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

Michael J. Brazell [*] Jay Sitaraman [†] Dimitri J. Mavriplis [‡]

A parallel high-order Discontinuous Galerkin (DG) method is used to solve the compressible Navier-Stokes equations on an overset mesh framework. The DG solver has many capabilities including: *hp*-adaption, curved cells, and support for hybrid, mixed element meshes. Combining these capabilities with overset grids allows the DG solver to be used in problems with bodies moving in relative motion to be combined 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, *hp*-adaption, and large scale parallel simulations in an overset framework.

I. Introduction

Achieving higher accuracy and fidelity in aerodynamic simulations using higher-order methods has received significant attention over the last decade. High-order methods are attractive because they provide higher accuracy with fewer degrees of freedom and at the same time relieve the burden of generating very fine meshes. Discontinuous Galerkin (DG) methods[1] have received particular attention for aerodynamic problems; these methods combine the ideas of finite-element and finite-volume methods allowing for high-order approximations and geometric flexibility.

The goal of this work is to devise an accurate, efficient and robust three-dimensional high-order method based on DG discretizations[2] for simulating a wide variety of aerodynamic flows in an overset grid framework. This is achieved with a three-dimensional DG solver that incorporates many of the techniques previously demonstrated by our group and others in the two-dimensional setting.[3–5] The DG solver supports hybrid, mixed-element, unstructured meshes including arbitrary combinations of tetrahedra, prisms, pyramids, and hexahedra. The solver has been designed to incorporate both p-enrichment and h-refinement capabilities using non-conforming elements (hanging nodes).

Recently overset grids have been gaining popularity. Although the DG solver can handle unstructured meshes there are some situations where even an unstructured solver could benefit from an overset grid framework. Bodies in relative motion such as helicopters or wind turbines are difficult to simulate with a single grid. Typically these simulations require mesh movement or re-meshing every time step. Overset grids solve this issue by allowing multiple grids to move relative to each other. Another advantage to overset grids is the ability to combine a near-body and an off-body solver. For example using the DG solver as a near-body solver combined with an efficient cartesian mesh off-body solver would greatly increase the overall efficiency and capabilities of the solver. This has been demonstrated with success in the HELIOS[6] framework which combines a near body solver NSU3D[7] with an off-body solver SAMARC.[8, 9]

Previous work by our group combined a hexahedral DG solver[10] with NSU3D. This work has been extended to the new DG solver using a newly developed high-order overset mesh framework called TIOGA (Topology Independent Overset Grid Assembler). The novelty in this approach is that the 3D hybrid, mixed element, curved cell, *hp*-adaption, and high order capabilities in the DG solver[2] can be used in an overset framework. This gives the ability to solve complicated relative motion problems at high order and to be

[*] Post Doctoral Research Associate, Dept of Mechanical Engineering, University of Wyoming

[†] Associate Professor, Dept of Mechanical Engineering, University of Wyoming

[‡] Professor, Dept of Mechanical Engineering, University of Wyoming

combined with an efficient off-body solver. This greatly increases the flexibility of the flow solver and achieves highly accurate solutions efficiently.

In the following sections, the governing equations are described, followed by the DG discretization and its implementation for three-dimensional problems. The solution methodology is described next and is followed by a set of illustrative results showing the performance of the DG code in an overset grid framework.

II. Governing Equations

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

$$\frac{\partial U_k}{\partial t} + \frac{\partial F_{ki}}{\partial x_i} = 0 \tag{1}$$

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

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

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:

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

where $\gamma = 1.4$ is the ratio of specific heats. In all of the following, Einstein notation is used where the subscripts of i and j represent spatial dimensions and have a range of 1 to 3 and the index k varies over the number of variables.

III. DG Formulation

In this section the DG finite-element formulation used to solve the Navier-Stokes equations is described. A crucial part to any finite element method is the choice of basis. Two sets of basis functions have been implemented for all types of elements (tetrahedra, pyramid, prism, and hexahedra). The first is a C^0 hierarchal modal basis and the second is a orthonormal hierarchal modal basis. The basis for the solution and the basis for the geometrical mapping can be chosen independently. Also, each element can have a different polynomial degree for the solution and geometrical mapping. For example, the orthonormal basis can be chosen for the solution and the C^0 basis for the geometrical mapping. Also, the solution can have a polynomial degree of $p = 4$ while the mapping could have a polynomial degree of $p = 1$, or the mapping could have a polynomial degree of $p = 5$ which could be used for curved boundaries.

To derive the weak form, equation (1) is first multiplied by a test function ϕ and integrated over the domain Ω to give:

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

To obtain the weak form, integration by parts is performed and the residual R_{nm} is defined as:

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

where ϕ are the basis functions and the solution is approximated using $U_k = \phi_m a_{km}$. The index m runs over the number of basis functions. The residual now contains integrals over faces Γ and special treatment is needed for the fluxes in these terms. The advective fluxes are calculated using a Riemann solver. Implemented Riemann solvers include: Lax-Friedrichs,[11] Roe,[12] and artificially upstream flux vector splitting scheme (AUFS).[13] The diffusive fluxes are handled using a symmetric interior penalty (SIP) method.[14,15]

IV. Solution Method

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

$$\mathbf{J}^k \Delta a^k = \left[\frac{\mathbf{M}}{\Delta t} + \left(\frac{\partial R}{\partial a} \right)^k \right] \Delta a^k = -R^k \tag{3}$$

where k 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 dampen the solution of the linear update problem.[5] The mass matrix $\mathbf{M}$, due to the discontinuous basis, only appears on the block diagonals. A local time step Δt is set on every element using

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

where h is a mesh size and c is the speed of sound. The CFL number is not based on an explicit stability limit, but rather is used to control the 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_{ns} by:

$$a^{k+1} = a^k + \Delta a^k.$$

To solve the linear system in equation (3), a flexible-GMRES[16] (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:

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

Preconditioners that have been implemented include Jacobi relaxation, Gauss-Seidel relaxation, line implicit Jacobi, and ILU(0).

V. Overset Mesh Assembly

The overset grid connectivities are handled through the TIOGA interface (Topology Independent Overset Grid Assembler). The TIOGA overset grid assembler relies on an efficient parallel implementation of Alternating Digital Trees (ADT) for point-cell inclusion tests. Multiple grids are loaded in parallel and TIOGA computes the IBLANKing information required by the flow solver, as well as the cell donor-receptor information. The receptor cells receive solution data from the donor cells through interpolation as shown in Figure 1. In order to interface TIOGA with a high-order method several call back functions need to be supplied. This allows high order interpolation which is required to maintain overall high-order solution accuracy.

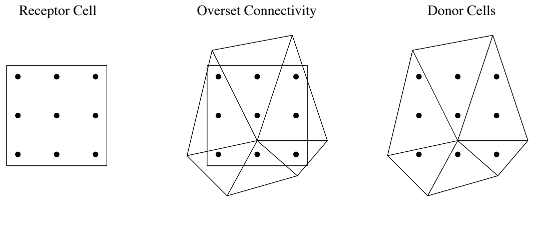

Figure 1. Receptor and donor cell overset connectivity.

There are four callback functions that are needed to interface TIOGA with a high-order method. The first callback function is a function which generates a list of receptor points. Typically these are the nodes or cell centers in a mesh, but for a high-order discretization additional points are needed within each cell. The second function is a high-order donor inclusion test which is necessary for curved elements. Once donor cells are located the third function returns the weights for the high-order interpolation on the donor cells. The last callback function converts the interpolated solution at the receptor points back into solution coefficients or modal coefficients on the receptor cell.

In this work we investigate two approaches for transferring solution 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 2 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 evaluation of the residual, while for the Vandermode matrix method the interpolation points consist of equidistant points. The interpolation points become the receptor points in the first callback function and then the receptor nodes are used by TIOGA to determine the donor cells.

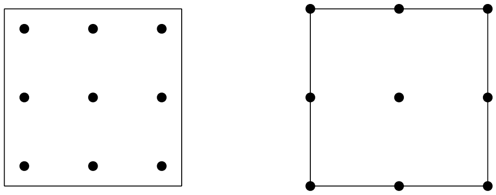

Figure 2. Locations of receptor nodes on a quadrilateral element for mass matrix method (left) and Vandermode matrix method (right).

To find the donor cells a call back function for a high-order curved cell inclusion test is used. TIOGA uses its 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 3. 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 coeffients 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:

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

Once the natural coordinates are found it is trivial to test if the point lies inside the element. For example, a tetrahedra has four faces that are each associated with a plane that can be tested for inclusion. If any of the following inequalities in Table 1 are true then the point does not lie inside the element.

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

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

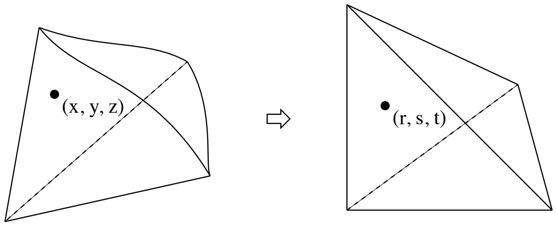

Figure 3. Curved tetrahedron in physical coordinates (left) and natural coordinates (right)

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

Table 1. Inequalities for each element type for inclusion.

where $\phi_m^d(r,s,t)$ are the donor cell basis functions (where superscript d represents donor) evaluated at the receptor node locations , n_{mode} are the number of solution basis functions, and a_m are the solution coefficients on the donor cell. TIOGA transfers, in parallel, all of the interpolated solutions from the donor cells to the receptor cells. The fourth callback function converts these solution values back into coefficients on the receptor cells. As discussed previously two approaches have been implemented; the first is the mass matrix method. This 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 basis functions (where superscript r represents receptor), n_{mode} is the number of solution receptor 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, and a_n are the receptor solution coefficients to be solved for. To solve for the coefficients both sides are multiplied by the receptor cells basis function and integrated over the cell:

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

This creates a mass matrix defined as:

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

and the right hand side becomes:

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

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:

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

The mass matrix is LU factorized ahead of time 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:

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

except now the Vandermode matrix:

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

is constructed by evaluating the basis 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:

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

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

There are two downsides to the mass matrix method. The first is that even if the mass matrix is diagonal the right hand side still requires integration with respect to the basis. The second is that, for tetrahedra, pyramids, and prisms, the number of quadrature points is greater than the number of basis modes. Also, for curved elements more quadrature points are needed making the mass matrix approach even more costly. Due to this the mass matrix method requires more computational work than the Vandermode matrix method. However, a benefit of the mass matrix method is that it gives a more optimal approximation to the interpolated values compared to the Vandermode approach at equidistant points. The Vandermode matrix 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.

VI. Fringe Removal

A high-order, finite-difference method uses a large stencil which requires many fringe points. Sometimes it may be difficult to obtain enough fringe points to complete the stencil and maintain accuracy. A benefit to using a DG discretization is that only a nearest neighbor stencil is required. TIOGA may return too many fringe cells and to minimize communication between grids these extra fringe cells can be removed. An example of an overset grid with too many fringe cells is shown in Figure 4.

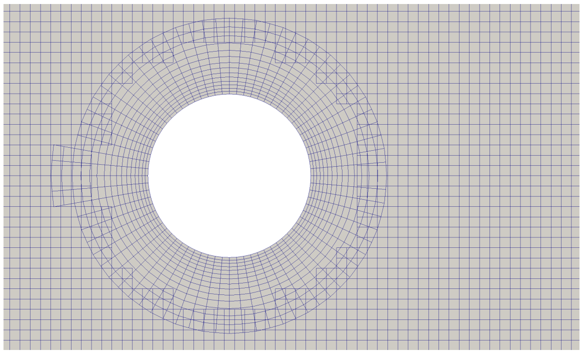

Figure 4. Overset grid for cylinder and background.

To remove 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 required by the DG solver. Figure 5 shows the background mesh IBLANK values from TIOGA and the IBLANK values after the fringe cells are removed. An IBLANK value of 1 is a field cell, 0 is a hole cell, and -1 is a fringe cell. 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 though there is not a hole in the domain this will force TIOGA to ignore these cells.

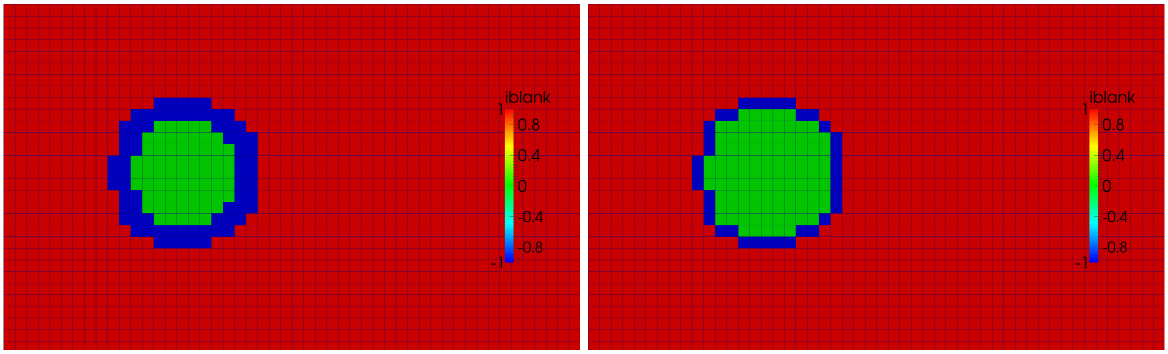

Figure 5. Initial cell IBLANK (left) and cell IBLANK after fringe removal (right) for background mesh.

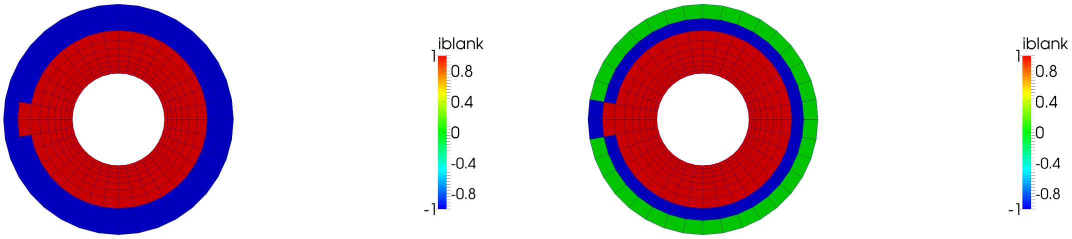

Figure 6. Initial cell IBLANK (left) and cell IBLANK after fringe removal (right) for cylinder mesh.

VII. Results

In this section four simulations are discussed. The first is a mesh resolution study to ensure 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 last two simulations are large scale simulations using the implicit solver to solve an unsteady viscous flow and a steady-state RANS simulation.

A. Ringleb Validation

Ringleb is a two dimensional exact solution to the Euler equations. This exact solution can be used to test the order of accuracy of the overset implementation. Figure 7 shows contours of pressure on an overset grid. The overset grid in this case uses hexahedra for the background grid and prisms for the interior grid. This figure shows that smooth contours of pressure are retained through the overlap regions.

To test the order of accuracy of the overset implementation the L_2 error norm is used:

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

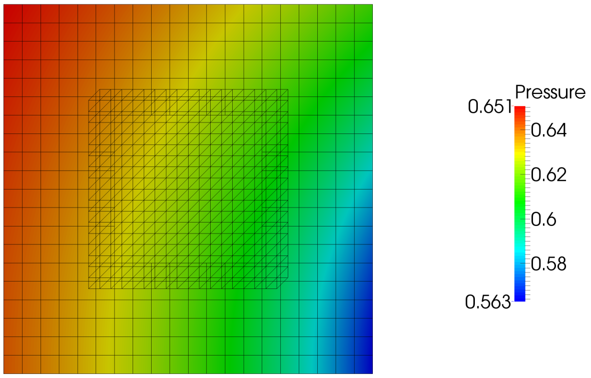

Figure 7. Pressure contours for Ringleb flow on an overset grid

This measures the error between the finite element solution u and the exact solution u_e, where V is the volume of each cell. Figure 8 shows the L_2 error for four sets of grids and polynomial degrees ranging from $p = 0$ to $p = 4$. It also contains single grid simulations for polynomial degrees ranging from $p = 0$ to $p = 2$. The overset grid simulations match closely in value and slope with the single grid simulations. For a finite element method error should decrease as Δx^{p+1}. The slopes are reported in Figure 8 and are close to the $p+1$ prediction demonstrating that the overset grid implementation maintains the correct order of accuracy.

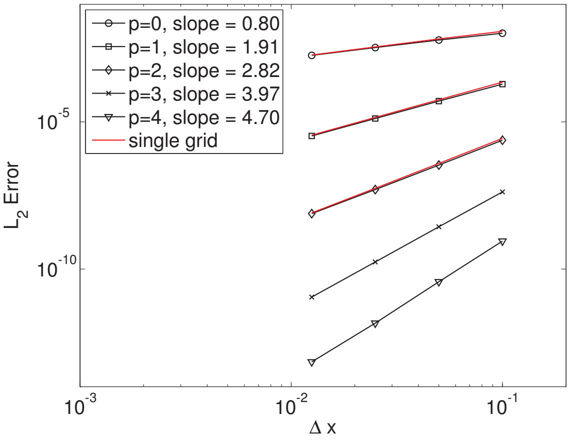

Figure 8. L_2 error for Ringleb flow on an overset grid for various grid sizes and polynomial degrees

The order of accuracy is maintained because the interpolation accuracy matches the solution accuracy of the finite element method. Figure 9 shows what happens if the interpolation accuracy is reduced from $p = 2$ to $p = 1$. The mixed approach has a slope that has deteriorated from 2.82 to 1.75 as the error becomes dominated by the interpolation processure in this case.

B. Isentropic vortex

In this section the accuracy of the Vandermode matrix method and the mass matrix method are compared with corresponding single grid error. An explicit unsteady simulation is carried out using a fourth-order Runge Kutta time stepping scheme (RK4) and an isentropic vortex is advected through the domain. The vortex was chosen because the exact solution is known. The overset grid consists of two hexahedral grids

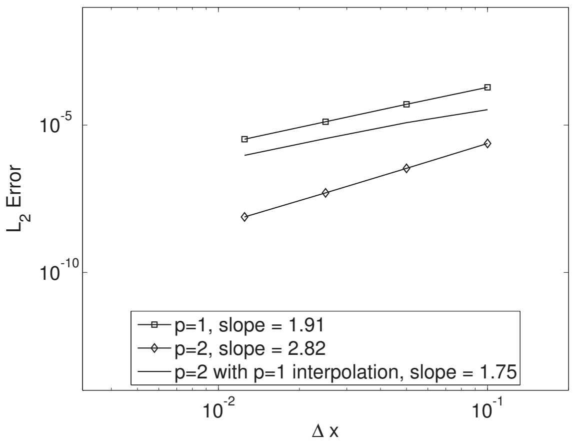

Figure 9. L_2 error for Ringleb flow on an overset grid with and without matching interpolation accuracy with solution accuracy

where the finer inner grid is rotated. The grids are shown in Figure 10 along with contours of pressure after the vortex exits the overset region.

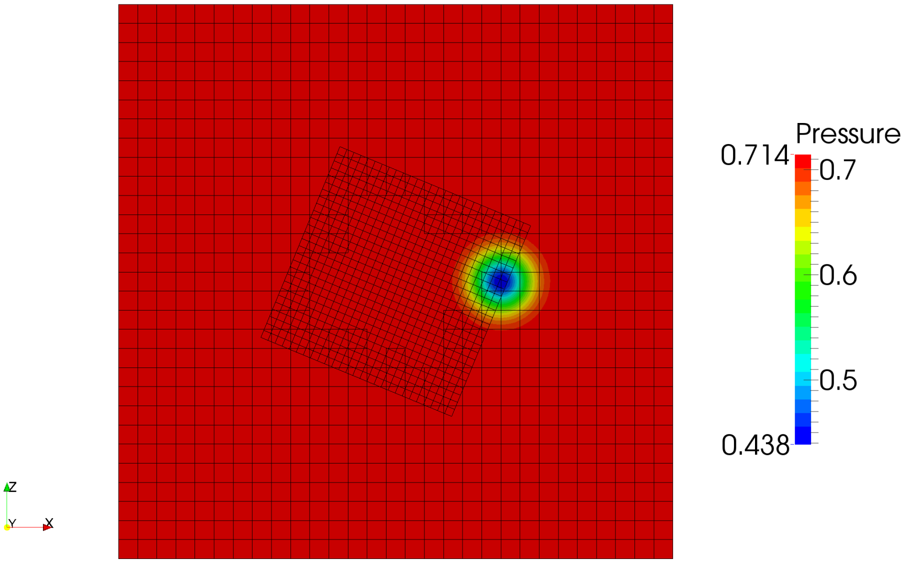

Figure 10. Contours of pressure for the Isentropic vortex on an overset mesh.

Figure 11 shows the L_2 error for a $p = 0$ simulation. For this case the Vandermode and mass matrix strategies give identical results. 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 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

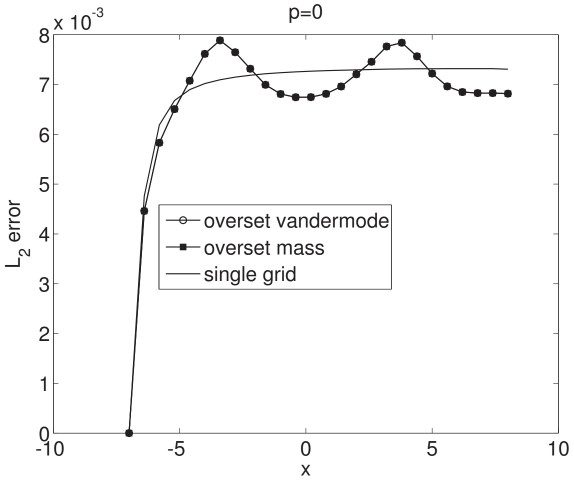

Figure 11. **L_2 Error for the Isentropic vortex, using $p = 0$ basis, with Vandermode overset approach, mass matrix overset approach, and single grid.**

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 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 around an order of magnitude compared to the $p = 2$ simulation.

C. 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[17] for the sphere, a structured hexahedral wake mesh surrounding the sphere, and an unstructured tetrahedral outer domain mesh as shown in Figure 15. At a polynomial degree of $p = 2$ which gives third order accuracy this simulation contains 6.8×10^6 degrees of freedom per solution variable. This simulation is solved on 208 processors. The time derivative is discretized using a second order backwards difference and the flow conditions are $Re = 1000$, and $Mach = 0.3$. Only qualitative results are shown in the form of contours of temperature in Figure 16. This simulation is meant to demonstrate the large scale capability and mixed element implementation of the overset framework.

D. RANS simulation of the DPW3 wing

Reynolds Averaged Navier-Stokes (RANS) simulations are performed on the third drag prediction workshop wing. The turbulence model used in this work is the negative variant of the Spalart-Allmaras (SA) model.[18] The overset grid consists of an unstructured hexahedral wing mesh and a refined structured hexahedral wake mesh as shown in Figure 17. The flow conditions are $Re = 5 \times 10^6$, $Mach = 0.76$, and $\alpha = 0.5$. The polynomial degree is $p = 2$ giving a total of 2.7×10^6 degrees of freedom. This simulation demonstrates that the wake mesh can resolve the tip-vortex generated on the wing as shown in Figure 18. This figure shows contours of eddy viscosity and it is important to note that the rotation curvature correction to the SA model is not used in this simulation. The coefficient of 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.[19]

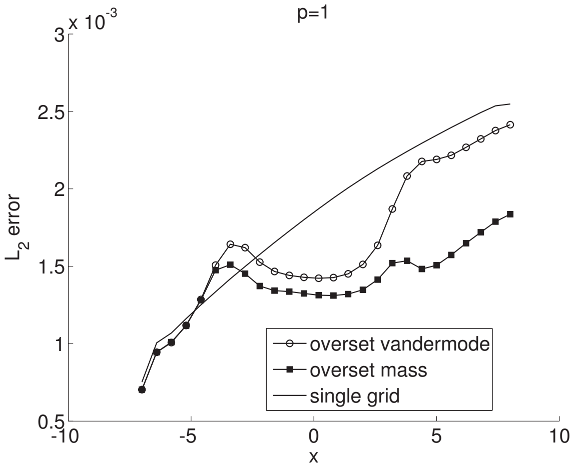

Figure 12. L_2 **Error for the Isentropic vortex, using** $p = 1$ **basis, with Vandermode overset approach, mass matrix overset approach, and single grid.**

VIII. Conclusions

In this work a high-order discontinuous Galerkin method for three dimensional viscous flows using hybrid unstructured meshes has been combined with an overset grid framework using 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. Using this solver complicated, large scale, simulations have been carried out. This is the first step in an effort to create a DG solver that can be used to solve problems with bodies in relative motion and be combined with an efficient off-body solver.

IX. Acknowledgments

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

References

[1] Reed, W. H. and Hill, T. R., "Triangular Mesh Methods for the Neutron Transport Equation," Tech. Rep. LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.

[2] Brazell, M. J. and Mavriplis, D. J., *3D Mixed Element Discontinuous Galerkin with Shock Capturing*, American Institute of Aeronautics and Astronautics, 2014/12/03 2013.

[3] Persson, P.-O. and Peraire, J., "Sub-cell shock capturing for discontinuous Galerkin methods," *Collection of Technical Papers - 44th AIAA Aerospace Sciences Meeting*, Vol. 2, 2006, pp. 1408 – 1420.

[4] Barter, G. and Darmofal, D., "Shock capturing with PDE-based artificial viscosity for DGFEM: Part I. Formulation," *J. Comput. Phys. (USA)*, Vol. 229, No. 5, 2010/03/01, pp. 1810 – 27.

[5] Burgess, N. K. and Mavriplis, D. J., "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*, 2011.

[6] Sankaran, V., Wissink, A., Datta, A., Sitaraman, J., Potsdam, M., Jayaraman, B., Katz, A., Kamkar, S., Roget, B., Mavriplis, D., Saberi, H., Chen, W.-B., Johnson, W., and Strawn, R., *Overview of the Helios Version 2.0 Computational Platform for Rotorcraft Simulations*, American Institute of Aeronautics and Astronautics, 2011.

[7] Mavriplis, D. J., "Three-dimensional multigrid Reynolds-averaged Navier-Stokes solver for unstructured meshes," *AIAA Journal*, Vol. 33, No. 3, 1995, pp. 445–453.

[8] Hornung, R., Wissink, A., and Kohn, S., "Managing complex data and geometry in parallel structured AMR applications," Vol. 22, No. 3-4, 2006, pp. 181–195.

[9] Pulliam, T. H., "Euler and Thin-Layer Navier-Stokes Codes: ARC2D, and ARC3D," *Notes for Computational Fluid Dynamics Users Workshop, The University of Tennesse Space Institute, Tullahoma, Tennessee*, 1984.

[10] Nastase, C., Mavriplis, D., and Sitaraman, J., *An Overset Unstructured Mesh Discontinuous Galerkin Approach for Aerodynamic Problems*, American Institute of Aeronautics and Astronautics, 2011.

[11] Lax, P. D., "Weak solutions of nonlinear hyperbolic equations and their numerical computation," *Communications on Pure and Applied Mathematics*, Vol. 7, No. 1, 1954, pp. 159–193.

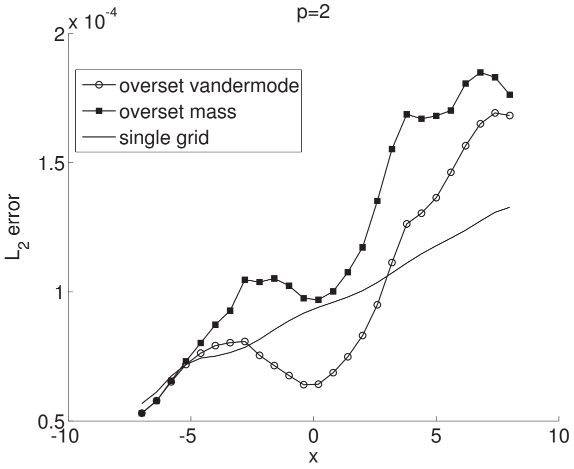

Figure 13. L_2 **Error for the Isentropic vortex, using** $p = 2$ **basis, with Vandermode overset approach, mass matrix overset approach, and single grid.**

[12] Roe, P., "Approximate Riemann solvers, parameter vectors, and difference schemes," *J. Comput. Phys. (USA)*, Vol. 43, No. 2, 1981/10/, pp. 357 – 72.

[13] Sun, M. and Takayama, K., "An artificially upstream flux vector splitting scheme for the Euler equations," *J. Comput. Phys. (USA)*, Vol. 189, No. 1, 2003/07/20, pp. 305 – 29.

[14] Hartmann, R. and Houston, P., "An optimal order interior penalty discontinuous Galerkin discretization of the compressible Navier-Stokes equations," *J. Comput. Phys. (USA)*, Vol. 227, No. 22, 2008/11/20, pp. 9670 – 85.

[15] Shahbazi, K., Mavriplis, D., and Burgess, N., "Multigrid algorithms for high-order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations," *J. Comput. Phys. (USA)*, Vol. 228, No. 21, 2009/11/20, pp. 7917 – 40.

[16] Saad, Y., "A flexible inner-outer preconditioned GMRES algorithm," *SIAM J. Sci. Comput.*, Vol. 14, No. 2, March 1993, pp. 461–469.

[17] Meakin, R., Wissink, A., Chan, W., and Pandya, S., *On Strand Grids for Complex Flows*, American Institute of Aeronautics and Astronautics, 2007.

[18] Allmaras, S., Johnson, F., and Spalart, P., "Modifications and Clarifications for the Implementation of the Spalart-Allmaras Turbulence Model," *7th International Conference on Computational Fluid Dynamics*, 2012.

[19] Vassberg, J. C., Tinoco, E. N., Mani, M., Brodersen, O. P., Eisfeld, B., Wahls, R. A., Morrison, J. H., Zickuhr, T., Laflin, K. R., and Mavriplis, D. J., "Abridged Summary of the Third AIAA Computational Fluid Dynamics Drag Prediction Workshop," *Journal of Aircraft*, Vol. 45, No. 3, 2008, pp. 781–798.

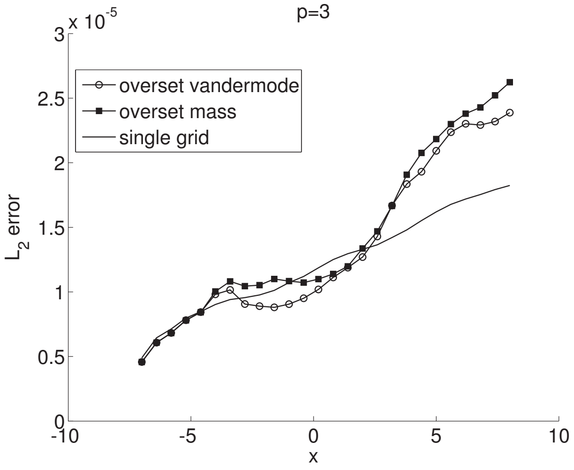

Figure 14. L_2 **Error for the Isentropic vortex, using** $p = 3$ **basis, with Vandermode overset approach, mass matrix overset approach, and single grid.**

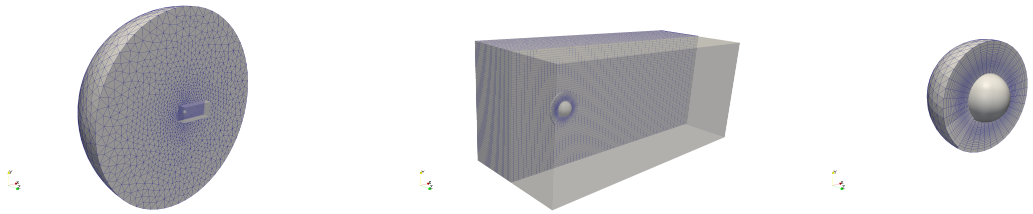

Figure 15. Overset mesh for sphere (left), wake and sphere strand mesh (center), and sphere strand mesh (right)

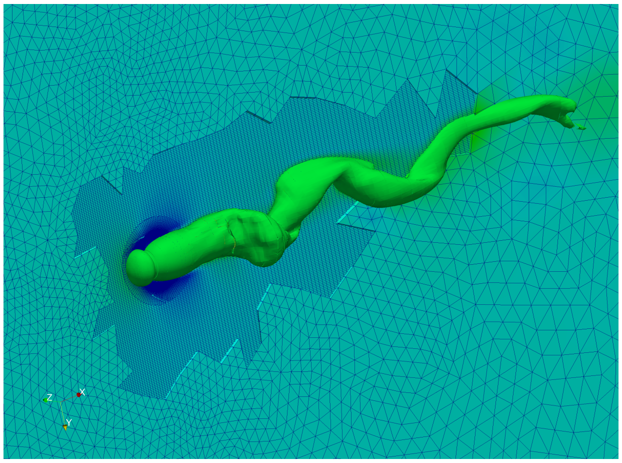

Figure 16. Contours of temperature for flow over sphere, $p = 2$, $Re = 1000$, $Mach = 0.3$

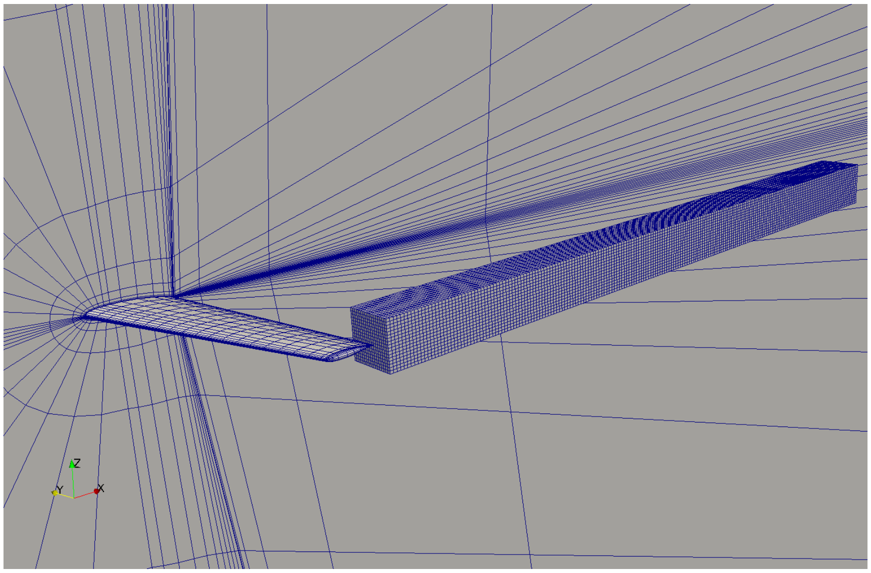

Figure 17. DPW3 wing and wake mesh

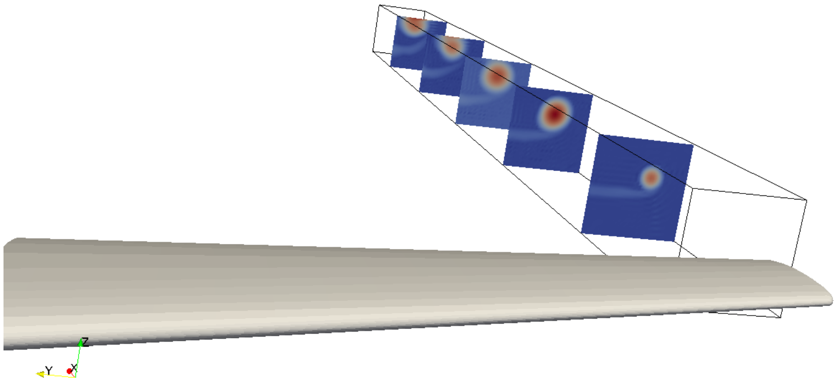

Figure 18. Contours of eddy viscosity over DPW3 wing, $p = 2$, $Re = 5 \times 10^6$, $Mach = 0.76$, $\alpha = 0.5$