



Third-Order Edge-Based Scheme for Unsteady Problems

Hiroaki Nishikawa* and Yi Liu[†]

National Institute of Aerospace, Hampton, VA 23666

In this paper, we discuss third-order edge-based schemes for unsteady problems. Second- and third-order unconditionally-stable implicit time-integration schemes, BDF2 and ESDIRK3, are compared, and effects of different orders of time accuracy are examined. Numerical results indicate that the orders of accuracy in time and space need to match to fully take advantage of the low-dispersive feature of third-order edge-based schemes. Furthermore, the third-order scheme combined with a low-dissipation numerical flux is shown to serve as a simple and economical scheme for high-resolution simulations. The third-order low-dissipation edge-based scheme has been implemented in NASA's FUN3D with a low-dissipative Roe flux, and demonstrated for a three-dimensional unsteady viscous flow over a cylinder. Results indicate that the third-order time-accurate edge-based scheme with a low-dissipation Roe flux is a practical approach to high-fidelity unstructured-grid simulations over complex geometries.

I. Introduction

Second-order node-centered edge-based method is one of the most successful practical algorithms in Computational Fluid Dynamics (CFD) solvers [1, 2, 3, 4, 5, 6, 7, 8]. Recent studies [9, 10, 11] revealed that third-order accuracy can be achieved by the edge-based method for first-order systems of conservation laws on arbitrary triangular and tetrahedral grids with quadratic least-squares (LSQ) gradients and linearly-extrapolated fluxes. The third-order edge-based method is a very efficient third-order algorithm in that the spatial residual can be computed over a single loop over edges with a single numerical flux evaluation per edge and second derivatives are not required at all. Another highly attractive feature is the ability to deliver third-order accurate solutions on linear triangular/tetrahedral grids even for curved geometries [12, 13], so that it can be immediately applicable to existing problems without generating high-order grids. Because of these very low-cost features, the third-order edge-based method has become a subject of great interest to CFD developers and practitioners; See Refs. [13, 14, 15, 16, 13, 17, 18, 19] for recent developments.

Very few work, however, has been reported in the literature regarding the application of the third-order edge-based scheme to unsteady problems. In Ref. [20], the third-order edge-based scheme is applied to the two-dimensional Navier-Stokes computations with a third-order conditionally-stable backward difference scheme, and demonstrated for an unsteady flow over a square cylinder. In Ref. [17], the third-order edge-based inviscid scheme was applied to three-dimensional viscous unsteady simulations, and demonstrated its superior accuracy. Although encouraging results have been reported, algorithmic details of the unsteady third-order edge-based schemes remain to be explored and understood.

In this study, we provide more detailed discussions on the time-accurate version of the third-order edge-based scheme. First, we discuss the spatial discretization of the time derivative term, which needs to be performed carefully in order to preserve third-order spatial accuracy. For this purpose, a special formula derived in Ref. [21] is employed, which does not require the computation and storage of the second-derivatives of the time derivative. For time-integration, we consider only L-stable methods: first- and second-order backward difference (BDF) schemes, and the third-order explicit first-stage singly diagonally implicit Runge-Kutta scheme (ESDIRK). A low-dissipation modification to upwind fluxes is also explored towards applications to third-order high-resolution simulations such as aero-acoustic propagation and large-eddy simulations.

We begin by testing the third-order unsteady edge-based scheme for a one-dimensional advection problem. Basic properties and relative merits of different time-integration schemes are discussed. Effects of a

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

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

low-dissipation upwind flux are also discussed. A few results for more realistic cases in three dimensions are presented, which were obtained with NASA's FUN3D code, in which the third-order unsteady edge-based scheme has already been implemented.

II. Model Equations

Consider the one-dimensional conservation law:

$$\partial_t \mathbf{u} + \partial_x \mathbf{f} = \mathbf{s}, \quad (\text{II.1})$$

where \mathbf{u} is the solution variable vector, and \mathbf{f} is the flux vector. In this study, we consider two scalar equations with $\mathbf{u} = u$: the linear advection-diffusion $\mathbf{f} = au - \nu \partial_x u$, and the viscous Burgers equation $\mathbf{f} = u^2/2 - \nu \partial_x u$, where a and ν are positive constants. We consider also the hyperbolic advection-diffusion system [22], which is used to solve the linear advection-diffusion and viscous Burgers' equations:

$$\mathbf{u} = \begin{bmatrix} u \\ 0 \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} f - \nu p \\ -u/T_r \end{bmatrix}, \quad \mathbf{s} = \begin{bmatrix} 0 \\ -p/T_r \end{bmatrix}, \quad (\text{II.2})$$

where $f = au - \nu p$ for the linear advection-diffusion equation, and $f = u^2/2 - \nu p$ for the viscous Burgers' equation. The relaxation time is defined by

$$T_r = L_r^2/\nu, \quad L_r = 1/2\pi, \quad (\text{II.3})$$

where the length scale L_r needs to be suitably modified for boundary layer problems [23] and dimensional problems [24]. The hyperbolic advection-diffusion system is equivalent to the scalar advection-diffusion and viscous Burgers' equations; it is hyperbolic in pseudo time as discussed below.

For implicit time-stepping schemes, the governing equation is typically written as

$$\partial_\tau \mathbf{u} + \partial_x \mathbf{f} = \mathbf{s} - \Delta_t \mathbf{u}, \quad (\text{II.4})$$

where

$$\Delta_t \mathbf{u} = \begin{bmatrix} \Delta_t u \\ 0 \end{bmatrix}, \quad (\text{II.5})$$

and $\Delta_t u$ is the physical time derivative discretized in time. The hyperbolic advection-diffusion system with $\partial_\tau \mathbf{u} = [\partial_\tau u, \partial_\tau p]$ is hyperbolic in the pseudo time τ , which guides us to construct an upwind numerical flux for the diffusive term [22,25]. Finally, after spacial discretization, we drop the pseudo time derivative, and solve the resulting global system of unsteady residual equations to obtain the solution at the next physical time level. The precise form of the unsteady residual depends on the time integration scheme as discussed in the next section.

III. Spatial Discretization: Third-Order Edge-Based Scheme

III.A. Discretization

The third-order edge-based discretization is defined by the residual at an interior node j on a one-dimensional grid $[x_0 = 0, x_1, x_2, \dots, x_N = 1]$ as

$$\mathbf{Res}_j = \mathbf{Res}_j^{\Delta x} - \mathbf{Res}_j^{\Delta t}, \quad (\text{III.1})$$

where $\mathbf{Res}_j^{\Delta x}$ is the spatial residual and $\mathbf{Res}_j^{\Delta t}$ is the physical time derivative discretized in space:

$$\mathbf{Res}_j^{\Delta x} = - \left(\frac{\Phi_{j+1/2} - \Phi_{j-1/2}}{\Delta x_j} \right) + \frac{1}{\Delta x_j} \int_{\Delta x_j} \mathbf{s} dx, \quad \mathbf{Res}_j^{\Delta t} = \frac{1}{\Delta x_j} \int_{\Delta x_j} \Delta_t \mathbf{u} dx, \quad j = 1, 2, \dots, N-1, \quad (\text{III.2})$$

Φ denotes a numerical flux and $\Delta x_j = (x_{j+1} - x_{j-1})/2$ is the dual control volume around the node j . The numerical flux Φ is a function of left and right states \mathbf{u}_L and \mathbf{u}_R :

$$\Phi(\mathbf{u}_L, \mathbf{u}_R) = \frac{1}{2} (\mathbf{f}_L + \mathbf{f}_R) - \frac{1}{2} \mathbf{Q}(\mathbf{u}_R - \mathbf{u}_L), \quad (\text{III.3})$$

where \mathbf{Q} is a dissipation coefficient (or a matrix for the hyperbolic scheme):

$$\mathbf{Q} = |a| + \frac{\alpha}{x_{j+1} - x_j}, \quad \text{for the linear advection-diffusion equation,} \quad (\text{III.4})$$

$$\mathbf{Q} = |u| + \frac{\alpha}{x_{j+1} - x_j}, \quad \text{for the viscous Burgers' equation,} \quad (\text{III.5})$$

$$\mathbf{Q} = \begin{bmatrix} |a| + \nu/L_r & 0 \\ 0 & \nu/L_r \end{bmatrix}, \quad \text{for the hyperbolic diffusion system.} \quad (\text{III.6})$$

The diffusive flux in Equations (III.4) and (III.5) is the alpha-damping scheme [26, 27, 28, 29] with $\alpha = 1$. The left and right states are computed by the linearly-extrapolated solutions and fluxes: for example at the face $j + 1/2$ between the nodes j and $j + 1$,

$$\mathbf{u}_L = \mathbf{u}_j + \frac{h_j}{2}(\partial_x \mathbf{u})_j, \quad \mathbf{u}_R = \mathbf{u}_{j+1} - \frac{h_j}{2}(\partial_x \mathbf{u})_{j+1}, \quad (\text{III.7})$$

$$\mathbf{f}_L = \mathbf{f}_j + \frac{h_j}{2} \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right)_j (\partial_x \mathbf{u})_j, \quad \mathbf{f}_R = \mathbf{f}_{j+1} - \frac{h_j}{2} \left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}} \right)_{j+1} (\partial_x \mathbf{u})_{j+1}, \quad (\text{III.8})$$

where $h_j = x_{j+1} - x_j$, and $(\partial_x \mathbf{u})_j$ denotes the solution gradient at j computed by a quadratic least-squares (LSQ) fit. The flux extrapolation is necessary to achieve third-order accuracy; the evaluation by the extrapolated solution, e.g., $\mathbf{f}_L = \mathbf{f}(\mathbf{u}_L)$, reduces the scheme to second-order accurate for nonlinear equations. In the hyperbolic scheme, the extra variable p replaces $\partial_x u$, and therefore the LSQ gradient is not needed for the primal variable, u ; this construction is called Scheme II in Refs.[25, 30]. Then, since the hyperbolic scheme achieves the same order of accuracy for the solution and the gradient, the accuracy of the gradient is equivalent to that of the quadratic LSQ gradient. As a result, the second-order hyperbolic scheme achieves third-order accuracy in the advection limit [25]. Further details on the discretization (LSQ methods, eigenvalue smoothing, weak boundary conditions, etc.), can be found in Ref.[23]. The source and physical time derivative terms are discretized by the compact formula, which was derived in Ref.[21]:

$$\int_{V_j} s dV = \sum_{k \in \{k_j\}} \frac{1}{2} (\mathbf{s}_L + \mathbf{s}_R) V_{jk}, \quad \mathbf{s}_L = \frac{7}{3} \mathbf{s}_j + \frac{1}{3} (x_k - x_j) \partial_x \mathbf{s}_j, \quad \mathbf{s}_R = -\frac{1}{3} \mathbf{s}_k, \quad (\text{III.9})$$

where V_j is the dual control volume $V_j = (x_{j+1} - x_{j-1})/2$, $\{k_j\} = \{j - 1, j + 1\}$, $V_{jk} = |x_k - x_j|/2$, and the derivative $\hat{\partial}_x \mathbf{s}_j$ is computed such that it is exact for quadratic functions, and similarly

$$\int_{\Delta x_j} \Delta_t \mathbf{u} dx = \sum_{k \in \{k_j\}} \frac{1}{2} [(\Delta_t \mathbf{u})_L + (\Delta_t \mathbf{u})_R] V_{jk}, \quad (\Delta_t \mathbf{u})_L = \frac{7}{3} (\Delta_t \mathbf{u})_j + \frac{1}{3} (x_k - x_j) \partial_x (\Delta_t \mathbf{u})_j, \quad (\Delta_t \mathbf{u})_R = -\frac{1}{3} (\Delta_t \mathbf{u})_k. \quad (\text{III.10})$$

As discussed in Ref.[21], the spatial discretizations of \mathbf{s} and $\Delta_t \mathbf{u}$ are required to preserve third-order spatial accuracy; there exist many accuracy-preserving source term quadrature formulas, and many of them require second-derivatives of \mathbf{s} and $\Delta_t \mathbf{u}$, including the one used in Ref.[20]. The formula (III.10) is a special formula derived in Ref.[21] that does not require second derivatives at all. It renders the third-order edge-based scheme completely free from second-derivatives on arbitrary simplex grids in all dimensions. The resulting pseudo-time system is written as

$$V_j \frac{d\mathbf{u}_j}{d\tau} = \mathbf{Res}_j, \quad (\text{III.11})$$

and by dropping the pseudo time derivative term, we obtain the unsteady residual:

$$0 = \mathbf{Res}_j, \quad (\text{III.12})$$

which we need to solve at each physical time step to obtain the solution at a new time level.

Note that the scheme reduces to second-order in space if a point evaluation is used for the source, a linear LSQ fit is used for the gradients, or the left and right fluxes evaluated by the extrapolated solutions: $\mathbf{f}_L = \mathbf{f}(\mathbf{u}_L)$ and $\mathbf{f}_R = \mathbf{f}(\mathbf{u}_R)$ for nonlinear equations. The third-order edge-based diffusion scheme in Ref.[20] may be employed to achieve third-order accuracy for the diffusion term. However, this particular approach requires a cubic LSQ fit and also high-order curved grids for curved geometries [31] in two and three dimensions. In this work, we focus only on methods that do not require high-order grids, and consider only the second-order scheme for diffusion in the scalar scheme. As a result, the third-order hyperbolic scheme is the only scheme that achieves third-order accuracy for both the advective and diffusive terms.

IV. Temporal Discretization: BDF and ESDIRK Schemes

At every node, we discretize the physical time derivative in time, resulting in a point value $(\Delta_t u)_j$. For the special source quadrature, the spatial derivative of $(\Delta_t u)_j$ is required. Below, we discuss how to evaluate these derivatives. In this study, we only consider implicit integration schemes, and therefore the spatial residual is evaluated by the solution at the next time level in all cases.

IV.A. BDF2

The BFD scheme discretizes the time derivative at a node j as

$$(\Delta_t \mathbf{u})_j = \alpha_p \mathbf{u}_j^{n+1} + \sum_{k=1}^l \alpha_{n-(k-1)} \mathbf{u}_j^{n-(k-1)}, \quad (\text{IV.1})$$

where n is a time level, u denotes the solution at the next time level $n+1$, and the coefficients α_p and $\{\alpha_l\}$ can be easily derived by fitting a polynomial of degree l over $l+1$ consecutive time steps [32]. In this work, we consider $l=1$ and 2 , i.e., the first-, second-order schemes, denoted respectively by BDF1 and BDF2. For variable time steps,

$$\Delta t_n = t^{n+1} - t^n, \quad \Delta t_{n-1} = t^n - t^{n-1}, \quad \Delta t_{n-2} = t^{n-1} - t^{n-2}, \quad (\text{IV.2})$$

BDF1 is given by

$$\alpha_p = -\alpha_n, \quad \alpha_n = -\frac{1}{\Delta t_n}, \quad \alpha_{n-1} = 0. \quad (\text{IV.3})$$

BDF2 is given by

$$\alpha_p = -(\alpha_n + \alpha_{n-1}), \quad \alpha_n = -\frac{\Delta t_{n-1} + 3\Delta t_n}{\Delta t_{n-1}(\Delta t_{n-1} + \Delta t_n)}, \quad \alpha_{n-1} = \frac{\Delta t_n}{\Delta t_{n-1}(\Delta t_{n-1} + \Delta t_n)}. \quad (\text{IV.4})$$

The BDF1 and BDF2 schemes are known to be L-stable. We do not consider the BDF3 scheme because it is not unconditionally stable. We consider only a constant time step Δt , but the BDF2 is not self-starting and needs to be started by the BDF1 scheme. The BDF1 scheme is used to start up the computation with $1/10$ of the actual time step Δt to minimize the effect of the lower-order accuracy. This is where the variable time-step formula is needed.

The BDF2 scheme is second-order accurate in time, but if it is used for the third-order edge-based scheme, the time derivative term must be discretized in space by the special source discretization to preserve third-order spatial accuracy. The special source discretization (III.10) requires the spatial gradient of the time-derivative $(\Delta_t u)_j$:

$$\partial_x (\Delta_t \mathbf{u})_j = \alpha_p \partial_x \mathbf{u}_j^{n+1} + \sum_{k=1}^l \alpha_{n-(k-1)} \partial_x \mathbf{u}_j^{n-(k-1)}, \quad (\text{IV.5})$$

and thus solution gradients at previous time levels need to be stored: two previous levels for BDF2 and one level for BDF1. In the case of the hyperbolic scheme, the solution gradients are already stored as the extra variable p , and therefore, no additional storage is required. As a result, both the scalar and hyperbolic schemes have the same storage requirement at previous time levels. This is true in all dimensions. For example, in 3D Navier-Stokes codes, a conventional approach requires the storage of the solution variables and their gradients

(5+15=20), and the hyperbolic approach requires the storage of the 20 solution variables, of which 15 correspond to the extra variables added to form a hyperbolic viscous system that are equivalent to the solution gradients.

The BDF2 scheme requires one nonlinear solve to advance the solution over one time step. The system of unsteady residual equations to be solved for \mathbf{u}^{n+1} is given by

$$\mathbf{Res}(\mathbf{u}^{n+1}) = \mathbf{Res}^{\Delta x}(\mathbf{u}^{n+1}) - \mathbf{Res}^{\Delta t}(\mathbf{u}^{n+1}, \mathbf{u}^n, \mathbf{u}^{n-1}) = 0. \quad (\text{IV.6})$$

IV.B. ESDIRK3

To achieve third-order accuracy in time, we employ the ESDIRK3 scheme [33] in the form:

$$\mathbf{u}_1 = \mathbf{u}^n, \quad (\text{IV.7})$$

$$\mathbf{Res}^{\Delta t}(\mathbf{u}_2, \mathbf{u}^n) = \sum_{k=1}^2 a_{2k} \mathbf{Res}^{\Delta x}(\mathbf{u}_k), \quad (\text{IV.8})$$

$$\mathbf{Res}^{\Delta t}(\mathbf{u}_3, \mathbf{u}^n) = \sum_{k=1}^3 a_{3k} \mathbf{Res}^{\Delta x}(\mathbf{u}_k), \quad (\text{IV.9})$$

$$\mathbf{Res}^{\Delta t}(\mathbf{u}^{n+1}, \mathbf{u}^n) = \sum_{k=1}^4 a_{4k} \mathbf{Res}^{\Delta x}(\mathbf{u}_k), \quad (\text{IV.10})$$

where the coefficients $\{a_{i,j}\}$ can be found in Ref.[33], and the j -th component of the temporal residual $\mathbf{Res}^{\Delta t}$ is defined by

$$\mathbf{Res}_j^{\Delta t}(\mathbf{v}, \mathbf{u}^n) = \frac{1}{\Delta x_j} \int_{\Delta x_j} \frac{\mathbf{v} - \mathbf{u}^n}{\Delta t} dx, \quad \mathbf{v} = \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}^{n+1}, \quad (\text{IV.11})$$

and the integral is evaluated by the special source discretization (III.10). The solution gradients are therefore required but only at the current and the previous level. The gradients need to be stored only at the previous time level for the scalar scheme; and no additional storage is required for the hyperbolic scheme. Moreover, the ESDIRK scheme is L-stable, and free from any explicit averaging process and thus suitable especially for the hyperbolic schemes. The ESDIRK scheme is self-starting, and thus the same constant time step can be used from the beginning and no special treatment at the first time step is required.

In contrast to the BDF2 scheme, the ESDIRK3 scheme requires three nonlinear solves. The three systems of unsteady residual equations are formed as follows:

$$\mathbf{Res}(\mathbf{u}_2) = \frac{1}{a_{22}} \mathbf{Res}^{\Delta t}(\mathbf{u}_2, \mathbf{u}^n) - \mathbf{Res}^{\Delta x}(\mathbf{u}_2) + \mathbf{G}_2 = 0, \quad \mathbf{G}_2 = -\frac{a_{21}}{a_{22}} \mathbf{Res}^{\Delta x}(\mathbf{u}_1), \quad (\text{IV.12})$$

$$\mathbf{Res}(\mathbf{u}_3) = \frac{1}{a_{33}} \mathbf{Res}^{\Delta t}(\mathbf{u}_3, \mathbf{u}^n) - \mathbf{Res}^{\Delta x}(\mathbf{u}_3) + \mathbf{G}_3 = 0, \quad \mathbf{G}_3 = -\sum_{k=1}^2 \frac{a_{3k}}{a_{33}} \mathbf{Res}^{\Delta x}(\mathbf{u}_k), \quad (\text{IV.13})$$

$$\mathbf{Res}(\mathbf{u}^{n+1}) = \frac{1}{a_{44}} \mathbf{Res}^{\Delta t}(\mathbf{u}^{n+1}, \mathbf{u}^n) - \mathbf{Res}^{\Delta x}(\mathbf{u}^{n+1}) + \mathbf{G}_4 = 0, \quad \mathbf{G}_4 = -\sum_{k=1}^3 \frac{a_{4k}}{a_{44}} \mathbf{Res}^{\Delta x}(\mathbf{u}_k). \quad (\text{IV.14})$$

These systems need to be solved, in order, for \mathbf{u}_2 , and \mathbf{u}_3 , and finally for \mathbf{u}^{n+1} .

IV.C. Implicit Solver

To solve each global system of unsteady residual equations,

$$\mathbf{Res}(\mathbf{v}) = 0, \quad (\text{IV.15})$$

where $\mathbf{v} = \mathbf{u}^{n+1}$ for the BDF2 scheme or the last stage of the ESDIRK3 scheme, or $\mathbf{v} = \mathbf{u}_2$ and $\mathbf{v} = \mathbf{u}_3$ for the second and third stages of the ESDIRK3 scheme, we employ the implicit defect-correction solver:

$$\mathbf{v}^{m+1} = \mathbf{v}^m + \Delta \mathbf{v}, \quad (\text{IV.16})$$

$$\frac{\partial \overline{\mathbf{Res}}}{\partial \mathbf{v}} \Delta \mathbf{v} = -\mathbf{Res}(\mathbf{v}^m), \quad (\text{IV.17})$$

where m is the iteration counter, and the Jacobian $\partial \overline{\mathbf{Res}} / \partial \mathbf{U}$ is the exact differentiation of the first-order version of the residual $\overline{\mathbf{Res}}$ (i.e., zero LSQ gradients). The linear system is relaxed to a specified tolerance by the Gauss-Seidel relaxation scheme. Note again that the pseudo-time term has been dropped and thus there is no CFL number to control.

V. Numerical Results

V.A. Accuracy Verification: Viscous Burgers

We consider a traveling wave solution of the viscous Burgers equation [34] with $\nu = 0.2$ and up to $t = 20$:

$$u(x, t) = u_R + \frac{1}{2}(u_L - u_R) \left[1 - \tanh \left(\frac{u_L - u_R}{4\nu} \left(x - \frac{u_L + u_R}{2} t \right) \right) \right], \quad (\text{V.1})$$

where $u_L = 1$ and $u_R = 0$. The time step is $\Delta t = 0.01$; it takes 2000 time steps to reach the final time $t = 20$. At each physical time step, the unsteady residuals are reduced by five orders of magnitude. The tolerance of five orders of magnitude residual reduction was found sufficient to observe the design order of accuracy for the schemes and grids used here. In practical applications, the unsteady-residual solver needs to be monitored and terminated by a suitable error control technique. In this problem, we compare four spatial discretization schemes: two scalar schemes and two hyperbolic schemes. A second-order scalar scheme uses the point source evaluation, linear LSQ gradients, and the solution extrapolation instead of flux; and the third-order scalar edge-based scheme, both of which use the second-order diffusion scheme. The second-order hyperbolic scheme uses the point source evaluation, linear LSQ gradients, and the flux extrapolation; and the third-order hyperbolic edge-based scheme. Note that the second-order hyperbolic scheme achieves third-order accuracy in the advective term. These schemes are combined with BDF2 and ESDIRK3, and compared.

Error convergence results are shown in Figure 1. As expected, only the third-order hyperbolic scheme achieves third-order accuracy in both the solution and the gradient, and only with the ESDIRK3 scheme. By comparing the BDF2 and ESDIRK3 schemes, we observe that the accuracy of the second-order scalar scheme is not affected by the time accuracy order; thus the BDF2 scheme is sufficient and the use of the third-order ESDIRK3 scheme for second-order schemes is a waste of resources for the time step $\Delta t = 0.01$. On the other hand, accuracy of third-order schemes is deteriorated by the BDF2 scheme on fine grids, whereas third-order accuracy is maintained with the ESDIRK3 scheme. Therefore, the third-order time integration for third-order schemes is important for highly accurate computations on refined grids. However, the error convergence results alone are not telling the whole story about the nature of the errors, i.e., dissipative and dispersive errors, which we investigate in the next section.

V.B. Sinusoidal Wave Packet: Linear Advection

To further investigate the behaviors of second- and third-order schemes, we consider a classic sinusoidal wave packet propagation problem. See Figure 2. The target equation is the linear advection equation with $a = 1$. Here, the second-order scalar advection scheme and the second-order hyperbolic scheme with $\nu = 10^{-12}$, which achieves third-order accuracy for advection dominated problems, are compared. In what follows, the conventional and hyperbolic schemes will be referred to as the second-order scheme and the third-order scheme, respectively. The domain is discretized with 479 nodes. The final time is $t = 2.0$ for all cases.

Figure 3 shows the results obtained for $\Delta t = 0.001$. As can be seen, all numerical solutions are heavily damped, and very inaccurate. However, the third-order scheme gives much more accurate solutions than the second-order scheme in terms of the phase error. This result is expected since it is the leading dispersive error in the second-order scheme, which is characterized by the third-order derivative $\partial_{xxx} u$ in the leading truncation error, that is removed by third-order schemes. It means also that leading dissipative errors, which are associated with a third-order truncation error proportional to $\partial_{xxxx} u$, are similar in second- and third-order schemes. Note that the time step is so small that the dispersive error is observed in the second-order scheme even with the third-order ESDIRK3 scheme.

To reduce the dissipation $\mathbf{Q}(\mathbf{u}_R - \mathbf{u}_L)$, there are at least two ways. One is to develop a higher-order reconstruction scheme that would reduce the jump, $\mathbf{u}_R - \mathbf{u}_L$ [35,36]. Note that any higher-order reconstruction does not improve the order of accuracy because the flux quadrature remains the central difference (and in

multi-dimensions, the flux quadrature remains the same edge-based formula with a single flux evaluation per edge); it merely reduces the jump. For the third-order edge-based scheme considered here, such a high-order reconstruction, if used in the flux extrapolation, is known to destroy third-order accuracy and the scheme reduces to second-order as demonstrated in Ref.[37]. The other approach is to reduce the dissipation coefficient \mathbf{Q} instead of the jump. In this study, we employ this strategy, which is simpler and more economical. For the model equation, we simply multiply the dissipation term by a small constant, K :

$$\mathbf{Q}^* = K\mathbf{Q}, \quad K = 0.01, \quad (\text{V.2})$$

where \mathbf{Q}^* is the reduced dissipation coefficient. The residual Jacobian is constructed by exactly differentiating the numerical flux with $K = 1$. A larger time step $\Delta t = 0.005$ is used here to illustrate the benefit of the high-order ESDIRK3 time-integration scheme. Results are shown in Figure 4. First of all, results with the BDF2 scheme are too dispersive and still dissipative for this time step size. See Figures 4(a) and 4(b). Note that second-order time integration schemes such as BDF2 has a leading local truncation error proportional to the third-order time-derivative $\partial_{ttt}u$ and we have $\partial_{ttt}u = a^3\partial_{xxx}u$ for the linear advection equation; therefore it introduces dispersive errors just like a second-order spatial discretization. It means that the use of the third-order edge-based scheme with BDF2 will be a waste of computational resources if the time step is so large that the time integration error dominates. Results obtained with the ESDIRK3 scheme, Figures 4(c) and 4(d), show that the second-order scheme still generates dispersive errors, and the third-order scheme gives an excellent result in terms of both dissipation and dispersion. These results indicate that a combination of the third-order scheme, the ESDIRK3 scheme, and a low-dissipation flux would be an effective and economical strategy for practical applications.

To investigate the performance on non-regular grids, we consider the same problem solved on an irregular grid, which is a perturbed version of the 479-node uniform grid used in the above, at $\Delta t = 0.0025$. The low-dissipation flux uses $K = 0.03$ for this problem. Results are shown in Figure 5. Unlike previous uniform-grid cases with a smaller Δt , quite different behaviors are observed for the BDF2 cases. Figures 5(a) and 5(b) show that the second-order scheme suffers from severe oscillations (dispersive errors) while the third-order scheme produces a smooth profile although a dispersive error due to the BDF2 scheme is observed. Therefore, the oscillations in the second-order scheme comes from the spatial discretization. This is confirmed also by the results for the ESDIRK3 scheme. See Figures 5(c) and 5(d). The second-order scheme again exhibits oscillations although the location of the wave packet is captured accurately due to ESDIRK3. On the other hand, the third-order scheme with ESDIRK3, which is third order accurate in space and time, produces a very accurate traveling wave packet solution.

The third-order scheme here is based on the hyperbolic formulation, and thus took a longer time (almost twice) to perform the calculations than the second-order scheme, for each time integration scheme. However, the third-order scheme based on the scalar equation with a quadratic fit (results not shown) took nearly the same CPU time as the second-order scheme. As expected, the ESDIRK versions took about three times more computing time than BDF2. However, such a comparison does not make sense for the results shown here because there is no point of obtaining worse solutions faster. For efficiency comparison, further studies are necessary to compare different schemes for a target accuracy, i.e., with different time steps or error control technique, different grid sizes, and etc.

V.C. Square Wave: Effect of Limiter

Effect of a limiter on second- and third-order schemes is investigated for a square-wave linear advection problem. The Van Albada limiter, as formulated in Ref.[38], was employed for its smooth property and popularity in practical codes. The advection speed is unity; the final time is 60, so that the wave travels at a distance of 60. The computational domain is taken to be $x \in [0, 25]$ with 127 nodes, but it is made periodic, and therefore there are no boundaries. The computation was performed with $\Delta t = 0.2$, and thus it takes 300 time steps for the wave to reach the final position. For this problem, two schemes were tested: the second-order scheme with BDF2, and the second-order hyperbolic advection-diffusion scheme with $\nu = 10^{-12}$ with ESDIRK3 as a third-order advection scheme. Without a limiter, the second-order scheme cannot well represent the square wave at the final time as shown in Figure 6(a). On the other hand, the third-order scheme well preserve the square profile with mild over and under shoots, demonstrating its low-dispersive nature. If the limiter is activated, the second-order result is further smoothed, apparently, due to the effect of first-order errors introduced by the limiter as shown in Figure 6(b). Therefore, no improvements are observed. It is conjectured that the limiter had to act extensively to deal with multiple extrema created by the dispersive nature of the second-order scheme. The third-order

limited scheme, on the other hand, produced an improved result with reduced over and under shoots. By the low-dispersion feature of the third-order scheme, there are only a small number of mild extrema created in the numerical solution. The limiter, therefore, needed to act only in that regions, and these regions are localized at the discontinuities. These results indicate that the third-order scheme can capture discontinuities with much less oscillations even without a limiter, and as a consequence, the solution is improved with a limiter that would work effectively in localized regions. Finally, it is pointed out that the limiting technique used here needs to be improved further to better bound the numerical solution with no new extrema at all.

VI. Three-Dimensional Viscous Flow Problem

Following the indication from the one-dimensional study, we consider a third-order low-dissipation scheme as an economical option for performing high-resolution viscous flow simulations. The third-order unsteady edge-based scheme with a low-dissipation inviscid flux has been implemented in NASA's FUN3D, and available to general users [17]. The low-dissipation flux is constructed based on the L²-Roe flux proposed in Ref.[39]. The L²-Roe flux uses Rieper's low-Mach fix [40], which eliminates the well-known low-Mach accuracy problem by limiting the normal velocity jump at a face:

$$\Delta u_n^* = z \Delta u_n, \quad z = \min(1, \max(M_L, M_R)), \quad (\text{VI.1})$$

where M_L and M_R are the Mach numbers based on the velocity magnitude at the left and right reconstructed states used for the flux evaluation. This is a very simple technique as one only needs to modify Δu_n in the dissipation term in an inviscid flux function. The factor z takes values smaller than 1 only for a local subsonic flow, and thus it recovers the original inviscid flux if the local Mach number is greater than or equal to 1. In effect, Rieper's technique reduces the dissipation, but Oßwald et al. [39] found when they applied it to the Roe flux that it was not sufficiently low for large-eddy simulations. To address the issue, they proposed to further reduce the dissipation by applying the factor z to all components of the velocity jump, instead of just the normal component:

$$\Delta \mathbf{v}^* = z \Delta \mathbf{v}, \quad (\text{VI.2})$$

where \mathbf{v} denotes the velocity vector. This modification has been shown to dramatically improve the resolution in a large-eddy-simulation [39] demonstrated its effectiveness for LES. We implemented the L²-Roe flux in FUN3D and performed a few tests, but encountered difficulties in iterative convergence of an implicit solver, which is an approximate Newton solver with an approximate Jacobian and a linear relaxation scheme applied to the linearized system. Note that an explicit time-stepping scheme is used in Ref.[39]. The implicit solver uses the Jacobian of the Roe flux, instead of the Jacobian of the L²-Roe flux, because the L²-Roe-flux Jacobian has reduced diagonal entries and leads to instability in the linear relaxation. Nevertheless, the implicit solver encounters difficulties in nonlinear convergence. Noting that the L²-Roe flux modifies the velocity jump but the dissipation matrix is evaluated by the original unmodified velocities, we found that the iterative convergence improved if the entire dissipation term (jumps and the matrix) was evaluated in a consistent manner. The idea is to use the Roe flux with the dissipation matrix is evaluated by a set of modified velocities such that the reduced jump (VI.2) is created automatically [41].

Our construction is based on a combination of the L²-Roe flux and the technique of Thornber et al. [41]. Thornber et al. proposed to modify the left and right velocities such that it creates $\Delta u_n = (\mathbf{v}_R - \mathbf{v}_L) \cdot \mathbf{n}$ by a factor of z but preserves the average $(\mathbf{v}_R + \mathbf{v}_L)/2$:

$$\mathbf{v}_L^* = \frac{\mathbf{v}_L + \mathbf{v}_R}{2} - \frac{z}{2}(\mathbf{v}_R - \mathbf{v}_L), \quad (\text{VI.3})$$

$$\mathbf{v}_R^* = \frac{\mathbf{v}_L + \mathbf{v}_R}{2} + \frac{z}{2}(\mathbf{v}_R - \mathbf{v}_L), \quad (\text{VI.4})$$

where z is defined as in Equation (VI.1), and then to compute a numerical flux with them. As discussed in Ref.[39], this method alters the averaged flux part of the Roe flux (i.e., it affects the consistency of the numerical flux), and therefore is not suitable if used as it is. Our proposal is to use these modified velocities to compute the dissipation term of the Roe flux in the original form. It produces the modified jump (VI.2) automatically, and the dissipation matrix is computed in a consistent manner. A similar low-dissipation inviscid flux had already been implemented in the commercial CFD software called scFlow [42] and demonstrated its high-resolution

feature for large-eddy-simulations. This technique is very simple to implement. Consider the Roe flux:

$$\frac{1}{2}(F_{nL} + F_{nR}) - \frac{1}{2}|\mathbf{A}_n|(\mathbf{U}_R - \mathbf{U}_L), \quad (\text{VI.5})$$

where F_{nL} and F_{nR} are the linearly extrapolated physical inviscid fluxes for third-order accuracy [13, 17], $|\mathbf{A}_n|$ is the absolute normal flux Jacobian evaluated at the Roe averages [43]. The first term is the averaged flux and the second term is the dissipation term, which is typically implemented in a characteristic form with jumps in the primitive variables and ambiguous tangent vectors eliminated [34]. Our proposal can be implemented as follows:

1. Compute the averaged flux with the original left and right states: \mathbf{U}_L and \mathbf{U}_R , and save it.
2. Modify the left and right velocities \mathbf{v}_L and \mathbf{v}_R as in Equations (VI.3) and (VI.4), respectively.
3. Compute the dissipation term $|\mathbf{A}_n|(\mathbf{U}_R - \mathbf{U}_L)$ with \mathbf{v}_L^* and \mathbf{v}_R^* .
4. Add the dissipation term to the averaged flux to complete the numerical flux.

Note that the only additional computation is the velocity modification in Step 2. In FUN3D, this numerical flux is designated as LDRoe, and can be activated with

```
flux_construction= "ldroe"
```

in the input file. Numerical experiments show that this version of the low-dissipation flux greatly improves the iterative convergence of the implicit solver. The third-order unsteady scheme can also be activated in FUN3D; it is expected to perform best with a Jacobian-Free Newton-Krylov solver that is currently under development (FUN3D uses only implicit time-stepping schemes).

A high-resolution capability of the low-dissipation flux has been demonstrated for an unsteady separated laminar flow over a cylinder [35]. The Mach number is 0.1, and the Reynolds number is 3900. The simulation was run with 20,000 time steps at $\Delta t = 0.05$ for the final time $t = 1000$. The time-integration was performed with the BDF2 scheme with an error control built in FUN3D. The cylinder of diameter 1.0 has a span-wise length of 2.0 and, the outer boundary is a rectangular box at the distance 100 from the center of the cylinder in all directions. The grid is a mixed grid with prisms around the cylinder and tetrahedra in the far field region as shown in Figure 7. The domain is taken as periodic at two planes with the minimum and maximum y -coordinates. The default FUN3D with the Roe flux and with the LDRoe flux are compared. Both use the default FUN3D viscous discretization, the Green-Gauss scheme, which is P_1 Galerkin on tetrahedral grids. The Q-isosurface (the vorticity magnitude) and the z -component of the velocity contours for the two cases are shown in Figure 8. As can be seen clearly, the LDRoe flux gives significantly higher resolution. These results indicate, in comparison with that in Ref.[35], where a high-order reconstruction technique is applied to the same cylinder test case, that higher-resolution can be achieved quite easily with the low-dissipation flux at almost no additional cost to the baseline scheme.

Another test was conducted with a pure tetrahedral grid, which was generated from the mixed grid by dividing the prisms into tetrahedra. For this case, we compared the default FUN3D and the third-order low-dissipation scheme. Again, the viscous scheme is the default P_1 Galerkin. As shown in Figure 9(a), the Q-isosurface is greatly diffused in the default FUN3D scheme. The third-order low-dissipation scheme produced a significantly better resolved solution as shown in Figure 9(b).

At the time of writing this paper, no statistical studies have been performed to quantitatively measure the improvements. Future work should investigate the quantitative comparison, also explore the application of a fully third-order and low-dissipation hyperbolic Navier-Stokes scheme with a stronger nonlinear solver and the ESDIRK3 time-integration scheme.

VII. Concluding Remarks

We have investigated the third-order edge-based scheme for various one-dimensional unsteady problems, and presented a preliminary result for a three-dimensional unsteady viscous flow. Numerical experiments show that a third-order time integration scheme, such as ESDIRK3, needs to be employed for the third-order edge-based scheme, otherwise third-order accuracy may not be obtained for a general time step size. If the second-order time integration error (e.g., BDF2) dominates, there will be no point of employing the third-order edge-based scheme. The study suggests that the third-order edge-based scheme with a low-dissipation flux would be an

economical and practical scheme for high-resolution simulations. The third-order scheme eliminates the leading dispersive error in the second-order scheme and the low-dissipation flux minimizes the dissipative error at almost no additional cost.

The third-order edge-based scheme has been implemented in NASA's FUN3D. A low-dissipation flux developed by combining two approaches in Refs.[39,41] has also been implemented in FUN3D. Numerical results shown for a three-dimensional unsteady viscous flow over a cylinder have demonstrated that the third-order low-dissipation flux is a very effective and economical approach to high-fidelity simulations, resolving the wake vortices with much higher resolution. The low-dissipation scheme in FUN3D is being used for various turbulent-flow applications such as landing-gear and rotorcraft simulations. Application to model-invariant RANS-LES computations can be found in Ref.[44].

Future work includes the development of a stronger solver based on the Jacobian-Free Newton-Krylov method with a variable preconditioner, a low-dissipation version of the hyperbolic viscous flux, and a compact implicit quadratic LSQ method. For NASA's FUN3D, the hyperbolic Navier-Stokes scheme needs to be implemented in the master code, and the ESDIRK3 scheme should be implemented also to fully benefit from the third-order edge-based scheme. The third-order hyperbolic Navier-Stokes scheme is capable of providing highly accurate derivative quantities (e.g., the viscous stresses and the heat fluxes) on irregular grids [13], and expected to play a vital role in high-fidelity turbulent simulations on adaptive grids.

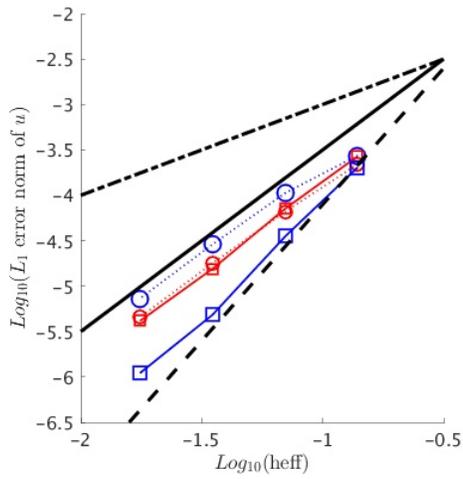
Acknowledgments

This work was funded by by NASA under Contract No. NNL09AA00A with Dr. Veer N. Vatsa as the technical monitor. The authors gratefully acknowledges support by the researchers of the FUN3D group, Computational Aerosciences Branch, NASA-LaRC., and Dr. Mujeeb Malik in the Revolutionary Computational Aerosciences (RCA) project within NASA's Transformational Tools and Technologies (TTT) Project of the Transformative Aeronautics Concepts Program under the Aeronautics Research Mission Directorate.

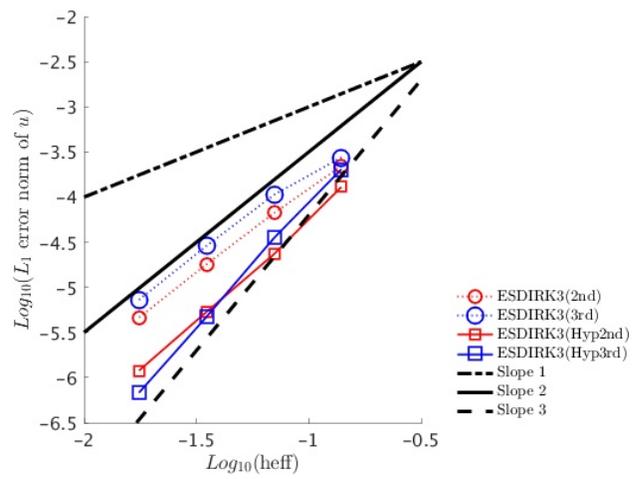
References

- ¹T. J. Barth. Numerical aspects of computing viscous high Reynolds number flows on unstructured meshes. AIAA Paper 91-0721, 1991.
- ²R. T. Biedron, J.-R. Carlson, J. M. Derlaga, P. A. Gnoffo, D. P. Hammond, W. T. Jones, B. Kleb, E. M. Lee-Rausch, E. J. Nielsen, M. A. Park, C. L. Rumsey, J. L. Thomas, and W. A. Wood. FUN3D manual: 12.9. *NASA-TM-2016-219012*, February 2016.
- ³Y. Nakashima, N. Watanabe, and H. Nishikawa. Development of an effective implicit solver for general-purpose unstructured CFD software. In *The 28th Computational Fluid Dynamics Symposium*, C08-1, Tokyo, Japan, 2014.
- ⁴N. Kroll, M. Abu-Zurayk, D. Dimitrov, T. Franz, T. Führer, T. Gerhold, S. Görtz, R. Heinrich, C. Ilic, J. Jepsen, J. Jägersküpper, M. Kruse, A. Krumbein, S. Langer, D. Liu, R. Liepelt, L. Reimer, M. Ritter, A. Schwöppe, J. Scherer, F. Spiering, R. Thormann, V. Togiti, D. Vollmer, and J.-H. Wendisch. Dlr project digital-x: Towards virtual aircraft design and flight testing based on high-fidelity methods. *CEAS Aeronautical Journal*, 7(1):3–27, March 2016.
- ⁵D. J. Mavriplis and M. Long. NSU3D Results for the Fourth AIAA Drag Prediction Workshop. AIAA Paper 2010-4363, 2010.
- ⁶H. Luo, J. D. Baum, and R. Löhner. High-reynolds number viscous flow computations using an unstructured-grid method. In *Proc. of 42nd AIAA Aerospace Sciences Meeting*, AIAA Paper 2004-1103, Reno, Nevada, 2004.
- ⁷T. Kozubskaya, I. Abalakin, A. Dervieux, and H. Ouvrard. Accuracy improvement for finite-volume vertex-centered schemes solving aeroacoustics problems on unstructured meshes. In *Proc. of 16th AIAA/CEAS Aeroacoustics Conference*, AIAA Paper 2010-3933, 2010.
- ⁸A. Haselbacher and J. Blazek. Accurate and efficient discretization of Navier-Stokes equations on mixed grids. *AIAA J.*, 38(11):2094–2102, 2000.
- ⁹A. Katz and V. Sankaran. Mesh quality effects on the accuracy of CFD solutions on unstructured meshes. *J. Comput. Phys.*, 230:7670–7686, 2011.
- ¹⁰A. Katz and V. Sankaran. An efficient correction method to obtain a formally third-order accurate flow solver for node-centered unstructured grids. *J. Sci. Comput.*, 51:375–393, 2012.
- ¹¹B. Diskin and J. L. Thomas. Effects of mesh regularity on accuracy of finite-volume schemes. In *Proc. of 50th AIAA Aerospace Sciences Meeting*, AIAA Paper 2012-0609, Nashville, Tennessee, 2012.
- ¹²H. Nishikawa. Accuracy-preserving boundary flux quadrature for finite-volume discretization on unstructured grids. *J. Comput. Phys.*, 281:518–555, 2015.
- ¹³Yi Liu and Hiroaki Nishikawa. Third-order inviscid and second-order hyperbolic Navier-Stokes solvers for three-dimensional inviscid and viscous flows. In *46th AIAA Fluid Dynamics Conference*, AIAA Paper 2016-3969, Washington, D.C., 2016.
- ¹⁴O. Tong, A. Katz, Y. Yanagita, A. Casey, and R. Schaap. High-order methods for turbulent flows on three-dimensional strand grids. *J. Sci. Comput.*, pages 1–19, 2015.
- ¹⁵A. Katz and Dalon Work. High-order flux correction/finite difference schemes for strand grids. *J. Comput. Phys.*, 282:360–380, 2015.

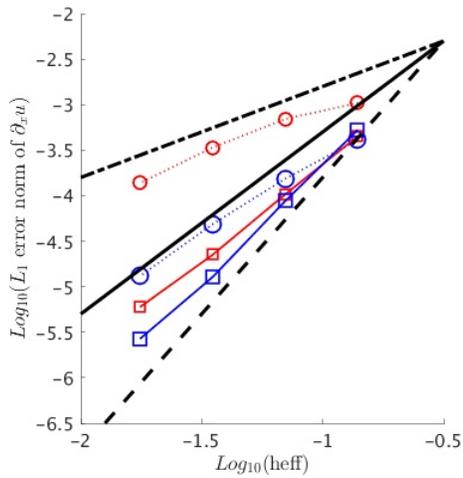
- ¹⁶Y. Nakashima, N. Watanabe, and H. Nishikawa. Hyperbolic Navier-Stokes solver for three-dimensional flows. In *54th AIAA Aerospace Sciences Meeting*, AIAA Paper 2016-1101, San Diego, CA, 2016.
- ¹⁷Yi Liu and Hiroaki Nishikawa. Third-order inviscid and second-order hyperbolic Navier-Stokes solvers for three-dimensional unsteady inviscid and viscous flows. In *55th AIAA Aerospace Sciences Meeting*, AIAA Paper 2017-0738, Grapevine, Texas, 2017.
- ¹⁸H. Nishikawa and Yi Liu. Hyperbolic Navier-Stokes method for high-Reynolds-number boundary-layer flows. In *55th AIAA Aerospace Sciences Meeting*, AIAA Paper 2017-0081, Grapevine, Texas, 2017.
- ¹⁹Yi Liu and Hiroaki Nishikawa. Third-order edge-based hyperbolic Navier-Stokes scheme for three-dimensional viscous flows. In *23rd AIAA Computational Fluid Dynamics Conference*, AIAA Paper 2017-3443, Denver, Colorado, 2017.
- ²⁰B. Pincock and A. Katz. High-order flux correction for viscous flows on arbitrary unstructured grids. *J. Sci. Comput.*, 61:454–476, 2014.
- ²¹H. Nishikawa and Y. Liu. Accuracy-preserving source term quadrature for third-order edge-based discretization. *J. Comput. Phys.*, 344:595–622, 2017.
- ²²H. Nishikawa. A first-order system approach for diffusion equation. II: Unification of advection and diffusion. *J. Comput. Phys.*, 229:3989–4016, 2010.
- ²³H. Nishikawa and Y. Liu. Hyperbolic advection-diffusion schemes for high-Reynolds-number boundary-layer problems. *J. Comput. Phys.*, 352:23–51, 2018.
- ²⁴H. Nishikawa and Y. Nakashima. Dimensional scaling and numerical similarity in hyperbolic method for diffusion. *J. Comput. Phys.*, 355:121–143, 2018.
- ²⁵H. Nishikawa. First, second, and third order finite-volume schemes for advection-diffusion. *J. Comput. Phys.*, 273:287–309, 2014.
- ²⁶H. Nishikawa, Y. Nakashima, and N. Watanabe. Effects of high-frequency damping on iterative convergence of implicit viscous solver. *J. Comput. Phys.*, 348:66–81, 2017.
- ²⁷H. Nishikawa. Robust and accurate viscous discretization via upwind scheme - I: Basic principle. *Comput. Fluids*, 49(1):62–86, October 2011.
- ²⁸H. Nishikawa. Beyond interface gradient: A general principle for constructing diffusion schemes. In *Proc. of 40th AIAA Fluid Dynamics Conference and Exhibit*, AIAA Paper 2010-5093, Chicago, 2010.
- ²⁹H. Nishikawa. Two ways to extend diffusion schemes to Navier-Stokes schemes: Gradient formula or upwinding. In *20th AIAA Computational Fluid Dynamics Conference*, AIAA Paper 2011-3044, Honolulu, Hawaii, 2011.
- ³⁰H. Nishikawa. First-, second-, and third-order finite-volume schemes for diffusion. *J. Comput. Phys.*, 256:791–805, 2014.
- ³¹D. G. Work and A. Katz. Aspects of the flux correction method for solving the Navier-Stokes equations on unstructured meshes. In *Proc. of 53rd AIAA Aerospace Sciences Meeting*, AIAA Paper 2015-0834, Kissimmee, Florida, January 2015.
- ³²Veer Vatsa and Mark Carpenter. Higher-order temporal schemes with error controllers for unsteady Navier-Stokes equations. In *Proc. of 17th AIAA Computational Fluid Dynamics Conference*, AIAA Paper 2005-5245, Toronto, 2005.
- ³³C. A. Kennedy and M. H. Carpenter. Additive runge-kutta schemes for convection-diffusion-reaction equations. *Appl. Numer. Math.*, 44:139–181, 2003.
- ³⁴Katate Masatsuka. I do like CFD, VOL.1, Second Edition, version 2.6. <http://www.cfdbooks.com>, 2018.
- ³⁵H. Q. Yang and Robert E. Harris. Development of vertex-centered high-order schemes and implementation in fun3d. *AIAA J.*, 54:3742–3760, 2016.
- ³⁶I. Abalakin, T. Kozubskaya, and A. Dervieux. High accuracy finite volume method for solving nonlinear aeroacoustics problems on unstructured meshes. *Chinese Journal of Aeroacoustics*, 19(2):97–104, 2006.
- ³⁷H. Nishikawa. Alternative formulations for first-, second-, and third-order hyperbolic Navier-Stokes schemes. In *Proc. of 22nd AIAA Computational Fluid Dynamics Conference*, AIAA Paper 2015-2451, Dallas, TX, 2015.
- ³⁸V. Venkatakrishnan. Convergence to steady state solutions of the euler equations on unstructured grids with limiters. *J. Comput. Phys.*, 118:120–130, 1995.
- ³⁹K. Oßwald, A. Siegmund, P. Birken, V. Hannemann, and A. Meister. L^2 Roe: a low dissipation version of Roes approximate riemann solver for low Mach numbers. *Int. J. Numer. Meth. Fluids*, 81:71–86, 2016.
- ⁴⁰F. Rieber. A low-mach number fix for Roes approximate riemann solver. *J. Comput. Phys.*, 230:5263–5287, 2011.
- ⁴¹B. Thornber, A. Mosedale, D. Drikakis, D. Youngs, and R. J. R. Williams. An improved reconstruction method for compressible flows with low mach number features. *J. Comput. Phys.*, 227:4873–4894, 2008.
- ⁴²scFlow, Software Cradle, MSC Software Company. <http://www.cradle-cfd.com/products/scflow>, visited on 06/14/2018.
- ⁴³P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.*, 43:357–372, 1981.
- ⁴⁴Sreekanth Ravindran and Stephen Woodruff. Model-invariant hybrid RANS-LES computations on unstructured meshes. In *48th AIAA AIAA Fluid Dynamics Conference*, AIAA Paper, Atlanta, GA, 2018. to be published as AIAA paper.



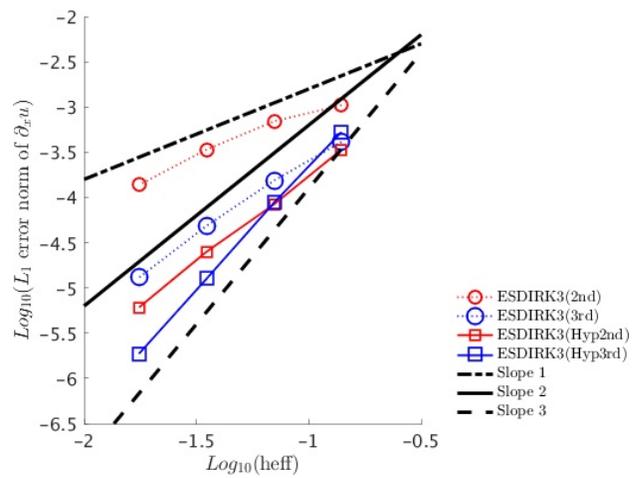
(a) Error in u with BDF2.



(b) Error in u with ESDIRK3.



(c) Error in $\partial_x u$ with BDF2.



(d) Error in $\partial_x u$ with ESDIRK3.

Figure 1: Error convergence results for the viscous Burgers case. Time step is $\Delta t = 0.01$; it takes 2000 time steps to reach the final time $t = 20$.

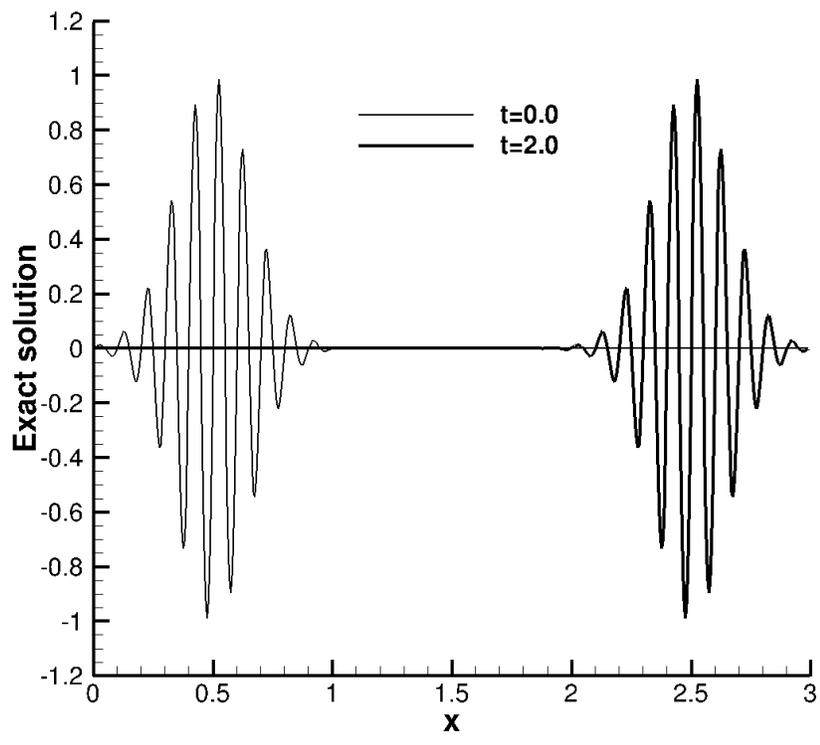
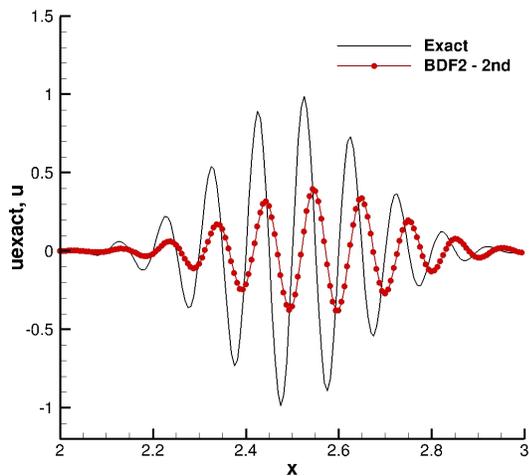
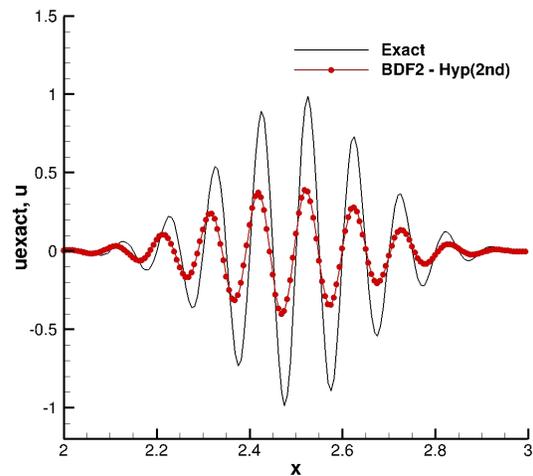


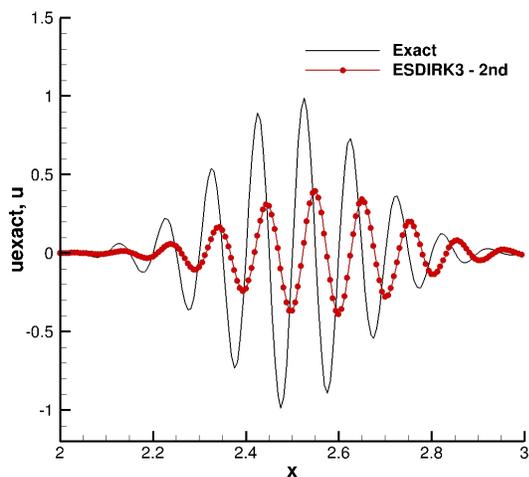
Figure 2: Sinusoidal wave packet propagation.



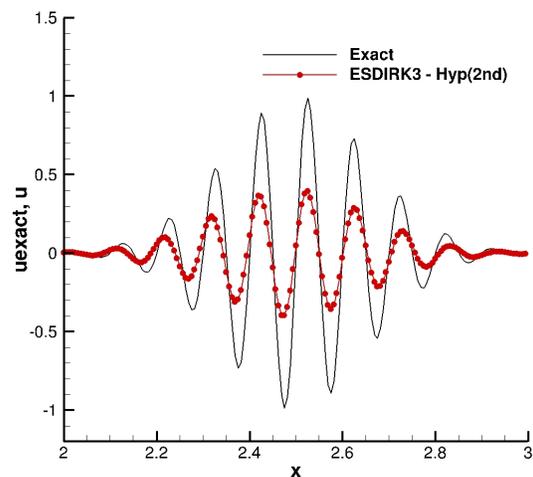
(a) Second-order scheme with BDF2.



(b) Third-order scheme with BDF2.

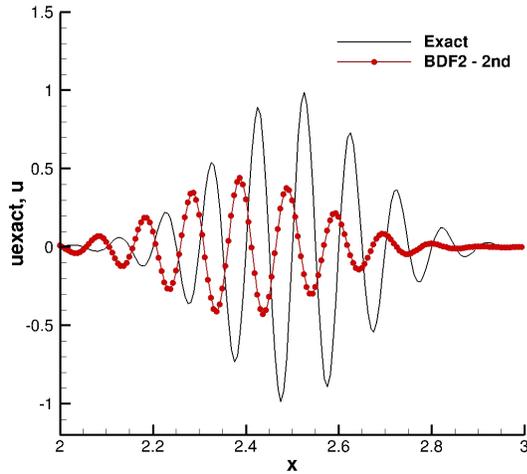


(c) Second-order scheme with ESDIRK3.

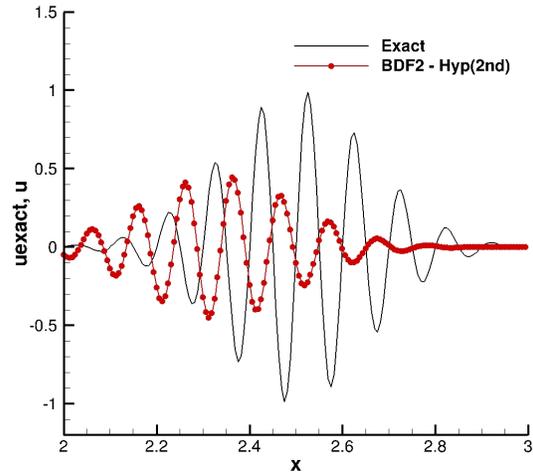


(d) Third-order scheme with ESDIRK3.

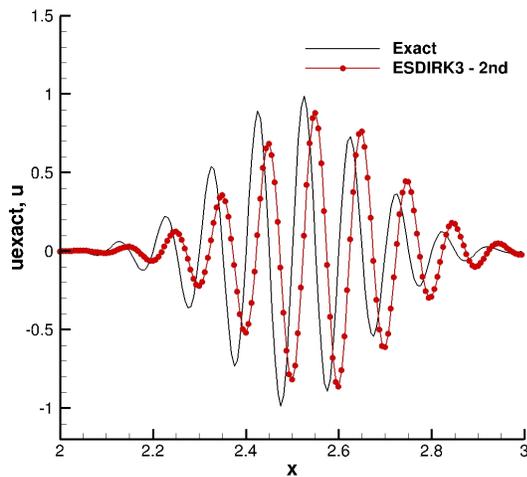
Figure 3: Upwind second- and third-order schemes for the sinusoidal wave-packet propagation on a regular grid with $\Delta t = 0.001$ at the final time $t = 2.0$. Hyp(2nd) is the second-order hyperbolic advection-diffusion scheme, which achieves third-order accuracy without quadratic LSQ gradients for advection problems.



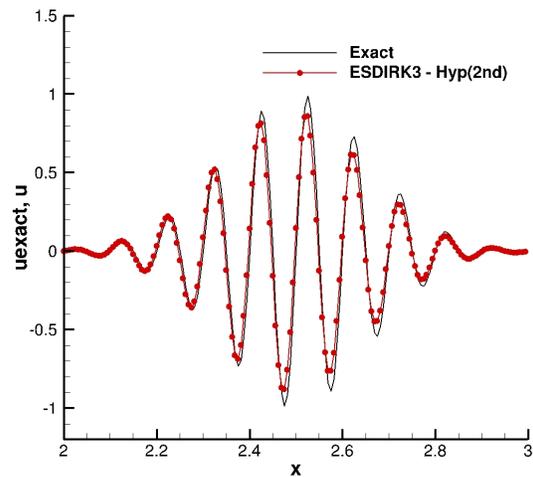
(a) Second-order scheme with BDF2.



(b) Third-order scheme with BDF2.

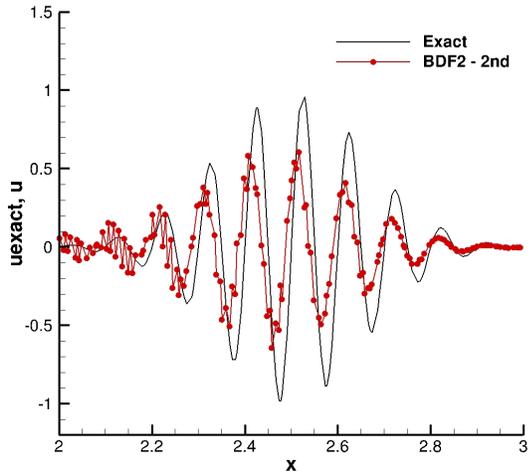


(c) Second-order scheme with ESDIRK3.

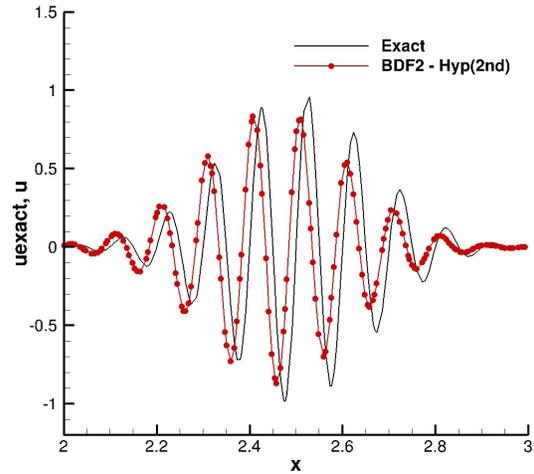


(d) Third-order scheme with ESDIRK3.

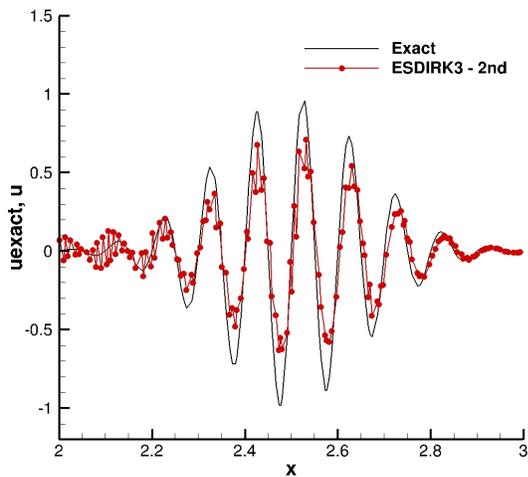
Figure 4: Low-dissipation second- and third-order schemes for the sinusoidal wave-packet propagation on a regular grid with $\Delta t = 0.005$ at the final time $t = 2.0$. Hyp(2nd) is the second-order hyperbolic advection-diffusion scheme, which achieves third-order accuracy without quadratic LSQ gradients for advection problems.



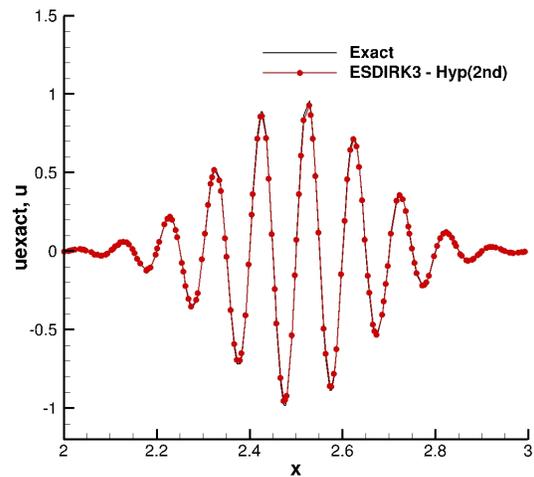
(a) Second-order scheme with BDF2.



(b) Third-order scheme with BDF2.

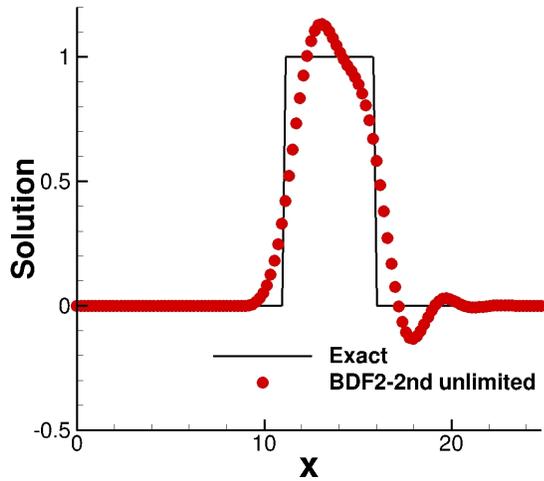


(c) Second-order scheme with ESDIRK3.

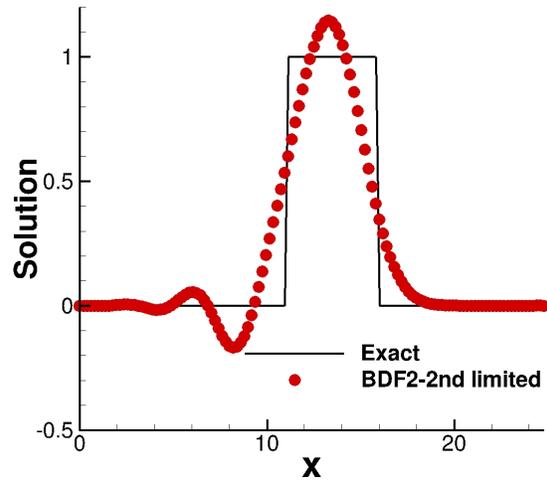


(d) Third-order scheme with ESDIRK3.

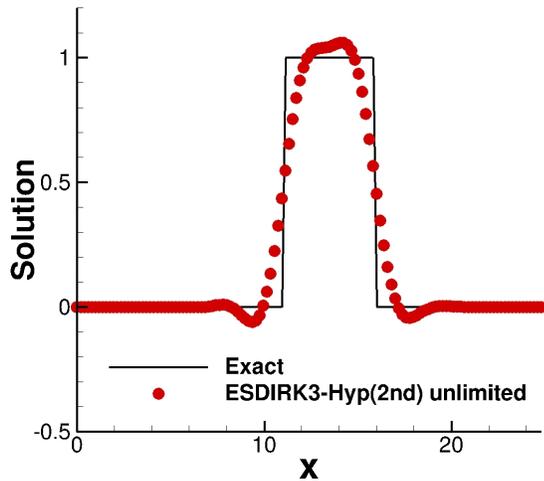
Figure 5: Low-dissipation second- and third-order schemes for the sinusoidal wave-packet propagation on an irregular grid with $\Delta t = 0.0025$ at the final time $t = 2.0$. Hyp(2nd) is the second-order hyperbolic advection-diffusion scheme, which achieves third-order accuracy without quadratic LSQ gradients for advection problems.



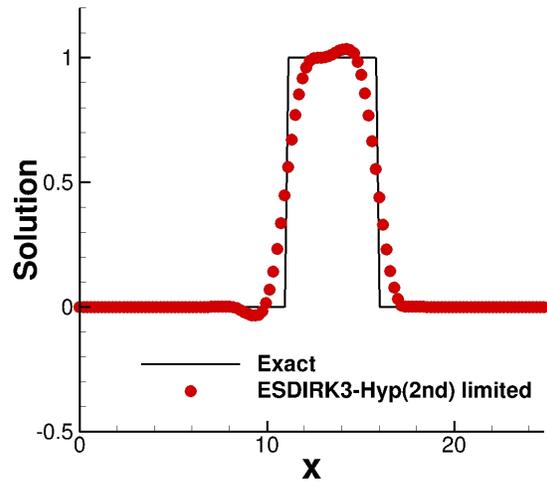
(a) Second-order scheme without a limiter.



(b) Second-order scheme with a limiter.

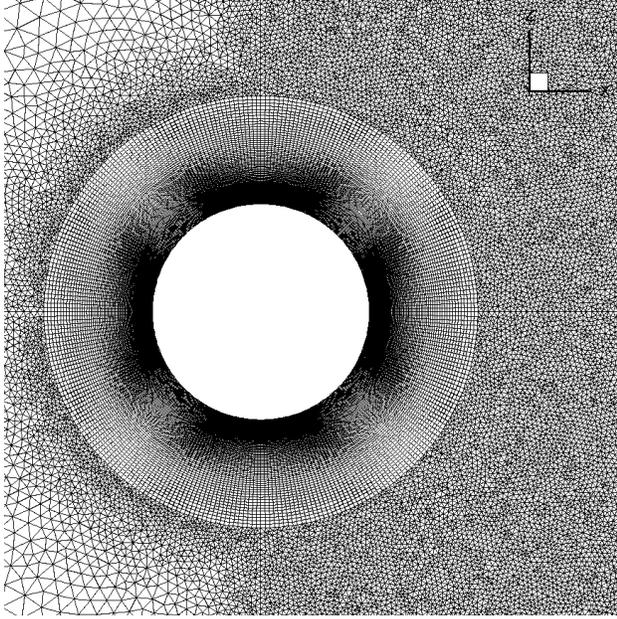


(c) Third-order scheme without a limiter.

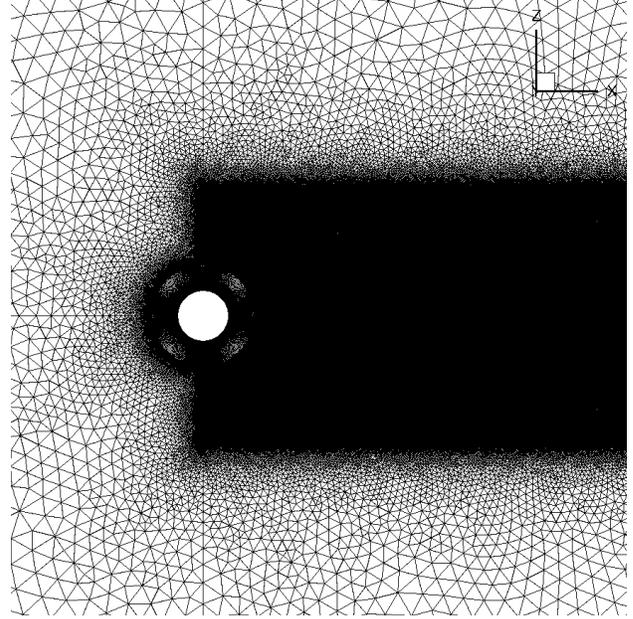


(d) Third-order scheme with a limiter.

Figure 6: Second- and third-order results for the square-wave linear convection problem at the unit convection speed on a uniform grid with $\Delta t = 0.2$ at the final time $t = 60.0$ with periodic boundary conditions. The distance that the wave traveled is 60. The second-order hyperbolic scheme is a third-order accurate without quadratic LSQ gradients for advection problems.

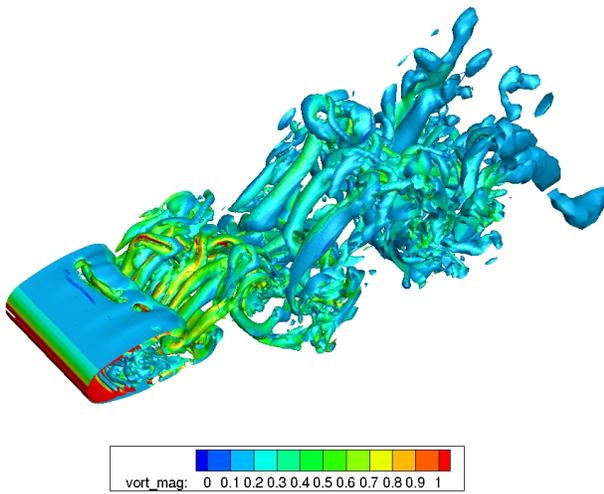


(a) Close view showing the prismatic region around the cylinder.

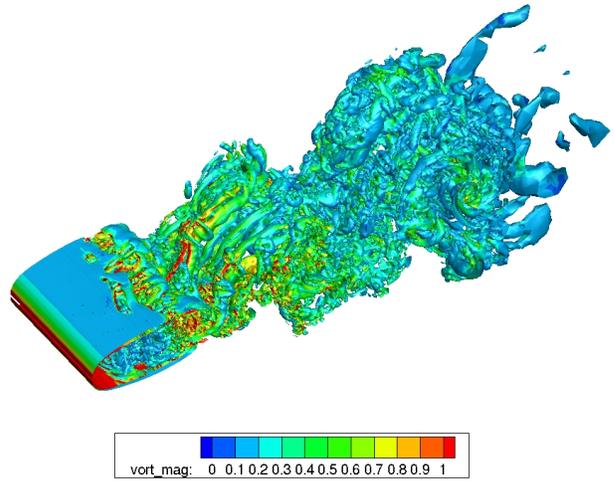


(b) Close view showing a highly-refined grid in the wake region.

Figure 7: Close section views of the three-dimensional mixed grid used for the cylinder flow simulations. Prisms around the cylinder and tetrahedra at a far field.

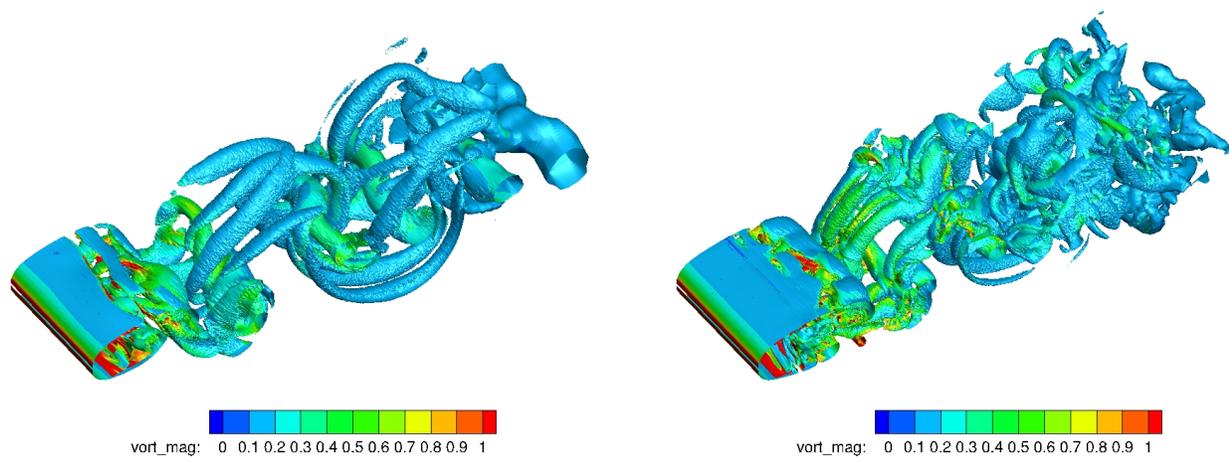


(a) FUN3D-Roe



(b) FUN3D-LDRoe

Figure 8: Q-isosurface comparison between the default FUN3D with the Roe flux and the low-dissipation LDRoe flux for a mixed grid.



(a) FUN3D-Roe

(b) FUN3D-i3rd-LDRoe

Figure 9: Q-isosurface comparison between the default FUN3D with the Roe flux and the third-order low-dissipation LDRoe scheme for a tetrahedral grid.