



Third-Order Edge-Based Hyperbolic Navier-Stokes Scheme for Three-Dimensional Viscous Flows

Yi Liu* and Hiroaki Nishikawa†

National Institute of Aerospace, Hampton, VA 23666

In this paper, we present a third-order edge-based scheme for the three-dimensional Navier-Stokes equations. The node-centered edge-based scheme is known to achieve third-order accuracy on tetrahedral grids with quadratic least-squares gradients and linear flux reconstruction for first-order hyperbolic systems. It is extended to the viscous terms by the hyperbolic Navier-Stokes method, in which the viscous terms are written as a first-order hyperbolic system with source terms. The source terms introduced by the hyperbolic formulation are discretized by a new formula recently discovered, which does not require second derivatives. The developed scheme is implemented in NASA's FUN3D code, and tested for three-dimensional laminar flow problems.

I. Introduction

This paper reports further progress in the development of an economical third-order edge-based scheme in NASA's FUN3D code [1]. Extending the previously developed third-order inviscid scheme to the viscous terms, we present a third-order edge-based Navier-Stokes scheme for three-dimensional viscous flows. To achieve third-order accuracy for the viscous terms, we employ the hyperbolic Navier-Stokes (HNS) method [2, 3, 4, 5], which reformulates the viscous terms as a hyperbolic system with the solution gradients introduced as additional unknowns. The developed scheme is implemented in FUN3D, verified, and applied for viscous flow problems.

An extension of the third-order edge-based scheme to the conventional viscous terms has been reported in Ref. [6]. The method requires cubic least-squares (LSQ) gradients for the viscous fluxes. They proposed to compute the cubic LSQ gradients on systematically constructed high-order elements. Despite the simplicity in the LSQ algorithm, this approach has been shown to require high-order curved grids for curved geometries [7]. Also, the method has been applied to three-dimensional problems with strand grids, where the two-dimensional scheme is applied on triangular layers and carefully combined with a high-order finite-difference method in the direction normal to solid bodies [8]. Therefore, their third-order edge-based scheme is essentially two-dimensional even for three-dimensional problems.

On the other hand, our focus is on the development of the third-order edge-based scheme on tetrahedral grids for improving practical unstructured-grid turbulent-flow solvers with arbitrary isotropic/anisotropic grid adaptation for complex geometries [9, 10, 11, 12, 13]. For the viscous terms, we avoid cubic LSQ fits by reformulating the viscous terms as a first-order hyperbolic system, so that the same algorithm can be applied to the inviscid and viscous terms [1, 2, 3, 5, 4], i.e., with quadratic LSQ fits for both fluxes. Also, we have derived a general accuracy-preserving boundary flux quadrature formula, which enables to achieve third-order accuracy on linear triangular/tetrahedral grids for curved geometries [1, 14]. Previously, we developed an efficient construction of the third-order edge-based scheme for the inviscid terms by using a second-order edge-based scheme for the hyperbolized viscous terms. In this work, we extend the scheme to a full third-order scheme for the Navier-Stokes equations. To construct a third-order viscous scheme, the source terms introduced by the hyperbolic viscous formulation need to be carefully discretized to satisfy a compatibility condition required to preserve third-order accuracy. Special source term discretization formulas previously developed [6, 15, 16] require the computation and storage of the second derivatives of the source terms, which can be quite demanding in three dimensions. A recent study [17] has revealed that there exists a one-parameter family of source discretization formulas that does not require second derivatives at all. Also, there is a unique formula in the one-parameter family that

*Senior Research Scientist (yi.liu@nianet.org), National Institute of Aerospace, 100 Exploration Way, Hampton, VA 23666 USA, Senior Member AIAA

†Associate Research Fellow (hiro@nianet.org), National Institute of Aerospace, 100 Exploration Way, Hampton, VA 23666 USA, Associate Fellow AIAA

does not even require the gradients at the neighbor nodes. In this study, we employ this efficient formula to discretize the source terms in the hyperbolic viscous terms and those for the method of manufactured solutions. The resulting third-order Navier-Stokes scheme is highly efficient: (1)the residual is computed in a loop over edges with a single numerical flux per edge; (2)high-order curved mesh is not required for curved geometries; (3)only quadratic LSQ gradients are needed for the viscous and unsteady problems; (4)derivative quantities such as the viscous stresses, the heat fluxes, and the vorticity are obtained with third-order accuracy on irregular tetrahedral grids; (5)convergence acceleration due to reduced numerical stiffness is achieved on refined grids.

II. Hyperbolic Navier-Stokes System in Three Dimensions

The governing system of equations is the HNS20 system in three dimensions [5]:

$$\mathbf{P}^{-1}\partial_{\tau}\mathbf{U} + \text{div}\mathbf{F} + \mathbf{S} = 0, \quad (\text{II.1})$$

where τ is a pseudo time, \mathbf{P}^{-1} is a diagonal scaling matrix, \mathbf{U} is the conservative variables augmented with the gradient variables ($\mathbf{g}_u, \mathbf{g}_v, \mathbf{g}_w, \mathbf{q}, \mathbf{r}$), \mathbf{F} is the flux tensor, and \mathbf{S} is a source term vector arising from the hyperbolic formulation of the viscous terms. The gradient variables are proportional to the gradients of the primitive variables when the pseudo-time term vanishes or is ignored:

$$\mathbf{r} = \nu_{\rho} \nabla \rho, \quad \mathbf{g} = [\mathbf{g}_u, \mathbf{g}_v, \mathbf{g}_w] = [\mu_v \nabla u, \mu_v \nabla v, \mu_v \nabla w], \quad \mathbf{q} = -\frac{\mu_h}{\gamma(\gamma - 1)} \nabla T, \quad (\text{II.2})$$

where ρ is the density, $\mathbf{v} = (u, v, w)$ is the velocity vector, T is the temperature, γ is the ratio of specific heats, and

$$\nu_{\rho} = V_{min}, \quad \mu_v = \frac{4\mu}{3}, \quad \mu_h = \frac{\gamma\mu}{Pr}, \quad (\text{II.3})$$

V_{min} is the minimum dual control volume of a given grid [5] (i.e., ν_{ρ} is a global constant), μ is the viscosity given by Sutherland's law, and Pr is the Prandtl number. The pressure gradient can be obtained directly from \mathbf{r} and \mathbf{q} :

$$\nabla p = -\frac{\rho(\gamma - 1)\mathbf{q}}{\mu_h} + \frac{T\mathbf{r}}{\gamma\nu_{\rho}}. \quad (\text{II.4})$$

See Refs.[3,5] for more details of the HNS20 system. All solutions are non-dimensionalized by free-stream values except that the velocity and the pressure are normalized by the free-stream speed of sound and the free-stream dynamic pressure, respectively [5, 18, 19].

In this paper, we focus on the source term:

$$\mathbf{S} = [0, 0, 0, 0, 0, \mathbf{g}_u/\mu_v, \mathbf{g}_v/\mu_v, \mathbf{g}_w/\mu_v, \mathbf{q}/\mu_h, \mathbf{r}/\nu_{\rho}]^t. \quad (\text{II.5})$$

A careful source term discretization is required to preserve third-order accuracy. The general divergence formulation of source terms in Ref.[16] or the extended Galerkin formula [6] can be directly applied, but it will require computations of the second derivatives of the source term. To avoid second derivatives, a hyperbolic formulation for these source terms have been developed [3, 20, 21]. The source term is written as a first-order hyperbolic system, and therefore can be discretized straightforwardly by the third-order edge-based scheme with the quadratic LSQ gradients. However, this approach requires the gradients at the neighbor nodes, which can largely increase the stencil size especially when the quadratic LSQ stencil extends to neighbors of neighbors. To further reduce the stencil size of the source discretization, we have recently derived new source term discretization formulas that do not require second derivatives as well as the gradients at the neighbor nodes [17]. Here, we employ this very economical formula to discretize the source term \mathbf{S} .

III. Third-Order Edge-Based Scheme

The HNS20 system is discretized on a tetrahedral grid by the node-centered edge-based method. The discretization at a node j is defined as

$$V_j \frac{d\mathbf{U}_j}{d\tau} = -\mathbf{P}_j \left(\sum_{k \in \{k_j\}} \Phi_{jk} |\mathbf{n}_{jk}| - \int_{V_j} \mathbf{S} dV \right), \quad (\text{III.1})$$

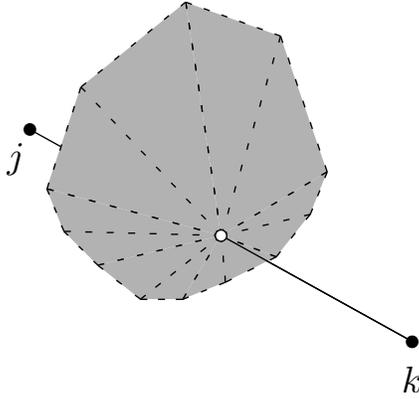


Figure 1: Dual control volume faces in a tetrahedral stencil.

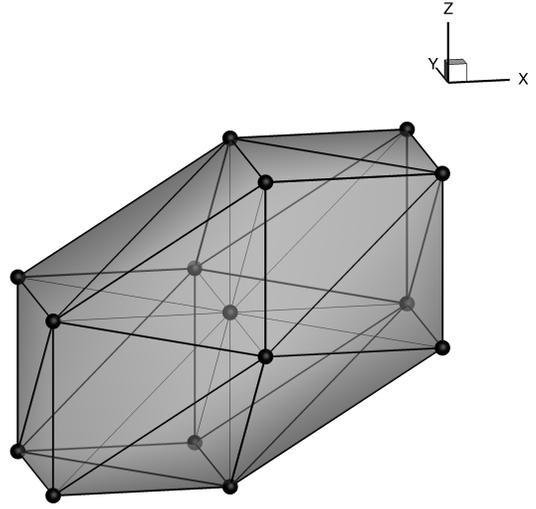


Figure 2: Regular tetrahedral grid.

where \mathbf{P}_j is the diagonal scaling matrix, V_j is the measure of the dual control volume around, $\{k_j\}$ is a set of neighbors of the node j , Φ_{jk} is a numerical flux, and \mathbf{n}_{jk} is the directed area vector, which is a sum of the directed-areas corresponding to the dual-triangular faces associated with all tetrahedral elements sharing the edge $[j, k]$ (see Figure 1). The numerical flux is upwind, and computed at the edge midpoint. The source term discretization is left open at this point.

To achieve third-order accuracy, the fluxes need to be linearly extrapolated to the edge midpoint, and the solution gradients need to be computed by a quadratic LSQ fit. In the previous work [1,5], the flux extrapolation was applied to the inviscid flux only and the quadratic fit was applied (with the two-step implementation as described in Ref.[21]) only to the primitive variables. Therefore, third-order accuracy was achieved only for the inviscid terms. In this work, we extend the third-order scheme to the viscous fluxes by performing the linear flux extrapolation for the viscous fluxes and computing the LSQ gradients of the gradient variables by the quadratic fit. In order to achieve third-order accuracy, the source term also needs to be discretized in a compatible manner as discussed in the next section.

IV. Source Term Discretizations

Third-order edge-based scheme relies on a special property of vanishing second-order truncation errors. On a regular tetrahedral grid as shown in Figure 2, the truncation error of the third-order edge-based scheme with $\mathbf{S} = 0$ is given by

$$\frac{1}{V_j} \sum_{k \in \{k_j\}} \Phi_{jk} |\mathbf{n}_{jk}| = \mathbf{R}_j - \frac{h^2}{12} [\partial_{xx} \mathbf{R}_j + \partial_{yy} \mathbf{R}_j + \partial_{zz} \mathbf{R}_j - \partial_{xy} \mathbf{R}_j - \partial_{yz} \mathbf{R}_j + \partial_{zx} \mathbf{R}_j] + O(h^3). \quad (\text{IV.1})$$

where h is a representative mesh spacing, and

$$\mathbf{R} = \text{div} \mathbf{F}. \quad (\text{IV.2})$$

In the pseudo steady state, we have $\mathbf{R} = \text{div} \mathbf{F} = 0$, and therefore the second-order truncation error vanishes:

$$\frac{1}{V_j} \sum_{k \in \{k_j\}} \Phi_{jk} |\mathbf{n}_{jk}| = \text{div} \mathbf{F} + O(h^3). \quad (\text{IV.3})$$

This is the mechanism by which the edge-based scheme achieves third-order accuracy by an algorithm almost like a second-order algorithm. Note that the above cancellation property is a requirement for third-order accuracy to be achieved on unstructured grids [17]. In order to preserve third-order accuracy with the source term, we

must discretize the source term such that

$$\frac{1}{V_j} \sum_{k \in \{k_j\}} \Phi_{jk} |\mathbf{n}_{jk}| - \int_{V_j} \mathbf{S} dV = \mathbf{R}'_j - \frac{h^2}{12} [\partial_{xx} \mathbf{R}'_j + \partial_{yy} \mathbf{R}'_j + \partial_{zz} \mathbf{R}'_j - \partial_{xy} \mathbf{R}'_j - \partial_{yz} \mathbf{R}'_j + \partial_{zx} \mathbf{R}'_j] + O(h^3), \quad (\text{IV.4})$$

where

$$\mathbf{R}' = \text{div} \mathbf{F} - \mathbf{S}, \quad (\text{IV.5})$$

so that we achieve

$$\frac{1}{V_j} \sum_{k \in \{k_j\}} \Phi_{jk} |\mathbf{n}_{jk}| - \int_{V_j} \mathbf{S} dV = \text{div} \mathbf{F} - \mathbf{S} + O(h^3), \quad (\text{IV.6})$$

Various source term quadrature formulas exist that satisfy this property [17]. In this work, we employ the most economical formula among those derived in Ref.[17]:

$$\int_{V_j} \mathbf{S} dV = \sum_{k \in \{k_j\}} \frac{(13\mathbf{S}_j + 3\partial_{jk}\mathbf{S}_j - 3\mathbf{S}_k) (\Delta \mathbf{x}_{jk} \cdot \mathbf{n}_{jk})}{60}, \quad (\text{IV.7})$$

where

$$\Delta \mathbf{x}_{jk} = (x_k - x_j, y_k - y_j, z_k - z_j), \quad \partial_{jk}\mathbf{S}_j = \Delta \mathbf{x}_{jk} \cdot \nabla \mathbf{S}_j. \quad (\text{IV.8})$$

This source term discretization requires only the source term gradients $\nabla \mathbf{S}_j$ at the node j , which are given by

$$\nabla \mathbf{S}_j = [0, 0, 0, 0, 0, \nabla(\mathbf{g}_u/\mu_v), \nabla(\mathbf{g}_v/\mu_v), \nabla(\mathbf{g}_w/\mu_v), \nabla(\mathbf{q}/\mu_h), \nabla(\mathbf{r}/\nu_\rho)]_j^t. \quad (\text{IV.9})$$

However, it actually does not require any additional LSQ gradient computations because these gradients are already available in the discretization of the HNS20 system. For example, consider $\nabla(\mathbf{g}_u/\mu_v)_j$:

$$\nabla(\mathbf{g}_u/\mu_v)_j = [\nabla \mathbf{g}_u/\mu_v - \mathbf{g}_u/\mu_v^2 \otimes \nabla \mu_v]_j \quad (\text{IV.10})$$

$$= \left[\nabla \mathbf{g}_u/\mu_v + \frac{4\gamma(\gamma-1)}{3\mu_v^2 \mu_h} \left(\frac{\partial \mu}{\partial T} \right) (\mathbf{g}_u \otimes \mathbf{q}) \right]_j. \quad (\text{IV.11})$$

The LSQ gradient $\nabla(\mathbf{g}_u)_j$ is already computed to perform the linear extrapolation, and all other quantities are available at the node j . Similarly, other source term gradients can be computed with available quantities at the node j :

$$\nabla(\mathbf{g}_v/\mu_v)_j = \left[\nabla \mathbf{g}_v/\mu_v + \frac{4\gamma(\gamma-1)}{3\mu_v^2 \mu_h} \left(\frac{\partial \mu}{\partial T} \right) (\mathbf{g}_v \otimes \mathbf{q}) \right]_j, \quad (\text{IV.12})$$

$$\nabla(\mathbf{g}_w/\mu_v)_j = \left[\nabla \mathbf{g}_w/\mu_v + \frac{4\gamma(\gamma-1)}{3\mu_v^2 \mu_h} \left(\frac{\partial \mu}{\partial T} \right) (\mathbf{g}_w \otimes \mathbf{q}) \right]_j, \quad (\text{IV.13})$$

$$\nabla(\mathbf{q}/\mu_h)_j = \left[\nabla \mathbf{q}/\mu_v + \frac{\gamma^2(\gamma-1)}{Pr\mu_h^3} \left(\frac{\partial \mu}{\partial T} \right) (\mathbf{q} \otimes \mathbf{q}) \right]_j, \quad (\text{IV.14})$$

$$\nabla(\mathbf{r}/\nu_\rho)_j = (\nabla \mathbf{r})_j / \nu_\rho. \quad (\text{IV.15})$$

Therefore, the stencil of the source term discretization matches that of the LSQ fit. In this work, the quadratic LSQ fit is used with neighbors and neighbors of the neighbors, and therefore the source discretization stencil extends to the neighbors of the neighbors, which is one-level smaller, at least, than those required by other techniques.

<u>Scheme</u>	<u>Discretization</u>						<u>HNS Source</u>
	Inviscid			Viscous			
	Flux		LSQ (ρ, \mathbf{v}, p)	Flux		LSQ ($\mathbf{r}, \mathbf{g}, \mathbf{q}$)	
FUN3D	Roe(2nd)	: 2	Linear	Galerkin(2nd)	: 1	None	None
FUN3D-i3rd	Roe(3rd)	: 3	Quadratic	Galerkin(2nd)	: 1	None	None
HNS20-IQ(2nd)	Roe(3rd)	: 2	C-quadratic	Upwind(2nd)	: 2	Linear	Point
HNS20-II(2nd)	Roe(3rd)	: 1	N/A	Upwind(2nd)	: 2	Linear	Point
HNS20-IQ(3rd)	Roe(3rd)	: 3	C-quadratic	Upwind(3rd)	: 3	Quadratic	Compact
HNS20-II(3rd)	Roe(3rd)	: 1	N/A	Upwind(3rd)	: 3	Quadratic	Compact

Table 1: Summary of discretizations. The number on the right side of each colon indicates the level of neighbors contributing to the residual at a node: 1 = up to the neighbors, 2 = up to neighbors of the neighbors, 3 = up to neighbors of the neighbors of the neighbors. Order of accuracy is indicated by 2nd and 3rd in the parentheses. Point indicates the point evaluation, and Compact is the compact formula (IV.7).

V. Results

The system of residual equations is solved by an implicit defect-correction solver similar to the one described specifically for the HNS schemes in Ref.[5]. The linear relaxation is performed by a multi-color Gauss-Seidel scheme available in FUN3D. In this study, we perform 15 linear relaxations per implicit iteration unless otherwise stated. The inviscid Jacobian is constructed by the linearization of Van Leer's flux vector splitting scheme for the inviscid terms. The viscous Jacobian is constructed by the exact linearization of the upwind flux for the hyperbolized viscous terms. The Jacobian is updated based on an efficient algorithm available in FUN3D, which automatically adjusts the Jacobian-update-frequency based on residual reduction. The pseudo-time term is discretized by the first-order backward Euler scheme, and the pseudo time step is defined locally with $CFL = 200$ for all cases unless otherwise stated.

Results are compared with FUN3D, FUN3D-i3rd, HNS20-IQ(2nd), and HNS20-II(2nd). See Ref.[1] for details of these schemes. We consider two versions of full third-order HNS schemes: HNS20-IQ(3rd) and HNS20-II(3rd). HNS20-IQ(3rd) uses the compact quadratic LSQ method for the primitive variables [1], and HNS20-II(3rd) directly replaces the LSQ gradients by the gradient variables for the primitive variables. HNS20-II schemes are used here for accuracy verification study only. It is still subject to an instability problem for boundary-layer problems [22], but can be employed for inviscid-dominated regions, e.g., a far-field vortical flow.

V.A. Accuracy Verification

Accuracy verification is performed by the method of manufactured solutions. Sine functions are made exact solutions to the Navier-Stokes equations by adding source terms. These source terms are discretized by the same source term quadrature formula as in Equation (IV.7). We solve a problem in a cubic domain with a family of consistently refined irregular tetrahedral grids of n^3 nodes: $n = 15, 20, 25, 30, 35, 40$. The code is fully parallelized, and 10 CPUs are used for this problem. See Figures 3 and 4. The boundary condition is of Dirichlet type, and thus the exact solution is strongly imposed at boundary nodes.

Error convergence results for the density, x -velocity, and pressure, and their x derivatives are shown in Figure 5 for $Re_\infty = 1$. Similar results have been obtained for other variables, and therefore are not shown. Third-order accuracy has been confirmed for the primitive variables in Figures 5(a), 5(b), and 5(c). For this low-Reynolds number problem, only the HNS20-IQ(3rd) and HNS20-II(3rd) schemes can achieve third-order accuracy. Others show second-order accuracy although the third-order inviscid approximation by FUN3D-i3rd and HNS20-IQ(2nd) leads to much lower levels of errors compared with FUN3D. Figures 5(d), 5(e), and 5(f) compare the errors in the gradients. Typically, the order of error convergence for gradients is one order lower than that of the primitive variables on irregular grids. Indeed, FUN3D and FUN3D-i3rd yield first-order accuracy in the gradients although FUN3D-i3rd gives much smaller errors and seemingly better than first-order

for the velocity gradients. On the other hand, HNS schemes are known to achieve the same order of error convergence for solution variables and their gradients. This is confirmed by the results: HNS20-IQ(2nd) and HNS20-II(2nd) schemes achieve second-order accuracy in the gradients, and HNS20-IQ(3rd) and HNS20-II(3rd) schemes achieve third-order accuracy in the gradients.

Second-order HNS schemes are known to achieve third-order accuracy in the primitive variables for high Reynolds numbers. Figure 6 shows results for $Re_\infty = 10^4$. As expected, HNS20-IQ(2nd) and HNS20-II(2nd) schemes yield third-order accuracy in the primitive variables while the gradients remain second-order accurate. Third-order accurate gradients are obtained only with third-order schemes, HNS20-IQ(3rd) and HNS20-II(3rd). This is confirmed by the results in Figures 6(d), 6(e), and 6(f).

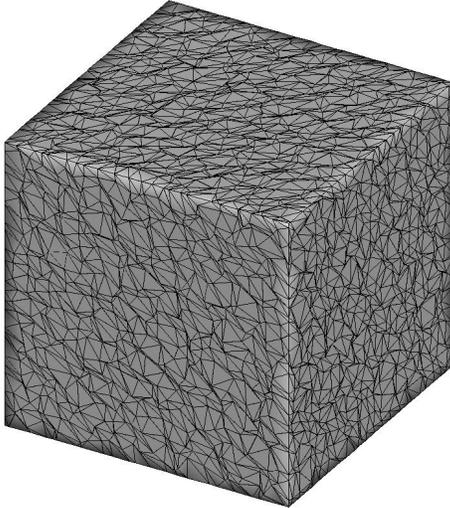


Figure 3: Irregular cube grid (25^3 nodes) for accuracy verification study.

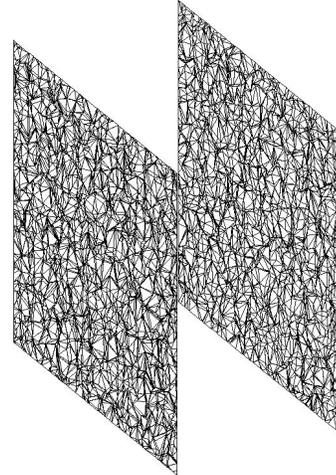


Figure 4: Slices of the irregular cube grid (25^3 nodes) for accuracy verification study.

V.B. Sphere

We consider a laminar flow over a sphere with $M_\infty = 0.15$ and $Re_\infty = 101$. The Reynolds number is defined based on the diameter of the sphere. An experimental value of the drag coefficient is given by $C_D = 1.08$ [23]. The sphere surface is systematically triangulated with no singularities (see Figure 7(b)), and a similar triangulation is used on the outer boundary, which is also a spherical surface and located at the distance of 100 times the diameter of the sphere. The viscous wall condition and the free stream condition are applied to the inner and outer boundaries, respectively. Four levels of tetrahedral grids have been generated with randomly perturbed nodes: 5668, 50228, 466830, and 1161591 nodes. The coarsest one is shown in Figures 7(a) and 7(b). The code was run with 10 CPUs for the first two coarse grids, and 240 for the last two fine grids. For this problem, the linear relaxation is performed to reduce the linear residual by one order of magnitude.

Contours of the z -component of the vorticity are compared for the finest grid in Figure 8. As typical for conventional schemes, the vorticity contours exhibit irregularity for FUN3D and even for FUN3D-i3rd, which uses a quadratic LSQ method. On the other hand, very smooth vorticity contours can be observed for the HNS schemes. Figure 9 shows convergence results for the finest grid. As can be seen in Figure 9(a), high-order schemes take more iterations to converge than FUN3D. However, Figure 9(b) shows that HNS20 schemes converge faster in CPU time than FUN3D and FUN3D-i3rd. This feature is unique to hyperbolic schemes, which eliminate numerical stiffness due to second derivatives in the viscous terms. The effect can be seen clearly in Figure 9(c), which shows the number of linear relaxations required per iteration. The number of linear relaxations increases quadratically for conventional viscous discretizations, and thus FUN3D and FUN3D-i3rd require a large number of linear relaxations especially on the finest grid. On the other hand, the HNS solvers converge rapidly even on the finest grid. The speed-up factor is $O(1/h)$, and thus the HNS solvers are expected to be more and more efficient than conventional solvers for further refined grids. This is a promising feature for high-fidelity large eddy or direct numerical simulations.

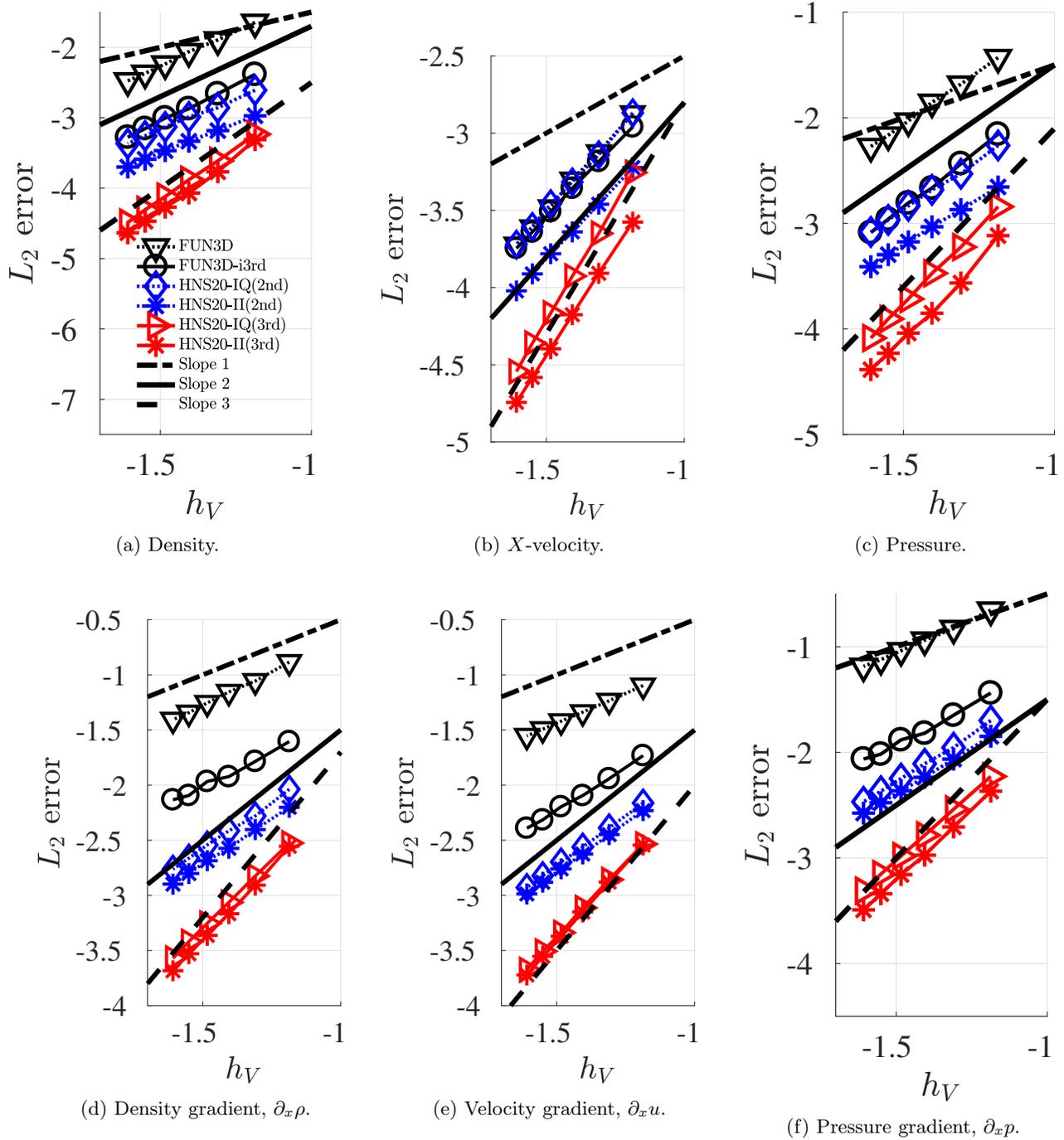


Figure 5: Error convergence results for $Re_\infty = 1$.

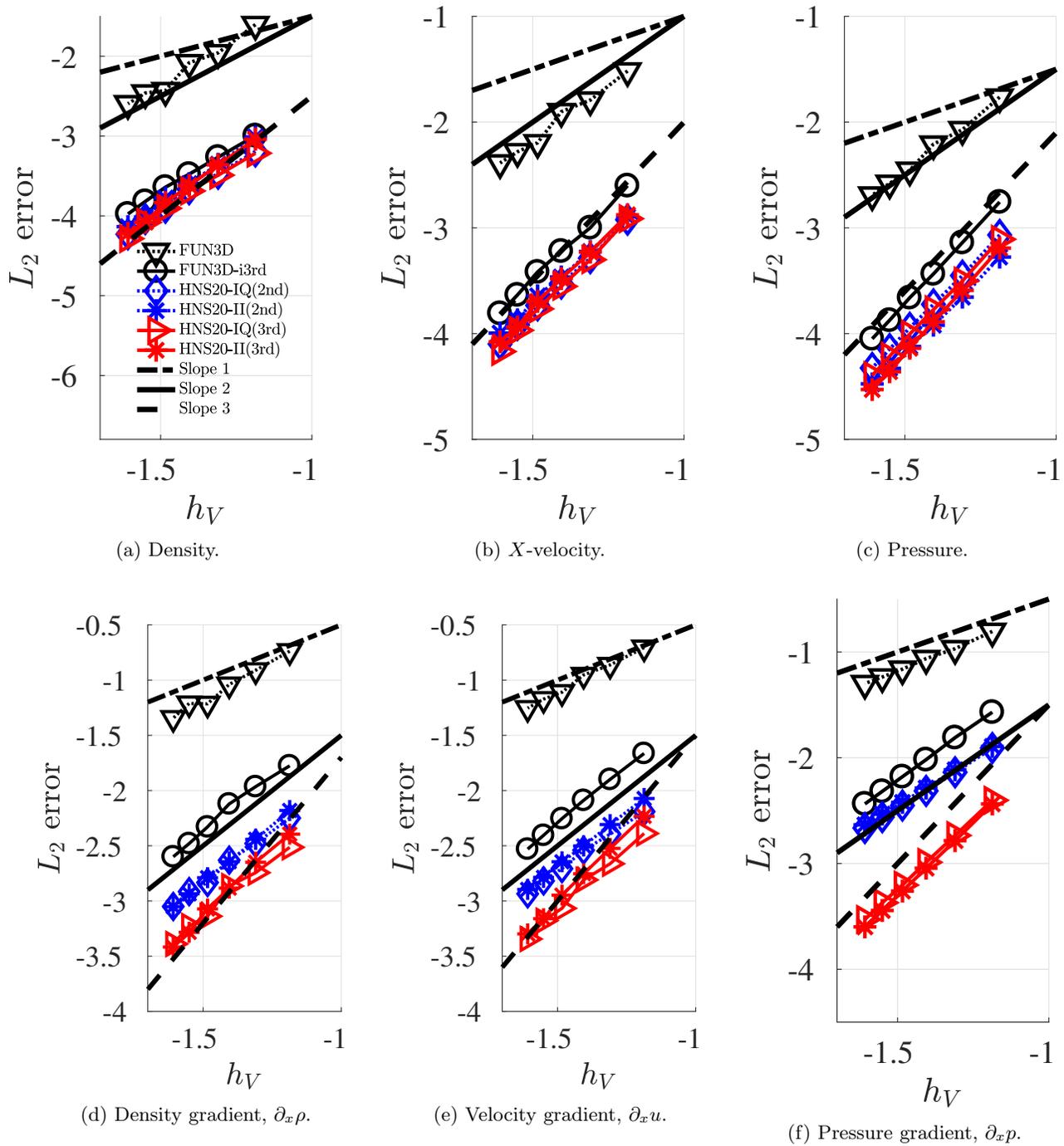
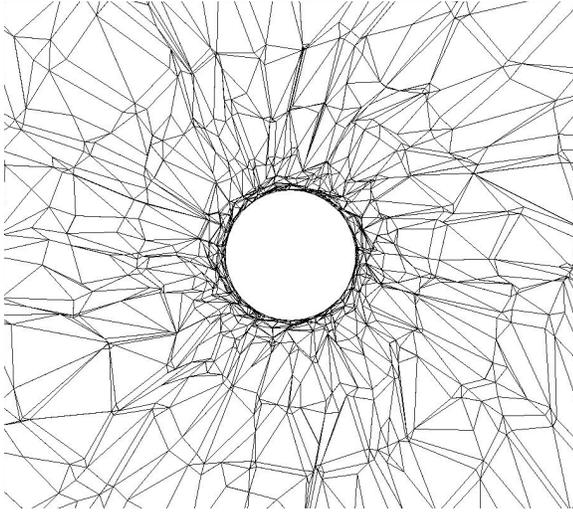
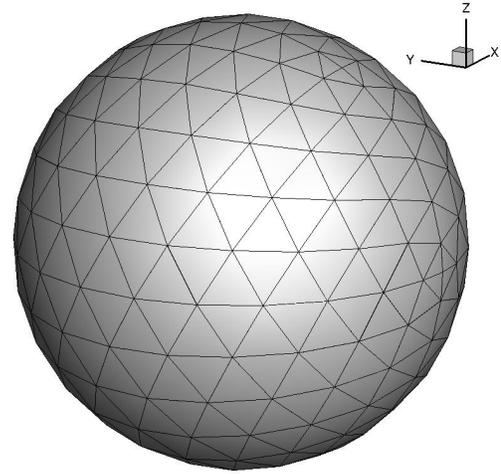


Figure 6: Error convergence results for $Re_\infty = 10^4$.



(a) Section plot at $x = 0.0$ for sphere.



(b) Surface grid for sphere.

Figure 7: Coarsest grid used for the sphere case.

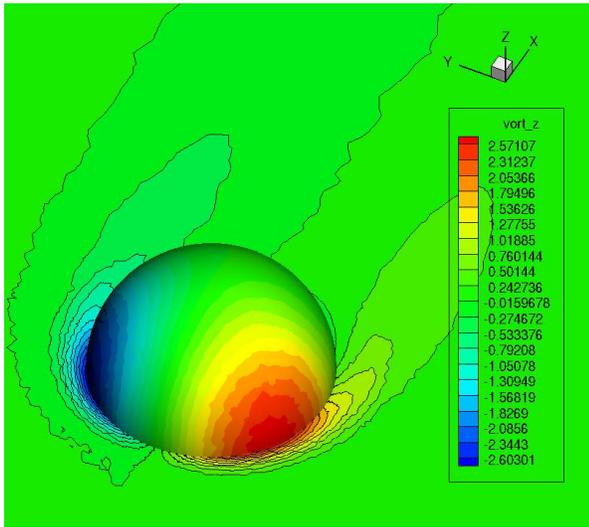
V.C. Joukowski Airfoil

A laminar flow over a Joukowski airfoil at the angle of attack 2° is considered for a moderate Reynolds number of 1000. The Reynolds number is defined based on the chord length of the airfoil. The flow conditions are $M_\infty = 0.5$ and $Re_\infty = 1,000$. The Joukowski airfoil is defined by a set of parameters: $\ell = 0.25$, $\epsilon = 0.3$, and $\kappa = 0.0$ (see Ref.[19] for the definitions of the parameters). The outer boundary is a circle located at the distance of 100 times the chord length. The viscous wall condition and the free stream condition are applied to the airfoil surface and the outer boundary, respectively. On the two boundary planes located at $y = 0$ and $y = 2$, a periodic boundary condition is imposed. Note that the domain is 3D, and therefore the case is equivalent to a 3D wing of infinite span. Four levels of tetrahedral grids have been generated with nodes randomly perturbed within (x, z) -plane: 9720 (10 CPUs), 64400 (120 CPUs), 462240 (240 CPUs), and 1500720 (240 CPUs) nodes; 81, 161, 321, 481 nodes on the airfoil. The coarsest grid is shown in Figures 10 and 11. It is emphasized that the grid lines are not orthogonal to the viscous surface due to the random perturbation, modeling adaptive viscous grids.

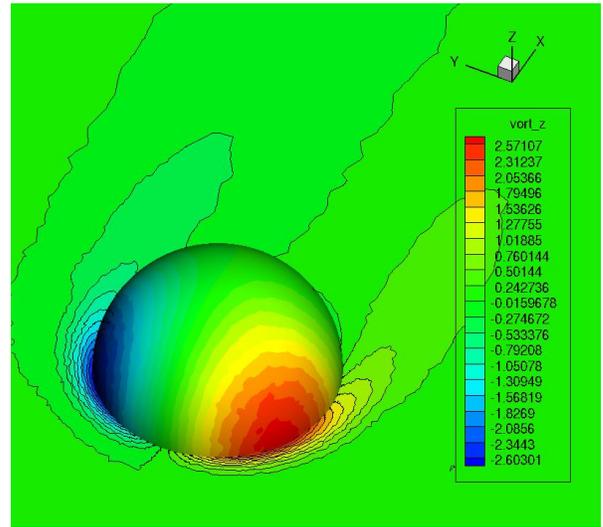
Vorticity contours are compared for the finest grid in Figure 12. The vorticity magnitude $|\text{curl } \mathbf{u}|$ was computed by LSQ gradients in the FUN3D and FUN3D-i3rd schemes, and by the gradient variables as $|\text{curl } \mathbf{u}| = |\mathbf{Eg}^t|$ in the HNS schemes, where \mathbf{E} is the third-rank alternating tensor of the Eddington epsilon [19]. It can be seen that contours exhibit irregularity in FUN3D as well as FUN3D-i3rd. HNS schemes, on the other hand, yield very smooth contours. These accurate vorticity contours directly impact the drag coefficient for the airfoil. Drag convergence is shown in Figure 13. The drag coefficients obtained on a series of grids by various schemes are plotted against the effective mesh spacing taking into account the fact that the HNS schemes has four times more discrete unknowns: $h_{DoF} = (4N)^{-1/3}$ for the HNS schemes and $h_{DoF} = N^{-1/3}$ for others, where N is the total number of nodes in the grid. Results show that HNS schemes give very fast drag convergence compared with FUN3D and FUN3D-i3rd.

V.D. Flat Plate

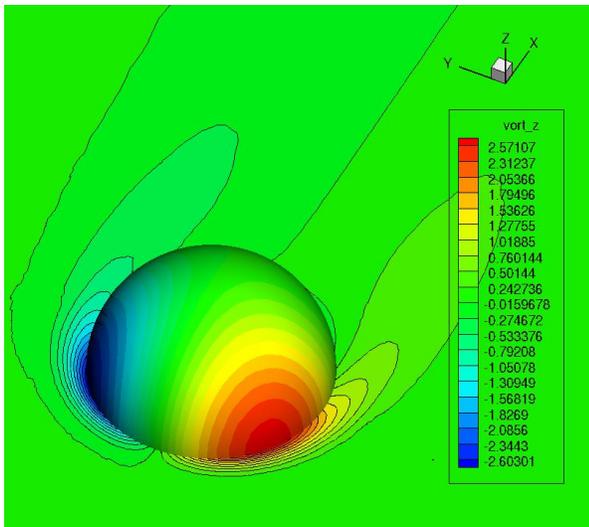
We consider a laminar flow over a flat plate at a Reynolds number of 1 million. The flow conditions are $M_\infty = 0.5$ and $Re_\infty = 10^6$. The Reynolds number is defined based on the length (in the x -direction) of the flat plate. The domain is taken to be rectangular $(x, y, z) \in [-2, 1] \times [0, 0.5] \times [0, 3]$, where the flat plate is located at the bottom ($z = 0$) and $(x, y) \in [0, 1] \times [0, 0.5]$. Four tetrahedral grids are generated over three identical 2D triangular-mesh planes, each of which consists of 64×32 (10 CPUs), 128×64 (120 CPUs), 256×128 (240 CPUs), and 384×192 nodes (400 CPUs). Figure 14 shows the coarsest grid. A parabolic stretching has been applied across the boundary layer in order to resolve the boundary layer equally along the flat plate (see Figure 14(b)). A reference value of the drag coefficient is 0.001328 based on a theory for incompressible viscous flows [24]. A periodic boundary condition is used at boundary planes at $y = 0$ and $y = 0.5$, and a symmetric condition is



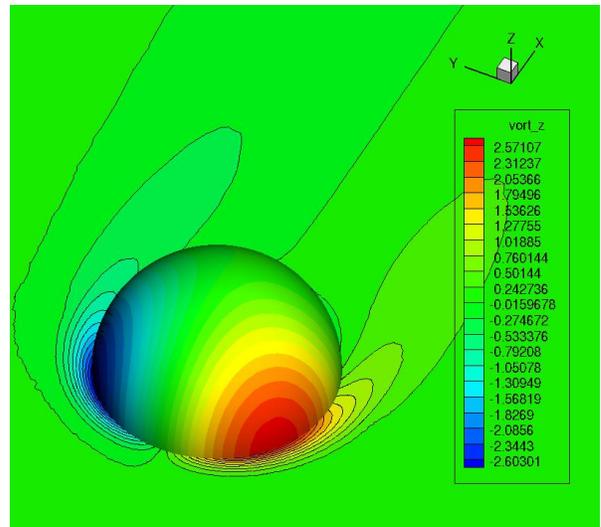
(a) FUN3D.



(b) FUN3D-i3rd.

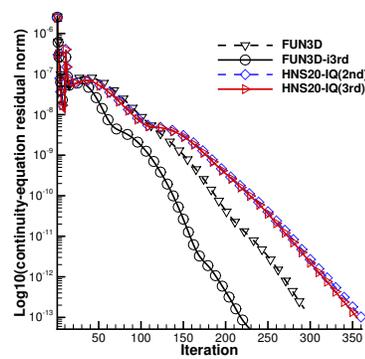


(c) HNS20-IQ(2nd).

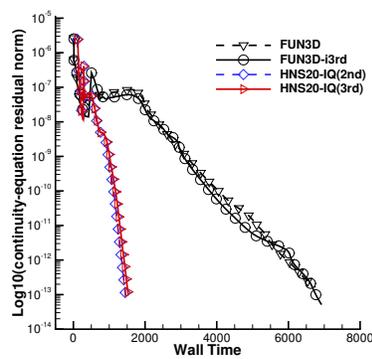


(d) HNS20-IQ(3rd).

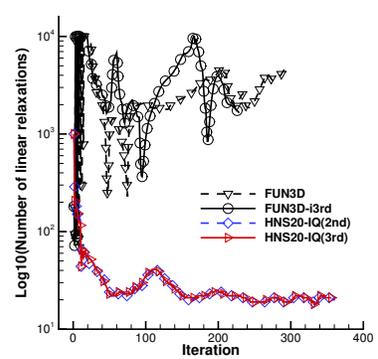
Figure 8: Contours of the magnitude of the vorticity for Sphere (Grid4).



(a) Iteration.



(b) CPU time.



(c) Linear relaxation.

Figure 9: Iterative convergence for the sphere case.

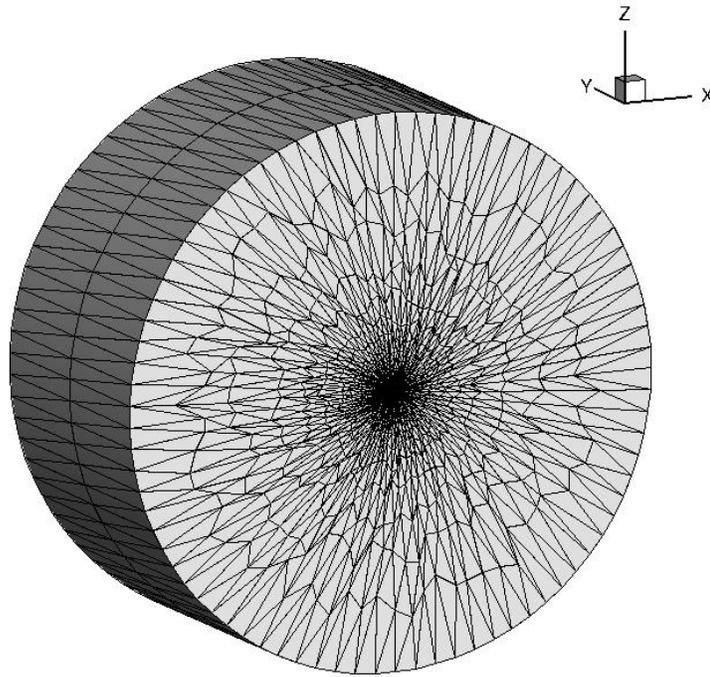
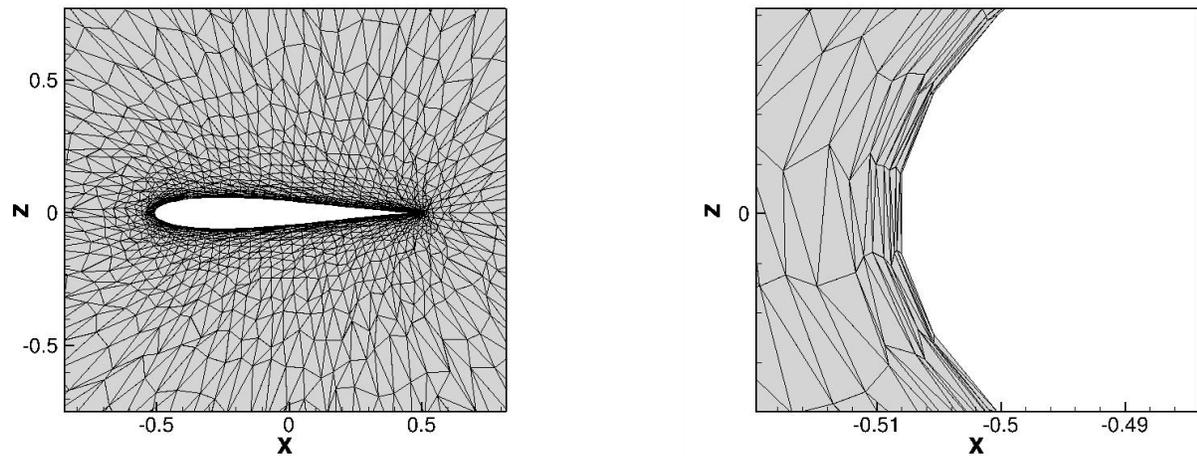


Figure 10: 3D grid for Joukowski airfoil (Grid1).



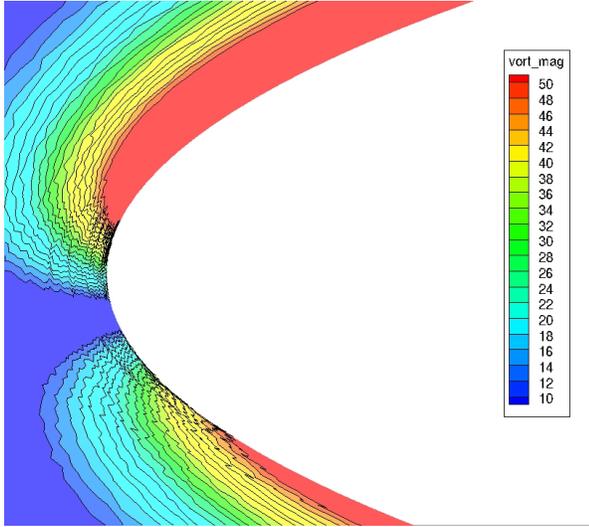
(a) Near-field view.

(b) Near-field view close to the viscous wall.

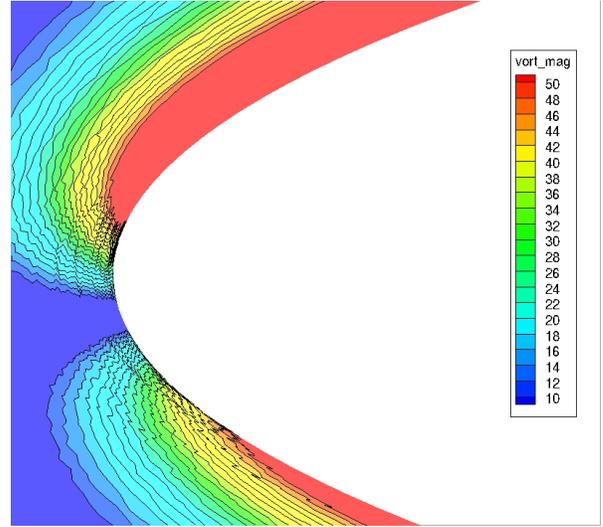
Figure 11: Near-field views in (x, z) -plane for Joukowski airfoil (Grid1).

used at $z = 0$ ahead of the flat plate. Free stream condition is used at the inflow boundary ($x = -2$), and the back pressure condition is used at the top ($z = 3$) and outflow ($x = 1$) boundaries.

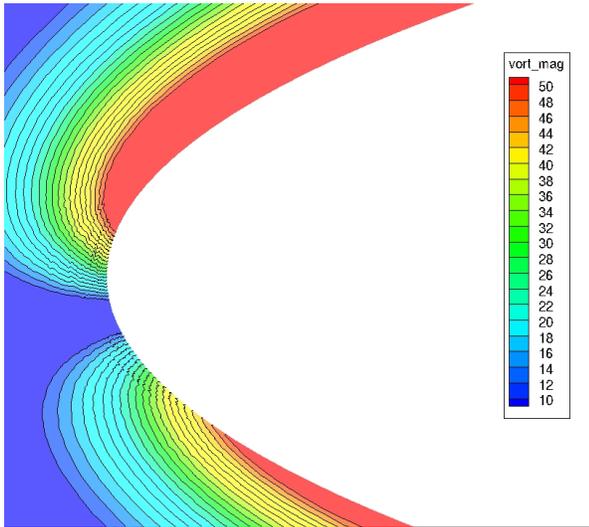
Drag convergence results are presented in Figure 15. It is observed that HNS schemes yield more accurate drag prediction than FUN3D and FUN3D-i3rd schemes do for the same number of discrete unknowns. The HNS20-IQ(3rd) scheme gives better prediction on coarse grids than other HNS schemes. To show that the HNS schemes give accurate predictions not only in the integrated drag but also in the local skin friction, the distributions of the local skin friction are compared on the coarsest grid in Figure 16. It can be seen that the HNS20-IQ(3rd) and HNS20-IQ(2nd) give more accurate local skin friction values than FUN3D.



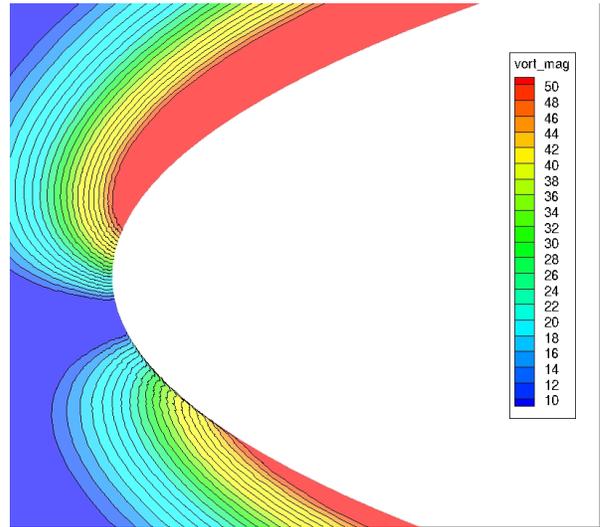
(a) FUN3D: Linear LSQ $|\text{curl } \mathbf{u}|$.



(b) FUN3D-i3rd: Quadratic LSQ $|\text{curl } \mathbf{u}|$.



(c) HNS20-IQ(2nd): $|\mathbf{Eg}^t| (= |\text{curl } \mathbf{u}|)$.



(d) HNS20-IQ(3rd): $|\mathbf{Eg}^t| (= |\text{curl } \mathbf{u}|)$.

Figure 12: Contours of the magnitude of the vorticity for Joukowski airfoil (Grid4).

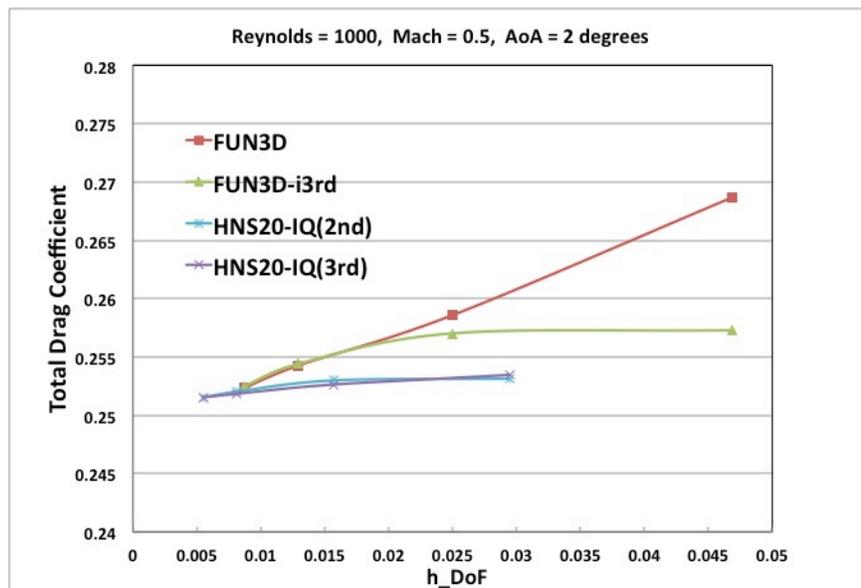
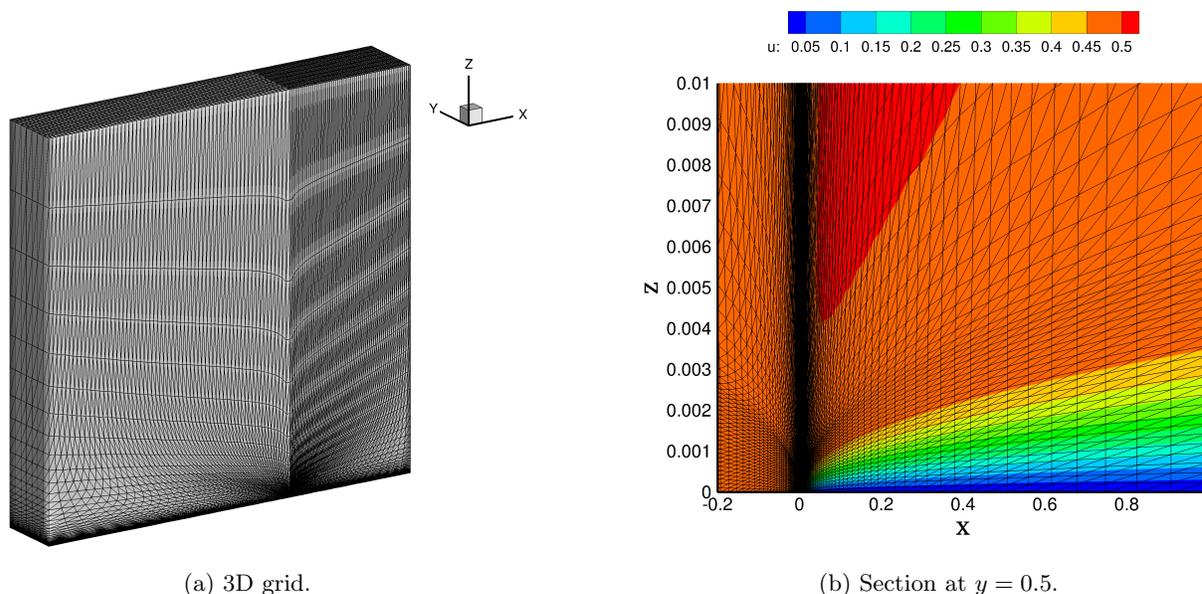


Figure 13: Drag coefficient convergence for Joukowsky airfoil.



(a) 3D grid.

(b) Section at $y = 0.5$.

Figure 14: Grid used for the 3D flat plate.

VI. Concluding Remarks

Third-order edge-based scheme has been extended to the Navier-Stokes equations for tetrahedral grids by using a hyperbolic viscous formulation. Source terms arising from the hyperbolic viscous formulation has been discretized by an efficient accuracy-preserving quadrature formula, which does not require second derivatives of the source terms. By the hyperbolic formulation, the viscous terms are discretized by the same algorithm used in the inviscid terms: a quadratic least-squares fit and a linear flux-extrapolation. Moreover, again due to the hyperbolic formulation, the resulting third-order Navier-Stokes scheme has been shown to achieve third-order accuracy for the primitive variables and their gradients. Since third-order accuracy in gradients generally requires a fourth-order scheme on unstructured grids, the ability of delivering third-order accurate gradients by a third-order scheme implies very high efficiency of the developed third-order edge-based scheme.

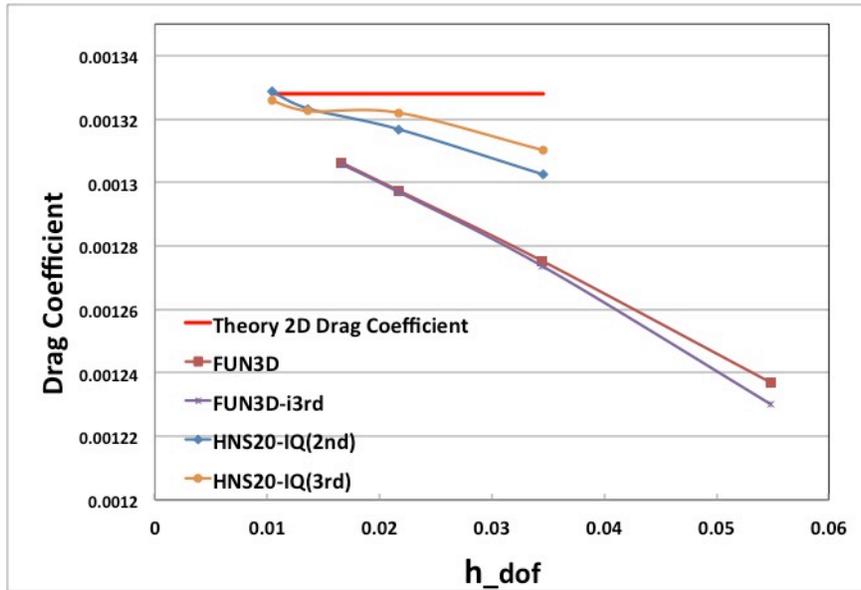


Figure 15: Drag coefficient convergence for the flat plate problem.

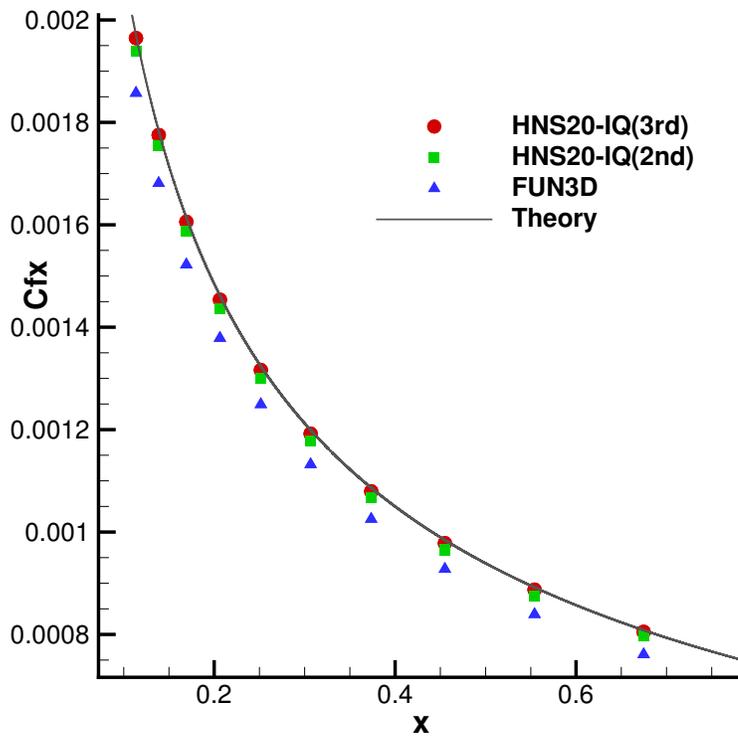


Figure 16: Skin friction comparison.

Acknowledgments

This work has been funded by NASA under Contract No. NNL09AA00A with Dr. Veer N. Vatsa as the technical monitor.

References

- ¹Liu, 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.
- ²Nishikawa, 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.
- ³Nishikawa, 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.
- ⁴Nishikawa, 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.
- ⁵Nakashima, 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.
- ⁶Pincock, B. and Katz, A., “High-Order Flux Correction for Viscous Flows on Arbitrary Unstructured Grids,” *J. Sci. Comput.*, Vol. 61, 2014, pp. 454–476.
- ⁷Work, D. G. and Katz, A., “Aspects of the Flux Correction Method for Solving the Navier-Stokes Equations on Unstructured Meshes,” *Proc. of 53rd AIAA Aerospace Sciences Meeting*, AIAA Paper 2015-0834, Kissimmee, Florida, January 2015.
- ⁸Tong, O., Katz, A., Yanagita, Y., Casey, A., and Schaap, R., “High-Order Methods for Turbulent Flows on Three-Dimensional Strand Grids,” *J. Sci. Comput.*, 2015, pp. 1–19.
- ⁹Park, M. A. and Darmofal, D. L., “Parallel Anisotropic Tetrahedral Adaptation,” *Proc. of 46th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper 2008-917, Reno, Nevada, 2008.
- ¹⁰Yano, M. and Darmofal, D. L., “An Optimization-Based Framework for Anisotropic Simplex Mesh Adaptation,” *Journal of Computational Physics*, Vol. 231, No. 22, Sept. 2012, pp. 7626–7649.
- ¹¹Michal, T. and Krakos, J., “Anisotropic Mesh Adaptation Through Edge Primitive Operations,” *Proc. of 50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, AIAA Paper 2012-0159, Nashville, Tennessee, 2012.
- ¹²Alauzet, F. and Loseille, A., “A Decade of Progress on Anisotropic Mesh Adaptation for Computational Fluid Dynamics,” *Computer-Aided Design*, March 2015, pp. 13–39.
- ¹³Park, M. A., Loseille, A., Krakos, J., Michal, T., and Alonso, J. J., “Unstructured Grid Adaptation: Status, Potential Impacts, and Recommended Investments Toward CFD Vision 2030,” *46th AIAA Fluid Dynamics Conference*, AIAA Paper 2016-3323, Washington, D.C., 2016.
- ¹⁴Nishikawa, H., “Accuracy-Preserving Boundary Flux Quadrature for Finite-Volume Discretization on Unstructured Grids,” *J. Comput. Phys.*, Vol. 281, 2015, pp. 518–555.
- ¹⁵Katz, A., “Source Term Discretizations for the Linear and Corrected Galerkin Schemes,” Unpublished, 2012.
- ¹⁶Nishikawa, H., “Divergence Formulation of Source Term,” *J. Comput. Phys.*, Vol. 231, 2012, pp. 6393–6400.
- ¹⁷Nishikawa, H. and Liu, Y., “Accuracy-Preserving Source Term Quadrature for Third-Order Edge-Based Discretization,” *J. Comput. Phys.*, Vol. 344, 2017, pp. 595–622.
- ¹⁸Biedron, R. T., Carlson, J.-R., Derlaga, J. M., Gnoffo, P. A., Hammond, D. P., Jones, W. T., Kleb, B., Lee-Rausch, E. M., Nielsen, E. J., Park, M. A., Rumsey, C. L., Thomas, J. L., and Wood, W. A., “FUN3D Manual: 12.9,” *NASA-TM-2016-219012*, February 2016.
- ¹⁹Masatsuka, K., “I do like CFD, VOL.1, Second Edition, Version 2.3,” <http://www.cfdbooks.com>, 2016.
- ²⁰Nishikawa, H., “First-, Second-, and Third-Order Finite-Volume Schemes for Diffusion,” *J. Comput. Phys.*, Vol. 256, 2014, pp. 791–805.
- ²¹Nishikawa, H., “First, Second, and Third Order Finite-Volume Schemes for Advection-Diffusion,” *J. Comput. Phys.*, Vol. 273, 2014, pp. 287–309.
- ²²Nishikawa, H. and Liu, Y., “Hyperbolic Navier-Stokes Method for High-Reynolds-Number Boundary-Layer Flows,” *55th AIAA Aerospace Sciences Meeting*, AIAA Paper 2017-0738, Grapevine, Texas, 2017.
- ²³Roos, F. W. and Willmarth, W. W., “Some Experimental Results on Sphere and Disk Drag,” *AIAA J.*, Vol. 9, No. 2, February 1971, pp. 285–291.
- ²⁴White, F. M., *Viscous Fluid Flow*, McGraw-Hill, 2nd ed., 1991.