3D Mixed Element Discontinuous Galerkin with Shock Capturing

Michael J. Brazell * and Dimitri J. Mavriplis †

Department of Mechanical Engineering, University of Wyoming, Laramie WY 82071, USA

A parallel high-order Discontinuous Galerkin method is developed for mixed elements to solve the Navier-Stokes equations. A PDE-based artificial viscosity is implemented to smooth and stabilize shocks. To solve this system of non-linear equations a Newton solver is implemented and preconditioned flexible-GMRES is used to solve the linear system arising from the Jacobian matrix. The preconditioners that are implemented include Jacobi relaxation, Gauss-Seidel relaxation, and a line solver. A wide variety of simulations are performed to demonstrate the capabilities of the DG solver. The inviscid simulations include a p-adapted subsonic flow over a cylinder, a $p = 0$ h-adapted hypersonic flow over a sphere, and a large scale $p = 2$ simulation of an aircraft with artificial viscosity to stabilize the shock formed on the wing. Two hypersonic viscous flows of a cylinder and sphere are simulated and compared to the NASA code LAURA. The solution matches closely to LAURA and the shock becomes more resolved as the polynomial degree is increased. The heating rate on the surface matches closely to LAURA at $p = 3$. Finally, the parallel scalability is tested and good speed up is obtained using up to 2048 processor cores. As the polynomial degree increases the scalability improves. Although, an ideal speedup was not shown this was contributed to load balancing. These simulations demonstrate the capability of the DG solver to handle strong shocks, complex geometry, hp-adaption, and parallel scalability.

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. However, similar to all high-order methods, DG discretizations are most effective for smooth solutions. For many aerodynamic problems, strong shock waves may be present and these have been found to pose significant challenges for high-order methods.

There are a number of methods developed to increase the robustness of DG to handle shocks. One is to reduce the polynomial degree to $p = 0$ in the vicinity of shocks, this essentially reduces DG to a cell centered finite volume method.[2,3] However this reduces the accuracy of the solution. Another method is to add artificial viscosity to the original set of equations first employed by Von Neumann and Richtmyer[4] and later extended to DG by Persson and Peraire.[5] This method uses a resolution sensor to detect shocks in a cell and then adds artificial viscosity to that cell. It is capable of smoothing out a shock and resolving it in high-order within a single cell. The downside to this method is that the artificial viscosity is piecewise constant between cells which can cause oscillations in the solution, especially for high Mach flows. A more robust extension of this method is a PDE-based artificial viscosity developed by Barter and Darmofal.[6] This method introduces a modified Helmholtz equation to resolve the artificial viscosity using a smooth high-order solution. Also, the shock detection is improved by detecting jumps in pressure between cells. Burgess and Mavriplis[7] implemented this method and were able to simulate strong shocks on a $M = 17.6$ flow over a cylinder.

*Post Doctoral Research Associate, AIAA Member

†Professor, AIAA Associate Fellow

The goal of this work is to devise an accurate, efficient and robust three-dimensional high-order method based on DG discretizations for simulating a wide variety of aerodynamic flows including flows with strong shock waves. This is achieved through the development of a new flexible three-dimensional DG solver that incorporates many of the techniques previously demonstrated by our group and others in the two-dimensional setting.[5–7] The DG solver supports hybrid, mixed-element, unstructured meshes including arbitrary combinations of tetrahedra, prisms, pyramids, and hexahedra. The implementation can handle a polynomial degree up to $p = 9$ (10th order). The solver has been designed to incorporate both p-enrichment and h-refinement capabilities using non-conforming elements (hanging nodes). For shock capturing, the PDE-based artificial viscosity developed by Barter and Darmofal[6] is implemented and combined with the cell pressure jump detection of Burgess and Mavriplis.[7] Steady-state solutions are solved using an effective preconditioned GMRES approach.

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 and its ability to capture strong shocks.

II. Governing Equations

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

$$\frac{\partial U_m}{\partial t} + \frac{\partial F_{mi}}{\partial x_i} = 0 \tag{1}$$

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

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

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 indices of m and n vary over the number of variables.

A. PDE-based artificial viscosity

To handle discontinuities formed by shocks, artificial viscosity can be added to the Navier-Stokes equations. In the absence of limiting this is necessary for high-order methods in order to smooth the solution and stabilize the method. To implement this method a modified Helmholtz equation is added to the governing equations to smooth the artificial viscosity which has the form:

$$\frac{\partial \epsilon}{\partial t} = \nabla \cdot \left(\left[\frac{\eta}{\tau} \nabla \epsilon \right] \right) + \frac{1}{\tau} \left(\frac{\bar{h}}{p} \lambda_{max} \tilde{s}_k - \epsilon \right). \tag{3}$$

The coefficients are defined as:

$$\tau = \frac{h_{min}}{c_1 p \lambda_{max}}, \quad \left[\frac{\eta}{\tau} \right] = \frac{c_1 c_2 p \lambda_{max}}{h_{min}} \mathrm{diag}([h_x^2, h_y^2, h_z^2]), \quad \lambda_{max} = |u| + |v| + |w| + c$$

$$\bar{h} = \frac{1}{3}(h_x + h_y + h_z), \quad h_{min} = \min(h_x, h_y, h_z), \quad c_1 = 3, \quad c_2 = 5$$

where p is the polynomial degree of the element, c is the speed of sound, and the mesh metrics h_x, h_y, h_z are the lengths of the sides of the smallest cartesian hexahedron enclosing each element. The source term $\tilde{s}_k$ is given by

$$\tilde{s}_k = \begin{cases} 0 & s_k < s_0 - \Delta\kappa \\ \frac{1}{2}\epsilon_0 \left(1 + \sin\left(\frac{1}{2}\frac{\pi(s_k - s_0)}{2\Delta\kappa}\right)\right) & s_0 - \Delta\kappa \leqslant s_k \leqslant s_o + \Delta\kappa \\ \epsilon_0 & s_k > s_0 + \Delta\kappa \end{cases}$$

where

$$s_0 = -(\kappa + c_{s0}\log_{10}(p)), \quad \Delta\kappa = \frac{1}{2}$$

and the user defined coefficients are set to $\epsilon_0 = 1, c_{s0} = 1, \kappa = 3$. The coefficient ϵ_0 controls the magnitude of the source term and the coefficients c_{s0} and κ control the minimum value of the shock detector that will trigger artificial viscosity.[7] The jump indicator s_k is calculated on each cell and is defined as:

$$s_k = \log_{10}\left(\frac{1}{\Gamma}\int_\Gamma \left|\frac{P_l - P_r}{(P_l + P_r)/2}\right| \mathrm{d}\Gamma\right)$$

where the integral is performed on each face of the cell using the pressure from the left (P_l) and right (P_r) cells.

The solution to equation (3) is then limited to give $\hat{\epsilon}$ and used as the artificial viscosity coefficient in the diffusion terms added to the continuity, momentum, and energy equations. $\hat{\epsilon}$ is limited by:

$$\hat{\epsilon} = \begin{cases} 0 & \epsilon < \hat{\epsilon}_{low} \\ \frac{1}{2}\left(1 + \sin\left(\pi\left[\frac{\epsilon - \hat{\epsilon}_{low}}{\hat{\epsilon}_{hi} - \hat{\epsilon}_{low}} - \frac{1}{2}\right]\right)\right) & \hat{\epsilon}_{low} \leqslant \epsilon \leqslant \hat{\epsilon}_{hi} \\ \hat{\epsilon}_{hi} & \epsilon > \hat{\epsilon}_{hi} \end{cases}$$

where $\hat{\epsilon}_{low} = \frac{\lambda_{max}}{100}\frac{\bar{h}}{p}$ and $\hat{\epsilon}_{hi} = \lambda_{max}\frac{\bar{h}}{p}$. The diffusive flux terms have the form:

$$F^v = -\begin{Bmatrix} \hat{\epsilon}\frac{h_x}{\bar{h}}\frac{\partial\rho}{\partial x} & \hat{\epsilon}\frac{h_y}{\bar{h}}\frac{\partial\rho}{\partial y} & \hat{\epsilon}\frac{h_z}{\bar{h}}\frac{\partial\rho}{\partial z} \\ \hat{\epsilon}\frac{h_x}{\bar{h}}\frac{\partial\rho u}{\partial x} & \hat{\epsilon}\frac{h_y}{\bar{h}}\frac{\partial\rho u}{\partial y} & \hat{\epsilon}\frac{h_z}{\bar{h}}\frac{\partial\rho u}{\partial z} \\ \hat{\epsilon}\frac{h_x}{\bar{h}}\frac{\partial\rho v}{\partial x} & \hat{\epsilon}\frac{h_y}{\bar{h}}\frac{\partial\rho v}{\partial y} & \hat{\epsilon}\frac{h_z}{\bar{h}}\frac{\partial\rho v}{\partial z} \\ \hat{\epsilon}\frac{h_x}{\bar{h}}\frac{\partial\rho w}{\partial x} & \hat{\epsilon}\frac{h_y}{\bar{h}}\frac{\partial\rho w}{\partial y} & \hat{\epsilon}\frac{h_z}{\bar{h}}\frac{\partial\rho w}{\partial z} \\ \hat{\epsilon}\frac{h_x}{\bar{h}}\frac{\partial\rho H}{\partial x} & \hat{\epsilon}\frac{h_y}{\bar{h}}\frac{\partial\rho H}{\partial y} & \hat{\epsilon}\frac{h_z}{\bar{h}}\frac{\partial\rho H}{\partial z} \end{Bmatrix}$$

and are added to the advective flux shown in equation (2). The diffusion term added to the energy equation is based on ρH instead of ρE to preserve total enthalpy.

III. DG Formulation

In this section the DG finite element formulation used to solve the Navier-Stokes equations along with the artificial viscosity equation 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 independently chosen. 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) augmented by the artificial dissipation equation (i.e. equation (3)) is first multiplied by a test function ϕ and integrated over the domain Ω to give:

$$\int_\Omega \phi_r \left(\frac{\partial U_m}{\partial t} + \frac{\partial F_{mi}}{\partial x_i}\right) \mathrm{d}\Omega = \int_\Omega \phi_r S_m \mathrm{d}\Omega$$

where S_m is a source term and is equal to zero for all equations except for the artificial viscosity equation. To obtain the weak form, integration by parts is performed and the residual R_{mr} is defined as:

$$R_{mr} = \int_\Omega \left(\phi_r \frac{\partial U_m}{\partial t} - \phi_r S_m - \frac{\partial \phi_r}{\partial x_i} F_{mi} \right) \mathrm{d}\Omega + \int_\Gamma \phi_r F_{mi} \mathrm{n}_i \mathrm{d}\Gamma = 0$$

where ϕ are the basis functions and the solution is approximated using $U_m = \phi_s a_{m,s}$. The index r and s run 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,[8] Roe,[9] and artificially upstream flux vector splitting scheme (AUFS).[10] The diffusive fluxes are handled using a symmetric interior penalty (SIP) method.[11,12]

To solve this set of equations it will be useful to have the exact Jacobian which can be used in a Newton method. The Jacobian is defined as the derivative of the residual with respect to the solution coefficients. In block matrix form the Jacobian is given by:

$$\frac{\partial R_{mr}}{\partial a_{ns}} = \int_\Omega \left(\frac{1}{\beta} \phi_r \delta_{mn} \phi_s - \phi_r \frac{\partial S_m}{\partial U_n} \phi_s - \frac{\partial \phi_r}{\partial x_i} \frac{\partial F_{mi}}{\partial U_n} \phi_s - \frac{\partial \phi_r}{\partial x_i} \frac{\partial F_{mi}}{\partial \frac{\partial U_n}{\partial x_j}} \frac{\partial \phi_s}{\partial x_j} \right) \mathrm{d}\Omega + \int_\Gamma \left(\phi_r \frac{\partial F_{mi}}{\partial U_n} \mathrm{n}_i \phi_s + \phi_r \frac{\partial F_{mi}}{\partial \frac{\partial U_n}{\partial x_j}} \mathrm{n}_i \frac{\partial \phi_s}{\partial x_j} \right) \mathrm{d}\Gamma$$

where β is a coefficient in a backwards difference formula times the inverse of the physical time step, δ_{mn} is the Kronecker delta, and n is the face normal.

IV. Solution Method

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

$$J^k_{mrns} \Delta a^k_{ns} = \left[\frac{\delta_{mn} M_{rs}}{\Delta t} + \frac{\partial R^k_{mr}}{\partial a^k_{ns}} \right] \Delta a^k_{ns} = -R^k_{mr} \tag{4}$$

where k is the non-linear iteration, $\mathbf{M}$ is a mass matrix and Δt is an element-wise time step which is used to dampen the solution.[7] The mass matrix $\mathbf{M}$ is defined as:

$$M_{rs} = \int_\Omega \phi_r \phi_s \mathrm{d}\Omega$$

which, 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 the CFL number is not based on an explicit stability limit, but rather is used to control the convergence characteristic of the implict scheme and is defined as:

$$CFL = \min \left(CFL_{min} \left(\frac{\|\mathbf{R}_0\|_2}{\|\mathbf{R}_k\|_2} \right)^\alpha, CFL_{max} \right)$$

where $\mathbf{R}_0$ is the initial residual and $\mathbf{R}_k$ is the kth non-linear residual, $CFL_{min} = 1.0$, $CFL_{max} = 1.0 \times 10^{16}$, and $\alpha = 1.25$.

Newton's method creates a linear system that must be solved to get the update to the coefficients a_{ns} by:

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

To solve the linear system in equation (4), a flexible-GMRES[13] (fGMRES) method is used. fGMRES only requires matrix vector products and for a high-order 3-D simulation the Jacobian matrix can require substantial memory storage. Therefore, two alternative methods have been implemented to reduce the memory requirement. The first is a Fréchet derivative which is defined as:

$$\mathbf{J}\Delta a \approx \frac{\mathrm{Im}\left(\mathbf{R}(a + \imath\epsilon\Delta a)\right)}{\epsilon}$$

which gives the approximation of $\mathbf{J}$ multiplied by a vector x by perturbing the complex residual. The second is the exact matrix vector product which formed by projecting the coefficients $\Delta a_{n,s}$ to give $\Delta U_n = \phi_s \Delta a_{n,s}$ and evaluating:

$$J_{mrns}\Delta a_{ns} = \int_\Omega \left(\left(\frac{1}{\beta} + \frac{1}{\Delta t} \right) \phi_r \delta_{mn} \Delta U_n - \phi_r \frac{\partial S_m}{\partial U_n} \Delta U_n - \frac{\partial \phi_r}{\partial x_i} \frac{\partial F_{mi}}{\partial U_n} \Delta U_n - \frac{\partial \phi_r}{\partial x_i} \frac{\partial F_{mi}}{\partial \frac{\partial U_n}{\partial x_j}} \frac{\partial \Delta U_n}{\partial x_j} \right) \mathrm{d}\Omega$$
$$+ \int_\Gamma \left(\phi_r \frac{\partial F_{mi}}{\partial U_n} \mathrm{n}_i \Delta U_n + \phi_r \frac{\partial F_{m,i}}{\partial \frac{\partial U_n}{\partial x_j}} \mathrm{n}_i \frac{\partial \Delta U_n}{\partial x_j} \right) \mathrm{d}\Gamma$$

which is only slightly more cost than the evaluation of the residual and the same storage as the solution vector.

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:

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

In serial, three preconditioners have been implemented. This includes Jacobi relaxation, Gauss-Seidel relaxation, and a line solver. However, the only preconditioner implemented in parallel to date is Jacobi relaxation.

V. Validation

In this section an exact solution will be used to verify the accuracy of the finite element formulation. Ringleb flow is an exact solution to the steady state Euler equations and is solved analytically using the hodograph method.[14] Although it is only a two-dimensional solution, the three-dimensional equations can be validated by setting the momentum in the third dimension to zero. Characteristic boundary conditions are used in the x and y directions and an inviscid wall is used in the z direction. The domain is a 1×1 square discretized with a hexahedral mesh. Four mesh resolutions are used and they consist of 5, 10, 20, and 40 elements in both the x and y directions and 1 element in the z direction. The flow is initialized using the exact solution and then the residual is driven to machine precision. The steady state solution of a $p = 3$ discretization with $40 \times 40 \times 1$ elements is plotted in Figure 1.

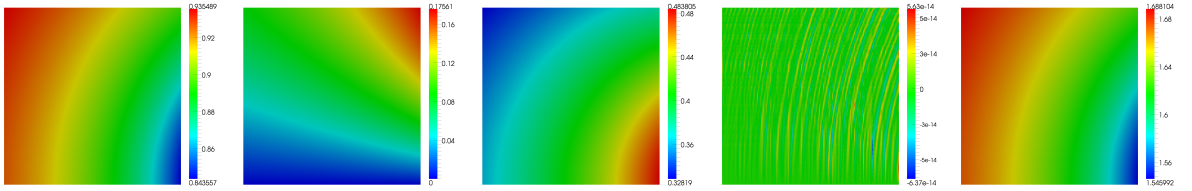

Figure 1. Ringleb flow: Contours of $\rho, \rho u, \rho v, \rho w, \rho E$

The errors in a discontinuous Galerkin discretization should decrease following the power law Ch^{p+1}, where C is a constant, h is the mesh size, and p is the polynomial degree. By comparing the analytic solution to the numerical solution the error can be measured. The error is measured using a L_2 norm for $p = 0, 1, 2, 3$ and all four mesh resolutions. The L_2 error versus mesh size h is plotted in Figure 2. As predicted the error decreases to the power of $p + 1$ or better.

VI. Results

Both subsonic and hypersonic results are shown in this section. The first three cases are all inviscid simulations. The first of these cases demonstrates a subsonic inviscid flow over a cylinder with p-adaption, the second case is a hypersonic flow over a sphere with h-adaption, and the third is an inviscid flow over an aircraft with artificial viscosity to smooth the shock formed on the wing. Then two viscous simulations of hypersonic flow over a cylinder and sphere are presented and compared to the NASA LAURA[15] code. Finally, results of the parallel scalability of the DG solver are presented.

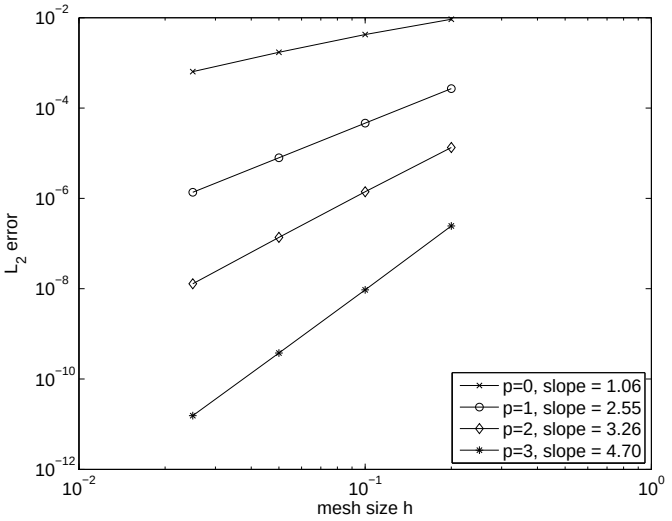

Figure 2. L_2 error vs. mesh size.

A. Inviscid Simulations

The first case is a $M = 0.5$ flow over a cylinder and was solved on a purely hexahedral mesh. The cylinder has a radius of 1 and the domain has an outer radius of 30. Characteristic boundary conditions are used on the inlet and outlet and slip walls are used on the cylinder. This simulation was solved by incrementing the polynomial degree starting at $p = 0$ and ending at $p = 2$. Then, the mesh is adapted and contains elements with polynomial degrees ranging from $p = 1$ to $p = 3$. Figure 3 shows the iterative convergence using the Newton fGMRES solver for each simulation. The Figure on the left shows the non-linear residual versus the number of non-linear Newton iterations while the Figure on the right shows the non-linear residual versus the number of linear iterations within fGMRES required for each non-linear update. The fGMRES solver exits when a relative tolerance of 1.0×10^{-3} for the linear residual or a maximum of 500 Krylov vectors and 10 restarts are reached. Starting with $p = 0$, Newton's method with damping is used and then $p = 1$ is initialized using the $p = 0$ solution. The damping term causes the initial slow transient for these two cases, but as the residual decreases the damping term is turned off and quadratic convergence is achieved. The remaining cases are simulated without damping as the solution is well developed. Again, the $p = 1$ simulation is used to initialize the $p = 2$ simulation and rapid convergence is observed. The final simulation is solved on an adapted mesh. The mesh is adapted by comparing the magnitude of the $p = 2$ modes. For regions with large high-order coefficients, the polynomial degree is increased to $p = 3$ and for small coefficients, the polynomial degree is decreased to $p = 1$. Again, rapid convergence is observed. The degrees of freedom for the $p = 0$, $p = 1$, $p = 2$, and $p = 1, 2, 3$ are 4500, 36000, 121500, and 113000 respectively. The solution to this final case is shown in Figure 4. The contour plot on the left shows the density field and the plot on the right shows the mesh and polynomial degree on each element.

The next simulation demonstrates the h-adaptation capabilities of the DG solver. All element types can be locally adapted and allow for hanging nodes between cells. Prisms and hexahedra are subdivided into eight self similar elements, tetrahedra are subdivided into eight tetrahedra using a shortest edge refinement[16] rule, and pyramids are broken into six pyramids and four tetrahedra. Mesh adaption is only implemented in serial for now which limits the size of the simulation that can be performed, therefore only a $p = 0$ simulation is shown. Figure 5 shows a $M = 17.6$ inviscid flow over a sphere. The Figure on the left shows the hexahedra mesh which has four levels of refinement. The Figure in the middle shows contours of Mach number and the Figure on the right shows contours of pressure.

The final inviscid case is a $M = .85$ flow over the NASA Common Research Model (CRM) at one degree angle of attack.[17,18] This simulation is solved in parallel on 4096 processors and the mesh contains 1.9 million tetrahedra. For a $p = 2$ simulation this gives 19 million degrees of freedom per solution variable. Figure 6 shows pressure and Mach contours on the surface of the CRM, at these conditions a shock has formed on the wing. The simulation uses a $p = 2$ basis which is stabilized using the artificial viscosity. Small fluctuations

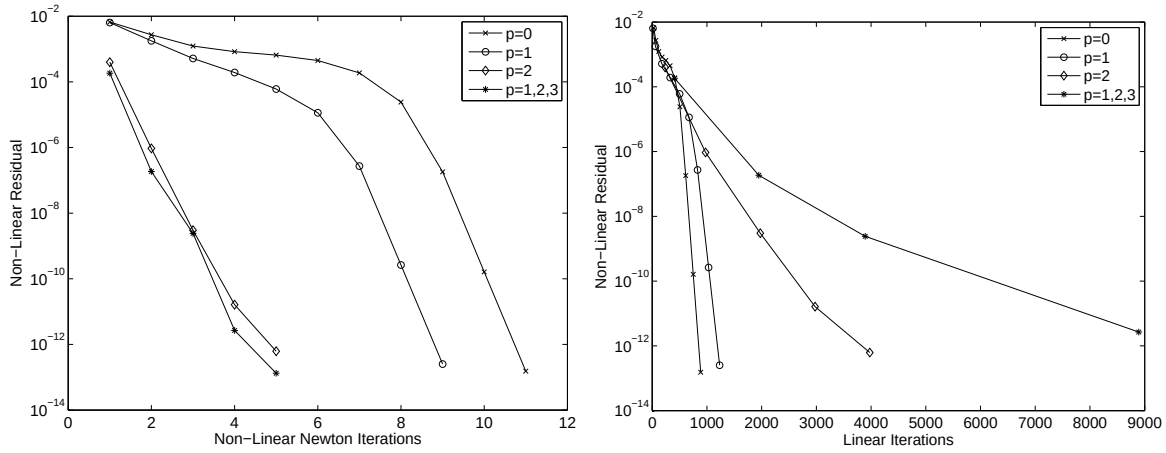

Figure 3. Non-Linear Residual vs. Non-Linear Newton Iterations (left) and Non-Linear Residual vs. Linear Iterations (right) for $M = 0.5$ flow over cylinder

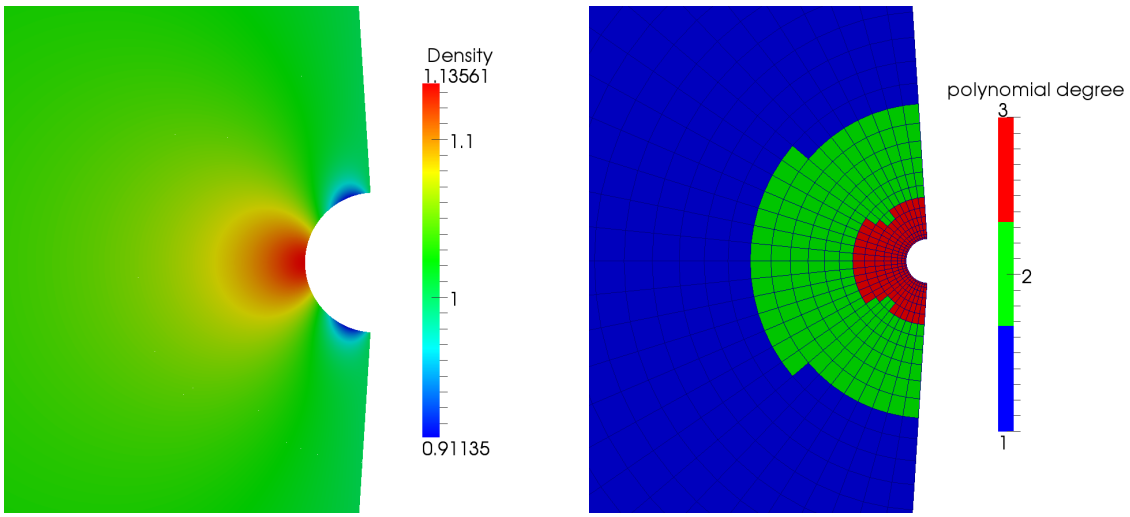

Figure 4. Density field (left) and mesh and polynomial degree (right) for $M = 0.5$ flow over a cylinder

are observed on the surface due to the use of straight sided elements. High order curved elements have been implemented for simple analytic geometries but not yet for CAD geometries. Nevertheless, this simulation shows the capability to handle shocks on complex geometries using a high-order method.

B. Viscous Hypersonic Simulations

In this section results are given for hypersonic viscous flow over a cylinder and sphere. Both of these geometries are standard test cases for hypersonic codes and have been extensively investigated using the NASA LAURA[15] code. To verify the implementation of the DG solver, both the cylinder and sphere is simulated at the conditions: $M = 17.605$, $Pr = .72$, and $Re = 376,930$. The inlet temperature is 200 K and the wall temperature is set to 500 K.

The cylinder is simulated using a shock fitted mesh generated by LAURA which contains 60×64 elements in the circumferential and radial directions and one element in the spanwise direction. Ten extra elements are added radially as the artificial viscosity spreads out the shock and this prevents the shock from touching the inlet. The Newton solver is used in each case with Jacobi preconditioning and CFL damping. The convergence of the residual for $p = 1, 2, 3$ is shown in the three plots in Figure 7. The $p = 1$ simulation

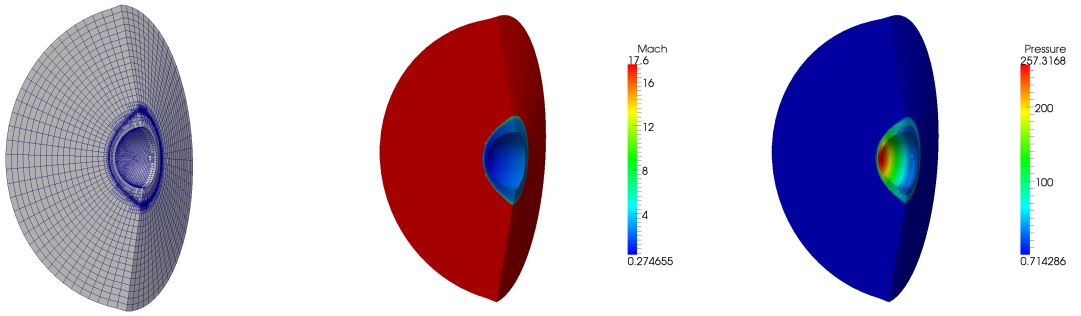

Figure 5. h-adapted $M = 17.6$ flow over sphere mesh (left), Mach contours (middle), and pressure contours (right)

has an initial transient that occurs as the shock is forming, then quick convergence is observed. The $p = 1$ simulation is then used to initialize the $p = 2$ simulation and therefore the convergence is quicker. The rate of convergence changes occasionally as the simulation is restarted with different CFL numbers. Finally, the $p = 3$ simulation is started using the $p = 2$ solution and fast convergence is observed after a small initial transient.

Figure 8 shows contours of temperature and pressure and Figure 9 shows contours of artificial viscosity and Mach number for the $p = 3$ simulation. The contour plots show that the shock is well resolved.

The non-dimensional pressure along the stagnation streamline is shown for $p = 1, 2, 3$ in Figure 10. As the polynomial degree increases the shock becomes less spread out and more resolved. The pressure for $p = 3$ DG simulation compares well to the LAURA simulation.

Similar results are shown in Figure 11 for the temperature along the stagnation streamline. The shock becomes better resolved for higher order and the $p = 3$ DG simulation gives very similar results compared to LAURA. The thermal boundary layer is well resolved for all cases as the boundary layer mesh is very fine.

The final comparison for the cylinder is the heating rate on the surface. This is difficult to obtain as it depends on temperature gradients and many simulations suffer from oscillations. Figure 12 shows the heating rate C_H versus theta. As the polynomial degree increases, the heating also increases. Comparable results with LAURA are only achieved with the $p = 3$ simulation. Also, the heating results from Burgess[19] on an h-adapted mesh using a high-order two-dimensional DG code are shown and compare well to the 3-D DG solver.

The next case is a viscous hypersonic flow over a sphere. This simulation is performed on a mesh created using Gmsh.[20] The mesh is structured and contains 242 thousand hexahedral elements with a boundary layer growth rate of 1.125. A $p = 1$ simulation is carried out but not fully converged and then used to initialize a $p = 2$ simulation. Figure 13 shows contours of non-dimensional pressure and temperature and Figure 14 shows contours of artificial viscosity and Mach number for the $p = 2$ simulation.

One-dimensional comparisons are made to LAURA along the stagnation streamline on the sphere. Figure 15 shows the density, pressure, and temperature of the $p = 2$ DG simulation compared to LAURA. The magnitudes and shock location are similar to LAURA but the shock is not as sharp as the Laura simulation. This is due to Laura using a shock fitted mesh and the DG simulation not using a shock fitted mesh. The DG simulation could be improved by using higher-order or by adapting the mesh.

In the sphere simulation, stability issues required that the inlet velocity was ramped up from $M = 3$ $M = 17.605$, also the preconditioner was not very effective and very slow convergence was observed. The simulation was terminated after the residual dropped five orders of magnitude. Further investigation is needed to fix the stability issues and also a parallel implementation of the line solver will help with the convergence. Due to the lack of convergence and only using a $p = 2$ basis, the heating rate does not compare well with LAURA and is therefore not presented.

C. Parallel Scalability

The final result is the parallel scalability of the DG solver. In this case a viscous hypersonic sphere is simulated. A coarse mesh with 44 thousand hexahedral elements is used and the timing includes twenty

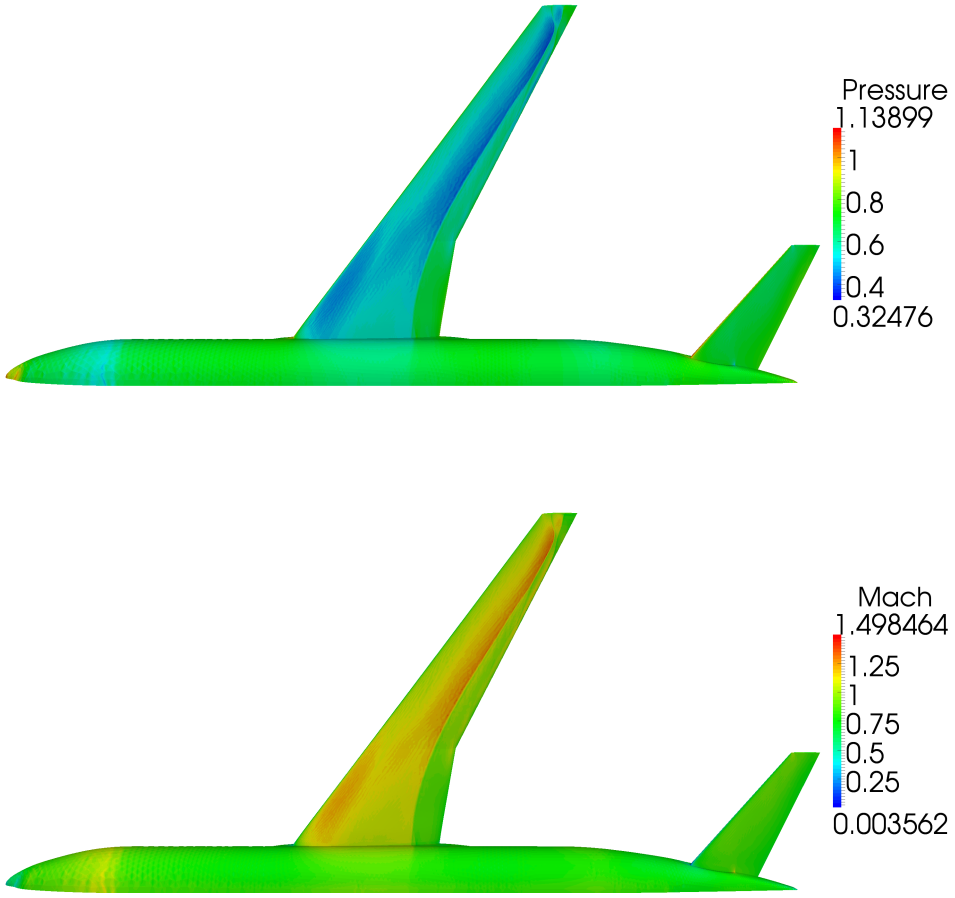

Figure 6. $M = .85$, $\alpha = 1$ **flow over NASA CRM, pressure contours (top) and Mach contours (bottom)**

Newton iterations. Each Newton iteration requires the assembly of the Jacobian, factorization of the Jacobian block diagonals, fGMRES, and Jacobi preconditioning on each Krylov vector. For the $p = 1, 2, 3$ cases the full Jacobian was stored, however for $p = 4$ only the block diagonals are stored and the complex Fréchet derivative is used to obtain Jacobian vector products that are exact to machine precision. Figure 16 shows the scaling from 32 processors to 2048 processors for $p = 1, 2, 3, 4$. The scaling improves as the polynomial degree increases. At 2048 processors the mesh contains an average of 21 cells per processor. The load balancing was not perfect and a minimum of 14 cells and a maximum of 31 cells were found within the partitioned mesh. Therefore the speed up of 1400 achieved with $p = 4$ on 2048 processors could theoretically rise to 2019 under the assumption of perfect load balancing (i.e. 21 cells per partition).

VII. Conclusions and Future Work

A high-order discontinuous Galerkin method for three dimensional viscous flows using hybrid unstructured meshes has been developed in this work. This new solver capability is based mostly on techniques previously developed in the two-dimensional setting and this work has successfully demonstrated their extension and applicability to large scale three-dimensional problems running on parallel computer architectures. In particular, the ability to capture strong hypersonic shocks at high order in three dimensions has been shown. Future work will focus on advancing the efficiency, robustness and capabilities of this solver. Firstly,

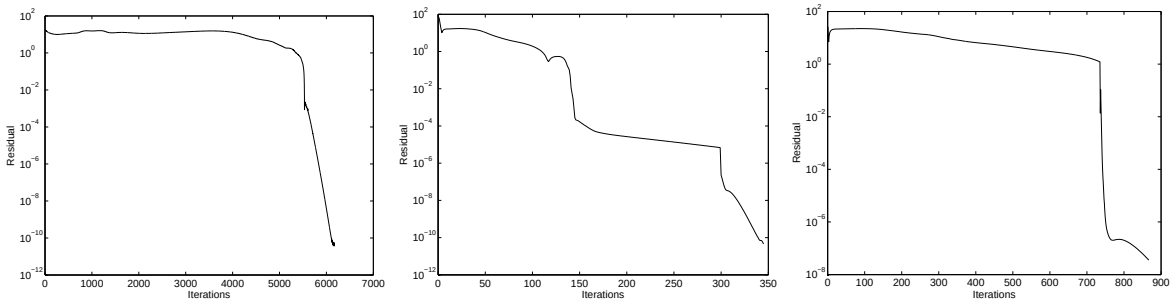

Figure 7. Residual convergence for cylinder $p = 1$ (left), $p = 2$ (middle), and $p = 3$ (right)

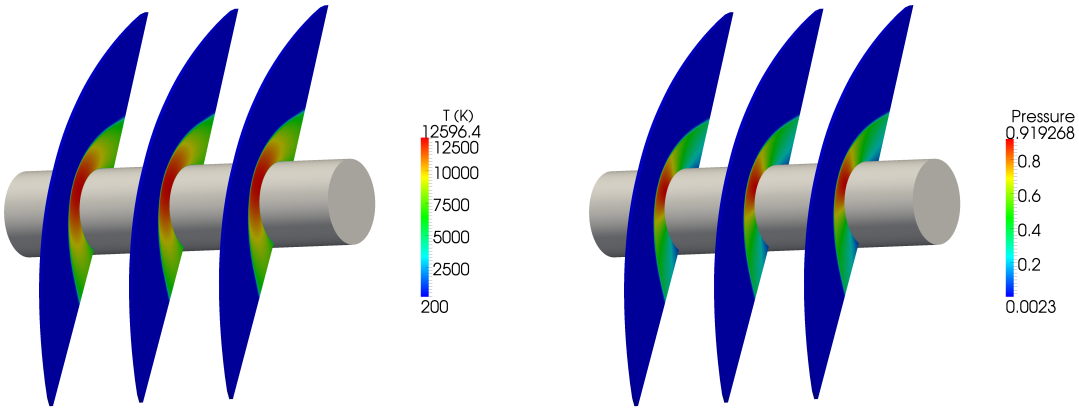

Figure 8. $M = 17.605$, $Re = 376,930$ flow over a cylinder, contours of temperature in Kelvin (left) and contours of non-dimensional pressure $P/\rho_\infty V_\infty^2$ (right)

stronger preconditioners will be investigated and implemented in parallel in order to increase solution efficiency. Secondly, although the ability to incorporate curved mesh elements has been implemented, at present only analytic geometry definitions can be handled, and thus an extension of this capability to handle CAD defined surface geometries will be pursued. Adaptive techniques have been developed but to make use of their full potential, these methods must be parallelized through the incorporation of dynamic load balancing. Target applications will include more realistic three-dimensional hypersonic re-entry geometries as well as lower speed flow transonic configurations.

VIII. Acknowledgments

This work was partially funded by AFOSR Contract FA9550-10-C-0051 and NASA Grant NNX12AF95A.

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]Burbeau, A., Sagaut, P., and Bruneau, C.-H., "A problem-independent limiter for high-order Runge-Kutta discontinuous Galerkin methods," *J. Comput. Phys. (USA)*, Vol. 169, No. 1, 2001/05/01, pp. 111 – 50.

[3]Baumann, C. and Oden, J., "A discontinuous hp finite element method for the Euler and Navier-Stokes equations," *Int. J. Numer. Methods Fluids (UK)*, Vol. 31, No. 1, 15 Sept. 1999, pp. 79 – 95.

[4]Von Neumann, J. and Richtmyer, R., "A method for the numerical calculation of hydrodynamic shocks," *Journal of Applied Physics*, Vol. 21, 1950/03/, pp. 232 – 237.

[5]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.

[6]Barter, G. and Darmofal, D., "Shock capturing with PDE-based artificial viscosity for DGFEM: Part I. Formulation," *J.*

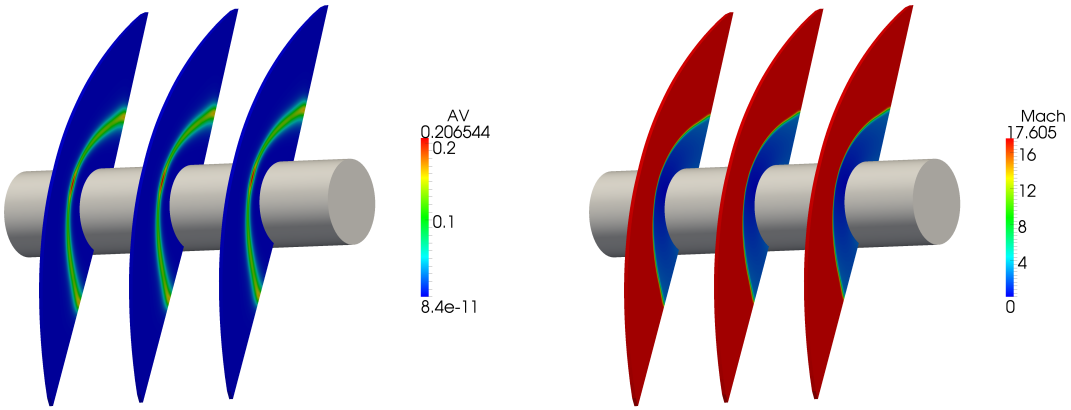

Figure 9. $M = 17.605$, $Re = 376,930$ **flow over a cylinder, contours of artificial viscosity (left) and contours of Mach number (right)**

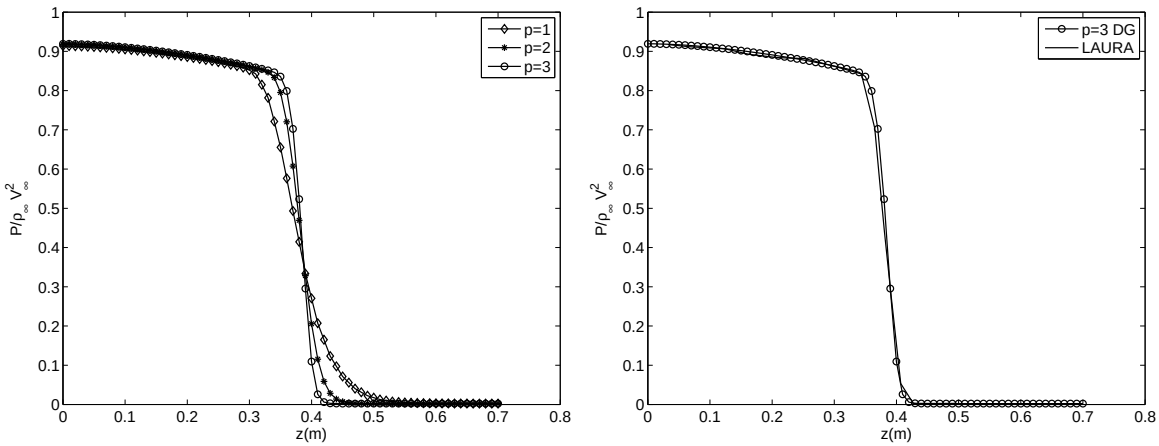

Figure 10. Stagnation streamline non-dimensional pressure for $M = 17.605$, $Re = 376,930$ **flow over a cylinder using** $p = 1, 2, 3$ **DG (left) and** $p = 3$ **DG compared to LAURA (right)**

Comput. Phys. (USA), Vol. 229, No. 5, 2010/03/01, pp. 1810 – 27.

[7]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.

[8]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.

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

[10]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.

[11]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.

[12]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.

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

[14]Ringleb, F., "Exakte Loesungen der Differentialgleichungen einer adiabatischen Gasstroemung," *A. Angew. Math. Mech.*, Vol. 20, No. 4, 1940, pp. 185–198.

[15]Mazaheri, A., Gnoffo, P. A., Johnston, C. O., and Kleb, B., *Laura Users Manual 5.2-43231. NASA TM 215944*, NASA, November 2009.

[16]Zhang, S., "Successive subdivisions of tetrahedra and multigrid methods on tetrahedral meshes," *Houston J. of Math*, Vol. 21, 1995, pp. 541–556.

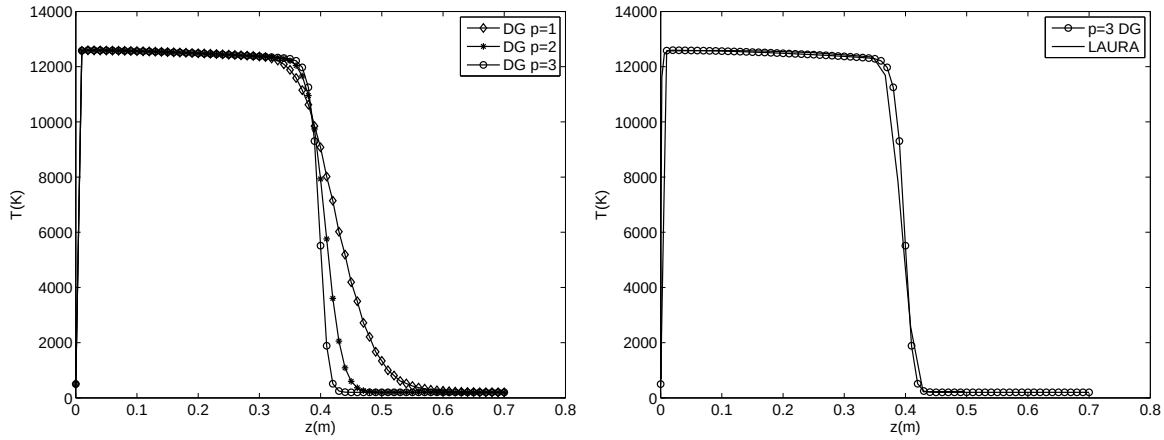

Figure 11. Stagnation streamline temperature (K) for $M = 17.605$, $Re = 376,930$ flow over a cylinder using $p = 1, 2, 3$ DG (left) and $p = 3$ DG versus LAURA (right)

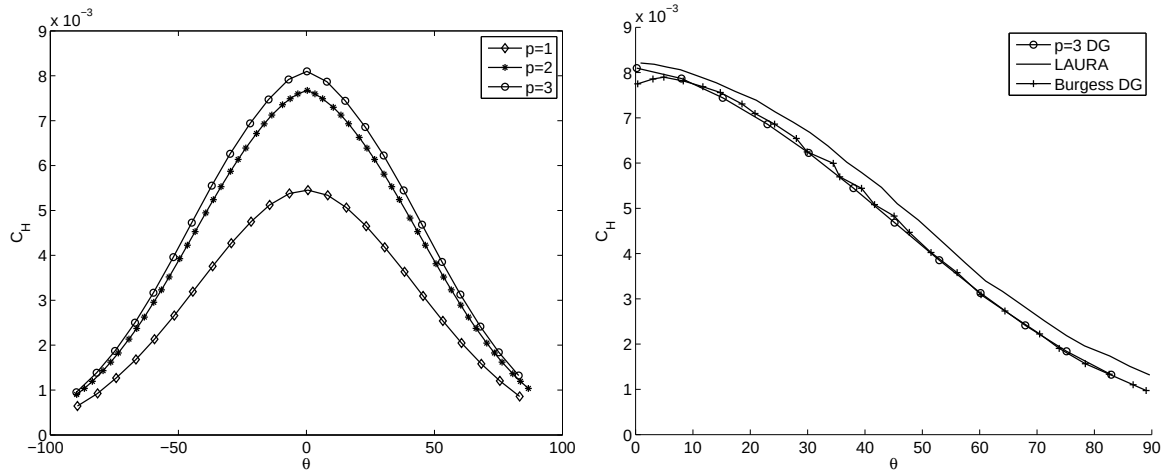

Figure 12. Heating rate (C_H) for $M = 17.605$, $Re = 376,930$ flow over a cylinder using $p = 1, 2, 3$ DG (left) and $p = 3$ DG compared to LAURA and Burgess DG (right)

[17]Vassberg, J. C., Tinoco, E. N., Mani, M., Zickuhr, T., Levy, D., Brodersen, O. P., Eisfeld, B., Wahls, R. A., Morrison, J. H., Mavriplis, D. J., and Murayama, M., "Summary of the Fourth AIAA CFD Drag Prediction Workshop," AIAA Paper 2010-4547.

[18]Vassberg, J. C., DeHaan, M. A., Rivers, S. M., and Wahls, R. A., "Development of a Common Research Model for Applied CFD Validation Studies," AIAA Paper 2008-6919.

[19]Burgess, N., *An Adaptive Discontinuous Galerkin Solver for Aerodynamic Flows*, Ph.D. thesis, University of Wyoming, 2011.

[20]Geuzaine, C. and Remacle, J.-F., "Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities," *International Journal for Numerical Methods in Engineering*, Vol. 79, No. 11, 2009, pp. 1309 – 1331.

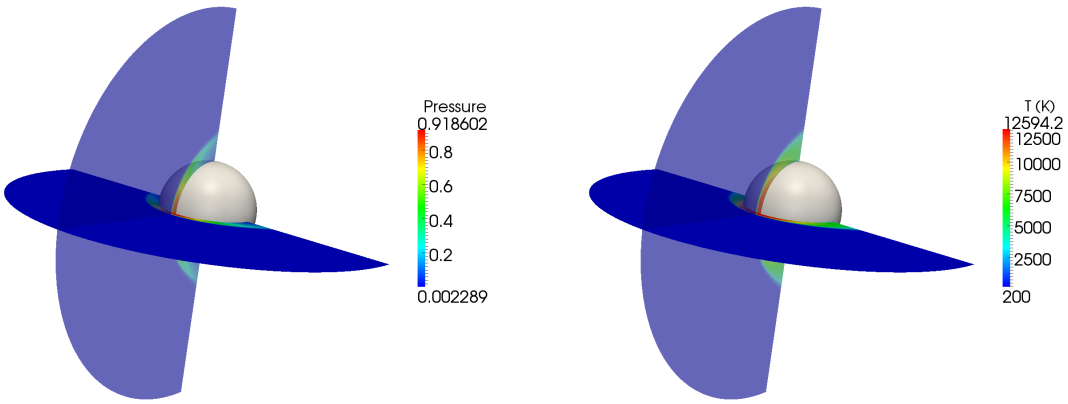

Figure 13. $M = 17.605$, $Re = 376,930$ **flow over a sphere: contours of non-dimensional pressure** $P/\rho_\infty V_\infty^2$ **(left) and contours of temperature (K) (right)**

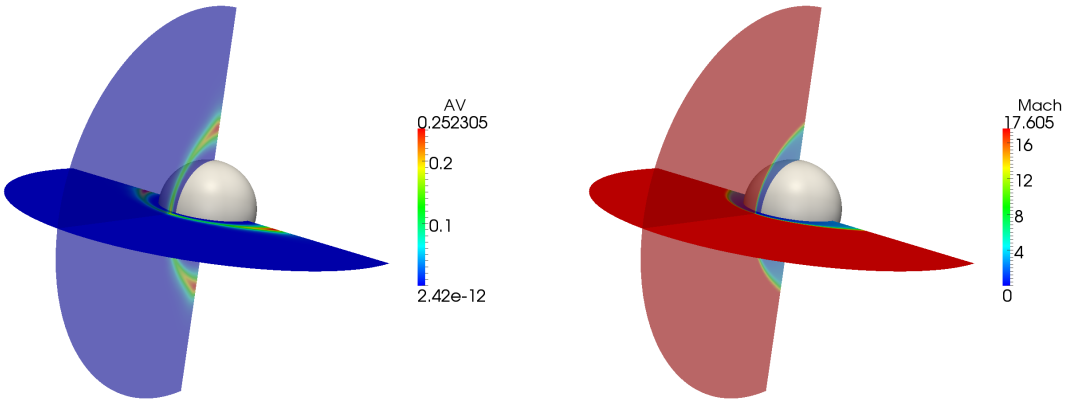

Figure 14. $M = 17.605$, $Re = 376,930$ **flow over a sphere: contours of artificial viscosity (left) and contours of Mach number (right)**

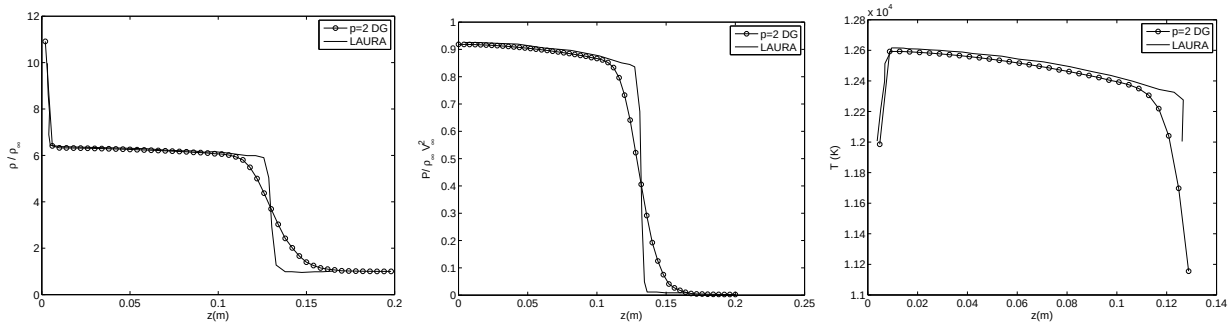

Figure 15. Stagnation streamline non-dimensional density (left), non-dimensional pressure (middle), and temperature (K) (right) for $M = 17.605$, $Re = 376,930$ **flow over a sphere comparing** $p = 2$ **DG to Laura**

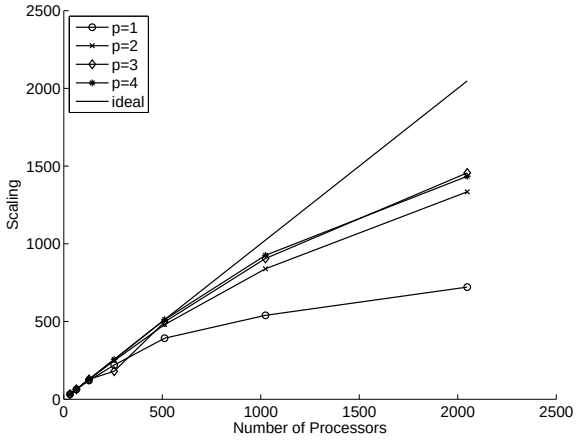

Figure 16. Parallel scaling of DG solver on a mesh using approximately 40 thousand hexahedra