Discontinuous Galerkin Turbulent Flow Simulations of NASA Turbulence Model Validation Cases and High Lift Prediction Workshop Test Case DLR-F11

Michael J. Brazell[a], Behzad R. Ahrabi[b], and Dimitri J. Mavriplis[c]

Department of Mechanical Engineering, University of Wyoming

A high-order Discontinuous Galerkin (DG) solver is used to simulate three turbulent flow simulations. Two of these simulations come from the NASA turbulence modeling resource and include turbulent flow over a hemisphere-cylinder and turbulent flow over a 3D bump in a channel. The third simulation is the DLR-F11 from the second high-lift prediction workshop. The flow is simulated by solving the Reynolds-Averaged Navier-Stokes equations closed by the negative Spalart-Allmaras turbulence model. For these simulations, lift, drag, pitching moment, pressure, and skin friction coefficients are provided for multiple grids and discretization orders and compared against other simulation results from well known finite-volume solvers. The simulations give very similar results to these benchmark solvers, pointing towards fully mesh resolved simulations and providing verification evidence of correct and consistent implementation of these discretizations. Results obtained using the high-order DG discretizations show higher accuracy using fewer degrees of freedom compared to the finite-volume discretizations. Also, it is shown for the DG simulations that p-refinement converges quicker to the mesh resolved solutions compared to h-refinement. However, there are still issues with the efficiency of DG and the time to solution is still an order of magnitude slower compared to finite-volumes methods such as NSU3D.

I. Introduction

Computational fluid dynamics (CFD) has matured to the state where many different types of discretizations can be considered for solving aerodynamic problems. Given the diversity of possible choices and the continual development of new schemes, it is of increasing interest to investigate the comparative accuracy and efficiency of the different approaches for solving standard test problems. However, in addition to different discretizations, different solvers may be used, and different implementations may also alter the results both in terms of accuracy and efficiency. Provided an implementation can be verified to demonstrate that the stated equations are discretized and implemented correctly, accuracy results should be independent of the particular solver and implementation. On the other hand, this is not the case for efficiency comparisons, which can depend on the chosen solver, the specific implementation, and the computational hardware.

In the interest of promoting proper verification and comparison of existing and new discretizations and their associated implementations, NASA has constructed and maintains the turbulence modeling resource (TMR) website [1]. This archive provides a database of Reynolds-averaged Navier-Stokes (RANS) simulations using benchmark solvers which are both highly resolved and well documented. The original purpose of the site is to provide computational fluid dynamics (CFD) developers a resource for implementing and validating RANS turbulence models, but this archive also serves as a verification database for legacy as well as new discretizations and their associated implementations.

[a]AIAA member, Post Doctoral Research Associate, mbrazell@uwyo.edu
[b]AIAA member, Post Doctoral Research Associate, brezaahr@uwyo.edu
[c]AIAA Associate Fellow, Professor, mavripl@uwyo.edu

Over the last decade, there has been significant interest in the development of higher-order discretizations for aerodynamics, due to the potential these methods offer for achieving higher accuracy at reduced cost [2–6]. In this work, we compare the accuracy of traditional second-order accurate finite-volume solvers with a high-order accurate discontinuous Galerkin (DG) discretization [7]. This work is a continuation of the verification of turbulent flows in 2D geometries that occurred in the 2015 AIAA SciTech conference [8] which included several high-order contributions [9–12]. In this paper, the previous work is extended to 3D configurations which include a 3D hemisphere-cylinder and a 3D bump. These two cases are provided on the NASA TMR website. These well documented cases include a series of highly resolved benchmark meshes with associated solutions from three well validated solvers (CFL3D [13], FUN3D [14], and USM3D [15]) including results such as force and moment coefficients, surface pressure and skin friction distributions, and off-body profiles. Also included is a realistic 3D configuration (DLR-F11) from the second AIAA CFD High Lift Prediction Workshop (HiLiftPW-2) [16].

In the following sections, the governing equations are described, followed by the DG discretization and implementations. The solution methodology is described next and is followed by a discussion of the simulation results of the hemisphere-cylinder, the 3D bump, and the DLR-F11 high-lift configuration.

II. Governing Equations

All simulations in this work are based on the solution of the steady-state Reynolds-averaged Navier-Stokes (RANS) equations with a single-equation turbulence model. In all of the following, Einstein notation is used where the subscripts of i, j, and k represent spatial dimensions and have a range of 1 to 3 and the indices of m and n vary over the number of conserved variables. The compressible Navier-Stokes equations 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 the flux vector F are defined as:

$$U = \begin{Bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho E \end{Bmatrix}, \quad F = \begin{Bmatrix} \rho u_1 & \rho u_2 & \rho u_3 \\ \rho u_1^2 + P - \tau_{11} & \rho u_1 u_2 - \tau_{12} & \rho u_1 u_3 - \tau_{13} \\ \rho u_1 u_2 - \tau_{21} & \rho u_2^2 + P - \tau_{22} & \rho u_2 u_3 - \tau_{23} \\ \rho u_1 u_3 - \tau_{31} & \rho u_2 u_3 - \tau_{32} & \rho u_3^2 + P - \tau_{33} \\ \rho u_1 H - \tau_{1j} u_j + q_1 & \rho u_2 H - \tau_{2j} u_j + q_2 & \rho u_3 H - \tau_{3j} u_j + q_3 \end{Bmatrix} \tag{2}$$

where ρ is the density, u_i 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 viscous stress tensor and heat flux are defined as:

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

$$q_i = -\gamma\left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t}\right)\left(\frac{\partial E}{\partial x_i} - u_j\frac{\partial u_j}{\partial x_i}\right)$$

where Pr and Pr_t are the Prandtl and turbulent Prandtl numbers, μ is the dynamic viscosity, and μ_t is the dynamic eddy viscosity. The dynamic eddy viscosity is only active when using a turbulence model which is described in the next section. The viscosity μ is a function of the reference viscosity μ_∞ and the temperature given by the Sutherland's formula:

$$\mu = \mu_\infty \left(\frac{RT}{RT_\infty}\right)^{3/2}\left(\frac{RT_\infty + RC}{RT + RC}\right).$$

Since the gas constant R can be eliminated from these expressions it never needs to be defined and $RT = P/\rho$ is used instead. In the Sutherland formula C is a scaled Sutherland constant defined as:

$$RC = \frac{S}{T_{ref}} RT_\infty$$

where $S = 110.\bar{3}K$ is the standard Sutherland constant, $T_{ref} = 300K$ is the reference temperature, and $RT_\infty = P_\infty/\rho_\infty$ is the free-stream temperature. These equations are closed using the ideal gas equation of state:

$$\rho E = \frac{P}{\gamma - 1} + \frac{1}{2}\rho(u_1^2 + u_2^2 + u_3^2)$$

where γ is the ratio of specific heats.

III. Single Equation Turbulence Model

Under turbulent flow conditions the compressible Navier-Stokes equations shown in the previous section can be used to solve the Reynolds Averaged Navier-Stokes (RANS) equations. This is done by adding a dynamic eddy viscosity μ_t to the viscous flux. The dynamic eddy viscosity is defined as:

$$\mu_t = \begin{cases} \mu' \rho \nu_t f_{v1} & \nu_t \geq 0 \\ 0 & \nu_t < 0 \end{cases}$$

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \quad \chi = \frac{\mu' \rho \nu_t}{\mu}$$

where μ' is a scaling constant, and ν_t is the rescaled kinematic eddy viscosity, or the turbulence model working variable. The ν_t variable is solved using the recent variant of the original one-equation Spalart-Allmaras (SA) turbulence model [17] designed to prevent solutions with negative values [18]. The turbulence model equation is given as:

$$\frac{\partial \rho \nu_t}{\partial t} + \frac{\partial \rho \nu_t u_i}{\partial x_i} - \frac{1}{\sigma}\frac{\partial}{\partial x_i}\left[(\mu + \mu' f_n \rho \nu_t)\frac{\partial \nu_t}{\partial x_i}\right] = \rho\,(P - D) + \frac{\mu' c_{b2} \rho}{\sigma}\frac{\partial \nu_t}{\partial x_i}\frac{\partial \nu_t}{\partial x_i} - \frac{1}{\sigma}\left(\frac{\mu}{\rho} + \mu' \nu_t\right)\frac{\partial \rho}{\partial x_i}\frac{\partial \nu_t}{\partial x_i}. \quad (3)$$

All of the terms are described below in the presence of the scaling factor μ' that has been introduced herein. The scaling factor is used to improve the convergence rate of the implicit Newton-Krylov DG-solver as described in [19].

In the negative-SA model the production and destruction terms depend on the sign of the eddy viscosity and are defined as:

$$P = \begin{cases} c_{b1}\,(1 - f_{t2})\,\tilde{s}\nu_t & \nu_t \geq 0 \\ c_{b1}\,(1 - c_{t3})\,s\nu_t & \nu_t < 0 \end{cases}, \quad D = \begin{cases} \mu'\left(c_{w1} f_w - \frac{c_{b1}}{\kappa^2} f_{t2}\right)\left(\frac{\nu_t}{d}\right)^2 & \nu_t \geq 0 \\ -\mu' c_{w1}\left(\frac{\nu_t}{d}\right)^2 & \nu_t < 0 \end{cases}$$

where s is the magnitude of vorticity:

$$s = \sqrt{\omega_i \omega_i},$$

$\tilde{s}$ is the modified vorticity:

$$\tilde{s} = \begin{cases} s + \bar{s} & \bar{s} \geq -c_{v2} s \\ s + \frac{s\left(c_{v2}^2 s + c_{v3}\bar{s}\right)}{(c_{v3} - 2c_{v2})s - \bar{s}} & \bar{s} < -c_{v2} s \end{cases},$$

where,

$$\bar{s} = \frac{\mu' \nu_t f_{v2}}{\kappa^2 d^2}, \quad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}},$$

and d is the distance to the closest wall. The function f_n and the laminar trip term f_{t2} are defined as:

$$f_n = \frac{c_{n1} + \chi^3}{c_{n1} - \chi^3}, \quad f_{t2} = c_{t3} e^{-c_{t4}\chi^2},$$

and the function f_w is defined as:

$$f_w = g\left[\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6}\right]^{1/6}, \quad g = r + c_{w2}\left(r^6 - r\right), \quad r = \min\left(\frac{\mu' \nu_t}{\tilde{s}\kappa^2 d^2}, r_{lim}\right).$$

The main trip term T containing f_{t1} [18] is excluded in the simulations presented in this paper. Lastly, the constants are taken as $\sigma = 2/3$, $c_{b1} = 0.1355$, $c_{b2} = 0.622$, $\kappa = 0.41$, $c_{w1} = c_{b1}/\kappa^2 + (1 + c_{b2})\,\sigma$, $c_{w2} = 0.3$, $c_{w3} = 2$, $c_{v1} = 7.1$, $c_{v2} = 0.7$, $c_{v3} = 0.9$, $c_{t1} = 1$, $c_{t2} = 2$, $c_{t3} = 1.2$, $c_{t4} = 0.5$, $r_{lim} = 10$, $c_{n1} = 16$.

IV. DG Formulation

A. Discretization

In this section, the discontinuous Galerkin (DG) finite element formulation used to solve the RANS equations is described. DG uses a basis that is continuous within an element but discontinuous between elements. This allows for great flexibility in the choice of bases. In this work, a hierarchical, modal basis is chosen. Each element can have a different polynomial degree p for the solution and q for the geometrical mapping. However, all of the simulations in this paper use a mapping basis of polynomial degree q which is at least $p+1$ or more. A higher-order mapping basis is needed due to the use of curved meshes, which are described in later sections.

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

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

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

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

where ϕ are the basis functions and the solution is approximated using $U_m = \phi_s a_{ms}$ where the index r and s run over the number of basis functions. The source term S_m only appears in the Spalart-Allmaras turbulence equation. The residual now contains integrals over faces Γ and special treatment is needed for the fluxes F^*_{mi} in these terms. The advective fluxes can be calculated using any of the following choices: Lax-Friedrichs [20], Roe [21], and artificially upstream flux vector splitting scheme (AUFS) [22]. The results in this paper use the Lax-Friedrichs flux and the diffusive fluxes are handled using a symmetric interior penalty (SIP) method [23, 24].

B. Solution Method

To solve the non-linear set of equations, a damped Newton-Rhapson 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 J_{mrns} is a block Jacobian matrix, k is the non-linear iteration, M_{rs} is a mass matrix, and Δt is an element-wise time step which is used for continuation of the solution to promote robust Newton convergence [25]. The mass matrix M_{rs} 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 h is a mesh size and c is the speed of sound. The mesh size h is defined as:

$$h = \frac{V_{cell}}{A_{face}(p+1)^2},$$

where V_{cell} is the cell volume and A_{face} is the surface area of the faces on the cell. The Newton-Rhapson method results in a linear system that must be solved at each non-linear iteration to obtain the update Δa^k_{ns} to the solution coefficients a_{ns} as:

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

where σ is a parameter between 0 and 1, whose optimal value is determined using a line search approach [19]. To solve the linear system in equation (4), a flexible-GMRES [26] (FGMRES) method is used. To further

improve convergence of FGMRES a preconditioner is applied to the system of equations. Preconditioners that have been implemented include Jacobi relaxation, Gauss-Seidel relaxation, line-implicit Jacobi, and ILU(0). The CFL number is used to control the convergence characteristic of the linear implicit scheme at each non-linear step and is ramped to high values in the final stages of convergence in order to recover the quadratic properties of a full Newton scheme.

V. 3D Hemisphere Cylinder Validation Case

The first test case investigated is turbulent flow over a smooth body of revolution. The body of revolution consists of a cylinder capped at the nose with a hemisphere; a slice of the domain is shown in Figure 1 (reproduced from the TMR website). This case is part of the TMR and also has experimental data collected at various angles and Mach numbers which can be used for validation purposes [27]. For comparison and verification the TMR website has provided the initial conditions, grids, and results from multiple finite-volume codes using the negative-SA single-equation turbulence model [18].

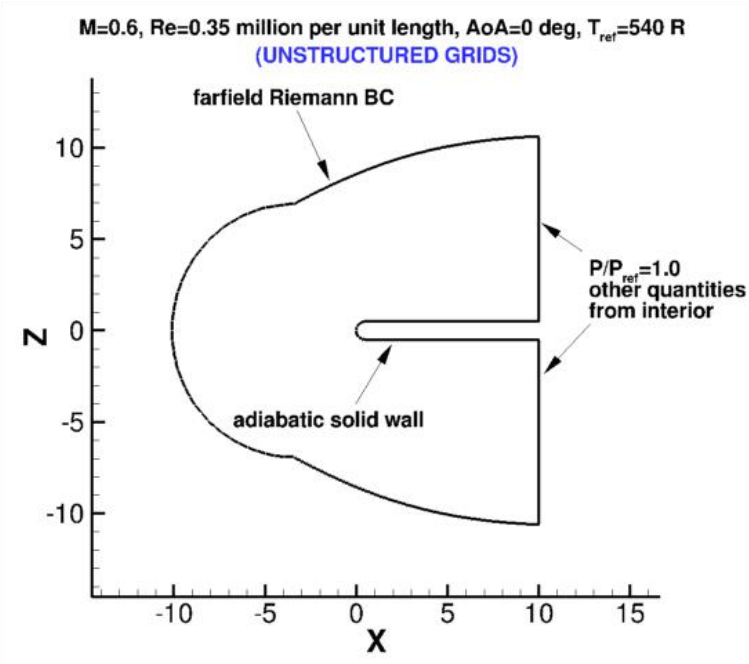


Figure 1: Domain for the hemisphere-cylinder reproduced from the TMR website.

A. Initial and Boundary Conditions

The free-stream conditions are taken from the NASA turbulence modeling resource. The flow conditions are: angle of attack $\alpha = 0°$, $Mach = 0.6$, $Re = 3.5 \times 10^5$, $\gamma = 1.4$, $Pr = 0.72$, and $Pr_t = 0.9$. The problem is initialized using uniform flow at the free-stream conditions. The turbulence equation is initialized everywhere as $\rho\nu_{t_\infty} = 3\mu_\infty/\mu'$. The far-field boundary conditions consist of non-reflecting Riemann invariants where the far-field values are taken as the free-stream conditions given above. For the outflow a constant pressure boundary condition is prescribed and the walls are assumed no-slip and adiabatic.

B. Grids

The cylinder surface mesh is created using a structured quadrilateral mesh and the hemisphere mesh is created using a unstructured triangular mesh. These surfaces are extruded outward to create hexahedral and prismatic cells. The unstructured grids for this case were created using the provided Fortran program on the TMR website. The inputs for the program for all grids are: target $y^+ = 0.5$, outer boundary distance equals 10, and the length of hemisphere-cylinder equals 10. The remaining inputs are different for each grid

and are shown in Table 1. Also, shown in the table are the number of hexahedra and prisms contained in each grid. The grid number in the table will be used in the results section to distinguish between simulations. Also, the degrees of freedom can be calculated using the formula:

$$DOF = n_{hex}(p+1)^3 + n_{prism}\frac{(p+1)^2(p+2)}{2}$$

where n_{hex} is the number of hexahedra, n_{prism} is the number of prisms, and p is the polynomial degree of the simulation.

Table 1: Grid descriptions with inputs and outputs of unstructured mesh generation routine and polynomial degrees.

Grid	Cylinder nodes	Division sectors	n_{hex}	n_{prism}	p
1	400	80	17,856,000	3,571,200	1
2	200	40	2,688,000	537,600	1, 2
3	100	20	372,000	74,400	1, 2, 3
4	50	10	42,000	8,400	2, 3, 4

C. Wall Distance

The wall distance for this simulation is calculated using the analytical definition of the geometry. For elements near the wall the curvature of the mesh does not exactly represent the geometry, this is because the mapping is defined by polynomials and the geometry is defined by a cylinder and hemisphere. For these elements the analytic wall distance could be negative or inaccurate. To fix these problematic cells, the wall distance is calculated on equidistant points on the element (the number of points is equal to the number of mapping basis functions), a Vandermode matrix is created and used to solve for a set of coefficients to represent the wall distance, and lastly the wall distance is projected onto the quadrature points.

D. Mesh Curving

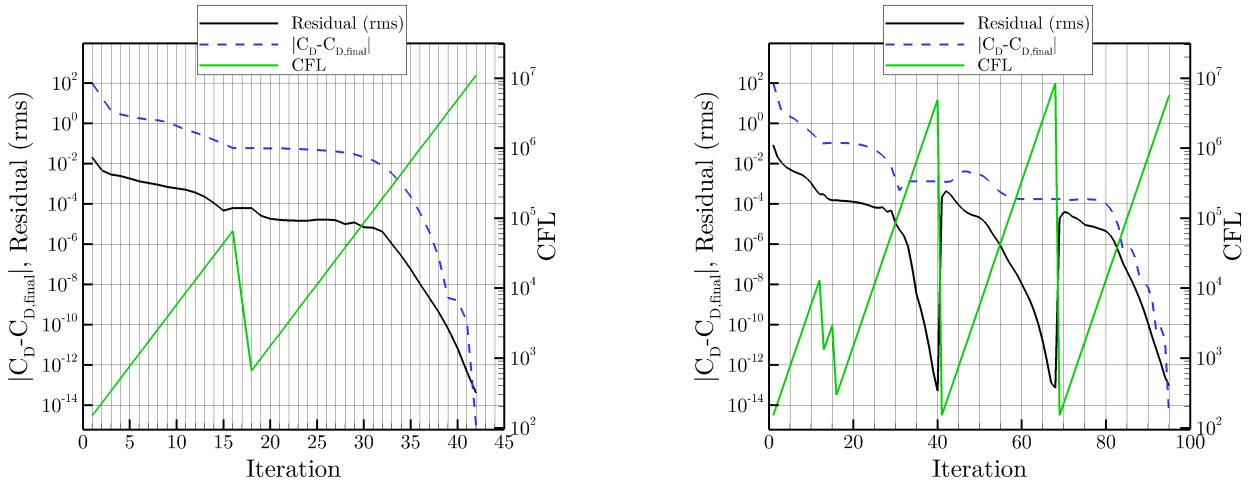
For discontinuous Galerkin discretizations of polynomial degree p, curved mesh elements of polynomial degree $q = p + 1$ must be used to obtain the full accuracy benefits of the discretization. To achieve the $p + 1$ curvature, the mesh is curved using the analytic definition of the geometry. To apply this curvature, the elements surrounding the hemisphere-cylinder are first initialized using straight-sided elements. Then the straight-sided elements are used to project equidistant points including the original corner nodes of the element (the number of points is equal to the number of mapping basis functions). The points on the corners of the elements already satisfy the analytic formula, while the remaining nodes are perturbed on the boundary face to satisfy this formula. This perturbation is pushed to the other nodes on the element in the normal direction. A Vandermode matrix is created and used to solve for the modal coefficients used in the mapping basis. This curvature is then pushed outward in the normal direction along lines from the surface to neighboring elements. This process is repeated for every element on the boundary.

E. Results

Nine simulations were carried out for the hemisphere-cylinder varying the polynomial degree and the grid size (see Table 1). For all of these simulations, partitioning of the grid is performed based on lines. Since DG is a cell-based discretization, the lines are based on the cells. On the cylinder, hexahedra form the lines from the surface all the way to the outer boundary and similarly on the hemisphere prisms form the lines. The lines allow for a line-implicit Jacobi iterative method to be used as a preconditioner to the FGMRES linear solver. The preconditioner is applied a maximum of 200 iterations or until the residual drops 2 orders of magnitude. If the preconditioner begins to diverge (relative residual rises to 1.5) the iterations are stopped prematurely and FGMRES moves on to the next Krylov vector. Each linear system is solved 5 orders of

magnitude and if the linear system fails to converge then the CFL is lowered. Also, if a change in pressure or density is too large (greater than 20%) the CFL is lowered [19].

Figure 2a shows the iterative convergence of the simulation using grid 1 (finest grid) and polynomial degree $p = 1$. The CFL number ramps up quickly until the turbulence model causes large changes in the solution at about iteration 16, the CFL adjusts and then begins to climb again. Newton convergence begins at about iteration 32 and a total of 42 iterations are needed to reach machine precision. Also, shown in this Figure is the iterative convergence of the drag which, as expected, closely follows the iterative convergence of the non-linear residual. This simulation has approximately 1.6×10^8 DOF's per solution variable solved on 16,384 processors in approximately 2.5×10^4 seconds (TauBench = 8.4 seconds). Figure 2b shows the iterative convergence for grid 3 and polynomial degrees $p = 1, 2, 3$. All three simulations are shown on the same Figure in order of polynomial degree. Similar to grid 1, $p = 1$, it takes around 40 non-linear iterations to reach machine precision. The $p = 2$ simulation is initialized using the $p = 1$ solution and then the $p = 3$ simulation is initialized using the $p = 2$ solution. For both the $p = 2$ and $p = 3$ simulations, the CFL increases at each iteration and the non-linear residual decreases rapidly to machine precision. For these higher order simulations faster convergence could probably be attained by starting with a higher initial CFL number. All three of these simulations are performed on 4,096 processors. For $p = 1$, there are 3.4×10^6 DOF's solved in 638 seconds, for $p = 2$, there are approximately 1.1×10^7 DOF's solved in 6,437 seconds, and for $p = 3$, there are approximately 2.6×10^7 DOF's solved in 3.9×10^4 seconds. The iterative convergence for the other simulations are not shown but have similar trends to the ones described here.



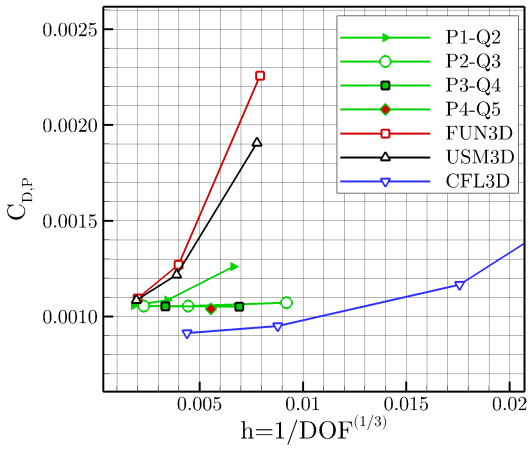
(a) Grid 1 and polynomial degree $p = 1$.

(b) Grid 3 and polynomial degrees $p = 1, 2, 3$.

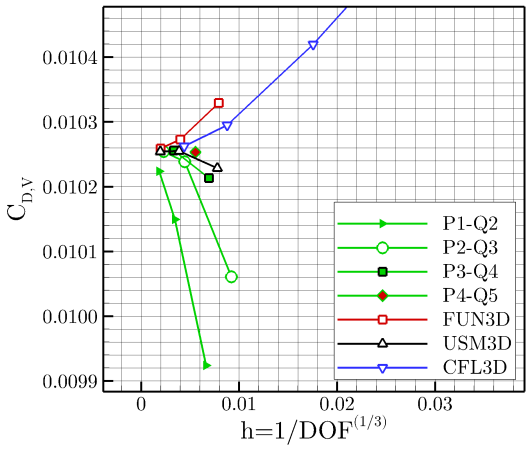
Figure 2: Iterative convergence of hemisphere-cylinder non-linear residual, CFL, and C_D.

For verification purposes a mesh resolution study is performed on drag coefficient. Figure 3a shows the drag coefficient contribution from pressure for all nine simulations compared to FUN3D, USM3D, and CFL3D. From this Figure it appears that the finite-volume solvers are approaching a mesh resolved solution but are still a few mesh refinements away. The DG simulations show a similar trend for $p = 1$ and are slightly better than the finite volume solvers. For the higher polynomial degrees the DG simulations are mesh converged even on the coarser meshes. This demonstrates that higher-order methods achieve better accuracy with fewer degrees of freedom compared to lower-order methods. Figure 3b shows the friction drag component of the drag coefficient. The finite volume solvers, the $p = 1$ DG simulations, and the $p = 2$ DG simulations are still not mesh converged. The finite volume solvers are more accurate than the $p = 1$ DG simulations. Only the $p = 3$ and $p = 4$ DG simulations appear to be close to the mesh converged solution. Figure 3c shows the total drag coefficient and show similar trends to the drag coefficient contribution from pressure shown in Figure 3a. This is because even though the magnitude of drag coefficient from friction drag is larger, the variations are smaller compared to the variations in the contribution from pressure. In this Figure only the $p = 2$ and $p = 3$ DG simulations are close to mesh resolved. To be within one drag

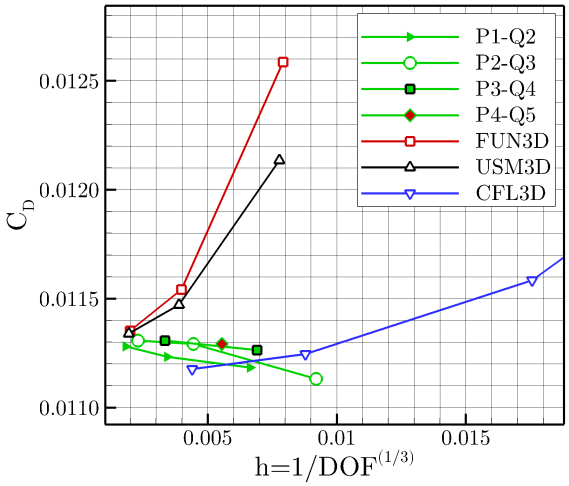
count of the mesh converged answer grid 3, $p = 3$ and grid 4, $p = 4$ DG simulations use the least amount of degrees of freedom.



(a) Drag coefficient contribution from pressure.



(b) Drag coefficient contribution from friction drag.

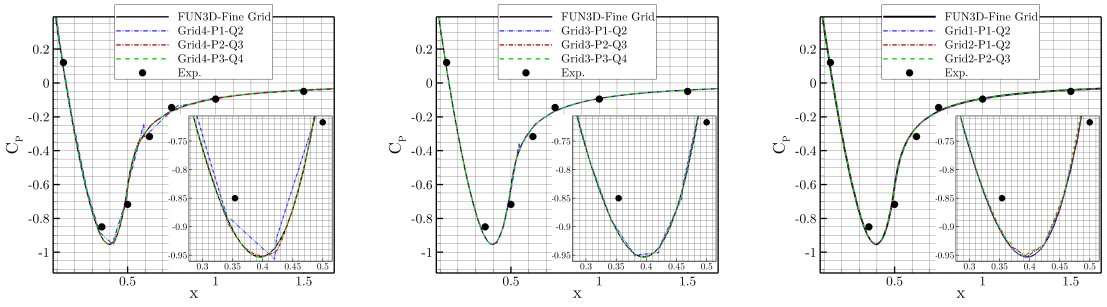


(c) Total drag coefficient.

Figure 3: Drag coefficient versus mesh size for hemisphere-cylinder compared to FUN3D, USM3D, and CFL3D

The computed surface pressure coefficients (C_P) are shown in Figure 4 for all nine simulations. The C_P is plotted along a line starting from the nose of the hemisphere along the axial direction (x-direction). All of the simulations show similar results to the experimental data. When comparing to the FUN3D results using the finest grid the DG simulations on the coarser grids 3 and 4 with lower polynomial degrees are not resolved enough. However, the simulations on Grids 1 and 2, shown in Figure 4c, are indistinguishable from each other at this plotting resolution.

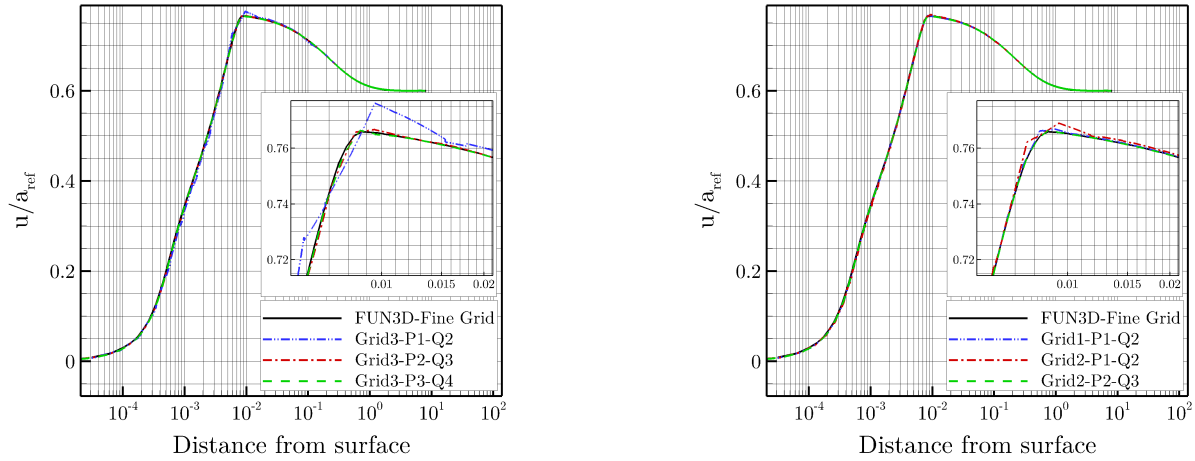
Wake profiles are also provided on the NASA turbulence modeling resource and compared to the DG simulations. Detailed wake profiles of C_P, x-velocity, and eddy viscosity are shown at $x/D = 0.5$, where $D = 1$ is the diameter of the cylinder. For these variables the solution is axisymmetric so they are plotted as distance from the surface radially outward. Figure 5 shows the x-velocity wake profile and Figure 6 shows



(a) (b) (c)

Figure 4: Computed surface pressure coefficients C_P for the hemisphere-cylinder on grids 1-4 compared to experimental data and FUN3D results on the finest grid.

the eddy viscosity working variable wake profile. Only grids 1-3 are shown in these Figures and are compared to FUN3D results using the finest grids. Grids 2-3 and polynomial degree $p = 1$ are not resolved enough. The remaining simulations are nearly indistinguishable from each other and FUN3D results.



(a) (b)

Figure 5: x-velocity wake profile at $x/D = 0.5$ for the hemisphere-cylinder on grids 1-3 compared to FUN3D results on the finest grid.

VI. 3D Modified Bump-in-channel Verification

The second test case investigated is turbulent flow over a modified bump in a channel. The modified bump is similar to the original two-dimensional bump except there are now span-wise variations making it a truly three-dimensionally problem. For comparison and verification the NASA turbulence modeling resource has provided the initial conditions, grids, and results from multiple finite-volume codes using the negative-SA model.

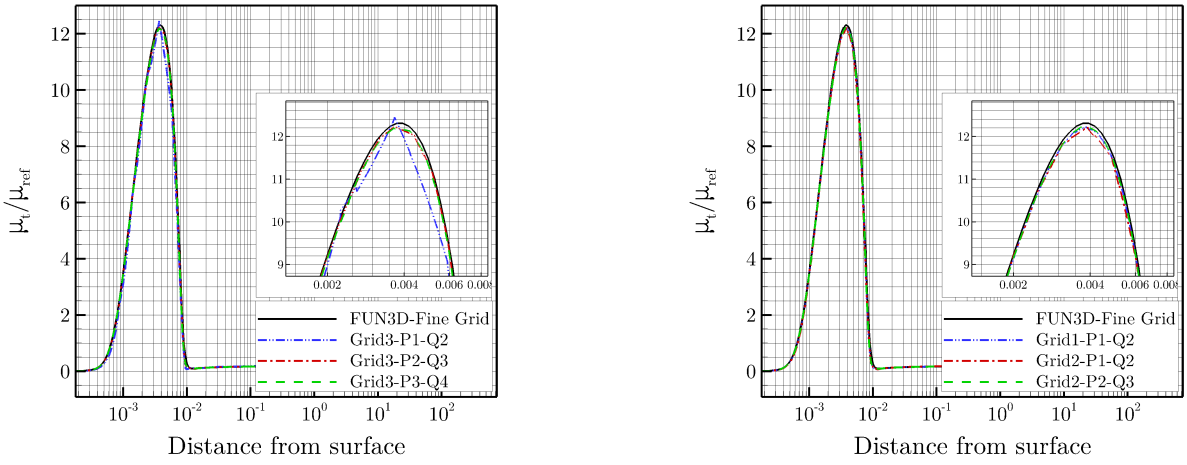


Figure 6: Eddy viscosity working variable wake profile at $x/D = 0.5$ for the hemisphere-cylinder on grids 1-3 compared to FUN3D results on the finest grid.

A. Initial and Boundary Conditions

The free-stream conditions are taken from the NASA turbulence modeling resource. The flow conditions are: angle of attack $\alpha = 0^o$, $Mach = 0.2$, $Re = 3 \times 10^6$, $\gamma = 1.4$, $Pr = 0.72$, and $Pr_t = 0.9$. The problem is initialized using uniform flow at the free-stream conditions. The turbulence equation is initialized everywhere as $\rho\nu_{t_\infty} = 3\mu_\infty/\mu'$. The far-field boundary conditions consist of non-reflecting Riemann invariants where the far-field values are taken as the free-stream conditions given above. For the outflow a constant pressure boundary condition is prescribed and the walls are assumed no-slip and adiabatic.

B. Grids

The structured grids for this case were created using the provided Fortran program on the NASA turbulence modeling resource website. The bottom surface of the grid at the bump is shown in Figure 7 (taken from TMR website). The inputs for the program are the original 2D bump grids and the 3D grid is made by lofting the 2D grid along an analytic function (equations are shown in Figure 7). Table 2 shows the number of hexahedra contained in each grid. The grid number in the table will be used in the results section to distinguish between simulations. The degrees of freedom can be calculated using the formula:

$$DOF = n_{hex}(p+1)^3$$

where n_{hex} is the number of hexahedra and p is the polynomial degree of the simulation.

C. Wall Distance

A brute force search method is used to calculate wall distance. This is performed by calculating the distance from every quadrature point to every triangular face on the wall boundary surface. Since the boundary is comprised of quadrilaterals, each quadrilateral is split into two sets of two triangles. This is because a quadrilateral may not be planar and both orientations need to be checked for distance. For each quadrature point the distance from the point to a planar triangular surface is calculated and the minimum distance is saved. Since this is performed on straight faces the wall distance is calculated before the mesh is curved. The assumption is that when the mesh is curved later the quadrature point and the face move approximately the same distance and direction. The brute force method has been compared to an analytic approach on different simple geometries and both methods yield similar wall distance and do not significantly change the solution.

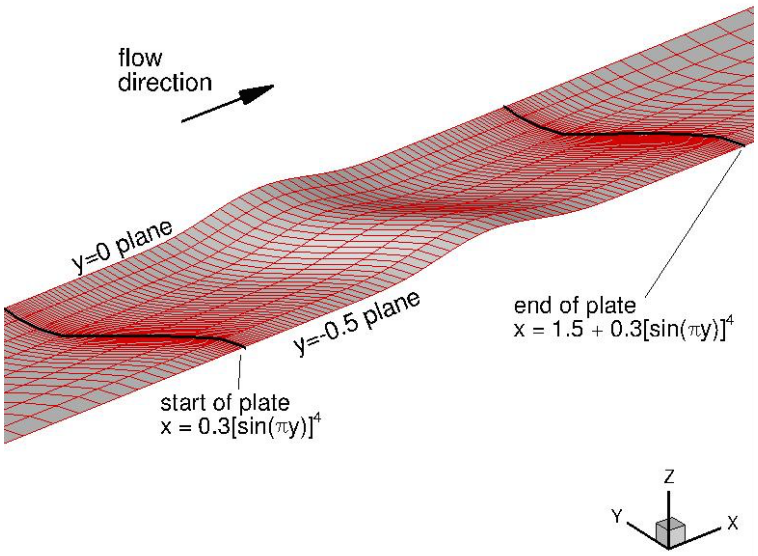


Figure 7: 3D modified-bump grid along bottom surface (taken from TMR website).

Table 2: Grid numbering and output of the structured mesh generation routine along with polynomial degrees simulated on each grid.

Grid	$n_y \times n_x \times n_z$	n_{hex}	p
2	$16 \times 352 \times 160$	901,120	1
3	$8 \times 176 \times 80$	112,640	1, 2
4	$4 \times 88 \times 40$	14,080	1, 2, 3
5	$2 \times 44 \times 20$	1,760	1, 2, 3

D. Mesh Curving

A parallel linear-elasticity solver is used to curve the modified bump mesh. The linear-elasticity solver is based on a continuous Galerkin discretization and has line-solvers, multigrid, and high-order mixed element capabilities. The boundary conditions for the solver are specified by using surfaces from finer grids provided by the NASA turbulence resource website. For example, to curve grid 5 ($2 \times 44 \times 20$ hexahedra) with a quadratic mapping, the surfaces from grid 4 ($4 \times 88 \times 40$ hexahedra) are used and for a quartic mapping the surfaces from grid 3 ($8 \times 176 \times 80$ hexahedra) are used. For the $p = 1$ modified bump simulations a quadratic mapping ($q = 2$) is used and for the $p = 2, 3$ simulations a quartic mapping ($q = 4$) mapping is used (note this is higher than $p + 1$ for the $p = 2$ simulation).

E. Results

Nine simulations were carried out for the modified-bump test case, varying polynomial degree and grid size (see Table 2). For all of these simulations, partitioning of the grid is performed based on lines. Since DG is a cell-based discretization, the lines are based on cells. The lines allow for a line-implicit Jacobi iterative method to be used as a preconditioner to the FGMRES linear solver. The preconditioner is applied a maximum of 200 iterations or until the residual drops 2 orders of magnitude. If the preconditioner begins to diverge (relative residual rises to 1.5) the iterations are stopped prematurely and FGMRES moves on to the next Krylov vector. Each linear system is solved 5 orders of magnitude and if the linear system fails to converge then the CFL is lowered. Also, if a change in pressure or density is too large (greater than 20%) the CFL is lowered [19].

Figure 8 shows the iterative convergence of the simulation using grid 4 and polynomial degrees $p = 1, 2, 3$. In this series of simulations, the $p = 1$ simulation is initialized with free-stream flow, while the $p = 2$ simulation is initialized using the $p = 1$ solution and then once again the $p = 3$ simulation is initialized using the $p = 2$ solution. For the first simulation ($p = 1$) the CFL number ramps up quickly until the turbulence model causes large changes in the solution at about iteration 15, the CFL adjusts and then begins to climb again. Newton convergence begins at about iteration 42 and a total of 48 iterations are needed to reach machine precision. Also, shown in this Figure is the iterative convergence of the drag which, as expected, closely follows the iterative convergence of the non-linear residual. The $p = 2$ simulation struggles to converge with non-monotone CFL ramping and does not achieve Newton-like quadratic convergence, although it does achieve a final residual norm of 10^{-10}. Consequently, two changes were made to the inputs of the solver to improve the convergence of the $p = 3$ simulation. The first is using a lower convergence tolerance on the linear system, to only requiring 3 orders of relative residual reduction. This will allow the CFL number to grow larger than if 5 orders or convergence were required. The second change is an increase in the number of quadrature points in the cell and face. It is believed (but not fully analyzed) that there are enough non-linearities in the mapping used for curving the mesh to cause issues in the non-linear convergence of the solver.

All three of these simulations (grid 4, $p = 1, 2, 3$) are performed on 256 processors. For $p = 1$, there are 1.1×10^5 DOF's solved in 981 seconds, for $p = 2$, there are 3.8×10^5 DOF's solved in 43,200 seconds, and for $p = 3$, there are 9.0×10^5 DOF's solved in 55,092 seconds. Typically, the $p = 3$ simulations are more time consuming compared to the $p = 2$ simulations, however with the changes to the solver inputs, Newton convergence is obtained and this speeds up the overall time to solution. Similar non-linear convergence issues occurred on finer grids and the solver inputs had to be modified accordingly. For grid 2 it was observed that the non-linear convergence path could lead to a negative pressure if the CFL was too large in the beginning of the simulation. This might attributed to using a line-solver preconditioner perpendicular to the flow direction and the lack of multigrid. As the grids get finer it takes more iterations for information to travel downstream. Therefore, the CFL must not grow too quickly until the wake is developed enough.

For verification purposes a mesh resolution study is performed on drag and lift coefficient. Figure 9 shows the drag coefficient broken into components, total drag coefficient, and lift coefficient for all nine simulations. Looking at the drag coefficient contribution from pressure in Figure 9a the $p = 1$ DG simulations converges similarly to CFL3D. All of the high-order DG simulations are mesh converged and are within one drag count of the mesh converged drag coefficient contribution from pressure. On the other hand, the drag coefficient contribution from friction drag does not look mesh converged and appears to be converging to a slightly different mesh converged solution although it is only around 0.25 of a drag count away from the finite-volume solvers. Figure 9c shows similar trends to the drag coefficient contribution from pressure. The $p = 1$

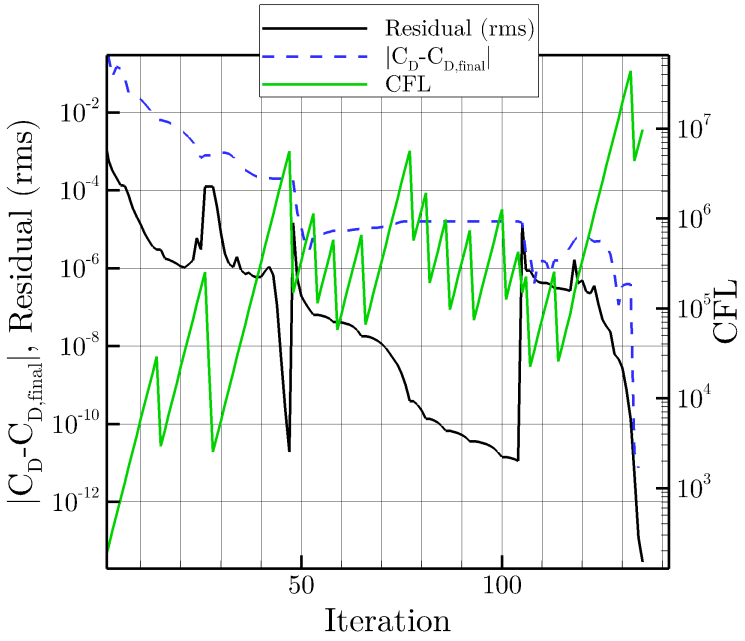


Figure 8: Iterative convergence of non-linear residual, CFL, and C_D for grid 4, $p = 1, 2, 3$

DG simulations are similar to the CFL3D results and the higher-order DG simulations appear all mesh resolved and within a drag count. Figure 9d shows the lift coefficient for all nine simulations. All of the DG simulations are converging to similar lift coefficients as FUN3D and USM3D. The $p = 2, 3$ DG simulations are converging to the mesh resolved lift coefficient using fewer degrees of freedom compared to the $p = 1$ simulation and all of the finite-volume solvers.

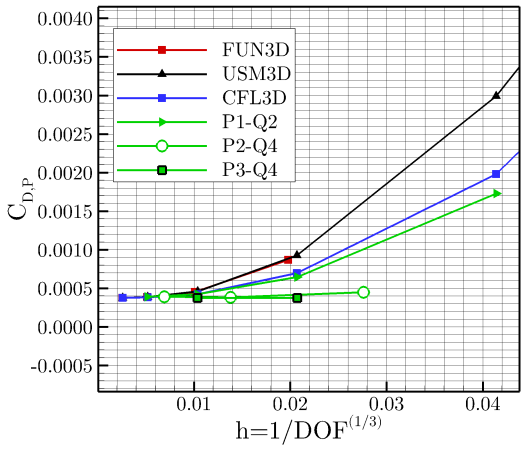
The computed surface pressure coefficients (C_P) are shown in Figure 10 for six out of nine simulations for the modified-bump problem. The C_P is plotted along a line in the axial direction (x-direction) at $y = 0$. Only grids 2-4 are shown because the grid 5 solutions are not resolved enough, due to only two elements being used in the span direction. When comparing to the FUN3D results only the grid 3, $p = 2$ and the grid 4, $p = 3$ simulations are mesh resolved. The remaining simulations are still not resolved enough. This is surprising given the accuracy shown for almost all cases in the mesh resolution study. This shows the importance of showing solution profiles and not just integrated values.

VII. High Lift Prediction of the DLR-F11

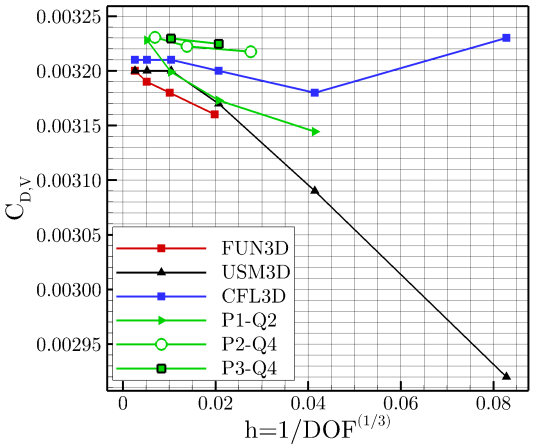
The second AIAA CFD High Lift Prediction Workshop (HiLiftPW-2) was held in June 2013. The focus of this workshop was on the validation of numerical methods for a realistic high-lift configuration of a modern transport aircraft. The DLR-F11 experiment was chosen for this purpose [28]. The DLR-F11 is a wing-body configuration with a full span leading-edge slat and single-slotted trailing-edge flap. A slat deflection of 26.5 degrees and a flap deflection of 32 degrees are specified for the workshop test cases. In this section, the results of the DLR-F11 simulation using the DG solver will be compared to results from a state-of-the-art finite-volume solver: NSU3D [29]. It is important to note that the DG solver uses the negative variant of the SA turbulence model and NSU3D uses the original SA model.

A. Initial and Boundary Conditions

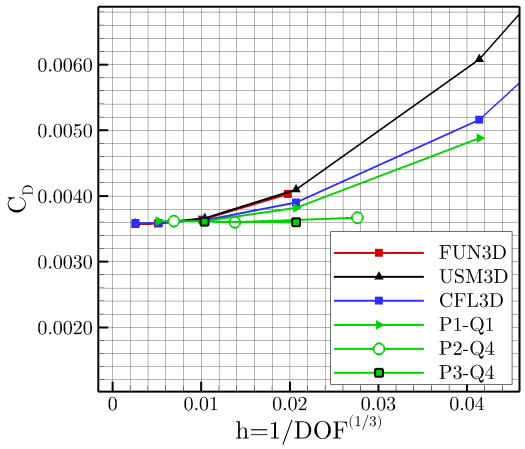
The flow conditions are: angle of attack $\alpha = 7^o$ and $\alpha = 16^o$, $Mach = 0.175$, $Re = 1.51 \times 10^7$, $\gamma = 1.4$, $Pr = 0.72$, and $Pr_t = 0.9$. The problem is initialized using uniform flow at the free-stream conditions. The turbulence equation is initialized everywhere as $\rho\nu_{t_\infty} = 3\mu_\infty/\mu'$. The far-field boundary conditions consist of non-reflecting Riemann invariants where the far-field values are taken as the free-stream conditions given



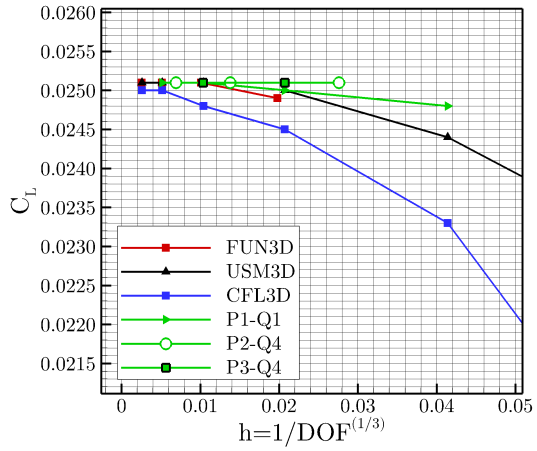
(a) Drag coefficient contribution from pressure.



(b) Drag coefficient contribution from friction drag.

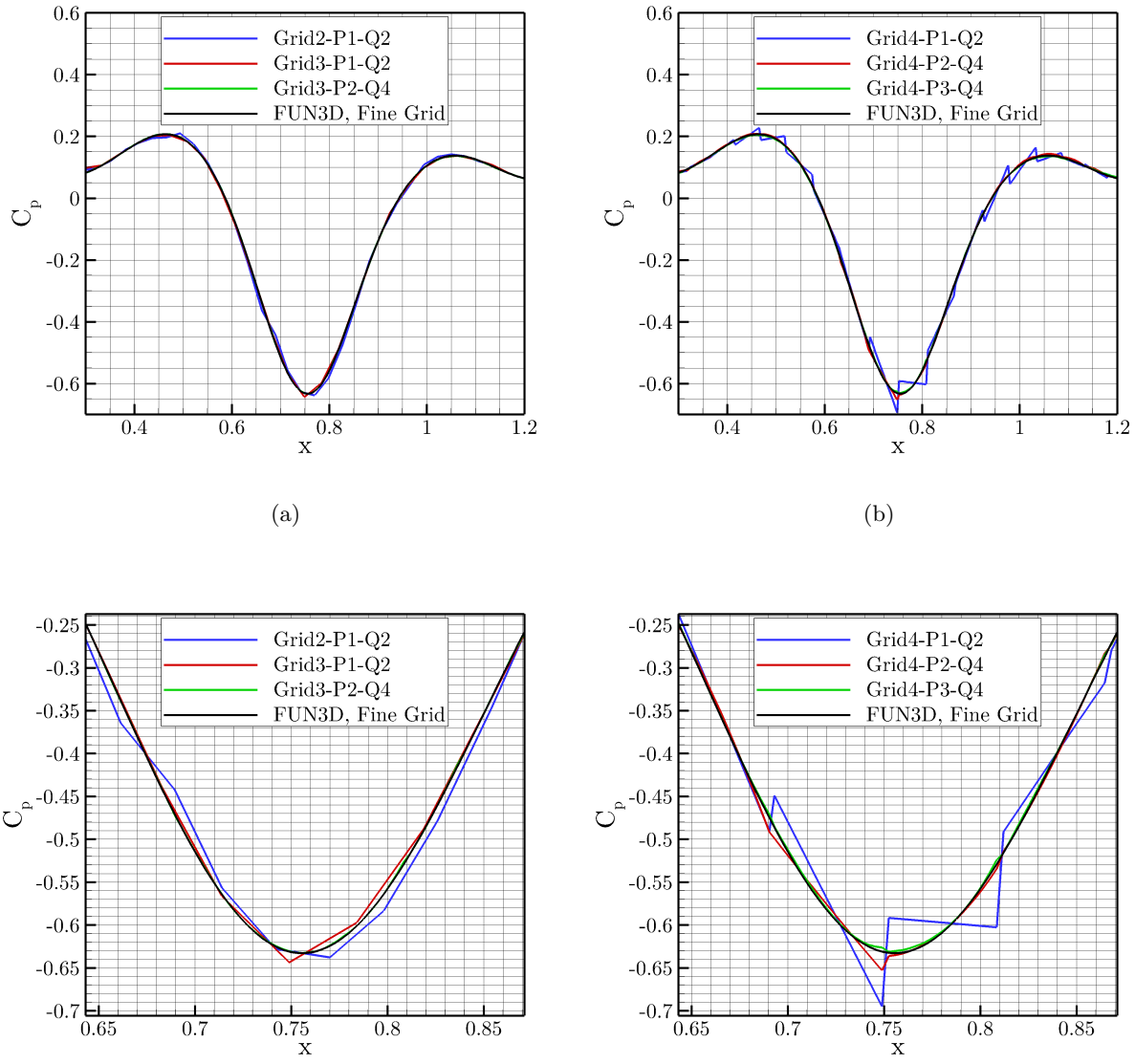


(c) Total drag coefficient.



(d) Total lift coefficient.

Figure 9: Drag and lift coefficient versus mesh size for the modified-bump compared to FUN3D, USM3D, and CFL3D.



(a)

(b)

(c) Close up view.

(d) Close up view.

Figure 10: Computed surface pressure coefficients C_P for the modified-bump on grids 2-4 compared to FUN3D results on the finest grid.

above.

B. Grid, Wall Distance, and Mesh Curving

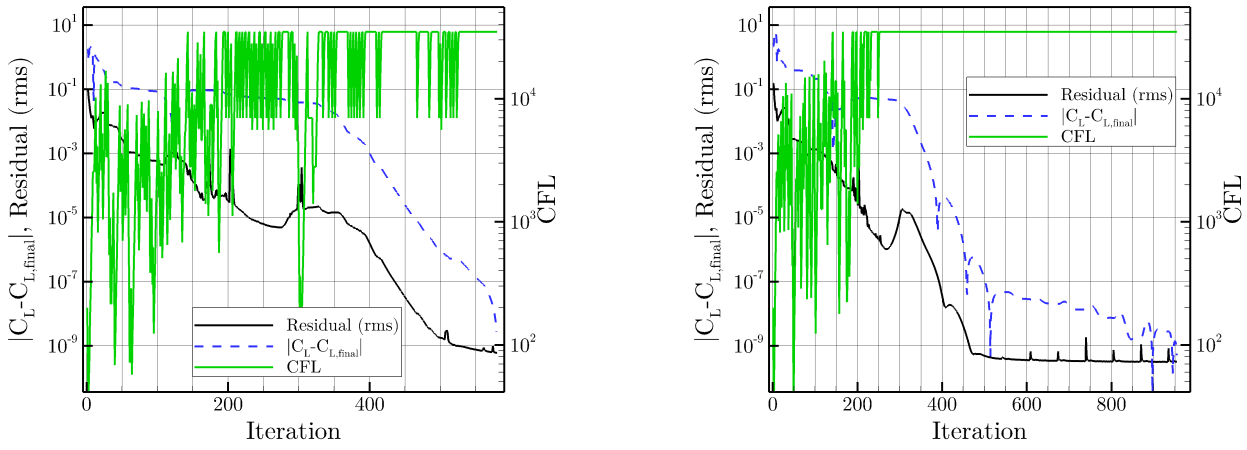
The geometry for this simulation comes from Case 1 of the HiLiftPW-2 and consists of a simple clean wing-slat-flap system with no fairings, supports or brackets. The grid is generated using VGRID and is similar to the grids used for the workshop [29]. The grid consists of 2.8×10^6 prisms, 4.4×10^4 pyramids, and 3.0×10^6 tetrahedra. The grid is shown in Figure 14. Compared to the grids in [29] this grid is coarser than the coarsest grid but because of the cell-based DG discretization, the degrees of freedom, using a $p = 1$ DG discretization, are comparable to the medium grid using a finite-volume discretization. The wall distance is calculated using a brute force search method on straight edged triangular faces. This is identical to the wall distance calculation discussed previously in section VI.C.

The DLR-F11 mesh is curved by first subdividing each face on the surface into self similar elements. This creates a new set of points which are snapped back onto the CAD geometry using Open Cascade [30]. However, the new points can cause negative volumes in the DG simulation. To address this, the newly created points are treated as boundary conditions to the linear elasticity solver previously discussed in VI.D. The linear elasticity solver curves the volume cells such that all of the negative volumes are properly eliminated.

C. Results

Two simulations were carried out for the DLR-F11 high-lift configuration varying only angle of attack ($\alpha = 7°$ and $16°$). For these simulations partitioning of the grid is performed based on lines on the prismatic boundary layer cells only. The preconditioner used in these cases is ILU(0) and the linear solver is FGMRES. For the FGMRES solver, 200 Krylov vectors are used with up to 40 restarts. Each linear system is converged three orders of magnitude and if the linear system fails to converge, then the CFL is lowered. Also, if a change in pressure or density is too large (greater than 20%) the CFL is lowered [19]. Along with these restrictions, a maximum CFL of 3.5×10^4 is enforced to avoid lack of convergence of the linear system and a subsequent drop in CFL number.

Figure 11a shows the iterative convergence of the DLR-F11 at angle of attack $\alpha = 7°$ and Figure 11b for angle of attack $\alpha = 16°$. Both simulations have been run at a polynomial degree of $p = 1$ with a total of 2.9×10^7 DOF's per solution variable. The simulations are solved on 8192 processors and each has taken around 6.5 hours (TauBench = 8.4 seconds) and 550 non-linear iterations to reduce the non-linear residual nine orders in magnitude. Compared to NSU3D, which takes approximately 1.5 hours on 1024 processors to run a similar case, there is still work needed to increase the solver efficiency of DG.

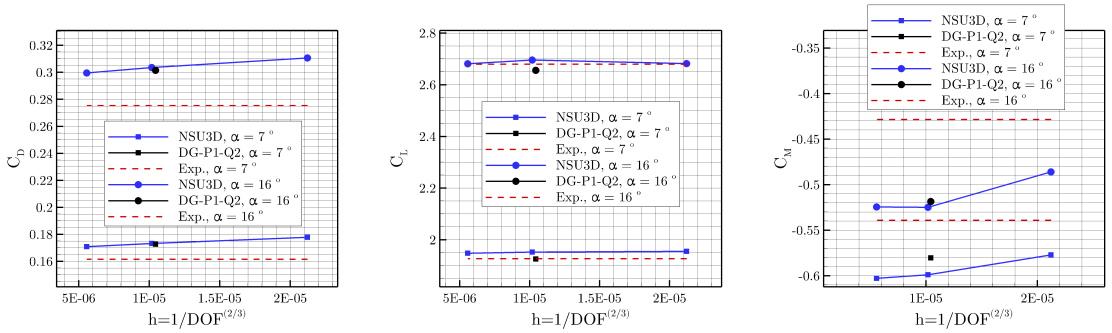


(a) Angle of attack 7°

(b) Angle of attack 16°

Figure 11: Iterative convergence of non-linear residual, C_L, and CFL for the DLR-F11 high-lift configuration.

A mesh resolution study can not be performed with only one data point for each angle of attack. Therefore, a strong argument can not be made when comparing the forces with the experiments and NSU3D data. Figure 12a shows the drag coefficient and the DG simulations show very similar results to the medium NSU3D grids. Figure 12b shows the lift coefficient and the DG simulation matches closer to the experiment at the lower angle of attack but slightly farther from the experimental value compared to NSU3D for the higher angle of attack. Figure 12c shows the Moment coefficient and both the DG simulation and NSU3D are far away from the experimental values. The DG simulation is similar to NSU3D at the higher angle of attack and somewhere in between the experiment and NSU3D at the lower angle of attack. Again, it is hard to draw many conclusions since there is only one data point for the DG simulations.



(a) Drag coefficient. (b) Lift coefficient. (c) Moment coefficient.

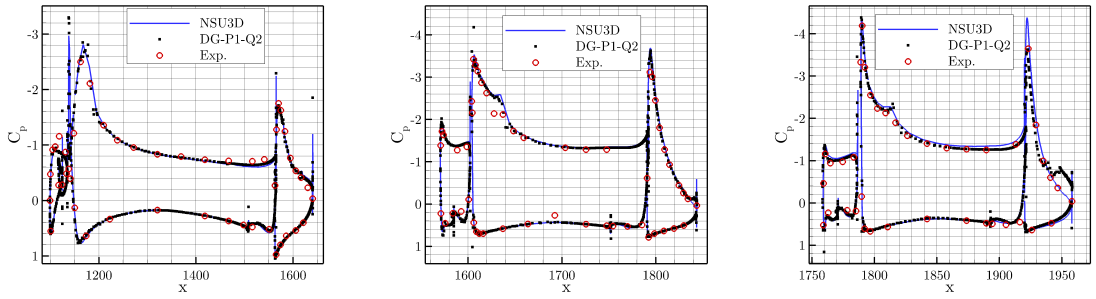
Figure 12: Coefficients of drag, lift, and moment for the DLR-F11 at angles of attack $\alpha = 7°$ and $16°$ compared to experiments and NSU3D.

The computed surface pressure coefficients (C_P) are shown in Figure 13 for the DLR-F11 at stations 1,6 and 10 for both angles of attack. Overall, the DG simulations compare well with NSU3D and the experiments. The only major discrepancies occur at station 10, $\alpha = 7°$ on the flap where a region of separation is observed in the DG simulation but not in the experiment or NSU3D. Further investigation into these discrepancies showed that FUN3D also observed separation at station 10 and $\alpha = 16°$ [31]. But they did not observe this until they performed several mesh adaption steps (using adjoint) resulting in 6×10^7 degrees of freedom. Since FUN3D only performed adaption at the higher angle of attack it is not clear if the separation should be observed at the lower angle of attack near station 10 as seen in the DG simulations. Contours of C_P are shown on the surface of the DLR-F11 in Figure 14 for $\alpha = 16°$. Clearly, in the outboard span area of the wing there is a region of lower C_P on the flap in the DG simulation.

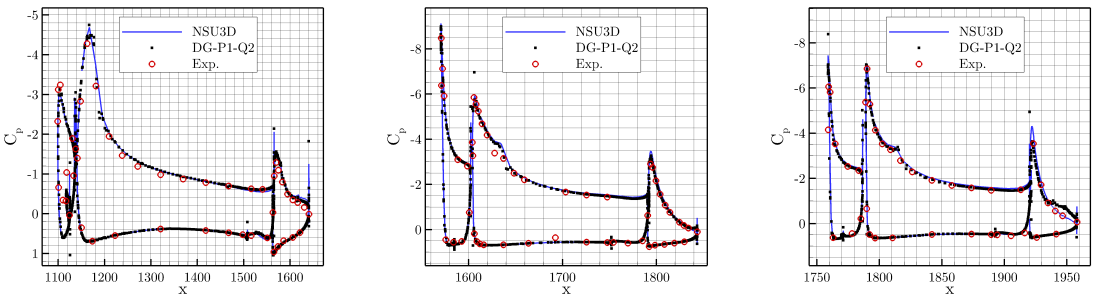
VIII. Conclusions

In this work, mesh resolved simulations are performed of the 3D hemisphere-cylinder and the 3D modified-bump. The NASA TMR website provides grids and reference solutions for the FUN3D, CFL3D, and USM3D solvers. These solutions are compared to DG simulations using a p-refinement study and a h-refinement study. Overall, the DG solver compares well with the finite-volume solvers. For the hemisphere, the finite-volume solvers are not fully mesh resolved. However on the finer grids and with higher polynomial degrees the DG simulations are mesh resolved. The bump is closer to being mesh resolved for the finite-volume solvers and again the DG solver compares well. The DG solver in both of these cases converges to the mesh resolved answers with fewer degrees of freedom using a higher order of accuracy, which is typical of high-order methods.

In terms of non-linear convergence the DG solver performs well for the hemisphere case. For all of the grids and polynomial degree $p = 1$, it takes around 40 non-linear iterations to solve to machine precision. This is due to the Newton-Krylov solver with line preconditioning and the CFL controller [19]. When the lower order solutions are used to initialize the higher order solutions even fewer iterations are required.



(a) Station 1, $\alpha = 7°$ (b) Station 6, $\alpha = 7°$ (c) Station 10, $\alpha = 7°$



(d) Station 1, $\alpha = 16°$ (e) Station 6, $\alpha = 16°$ (f) Station 10, $\alpha = 16°$

Figure 13: Computed surface pressure coefficients C_P for the DLR-F11 compared to NSU3D results on the finest grid and experimental data.

The bump performs almost as well the hemisphere but convergence is more challenging on the finer grids. Relaxing the tolerances on the linear solve helps with achieving Newton quadratic convergence in this case.

The simulation of the DLR-F11 compares well to NSU3D results for both angles of attack $\alpha = 7^\circ$ and $\alpha = 16^\circ$. Only one $p = 1$ simulation was performed at each angle of attack, so not many conclusions can be drawn from these comparisons. The drag, lift, and moment coefficients compare well with NSU3D. Also, C_P line plots at stations 1 and 6 at both angles of attack compare well with NSU3D. Separation was observed at station 10, $\alpha = 7^\circ$, in the DG solution that is not observed in NSU3D or the experiments. However, separation at station 10, $\alpha = 16^\circ$, is observed again in the DG solution but also occurs in a mesh adapted solution using FUN3D. This needs to be further investigated along with running finer grids and higher orders of accuracy.

IX. Acknowledgments

Computer time was provided by the NCAR-Wyoming Supercomputing Center (NWSC) and University of Wyoming Advanced Research Computing Center (ARCC).

References

[1]Rumsey, C., "Turbulence Modeling Resource," turbmodels.larc.nasa.gov, August 2014.

[2]Fidkowski, K. J. and Darmofal, D. L., "An adaptive simplex cut-cell method for discontinuous Galerkin discretizations of the Navier-Stokes equations," *AIAA Paper*, Vol. 3941, 2007, pp. 2007.

[3]Wang, L. and Mavriplis, D. J., "Adjoint-based h–p adaptive discontinuous Galerkin methods for the 2D compressible Euler equations," *Journal of Computational Physics*, Vol. 228, No. 20, 2009, pp. 7643–7661.

[4]Hartmann, R. and Houston, P., "Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations," *Journal of Computational Physics*, Vol. 183, No. 2, 2002, pp. 508–532.

[5]Ahrabi, B. R., Anderson, W. K., and Newman, J. C., *High-Order Finite-Element Method and Dynamic Adaptation for Two-Dimensional Laminar and Turbulent Navier-Stokes*, AIAA Paper 2014-2983, 32nd AIAA Applied Aerodynamics Conference, Atlanta GA, June 2014.

[6]Ahrabi, B. R., Anderson, W. K., and Newman, J. C., *An Adjoint-Based hp-Adaptive Petrov-Galerkin Method for Turbulent Flows*, AIAA Paper 2015-2603, 22nd AIAA Computational Fluid Dynamics Conference, Dallas, TX, June 2015.

[7]Brazell, M. J. and Mavriplis, D. J., *3D Mixed Element Discontinuous Galerkin with Shock Capturing*, AIAA Paper 2013-2855, 21st AIAA CFD Conference, San Diego, CA, June 2013.

[8]Diskin, B., Thomas, J., Rumsey, C. L., and Schwoeppe, A., *Grid Convergence for Turbulent Flows (Invited)*, AIAA Paper 2015-1746, 53rd AIAA Aerospace Sciences Meeting, Kissimmee, FL, January 2015.

[9]Brazell, M. J. and Mavriplis, D. J., *High-Order Discontinuous Galerkin Mesh Resolved Turbulent Flow Simulations of a NACA 0012 Airfoil (Invited)*, AIAA Paper 2015-1529, 53rd AIAA Aerospace Sciences Meeting, Kissimmee, FL, January 2015.

[10]Ceze, M. and Fidkowski, K., *High-Order Output-Based Adaptive Simulations of Turbulent Flow in Two Dimensions (Invited)*, AIAA Paper 2015-1532, 53rd AIAA Aerospace Sciences Meeting, Kissimmee, FL, January 2015.

[11]Hu, Y., Wagner, C., Allmaras, S., Galbraith, M., and Darmofal, D. L., *Application of a Higher-order Adaptive Method to RANS Test Cases (Invited)*, AIAA Paper 2015-1530, 53rd AIAA Aerospace Sciences Meeting, Kissimmee, FL, January 2015.

[12]Anderson, W. K., Ahrabi, B. R., and Newman, J., *Finite-Element Solutions for Turbulent Flow over the NACA 0012 Airfoil (Invited)*, AIAA Paper 2015-1531, 53rd AIAA Aerospace Sciences Meeting, Kissimmee, FL, January 2015.

[13]Rumsey, C. L., Biedron, R. T., and Thomas, J. L., *CFL3D, Its History and Some Recent Applications*, Vol. 112861, National Aeronautics and Space Administration, Langley Research Center, 1997.

[14]Anderson, W. K. and Bonhaus, D. L., "An implicit upwind algorithm for computing turbulent flows on unstructured grids," *Computers & Fluids*, Vol. 23, No. 1, 1994, pp. 1–21.

[15]Frink, N. T., "Tetrahedral Unstructured Navier-Stokes Method for Turbulent Flows," *AIAA Journal*, Vol. 36, No. 11, 2015/12/08 1998, pp. 1975–1982.

[16]Rumsey, C., "2nd AIAA CFD High Lift Prediction Workshop," http://hiliftpw.larc.nasa.gov/index-workshop2.html, May 2015.

[17]Spalart, P. and Allmaras, S., *A one-equation turbulence model for aerodynamic flows*, Vol. 1, Le Recherche Aerospátiale, 1994, pp. 5–21.

[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]Ceze, M. A. and Fidkowski, K. J., "Constrained pseudo-transient continuation," *International Journal for Numerical Methods in Engineering*, Vol. 102, 2015, pp. 1683–1703.

[20]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.

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

[22]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.

[23]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.

[24]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.

[25]Burgess, N. K. and Mavriplis, D. J., "An hp-adaptive Discontinuous Galerkin solver for aerodynamic flows on mixed-element meshes," *AIAA Paper 2011-490, 49th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, January 2011.

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

[27]Hsieh, T., "An Investigation of Separated Flow about a Hemisphere-Cylinder at 0- to 19-Deg Incidence in the Mach Number Range from 0.6 to 1.5," Tech. rep., AEDC-TR-76-112, 1974.

[28]Rudnik, R., Huber, K., and Melber-Wilkending, S., *EUROLIFT Test Case Description for the 2nd High Lift Prediction Workshop*, AIAA Paper 2012-2924, 30th AIAA Applied Aerodynamics Conference, June 2012.

[29]Mavriplis, D., Long, M., Lake, T., and Langlois, M., "NSU3D Results for the Second AIAA High-Lift Prediction Workshop," *Journal of Aircraft*, Vol. 52, No. 4, 2015/11/22 2015, pp. 1063–1081.

[30]"OpenCascade - an open source library for BRep solid modeling," http://www.opencascade.com, 2000.

[31]Lee-Rausch, E. M., Rumsey, C. L., and Park, M. A., "Grid-Adapted FUN3D Computations for the Second High-Lift Prediction Workshop," *Journal of Aircraft*, Vol. 52, No. 4, 2015/11/29 2015, pp. 1098–1111.

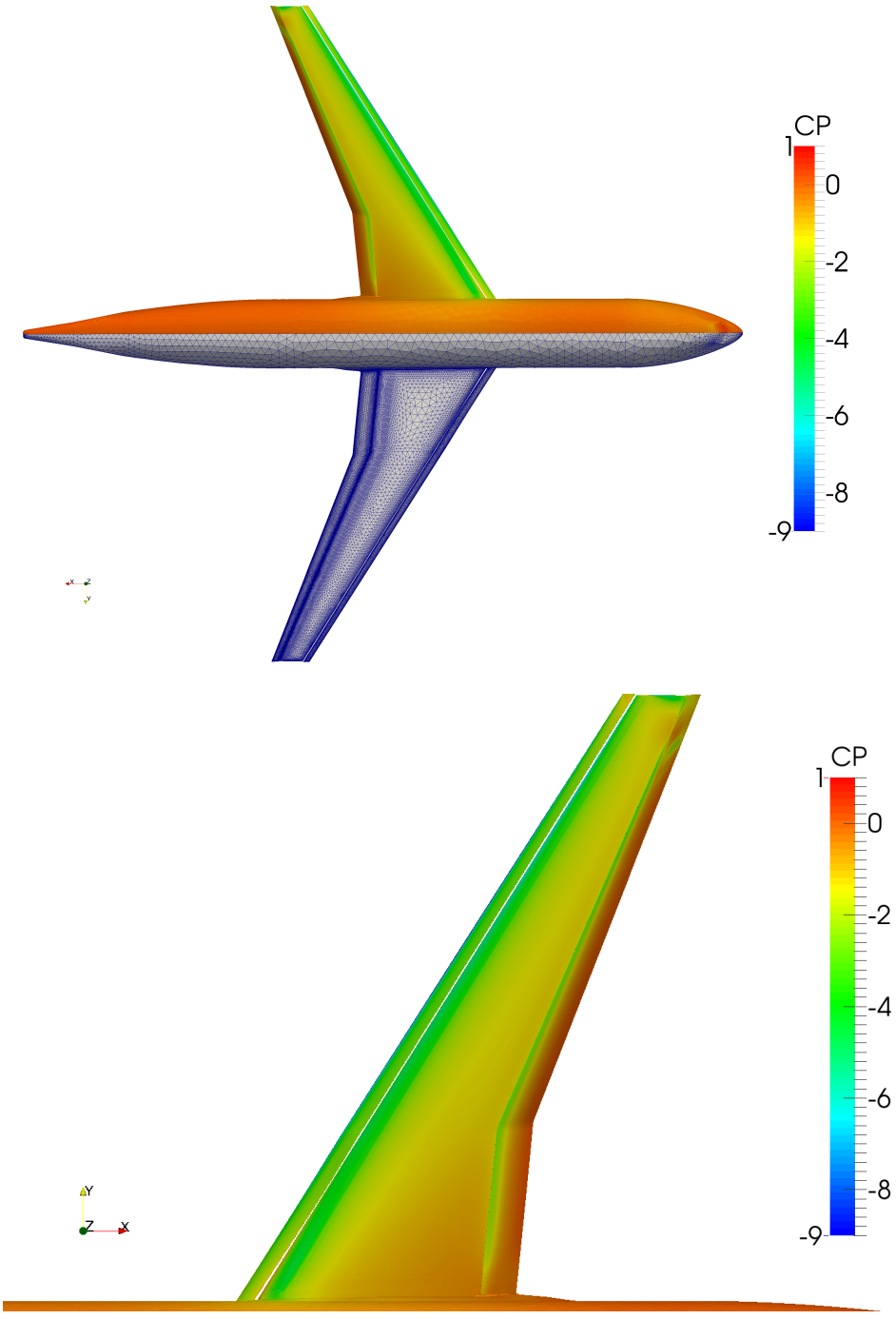


Figure 14: Contours of computed surface pressure coefficients C_P for the DLR-F11 at $\alpha = 16°$ and mesh.