



An Implicit Gradient Method for Cell-Centered Finite-Volume Solver on Unstructured Grids

Hiroaki Nishikawa*

National Institute of Aerospace, Hampton, VA 23666, USA

In this paper, we investigate an implicit gradient method for a second-order cell-centered finite-volume method on unstructured grids. In the implicit gradient method, solution gradients are obtained by solving a global system of linear equations, but they can be computed iteratively along with an implicit finite-volume solver iteration for the Euler or Navier-Stokes equations. The cost of the implicit gradient computation is thereby made comparable to or even cheaper than that of explicit gradient methods such as a least-squares method. Furthermore, a known stability issue with a face-neighbor gradient stencil is effectively circumvented by expanding the gradient stencil to the entire domain. We discuss the relative performance of the implicit gradient method and a least-squares method for various inviscid and viscous flow computations on unstructured grids.

1. Introduction

Convergence and accuracy of implicit finite-volume solvers widely used in practical computational fluid dynamics (CFD) codes are greatly affected by gradient computation methods, especially on unstructured grids. Despite great progress made over the last decades, represented by an extensive study on least-squares gradients in Ref.[1], current state-of-the-art unstructured-grid finite-volume solvers still encounter robustness issues when dealing with highly-distorted grids as typical in practical applications involving complex geometries. In practical solvers, inconsistent gradient methods such as the Green-Gauss method are locally employed for robustness [2,3,4], or simply gradients are ignored in problematic regions (e.g., in a cell having a large face angle). Also, stable computations often require different types of gradient methods for different terms, e.g., unweighted least-squares (LSQ) gradients for the inviscid terms, and weighted LSQ gradients for the viscous terms and the source terms [2,5]. Another known problem is that finite-volume schemes with LSQ gradients computed over face neighbors are unstable for triangular and tetrahedral grids [6,7]. Development of gradient methods for increasing robustness and efficiency without degrading accuracy, therefore, remains an important subject of research [8,9,10].

Recently, in a series of papers [11,12,13], a new approach called the variational reconstruction (VR) method was proposed. In this method, a globally coupled system of linear equations for the gradients is derived by minimizing the solution jumps at faces. The system is iteratively solved along with a finite-volume flow solver. If only one iteration is performed for the gradients per solver iteration, then the cost per iteration is almost the same as the explicit methods such as the GG and LSQ methods, or more efficient than LSQ methods with extended stencils beyond face neighbors. The stability issue mentioned earlier is effectively circumvented by the gradient stencil spanning the entire grid. A further advantage lies in the straightforward extension to high-order through minimizing the jumps of high-order polynomials [11,12,13,14,15]. Inspired by the VR method, we developed a similar implicit gradient methodology called the implicit Green-Gauss (IGG) gradient method [16], which was derived from a second-order finite-volume discretization of a hyperbolic diffusion model. In Ref.[16], the IGG gradient method was demonstrated, for model advection-diffusion problems, to produce accurate gradients on highly distorted grids including highly-curved thin grids, and also allow implicit finite-volume solvers to converge for a discontinuous solution. In this paper, we investigate the IGG gradient method for inviscid and viscous flow problems.

*Associate Research Fellow (hiro@nianet.org), 100 Exploration Way, Hampton, VA 23666 USA, Associate Fellow AIAA

2. Governing Equations

Consider the compressible Navier-Stokes (NS) equations in two dimensions:

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}_x}{\partial x} + \frac{\partial \mathbf{f}_y}{\partial y} = 0, \quad (2.1)$$

where

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}, \quad \mathbf{f}_x = \mathbf{f}_x^{inv} + \mathbf{f}_x^{vis} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho u H \end{bmatrix} + \begin{bmatrix} 0 \\ -\tau_{xx} \\ -\tau_{xy} \\ -\tau_{xx}u - \tau_{xy}v + q_x \end{bmatrix}, \quad (2.2)$$

$$\mathbf{f}_y = \mathbf{f}_y^{inv} + \mathbf{f}_y^{vis} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho v H \end{bmatrix} + \begin{bmatrix} 0 \\ -\tau_{yx} \\ -\tau_{yy} \\ -\tau_{yx}u - \tau_{yy}v + q_y \end{bmatrix}, \quad (2.3)$$

ρ is the density, (u, v) is the velocity vector, p is the pressure, E is the specific total energy, and $H = E + p/\rho$ is the specific total enthalpy. The viscous stresses $(\tau_{xx}, \tau_{xy}, \tau_{yy})$, and the heat fluxes q_x and q_y are given by

$$\tau_{xx} = \frac{2}{3}\mu \left(2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \right), \quad \tau_{xy} = \tau_{yx} = \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right), \quad \tau_{yy} = \frac{2}{3}\mu \left(2\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x} \right), \quad (2.4)$$

$$q_x = -\frac{\mu}{Pr(\gamma-1)} \frac{\partial T}{\partial x}, \quad q_y = -\frac{\mu}{Pr(\gamma-1)} \frac{\partial T}{\partial y}. \quad (2.5)$$

where T is the temperature, γ is the ratio of specific heats, Pr is the Prandtl number, and μ is the viscosity defined by Sutherland's law. All the quantities are assumed to have been nondimensionalized by their free-stream values except that the velocity and the pressure are scaled by the free-stream speed of sound and the free-stream dynamic pressure, respectively (see Ref.[17]). Thus, the viscosity is given by the following form of Sutherland's law:

$$\mu = \frac{M_\infty}{Re_\infty} \frac{1 + C/\tilde{T}_\infty}{T + C/\tilde{T}_\infty} T^{\frac{3}{2}}, \quad (2.6)$$

where \tilde{T}_∞ is the dimensional free stream temperature in Kelvin (K), and $C = 110.5$ [K] is the Sutherland constant. The ratio of the free stream Mach number M_∞ to the free stream Reynolds number Re_∞ arises from the nondimensionalization. The system is closed by the nondimensionalized equation of state for ideal gases: $\gamma p = \rho T$.

3. Finite-Volume Discretization

A cell-centered finite-volume discretization is employed for the discretization, where the residual is defined as an approximation to the NS system integrated over a computational cell j (see Figure 1) with the midpoint rule:

$$\mathbf{Res}_j = \sum_{k \in \{k_j\}} \Phi_{jk} A_{jk}, \quad (3.1)$$

where $\{k_j\}$ is a set of neighbors of the cell j , A_{jk} is the length of the face across j and k , and Φ_{jk} is a numerical flux.

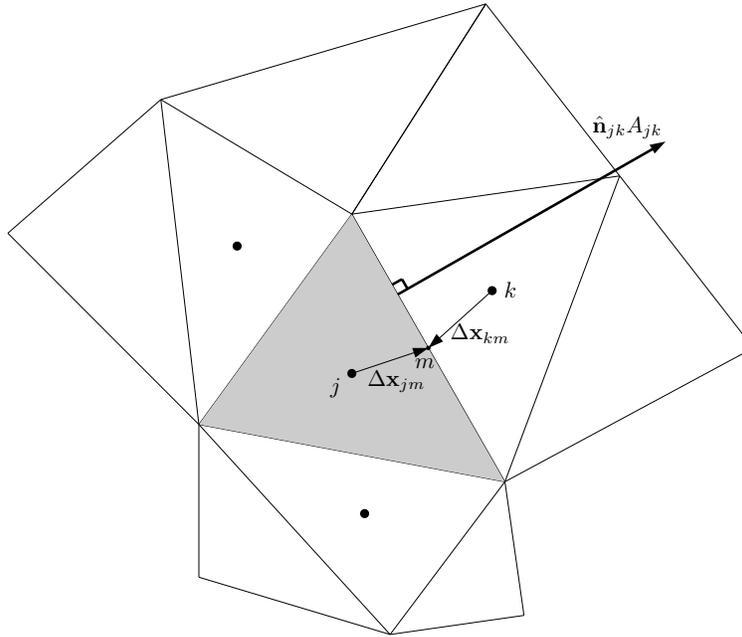


Figure 1: Stencil for cell-centered finite-volume discretization.

In this work, the Roe [18] and HLL [19] fluxes are used for the inviscid terms, and the alpha-damping flux [20, 21, 22, 23] for the viscous terms. These numerical fluxes are functions of the primitive-variable gradients, $\nabla \mathbf{W}_j$ and $\nabla \mathbf{W}_k$, and the primitive variables $\mathbf{W} = (\rho, u, v, p)$ linearly extrapolated at the face midpoint from the two adjacent cells j and k :

$$\mathbf{W}_L = \mathbf{W}_j + \phi_j \nabla \mathbf{W}_j \cdot \Delta \mathbf{x}_{jm}, \quad \mathbf{W}_R = \mathbf{W}_k + \phi_k \nabla \mathbf{W}_k \cdot \Delta \mathbf{x}_{km}, \quad (3.2)$$

where $\Delta \mathbf{x}_{jm} = \mathbf{x}_m - \mathbf{x}_j$, $\Delta \mathbf{x}_{km} = \mathbf{x}_m - \mathbf{x}_k$, \mathbf{x}_m is the face-midpoint position, \mathbf{x}_j and \mathbf{x}_k are the centroid coordinates of the cells j and k , respectively, and ϕ_j and ϕ_k are the Venkat limiter functions computed based on the enforcement at nodes as described in Refs.[24, 25]. The limiter function is defined by the minimum, as typically done, of those computed for all primitive variables. The cell gradients, $\nabla \mathbf{W}_j$ and $\nabla \mathbf{W}_k$, need to be computed from the numerical solutions stored at cells. These gradients are typically computed by a LSQ method or the Green-Gauss method. In this study, the IGG gradient method [16] is employed and its performance is investigated and compared with a LSQ gradient method. For the viscous flux, these gradients will be used to evaluate the consistent term of the alpha-damping viscous flux: for example, the x -velocity gradient is given by

$$\nabla u|_{face} = \frac{1}{2} [\nabla u_j + \nabla u_k] + \frac{\alpha}{|\mathbf{e}_{jk} \cdot \hat{\mathbf{n}}_{jk}|} (u_R - u_L) \hat{\mathbf{n}}_{jk}, \quad \mathbf{e}_{jk} = \mathbf{x}_k - \mathbf{x}_j, \quad (3.3)$$

where the first term is the consistent term (i.e., consistently approximating the gradient), the second term is the damping term (i.e., responsible for damping high-frequency errors) with the damping coefficient α given an optimal value: $\alpha = 4/3$ (see Refs.[20, 22] for details), and $\hat{\mathbf{n}}_{jk} = (\hat{n}_x, \hat{n}_y)$ is the unit vector normal to the face pointing from j to k .

4. Implicit Green-Gauss Gradient (IGG) Method

The IGG gradient method [16] is an implicit method that determines the cell-gradients ($\partial_x w_j, \partial_y w_j$) for a given set of numerical solution values, $\{w_j\} = \{\rho_j\}, \{u_j\}, \{v_j\}$, or $\{p_j\}$, as a solution to a global linear system, where the equations corresponding to a cell j is given by

$$\mathbf{M}_{jj} \mathbf{g}_j + \sum_{k \in \{k_j\}} \mathbf{M}_{jk} \mathbf{g}_k = \mathbf{b}_j, \quad (4.1)$$

where $\{k_j\}$ is a set of face-connected neighbors of the cell j , the 2×2 matrices \mathbf{M}_{jj} and \mathbf{M}_{jk} are given by

$$\mathbf{M}_{jj} = \frac{1}{2V_j} \sum_{k \in \{k_j\}} \left\{ - \begin{bmatrix} \Delta x_{jm} \hat{n}_x & \Delta y_{jm} \hat{n}_x \\ \Delta x_{jm} \hat{n}_y & \Delta y_{jm} \hat{n}_y \end{bmatrix} + L_r \begin{bmatrix} \hat{n}_x^2 & \hat{n}_x \hat{n}_y \\ \hat{n}_x \hat{n}_y & \hat{n}_y^2 \end{bmatrix} + c_j \begin{bmatrix} \Delta \mathbf{x}_{jm} \cdot \hat{\mathbf{n}}_{jk} & 0 \\ 0 & \Delta \mathbf{x}_{jm} \cdot \hat{\mathbf{n}}_{jk} \end{bmatrix} \right\} A_{jk}, \quad (4.2)$$

$$\mathbf{M}_{jk} = \frac{1}{2V_j} \left\{ - \begin{bmatrix} \Delta x_{km} \hat{n}_x & \Delta y_{km} \hat{n}_x \\ \Delta x_{km} \hat{n}_y & \Delta y_{km} \hat{n}_y \end{bmatrix} - L_r \begin{bmatrix} \hat{n}_x^2 & \hat{n}_x \hat{n}_y \\ \hat{n}_x \hat{n}_y & \hat{n}_y^2 \end{bmatrix} + (1 - c_j) \begin{bmatrix} \Delta \mathbf{x}_{jm} \cdot \hat{\mathbf{n}}_{jk} & 0 \\ 0 & \Delta \mathbf{x}_{jm} \cdot \hat{\mathbf{n}}_{jk} \end{bmatrix} \right\} A_{jk}, \quad (4.3)$$

and

$$\mathbf{g}_j = \begin{bmatrix} \partial_x w_j \\ \partial_y w_j \end{bmatrix}, \quad \mathbf{g}_k = \begin{bmatrix} \partial_x w_k \\ \partial_y w_k \end{bmatrix}, \quad \mathbf{b}_j = \frac{1}{2V_j} \sum_{k \in \{k_j\}} \begin{bmatrix} (w_j + w_k) \hat{n}_x \\ (w_j + w_k) \hat{n}_y \end{bmatrix} A_{jk}, \quad (4.4)$$

where $\Delta \mathbf{x}_{jm} = (x_{jm}, y_{jm})$, $\Delta \mathbf{x}_{km} = (y_{km}, y_{km})$, V_j is the area of the cell j . Note that the right hand side \mathbf{b}_j is the Green-Gauss gradient over the cell j , and the gradients are coupled with neighbors on the left hand side; thus, it is called the implicit Green-Gauss gradient method. As discussed in Ref.[16], the IGG method produces first-order accurate gradients on arbitrary unstructured grids whereas the Green-Gauss method loses accuracy, as is well known, on irregular grids. The parameter c_j is defined in the cell j and changes its value depending on the grid skewness as described in Ref.[16]. The length L_r is defined based also on the skewness:

$$L_r = \alpha_g |\mathbf{e}_{jk} \cdot \hat{\mathbf{n}}_{jk}|, \quad \mathbf{e}_{jk} = \mathbf{x}_k - \mathbf{x}_j, \quad (4.5)$$

where α_g is a constant. As discussed in Ref.[16], the IGG gradient method can be considered as an extension of the compact finite-difference methods to unstructured grids; the parameter α_g controls the resolution in much the same way as in the compact finite-difference schemes. A large value of α_g has the effect of smoothing the gradient at a discontinuous solution and allows an implicit finite-volume solver to converge without a limiter. On the other hand, decreasing α_g has the effect of picking up more high-frequency modes, and eventually achieves fourth-order accuracy on a rectangular grid at $\alpha_g = 1/6$ through a domain boundary. The IGG method is, therefore, a convenient way of implementing a compact scheme without any special treatment at domain boundaries. See Ref.[16] for further details.

5. Implicit Defect-Correction (IDC) and IGG Solver

The set of discrete equations to be solved are the residual equation for the NS equations:

$$0 = \mathbf{Res}(\mathbf{U}), \quad (5.1)$$

where \mathbf{U} denotes the global vector of the conservative variables, and the residual equation for the gradients of each primitive variable, i.e., a global expression of Equation (4.1):

$$0 = \mathbf{Res}^g(\mathbf{g}), \quad (5.2)$$

where \mathbf{g} denotes the global vector of the gradients. Note that there are four gradient residual equations for $w = \rho, u, v,$ and p . There can be various different strategies for solving these equations. In this paper, we consider a loosely-coupled implicit defect-correction (IDC) solver.

The IDC solver for the NS residual is given by

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \Delta \mathbf{U}, \quad (5.3)$$

$$\left(\mathbf{D} + \frac{\partial \overline{\mathbf{Res}}}{\partial \mathbf{U}} \right) \Delta \mathbf{U} = -\mathbf{Res}(\mathbf{U}^n), \quad (5.4)$$

where n is the iteration counter, \mathbf{D} is the pseudo-time diagonal matrix with the j -th diagonal block, which corresponds to the cell j , is defined by $\mathbf{D}_{jj} = V_j / \Delta \tau_j \mathbf{I}$, where \mathbf{I} is the 4×4 identity matrix and $\Delta \tau_j$ is a local

pseudo time step, and the Jacobian $\partial \overline{\mathbf{Res}} / \partial \mathbf{U}$ is the exact differentiation of the low-order compact residual $\overline{\mathbf{Res}}$ with zero gradients (i.e., first-order accurate for the inviscid terms and zero-th order for the viscous terms). The linear system is relaxed by the block Gauss-Seidel relaxation scheme, which is written for a cell j as

$$\Delta \mathbf{U}_j^{m+1} = \Delta \mathbf{U}_j^m + \omega \mathbf{r}_j, \quad \mathbf{r}_j = \left(\frac{\partial \overline{\mathbf{Res}}_j}{\partial \mathbf{U}_j} \right)^{-1} \left[\sum_{k \in \{k_j\}} \frac{\partial \overline{\mathbf{Res}}_j}{\partial \mathbf{U}_k} - \mathbf{Res}_j(\mathbf{U}^n) \right], \quad (5.5)$$

where m is the relaxation counter, ω is a relaxation factor, and $\frac{\partial \overline{\mathbf{Res}}_j}{\partial \mathbf{U}_j}$ and $\frac{\partial \overline{\mathbf{Res}}_j}{\partial \mathbf{U}_k}$ are the 4×4 diagonal and off-diagonal matrices, respectively.

The gradient residual equation is relaxed for each primitive variable, $\mathbf{g}_j = (\partial_x w_j, \partial_y w_j)$, where $w_j = \rho_j, u_j, v_j$, or p_j , by directly applying the GS relaxation scheme [16]:

$$\mathbf{g}_j^{n+1} = \mathbf{g}_j^n + \omega_g \left[\sum_{k \in \{k_j\}} \{-\mathbf{C}_{jk} \mathbf{g}_k^n + \mathbf{c}_{jk}(w_j + w_k)\} - \mathbf{g}_j^n \right], \quad (5.6)$$

where ω_g is a relaxation factor, and

$$\mathbf{C}_{jk} = \mathbf{M}_{jj}^{-1} \mathbf{M}_{jk}, \quad \mathbf{c}_{jk} = \mathbf{M}_{jj}^{-1} \frac{1}{2V_j} \hat{\mathbf{n}}_{jk} A_{jk}. \quad (5.7)$$

It is pointed out that there is a typo in the GS relaxation scheme in Ref.[16]; the above is the correct form. Note that the same iteration counter n is used in both iterations, meaning that the gradient relaxation is performed once per IDC iteration. As discussed in Ref.[16], this iteration strategy renders the IGG gradient method comparable or more economical than the IDC solver with the LSQ gradient method (with augmented stencils beyond face neighbors) in terms of both computing time and storage, and in both two and three dimensions. This strategy has successfully been employed in other works related to the VR gradient method [11, 12, 13, 14, 15]. In the rest of the paper, the IDC solver with the IGG Gauss-Seidel relaxation is referred to as IDC-IGG while the IDC solver with the LSQ gradient method is referred to as IDC-LSQ.

The IDC-IGG solver is controlled by the following parameters:

| | | |
|---------------------------------|---|--|
| <code>tolerance_u</code> | = | Tolerance for the flow-equation residuals. |
| <code>tolerance_g</code> | = | Tolerance for the gradient residuals. |
| <code>max_relax_linear_u</code> | = | Maximum number of relaxations for the flow-equation GS relaxation. |
| <code>max_relax_linear_g</code> | = | Maximum number of relaxations for the gradient GS relaxation. |
| <code>tolerance_linear_u</code> | = | Tolerance for the flow-equation GS relaxation. |
| <code>tolerance_linear_g</code> | = | Tolerance for the gradient GS relaxation. |
| <code>omega</code> | = | Relaxation factor for the flow-equation GS relaxation. |
| <code>omega_g</code> | = | Relaxation factor for the gradient GS relaxation. |

The first two determines convergence by the order of magnitude of the residual reduction achieved in the L_1 norm for all the Euler/NS residuals and the gradient residuals. For problems considered in this paper, we choose `tolerance_u` = 1.0e-05 and `tolerance_g` = 1.0e-03 since it was observed that any further residual reduction in the gradient residuals did not bring a noticeable change in the solution. For the linear relaxation, we typically require one order of magnitude reduction; the maximum numbers of relaxations depend on the grid size and are given sufficiently larger values to meet the tolerance in all cases considered in this paper. The relaxation factor parameter `omega` sets an initial value; it is adaptively modified based on the history of the linear residual norm $\|\mathbf{r}\|_2$ during the relaxation: increased by 25 % if the norm is below the initial value while decreased by 5% if above, with the minimum and maximum values of 0.05 and 1.0, respectively. For the gradient relaxation, the relaxation factor `omega_g` is fixed as 1.0, 0.75, or 0.5; we will discuss later.

We remark that the IDC-IGG solver as described above can be considered as a simplified version of a fully-coupled solver. If we add all the gradients to the solution vector \mathbf{U} and add the gradient residuals to the residual equation (5.1), then we can construct Newton's method:

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \Delta \mathbf{U}, \quad (5.8)$$

applied to the gradients. The pseudo time step is defined by

$$\Delta\tau_j = \text{CFL} \frac{2V_j}{\sum_{k \in \{k_j\}} [|u_n| + a + a_h] A_{jk}}, \quad (6.1)$$

where u_n is the flow speed in the direction normal to the face, a is the speed of sound at the face, and a_h is a characteristic speed associated with the viscous heating wave (see Ref.[21]):

$$a_h = \frac{2\alpha\gamma\mu}{\rho P_r |\mathbf{e}_{jk} \cdot \hat{\mathbf{n}}_{jk}|}. \quad (6.2)$$

All fluid properties are set based on a standard air: $\gamma = 1.4$, $P_r = 0.72$, at the free stream temperature $\tilde{T}_\infty = 300[K]$. Boundary conditions are imposed weakly through a numerical flux with the right state specified by physical boundary conditions.

6.1. Euler Equations

6.1.1. Subsonic flow over a circular cylinder at $M_\infty = 0.3$

We consider an inviscid compressible flow over a circular cylinder at $M_\infty = 0.3$. The domain is defined by the cylinder of unit diameter and the circular outer boundary of radius 100. The problem is solved on a triangular grid shown in Figures 2(a) and 2(b): 12800 nodes, 25280 triangles, 161 nodes over the cylinder, and the minimum and maximum skewness measures of 0.720 and 1.000 respectively. For the IGG gradient method, $\alpha_g = 1.0$ is used. In all calculations, we set $\text{CFL} = 10^{15}$. Both the IDC-LSQ and IDC-IGG solvers converged as shown in Figure 2(e), and similar solutions are obtained as in Figures 2(c) and 2(d). For this problem, the solver converged with a fewer number of iterations with the IGG gradients than with the LSQ gradients. As can be expected, it leads to faster convergence in CPU time as shown in Figure 2(f).

6.1.2. Transonic flow over a Joukowsky airfoil at $M_\infty = 0.8$

The next inviscid test case is a transonic flow over a Joukowsky airfoil of a unit chord at $M_\infty = 0.8$ at the angle of attack 1.25 degrees. The outer boundary is taken as a circle of radius 50. The grid is an irregular triangular grid as shown in Figures 3(a) and 3(b): 4235 nodes, 8228 triangles, 122 nodes over the airfoil, the minimum and maximum skewness measures of 0.0273 and 0.9999 respectively. In all calculations, we set $\text{CFL} = 10$. The LSQ method was applied with the Venkat limiter [24] as described in Section 3. The IGG method was applied with $\alpha_g = 4.0$ to smooth the gradients across shocks and with and without the Venkat limiter. Otherwise, the same parameter setting was used as in the previous case. The limiter was applied with the parameter $K = 5.0$ for the LSQ gradients and $K = 10.0$ (less limiting) for the IGG gradients [24].

Convergence histories are shown in Figures 4(a) and 4(b); the IDC-LSQ solver stalls (not shown but diverged without a limiter), while the IDC-IGG solver converges with and without the limiter (denoted by IGG-Limited and IGG, respectively, in the figure). Note that the IDC-LSQ solver converges on a structured quadrilateral/triangular grid without a limiter; the divergence without a limiter is therefore considered due to the irregularity. The limited version of the IDC-IGG solver is slightly slower in CPU time apparently due to the limiter calculation. Solutions are compared in Figure 5. As expected, the IGG solution without a limiter has over- and under-shoots around the shock on the upper surface (see Figure 5(e)); and they are greatly reduced by the limiter as can be seen in Figure 5(f). Contours of the limiter function are plotted in Figures 5(g)-5(i). It is observed that the limiter acts much less on the IGG gradients than on the LSQ gradients, thus preserving accuracy better with the IGG gradients.

6.1.3. Hypersonic flow over a cylinder at $M_\infty = 5.2$

To further demonstrate the robustness of the IDC-IGG solver, we consider a hypersonic flow over a cylinder at $M_\infty = 5.2$ on a triangular grid as shown in Figures 6(a) and 6(b): 1089 nodes, 2048 triangles, 32 nodes over the cylinder surface, the minimum and maximum skewness measures of 0.3037 and 1.000 respectively. The LSQ method was applied with the Venkat limiter ($K = 0.1$) [24]. The IGG method was applied with $\alpha_g = 3.5$ to smooth the gradients across the bow shock with the Venkat limiter ($K = 5.0$). For this problem, the CFL number is initially set to be 0.1 and linearly increased in terms of the iteration

number towards 10 over 50000 iterations. The linear relaxation tolerance for the Euler equations is set to be 0.001: `tolerance_linear_u` = 0.001. The HLL flux is used for the inviscid flux instead of the Roe flux.

As shown in Figures 6(c) and 6(d), the IDC-IGG solver converges whereas the IDC-LSQ solver gradually diverges up to the maximum iteration of 25000. Despite the gradual divergence, the solution looks reasonably accurate and very similar to the converged solution obtained by the IDC-IGG solver. See Figures 6(e) and 6(f); the corresponding limiter function distributions are shown in Figures 6(g) and 6(h).

6.2. Navier-Stokes Equations

6.2.1. Viscous flow over a circular cylinder at $M_\infty = 0.3$ and $Re_\infty = 40$

The first viscous test case is a viscous flow over a circular cylinder at $M_\infty = 0.3$ and $Re_\infty = 40$ with the same grid used in the inviscid cylinder test case. The IGG method was applied with $\alpha_g = 1/6$. It was found that the IDC-IGG solver diverged unless the relaxation factor `omega_g` is reduced to 0.75. Results shown are obtained with `omega_g=0.75`. The CFL number is 10^{15} for all cases.

Convergence results are shown in Figures 7(a) and 7(b). For this problem, the IDC-IGG solver is twice as fast as the IDC-LSQ solver in both iteration and CPU time, producing nearly identical solutions as in Figures 7(c)-7(f).

6.2.2. Viscous flow over a flat plate at $M_\infty = 0.15$ and $Re_\infty = 10^6$

Finally, we consider a high-Reynolds-number laminar flow over a flat plate at $M_\infty = 0.15$ and $Re_\infty = 10^6$. The grid is a mixed-element grid with 3264 nodes, 3638 triangles, 1330 quadrilaterals (which covers the boundary layer), 30 nodes over the flat plate located at $y = 0$ and $x \in [0, 2]$, and the minimum and maximum skewness measures of 0.0049 and 1.000 respectively. See Figures 8(a) and 8(b); the latter shows that the interface between triangular and quadrilateral regions is not smooth. The IGG method was applied with $\alpha_g = 1/6$. For this problem, the IDC-IGG solver diverged even with `omega_g=0.75`, and thus further reduced to `omega_g=0.5`.

Convergence results are shown in Figures 8(c) and 8(d). In this case, the IDC-IGG solver takes a larger number of iterations, and therefore a longer CPU time to converge. However, the solution obtained with the IGG gradient method is more accurate (closer to the Blasius solution) than that with the LSQ gradients. See Figures 8(e)-8(g), where the numerical solutions are compared with the Blasius solution at $x = 0.9$: the x -velocity u , the y -velocity $v\sqrt{Re_x}$, where Re_x is the Reynolds number based on the distance along the plate from the leading edge, and the vorticity in the normalized coordinate $\eta = y\sqrt{Re_x}/x$.

7. Concluding Remarks

The implicit Green-Gauss gradient method was tested and compared with the least-squares gradient method for a cell-centered finite-volume method on unstructured grids for inviscid and viscous problems. The gradients are computed by iteratively solving a global system of linear equations with the Gauss-Seidel relaxation scheme. An implicit defect-correction solver was tested, where the Gauss-Seidel relaxation is performed once for the gradient in each nonlinear iteration. Numerical results have confirmed that the per-iteration cost of the implicit Green-Gauss gradient method is comparable to the LSQ gradient computation. The implicit solver has been demonstrated to converge to a specified tolerance for all problems (subsonic, transonic, and hypersonic inviscid flows, and low- and high-Reynolds-number viscous flows) even with shock waves and limiters whereas it stalled with the least-squares gradients; and it converged faster in CPU time than the solver with the LSQ gradients in all cases except the flat plate case. The solution, however, was found to be more accurate with the implicit Green-Gauss gradient method.

Despite the demonstrated robustness, the solver needs further improvements. The need for the relaxation factor adjustment in the IGG relaxation seems to imply that the IDC-IGG solver can become unstable if the gradients gain accuracy too fast. Apparently, the implicit Euler/NS solver and the implicit Green-Gauss gradient relaxation are not always compatible with each other. A fully-coupled solver is one possible strategy to address the issue. There can be, at least, two variants. One is Newton's method applied to the entire set of residual equations, where the Jacobian is compact and exact but larger (i.e., 20×20 blocks in three dimensions). Another would be a Jacobian-Free Newton-Krylov (JFNK) method with the IDC-IGG solver used as a preconditioner. These solvers will be particularly useful for allowing parameters (e.g.,

α_g) to be defined locally to adapt the gradients to local flow features while keeping different values in the Jacobian/preconditioner for robust convergence. A preliminary study indicates that the Newton solver is much more robust than the IDC-IGG and JFNK solvers, but requires a very efficient linear solver (e.g., multigrid) to minimize the overall computing time; it seems suggest that the hyperbolic Navier-Stokes formulation is better suited since the linear relaxation converges rapidly by the reduced condition number (due to the elimination of second derivatives) [26, 27, 28, 29, 30, 31].

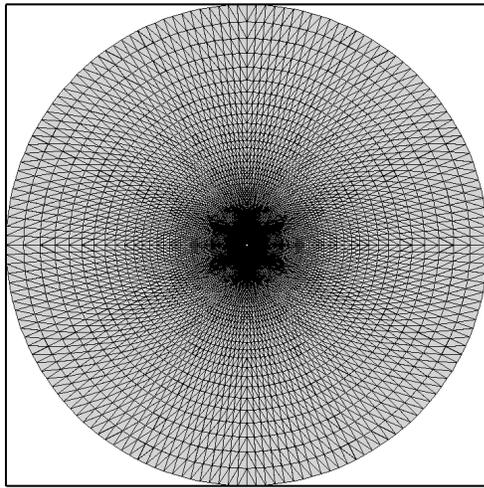
Acknowledgments

This work was supported by the Hypersonic Technology Project, through the Hypersonic Airbreathing Propulsion Branch of the NASA Langley Research Center, under Contract No. 80LARC17C0004. The author would like to thank Jeffery A. White (NASA Langley Research Center) for valuable comments and discussions.

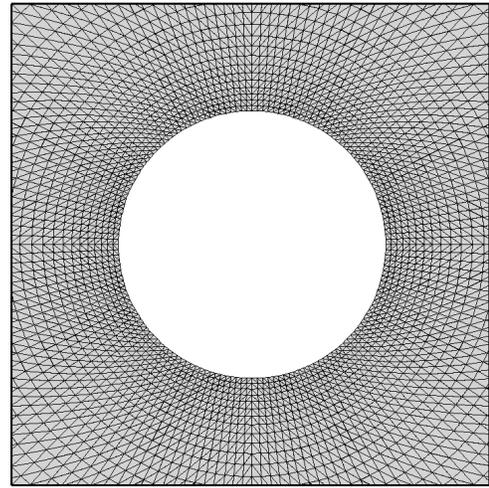
References

- ¹Diskin, B. and Thomas, J. L., “Accuracy of Gradient Reconstruction on Grids with High Aspect Ratio,” *NIA Report No. 2008-12*, 2008.
- ²Mavriplis, D. J., “Revisiting the Least-Squares Procedure for Gradient Reconstruction on Unstructured Meshes,” *Proc. of 16th AIAA Computational Fluid Dynamics Conference*, AIAA Paper 2003-3986, Orlando, Florida, 2003.
- ³Shima, E., Kitamura, K., and Haga, T., “Green-Gauss/Weighted-Least-Squares Hybrid Gradient Reconstruction for Arbitrary Polyhedra Unstructured Grids,” *AIAA J.*, Vol. 51, No. 11, 2013, pp. 2740–2747.
- ⁴Moukalled, F., Mangani, L., and Darwish, M., *The finite volume method in computational fluid dynamics: An introduction with OpenFOAM and matlab*, Fluid Mechanics and Its Applications, Volume 13, Springer International Publishing, 2015.
- ⁵May, G. and Jameson, A., “Unstructured algorithms for inviscid and viscous flows embedded in a unified solver architecture,” *Proc. of 43rd AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper 2005-318, Reno, Nevada, 2005.
- ⁶Diskin, B. and Thomas, J. L., “Comparison of Node-Centered and Cell-Centered Unstructured Finite-Volume Discretizations: Inviscid Fluxes,” *AIAA J.*, Vol. 49, No. 4, 2011, pp. 836–854.
- ⁷Haider, F., Croisille, J.-P., and Coubet, B., “Stability analysis of the cell centered finite-volume MUSCL method on unstructured grids,” *Numer. Math.*, Vol. 113, No. 4, 2009, pp. 555–600.
- ⁸White, J. A., Baurle, R., Passe, B. J., Spiegel, S. C., and Nishikawa, H., “Geometrically Flexible and Efficient Flow Analysis of High Speed Vehicles Via Domain Decomposition, Part 1, Unstructured-grid Solver for High Speed Flows,” *JANNAF 48th Combustion 36th Airbreathing Propulsion, 36th Exhaust Plume and Signatures, 30th Propulsion Systems Hazards, Joint Subcommittee Meeting, Programmatic and Industrial Base Meeting*, Newport News, VA, 2017.
- ⁹Zangeneh, R. and Ollivier-Gooch, C., “Reconstruction Map Stability Analysis for Cell Centered Finite Volume Methods on Unstructured Meshes,” *55th AIAA Aerospace Sciences Meeting*, AIAA Paper 2017-0734, Grapevine, Texas, 2017.
- ¹⁰Zhang, F., “A vertex-weighted-Least-Squares gradient reconstruction,” arXiv:1702.04518 [physics.flu-dyn], 2017.
- ¹¹Wang, Q., Ren, Y.-X., Pan, J., and Li, W., “Compact high order finite volume method on unstructured grids I: Basic formulations and one-dimensional schemes,” *J. Comput. Phys.*, Vol. 314, 2016, pp. 863–882.
- ¹²Wang, Q., Ren, Y.-X., Pan, J., and Li, W., “Compact high order finite volume method on unstructured grids II: Extension to two-dimensional Euler equations,” *J. Comput. Phys.*, Vol. 314, 2016, pp. 883–908.
- ¹³Wang, Q., Ren, Y.-X., Pan, J., and Li, W., “Compact high order finite volume method on unstructured grids III: Variational reconstruction,” *J. Comput. Phys.*, Vol. 337, 2017, pp. 1–26.
- ¹⁴Li, L., Liu, X., Lou, J., Luo, H., Nishikawa, H., and Ren, Y., “A Finite Volume Method Based on Variational Reconstruction for Compressible Flows on Arbitrary Grids,” *23rd AIAA Computational Fluid Dynamics Conference*, AIAA Paper 2017-3097, Denver, Colorado, 2017.
- ¹⁵Li, L., Liu, X., Lou, J., Luo, H., Nishikawa, H., and Ren, Y., “A Discontinuous Galerkin Method Based on Variational Reconstruction for Compressible Flows on Arbitrary Grids,” *56th AIAA Aerospace Sciences Meeting*, AIAA Paper 2018-0831, Kissimmee, Florida, 2018.
- ¹⁶Nishikawa, H., “From Hyperbolic Diffusion Scheme to Gradient Method: Implicit Green-Gauss Gradients for Unstructured Grids,” *J. Comput. Phys.*, Vol. 372, 2018, pp. 126–160.
- ¹⁷Masatsuka, K., “I do like CFD, VOL.1, Second Edition,” <http://www.cfdbooks.com>, 2013.
- ¹⁸Roe, P. L., “Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes,” *J. Comput. Phys.*, Vol. 43, 1981, pp. 357–372.
- ¹⁹Harten, A., Lax, P. D., and van Leer, B., “On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws,” *SIAM Rev.*, Vol. 25, No. 1, 1983, pp. 35–61.
- ²⁰Nishikawa, H., “Beyond Interface Gradient: A General Principle for Constructing Diffusion Schemes,” *Proc. of 40th AIAA Fluid Dynamics Conference and Exhibit*, AIAA Paper 2010-5093, Chicago, 2010.
- ²¹Nishikawa, H., “Two Ways to Extend Diffusion Schemes to Navier-Stokes Schemes: Gradient Formula or Upwinding,” *20th AIAA Computational Fluid Dynamics Conference*, AIAA Paper 2011-3044, Honolulu, Hawaii, 2011.
- ²²Nishikawa, H., “Robust and Accurate Viscous Discretization via Upwind Scheme - I: Basic Principle,” *Comput. Fluids*, Vol. 49, No. 1, October 2011, pp. 62–86.

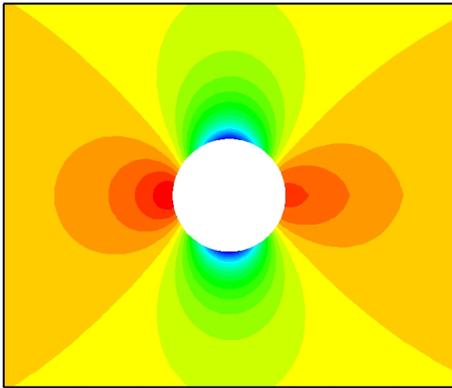
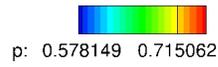
- ²³Nishikawa, H., Nakashima, Y., and Watanabe, N., "Effects of High-Frequency Damping on Iterative Convergence of Implicit Viscous Solver," *J. Comput. Phys.*, Vol. 348, 2017, pp. 66–81.
- ²⁴Venkatakrishnan, V., "Convergence to Steady State Solutions of the Euler Equations on Unstructured Grids with Limiters," *J. Comput. Phys.*, Vol. 118, 1995, pp. 120–130.
- ²⁵Seok Park, J., Hwan Yoon, S., and Kim, C., "Multi-dimensional limiting process for hyperbolic conservation laws on unstructured grids," *J. Comput. Phys.*, Vol. 229, 2010, pp. 788–812.
- ²⁶Nishikawa, H., "New-Generation Hyperbolic Navier-Stokes Schemes: $O(1/h)$ Speed-Up and Accurate Viscous/Heat Fluxes," *Proc. of 20th AIAA Computational Fluid Dynamics Conference*, AIAA Paper 2011-3043, Honolulu, Hawaii, 2011.
- ²⁷Nishikawa, H., "First, Second, and Third Order Finite-Volume Schemes for Navier-Stokes Equations," *Proc. of 7th AIAA Theoretical Fluid Mechanics Conference, AIAA Aviation and Aeronautics Forum and Exposition 2014*, AIAA Paper 2014-2091, Atlanta, GA, 2014.
- ²⁸Nishikawa, H., "Alternative Formulations for First-, Second-, and Third-Order Hyperbolic Navier-Stokes Schemes," *Proc. of 22nd AIAA Computational Fluid Dynamics Conference*, AIAA Paper 2015-2451, Dallas, TX, 2015.
- ²⁹Nakashima, Y., Watanabe, N., and Nishikawa, H., "Hyperbolic Navier-Stokes Solver for Three-Dimensional Flows," *54th AIAA Aerospace Sciences Meeting*, AIAA Paper 2016-1101, San Diego, CA, 2016.
- ³⁰Liu, Y. and Nishikawa, H., "Third-Order Inviscid and Second-Order Hyperbolic Navier-Stokes Solvers for Three-Dimensional Inviscid and Viscous Flows," *46th AIAA Fluid Dynamics Conference*, AIAA Paper 2016-3969, Washington, D.C., 2016.
- ³¹Li, L., Lou, J., Luo, H., and Nishikawa, H., "A New Formulation of Hyperbolic Navier-Stokes Solver based on Finite Volume Method on Arbitrary Grids," *AIAA 2018 Fluid Dynamics Conference*, AIAA Paper 2018-4160, Atlanta, Georgia, 2018.



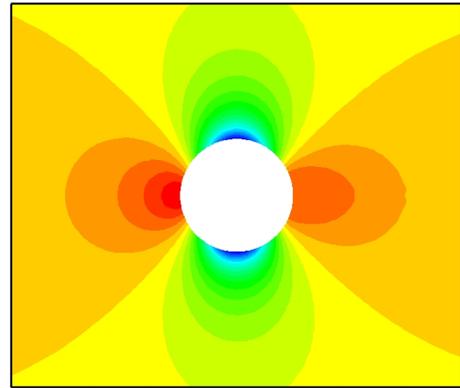
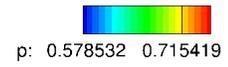
(a) Grid.



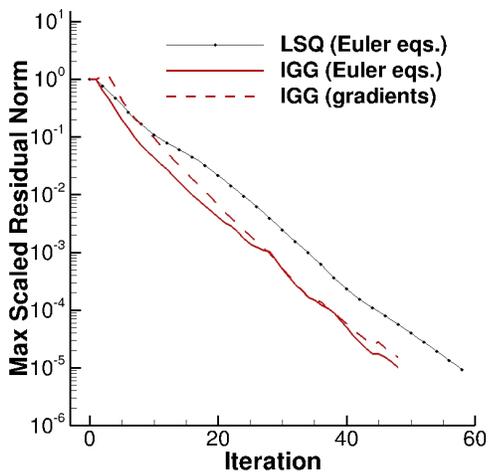
(b) Close view of the grid.



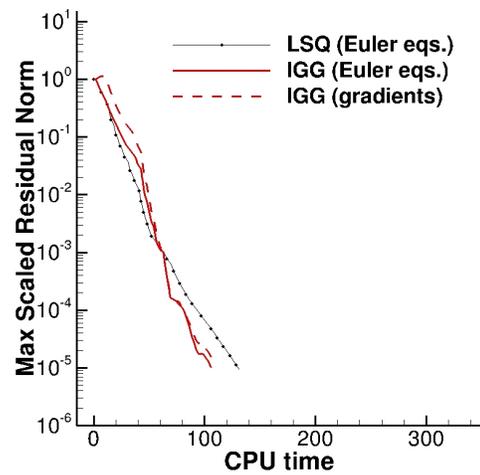
(c) Pressure contours: LSQ.



(d) Pressure contours: IGG.



(e) Maximum residual versus iteration.



(f) Maximum Residual versus CPU time.

Figure 2: Inviscid flow over a cylinder at $M_\infty = 0.3$ on a triangular grid.

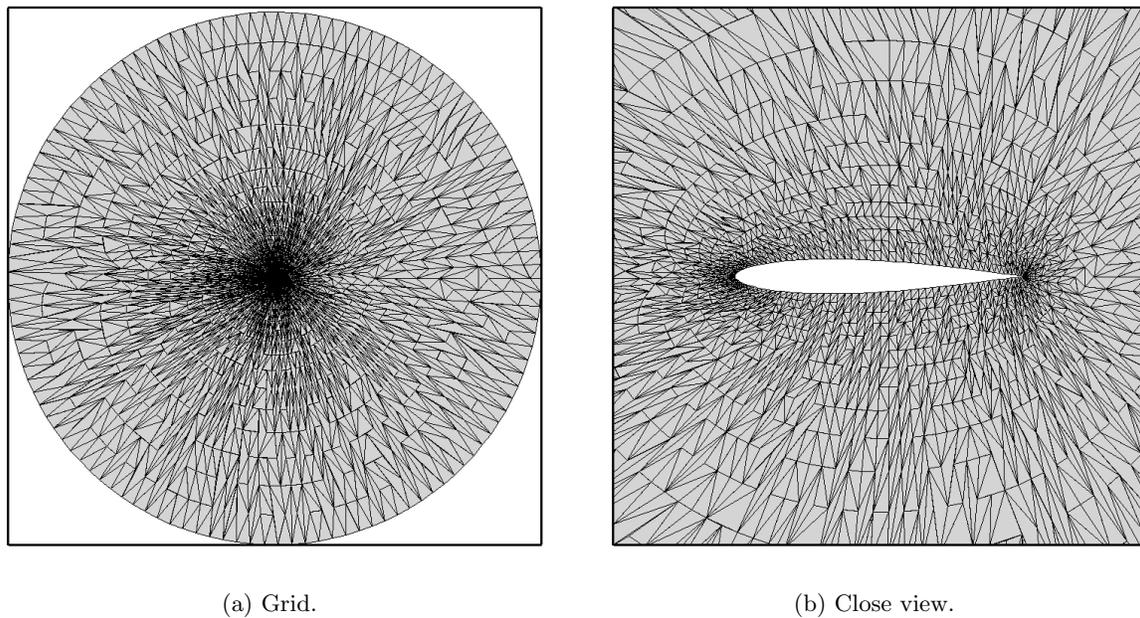


Figure 3: Irregular triangular grid used for the inviscid flow over a cylinder at $M_\infty = 0.3$.

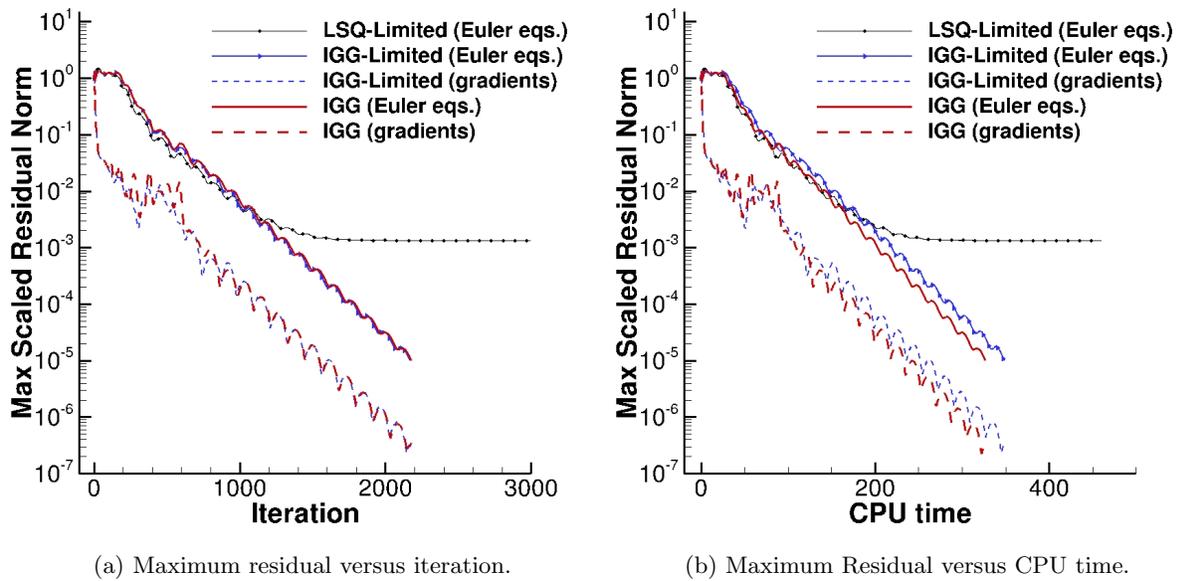


Figure 4: Convergence history for the airfoil case.

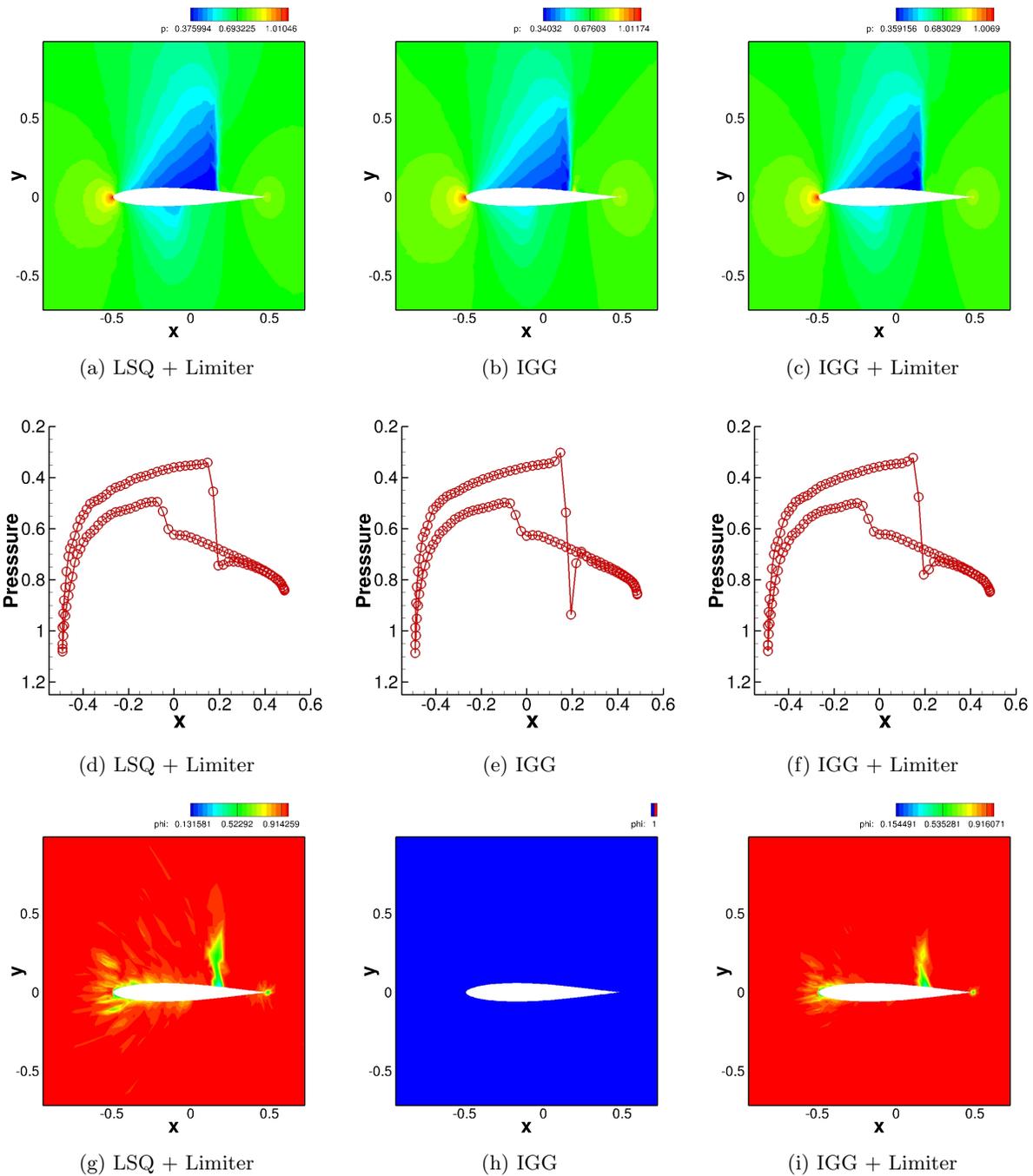
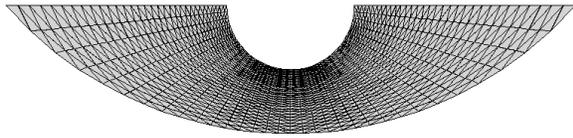
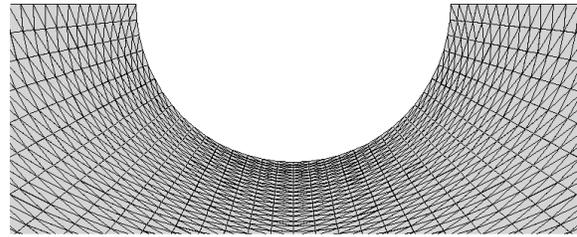


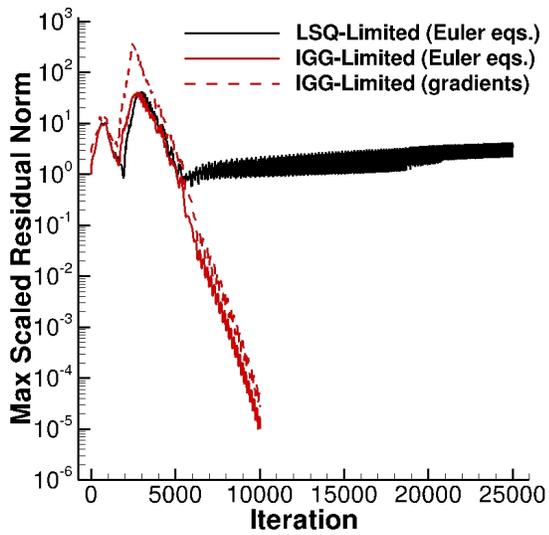
Figure 5: Inviscid flow over a Joukowski airfoil at $M_\infty = 0.85$ on a triangular grid.



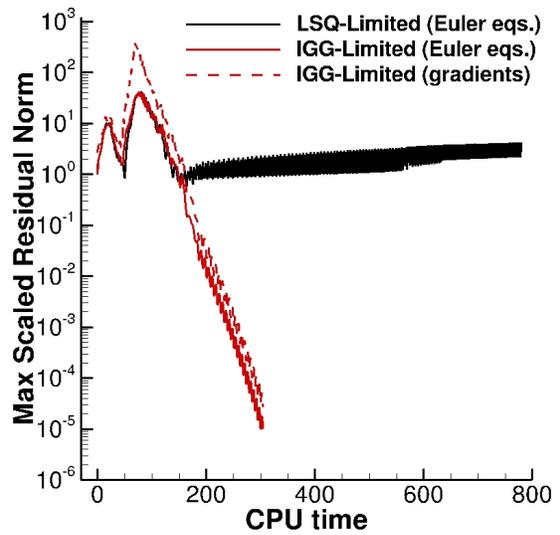
(a) Grid.



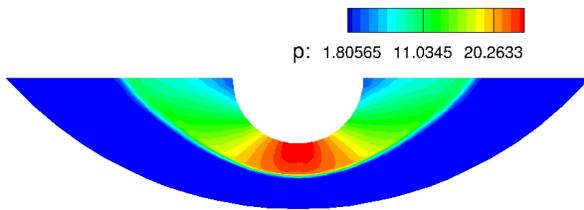
(b) Close view of the grid.



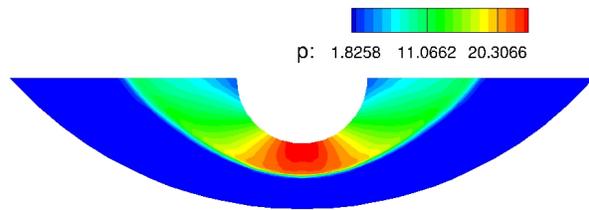
(c) Maximum residual versus iteration.



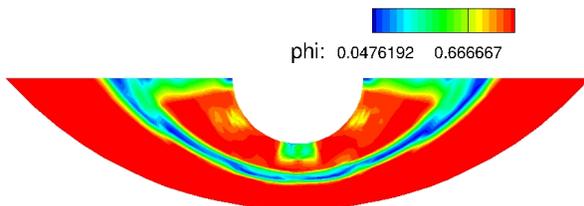
(d) Maximum Residual versus CPU time.



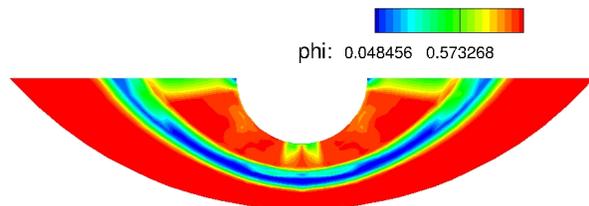
(e) Pressure contours: LSQ.



(f) Pressure contours: IGG.

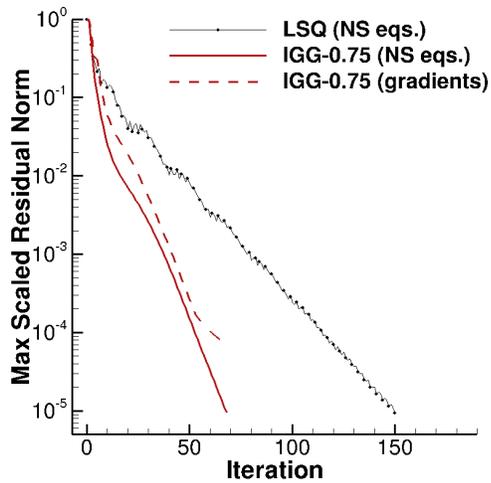


(g) Limiter contours: LSQ.

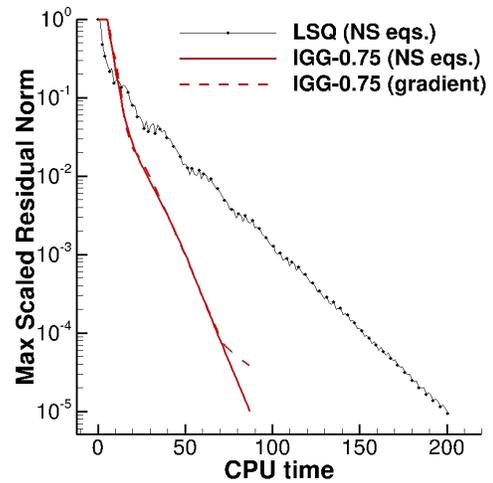


(h) Limiter contours: IGG.

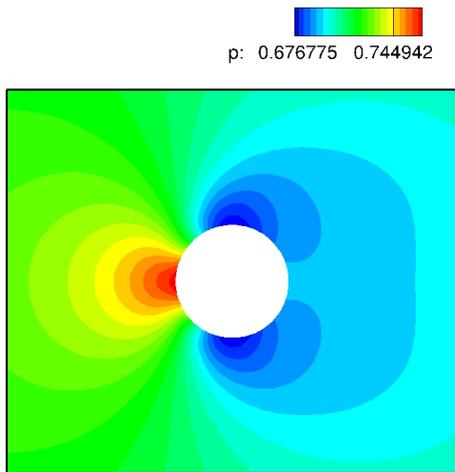
Figure 6: Hypersonic flow over a cylinder at $M_\infty = 5.2$ on a triangular grid.



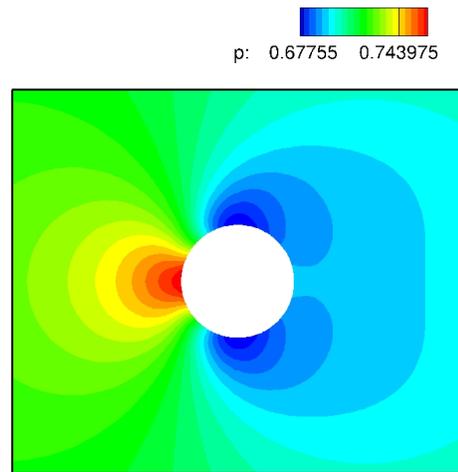
(a) Maximum residual versus iteration.



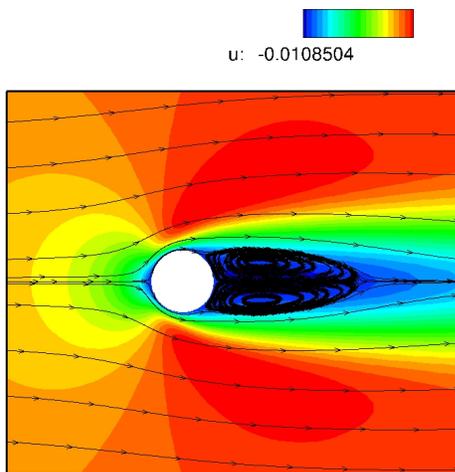
(b) Maximum Residual versus CPU time.



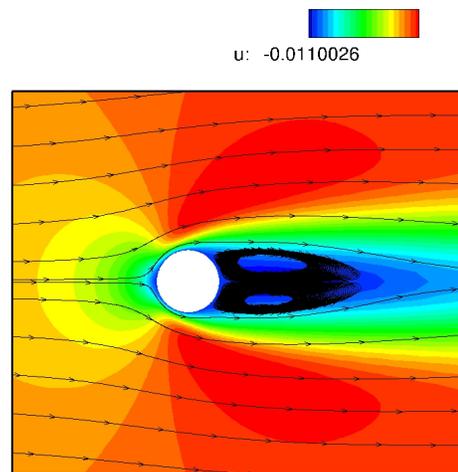
(c) Pressure contours: LSQ.



(d) Pressure contours: IGG.



(e) Streamlines and x -velocity contours: LSQ.



(f) Streamlines and x -velocity contours: IGG.

Figure 7: Viscous flow over a cylinder at $M_\infty = 0.3$ and $Re_\infty = 40$ on a triangular grid.

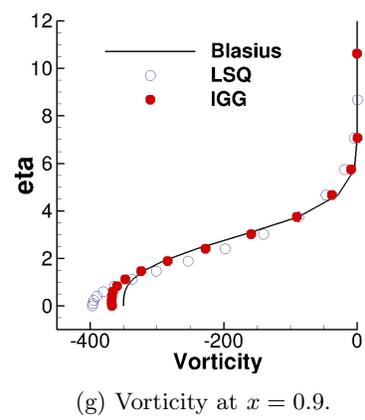
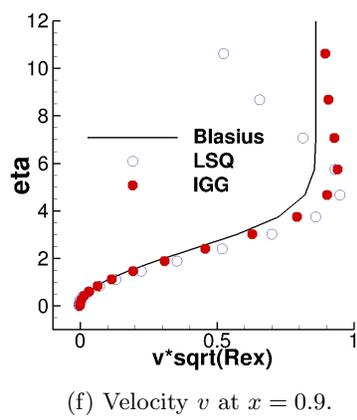
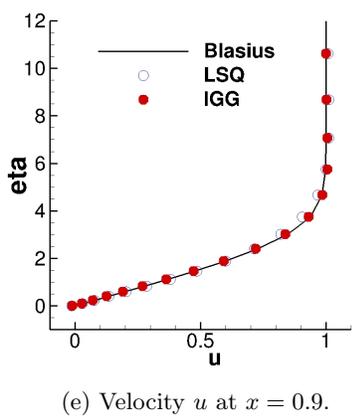
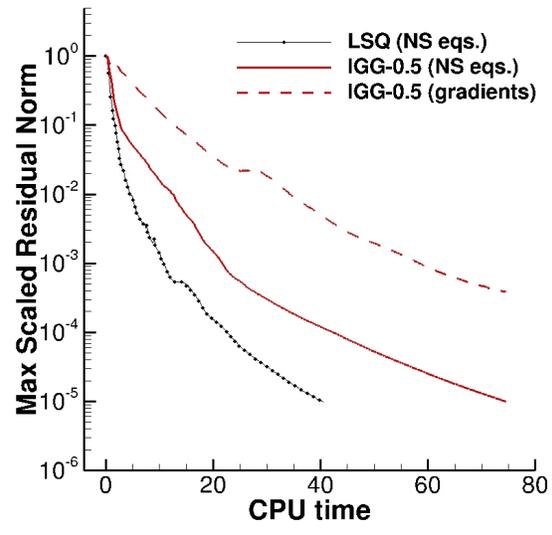
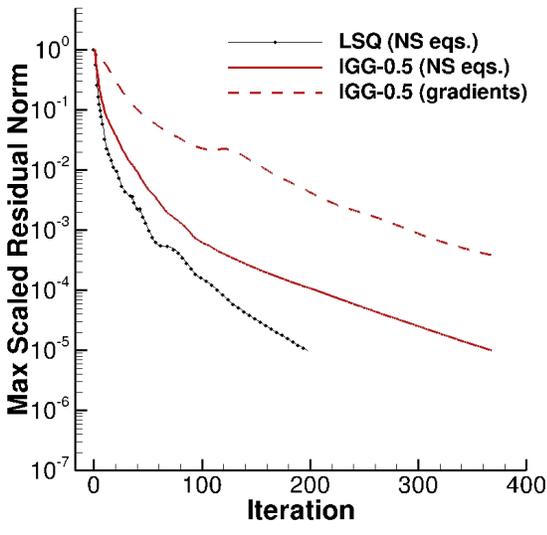
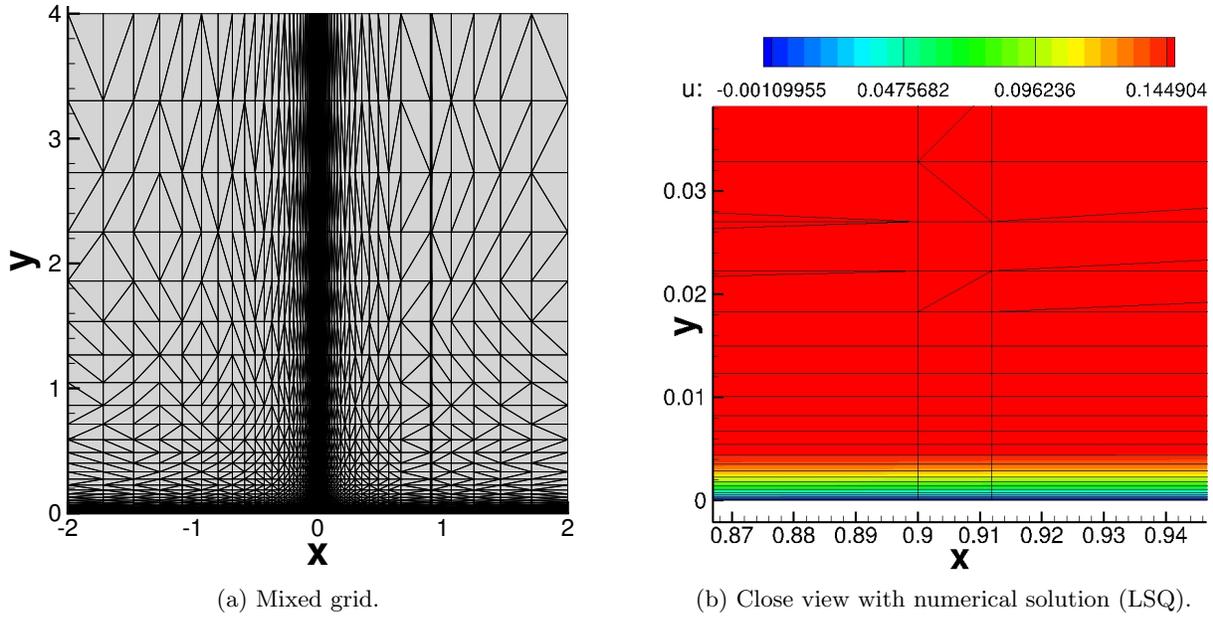


Figure 8: Viscous flow over a flat plate on a mixed grid at $M_\infty = 0.15$ and $Re_\infty = 10^6$.