is more accurate than the mass matrix method. There

Figure 9: L_2 error for Ringleb flow on an overset grid with and without matching interpolation accuracy with solution accuracy

is increased error in the grid overlap fringe region and decreased error in the inner refined overset mesh region. Like the $p = 2$ case, the single grid simulation outperforms the overset grid simulation. The overall error has dropped by another order of magnitude compared to the $p = 2$ simulation.

Although, the DG method used in this work is a conservative formulation, the use of overset grids can affect the conservation properties of the method. Mass conservation properties for steady state overset problems have been studied extensively for DG; for these cases even though there is a loss of conservation the magnitude of this loss diminishes when the mesh is refined or the polynomial degree is increased [30]. Mass conservation for the unsteady isentropic vortex problem will be investigated in this work. For this case the vortex is initialized in the refined region and advects through the domain multiple times using periodic boundary conditions. To calculate the mass conservation the density is integrated over all the valid cells (IBLANK=1) and divided by the total volume of the valid cells of the two grids. Figure 15 shows the mass conservation for the isentropic vortex for a single grid and an overset grid. The single grid, as expected, conserves mass to machine precision for every polynomial degree. The $p = 0$ simulation initially does not conserve mass, however, as the vortex dissipates and the flow reverts to freestream, the mass is conserved. The $p = 1$ simulation does not dissipate as quickly as the $p = 0$ simulation and the mass varies over time.

Figure 10: Contours of pressure for the isentropic vortex on an overset mesh.

328 The mass matrix approach conserves mass more accurately than the Van-
 329 dermode approach. The $p = 2$ simulation is a slight improvement for mass
 330 conservation compared to $p = 1$, however it is not clear whether the mass ma-
 331 trix or Vandermode approach is better. Also, the $p = 2$ simulation loses mass
 332 similar to $p = 1$ but also gains more mass than the initial mass as the vortex
 333 moves through the refined region. Overall, the loss of mass conservation is
 334 minimal for these simulations but it does allow us to compare the different
 335 polynomial degrees and Vandermode versus mass matrix approaches to each
 336 other. Further work is needed to understand the effect overset grids have on
 337 mass conservation when a boundary or higher compressibility is taken into
 338 consideration for unsteady problems.

339 8.3. Inviscid cylinder implicit solver performance

340 This section demonstrates the performance of the implicit solver on both
 341 a single grid and an overset grid configuration. The simulation consists of
 342 inviscid flow over a cylinder at $Mach = 0.3$ using a $p = 1$ polynomial de-
 343 gree basis (second order accuracy). Figure 16 shows contours of pressure
 344 along with the overset grid used. Both the inner and outer grids contain

Figure 11: L_2 Error for the isentropic vortex, using $p = 0$ basis, with Vandermode overset approach, mass matrix overset approach, and single grid.

345 hexahedral elements (one element thick in the span direction). The single
 346 grid simulation only consists of the inner mesh, which is not large enough
 347 for accurate solutions. However for this demonstration, only the non-linear
 348 convergence is of importance. Figure 17 shows the non-linear convergence
 349 of both the single and overset grid configurations. Both implicit solves take
 350 exactly 25 non-linear iterations to reach a tolerance of 1×10^{-12} , however
 351 the non-linear solution path is slightly different. This is to be expected since
 352 the domains are relatively small and the overset connectivity happens near
 353 the cylinder. Both simulations are converged two orders of magnitude using
 354 FGMRES with line-implicit Jacobi smoothing as a preconditioner. The
 355 preconditioner has a tolerance of two orders of magnitude or a max of 200
 356 smoothing steps. The total number of linear iterations for the single grid is
 357 69 and for the overset grid is 72 iterations. This case demonstrates that the
 358 overset grid simulation converges similarly to the single grid simulation and
 359 both obtain quadratic convergence. Quadratic convergence is typically only
 360 observed in Newton's method if an exact linearization is used which can only
 361 occur with tight coupling between overset domains.

Figure 12: L_2 Error for the isentropic vortex, using $p = 1$ basis, with Vandermode oversight approach, mass matrix oversight approach, and single grid.

362 8.4. Pitching airfoil

One of advantages of overset grids is the ability to move bodies in relative motion without the need to re-generate a new grid at every time step. To demonstrate the accuracy of a moving grid in an overset framework a simulation of a pitching NACA 0012 airfoil is carried out [49]. The airfoil pitches at the quarter chord with a time dependent angle given by the formula:

$$\alpha(t) = 2.5 \sin(\omega t) + \sin(7\omega t)$$

where ω is the angular frequency defined by:

$$\omega = \frac{2U_\infty k_c}{c}$$

363 The freestream velocity is $U_\infty = 0.5$, the reduced frequency $k_c = 0.02$, and
 364 the chord length $c = 1$. The time derivative is discretized using the second
 365 order backwards difference formula and the physical time step is $\Delta t = 1/2$.
 366 In space a $p = 1$ polynomial basis is used giving second order accuracy. The
 367 flow is assumed inviscid and the flux is modified with arbitrary Lagrangian-
 368 Eulerian (ALE) terms to account for the rigid movement of the grid.

369 To demonstrate the accuracy of the pitching airfoil with overset grids the
 370 simulation is compared with a single rigid moving grid simulation. Figure 18

Figure 13: L_2 Error for the isentropic vortex, using $p = 2$ basis, with Vandermode overset approach, mass matrix overset approach, and single grid.

shows the overset grid connectivity. The blue grid is used in both the rigid single grid case and the overset grid case. The grid extends to a radius of 50 chord lengths and is comprised of one element thick prisms. The outer green grid is used in the overset grid case only and it extends to a radius of 60 chord lengths. The outer grid conforms to a slightly larger domain and is slightly finer near the airfoil so that the overset connectivity occurs relatively close to the airfoil. The overlap in the overset connectivity of these grids is large due to the inner grid not being trimmed near the airfoil and the inner and outer grid having similar mesh resolution. Figure 19 shows the coefficient of lift and moment over non-dimensional time for both the rigid and overset grid simulations. The two simulations closely match each other and make it hard to distinguish between the two. To help show the differences the absolute difference of coefficient of lift, moment, and drag between the two simulations are shown on a log scale in Figure 20.

8.5. Unsteady viscous simulation of a sphere

Unsteady viscous flow around a sphere is solved to demonstrate a large scale simulation using an overset framework. The overset grid consists of a prismatic strand mesh [50] for the sphere, a structured hexahedral wake mesh surrounding the sphere, and an unstructured tetrahedral outer domain mesh as shown in Figure 21. At a polynomial degree of $p = 2$, which gives

Figure 14: L_2 Error for the isentropic vortex, using $p = 3$ basis, with Vandermode overset approach, mass matrix overset approach, and single grid.

391 third order accuracy, this simulation contains 6.8×10^6 degrees of freedom per
 392 equation. This simulation is solved on 208 processors. The time derivative is
 393 discretized using a second order backwards difference and the flow conditions
 394 are $Re = 1000$, and $Mach = 0.3$. Only qualitative results are shown in the
 395 form of contours of temperature in Figure 22. This simulation is meant to
 396 demonstrate the large scale capability and mixed element implementation of
 397 the overset framework.

398 8.6. Turbulent simulation of the DPW3 wing

399 A Reynolds-Averaged Navier-Stokes (RANS) simulation is performed on
 400 a wing configuration taken from the third AIAA Drag Prediction Workshop
 401 (DPW) test cases [51]. The turbulence model used in this work is the negative
 402 variant of the Spalart-Allmaras (SA) model [34]. The overset grid consists of
 403 an unstructured hexahedral wing mesh and a refined structured hexahedral
 404 wake mesh as shown in Figure 23. The flow conditions are $Re = 5 \times 10^6$,
 405 $Mach = 0.76$, and an angle of attack $\alpha = 0.5$. The polynomial degree is
 406 $p = 2$ (third order accuracy) giving a total of 2.7×10^6 degrees of freedom.
 407 This simulation demonstrates that the wake mesh can resolve the tip-vortex
 408 generated on the wing as shown in Figure 24. This figure shows contours
 409 of eddy viscosity and it is important to note that the rotation curvature
 410 correction to the SA model is not used in this simulation. The coefficient of

Figure 15: Mass conservation for the isentropic vortex with Vandermode overset approach, mass matrix overset approach, and single grid.

drag is $C_D = .001955$ and the lift is $C_L = 0.4781$ which are within the range of solutions found at the drag prediction workshop [51].

9. Conclusions

In this work a high-order discontinuous Galerkin method is used in an overset framework to solve three-dimensional viscous flows on hybrid unstructured meshes. The overset domain connectivity is handled using a Topology Independent Overset Grid Assembler (TIOGA). To enable high-order capabilities within TIOGA high-order call back functions are added to the DG code. This increases the accuracy of the interpolation and also allows for inclusion tests on curved cells within TIOGA. The solver has been validated by showing that an overset grid simulation can maintain the same order of accuracy compared to an equivalent single grid calculation. The accuracy

Figure 16: Overset mesh and contours of pressure for flow over a cylinder.

423 of two interpolation schemes has been investigated and shows that the Van-
 424 dermode matrix approach gives lower error than a mass matrix approach at
 425 moderate polynomial degrees. Also, the Vandermode approach is more com-
 426 putationally efficient compared to the mass matrix approach. The implicit
 427 solver is demonstrated on flow over a cylinder and full Newton convergence
 428 is achieved which closely follows a single grid simulation. Mesh movement is
 429 also demonstrated to give accurate solutions within the overset framework.
 430 Also, using this solver, complex, large scale, simulations have been carried
 431 out. This work constitutes the first step in an effort to create a DG solver
 432 that can be used to solve problems with bodies in relative motion and be
 433 combined with an efficient off-body solver.

434 10. Acknowledgments

435 This work was funded by ONR Grant N00014-14-1-0045.

Figure 17: Non-linear convergence of flow over a cylinder for a single grid and overset grid configuration using a $p = 1$ basis (second order accurate).

References

- [1] W. H. Reed, T. R. Hill, Triangular mesh methods for the neutron transport equation, Tech. Rep. LA-UR-73-479, Los Alamos Scientific Laboratory (1973).
- [2] B. Cockburn, G. E. Karniadakis, C.-W. Shu, The development of discontinuous Galerkin methods, Springer, 2000.
- [3] T. Warburton, I. Lomtev, Y. Du, S. Sherwin, G. Karniadakis, Galerkin and discontinuous Galerkin spectral/hp methods, Computer methods in applied mechanics and engineering 175 (3) (1999) 343–359.
- [4] S. J. Sherwin, G. E. Karniadakis, A new triangular and tetrahedral basis for high-order (hp) finite element methods, International Journal for Numerical Methods in Engineering 38 (22) (1995) 3775–3802.

Figure 18: Overset mesh for pitching airfoil

- 448 [5] B. Cockburn, C.-W. Shu, Runge–Kutta discontinuous Galerkin meth-
 449 ods for convection-dominated problems, *Journal of scientific computing*
 450 16 (3) (2001) 173–261.
- 451 [6] B. Cockburn, C.-W. Shu, The local discontinuous Galerkin method for
 452 time-dependent convection-diffusion systems, *SIAM Journal on Numer-
 453 ical Analysis* 35 (6) (1998) 2440–2463.
- 454 [7] B. Cockburn, C.-W. Shu, TVB Runge-Kutta local projection discontin-
 455 uous Galerkin finite element method for conservation laws. ii. general
 456 framework, *Mathematics of computation* 52 (186) (1989) 411–435.
- 457 [8] B. Cockburn, S.-Y. Lin, C.-W. Shu, TVB Runge-Kutta local projec-
 458 tion discontinuous Galerkin finite element method for conservation laws

Figure 19: Coefficient of Lift and Moment for rigid and overset pitching airfoil.

- 459 iii: one-dimensional systems, *Journal of Computational Physics* 84 (1)
460 (1989) 90–113.
- 461 [9] B. Cockburn, C.-W. Shu, The Runge–Kutta discontinuous Galerkin
462 method for conservation laws v: multidimensional systems, *Journal of*
463 *Computational Physics* 141 (2) (1998) 199–224.
- 464 [10] B. Cockburn, S. Hou, C.-W. Shu, The Runge-Kutta local projection
465 discontinuous Galerkin finite element method for conservation laws. iv.
466 the multidimensional case, *Mathematics of Computation* 54 (190) (1990)
467 545–581.
- 468 [11] B. Van Leer, Towards the ultimate conservative difference scheme. ii.
469 monotonicity and conservation combined in a second-order scheme,
470 *Journal of computational physics* 14 (4) (1974) 361–370.
- 471 [12] B. Van Leer, Towards the ultimate conservative difference scheme iii.
472 upstream-centered finite-difference schemes for ideal compressible flow,
473 *Journal of Computational Physics* 23 (3) (1977) 263–275.
- 474 [13] B. Van Leer, Towards the ultimate conservative difference scheme. iv. a
475 new approach to numerical convection, *Journal of computational physics*
476 23 (3) (1977) 276–299.
- 477 [14] B. Van Leer, Towards the ultimate conservative difference scheme. v.

Figure 20: Absolute difference of Lift, Drag, and Moment coefficients for rigid and overset pitching airfoil.

- 478 a second-order sequel to Godunov's method, *Journal of computational*
479 *Physics* 32 (1) (1979) 101–136.
- 480 [15] F. Bassi, S. Rebay, High-order accurate discontinuous finite element
481 solution of the 2d Euler equations, *Journal of computational physics*
482 138 (2) (1997) 251–285.
- 483 [16] F. Bassi, S. Rebay, A high-order accurate discontinuous finite element
484 method for the numerical solution of the compressible Navier–Stokes
485 equations, *Journal of computational physics* 131 (2) (1997) 267–279.
- 486 [17] K. S. Bey, J. T. Oden, hp-version discontinuous Galerkin methods for
487 hyperbolic conservation laws, *Computer Methods in Applied Mechanics*
488 and *Engineering* 133 (3) (1996) 259–286.
- 489 [18] P.-O. Persson, J. Peraire, Sub-cell shock capturing for discontinu-

Figure 21: Overset mesh for sphere strand mesh (left), wake and sphere strand mesh (center), and the previous two meshes including the sphere outer mesh (right)

Figure 22: Contours of temperature for flow over sphere, $p = 2$, $Re = 1000$, $Mach = 0.3$

- ous Galerkin methods, Collection of Technical Papers - 44th AIAA
Aerospace Sciences Meeting 2 (2006) 1408 – 1420.
- [19] G. Barter, D. Darmofal, Shock capturing with PDE-based artificial vis-
cosity for DGFEM: Part I. formulation, J. Comput. Phys. (USA) 229 (5)
(2010/03/01) 1810 – 27.
URL <http://dx.doi.org/10.1016/j.jcp.2009.11.010>
- [20] N. K. Burgess, D. J. Mavriplis, An hp-adaptive discontinuous Galerkin
solver for aerodynamic flows on mixed-element meshes, 49th AIAA
Aerospace Sciences Meeting Including the New Horizons Forum and
Aerospace Exposition.
- [21] C. C. de Wiart, K. Hillewaert, L. Bricteux, G. Winckelmans, Implicit

Figure 23: DPW3 wing and wake mesh

- 501 les of free and wall-bounded turbulent flows based on the discontin-
 502 uous Galerkin/symmetric interior penalty method, International Jour-
 503 nal for Numerical Methods in Fluids 78 (6) (2015) 335–354, fld.4021.
 504 doi:10.1002/fld.4021.
 505 URL <http://dx.doi.org/10.1002/fld.4021>
- 506 [22] F. Hindenlang, G. J. Gassner, C. Altmann, A. Beck, M. Stau-
 507 denmaier, C.-D. Munz, Explicit discontinuous Galerkin methods
 508 for unsteady problems, Computers & Fluids 61 (2012) 86–93.
 509 doi:<http://dx.doi.org/10.1016/j.compfluid.2012.03.006>.
 510 URL <http://www.sciencedirect.com/science/article/pii/S004579301200093X>
- 511 [23] M. J. Brazell, D. J. Mavriplis, 3D Mixed Element Discontinuous
 512 Galerkin with Shock Capturing, American Institute of Aeronautics and
 513 Astronautics, 2013. doi:10.2514/6.2013-3064.
 514 URL <http://dx.doi.org/10.2514/6.2013-3064>
- 515 [24] V. Sankaran, A. Wissink, A. Datta, J. Sitaraman, M. Potsdam, B. Ja-
 516 yaraman, A. Katz, S. Kamkar, B. Roget, D. Mavriplis, H. Saberi, W.-B.
 517 Chen, W. Johnson, R. Strawn, Overview of the Helios Version 2.0 Com-

Figure 24: Contours of eddy viscosity over DPW3 wing, $p = 2$, $Re = 5 \times 10^6$, $Mach = 0.76$, $\alpha = 0.5$

- 518 putational Platform for Rotorcraft Simulations, American Institute of
519 Aeronautics and Astronautics, 2011. doi:doi:10.2514/6.2011-1105.
520 URL <http://dx.doi.org/10.2514/6.2011-1105>
- 521 [25] D. J. Mavriplis, Three-dimensional multigrid Reynolds-averaged Navier-
522 Stokes solver for unstructured meshes, *AIAA Journal* 33 (3) (1995) 445–
523 453. doi:10.2514/3.12597.
524 URL <http://dx.doi.org/10.2514/3.12597>
- 525 [26] R. Hornung, A. Wissink, S. Kohn, Managing complex data and geom-
526 etry in parallel structured amr applications 22 (3-4) (2006) 181–195.
527 doi:10.1007/s00366-006-0038-6.
528 URL <http://dx.doi.org/10.1007/s00366-006-0038-6>
- 529 [27] T. H. Pulliam, Euler and thin-layer Navier-Stokes codes: ARC2D, and
530 ARC3D, in: *Notes for Computational Fluid Dynamics Users Workshop*,
531 The University of Tennessee Space Institute, Tullahoma, Tennessee, 1984.
- 532 [28] D. J. Mavriplis, Third drag prediction workshop results using the nsu3d
533 unstructured mesh solver, *Journal of Aircraft* 45 (3) (2008) 750–761.

- doi:10.2514/1.29828.
 URL <http://arc.aiaa.org/doi/abs/10.2514/1.29828>
- [29] C. Nastase, D. Mavriplis, J. Sitaraman, An Overset Unstructured Mesh Discontinuous Galerkin Approach for Aerodynamic Problems, American Institute of Aeronautics and Astronautics, 2011. doi:doi:10.2514/6.2011-195.
 URL <http://dx.doi.org/10.2514/6.2011-195>
- [30] M. C. Galbraith, J. A. Benek, P. D. Orkwis, M. G. Turner, A discontinuous Galerkin chimera scheme, Computers & Fluids 98 (0) (2014) 27 – 53. doi:<http://dx.doi.org/10.1016/j.compfluid.2014.03.014>.
 URL <http://www.sciencedirect.com/science/article/pii/S0045793014001133>
- [31] B. Zhang, C. Liang, A Simple, Efficient, High-Order Accurate Sliding-mesh Interface Approach to FR/CPR Method on Coupled Rotating and Stationary Domains, American Institute of Aeronautics and Astronautics, 2015. doi:doi:10.2514/6.2015-1742.
 URL <http://dx.doi.org/10.2514/6.2015-1742>
- [32] C. Liu, J. C. Newman, W. K. Anderson, Petrov–Galerkin overset grid scheme for the Navier–Stokes equations with moving domains, AIAA Journal (2015) 1–16doi:10.2514/1.J053925.
 URL <http://dx.doi.org/10.2514/1.J053925>
- [33] B. E. Merrill, Y. T. Peet, P. F. Fischer, J. W. Lottes, A spectrally accurate method for overlapping grid solution of incompressible Navier–Stokes equations, Journal of Computational Physics 307 (2016) 60 – 93. doi:<http://dx.doi.org/10.1016/j.jcp.2015.11.057>.
 URL <http://www.sciencedirect.com/science/article/pii/S0021999115008050>
- [34] S. Allmaras, F. Johnson, P. Spalart, Modifications and clarifications for the implementation of the Spalart–Allmaras turbulence model, in: 7th International Conference on Computational Fluid Dynamics, 2012.
- [35] M. J. Brazell, D. J. Mavriplis, High-Order Discontinuous Galerkin Mesh Resolved Turbulent Flow Simulations of a NACA 0012 Airfoil (Invited), American Institute of Aeronautics and Astronautics, 2015. doi:doi:10.2514/6.2015-1529.
 URL <http://dx.doi.org/10.2514/6.2015-1529>

- [36] M. J. Brazell, D. J. Mavriplis, J. Sitaraman, An Overset Mesh Approach for 3D Mixed Element High Order Discretizations, American Institute of Aeronautics and Astronautics, 2015. doi:doi:10.2514/6.2015-1739.
URL <http://dx.doi.org/10.2514/6.2015-1739>
- [37] R. Cools, An encyclopaedia of cubature formulas, Journal of complexity 19 (3) (2003) 445–453.
- [38] P. D. Lax, Weak solutions of nonlinear hyperbolic equations and their numerical computation, Communications on Pure and Applied Mathematics 7 (1) (1954) 159–193. doi:10.1002/cpa.3160070112.
URL <http://dx.doi.org/10.1002/cpa.3160070112>
- [39] P. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, J. Comput. Phys. (USA) 43 (2) (1981/10/) 357 – 72.
- [40] M. Sun, K. Takayama, An artificially upstream flux vector splitting scheme for the Euler equations, J. Comput. Phys. (USA) 189 (1) (2003/07/20) 305 – 29.
URL [http://dx.doi.org/10.1016/S0021-9991\(03\)00212-2](http://dx.doi.org/10.1016/S0021-9991(03)00212-2)
- [41] R. Hartmann, P. Houston, An optimal order interior penalty discontinuous Galerkin discretization of the compressible Navier-Stokes equations, J. Comput. Phys. (USA) 227 (22) (2008/11/20) 9670 – 85.
URL <http://dx.doi.org/10.1016/j.jcp.2008.07.015>
- [42] K. Shahbazi, D. Mavriplis, N. Burgess, Multigrid algorithms for high-order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations, J. Comput. Phys. (USA) 228 (21) (2009/11/20) 7917 – 40.
URL <http://dx.doi.org/10.1016/j.jcp.2009.07.013>
- [43] M. Drosson, K. Hillewaert, On the stability of the symmetric interior penalty method for the Spalart–Allmaras turbulence model, Journal of Computational and Applied Mathematics 246 (2013) 122 – 135. doi:<http://dx.doi.org/10.1016/j.cam.2012.09.019>.
URL <http://www.sciencedirect.com/science/article/pii/S0377042712003834>
- [44] T. S. Coffey, C. T. Kelley, D. E. Keyes, Pseudotransient continuation and differential-algebraic equations, SIAM J. Sci. Comput. 25 (2) (2003)

- 553–569. doi:10.1137/S106482750241044X.
 URL <http://dx.doi.org/10.1137/S106482750241044X>
- [45] M. Ceze, K. J. Fidkowski, Constrained pseudo-transient continuation, *International Journal for Numerical Methods in Engineering* 102 (11) (2015) 1683–1703, nme.4858. doi:10.1002/nme.4858.
 URL <http://dx.doi.org/10.1002/nme.4858>
- [46] Y. Saad, A flexible inner-outer preconditioned GMRES algorithm, *SIAM J. Sci. Comput.* 14 (2) (1993) 461–469. doi:10.1137/0914028.
 URL <http://dx.doi.org/10.1137/0914028>
- [47] G. J. Gassner, A. D. Beck, On the accuracy of high-order discretizations for underresolved turbulence simulations, *Theoretical and Computational Fluid Dynamics* 27 (3-4) (2013) 221–237.
- [48] F. Ringleb, Exakte loesungen der differentialgleichungen einer adiabatischen gasstroemung, *A. Angew. Math. Mech.* 20 (4) (1940) 185–198.
- [49] N. L. Mundis, D. J. Mavriplis, GMRES applied to the Time Spectral and Quasi-periodic Time Spectral Methods, *AIAA Paper* 2013-3084, 21st AIAA Computational Fluid Dynamics Conference, 2013. doi:doi:10.2514/6.2013-3084.
 URL <http://dx.doi.org/10.2514/6.2013-3084>
- [50] R. Meakin, A. Wissink, W. Chan, S. Pandya, On Strand Grids for Complex Flows, *American Institute of Aeronautics and Astronautics*, 2007. doi:doi:10.2514/6.2007-3834.
 URL <http://dx.doi.org/10.2514/6.2007-3834>
- [51] J. C. Vassberg, E. N. Tinoco, M. Mani, O. P. Brodersen, B. Eisfeld, R. A. Wahls, J. H. Morrison, T. Zickuhr, K. R. Laflin, D. J. Mavriplis, Abridged summary of the third AIAA computational fluid dynamics drag prediction workshop, *Journal of Aircraft* 45 (3) (2008) 781–798. doi:10.2514/1.30572.
 URL <http://arc.aiaa.org/doi/abs/10.2514/1.30572>