High–Order Hyperbolic Residual–Distribution Schemes on Arbitrary Triangular Grids

Alireza Mazaheri*

NASA Langley Research Center, Hampton, VA 23681

Hiroaki Nishikawa[†]

National Institute of Aerospace, Hampton, VA 23666

In this paper, we construct high–order hyperbolic residual–distribution schemes for general advection–diffusion problems on arbitrary triangular grids. We demonstrate that the second–order accuracy of the hyperbolic schemes can be greatly improved by requiring the scheme to preserve exact quadratic solutions. We also show that the improved second–order scheme can be easily extended to third–order by further requiring the exactness for cubic solutions. We construct these schemes based on the LDA and the SUPG methodology formulated in the framework of the residual–distribution method. For both second– and third-order–schemes, we construct a fully implicit solver by the exact residual Jacobian of the second–order scheme, and demonstrate rapid convergence of 10–15 iterations to reduce the residuals by 10 orders of magnitude. We demonstrate also that these schemes can be constructed based on a separate treatment of the advective and diffusive terms, which paves the way for the construction of hyperbolic residual–distribution schemes for the compressible Navier–Stokes equations. Numerical results show that these schemes produce exceptionally accurate and smooth solution gradients on highly skewed and anisotropic triangular grids, including curved boundary problems, using linear elements. We also present Fourier analysis performed on the constructed linear system and show that an under–relaxation parameter is needed for stabilization of Gauss-Seidel relaxation.

I.   Introduction

In many flow simulations, accurate prediction of solution gradients, such as velocity and temperature gradients, are essential for design and analysis purposes as they are directly related to the physical quantities of interest: the viscous stresses, the vorticity, and the heat fluxes. However, it is widely accepted that accurate and smooth solution gradients cannot be achieved with conventional schemes on fully irregular unstructured grids.[1,2] In conventional schemes, the gradients are obtained typically with a lower order of accuracy (typically through reconstruction) and they are also subject to numerical oscillations on such grids. The resolution of this issue is very important for justifying the use of high–fidelity models in engineering design, analysis, and optimization, especially for applications involving complex geometries. The ability to predict the gradients on irregular grids is even more critical for grid adaptation, a vital technique for efficient CFD calculations in high–order methods,[3] because the grid adaptation almost necessarily introduces irregularity in the grid. In fact, current practices in grid adaptation often avoid adaptation in certain regions such as within boundary layers where grid irregularity has a severe impact on the solution quality.[4] Therefore, numerical schemes that can accurately predict solution gradients on irregular grids need to be developed, so that the power of grid adaptation can be fully exploited.

A method that enables the construction of such schemes is the first–order hyperbolic system method, or the hyperbolic method for short, which was proposed in 2007 for diffusion.[5] In the hyperbolic method, which is fundamentally different from conventional methods, the solution and the solution gradients are simultaneously computed by solving a hyperbolic system for diffusion. The hyperbolic method was then studied

---

*Research Aerospace Engineer, Aerothermodynamics Branch, M/S 408A, Ali.R.Mazaheri@NASA.gov.

[†]Associate Research Fellow, 110 Exploration Way, Hampton, VA 23666.

for advection–diffusion in Ref. 6 with Residual–Distribution (RD) schemes. Later, the method was demonstrated for the compressible Navier-Stokes equations by a second–order Finite–volume (FV) scheme.[7] Since then, there have been efforts in developing high–order hyperbolic schemes in the finite–volume method,[8–10] in the active flux method,[11] and in the RD method[12,13] for unsteady computations.

In this paper, we focus on the development of hyperbolic RD schemes for two–dimensional problems, extending the previous work,[5,6] with several important advances. Improved Accuracy: We propose to construct a second–order scheme such that it preserves exact quadratic solutions, which can be accomplished by the curvature correction technique.[14] The resulting scheme remains compact, and produces significantly improved solution gradients over the previous schemes, which do not preserve exact quadratic solutions. Third–Order Accuracy: Extending the improved second–order scheme, we construct a third–order scheme that preserves exact cubic solutions. The construction requires quadratic least-squares (LSQ) gradients and a high–order source term discretization. Nonlinear Equation: The improved schemes are extended to a nonlinear advection–diffusion equation by the preconditioned conservative formulation introduced in Ref. 7. High–Order on Linear Elements: We demonstrate that the third–order scheme does not require curved elements for curved boundary problems; it gives more accurate solution and gradients than the second–order scheme on the same linear grids. This is a significant advantage, because most high–order methods require curved geometries to be represented by high–order curved elements; see Ref. 3. Non–Unified Approach: Instead of the fully integrated approach of discretizing the hyperbolic advection–diffusion system as in Ref. 6, we discretize the advective and diffusive terms separately. This approach will enable the extension to the compressible Navier–Stokes equations for which the eigenstructure of the whole system has not been discovered yet. Fully Implicit Solver: We construct a fully implicit solver for both second– and third–order schemes. For practical applications, explicit iterations considered in Refs. 5,6 are not efficient enough, and a fully implicit solver is needed. The implicit solver is constructed by the exact residual Jacobian of the second–order scheme. It converges in at most 10 iterations to reduce the residual by 10 orders of magnitude for both second– and third–order schemes. We demonstrate these features for a series of test problems involving fully irregular isotropic and anisotropic triangular grids and curved boundaries. This work serves as a basis for the development of high–order multidimensional hyperbolic RD schemes for more complex equations such as the Navier–Stokes equations. The extension of the proposed RD schemes to Navier–Stokes equations and problems with shocks and discontinuities will be addressed in subsequent papers.

The paper is organized as follows. In the next section, we describe the basics of the hyperbolic RD scheme: residual evaluations, boundary conditions, and an implicit solver. In Section III, we describe the construction of the LDA and the SUPG distribution matrices based on a non–unified approach, in which the advective and diffusive terms are treated independently. In Section IV, we discuss the accuracy issue of the previous schemes, and propose a guiding principle for constructing improved schemes. A new second–order scheme is presented in Section V, followed by the extension to third–order in Section VI. The extension to nonlinear advection–diffusion equations is explained in Section VII. Numerical results are then presented in Section VIII for both linear and nonlinear problems, followed by concluding remarks in Section IX. For completeness, Fourier analysis of the constructed linear system is also presented in Appendix A.

II.   Baseline RD scheme for advection–diffusion

This baseline scheme is the basis for the development of improved second–order and third–order schemes discussed later. For the purpose of our discussion and simplicity, we first describe the details for the linear advection–diffusion equation. Extension to the nonlinear equation is discussed in Sec. VII.

A.   Hyperbolic advection-diffusion system

Consider a two–dimensional advection–diffusion equation:

$$\partial_t u + a\,\partial_x u + b\,\partial_y u = \nu\,(\partial_{xx} u + \partial_{yy} u) + \tilde{s}(x, y, u), \tag{1}$$

where a and b are constant advection speeds, respectively, in x and y directions, $\nu\ (>0)$ is the diffusion coefficient, and $\tilde{s}(x, y, u)$ is a source term. Following Refs.,[6,12] we rewrite the above equation as a first–order

hyperbolic advection–diffusion system:

$$\partial_\tau u + a\,\partial_x u + b\,\partial_y u \;=\; \nu\,(\partial_x p + \partial_y q) + s(x,y,u), \tag{2}$$

$$\partial_\tau p \;=\; (\partial_x u - p)/T_r, \tag{3}$$

$$\partial_\tau q \;=\; (\partial_y u - q)/T_r, \tag{4}$$

where τ is a pseudo time, $T_r > 0$ is the relaxation time, and $s(x,y,u)$ is a sum of $\tilde{s}(x,y,u)$ and a physical time–derivative term discretized by an implicit time–stepping scheme (see Refs.[12,13] for more details). In the vector form, the system is written as

$$\frac{\partial \mathbf{U}}{\partial \tau} + \mathbf{A}\frac{\partial \mathbf{U}}{\partial x} + \mathbf{B}\frac{\partial \mathbf{U}}{\partial y} = \mathbf{Q}, \tag{5}$$

where

$$\mathbf{U} = \begin{bmatrix} u \\ p \\ q \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} a & -\nu & 0 \\ -1/T_r & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} b & 0 & -\nu \\ 0 & 0 & 0 \\ -1/T_r & 0 & 0 \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} s \\ -p/T_r \\ -q/T_r \end{bmatrix}. \tag{6}$$

The system is hyperbolic in the pseudo time as shown in Refs. 6. Towards the pseudo steady state, the variables p and q approach the solution gradients u_x and u_y and hence the above equation becomes identical to the original advection–diffusion equation with the physical time–derivative discretized by an implicit time–stepping scheme (see Ref.[12]). This is true for any T_r, and thus T_r is determined not by physical constraints but by optimal steady convergence criteria,[6] leading to a non–stiff hyperbolic formulation for diffusion. Note that simply dropping the pseudo time derivative will also recover the original equation. In this paper, we focus on steady problems, but the resulting steady schemes can be made time accurate by including a physical time derivative term in the source term, $s(x,y,u)$, as described in details in Refs.[12,13] In either case, an efficient steady solver is required, and its development is one of the objectives of the present paper.

B.   RD scheme: cell–residual, distribution matrix, nodal residual

We discretize the hyperbolic advection–diffusion system on unstructured triangular grids. The domain is divided into a set $\{E\}$ of arbitrary triangular cells (or elements), and an associated set $\{J\}$ of nodes (or vertices). The total number of nodes is denoted by N. We store the solutions (u_j, p_j, q_j) at each node $j \in \{J\}$. In the RD method, we first evaluate the residuals over the elements and then distribute the residuals to the nodes with a distribution matrix. The cell–residual over an element $E \in \{E\}$ is defined as
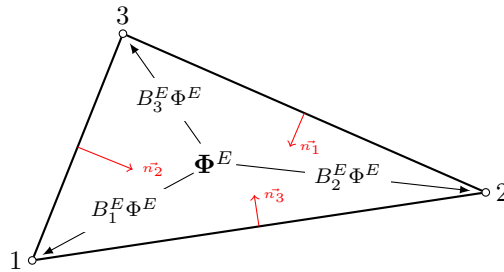

Figure 1.   Schematic of the residual distribution to local nodes and definition of the inward unit normals (not-to-scale).

$$\boldsymbol{\Phi}^E = \begin{bmatrix} \Phi_u^E \\ \Phi_p^E \\ \Phi_q^E \end{bmatrix} = \int_E (-\mathbf{A}\mathbf{U}_x - \mathbf{B}\mathbf{U}_y + \mathbf{Q})\, dx\, dy. \tag{7}$$

American Institute of Aeronautics and Astronautics

We assume a piecewise linear variation of $\mathbf{U}$ (i.e. $c_0 + c_x(x - x_c) + c_y(y - y_c)$) over the element, which interpolates the three nodal solutions, and perform the integration to obtain

$$\mathbf{\Phi}^E = -\sum_{i=1}^{3} \mathbf{K}_i \mathbf{U}_i + \overline{\mathbf{Q}}^E d\Omega^E, \tag{8}$$

where i is the local vertex counter (i.e., $i = 1, 2, 3$) for the element E, $d\Omega^E$ is the element area, and

$$\mathbf{K}_i = \frac{1}{2}(\mathbf{A}\,\hat{n}_{i_x} + \mathbf{B}\,\hat{n}_{i_y})|\mathbf{n}_i| = \frac{1}{2}\mathbf{A}_{n_i}|\mathbf{n}_i|, \quad \overline{\mathbf{Q}}^E = \frac{1}{3}\sum_{i=1}^{3} \mathbf{Q}_i. \tag{9}$$

Note that $\mathbf{n}_i = (n_{i_x}, n_{i_y})$ is the scaled inward normal to the face (edge) opposed to the vertex i, $|\mathbf{n}_i| = \sqrt{n_{i_x}^2 + n_{i_y}^2}$ (see Fig. 1), and $\hat{\mathbf{n}}_i = (\hat{n}_{i_x}, \hat{n}_{i_y})$ is the unit inward normal. We note that for every element $\sum_{i=1}^{3} \mathbf{n}_i = 0$ and therefore we have $\sum_{i=1}^{3} \mathbf{K}_i = 0$.

The next step is to distribute the residuals to the three vertices by the distribution matrices, $\mathbf{B}_1^E$, $\mathbf{B}_2^E$, $\mathbf{B}_3^E$, as illustrated in Fig. 1. The discussion of the distribution matrix is one of the key contributions of the present paper, but we leave the choice open at this point. We only mention here that $\mathbf{B}_i^E$ is a 3×3 matrix, which sums up to the identity matrix over the element for conservation. The distribution process results in the nodal residual at node j:

$$\mathbf{Res}_j = \frac{1}{d\Omega_j} \sum_{E \in \{E_j\}} \mathbf{B}_j^E \mathbf{\Phi}_j^E, \tag{10}$$

where $d\Omega_j$ is the median dual volume around the node j (see Fig. 2), and $\{E_j\}$ is a set of triangular elements sharing the node j. The nodal residual is an approximation of the spatial part of the target equations, and thus it leads to a semi–discrete form:

$$\frac{d\mathbf{U}_j}{d\tau} = \mathbf{Res}_j. \tag{11}$$

It can be integrated to the steady state by explicit pseudo–time stepping schemes as in Refs. 5, 6, which is significantly faster than conventional explicit schemes because the hyperbolic formulation eliminates a typical diffusion constraint, $O(h^2)$, where h is a representative mesh spacing, on the explicit time step.[5,6] However, it still requires a large number of iterations, especially for anisotropic grids. To improve the convergence, we develop an implicit solver in this work, which is explained in the next section.
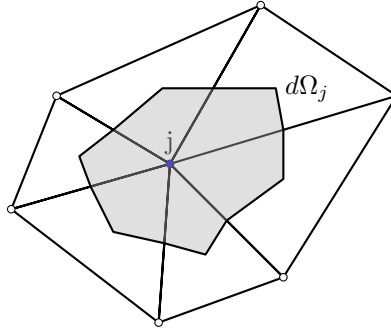

Figure 2. Schematic of a median dual volume around the node j.

C.   Implicit solver

To solve Eq. (11) for the pseudo steady state, we drop the pseudo–time derivative, and define the global system of steady residual equations,

$$0 = \mathbf{Res}, \tag{12}$$

which consists of the right hand side of Eq. (11) for all nodes. Note that the resulting residual equation has become consistent with the advection–diffusion equation (1) because the pseudo–time derivative has been

American Institute of Aeronautics and Astronautics

dropped. We solve the system by Newton's method:

$$\mathbf{U}^{l+1} = \mathbf{U}^l + \Delta\mathbf{U}^l, \tag{13}$$

where $\mathbf{U} = (u_1, p_1, q_1, u_2, p_2, q_2, \ldots, u_N, p_N, q_N)$ and l is the iteration counter. The correction $\Delta\mathbf{U}^l = \mathbf{U}^{l+1} - \mathbf{U}^l$ is determined as the solution to the linear system:

$$-\frac{\partial\mathbf{Res}}{\partial\mathbf{U}}\Delta\mathbf{U}^l = \mathbf{Res}^l(\mathbf{U}^l), \tag{14}$$

where $\mathbf{Res}^l$ is the steady residual vector evaluated by $\mathbf{U}^l$. The Jacobian matrix $\partial\mathbf{Res}/\partial\mathbf{U}$ is exact and sparse because the spatial discretization is compact. For each node $j \in \{J\}$, it involves $(k+1)$ 3×3 blocks, where k is the number of immediate neighboring nodes to the node j. For example, $k = 6$ for the node j shown in Fig. 2 and therefore, for this particular node, there are seven 3×3 blocks. For a node j, we have

$$-\mathbf{J}_j\Delta\mathbf{U}_j^l - \sum_{m=1}^{k}\mathbf{J}_{j,m}\Delta\mathbf{U}_m^l = \mathbf{Res}_j(\mathbf{U}^l), \tag{15}$$

where

$$\mathbf{J}_j = \frac{\partial\mathbf{Res}_j}{\partial\mathbf{U}_j}, \quad \mathbf{J}_{j,m} = \frac{\partial\mathbf{Res}_j}{\partial\mathbf{U}_m}. \tag{16}$$

We may analytically evaluate the diagonal and off–diagonal entries of the Jacobian matrix, i.e., $\mathbf{J}_j$ and $\mathbf{J}_{j,m}$, but this process is rather tedious for general time–dependent advection–diffusion problems, and even more difficult for complex systems such as the Navier–Stokes equations. To overcome the difficulty, we implemented an *Automatic Differentiation* (AD) tool based on an operator–overloading technique to evaluate the exact Jacobians numerically through chain rule. Thus, the Jacobian matrix is exact up to the round–off error for the baseline and second–order schemes. The linear system is relaxed by the sequential Gauss–Seidel relaxation with an under–relaxation parameter introduced for stabilization based on Fourier analysis of the linear system (see Appendix A for more details on stability analysis) . Typically, the relaxation is performed until the linear residual is reduced by five orders of magnitude with a maximum relaxation steps of 1000.

D.   Implicit boundary condition

Here, we discuss two Boundary Condition (BC) types: Dirichlet (i.e., u is known), and Neumann (i.e., $\partial_n u$ is known). We note that because gradients are also the primary variables in the hyperbolic method (i.e., not reconstructed from the solution variable), the Neumann type BC is treated similarly as the Dirichlet type BC.
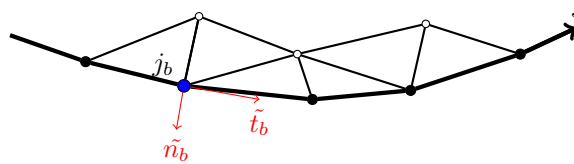

Figure 3. **Schematic of boundary nodes for formulation of implicit boundary condition.**

For a Dirichlet type BC, the following set of equations is imposed at the boundary nodes:

$$
\begin{align}
u - u_b &= 0, \tag{17}\\
(p, q) \cdot \tilde{t}_b &= \partial_s u_b, \tag{18}\\
(\partial_x u - p, \partial_y u - q) \cdot \tilde{n}_b &= 0, \tag{19}
\end{align}
$$

where u_b is the given boundary value and $\partial_s u_b$ is the derivative of the u_b along the boundary, which can be computed with the given u_b. The $\tilde{n}_b$ and $\tilde{t}_b$ are, respectively, the unit normal and tangent vectors at the boundary nodes (see Fig. 3). For a strong Dirichlet type BC, we specify the BC value at the boundary nodes and solve the hyperbolic system accordingly. Here, we discretize and solve Eq. (19) according to

American Institute of Aeronautics and Astronautics

the hyperbolic RD scheme described in the previous section. That is, the residual vector at the node j_b is redefined as

$$
\begin{bmatrix}
u_{j_b} - u_b \\
(p_{j_b}, q_{j_b}) \cdot \tilde{t}_b - \partial_s u_b \\
\mathbf{Res}_{j_b}(2)\tilde{n}_{b_x} + \mathbf{Res}_{j_b}(3)\tilde{n}_{b_y}
\end{bmatrix},
\tag{20}
$$

where $\mathbf{Res}_{j_b}(2)$ and $\mathbf{Res}_{j_b}(3)$ are the nodal residuals of the second and third equations of the first–order hyperbolic system (i.e., Eq. 3 and Eq. 4, respectively), computed as described in the previous sections.

For a Neumann type BC, we impose the following equations to the boundary nodes:

$$
\begin{aligned}
\partial_\tau u + a\partial_x u + b\partial_y u &= \nu(\partial_x p + \partial_y q) + s, & (21) \\
(p, q) \cdot \tilde{n}_b &= \partial_n u_b, & (22) \\
(\partial_x u - p, \partial_y u - q) \cdot \tilde{t}_b &= 0, & (23)
\end{aligned}
$$

where $\partial_n u_b$ is the given gradient of the u on the boundary in the normal direction, and therefore is known. For the strong formulation of a Neumann type BC, we discretize and solve Eqs. (21) and (23) according to the RD scheme; that is

$$
\begin{bmatrix}
\mathbf{Res}_{j_b}(1) \\
(p_{j_b}, q_{j_b}) \cdot \tilde{n}_b - \partial_n u_b \\
\mathbf{Res}_{j_b}(2)\tilde{t}_{b_x} + \mathbf{Res}_{j_b}(3)\tilde{t}_{b_y}
\end{bmatrix},
\tag{24}
$$

where $\mathbf{Res}_{j_b}(1)$ is the nodal residual of the first equation of the first–order hyperbolic system computed as described in the previous sections.

In either of the Dirichlet or the Neumann type BC, the modified residual is incorporated into the implicit solver with the exact Jacobians. In doing so, a care must be taken to avoid zero diagonal entries in the Jacobian; the second and third components in the boundary residual need to be exchanged, depending on the magnitude of $\tilde{n}_{b_x}$ and $\tilde{n}_{b_y}$, to avoid zero diagonals.

E.   Remark on variable advection coefficients

If the advection coefficients a and b are functions of (x, y), advective fluxes are not necessarily conservative. In this case, we estimate the advection vector by the arithmetic average over an element to write

$$
\frac{\partial \mathbf{U}}{\partial \tau} + \overline{\mathbf{A}}\frac{\partial \mathbf{U}}{\partial x} + \overline{\mathbf{B}}\frac{\partial \mathbf{U}}{\partial y} = \mathbf{Q},
\tag{25}
$$

where $\overline{\mathbf{A}}$ and $\overline{\mathbf{B}}$ are defined with the averaged advection vector $(\bar{a}, \bar{b})$:

$$
\mathbf{A} = \begin{bmatrix}
\bar{a} & -\nu & 0 \\
-1/T_r & 0 & 0 \\
0 & 0 & 0
\end{bmatrix}, \quad
\mathbf{B} = \begin{bmatrix}
\bar{b} & 0 & -\nu \\
0 & 0 & 0 \\
-1/T_r & 0 & 0
\end{bmatrix}.
\tag{26}
$$

The baseline RD scheme is then applicable. If conservative fluxes exist, then the nonlinear formulation described in Sec. VII can be employed to construct an RD scheme.

III.   Distribution matrices: non–unified approach

The RD scheme is defined by the combination of the cell–residuals and the distribution matrix. The distribution matrix is mainly responsible for the dissipative behavior and the stability while the cell–residuals determine the order of accuracy. Unbounded distribution matrices can also affect the order of accuracy, but we consider only bounded ones in this paper. The construction of the distribution matrix often requires

American Institute of Aeronautics and Astronautics

the knowledge of the eigenstructure of the target system. The eigenstructure of the hyperbolic advection–diffusion system is known[6] and various distribution matrices can be constructed. However, the extension to the compressible Navier–Stokes equations is not possible at the moment because the eigenstructure has not been successfully worked out yet. In order to develop hyperbolic RD schemes that can be extended to the compressible Navier–Stokes equations, we consider a non–unified approach introduced for FV schemes in Refs. 7 and 9, respectively, for the compressible Navier-Stokes and advection–diffusion equations. In this approach, we construct distribution matrices for advection and diffusion separately and then combine them to derive a distribution scheme for the advection–diffusion equation. This approach easily extends to the compressible Navier–Stokes equations because the eigenstructures of the inviscid terms and the hyperbolized viscous terms can be fully analyzed independently.[7]

A.   Unified approach

To illustrate the point of the non–unified approach, we first recall the unified approach of Ref. 6. Consider an arbitrary unit vector $\hat{\mathbf{n}} = (\hat{n}_x, \hat{n}_y)$. The unified advection–diffusion flux Jacobian is defined as:

$$\mathbf{A}_n = \mathbf{A}\hat{n}_x + \mathbf{B}\hat{n}_y = \begin{bmatrix} a_n & -\nu\,\hat{n}_x & -\nu\,\hat{n}_y \\ -\hat{n}_x/T_r & 0 & 0 \\ -\hat{n}_y/T_r & 0 & 0 \end{bmatrix}, \tag{27}$$

where $a_n = a\,\hat{n}_x + b\,\hat{n}_y$ is the advection velocity projected onto the unit vector $\hat{\mathbf{n}}$. The Jacobian $\mathbf{A}_n$ has the following real eigenvalues:

$$\lambda_1 = \frac{1}{2}\left[a_n - \sqrt{a_n^2 + \frac{4\nu}{T_r}}\,\right], \quad \lambda_2 = \frac{1}{2}\left[a_n + \sqrt{a_n^2 + \frac{4\nu}{T_r}}\,\right], \quad \lambda_3 = 0, \tag{28}$$

where $\lambda_1 < 0$ and $\lambda_2 > 0$. The arbitrary but positive relaxation time, T_r, is defined as

$$T_r = \frac{L_r}{\sqrt{a^2 + b^2} + \nu/L_r}, \tag{29}$$

where the length scale L_r is a quantity of $O(1)$, which may be taken as $L_r = 1/2\pi$, as used here, or from an optimal formula as derived in Ref. 6. The right and left eigenvectors for the Jacobian $\mathbf{A}_n$ are given by

$$\mathbf{R}_n = \begin{bmatrix} -\lambda_1 T_r & -\lambda_2 T_r & 0 \\ \hat{n}_x & \hat{n}_x & -\hat{n}_y \\ \hat{n}_y & \hat{n}_y & \hat{n}_x \end{bmatrix}, \quad \mathbf{L}_n = \frac{1}{\lambda_1 - \lambda_2}\begin{bmatrix} -1/T_r & -\lambda_2\hat{n}_x & -\lambda_2\hat{n}_y \\ 1/T_r & \lambda_1\hat{n}_x & \lambda_1\hat{n}_y \\ 0 & -(\lambda_1 - \lambda_2)\hat{n}_y & (\lambda_1 - \lambda_2)\hat{n}_x \end{bmatrix}. \tag{30}$$

The unified Jacobian $\mathbf{A}_n$ can be rewritten based on its right and left eigenvectors as

$$\mathbf{A}_n = \mathbf{R}_n \mathbf{\Lambda}_n \mathbf{L}_n, \tag{31}$$

where $\mathbf{\Lambda}$ is the diagonal matrix with the entries of the eigenvalues defined in Eq. (28). The Jacobian matrix can be decomposed as (see Ref. 6)

$$\mathbf{A}_n = \lambda_1 \mathbf{\Pi}_1 + \lambda_2 \mathbf{\Pi}_2 = \mathbf{A}_n^- + \mathbf{A}_n^+, \tag{32}$$

where

$$\mathbf{\Pi}_1 = \frac{1}{\lambda_1 - \lambda_2}\begin{bmatrix} \lambda_1 & \lambda_1\lambda_2 T_r\hat{n}_x & \lambda_1\lambda_2 T_r\hat{n}_y \\ -\hat{n}_x/T_r & -\lambda_2\hat{n}_x^2 & -\lambda_2\hat{n}_x\hat{n}_y \\ -\hat{n}_y/T_r & -\lambda_2\hat{n}_x\hat{n}_y & -\lambda_2\hat{n}_y^2 \end{bmatrix}, \tag{33}$$

$$\mathbf{\Pi}_2 = \frac{1}{\lambda_1 - \lambda_2}\begin{bmatrix} -\lambda_2 & -\lambda_1\lambda_2 T_r\hat{n}_x & -\lambda_1\lambda_2 T_r\hat{n}_y \\ \hat{n}_x/T_r & \lambda_1\hat{n}_x^2 & \lambda_1\hat{n}_x\hat{n}_y \\ \hat{n}_y/T_r & \lambda_1\hat{n}_x\hat{n}_y & \lambda_1\hat{n}_y^2 \end{bmatrix}. \tag{34}$$

American Institute of Aeronautics and Astronautics

Thus, the $\mathbf{K}_i$ defined in Eq. (9) can be written as

$$\mathbf{K}_i = \mathbf{K}_i^- + \mathbf{K}_i^+, \tag{35}$$

where

$$\mathbf{K}_i^- = \frac{1}{2}\lambda_{1,i}\,\mathbf{\Pi}_{1,i}\,|\mathbf{n}_i| = \frac{1}{2}\mathbf{A}_{n_i}^-|\mathbf{n}_i|, \tag{36}$$

$$\mathbf{K}_i^+ = \frac{1}{2}\lambda_{2,i}\,\mathbf{\Pi}_{2,i}\,|\mathbf{n}_i| = \frac{1}{2}\mathbf{A}_{n_i}^+|\mathbf{n}_i|. \tag{37}$$

These matrices are often required for the construction of the distribution matrix, and as can be seen from the above, they require the eigenvalues and the eigenvectors, which are not known at present for a hyperbolic formulation of the compressible Navier–Stokes equations.

B.   Non–unified approach

In the non–unified approach,[7,9] we treat the advective and diffusive terms separately. Therefore we rewrite Eq. (5) as

$$\frac{\partial \mathbf{U}}{\partial \tau} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = \mathbf{Q}, \tag{38}$$

where

$$\mathbf{F} = \mathbf{F}^a + \mathbf{F}^d = \begin{bmatrix} au \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -\nu p \\ -u/T_r \\ 0 \end{bmatrix}, \tag{39}$$

$$\mathbf{G} = \mathbf{G}^a + \mathbf{G}^d = \begin{bmatrix} bu \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -\nu q \\ 0 \\ -u/T_r \end{bmatrix}. \tag{40}$$

Similarly, the flux Jacobian $\mathbf{A}_n$ is decomposed into two advective and diffusive fluxes, $\mathbf{A}_n^a$ and $\mathbf{A}_n^d$, respectively:

$$\mathbf{A}_n = \mathbf{A}_n^a + \mathbf{A}_n^d, \tag{41}$$

where

$$\mathbf{A}_n^a = \begin{bmatrix} a_n & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{A}_n^d = \begin{bmatrix} 0 & -\nu\hat{n}_x & -\nu\hat{n}_y \\ -\hat{n}_x/T_r & 0 & 0 \\ -\hat{n}_y/T_r & 0 & 0 \end{bmatrix}. \tag{42}$$

The distribution matrix is constructed separately for the advective and diffusive terms based on the corresponding flux Jacobian, and then the two matrices are combined to form a matrix for the advection–diffusion equation. This approach can be extended to the compressible Navier–Stokes equations because the eigenstructure of the hyperbolized viscous terms is available.[7]

The advective Jacobian has only one non–zero eigenvalue ($\lambda^a = a_n$) and thus we can simply decompose it according to the sign of λ^a:

$$\mathbf{A}_n^a = \mathbf{A}_n^{a^+} + \mathbf{A}_n^{a^-} = \begin{bmatrix} \max(0, a_n) & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} \min(a_n, 0) & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{43}$$

The diffusive Jacobian has two non–zero eigenvalues, $\lambda_1^d = -\sqrt{\nu/T_r}$, $\lambda_2^d = +\sqrt{\nu/T_r}$, with the right and left eigenvectors formally in the form of Eq. (30). Note, however, that they are numerically different because the eigenvalues are different. The diffusive Jacobian can be split as

$$\mathbf{A}_n^d = \mathbf{R}_n^d \mathbf{\Lambda}_n^d \mathbf{L}_n^d = \lambda_1^d \mathbf{\Pi}_1 + \lambda_2^d \mathbf{\Pi}_2 = \mathbf{A}_n^{d^+} + \mathbf{A}_n^{d^-}, \tag{44}$$

where $\mathbf{\Pi}_1$ and $\mathbf{\Pi}_2$ are given by Eq. (33) and Eq. (34), respectively, with $\lambda_1 = \lambda_1^d$ and $\lambda_2 = \lambda_2^d$.

We define $\mathbf{K}_i^d$ by

$$\mathbf{K}_i^d = \frac{1}{2}\mathbf{A}_{n_i}^d |\mathbf{n}_i| = \mathbf{K}_i^{d^-} + \mathbf{K}_i^{d^+}, \tag{45}$$

where the negative and positive matrices, $\mathbf{K}_i^{d^-}$ and $\mathbf{K}_i^{d^+}$ are given by

$$\mathbf{K}_i^{d^-} = \frac{1}{2}\lambda_{1,i}^d \mathbf{\Pi}_{1,i} |\mathbf{n}_i| = \frac{1}{2}\mathbf{A}_{n_i}^{d^-} |\mathbf{n}_i|, \tag{46}$$

$$\mathbf{K}_i^{d^+} = \frac{1}{2}\lambda_{2,i}^d \mathbf{\Pi}_{2,i} |\mathbf{n}_i| = \frac{1}{2}\mathbf{A}_{n_i}^{d^+} |\mathbf{n}_i|. \tag{47}$$

Various distribution matrices can be employed for the hyperbolic RD schemes: Lax–Wendroff (LW),[5] Low–Diffusion–A (LDA),[6] and Streamline–Upwind–Petrov–Galerkin (SUPG) distribution matrices. In the following sections, we describe the LDA and the SUPG distribution schemes and their formulations in the context of hyperbolic RD scheme (the LW formulation is similar to the SUPG). For completeness, both unified and non–unified approaches are presented.

C. Distribution matrix: Low–Diffusion–A (LDA)

The LDA scheme is a multidimensional upwind scheme, meaning that $\mathbf{B}_i^{\text{LDA}}$ becomes zero when no signal is being sent to the the upwind nodes. In the unified approach, the distribution matrix $\mathbf{B}_i^E$ for the LDA scheme[15,16] is defined as

$$\mathbf{B}_i^{\text{LDA}} = \mathbf{K}_i^+ \left(\sum_{j=1}^3 \mathbf{K}_j^+ \right)^{-1}. \tag{48}$$

For the non–unified approach, the distribution matrix should separately be constructed with the $\mathbf{K}_i^{+\,a}$ and the $\mathbf{K}_i^{+\,d}$ terms, corresponding to the advection and the diffusion components, respectively. The diffusion component of the distribution matrix, $\mathbf{B}_i^d$, is similar to the Eq. (51), except that the $\mathbf{K}_i^+$ is replaced with the $\mathbf{K}_i^{+\,d}$:

$$\mathbf{B}_i^d = \mathbf{K}_i^{+\,d} \left(\sum_{j=1}^3 \mathbf{K}_j^{+\,d} \right)^{-1}. \tag{49}$$

Note that the inverse matrix is usually evaluated numerically, as done here, but the analytical expression for $\mathbf{B}_i^d$ is also available in Ref. 17, which we used it only for verification purposes. The distribution matrix for the advection term, $\mathbf{B}_i^a$, reduces, however, to a scalar equation as given below because there is no contribution from the gradient equations:

$$\mathbf{B}_i^a = \begin{bmatrix} \beta_i^{\text{LDA}} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \beta_i^{\text{LDA}} = \frac{\max(0, a_{n_i})|n_i|}{\sum_{j=1}^3 \max(0, a_{n_j})|n_j|}. \tag{50}$$

These two matrices are combined in the following form to define the LDA scheme for the advection–diffusion equation:

$$\mathbf{B}_i^{\text{LDA}} = (\mathbf{I} - \mathbf{I}_\omega)\mathbf{B}_i^a + \mathbf{I}_\omega \mathbf{B}_i^d, \tag{51}$$

where

$$\mathbf{I}_\omega = \begin{bmatrix} \omega & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{52}$$

and ω is a weighting function. From a detailed analysis of a one–dimensional hyperbolic advection–diffusion system, the weighting function can be evaluated as[6]

$$\omega = \frac{2}{\text{Re} + 2}, \tag{53}$$

where we define the Re for the two–dimensional system as

$$\text{Re} = \sqrt{a^2 + b^2}/\nu. \tag{54}$$

The weights have been introduced for two purposes. One is to guarantee the conservation:

$$\sum_{i=1}^{3} \mathbf{B}_i^{\text{LDA}} = \mathbf{I}, \tag{55}$$

and the other is to ensure that the scheme approaches a pure advection scheme in the limit of $Re \to \infty$, and a pure diffusion scheme as $Re \to 0$. Note, for example, that β_i^{LDA} is $O(1)$ even for a vanishingly small advective vector, and thus it will affect the distribution even in the diffusion limit if the weighing function is independent of Re.

In our experience, hyperbolic RD schemes with the LDA distribution matrix, constructed by the non–unified approach, have an instability problem in the linear relaxation on some anisotropic triangular grids even with the employment of uder–relaxation parameter obtained from Fourier analysis (see Appendix A). On the other hand, hyperbolic RD schemes can be successfully stabilized with a suitable under–relaxation parameter in the case of the SUPG distribution matrix, which is discussed next.

D.   Distribution matrix: Streamline–Upwind–Petrov–Galerkin (SUPG)

The finite–element SUPG scheme[18] is known to be applicable to the RD framework.[19] In the unified approach, the SUPG distribution matrix for triangular grids is given by

$$\tilde{\mathbf{B}}_i^{\text{SUPG}} = \frac{1}{3}\mathbf{I} + \frac{1}{2}\mathbf{K}_i \left(\sum_{l=1}^{3} \mathbf{K}_l^+ \right)^{-1}, \tag{56}$$

which consists of the Galerkin part (the first term) and the stabilization part (the second term). In the non–unified approach, we construct the SUPG distribution matrix as follows:

$$\mathbf{B}_i^{\text{SUPG}} = \frac{1}{3}\mathbf{I} + \mathbf{D}_i^a + \mathbf{D}_i^d, \tag{57}$$

where $\mathbf{D}_i^a$ and $\mathbf{D}_i^d$ are the stabilization terms defined independently for the advective and diffusive terms,

$$\mathbf{D}_i^a = \begin{bmatrix} d_i^{\text{SUPG}} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad d_i^{\text{SUPG}} = \frac{1}{2}\frac{a_{n_i}|n_i|}{\sum_{l=1}^{3}\max(0, a_{n_l})|n_l|}, \tag{58}$$

$$\mathbf{D}_i^d = \frac{1}{2}\mathbf{K}_i^d \left(\sum_{l=1}^{3} \mathbf{K}_l^{d+} \right)^{-1}. \tag{59}$$

Note that the denominator of Eq. (58) cannot vanish unless the advection vector is exactly zero. A small numerical value of the order of machine zero may be added to the denominator in order to avoid zero division in the case of zero advection vector. Another possibility is to completely remove D_i^a term for zero advection vector; both approaches yield identical result.

It is also important to note that we have $\sum_{i=1}^{3} \mathbf{D}_i^a = \sum_{i=1}^{3} \mathbf{D}_i^d = 0$, and therefore the conservation property is guaranteed:

$$\sum_{i=1}^{3} \mathbf{B}_i^{\text{SUPG}} = \mathbf{I}. \tag{60}$$

A weighting function can be introduced in the stabilization terms of Eq. (57),

$$\mathbf{B}_i^{\text{SUPG}} = \frac{1}{3}\mathbf{I} + (1 - \omega)\mathbf{D}_i^a + \omega\mathbf{D}_i^d, \tag{61}$$

where ω is defined such that $\omega \to 0$ as $Re \to \infty$, and $\omega \to 1$ as $Re \to 0$, so that the scheme will properly reduce to pure advection and diffusion schemes as $Re \to \infty$ and $Re \to 0$, respectively. The weighting function ω can be evaluated, based on a detailed analysis for a one–dimensional hyperbolic advection–diffusion system,[6] as

$$\omega = \frac{2}{Re + 2}, \tag{62}$$

where we define the Re for the two–dimensional system as

$$Re = \sqrt{a^2 + b^2}/\nu. \tag{63}$$

In our study, however, we observed no noticeable change in the results between Eq. (57) and Eq. (61) with the above weighting function. Therefore, the results shown in this paper are all based on the SUPG distribution matrix without the weighting function. Details of the effects of the weighting function is a subject for future study. We remark that unlike in the LDA formulation, where the weighting function is required to preserve conservation, the SUPG formulation does not require any weighing function because conservation is guaranteed regardless.

As can be seen from above, the SUPG distribution matrix is constructed based on a decomposition of the stabilization term into advective and diffusive stabilization terms. Therefore, the SUPG distribution matrix can be extended to a hyperbolic formulation of the compressible Navier–Stokes equations, for which the inviscid and viscous terms can be analyzed independently.[7]

IV.   New design principle for advection–diffusion equation

An important feature of the RD schemes critical to the accuracy condition is the ability to preserve exact polynomial solutions on arbitrary grids. Historically, RD schemes have been designed to preserve linear exact solutions for hyperbolic systems of conservation laws. This is called the linearity preservation. Such schemes are known to yield second–order discretization errors. The baseline hyperbolic RD scheme[5] is constructed based on the linearity preserving property. Although it gives second–order accuracy for all variables on regular triangular grids,[5] it was found later that the order of accuracy of the gradient variables (i.e., p and q) deteriorates to first–order on irregular grids. In a subsequent work presented in Ref. 6, a better scheme was proposed, which attempted to improve the accuracy of the gradients by upgrading the evaluation of the second and third components of the residual. In this paper, the scheme of Ref. 6 is referred to as the RD–JCP2010 scheme. Results given in Ref. 6 showed that the RD–JCP2010 scheme, relative to the baseline scheme, improves the order of accuracy of the gradient variables, but no results were presented for the quality of the predicted gradients. As will be shown later in this paper, we found that for a different test problem and a more general exact solution, the same RD–JCP2010 scheme does not yield second–order accuracy for the gradient variables (i.e., p and q), and also generates severe oscillations in these variables within the domain. It appears that the discretization of the source terms (including those arising from the hyperbolic formulation) was not fully compatible with the flux terms, and it is not clear to us, at the moment, how to improve the source term discretization to develop a fully second–order scheme. To overcome this difficult problem, we instead propose to improve the flux terms such that the residual vanishes for quadratic solutions for the advection–diffusion equation. The exactness, as will be demonstrated later, has a significant impact on the accuracy and the quality of the gradient variables on irregular grids. The construction of such schemes is very difficult in general because second–order gradients need to be recovered from second–order accurate solutions on irregular grids for the diffusive term. However, the construction is quite straightforward in the hyperbolic method since the gradients are computed simultaneously with the solution variable u. The scheme also extends easily to third–order by imposing the exactness for cubic solutions. Note that the exactness is not a necessary condition for accuracy. For example, the third-order RD scheme proposed in Ref. 20 does not preserve cubic solutions, but was shown to produce third–order accuracy on smooth unstructured triangular grids.

In the next two sections, we describe how to construct second– and third–order accurate schemes that, respectively, preserve quadratic and cubic solutions on arbitrary triangular grids.

American Institute of Aeronautics and Astronautics

V.   Improved second–order scheme: RD–CC2

Consider the cell–residuals for the baseline hyperbolic RD scheme:

$$\Phi_u^E = \int_E (-au_x - bu_y + \nu p_x + \nu q_y + s)dxdy = \frac{1}{2}\sum_{i=1}^{3}\left(-a_{n_i}u_i|n_i| + \nu p_i n_{i_x} + \nu q_i n_{i_y}\right) + \overline{s}^E d\Omega^E, \qquad (64)$$

$$\Phi_p^E = \frac{1}{T_r}\int_E (u_x - p)dxdy = \frac{1}{T_r}\left(\frac{1}{2}\sum_{i=1}^{3}u_i n_{i_x} - \overline{p}^E d\Omega^E\right), \qquad (65)$$

$$\Phi_q^E = \frac{1}{T_r}\int_E (u_y - q)dxdy = \frac{1}{T_r}\left(\frac{1}{2}\sum_{i=1}^{3}u_i n_{i_y} - \overline{q}^E d\Omega^E\right). \qquad (66)$$

These residuals are constructed based on the assumption that all variables (i.e., u, p, q) are linear over the element, and therefore they do not preserve quadratic solutions. Note that quadratic solutions imply linear gradients (i.e., u_x, u_y, p, q), and the above cell residuals are already exact for linear gradients. Therefore, to preserve quadratic solutions, the integrals of u_x and u_y terms need to be exact for quadratic solutions. A simple way to accomplish this is to reconstruct a quadratic element by interpolating the solution at the midpoint of each side in the element (see Fig. 4). The midpoint value u_{m_i} is estimated by the Hermite
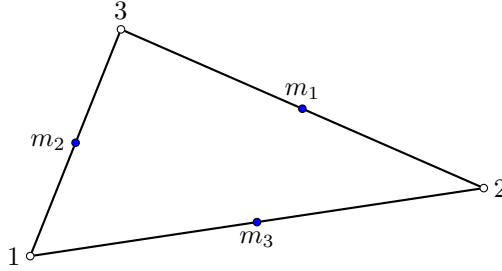
Figure 4. Reconstructed P_2 element with virtual midpoints.

interpolation along the side:

$$u_{m_i} = \overline{u}_i - \frac{1}{8}\left(\Delta p_i \Delta x_i + \Delta q_i \Delta y_i\right), \qquad (67)$$

where $\overline{u}_i$ is the arithmetic average of the solution values at the two end nodes, and $\Delta(\ )_i$ denotes the difference of the nodal values taken counterclockwise along the edge opposite to the node i, e.g., $\Delta x_3 = x_1 - x_2$. Note that the midpoint value u_{m_i} is exact for quadratic solutions (and linear gradients). Once the quadratic element is reconstructed, the cell–residuals are evaluated as line integrals with Simpson's rule applied along each side. This reconstruction approach was first introduced in Refs. 21, 22, and later employed in the form of the Green–Gauss gradient with a high–order curvature correction term.[6, 14, 23] In this work, we employ the latter implementation to upgrade the baseline residuals. Note that the formula (67) usually requires LSQ gradient reconstruction to obtain the nodal gradients,[14, 21–23] but it is not necessary in the hyperbolic method because $p(= u_x)$ and $q(= u_y)$ are already available at nodes as a part of the primary variables.

Applying the curvature correction, we modify the cell–residuals of the baseline RD scheme as

$$\tilde{\Phi}_u^E = -\frac{1}{2}\sum_{i=1}^{3}a_{n_i}(u_i + \delta_i^u)|n_i| + \frac{\nu}{2}\sum_{i=1}^{3}\left(p_i n_{i_x} + q_i n_{i_y}\right) + \overline{s}^E d\Omega^E, \qquad (68)$$

$$\tilde{\Phi}_p^E = \frac{1}{T_r}\left(\frac{1}{2}\sum_{i=1}^{3}(u_i + \delta_i^u)n_{i_x} - \overline{p}^E d\Omega^E\right), \qquad (69)$$

$$\tilde{\Phi}_q^E = \frac{1}{T_r}\left(\frac{1}{2}\sum_{i=1}^{3}(u_i + \delta_i^u)n_{i_y} - \overline{q}^E d\Omega^E\right), \qquad (70)$$

where the curvature correction term, δ_i^u, is defined as

$$\delta_i^u = \frac{1}{6}(\Delta p_i \Delta x_i + \Delta q_i \Delta y_i). \tag{71}$$

Note that in Ref. 6, δ_i^u was mistakenly shown with a negative sign. One may find it convenient that the compact RD–CC2 scheme can be implemented as an add–on to the baseline hyperbolic RD scheme:

$$\tilde{\Phi}_u^E = \Phi_u^E - \frac{1}{2}\sum_{i=1}^{3} a_{n_i} \delta_i^u |n_i|, \tag{72}$$

$$\tilde{\Phi}_p^E = \Phi_p^E + \frac{1}{T_r}\left(\frac{1}{2}\sum_{i=1}^{3} \delta_i^u n_{x_i}\right), \tag{73}$$

$$\tilde{\Phi}_q^E = \Phi_q^E + \frac{1}{T_r}\left(\frac{1}{2}\sum_{i=1}^{3} \delta_i^u n_{y_i}\right). \tag{74}$$

The modified cell–residuals are distributed by the SUPG distribution matrix. The RD scheme based on these cell–residuals is referred to as the RD–CC2 scheme. This scheme is compact because no LSQ gradient reconstruction is necessary. The exactness for quadratic solutions has been confirmed numerically, and are compared with the baseline RD and RD–JCP2010 schemes. These results are summarized in Table 1,

Table 1. Exactness of the hyperbolic RD equations for exact quadratic and cubic polynomial solutions on irregular grids. The last column indicates whether the exactness ($\checkmark$) is based on cell and/or nodal residuals.

Hyperbolic RD Scheme	Quadratic solution			Cubic solution			Cell/Nodal
	u eq.	p eq.	q eq.	u eq.	p eq.	q eq.	
Baseline	✗	✗	✗	✗	✗	✗	none
RD–JCP2010	✗	✓	✓	✗	✗	✗	cell
RD–CC2	✓	✓	✓	✗	✗	✗	both

The truncation error orders for the RD–CC2 scheme, the baseline RD scheme, and the RD–JCP2010 scheme are provided in Table 2. The truncation error orders have been determined by substituting a smooth exact solution (taken from Ref. 24) into the residuals for a series of irregular triangular grids. Clearly, all schemes have the same truncation error order for the nodal residual. A known theory for hyperbolic conservation laws[25, 26] states that the discretization error of $O(h^2)$ leads to the truncation error of $O(h)$. However, our numerical results indicate that the converse is not necessarily true. As we will show later, only the RD–CC2 scheme, which has $O(h)$ truncation order, gives discretization error of $O(h^2)$.

Table 2. Comparison between numerical truncation error of the nodal residuals (i.e., T.E. $= \frac{1}{d\Omega_j}\sum_j \mathbf{B}^E \mathbf{\Phi}^E$) obtained with RD–CC2 and non–genuine second–order RD schemes on irregular grids.

Hyperbolic RD Scheme	Nodal Residuals		
	u equation	p equation	q equation
Baseline	$O(h)$	$O(h)$	$O(h)$
RD–JCP2010	$O(h)$	$O(h)$	$O(h)$
RD–CC2	$O(h)$	$O(h)$	$O(h)$

We remark that the RD–CC2 scheme is different from the RD–JCP2010 scheme,[6] which has the curvature correction only in $\tilde{\Phi}_p^E$ and $\tilde{\Phi}_q^E$ residuals. Thus, the whole residual vector of RD–JCP2010 is not exact for quadratic solutions as confirmed in Table 1. The RD–CC2 scheme, on the other hand, has the curvature correction applied to all the residuals, and therefore is exact for quadratic solutions (and linear gradients).

To solve the discrete equations, we employ the implicit solver described in Sec. C. The RD–CC2 scheme is compact and therefore we linearize the residual vector exactly by using the AD tool. The implicit solver

is thus Newton's method and converges to machine zero (i.e., ten orders of magnitude reduction in all the residuals) typically within 3–5 Newton iterations.

VI.   Third–order scheme: RD–CC3

To extend the compact RD–CC2 scheme to a third–order scheme, we modify the cell–residuals such that they preserve exact cubic solutions and quadratic gradients. Following the procedure explained in Sec. V for the RD–CC2, and thanks to the fact that both the Hermite interpolation and Simpson's rule are exact for cubic functions, we employ, again, the curvature correction technique. Extension to third–order, therefore, requires two additional steps: 1) high–order source term discretization, and 2) quadratic LSQ gradients for p, q, and the source term s. The high–order discretization of the source term is a critical piece for achieving third–order accuracy. We derive a formula by estimating the midpoint value by the Hermite interpolation and then exactly integrating the source terms assuming a quadratic variation over the element:

$$\int_E s \, dx dy = \left(\overline{s}^E - \frac{1}{4} \sum_{i=1}^{3} \delta_i^s \right) d\Omega^E, \tag{75}$$

$$\int_E p \, dx dy = \left(\overline{p}^E - \frac{1}{4} \sum_{i=1}^{3} \delta_i^p \right) d\Omega^E, \tag{76}$$

$$\int_E q \, dx dy = \left(\overline{q}^E - \frac{1}{4} \sum_{i=1}^{3} \delta_i^q \right) d\Omega^E, \tag{77}$$

where

$$\delta_i^s = \frac{1}{6} (\Delta s_{x_i} \Delta x_i + \Delta s_{y_i} \Delta y_i), \tag{78}$$

$$\delta_i^p = \frac{1}{6} (\Delta p_{x_i} \Delta x_i + \Delta p_{y_i} \Delta y_i), \tag{79}$$

$$\delta_i^q = \frac{1}{6} (\Delta q_{x_i} \Delta x_i + \Delta q_{y_i} \Delta y_i). \tag{80}$$

Using these high–order source discretizations, we construct the cell–residuals that preserve cubic solutions (and quadratic gradients) as follows:

$$\tilde{\tilde{\Phi}}_u^E = \Phi_u^E - \frac{1}{2} \sum_{i=1}^{3} \left(a_{n_i} \delta_i^u |n_i| - \nu (\delta_i^p n_{x_i} + \delta_i^q n_{y_i}) - \frac{1}{2} \delta_i^s d\Omega^E \right), \tag{81}$$

$$\tilde{\tilde{\Phi}}_p^E = \Phi_p^E + \frac{1}{2T_r} \sum_{i=1}^{3} \left(\delta_i^u n_{x_i} + \frac{1}{2} \delta_i^p d\Omega^E \right), \tag{82}$$

$$\tilde{\tilde{\Phi}}_q^E = \Phi_q^E + \frac{1}{2T_r} \sum_{i=1}^{3} \left(\delta_i^u n_{y_i} + \frac{1}{2} \delta_i^q d\Omega^E \right), \tag{83}$$

which can also be written in the form of an extension to the compact RD–CC2 scheme:

$$\tilde{\tilde{\Phi}}_u^E = \tilde{\Phi}_u^E + \frac{1}{2} \sum_{i=1}^{3} \left(\nu (\delta_i^p n_{x_i} + \delta_i^q n_{y_i}) + \frac{1}{2} \delta_i^s d\Omega^E \right), \tag{84}$$

$$\tilde{\tilde{\Phi}}_p^E = \tilde{\Phi}_p^E + \frac{1}{4T_r} \sum_{i=1}^{3} \delta_i^p d\Omega^E, \tag{85}$$

$$\tilde{\tilde{\Phi}}_q^E = \tilde{\Phi}_q^E + \frac{1}{4T_r} \sum_{i=1}^{3} \delta_i^q d\Omega^E. \tag{86}$$

The curvature correction terms require quadratic LSQ gradients for p, q, and s, but not for u because $p(= u_x)$ and $q(= u_y)$ are already available at nodes. Again, the cell–residuals are distributed by the SUPG

distribution matrix. The hyperbolic RD scheme based on these cell–residuals is referred to as the RD–CC3 scheme. The exactness for cubic solutions has been confirmed numerically as shown in Table 3. The results of a numerical truncation error analysis for the RD–CC3 are provided in Table 4 and compared with the truncation error of RD–CC2.

Table 3. Exactness of the hyperbolic RD equations for exact quadratic and cubic polynomial solutions on irregular grids. The last column indicates whether the exactness ($\checkmark$) is based on cell and/or nodal residuals.

Hyperbolic RD Scheme	Quadratic solution			Cubic solution			Elemental/Nodal
	u eq.	p eq.	q eq.	u eq.	p eq.	q eq.	
RD–CC2	$\checkmark$	$\checkmark$	$\checkmark$	$\times$	$\times$	$\times$	both
RD–CC3	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	both

Table 4. Comparison between numerical truncation error of the nodal residuals (i.e., T.E. $= \frac{1}{d\Omega_j} \sum_j \mathbf{B}^E \mathbf{\Phi}^E$) obtained with the RD–CC2 and RD–CC3 schemes on irregular grids.

Hyperbolic RD Scheme	Nodal Residuals		
	u equation	p equation	q equation
RD–CC2	$O(h)$	$O(h)$	$O(h)$
RD–CC3	$O(h^2)$	$O(h^2)$	$O(h^2)$

The discrete equations are solved by the implicit solver described in Sec. C. We do not attempt to linearize the RD–CC3 scheme exactly, but simply use the exact Jacobian of the compact RD–CC2 scheme. The implicit solver is therefore not precisely Newton's method, but it converges so rapidly that fully converged numerical solutions (i.e., ten orders of magnitude reductions for all residuals) are obtained typically within 10–15 Newton iterations as will be demonstrated later in Sec. VIII.

Remark: The RD–CC2 and RD–CC3 schemes with the SUPG distribution matrix can be considered as economical and powerful alternatives to high–order finite–element methods.[27] They are economical mainly for three reasons: 1) the number of linear relaxations in the implicit solver increasers linearly, not quadratically as typical for diffusion problems, with grid refinement. 2) the second–order advection–diffusion scheme is compact, and thus allows an efficient construction of Newton's method. Furthermore, the same compact Jacobian still yields rapid convergence (comparable to Newton's method) for the third–order scheme. 3) second– and third–order accuracy can be achieved (as will be demonstrated later in Sec. VIII) on linear triangular elements without introducing curved elements. Thus, high–order curved grids are not needed, and these schemes are directly applicable to existing grids composed of linear triangular elements. These schemes are also powerful in that they are capable of producing highly accurate and smooth gradients, to the same order of accuracy as that of the main solution variable u, on isotropic and anisotropic irregular grids. These advantages will be demonstrated in Sec. VIII.

VII. Nonlinear advection–diffusion equation

In this section, we describe the discretization of a nonlinear hyperbolic advection–diffusion equation. As an example, we discuss the construction of the RD–CC3 scheme with the SUPG distribution matrix. Note that the compact RD–CC2 is a subset of RD–CC3 scheme and therefore the discussion here also applies to the RD–CC2 scheme. A similar procedure can be applied to other distribution schemes. Throughout the discussion, we only consider the non–unified approach, which is applicable to more complex systems such as the compressible Navier–Stokes equations.

A. Nonlinear hyperbolic advection–diffusion equation

Consider the following two–dimensional nonlinear advection–diffusion equation:

$$\partial_t u + \partial_x f + \partial_y g = \partial_x(\nu \partial_x u) + \partial_y(\nu \partial_y u) + \tilde{s}(x, y, u), \tag{87}$$

American Institute of Aeronautics and Astronautics

where f and g are nonlinear functions of u, and $\nu = \nu(u)$. The advection speeds in x and y directions are therefore $a(u) = \partial f / \partial u$ and $b(u) = \partial g / \partial u$, respectively. Using the preconditioned formulation proposed for nonlinear systems in Ref. 7, we construct a hyperbolic system as

$$\partial_\tau u + \partial_x f + \partial_y g \;=\; \partial_x p + \partial_y q + s, \tag{88}$$

$$\frac{T_r}{\nu}\partial_\tau p \;=\; \partial_x u - p/\nu, \tag{89}$$

$$\frac{T_r}{\nu}\partial_\tau q \;=\; \partial_y u - q/\nu, \tag{90}$$

where the variables p and q are, in the pseudo steady state, equivalent to the diffusive fluxes in x and y directions, respectively. As before, the physical time derivative can be incorporated into the source term s as done in Ref. 13, but we focus on the steady equation here. Note that the system reduces to the target equation, i.e., Eq. (87), in the pseudo steady state. The system is written in the vector form as a preconditioned conservative equation, with the preconditioning matrix $\mathbf{P}$:

$$\mathbf{P}^{-1}\frac{\partial \mathbf{U}}{\partial \tau} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = \mathbf{Q}, \tag{91}$$

where

$$\mathbf{P}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & T_r/\nu(u) & 0 \\ 0 & 0 & T_r/\nu(u) \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} s \\ -p/\nu(u) \\ -q/\nu(u) \end{bmatrix}, \tag{92}$$

$$\mathbf{F} = \mathbf{F}^a + \mathbf{F}^d = \begin{bmatrix} f \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -p \\ -u \\ 0 \end{bmatrix}, \quad \mathbf{G} = \mathbf{G}^a + \mathbf{G}^d = \begin{bmatrix} g \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -q \\ 0 \\ -u \end{bmatrix}. \tag{93}$$

The flux Jacobians can be obtained by multiplying both sides of Eq. (91) by the preconditioned matrix $\mathbf{P}$, and arrive at

$$\mathbf{PF}_x = \mathbf{P}\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\mathbf{U}_x = \mathbf{A}\mathbf{U}_x, \quad \mathbf{PG}_y = \mathbf{P}\frac{\partial \mathbf{G}}{\partial \mathbf{U}}\mathbf{U}_y = \mathbf{B}\mathbf{U}_y, \tag{94}$$

where

$$\mathbf{A} = \begin{bmatrix} a(u) & -1 & 0 \\ -\nu(u)/T_r & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} b(u) & 0 & -1 \\ 0 & 0 & 0 \\ -\nu(u)/T_r & 0 & 0 \end{bmatrix}. \tag{95}$$

Note that these preconditioned Jacobian matrices (i.e., $\mathbf{A}$ and $\mathbf{B}$) are slightly different from the ones obtained for the linear hyperbolic advection–diffusion system. Hence, their eigen–structures are also different from those given for the linear system in Sec. II. The differences in the eigen–structures influence the formulation of the distribution matrices as described next. Below, we first describe the baseline RD scheme for the preconditioned nonlinear system, and then upgrade it to higher order by the correction approach.

B.   Baseline RD scheme

The cell–residual for the preconditioned system is evaluated in two steps. First, we evaluate the cell–residuals of the nonlinear equation by integrating right hand side of Eq. (91) over an element $E \in \{E\}$ to get the unpreconditioned cell–residuals $\mathbf{\Psi}$:

$$\mathbf{\Psi}^E = \begin{bmatrix} \Psi_u^E \\ \Psi_p^E \\ \Psi_q^E \end{bmatrix} \;=\; \int_E (-\mathbf{F}_x - \mathbf{G}_y + \mathbf{Q})dxdy,$$

$$\;=\; -\frac{1}{2}\sum_{i=1}^{3}\left(\mathbf{F}_i \hat{n}_{i_x} + \mathbf{G}_i \hat{n}_{i_y}\right)|\mathbf{n}_i| + \overline{\mathbf{Q}}^E d\Omega^E, \tag{96}$$

American Institute of Aeronautics and Astronautics

The preconditioned cell–residual, which is distributed to the element vertices, is then defined as

$$\mathbf{\Phi}^E = \begin{bmatrix} \Phi_u^E \\ \Phi_p^E \\ \Phi_q^E \end{bmatrix} = \overline{\mathbf{P}}\mathbf{\Psi}^E, \tag{97}$$

where $\overline{\mathbf{P}}$ is evaluated by the arithmetic average of the solution $\mathbf{U}$ within the element. The matrix $\mathbf{K}_i$ corresponding to the preconditioned system is defined by

$$\begin{aligned} \mathbf{K}_i &= \frac{1}{2}\overline{\mathbf{P}}\left(\overline{\frac{\partial \mathbf{F}}{\partial \mathbf{U}}}\hat{n}_{i_x} + \overline{\frac{\partial \mathbf{G}}{\partial \mathbf{U}}}\hat{n}_{i_y}\right)|\mathbf{n}_i|, \\ &= \mathbf{K}_i^a + \mathbf{K}_i^d, \end{aligned} \tag{98}$$

where the flux Jacobians are evaluated by the arithmetic average of the solution, and $\mathbf{K}_i^a$ and $\mathbf{K}_i^d$ are, respectively, the contributions from the advection and diffusion components, defined as

$$\mathbf{K}_i^a = \frac{1}{2}\overline{\mathbf{P}}\left(\overline{\frac{\partial \mathbf{F}^a}{\partial \mathbf{U}}}n_{i_x} + \overline{\frac{\partial \mathbf{G}^a}{\partial \mathbf{U}}}n_{i_y}\right) = \frac{1}{2}\mathbf{A}_{n_i}^a|\mathbf{n}_i|, \tag{99}$$

$$\mathbf{K}_i^d = \frac{1}{2}\overline{\mathbf{P}}\left(\overline{\frac{\partial \mathbf{F}^d}{\partial \mathbf{U}}}n_{i_x} + \overline{\frac{\partial \mathbf{G}^d}{\partial \mathbf{U}}}n_{i_y}\right) = \frac{1}{2}\mathbf{A}_{n_i}^d|\mathbf{n}_i|, \tag{100}$$

where

$$\mathbf{A}_{n_i}^a = \begin{bmatrix} a_{n_i} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{A}_{n_i}^d = \begin{bmatrix} 0 & -\hat{n}_{i_x} & -\hat{n}_{i_y} \\ -\overline{\nu}\hat{n}_{i_x}/T_r & 0 & 0 \\ -\overline{\nu}\hat{n}_{i_y}/T_r & 0 & 0 \end{bmatrix}, \tag{101}$$

and $\overline{\nu}$ is the arithmetic average of the diffusion coefficient over the element, and

$$a_n = \frac{\partial f}{\partial u}\hat{n}_x + \frac{\partial g}{\partial u}\hat{n}_y. \tag{102}$$

It is clear that the eigenvalues of the advective and diffusive components of the nonlinear hyperbolic advection–diffusion system are equivalent to the linear system:

$$\lambda^a = a_n, \quad \lambda_1^d = -\sqrt{\frac{\overline{\nu}}{T_r}}, \quad \lambda_2^d = +\sqrt{\frac{\overline{\nu}}{T_r}}. \tag{103}$$

Similarly to the linear advection–diffusion system, we construct the SUPG distribution matrix of the nonlinear system over the element E as:

$$\mathbf{B}_i^{SUPG} = \frac{1}{3}\mathbf{I} + \mathbf{D}_i^a + \mathbf{D}_i^d, \tag{104}$$

where

$$\mathbf{D}_i^a = \begin{bmatrix} d_i^{\text{SUPG}} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad d_i^{\text{SUPG}} = \frac{1}{2}\frac{\overline{a}_{n_i}|n_i|}{\sum_{l=1}^{3}\max(0, \overline{a}_{n_l})|n_l|}, \tag{105}$$

and

$$\mathbf{D}_i^d = \frac{1}{2}\frac{\mathbf{K}_i^d}{\sum_{l=1}^{3}\mathbf{K}_l^{d+}}, \quad \mathbf{K}_i^d = \frac{1}{2}\mathbf{A}_{n_i}^d|\mathbf{n}_i|, \tag{106}$$

where $\mathbf{K}_i^{d+}$ is constructed based on the projection of the $\mathbf{A}_n^d$ onto the λ_2^d running wave. We note here that the diffusive flux Jacobian matrix $\mathbf{A}_n^d$ has the following eigenvectors:

$$\mathbf{R}_n = \begin{bmatrix} -1/\lambda_1 & -1/\lambda_2 & 0 \\ \hat{n}_x & \hat{n}_x & -\hat{n}_y \\ \hat{n}_y & \hat{n}_y & \hat{n}_x \end{bmatrix}, \quad \mathbf{L}_n = \frac{1}{\lambda_1 - \lambda_2}\begin{bmatrix} \lambda_1\lambda_2 & \lambda_1\hat{n}_x & \lambda_1\hat{n}_y \\ -\lambda_1\lambda_2 & -\lambda_2\hat{n}_x & -\lambda_2\hat{n}_y \\ 0 & -(\lambda_1 - \lambda_2)\hat{n}_y & (\lambda_1 - \lambda_2)\hat{n}_x \end{bmatrix}, \tag{107}$$

American Institute of Aeronautics and Astronautics

which is different from the linear case. Thus, the projection matrix $\mathbf{\Pi}_2$ for the nonlinear case becomes

$$\mathbf{\Pi}_2 = \frac{1}{\lambda_1 - \lambda_2} \begin{bmatrix} \lambda_1 & \hat{n}_x & \hat{n}_y \\ -\lambda_1 \lambda_2 \hat{n}_x & -\lambda_2 \hat{n}_x^2 & -\lambda_2 \hat{n}_x \hat{n}_y \\ -\lambda_1 \lambda_2 \hat{n}_y & -\lambda_2 \hat{n}_x \hat{n}_y & -\lambda_2 \hat{n}_y^2 \end{bmatrix}, \tag{108}$$

and the $\mathbf{K}_i^{d+}$ can be expressed as

$$\mathbf{K}_i^{d+} = \frac{1}{2} \lambda_{2,i}^d \mathbf{\Pi}_{2,i} |\mathbf{n}_i|. \tag{109}$$

The residual at a node j is then defined in the form of Eq. (10). This completes the construction of the baseline RD scheme for the nonlinear advection–diffusion equation.

C. RD–CC2 and RD–CC3 schemes

The high–order schemes, RD–CC2 and RD–CC3, can be constructed by improving the cell–residuals with the high–order curvature correction approach discussed in Sec. V and VI. A special attention should be made, however, in constructing the high–order hyperbolic RD schemes for nonlinear equations: the Δp and Δq terms in Eq. (71) require the solution gradients, while p and q variables in the nonlinear formulation are the diffusive fluxes. With this in mind, the construction is straightforward.

The cell–residuals of the RD–CC3 scheme for the nonlinear advection–diffusion equation are given by

$$\tilde{\tilde{\Phi}}_u^E = \Phi_u^E - \frac{1}{2} \sum_{i=1}^{3} \left[\left(\delta_i^f - \delta_i^p \right) n_{x_i} + \left(\delta_i^g - \delta_i^q \right) n_{y_i} + \frac{1}{2} \delta_i^s d\Omega^E \right], \tag{110}$$

$$\tilde{\tilde{\Phi}}_p^E = \Phi_p^E + \frac{\overline{\nu}}{2T_r} \sum_{i=1}^{3} \left(\delta_i^u n_{x_i} + \frac{1}{2} \delta_i^{p/\nu} d\Omega^E \right), \tag{111}$$

$$\tilde{\tilde{\Phi}}_q^E = \Phi_q^E + \frac{\overline{\nu}}{2T_r} \sum_{i=1}^{3} \left(\delta_i^u n_{y_i} + \frac{1}{2} \delta_i^{q/\nu} d\Omega^E \right), \tag{112}$$

where δ_i^s, δ_i^p and δ_i^q are given by Equations (78), (79) and (80), respectively, and

$$\delta_i^f = \frac{1}{6} \left(\Delta f_{x_i} \Delta x_i + \Delta f_{y_i} \Delta y_i \right), \tag{113}$$

$$\delta_i^g = \frac{1}{6} \left(\Delta g_{x_i} \Delta x_i + \Delta g_{y_i} \Delta y_i \right), \tag{114}$$

$$\delta_i^u = \frac{1}{6} \left(\Delta (p/\nu)_i \Delta x_i + \Delta (q/\nu)_i \Delta y_i \right), \tag{115}$$

$$\delta_i^{p/\nu} = \frac{1}{6} \left(\Delta (p/\nu)_{x_i} \Delta x_i + \Delta (p/\nu)_{y_i} \Delta y_i \right), \tag{116}$$

$$\delta_i^{q/\nu} = \frac{1}{6} \left(\Delta (q/\nu)_{x_i} \Delta x_i + \Delta (q/\nu)_{y_i} \Delta y_i \right). \tag{117}$$

For the flux gradients in the curvature correction terms for f and g, we employ the chain rule to avoid the LSQ gradient reconstruction:

$$f_x = \frac{\partial f}{\partial u} u_x = \frac{\partial f}{\partial u} (p/\nu), \quad f_y = \frac{\partial f}{\partial u} u_y = \frac{\partial f}{\partial u} (q/\nu), \tag{118}$$

and similarly for the flux g. The gradients of s, p, q, p/ν, q/ν, and the source term, s, are computed by the quadratic LSQ fit. The RD–CC3 scheme is obtained by distributing the above residuals by the SUPG distribution matrix. The RD–CC2 scheme is obtained simply by removing δ_i^p, δ_i^q, δ_i^s, $\delta_i^{p/\nu}$, and $\delta_i^{q/\nu}$ terms from the cell–residuals. Consequently, the RD–CC2 scheme does not require any LSQ gradient reconstruction, and therefore it is compact as in the linear case. We confirmed by numerical experiments that the RD–CC2 and RD–CC3 schemes preserve exact quadratic and cubic solutions, respectively, by the method of manufactured solutions.

VIII. Results

In this section, we verify and examine the accuracy of the hyperbolic RD schemes. In all the test problems, we specify the exact solution u on the boundaries, and initialize all the variables in the interior of the domain with a zero value, and solve the hyperbolic system for u, p, and q. The linear relaxation is performed with a Gauss–Seidel (GS) algorithm to reduce the linear residual by five orders of magnitude with the maximum of 1000 relaxations. Typically, it takes 300–600 relaxations with the under–relaxation parameter in the range of 0.5 to 0.8. The implicit solver is taken to be converged at ten orders of magnitude residual reduction for all equations; this is obtained typically with 3–5 Newton iterations for the RD–CC2 scheme, and 10–15 Newton iterations for the RD–CC3 scheme.

Simulations were performed using both the *unified* and the *non–unified* approaches discussed in Sec. III. The convergence speed and the order of accuracy results were found to be nearly identical with either of the two approaches. Therefore, we only present the results obtained with the *non–unified* approach.

We also note that for some anisotropic, stretched grids, we could not obtain solution with LDA because the GS linear solver was unstable. A more efficient and stable linear solver is needed, particularly for the LDA scheme, but we did not pursue this. Therefore, we only focus on the hyperbolic RD schemes with SUPG distribution method.

A. Linear advection–diffusion problem

For the linear advection–diffusion equation (1), we verify the order of accuracy and the quality of the predicted solution and solution gradients for isotropic and anisotropic irregular triangular grids, and for a problem with curved boundaries.

1. Isotropic Grids

Consider a steady two–dimensional linear advection–diffusion equation, Eq. (1), with $\tilde{s} = 0$. This equation has an exact exponential solution[28] of the form:

$$u(x,y) = C \cos(A\pi\eta) \exp\left(\frac{1 - \sqrt{1 + 4A^2\pi^2\nu^2}}{2\nu}\xi\right),\tag{119}$$

where A and C are arbitrary constants, and $\xi = ax + by$ and $\eta = bx - ay$. This solution is smooth and has no singularly on the boundariesand therefore it is appropriate for the order of accuracy verification.

A series of isotropic irregular grids was generated and the accuracy of the proposed high–order schemes is verified against the baseline RD and RD–JCP2010 schemes. The first three coarse grids are shown in Fig. 5. Error convergence results are shown in Fig. 6 along with a more detailed information in Tables 5–7. For the variable u, all second-order schemes give truly second–order accuracy, and the RD–CC3 scheme gives fourth-order accuracy. For the gradient variables, p and q, on the other hand, the RD–CC2 and RD–CC3 schemes yield, respectively, second– and third–order accuracy, but the baseline RD and RD–JCP2010 do not give truly second–order accuracy.

Table 5. $L1$ **error convergence of** u **for hyperbolic RD schemes with the SUPG distribution on irregular grids** ($A = 2$, $C = -1$, $a = 2$, $b = 1$, $\nu = 0.01$).

Grids	RD	Order	RD–JCP2010	Order	RD–CC2	Order	RD–CC3	Order
32×32	5.66E-05	–	6.98E-05	–	1.91E-05	–	2.31E-06	–
64×64	1.08E-05	2.39	1.31E-05	2.41	4.11E-06	2.22	1.43E-07	4.01
128×128	2.47E-06	2.13	2.82E-06	2.22	9.87E-07	2.06	1.02E-08	3.81
256×256	5.73E-07	2.11	6.18E-07	2.19	2.38E-07	2.02	6.94E-10	3.88

Iterative convergence results are provided in Table 8 with the total number of Newton iterations and the number of GS linear relaxations per Newton iteration. As expected, the implicit solver, which is Newton's method for the baseline, the RD–JCP2010 and the RD–CC2 schemes, converges with 3–5 Newton iterations. For the RD–CC3 scheme, the implicit solver, which is not truly Newton's method, converges also very
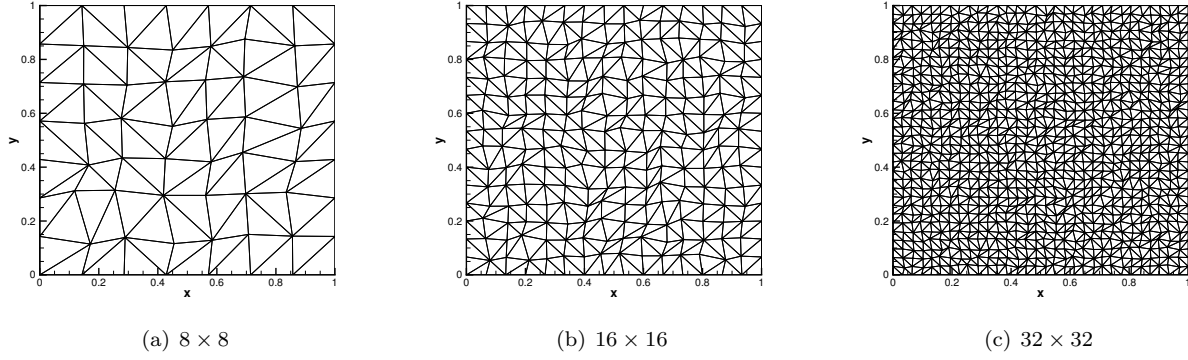
(a) 8×8

(b) 16×16

(c) 32×32

Figure 5. Samples of the irregular grids used in this study.

(a) u
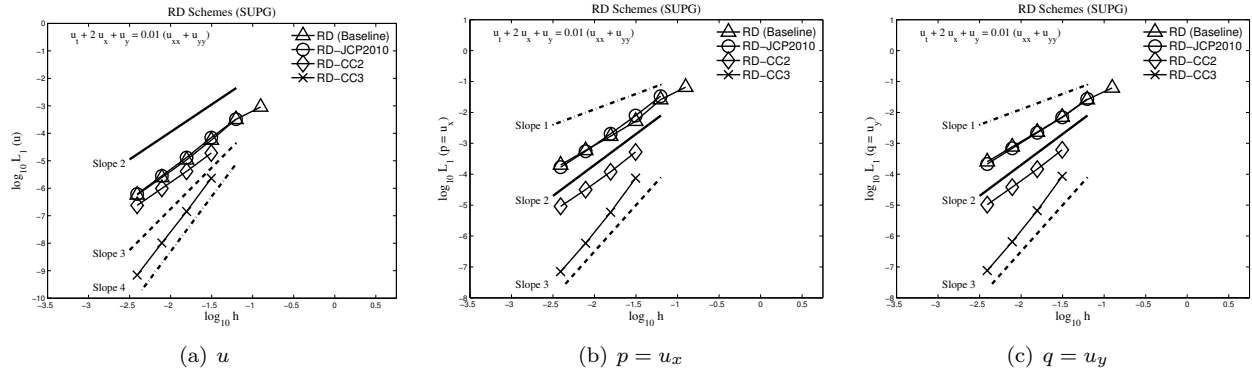
(b) $p = u_x$

(c) $q = u_y$

Figure 6. Order of accuracy comparisons of the baseline RD, RD-JCP2010, RD–CC2, and RD–CC3 schemes.

Table 6. $L1$ **error convergence of** p $(= u_x)$ **for hyperbolic RD schemes with the SUPG distribution on irregular grids** ($A = 2$, $C = -1$, $a = 2$, $b = 1$, $\nu = 0.01$)**.**

Grids	RD	Order	RD–JCP2010	Order	RD–CC2	Order	RD–CC3	Order
32×32	5.19E-03	–	7.87E-03	–	5.25E-04	–	7.43E-05	–
64×64	1.70E-03	1.61	2.04E-03	1.94	1.20E-04	2.13	5.88E-06	3.66
128×128	5.88E-04	1.53	5.54E-04	1.88	3.18E-05	1.92	5.85E-07	3.33
256×256	1.99E-04	1.56	1.69E-04	1.71	9.14E-06	1.80	7.06E-08	3.05

Table 7. $L1$ **error convergence of** q $(= u_y)$ **for hyperbolic RD schemes with the SUPG distribution on irregular grids** ($A = 2$, $C = -1$, $a = 2$, $b = 1$, $\nu = 0.01$)**.**

Grids	RD	Order	RD–JCP2010	Order	RD–CC2	Order	RD–CC3	Order
32×32	7.04E-03	–	6.93E-03	–	6.16E-04	–	8.58E-05	–
64×64	2.32E-03	1.60	2.20E-03	1.65	1.46E-04	2.08	6.61E-06	3.70
128×128	7.67E-04	1.60	6.91E-04	1.67	3.82E-05	1.93	6.50E-07	3.35
256×256	2.54E-04	1.59	2.14E-04	1.69	1.03E-05	1.89	7.62E-08	3.09

American Institute of Aeronautics and Astronautics

Table 8. Iterative convergence of the various hyperbolic RD schemes on irregular grids ($A = 2$, $C = -1$, $a = 2$, $b = 1$, $\nu = 0.01$). Solutions are considered converged when the minimum of ten orders of magnitude reduction is achieved for all the nodal-residuals. The Gauss–Seidel relaxation is performed for each Newton iteration until at least five orders of magnitude reduction is obtained in the linear residuals.

Grids	RD		RD–JCP2010		RD–CC2		RD–CC3	
	Newton (Nt)	GS/Nt	Newton	GS/Nt	Newton	GS/Nt	Newton	GS/Nt
32×32	3	68	3	48	3	72	10	76
64×64	3	103	3	69	3	76	10	83
128×128	3	165	3	111	3	118	10	170
256×256	3	273	3	190	3	249	10	361

rapidly in 10 Newton iterations. Note also that asymptotically the number of linear relaxations increases linearly (not quadratically as typically in conventional methods) with the grid sizes. This is a feature of the hyperbolic method, arising from the elimination of the second derivatives in the target equation.[5]

To demonstrate the high quality of the predicted solution gradients within the domain by the improved RD–CC2 and RD–CC3 schemes, we show results for the 32×32 irregular grid shown in Fig. 5b. Accurate prediction of the solution gradients within the domain is of great interest in many applications (e.g. turbulent flows, reacting flows, etc.). Figure 7 shows the solution gradients predicted by the baseline RD and RD–JCP2010 schemes, and compares them with the exact solutions. As can be seen clearly, the predicted results

(a) u_x; RD (baseline) (b) u_x; RD–JCP2010 (c) u_x (exact)

(d) u_y; RD (baseline) (e) u_y; RD–JCP2010 (f) u_y (exact)
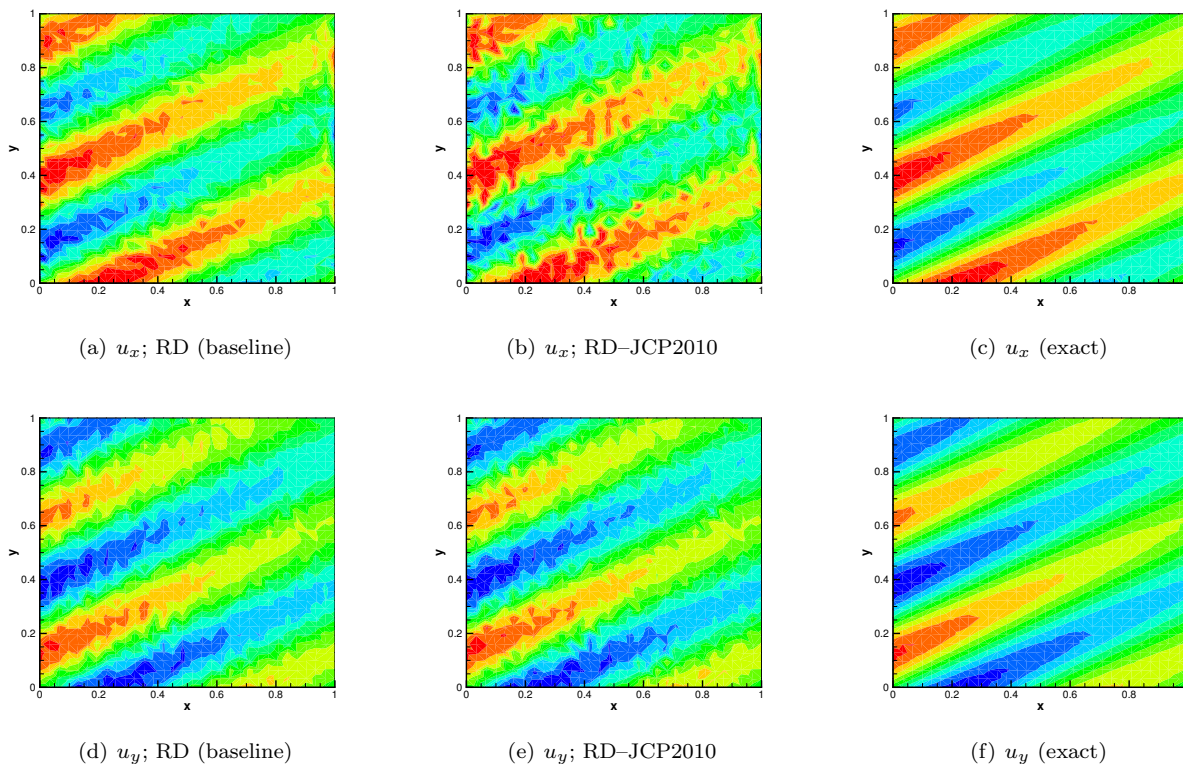
Figure 7. Comparisons between solution gradients (u_x and u_y) predicted by the baseline and RD–JCP2010 schemes on irregular 32×32 grid (see Fig. 5b) with the exact values ($A = 2$, $C = -0.009$, $a = 2$, $b = 1$, $\nu = 0.01$).

are very oscillatory and inaccurate. On the other hand, as shown in Fig. 8, the gradients predicted by the proposed RD–CC2 and RD–CC3 schemes are very accurate and smooth. Note that, our numerical experiments suggest that in some cases the noise produced by the baseline RD and RD–JCP2010 schemes will not completely disappear in the entire domain even on a highly refined grid.

American Institute of Aeronautics and Astronautics

(a) u_x; RD–CC2

(b) u_x; RD–CC3

(c) u_x (exact)

(d) u_y; RD–CC2

(e) u_y; RD–CC3

(f) u_y (exact)
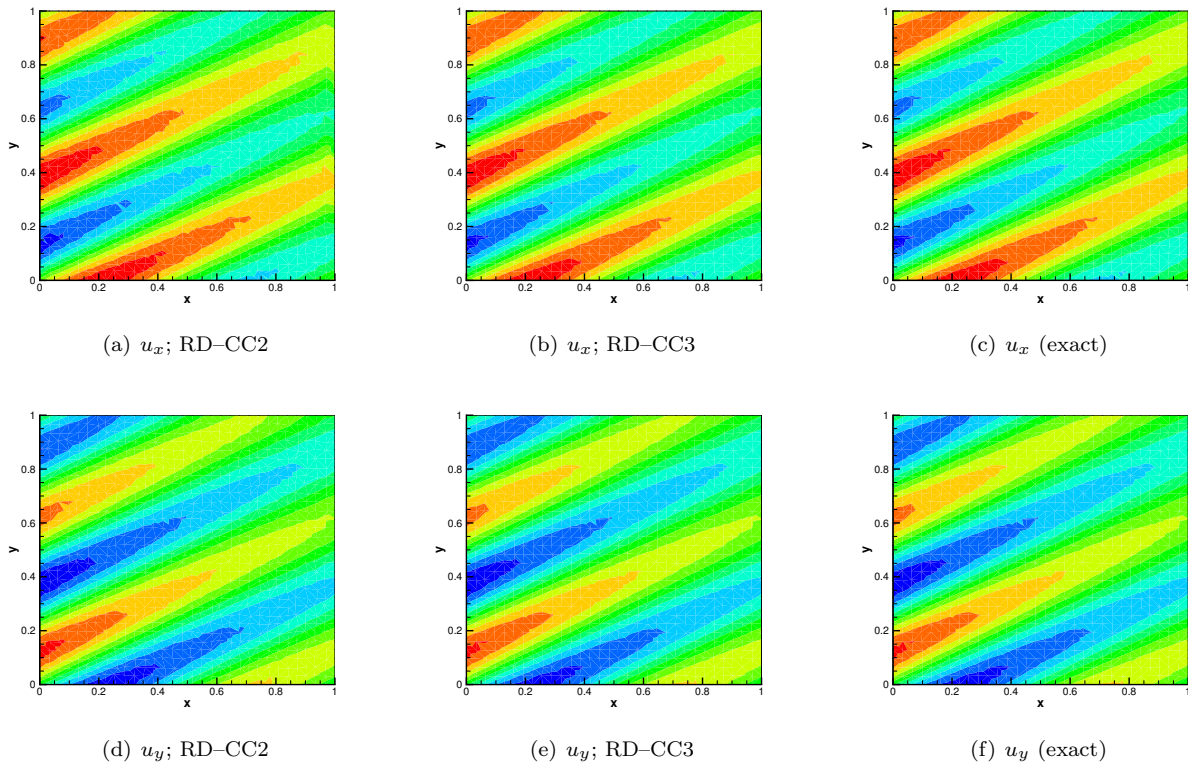
Figure 8. Comparisons between solution gradients (u_x and u_y) predicted by the RD–CC2 and RD–CC3 schemes on irregular 32×32 grid (see Fig. 5b) with the exact values ($A = 2$, $C = -0.009$, $a = 2$, $b = 1$, $\nu = 0.01$).

Shown in Figs. 9 and 10 are the reconstructed gradients computed by the quadratic LSQ fits from the solution of the RD–CC3 scheme, compared with the gradients predicted directly with the RD–CC3 scheme. Clearly the reconstructed gradients are less accurate and deteriorate even more within the domain. This comparison emphasizes the difficultly in obtaining accurate solution gradients by reconstruction techniques, a common practice with conventional schemes.

(a) Reconstructed; u_x

(b) Predicted; u_x

(c) Exact; u_x

(d) u_x at $x = 0$
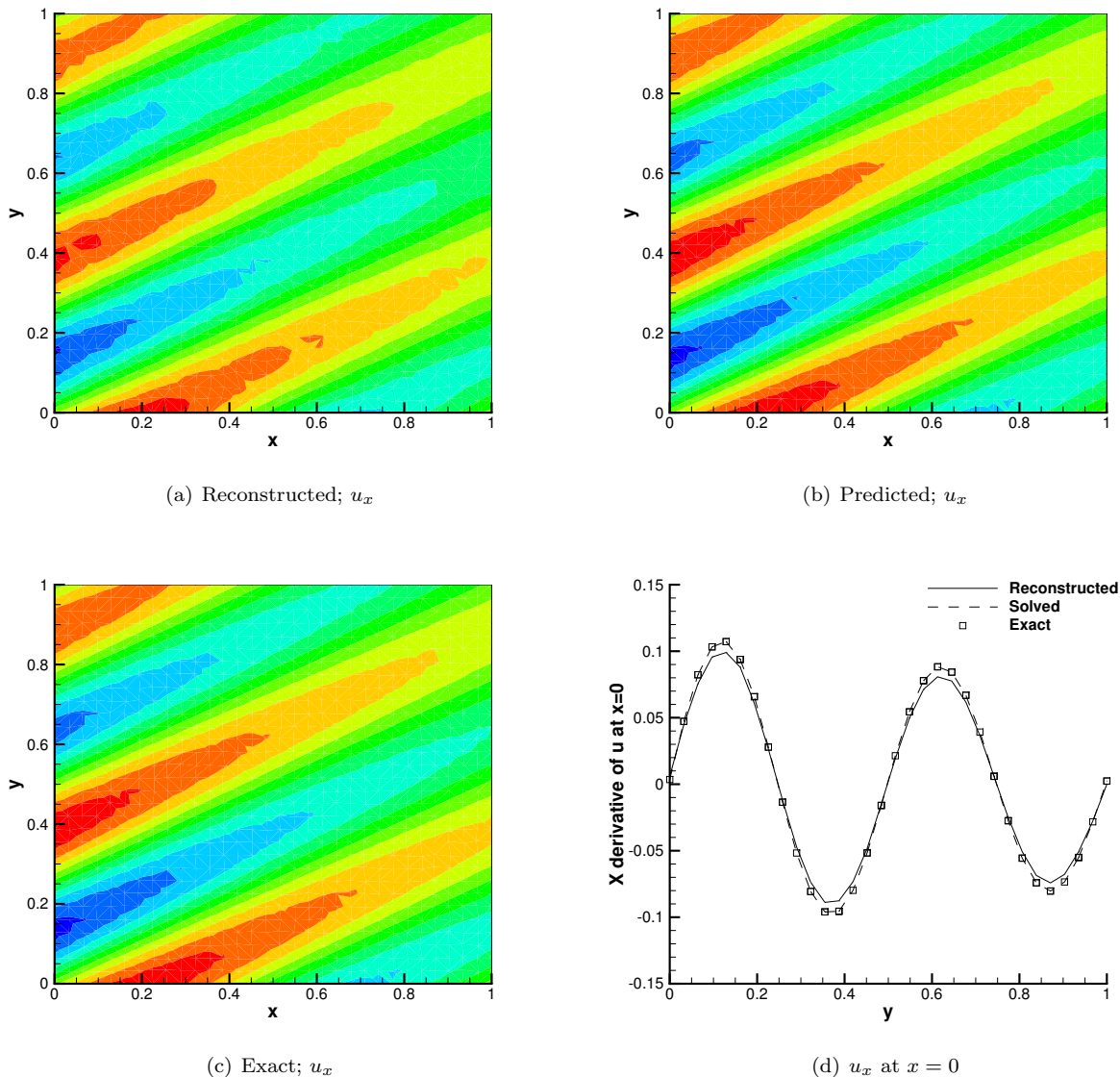
Figure 9. Comparison of u_x reconstructed with the quadratic LSQ fit from the solution of the RD–CC3 scheme and the one predicted directly with the RD–CC3 scheme on irregular 32×32 grid in Fig. 5b. ($A = 2$, $C = -0.009$, $a = 2$, $b = 1$, $\nu = 0.01$).

2. Uniform accuracy over $Re = a/\nu$

In this section we demonstrate uniform accuracy predicted by the RD–CC2 and RD–CC3 schemes for both the solution and the solution gradients over ranges of Re with different irregular grids. These results are shown in Fig. 11, which verifies that the RD–CC2 and RD–CC3 schemes produce, respectively, at a minimum, second– and third–order solution for all the variables. Note that in the advection–limit ($\nu \to 0$) the hyperbolic advection–diffusion system reduces to simple scalar advection equation, for which the RD–CC2 scheme is third–order accurate for all the variables. In the diffusion–limit ($\nu \to \infty$), the hyperbolic advection–diffusion

American Institute of Aeronautics and Astronautics
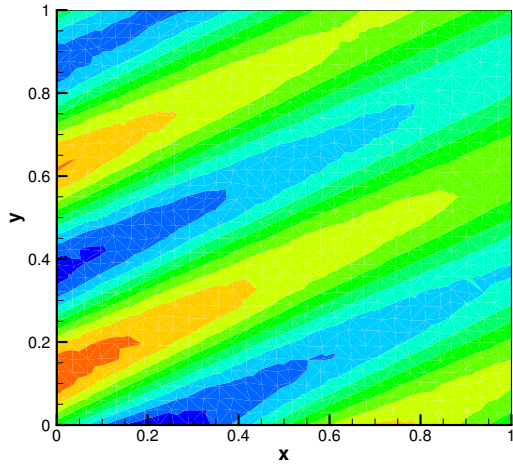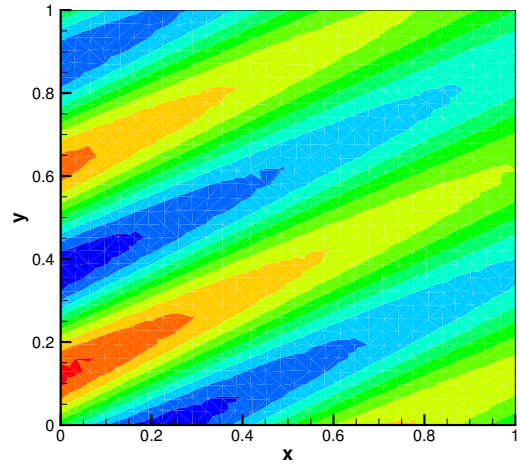
(a) Reconstructed; u_y

(b) Predicted; u_y
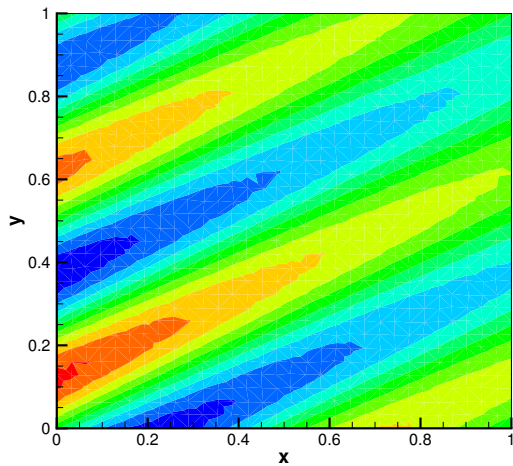
(c) Exact; u_y
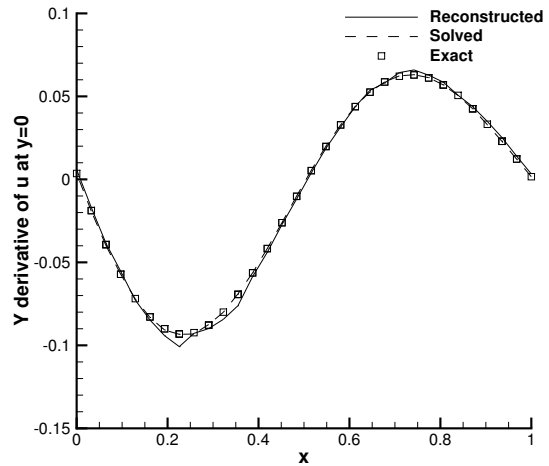
(d) u_y at $y = 0$

Figure 10. Comparison of u_y reconstructed with the quadratic LSQ fit from the solution of the RD–CC3 scheme and the one predicted directly with the RD–CC3 scheme on irregular 32×32 grid in Fig. 5b. ($A = 2$, $C = -0.009$, $a = 2$, $b = 1$, $\nu = 0.01$).

system becomes hyperbolic diffusion system, for which the RD–CC3 schemes is fourth–order for u but remains third–order for the solution gradients.

(a) RD–CC2; u (b) RD–CC2; u_x (c) RD–CC2; u_y

(d) RD–CC3; u (e) RD–CC3; u_x (f) RD–CC3; u_y
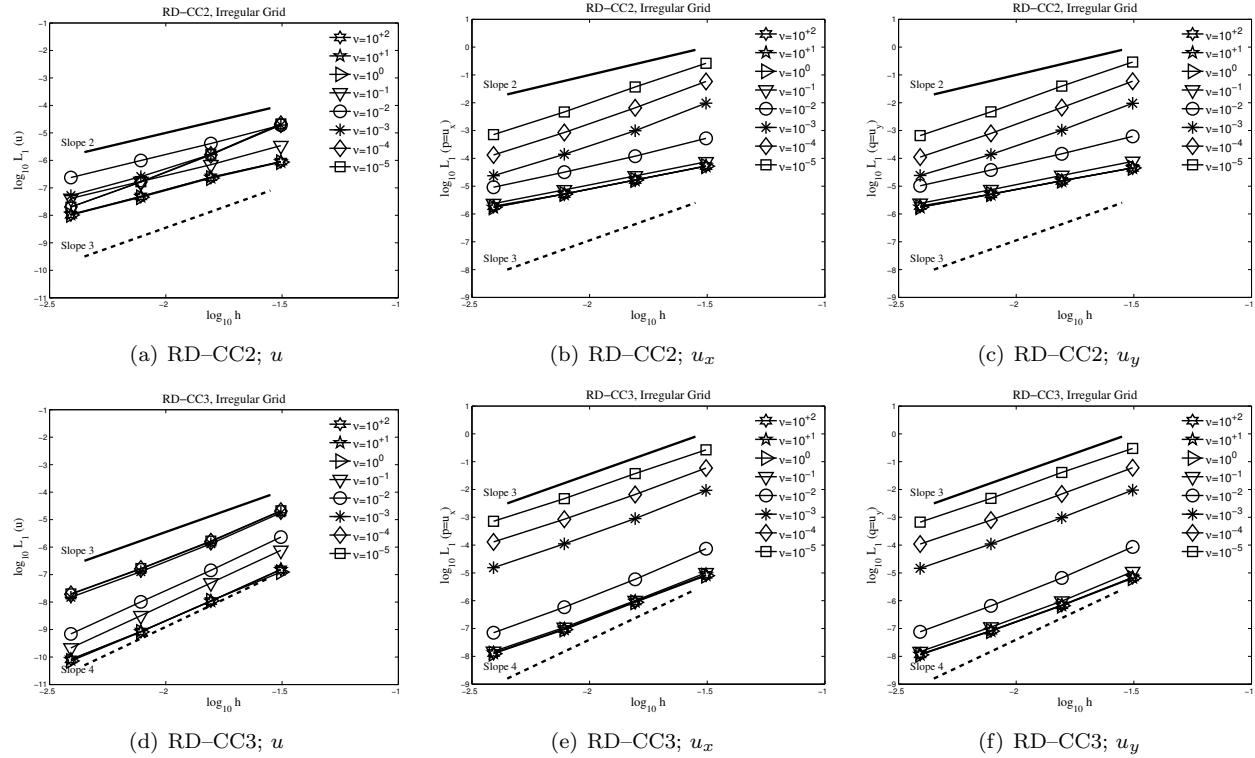
Figure 11. Uniform accuracy over ranges of Re with the proposed RD–CC2 and RD–CC3 schemes on irregular grids ($A = 2$, $C = -0.009$, $a = 2$, $b = 1$, $Re = \sqrt{5}/\nu$).

Figure 12 shows Newton iteration versus Re for both RD–CC2 and RD–CC3 schemes. It is clear that the number of Newton iterations is independent of the Re and grid sizes. We remark that some instabilities were observed with the linear relaxation for high–Re cases ($Re \geq 10^3$), but found that both RD–CC2 and RD–CC3 schemes become stable if the curvature correction term applied to the advection fluxes (i.e., δ_i^u), is evaluated with quadratic LSQ reconstructions; that is, u_x and u_y variables present in the δ_i^u term (see Eq. 71) are reconstructed from the solution variable u, instead of being substituted by the gradient variables p and q. The jump in the number of Newton iteration shown in Fig. 12 for RD–CC2 is due to this switch, which is implemented to avoid instability for high–Re cases. We also remark that if the switch is activated for all the Re, the jump will disappear and a similar result shown for RD–CC3 scheme will be recovered.

A linear convergence, $O(N)$, was also obtained for the ranges of Re. This is shown in Fig. 13. For high–Re cases, a better than linear convergence is obtained because of high mesh–Re (i.e., ah/ν) values used with coarser grids. For example, for the $\nu = 10^{-3}$ case ($Re = 1000\sqrt{5}$), linear convergence is obtained from the second coarsest grid (mesh–$Re < 35$). Note that the RD–CC2 and RD–CC3 schemes are efficient enough in converging high–Re cases on grids with high mesh–Re, and therefore we did not attempt to fix a mesh–Re on these cases.

3. High aspect–ratio grids

In practical applications, we are often interested in the gradients at the boundary and with highly anisotropic grids, e.g., viscous drag or heat flux predictions in high–Reynolds–number viscous flows. To demonstrate the benefit of the hyperbolic schemes for such applications, we consider two highly stretched anisotropic irregular grids (see Fig. 14). The y–derivative of u at $y = 0$ is plotted and compared with the exact solution in Fig. 14. The predicted normal gradient is very accurate and smooth on such relatively coarse irregular grids. Similar results have been obtained by other schemes, and therefore not shown.

This exceptional feature of the hyperbolic RD schemes is further illustrated by comparing the normal gradients predicted by the baseline RD scheme and a conventional FV scheme on a high aspect–ratio (AR)
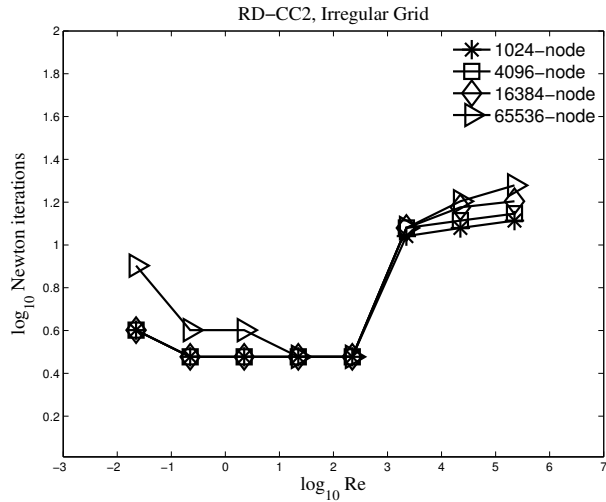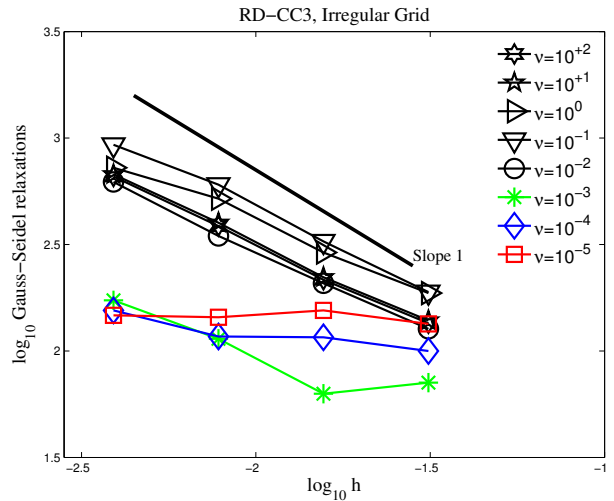
(a) RD–CC2
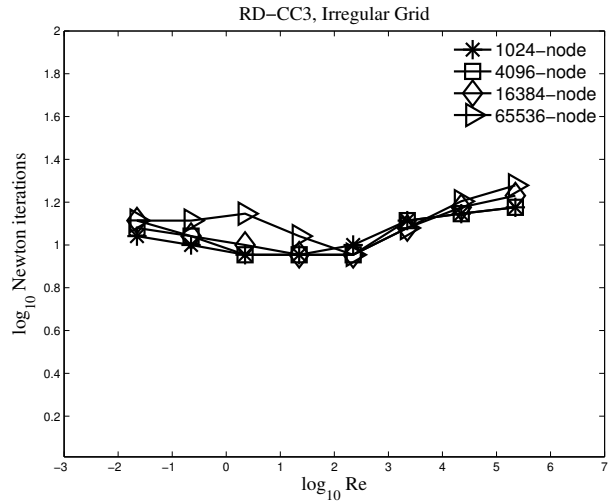
(b) RD–CC3

Figure 12. Newton iterations used by the proposed RD–CC2 and RD–CC3 schemes on irregular grids over ranges of Re **and grids** ($A = 2$, $C = -0.009$, $a = 2$, $b = 1$, $Re = \sqrt{5}/\nu$)**.**
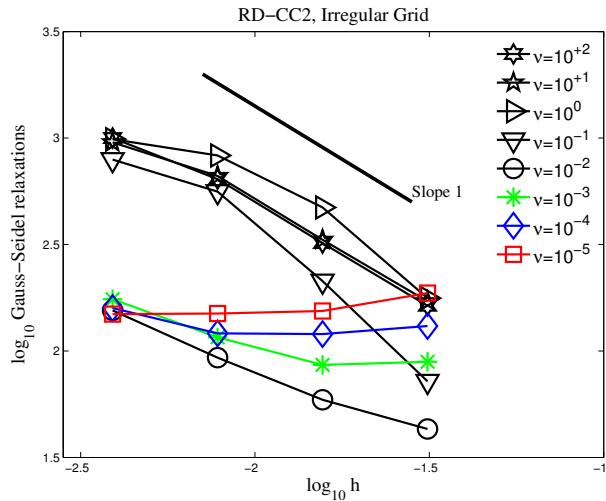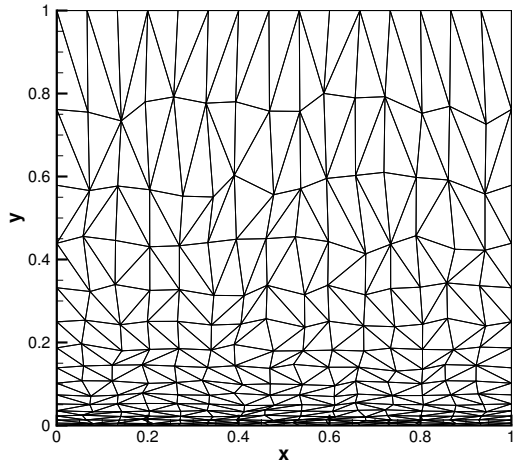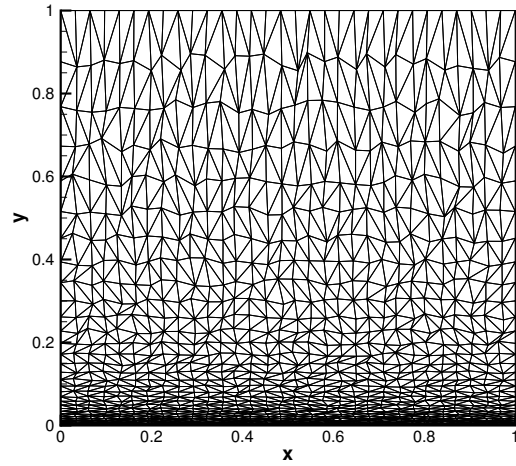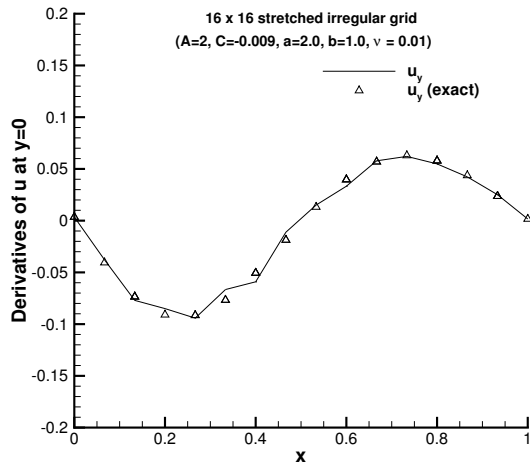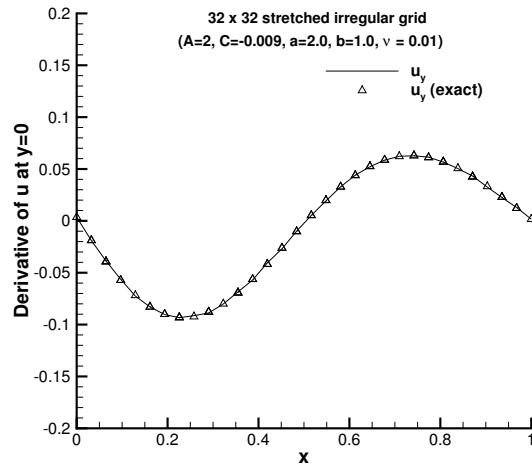

(a) RD–CC2

(b) RD–CC3

Figure 13. Average Gauss-Seidel relaxations per Newton iteration used by the proposed RD–CC2 and RD–CC3 schemes on irregular grids over ranges of Re **and grids** ($A = 2$, $C = -0.009$, $a = 2$, $b = 1$, $Re = \sqrt{5}/\nu$)**.**

(a) 16×16

(b) 32×32

(c) 16×16

(d) 32×32

Figure 14. **Anisotropic, stretched, and irregular grids, and the corresponding predictions of the normal gradients along the** $y = 0$ **boundary with the baseline hyperbolic RD scheme** ($A = 2$, $C = -0.009$, $a = 2$, $b = 1$, $\nu = 0.01$)**.**

American Institute of Aeronautics and Astronautics

grid. The grid and the results are shown in Fig. 15 where the normal gradients at the $y = 0$ and the $x = 0$ boundaries predicted with the hyperbolic RD scheme are compared with those obtained by the linear LSQ reconstruction from the solution of a conventional FV scheme taken from Ref. 8, where the scalar advection–diffusion equation is solved by the second–order edge–based advection scheme and the second–order Galerkin discretization for diffusion (referred to as Galerkin in the figure). The predicted normal derivative using the baseline RD scheme is remarkably smooth and accurate, while the gradients obtained by the conventional scheme shows oscillatory behavior and inaccurate predictions. For this problem also, similar results have been obtained by other schemes, and therefore not shown.

These results indicate that even the baseline RD and RD–JCP2010 schemes can produce accurate gradients on boundaries although they yield oscillations in the interior as shown in the previous section. Also, the results demonstrate that the RD–CC2 and RD–CC3 schemes perform successfully even on irregular high–aspect–ratio grids, which are representatives of adapted viscous grids.

| (a) High AR grid | (b) u_y | (c) u_x |
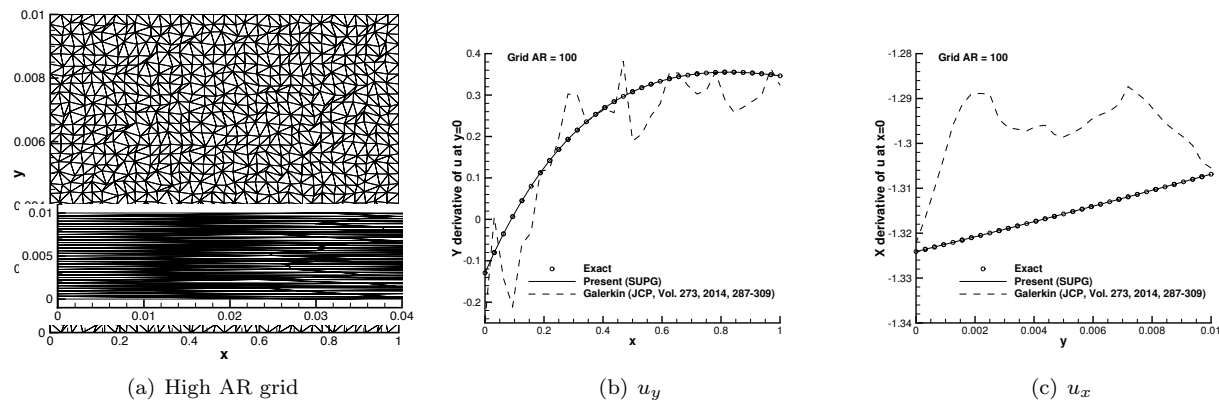
Figure 15. Comparisons between the gradients (u_x and u_y) predicted with the baseline hyperbolic RD scheme and the published result of the conventional Galerkin method[8] on a high aspect-ratio grid with AR $= 100$; the inset figure shows a portion of the grid with 1:1 scaling on both axes.($A = 2$, $C = 1$, $a = 1.23$, $b = 0.12$, $\nu = 0.1235839795$).

B. Domain with curved geometrical boundaries

In this section, we verify the accuracy of the RD–CC2 and RD–CC3 schemes for a problem with a curved boundary. The problem is a potential flow over a unit circle, taken from Ref.[29] The governing equation is the Laplace equation for the stream function, ψ:

$$\partial_{xx}\psi + \partial_{yy}\psi = 0. \tag{120}$$

We solve the above equation by solving the following hyperbolic system of equations:

$$\partial_\tau \psi = \partial_x p + \partial_y q, \quad \partial_\tau p = \frac{1}{T_r}\left(\partial_x \psi - p\right), \quad \partial_\tau q = \frac{1}{T_r}\left(\partial_y \psi - q\right), \tag{121}$$

where the x– and y–velocity components are related to the gradient of the main variable ψ; i.e., $\psi_x = -v$ and $\psi_y = u$.

We discretize the hyperbolic potential flow system of equations by the RD–CC2 and RD–CC3 schemes, described in Secs. V and VI, and verify their predicted orders of accuracy over a set of eight irregular anisotropic triangular grids of 441, 1681, 3721, 6561, 10201, 14641, 25921, and 40401 nodes. The grids are composed of linear elements only, and no curved elements are used. The domain and a sample grid (1681 nodes) is shown in Fig. 16a. We impose the exact solution on the outer domain, and specify $\psi = 0$ at the solid boundary nodes to focus on the order of accuracy in the gradients. Note that the gradients at boundary nodes are predicted by the numerical schemes. Figure 16 shows the contours of ψ_x obtained with the RD–CC3 scheme on the 1681–node grid. The excellent quality of the predicted derivatives is observed in comparison with the exact solution. The predicted streamline is also shown in the same figure.
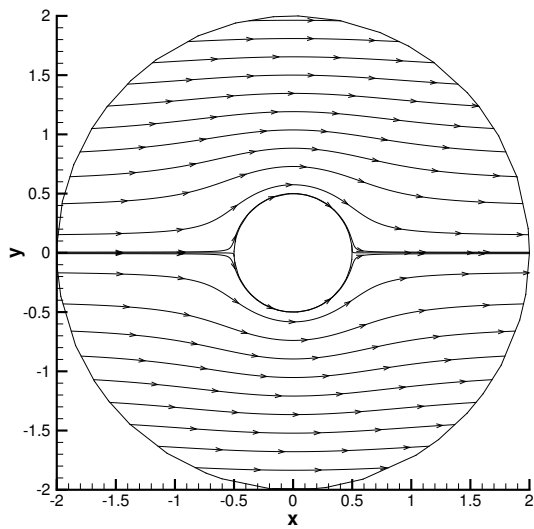
The order of accuracy of theses schemes on predicting solution gradients on curved boundaries is also verified. The error convergence results for the boundary nodes and the entire domain are shown in Fig. 17.

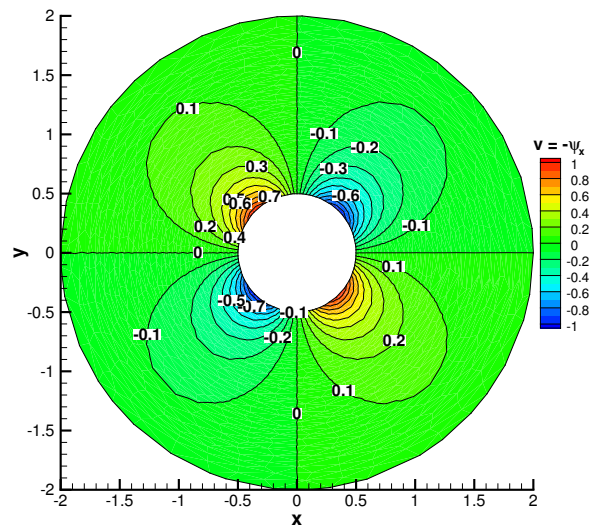American Institute of Aeronautics and Astronautics
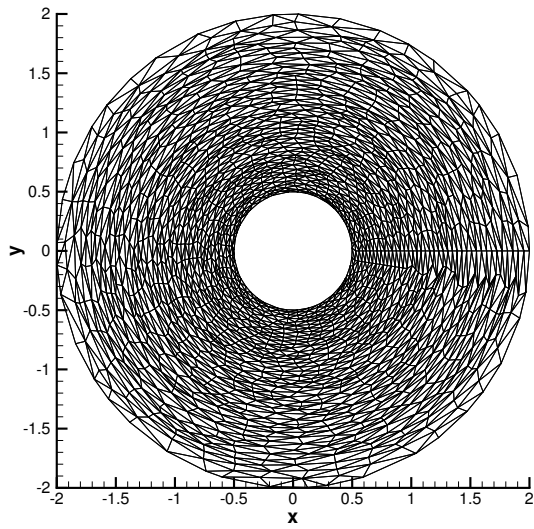
(a) Irregular grid (1681 nodes)
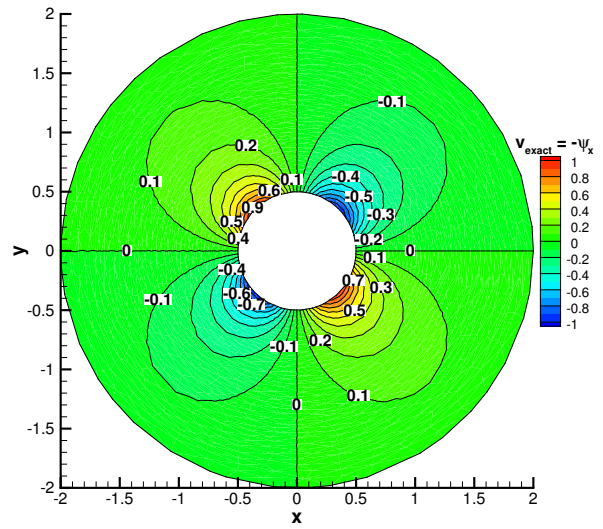
(b) Exact $(-\psi_x)$ on the grid of 1681 nodes

(c) Predicted streamline on the grid of 1681 nodes.

(d) Predicted $(-\psi_x)$ on the grid of 1681 nodes.

Figure 16. Potential flow $(\psi_{xx} + \psi_{yy} = 0)$ over a unit circle on irregular and stretched grids with RD–CC3. The contours show the x–gradient of the solution variable ψ.
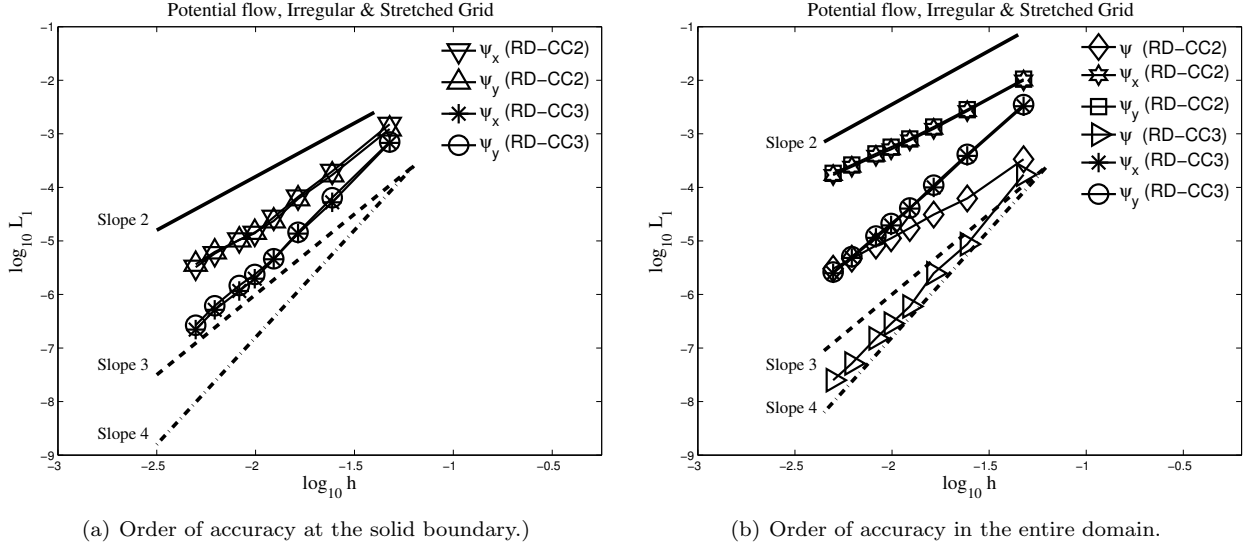
American Institute of Aeronautics and Astronautics

(a) Order of accuracy at the solid boundary.) (b) Order of accuracy in the entire domain.

Figure 17. Potential flow ($\psi_{xx} + \psi_{yy} = 0$) over a unit circle on several irregular and stretched grids with the RD–CC2 and RD–CC3 schemes.

As it is shown, the design order of accuracy is clearly observed through the boundary nodes. It is also evident that the RD–CC3 scheme is more accurate than the RD–CC2 scheme, even on the coarsest grid level, indicating that the third–order scheme, which is constructed for linear elements, is, in fact, solving the correct problem on the curved boundary that is defined only with linear elements.

C.   Nonlinear advection–diffusion problem

Consider the following nonlinear advection–diffusion equation

$$\partial_t u + \partial_x f + \partial_y g = \partial_x(\nu \partial_x u) + \partial_y(\nu \partial_y u) + \tilde{s}(x, y), \tag{122}$$

where $f = g = u^2/2$, $\nu = u$. The source term $\tilde{s}(x, y)$ is defined by

$$\tilde{s}(x, y) = u^e(u_x^e + u_y^e) - (u_x^e)^2 - (u_y^e)^2 - u^e(u_{xx}^e + u_{yy}^e), \tag{123}$$

$$u^e = C \cos(A\pi\eta) \exp\left(\frac{1 - \sqrt{1 + 4A^2\pi^2\nu_0^2}}{2\nu_0}\xi\right) + C_0, \tag{124}$$

where $C = -0.009$, $A = 2.0$, $\nu_0 = 0.01$, and $C_0 = 1$, so that u^e is the exact solution to the nonlinear advection–diffusion equation (122). Note that C_0 must be greater than C in order for the diffusion coefficient to be positive.

We solve the nonlinear advection–diffusion equation in a square domain of $[0, 1] \times [0, 1]$, and verify the order of accuracy of the solution and solution gradients predicted by the baseline RD, the RD–CC2, and the RD–CC3 schemes. The order of accuracy comparison is shown in Fig. 18 along with a more detailed information in Tables 9–11, illustrating that the RD–CC2 and RD–CC3 schemes achieve an improved order of accuracy for both the solution and the solution gradients on irregular grids. On the other hand, the baseline RD scheme suffers from the reduced order of accuracy in the gradients.

IX.   Conclusions

We have presented detailed formulation and implementation procedures for hyperbolic RD schemes for general advection–diffusion problems on arbitrary triangular grids. Improved upon the previous work,[5, 6] we developed new second– and third–order hyperbolic RD schemes that preserve, respectively, quadratic and cubic solutions on arbitrary triangular grids. We also developed a high–order source term discretization, which was found essential in constructing the third–order scheme with linear elements. We showed that the

American Institute of Aeronautics and Astronautics
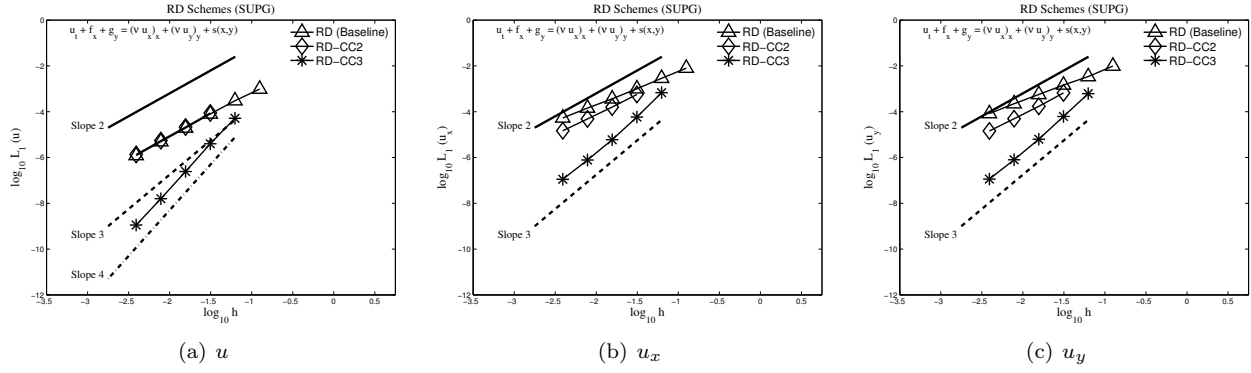
(a) u (b) u_x (c) u_y

Figure 18. Order of accuracy comparisons between the baseline and the proposed hyperbolic RD schemes on irregular grids for the nonlinear advection–diffusion problem ($f = g = u^2/2$, $\nu = u$).

Table 9. L_1 **error convergence of** u **for the hyperbolic RD schemes with the SUPG distribution on irregular grids, for the nonlinear advection–diffusion problem.**

Grids	RD	Order	RD–CC2	Order	RD–CC3	Order
32×32	7.93E-05	–	8.79E-05	–	3.98E-06	–
64×64	1.93E-05	2.04	2.15E-05	2.03	2.43E-07	4.04
128×128	4.90E-06	1.98	5.46E-06	1.98	1.59E-08	3.93
256×256	1.22E-06	2.01	1.36E-06	2.01	1.12E-09	3.83

Table 10. L_1 **error convergence of** u_x **for the hyperbolic RD schemes with the SUPG distribution on irregular grids, for the nonlinear advection–diffusion problem.**

Grids	RD	Order	RD–CC2	Order	RD–CC3	Order
32×32	1.00E-03	–	5.66E-04	–	5.84E-05	–
64×64	3.63E-04	1.46	1.59E-04	1.83	5.94E-06	3.30
128×128	1.44E-04	1.33	4.84E-05	1.72	7.75E-07	2.94
256×256	5.31E-05	1.44	1.47E-05	1.72	1.12E-07	2.79

Table 11. L_1 **error convergence of** u_y **for the hyperbolic RD schemes with the SUPG distribution on irregular grids, for the nonlinear advection–diffusion problem.**

Grids	RD	Order	RD–CC2	Order	RD–CC3	Order
32×32	1.46E-03	–	6.59E-04	–	6.17E-05	–
64×64	5.62E-04	1.38	1.71E-04	1.95	6.28E-06	3.30
128×128	2.23E-04	1.33	4.91E-05	1.80	7.94E-07	2.98
256×256	8.23E-05	1.44	1.44E-05	1.77	1.14E-07	2.80

American Institute of Aeronautics and Astronautics

second– and third–order schemes can be constructed by a separate treatment of the advective and diffusive terms, enabling development of hyperbolic residual–distribution schemes for the compressible Navier–Stokes equations. The superiority of the proposed schemes was further demonstrated with accurate and smooth predictions of the gradients within domains, on both flat and curved boundaries. The new second–order scheme, which is still defined within a compact stencil, allows an efficient construction of Newton's method with the exact residual Jacobian computed by the automatic differentiation. We demonstrated that the resulting implicit solver is capable of reducing the residuals of all equations by at least ten orders of magnitude with typically less than 10 Newton iterations for the second–order scheme. For the third–order scheme, we showed that the linear solver converges also very rapidly with 10–15 Newton iterations with the second–order Jacobian. Because of the hyperbolic reformulation of the advection–diffusion equation, the number of linear relaxations per Newton iteration was also shown to increase linearly, not quadratically as for typical diffusion problems, with grid refinement.

Acknowledgments

A.   Fourier analysis

In this appendix, we present the Fourier analysis to investigate the stability of relaxation schemes applied to the linear system (14). The analysis is performed for the baseline RD scheme only, but the results are suggestive and in fact effective to stabilize other schemes.

Consider a two–dimensional stencil shown in Fig. 19, where we seek the solution at node j, which is
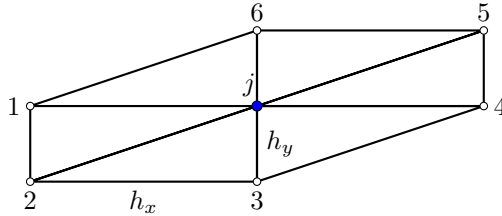
Figure 19. Schematic of the stencil used in Fourier analysis.

surrounded by six elements and and six nodes. The grid aspect radio (AR) is defined as h_x/h_y. We now define the vectors $\mathbf{U}$ and $\mathbf{Q}$ to all the nodes; that is:

$$\mathbf{U}_j = \begin{bmatrix} u_j \\ p_j \\ q_j \end{bmatrix}, \quad \mathbf{U}_k = \begin{bmatrix} u_k \\ p_k \\ q_k \end{bmatrix} \text{ for } k = 1..6, \tag{125}$$

$$\mathbf{Q}_j = \begin{bmatrix} 0 \\ p_j/T_r \\ q_j/T_r \end{bmatrix}, \quad \mathbf{Q}_k = \begin{bmatrix} 0 \\ p_k/T_r \\ q_k/Tr \end{bmatrix} \text{ for } k = 1..6. \tag{126}$$

The cell–residual $\mathbf{\Phi}^T$ then becomes:

$$\mathbf{\Phi}^T = -\sum_{i \in T} \mathbf{K}_i \mathbf{U}_i + \bar{\mathbf{Q}}^T S^T. \tag{127}$$

For example, for the element containing nodes $(j, 1, 2)$, we have:

$$\mathbf{\Phi}^{T \in (j,1,2)} = -\mathbf{K}_j \mathbf{U}_j - \mathbf{K}_1 \mathbf{U}_1 - \mathbf{K}_2 \mathbf{U}_2 + \frac{1}{3}(\mathbf{Q}_j + \mathbf{Q}_1 + \mathbf{Q}_2)\frac{h_x h_y}{2}. \tag{128}$$

American Institute of Aeronautics and Astronautics

We remark that $\mathbf{K}$ is an element dependent matrix and therefore special care must be taken in evaluating the elemental residual $\mathbf{\Phi}^T$.

Using Eq. (11), we evaluate the residual of the node j, noting that the median dual volume S_j for our stencil is simply $h_x h_y$. We now start the Fourier analysis by writing the $\mathbf{Res}_j$ as

$$\mathbf{Res}_j = \mathbf{C}_j \mathbf{U}_j + \sum_{k=1}^{6} \mathbf{C}_k \mathbf{U}_k, \tag{129}$$

where $\mathbf{C}_j$ and $\mathbf{C}_k$ are:

$$\mathbf{C}_j = \frac{\partial \mathbf{Res}_j}{\partial \mathbf{U}_j}, \tag{130}$$

$$\mathbf{C}_k = \frac{\partial \mathbf{Res}_j}{\partial \mathbf{U}_k}. \tag{131}$$

We then transform the residual $\mathbf{Res}_j$ into the corresponding Fourier modes by replacing the vector $\mathbf{U}_k$ with a Fourier mode of the phase angle $\beta = (\beta_x, \beta_y)$ with $\beta \in [0, \pi]$:

$$\mathbf{U}_k = \mathbf{U}_j e^{i\left(\beta_x \frac{x_k - x_j}{h_x} + \beta_y \frac{y_k - y_j}{h_y}\right)}, \tag{132}$$

where $i = \sqrt{-1}$, and (x_j, y_j) and (x_k, y_k) are the nodal coordinates. Equation 129 can now be written as:

$$\mathbf{Res}_j = \mathbf{U}_j \left(\mathbf{C}_j + e^{-i\beta_x} \mathbf{C}_1 + e^{-i(\beta_x + \beta_y)} \mathbf{C}_2 + e^{-i\beta_y} \mathbf{C}_3 + e^{i\beta_x} \mathbf{C}_4 + e^{i(\beta_x + \beta_y)} \mathbf{C}_5 + e^{i\beta_y} \mathbf{C}_6 \right) \tag{133}$$

The solution of $\mathbf{U}_j$ using the Jacobi relaxation is therefore become:

$$\mathbf{U}_j^{n+1} = \mathbf{M}_{JB} \mathbf{U}_j^n \tag{134}$$

where the mass matrix $\mathbf{M}$ is defined as

$$\mathbf{M}_{JB} = \mathbf{C}_j^{-1} \left(e^{-i\beta_x} \mathbf{C}_1 + e^{-i(\beta_x + \beta_y)} \mathbf{C}_2 + e^{-i\beta_y} \mathbf{C}_3 + e^{i\beta_x} \mathbf{C}_4 + e^{i(\beta_x + \beta_y)} \mathbf{C}_5 + e^{i\beta_y} \mathbf{C}_6 \right) \tag{135}$$

Note that this is equivalent to the Jabobi relaxation applied to the linear system (14) without the right hand side, which is irrelevant to the stability analysis.

The three complex eigenvalues of the matrix, $\mathbf{M}_{JB}$, are computed for several RD schemes with grid AR of 1 and 10^6. Here we used 50 equally distributed points in β_x and β_y directions, for a total of 2500 points (i.e., 50×50). We first present the results of the first–order Lax–Friedrichs (or Rusanov) scheme[30] (Fig. 20) for the hyperbolic diffusion problem. The x- and y-axis are the real and imaginary part of the matrix eigenvalues. The colors in the plots correspond to the three eigenvalues. This Rusanov scheme is stable with GS for any regular grid. Note that the imaginary component of the eigenvalues are very small.

Figure 21 shows the results for the hyperbolic diffusion system using LDA scheme. Clearly, the results exhibit modes with magnitudes larger than one, specially on the real axis. These modes indicate that the errors could amplify and may not be damped. Therefore, the LDA scheme is unstable with the Jacobi relaxation for our system equation. Similar results are obtained with the Gauss–Seidel relaxation. The results are consistent with our developed two-dimensional RD code. We remark that while the figures are shown for $AR = 10^6$ are identical to results obtained on uniform grids with arbitrary AR. Similar results are also obtained with a pure hyperbolic problem, the wave equation, $u_{tt} = c^2(u_{xx} + u_{yy})$ (see Fig. 22). The results are nearly identical to the hyperbolic diffusion problem shown in Fig. 21 particularly with $AR = 1$. The results indicate that the instability of the LDA scheme is not caused by the first–order hyperbolic system method. We have also examined the stability of the LDA scheme for a scalar linear advection equation. Note that there is only one eigenvalue for the pure advection problem. These results are presented in Fig. 23, which indicate non-damping modes specially for grids with $AR = 1$. The results confirm that the presented instability is not due to the formulation of the diffusion problems using hyperbolic first–order system.

The SUPG, LW, and a stabilized–LDA schemes also exhibit similar instability with the GS relaxation for both diffusion and scalar linear advection problems (see Figs. 24-28). Note that the stabilized–LDA scheme consists of the LDA scheme and a stabilization term defined as $\mathbf{K}$.
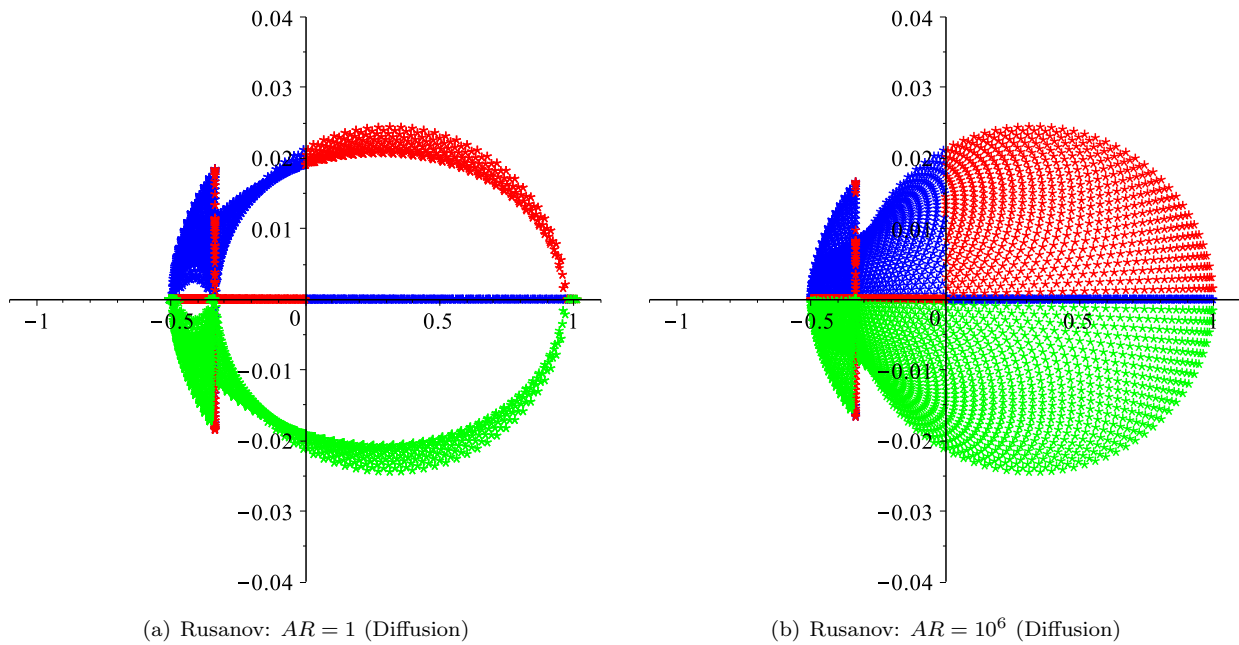
American Institute of Aeronautics and Astronautics

(a) Rusanov: $AR = 1$ (Diffusion)

(b) Rusanov: $AR = 10^6$ (Diffusion)

Figure 20. Fourier analysis of the Rusanov scheme for the hyperbolic diffusion problem: $\nu = 1$, $h_x = 0.1$.

(a) LDA: $AR = 1$ (Diffusion)
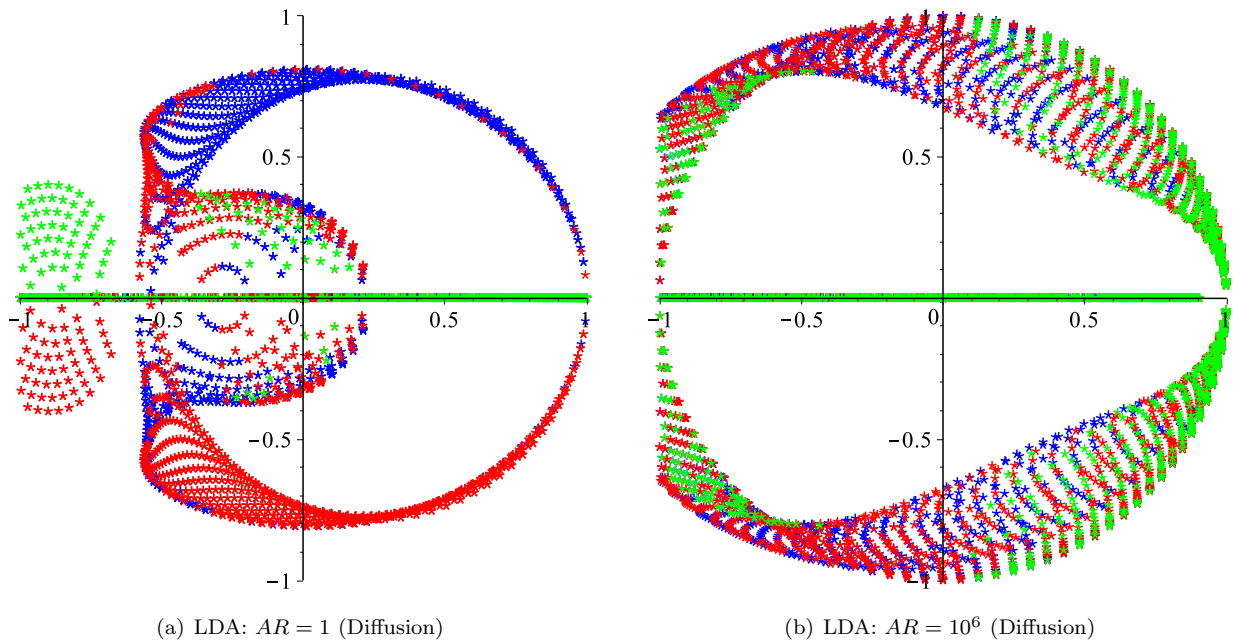
(b) LDA: $AR = 10^6$ (Diffusion)

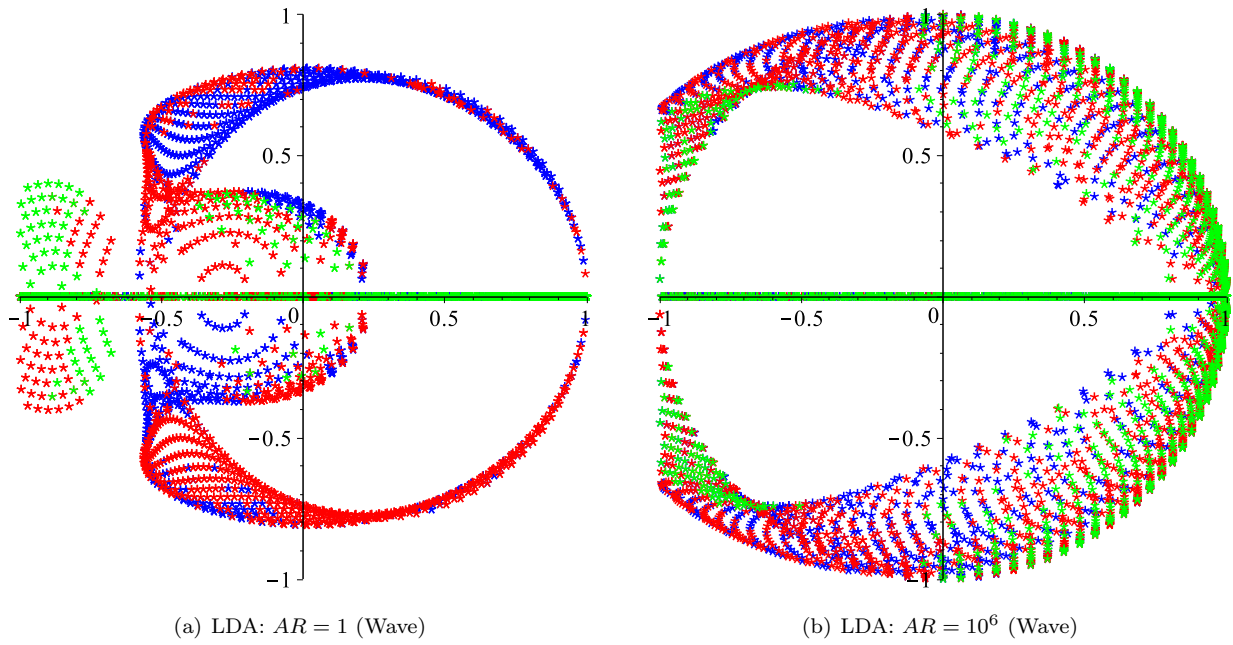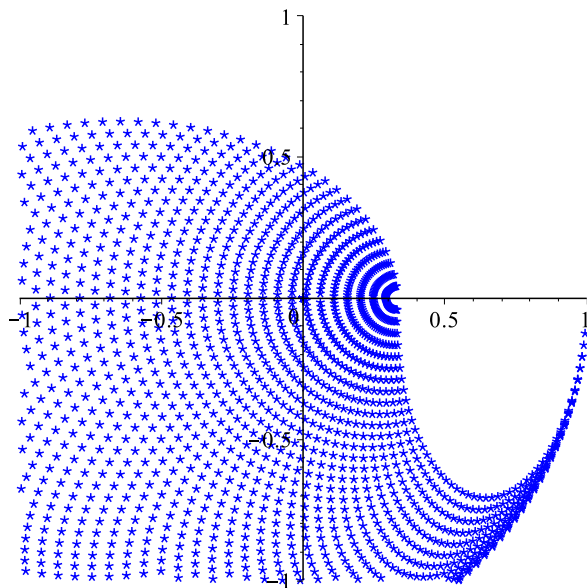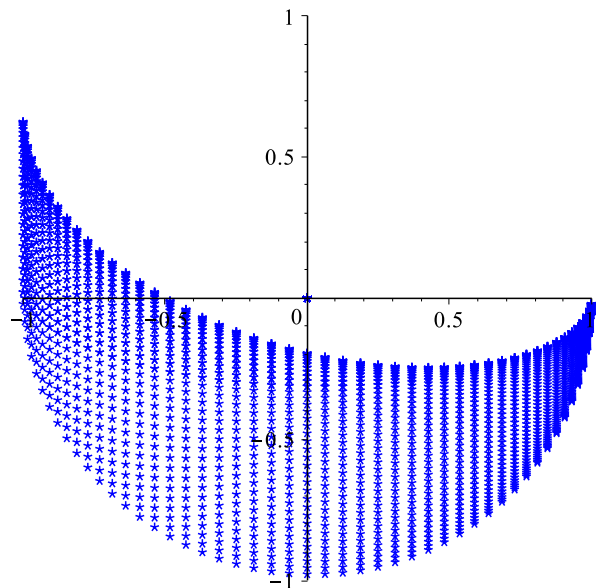**Figure 21. Fourier analysis of the LDA scheme for the hyperbolic diffusion system:** $\nu = 1$, $h_x = 0.1$.

American Institute of Aeronautics and Astronautics

(a) LDA: $AR = 1$ (Wave)

(b) LDA: $AR = 10^6$ (Wave)

Figure 22. Fourier analysis of the LDA scheme for the hyperbolic wave problem: $u_{tt} = c^2(u_{xx} + u_{yy})$, $c = 1$, $h_x = 0.1$.

(a) LDA: $AR = 1$ (Advection)
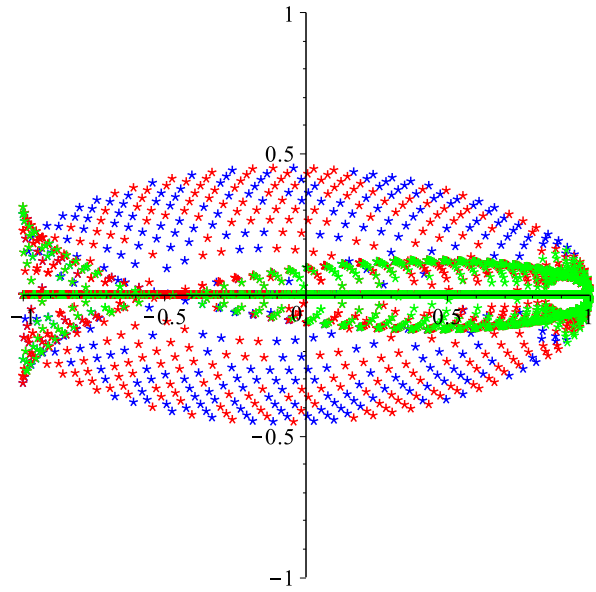

(b) LDA: $AR = 10^6$ (Advection)
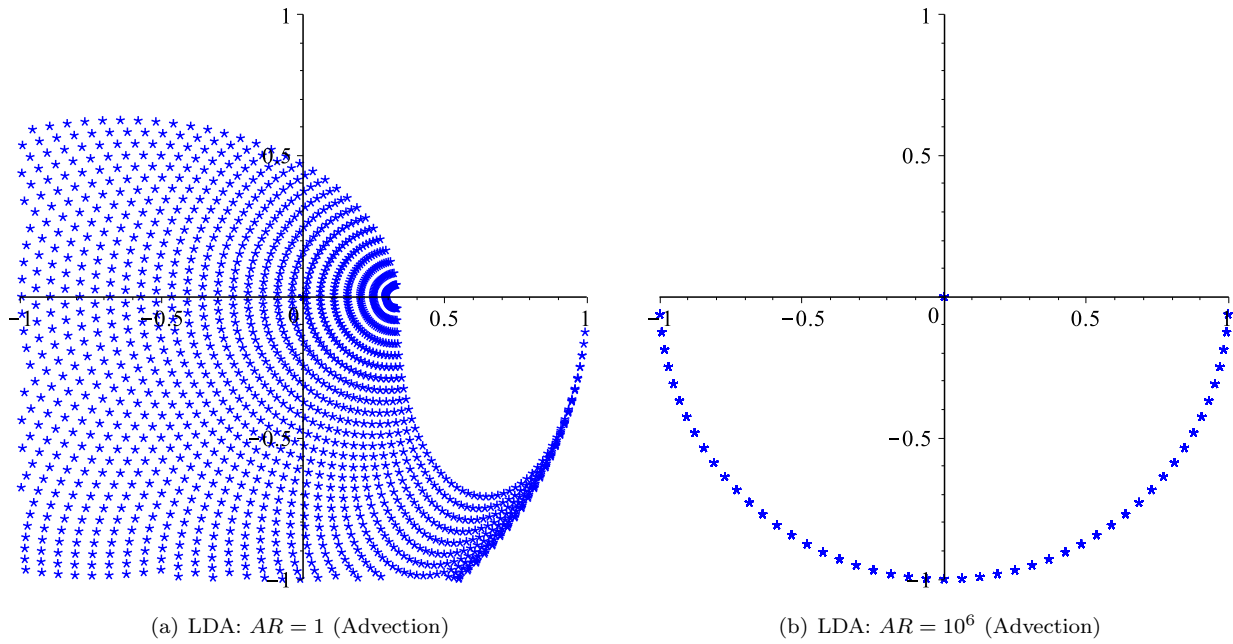
Figure 23. Fourier analysis of the LDA scheme for linear advection equation: $a = 2$, $b = 1$, $h_x = 0.1$.

American Institute of Aeronautics and Astronautics
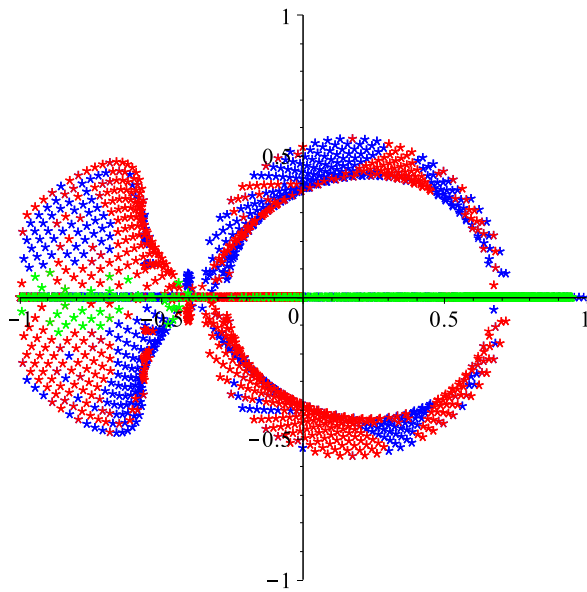
(a) SUPG: $AR = 1$ (Diffusion)
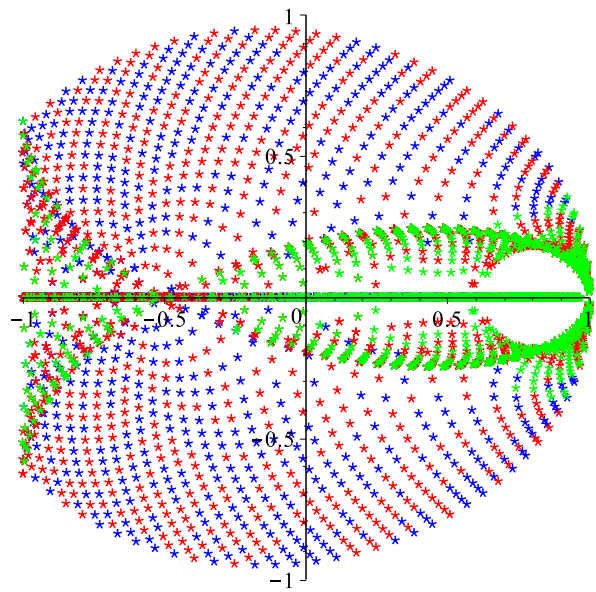
(b) SUPG: $AR = 10^6$ (Diffusion)

Figure 24. Fourier analysis of the SUPG scheme for the hyperbolic diffusion problem: $\nu = 1$, $h_x = 0.1$.

(a) SUPG: $AR = 1$ (Advection)

(b) SUPG: $AR = 10^6$ (Advection)

Figure 25. Fourier analysis of the SUPG scheme for linear advection equation: $a = 2$, $b = 1$, $h_x = 0.1$.
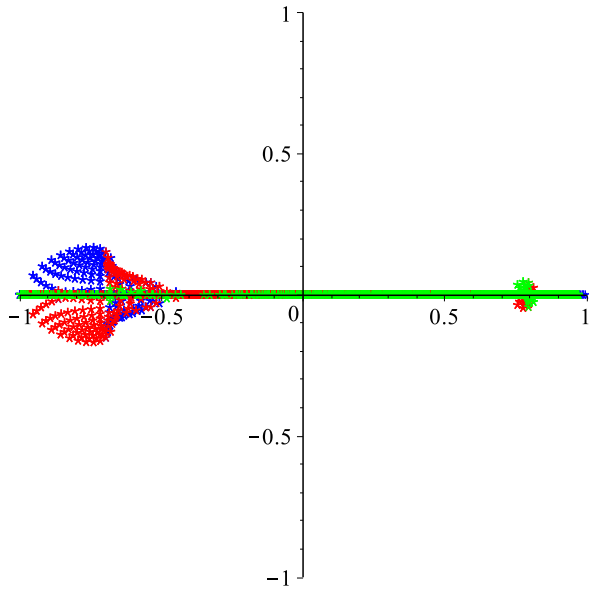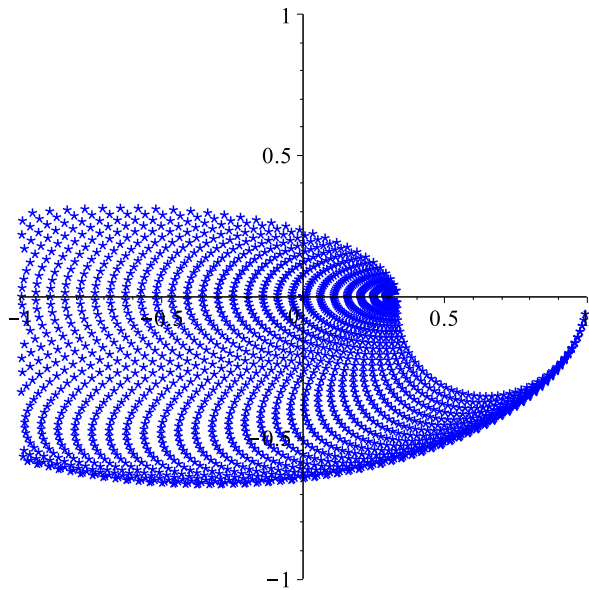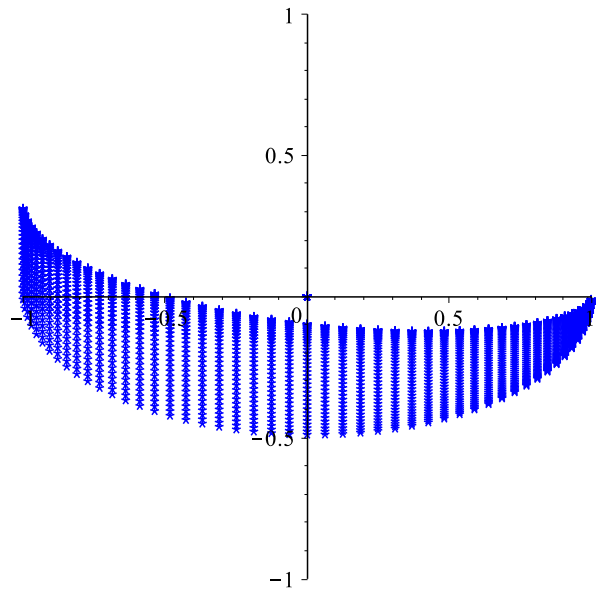
(a) LW: $AR = 1$ (Diffusion)

(b) LW: $AR = 10^6$ (Diffusion)

Figure 26. Fourier analysis of the Lax–Wendrof (LW) scheme for the hyperbolic diffusion problem: $\nu = 1$, $h_x = 0.1$.

(a) LW: $AR = 1$ (Advection)

(b) LW: $AR = 10^6$ (Advection)

Figure 27. Fourier analysis of the Lax–Wendrof (LW) scheme for linear advection equation: $a = 2$, $b = 1$, $h_x = 0.1$.
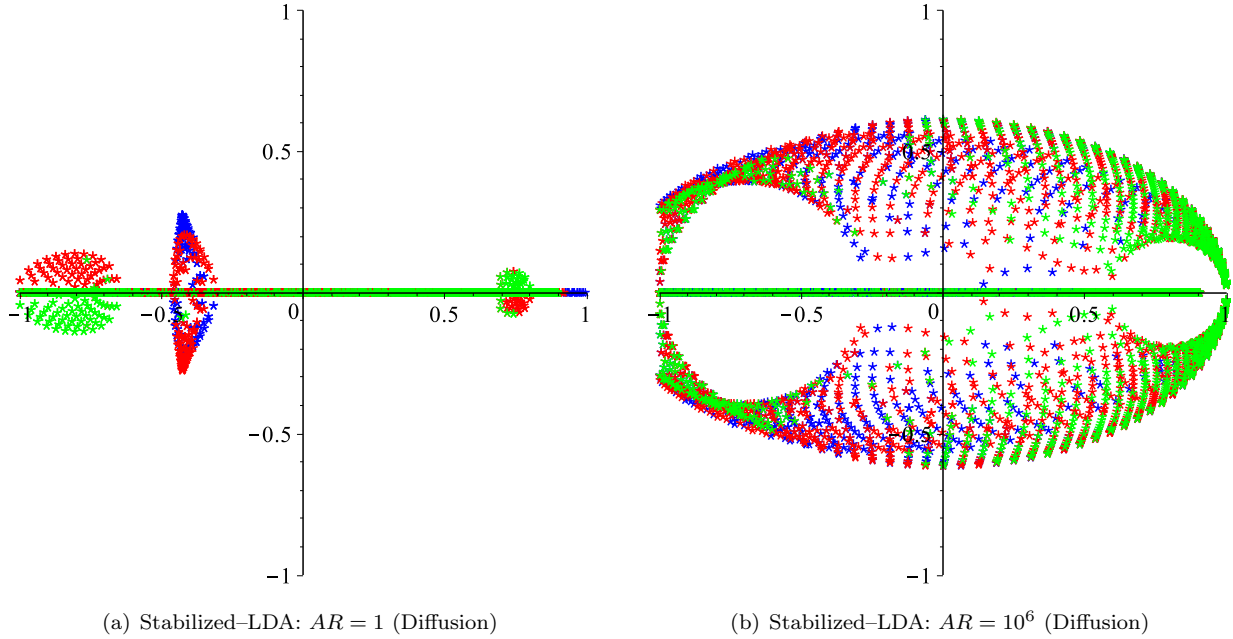
(a) Stabilized–LDA: $AR = 1$ (Diffusion)

(b) Stabilized–LDA: $AR = 10^6$ (Diffusion)

Figure 28. Fourier analysis of the stabilized–LDA scheme for the hyperbolic advection–diffusion problem: $\nu = 1$, $h_x = 0.1$.

We further examined the instability of these schemes with weighted Jacobi and under–relaxed GS relaxations. The matrix for the weighted Jacobi relaxation, $\mathbf{M}_{WJB}$ is:

$$\mathbf{M}_{WJB} = (1 - \omega)\mathbf{I} + \omega\mathbf{M}_{JB}, \tag{136}$$

where $\mathbf{I}$ is an identity matrix and $\mathbf{M}_{JB}$ is defined in Eq. 135. We found that generally these schemes are stable with weighted Jacobi or under–relaxed GS relaxations with $\omega = 0.7 - 0.8$. For instance, Fig. 29 presents the stability plots for the linear advection problem using SUPG formulation. The results indicate that the linear solver is stable on arbitrary regular grids with $\omega = 0.7$ as the under–relaxation value. The same technique applied to the hyperbolic diffusion problem as shown in Fig. 30. Similar results were obtained with other presented schemes and therefore not repeated here. Note that in the Fourier analysis, effects of the boundary points and grid irregularity could not be included. However, we found that the under-relaxation parameter of $\omega = 0.6 - 0.8$ is typically sufficient to make the linear relaxation stable on arbitrary anisotropic irregular grids for general advection–diffusion problems.

References

[1]P.A. Gnoffo. Multi–dimensional, inviscid flux reconstruction for simulation of hypersonic heating on tetrahedral grids. In *Proc. of 47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition*, AIAA Paper 2009–599, Orlando, Florida, 2009.

[2]K. Kitamura, E. Shima, Y. Nakamura, and P.L. Roe. Evaluation of euler fluxes for hypersonic heating computations. *AIAA J.*, 48(4):763–776, 2010.

[3]Z.J. Wang, K. Fidkowski, R. Abgral, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H.T. Huynh, N. Kroll, G. May, P.-O Persson, B. Van Leer, and M. Visbal. High–order CFD methods: Current status and perspective. *Int. J. Numer. Meth. Fluids*, 72:811–845, 2012.

[4]M.A. Park, V.N. Vatsa, M.R. Khorram, and D.P. Lockard. Aeroacoustic simulation of nose landing gear on adaptive unstructured grids with FUN3D. In *Proc. of 19th AIAA/CEAS Aeroacoustics Conference*, AIAA Paper 2013–2071, Berlin, Germany, 2011.

[5]H. Nishikawa. A first–order system approach for diffusion equation. I: Second order residual distribution schemes. *J. Comput. Phys.*, 227:315–352, 2007.

[6]H. Nishikawa. A first–order system approach for diffusion equation. II: Unification of advection and diffusion. *J. Comput. Phys.*, 229:3989–4016, 2010.

[7]H. Nishikawa. New–generation hyperbolic Navier-Stokes schemes: $O(1/h)$ speed–up and accurate viscous/heat fluxes. In *Proc. of 20th AIAA Computational Fluid Dynamics Conference*, AIAA Paper 2011–3043, Honolulu, Hawaii, 2011.

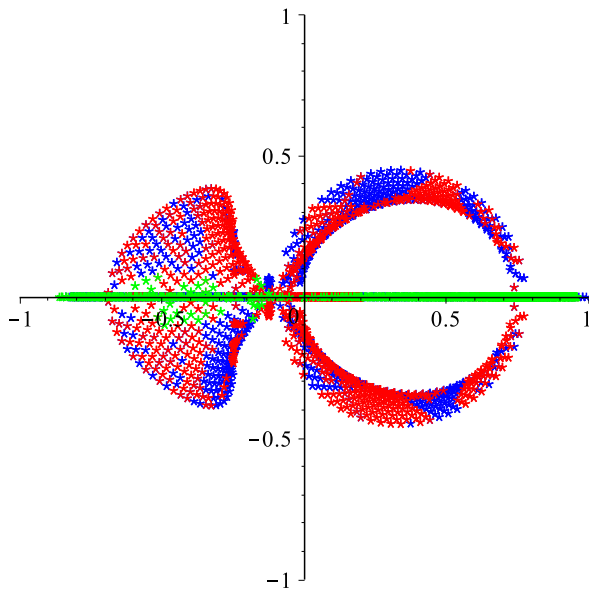American Institute of Aeronautics and Astronautics
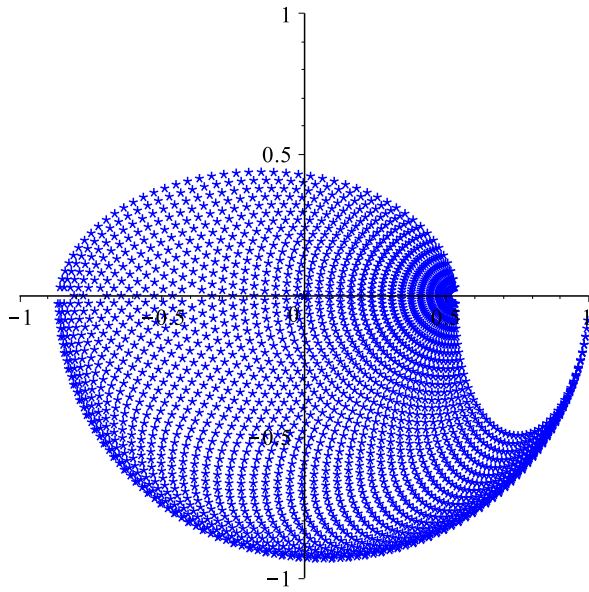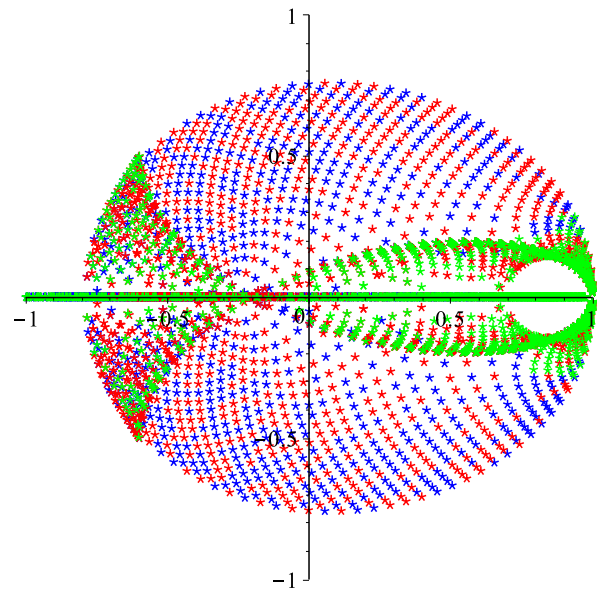
(a) SUPG: $AR = 1$, $\omega = 0.7$ (Advection)
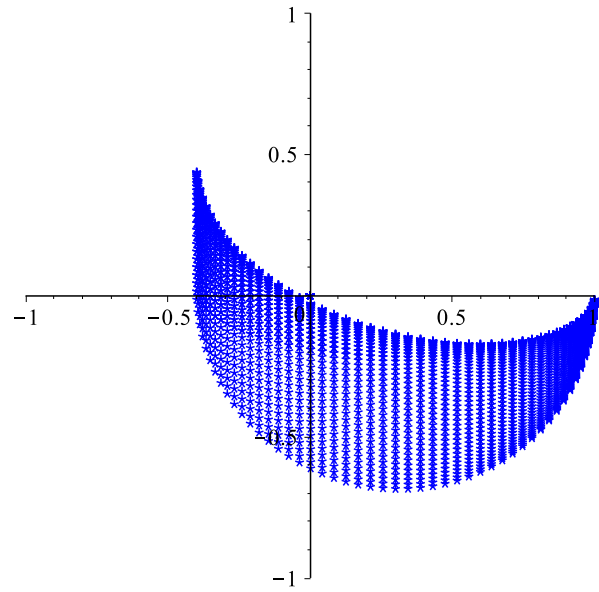
(b) SUPG: $AR = 10^6$, $\omega = 0.7$ (Advection)

Figure 29. Fourier analysis of the SUPG scheme for the linear advection problem ($a = 2$, $b = 1$, $h_x = 0.1$) using weighted Jacobi iteration ($\omega = 0.7$).

(a) SUPG: $AR = 1$, $\omega = 0.8$ (Diffusion)

(b) SUPG: $AR = 10^6$, $\omega = 0.8$ (Diffusion)

Figure 30. Fourier analysis of the SUPG scheme for the linear advection problem ($\nu = 1$, $h_x = 0.1$) with weighted Jacobi iteration ($\omega = 0.8$).

[8] H. Nishikawa. First–, second–, and third–order finite–volume schemes for diffusion. *J. Comput. Phys.*, 256:791–805, 2014.

[9] H. Nishikawa. First, second, and third order finite–volume schemes for advection–diffusion. *J. Comput. Phys.*, 273:287–309, 2014.

[10] H. Nishikawa. First, second, and third order finite–volume schemes for Navier–Stokes equations. In *Proc. of 7th AIAA Theoretical Fluid Mechanics Conference, AIAA Aviation and Aeronautics Forum and Exposition 2014*, AIAA Paper 2014–2091, Atlanta, GA, 2014.

[11] H. Nishikawa and P.L. Roe. Active flux for diffusion. In *Proc. of 7th AIAA Theoretical Fluid Mechanics Conference, AIAA Aviation and Aeronautics Forum and Exposition 2014*, AIAA Paper 2014-2092, Atlanta, GA, 2014.

[12] A. Mazaheri and H. Nishikawa. First–order hyperbolic system method for time–dependent advection–diffusion problems. Technical Report NASA/TM–2014–218175, March 2014.

[13] A. Mazaheri and H. Nishikawa. Very efficient high–order hyperbolic schemes for time–dependent advection–diffusion problems: Third–, fourth–, and sixth–order. *Computers and Fluids*, 102:131–147, 2014.

[14] H. Nishikawa, M. Rad, and P. Roe. A third–order fluctuation–splitting scheme that preserves potential flow. In *Proc. of 15th AIAA Computational Fluid Dynamics Conference*, AIAA Paper 01–2595, Anaheim, CA, 2001.

[15] E. van der Weide and H. Deconinck. Positive matrix distribution schemes for hyperbolic systems. In *Computational Fluid Dynamics*, pages 747–753. Wiley, New York, 1996.

[16] E.van der Weide, H. Deconinck, E. Issman, and G. Degrez. A parallel, implicit, multi–dimentional upwind residual distribution method for the Navier–Stokes equations on unstructured grids. *Computational Mechanics*, 23:199–208, 1999.

[17] H. 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.

[18] A. Brooks and T.J.R. Hughes. Streamline upwind petrov–galerkin formulation for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 32:199–259, 1982.

[19] A. De Palma, G. Pascazio, D.T. Rubino, and M. Napolitano. Residual–distribution schemes for advection and advection–diffusion problems on quadrilateral cells. *J. Comput. Phys.*, 218:159–199, 2006.

[20] Remi Abgrall, Dante De Santis, and Mario Ricchiuto. High-order preserving residual distribution schemes for advection-diffusion scalar problems on arbitrary grids. *SIAM J. Sci. Comput.*, 36(3):A955–A983, 2014.

[21] D. Caraeni and L. Fuchs. Compact third–order multidimensional upwind scheme for Navier–Stokes simulations. *Appl. Math.*, 15:373–401, 2002.

[22] D. Caraeni and L. Fuchs. A new compact high order multidimensional upwind discretization. In *Proc. of 4th World CSCC conference*, Vouliagmeni, Greece, July 10–15 2000.

[23] H. Nishikawa. Multigrid third–order least–squares solution of Cauchy–Riemann equations on unstructured triangular grids. *Int. J. Numer. Meth. Fluids*, 53:443–454, 2007.

[24] Katate Masatsuka. *I do like CFD, VOL.1, Second Edition.* Self published, 2013.

[25] R. Abgrall. Toward the ultimate conservative scheme: following the quest. *J. Comput. Phys.*, 167:277–315, 2001.

[26] R. Abgrall and P.L. Roe. High–order fluctuation schemes on triangular meshes. *J. Sci. Comput.*, 19:3–36, 2002.

[27] Li Wang, W. Kyle Anderson, J. Taylor Erwin, and Sagar Kapadia. Discontinuous galerkin and petrov galerkin methods for compressible viscous flows. *Computers and Fluids*, 100:1329, 2014.

[28] H. Nishikawa and P.L. Roe. On high–order fluctuation–splitting schemes for Navier–Stokes equations. In C. Groth and D.W. Zingg, editors, *Computational Fluid Dynamics 2004*, pages 799–804. Springer–Verlag, 2004.

[29] H. Nishikawa. Accuracy-preserving boundary flux quadrature for finite-volume discretization on unstructured grids. *J. Comput. Phys.*, 281:518–555, 2015.

[30] R. Abgrall. Essentially non–oscillatory residual distribution schemes for hyperbolic problems. *J. Comput. Phys.*, 214:773–808, 2006.