Preprint accepted for publication in Journal of Computational Physics, 2014

http://dx.doi.org/10.1016/j.jcp.2014.05.021

First, Second, and Third Order Finite-Volume Schemes for Advection-Diffusion

Hiroaki Nishikawa *
National Institute of Aerospace, Hampton, VA 23666

June 7, 2014

Abstract

In this paper, we present first, second, and third order implicit finite-volume solvers for advection-diffusion problems based on the first-order hyperbolic system method. In particular, we demonstrate that the construction of an uniformly accurate third-order advection-diffusion scheme is made trivial by the hyperbolic method while a naive construction of adding a third-order diffusion scheme to a third-order advection scheme can fail to yield third-order accuracy. We demonstrate also that the gradients are computed simultaneously to the same order of accuracy as that of the solution variable on irregular triangular grids: first, second and third order accurate gradients by the first, second, and third order schemes, respectively. Furthermore, the first and second order schemes are shown to achieve one order higher accuracy for the solution variable in the advection limit. It is also shown that these schemes are capable of producing highly accurate and smooth solution gradients along the boundary in a highly-skewed anisotropic irregular triangular grid while conventional schemes suffer from oscillations on such a grid. Numerical results show that these schemes are capable of delivering high accuracy over conventional schemes at a significantly reduced cost.

1 Introduction

This paper is a sequel to the previous paper [1], where we presented first, second, and third order diffusion schemes constructed based on the the first-order hyperbolic system method [2]. In this method, diffusion schemes are constructed from advection schemes via an equivalent hyperbolic system. We demonstrated that first, second, and third order diffusion schemes constructed by the upwind flux yield first, second, and third order accurate solution and gradients, respectively, on irregular triangular grids, with significant acceleration in convergence over a traditional scheme. In particular, the third-order diffusion scheme was shown to be incomparably more accurate and efficient, providing third-order accuracy in the solution as well as in the gradients at a significantly reduced cost over conventional schemes.

Towards the development of a robust, accurate, and efficient three-dimensional viscous solver capable of producing high accurate derivatives (e.g, viscous stresses, heat fluxes, and vorticity) on unstructured grids, we now consider the advection-diffusion equation. The main focus of the paper is on the uniform third-order accuracy from the advection limit to the diffusion limit. The third-order scheme considered in the current study belongs to the class of numerical schemes based on vanishing residuals [3, 4, 5]. The second-order error term contains the residual that vanishes in the steady state; the leading error is then upgraded to third order. This type of scheme is known to deliver high-order accuracy on a relatively compact stencil (e.g., third-order accuracy on a second-order stencil), and thus very attractive for practical computations where high accuracy is demanded at a minimal additional cost. Successful applications of these schemes to the advection-diffusion equation require a careful construction to ensure the property of vanishing residuals. If not designed properly, the scheme loses the design accuracy at least by one order as shown in Ref.[4] for a similar high-order scheme, and in Ref.[6] for conservation laws with source terms. To achieve third-order accuracy for the advection-diffusion

*National Institute of Aerospace, 100 Exploration Way, Hampton, VA 23666-6147, USA. Tel.:+1 757 864 7244. *E-mail address:* hiro@nianet.org

equation, the advective and diffusive terms must be discretized not only to third order but also in a compatible manner as we discuss later in details. Uniform accuracy is very important for advection-diffusion problems for a practical reason. Since such accuracy deterioration occurs typically where the advective and diffusive terms are equally important, it can happen anywhere the local Reynolds number falls within such a range in practical viscous flows, e.g., a boundary layer. Consequently, improperly-designed high-order schemes will only increase the computational cost without delivering high accuracy; locally generated lower error may then propagate through the downstream and affect other regions where high-accuracy is maintained otherwise.

A radical approach to ensure the uniform accuracy is to integrate the advective and diffusive terms into a single hyperbolic system as proposed in Ref.[7]. In this approach, the accuracy degradation cannot occur because there is only a single hyperbolic system. However, the extension to the compressible Navier-Stokes equations is not straightforward. A full eigen-structure has not been found yet for the hyperbolized compressible Navier-Stokes equations, and therefore a full integration of the inviscid and the viscous terms remains a challenge. To overcome the difficulty, a simplified approach was proposed in Ref.[8], where the inviscid and viscous terms are analyzed and discretized independently. As demonstrated in Ref.[8], the hyperbolic viscous system alone can be fully analyzed and the eigen-structure can be derived to construct the upwind viscous flux. A full Navier-Stokes discretization is then constructed by adding the upwind viscous flux to the inviscid flux. For the type of scheme considered in the current study, such an approach often fails for conventional methods, but it works trivially for hyperbolized systems because all terms are discretized by the same scheme. In this paper, we discuss the simplified approach for the model equation, and thereby lay a foundation for the construction of first, second, and third order hyperbolic Navier-Stokes schemes.

The paper begins with the hyperbolic formulation for the advection-diffusion system, describes implicit first, second, and third order node-centered edge-based finite-volume schemes, including the discussion on the compatibility problem for uniform third-order accuracy, then presents numerical results and discussion, and finally concludes with remarks.

2 Hyperbolic Advection-Diffusion System

Consider the advection-diffusion equation in two dimensions:

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

where u is the solution variable, (a, b) is a constant advection vector, and ν is a constant diffusion coefficient. Construction of numerical schemes for the advective term is relatively straightforward, but the same is not necessarily true for the diffusive term of second derivatives, especially for high-order methods and unstructured grids. A radical approach to diffusion is to convert the diffusive term into a hyperbolic system as proposed in Ref.[2], which is extended to the advection-diffusion in the following form:

$$\begin{aligned} \partial_t u + a\,\partial_x u + b\,\partial_y u &= \nu\,(\partial_x p + \partial_y q), \\ \partial_t p &= (\partial_x u - p)/T_r, \\ \partial_t q &= (\partial_y u - q)/T_r, \end{aligned} \tag{2}$$

where p and q are called the gradient variables which relax to the solution derivatives, $\partial_x u$ and $\partial_y u$, respectively, in the steady state, and T_r is a *free parameter* called the relaxation time. The system is equivalent to the advection-diffusion equation (1) in the steady state for any nonzero T_r. Therefore, the steady solution to the advection-diffusion equation (1) can be computed by solving the first-order system. This idea was first introduced in Ref.[2], and subsequently extended to the advection-diffusion equation [7] and the Navier-Stokes equations [8]. Discretization is made simple because there are no second derivatives and the system is *hyperbolic*[7] for which a variety of well established techniques are available. The hyperbolic system here designed specifically for steady computations is simply called the hyperbolic advection-diffusion system. This formulation clearly shows that the method is different from other relaxation models[9, 10, 11, 12] . Our method introduces relaxation only in the diffusive fluxes, and our target equation is exactly the advection-diffusion equation (1), not an asymptotic approximation. In other words, our hyperbolic model reduces to the advection-diffusion equation exactly in the steady state for any nonzero T_r, whereas other models are equivalent to the advection-diffusion equation only in the limit $T_r \to 0$, resulting in a stiff relaxation system. In our model, the stiffness is not an issue because T_r is a free parameter and does not have to be small. The relaxation time can be determined not by physical consideration but by numerical consideration, i.e., fast steady convergence. Also, our target

applications are second- or higher-order partial differential equations, such as the Navier-Stokes equations, not specific to rarefied gasdynamics or radiation hydrodynamics. A similar hyperbolic system is utilized in a recent work [13] [1], where high-order explicit time-stepping schemes are constructed based on the first-order hyperbolic formulation of diffusion. However, again, the relaxation time is restricted as it has to be proportional to the mesh spacing to the power of the order of accuracy, e.g., $T_r = O(h^2)$ for second-order schemes, so that the hyperbolic system accurately represents the original diffusion term at any instant of time. Consequently, the $O(1/h)$-type fast convergence, which is one of the distinguished features of our method, cannot be achieved in their method. Also, it is not known whether their method can produce the gradients to the same order of accuracy as that of the solution on irregular grids. We emphasize again that the key idea of the hyperbolic method considered in this paper is to construct a hyperbolic system that recovers the original equation in the steady state. The time evolution of the solution will not be accurate for a large value of T_r designed for fast steady convergence, and therefore explicit time-stepping schemes cannot be constructed for unsteady problems. Time-accurate computations are possible by implicit time-stepping (or space-time) schemes, where a fast steady solver constructed in this work can be used to solve a system of globally coupled residual equations arising from the implicit time integration scheme. Note that this is not a severe limitation of the hyperbolic method. Many practical simulation codes employ implicit time-stepping schemes for efficiency and robustness purposes especially for high-Reynolds-number applications, and demand powerful steady solvers. The steady solver constructed based on the hyperbolic method has a potential for significantly improving the efficiency of the implicit time-integration scheme as well as the accuracy in the gradients. In fact, the hyperbolic method has already been extended to unsteady problems, first in Ref.[14] for a second-order residual-distribution method, and then in Ref[15] for a very high-order residual-distribution method. In both, the second- and higher-order backward Euler time-integration schemes are used, and a steady solver have been shown to be capable of solving the unsteady residual equations very efficiently as well as producing highly accurate solution gradients at every physical time step. The methodology is equally applicable to the finite-volume schemes. Further discussion on time accurate schemes is beyond the scope of the present paper.

We write the system in the vector form,

$$\partial_t \mathbf{U} + \partial_x \mathbf{F} + \partial_y \mathbf{G} = \mathbf{S}, \tag{3}$$

where

$$\mathbf{U} = \begin{bmatrix} u \\ p \\ q \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} au - \nu p \\ -u/T_r \\ 0 \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} bu - \nu q \\ 0 \\ -u/T_r \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} 0 \\ -p/T_r \\ -q/T_r \end{bmatrix}. \tag{4}$$

In Ref.[7], the system is taken as a single hyperbolic system. In this paper, we consider the advective term and the diffusive term separately.

$$\mathbf{F} = \mathbf{F}^a + \mathbf{F}^d = \begin{bmatrix} au \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -\nu p \\ -u/T_r \\ 0 \end{bmatrix}, \quad \mathbf{G} = \mathbf{G}^a + \mathbf{G}^d = \begin{bmatrix} bu \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -\nu q \\ 0 \\ -u/T_r \end{bmatrix}. \tag{5}$$

The flux Jacobian projected along an arbitrary vector, $\mathbf{n} = (n_x, n_y)$ is given by

$$\mathbf{A}_n = \frac{\partial \mathbf{F}}{\partial \mathbf{U}} n_x + \frac{\partial \mathbf{G}}{\partial \mathbf{U}} n_y = \mathbf{A}_n^a + \mathbf{A}_n^d, \tag{6}$$

where $\mathbf{A}_n^a$ and $\mathbf{A}_n^d$ are the advective and diffusive Jacobians, respectively,

$$\mathbf{A}_n^a = \frac{\partial \mathbf{F}^a}{\partial \mathbf{U}} n_x + \frac{\partial \mathbf{G}^a}{\partial \mathbf{U}} n_y = \begin{bmatrix} a_n & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{A}_n^d = \frac{\partial \mathbf{F}^d}{\partial \mathbf{U}} n_x + \frac{\partial \mathbf{G}^d}{\partial \mathbf{U}} n_y = \begin{bmatrix} 0 & -\nu n_x & -\nu n_y \\ -n_x/T_r & 0 & 0 \\ -n_y/T_r & 0 & 0 \end{bmatrix}, \tag{7}$$

[1] In Ref.[13], it is stated that the first-order hyperbolic system method was first introduced in the paper [4], but it is not true. The method was first introduced in Ref. [2].

and $a_n = an_x + bn_y$. The advective Jacobian has the eigenvalue, a_n, and the diffusive Jacobian has the following eigenvalues:

$$\lambda = \pm\sqrt{\frac{\nu}{T_r}}, \quad 0. \tag{8}$$

The zero eigenvalue corresponds to the inconsistency damping mode[2], acting on the components of p and q such that $q_x - p_y \neq 0$. It can be made nonzero by the fully hyperbolic formulation [1], which we shall employ in this study for constructing a third-order scheme.

The relaxation time is defined as $T_r = L_r/(|a_n| + \nu/L_r)$ in Ref.[7] based on a unified treatment of the advective and diffusive terms. However, in this paper, we treat the diffusive part independently and thus define the relaxation time as suggested for pure diffusion in the previous paper:

$$T_r = \frac{L_r^2}{\nu}, \quad Lr = \frac{1}{2\pi}. \tag{9}$$

With the purely diffusive relaxation time, the system reduces to the scalar advection equation in the advection limit, $\nu \to 0$:

$$\begin{aligned} \partial_t u + a\,\partial_x u + b\,\partial_y u &= 0, \\ \partial_t p &= 0, \\ \partial_t q &= 0, \end{aligned} \tag{10}$$

whereas for $T_r = L_r/(|a_n| + \nu/L_r)$, it remains finite ($T_r \to L_r/|a_n|$) and thus generates a coupled system:

$$\begin{aligned} \partial_t u + a\,\partial_x u + b\,\partial_y u &= 0, \\ \partial_t p &= (\partial_x u - p)/T_r, \\ \partial_t q &= (\partial_y u - q)/T_r. \end{aligned} \tag{11}$$

Anticipating the extension to the Navier-Stokes equations, we find the decoupling as in Equation (10) more suitable for modeling the hyperbolic Navier-Stokes system[8], where the viscous stresses and heat fluxes are taken as the gradient variables, which are not physically coupled with the inviscid terms. In the inviscid limit, the hyperbolic Navier-Stokes system that decouples reduces exactly to the Euler equations. Also, the separate treatment of the inviscid and viscous terms dramatically simplifies the construction of numerical schemes because it only requires the eigen-structure of each term, which can be fully analyzed independently. In other words, the inviscid scheme can be chosen independently from the choice of the hyperbolic viscous scheme. This simplified approach was first considered for the hyperbolic Navier-Stokes system in Ref.[8], but not studied for the model equation before. This paper only considers the linear model equation, but the method can be extended to nonlinear equations by the preconditioned conservative formulation proposed in Ref.[8].

3 Node-Centered Edge-Based Finite-Volume Scheme

3.1 Discretization

The node-centered edge-based finite-volume scheme for Equation (3) is given by

$$V_j \frac{d\mathbf{U}_j}{dt} = -\sum_{k\in\{k_j\}} \Phi_{jk} A_{jk} + \mathbf{S}_j V_j, \tag{12}$$

where V_j is the measure of the dual control volume around node j in the set $\{J\}$ of nodes, $\{k_j\}$ is a set of neighbors of j, Φ_{jk} is a numerical flux, and A_{jk} is the magnitude of the directed area vector, i.e., $A_{jk} = |\mathbf{n}_{jk}| = |\mathbf{n}^{\ell}_{jk} + \mathbf{n}^{r}_{jk}|$ (see Figure 1). This formulation is valid for triangular, quadrilateral, or mixed grids, and all schemes developed below can be directly applied to any grid except the third-order scheme, which is third-order accurate only on triangular grids. For the third-order scheme, the point-source integration in Equation (12) cannot be employed; it needs to be discretized carefully to preserve the accuracy as will be discussed later. Note also that an appropriate boundary flux must be supplied at the boundary node. For first-order schemes, a point evaluation

is sufficiently accurate, but for second-order schemes, a different quadrature is required for the linear exactness in the flux integration. See Appendix E in Ref.[16] for a comprehensive list of linearity preserving boundary quadrature formulas in two and three dimensions. For the third-order scheme, yet another formula is required for preserving quadratic fluxes. In this paper, however, we focus on test problems where the point evaluation is sufficient for all schemes: uniformly spaced flat boundary grids. Details on the accuracy-preserving boundary flux quadrature formulas will be given in a separate paper.

3.2 Numerical Flux

The numerical flux is computed by the upwind flux:

$$\Phi_{jk} = \frac{1}{2}(\mathbf{H}_L + \mathbf{H}_R) \cdot \hat{\mathbf{n}}_{jk} - \frac{1}{2}|\mathbf{A}_n|(\mathbf{U}_R - \mathbf{U}_L), \tag{13}$$

where $\mathbf{H}_L = [\mathbf{F}_L, \mathbf{G}_L]$, $\mathbf{H}_R = [\mathbf{F}_R, \mathbf{G}_R]$, and $\hat{\mathbf{n}}_{jk} = (n_x, n_y)$ is the unit directed area vector. The left and right fluxes and solutions are defined at the edge midpoint and evaluated by the nodal values for first-order accuracy and by the linear extrapolation from the nodes for second/third-order accuracy as discussed later. $\mathbf{A}_n$ is the flux Jacobian projected along $\hat{\mathbf{n}}_{jk}$. The construction of the absolute Jacobian, $|\mathbf{A}_n|$, requires the full eigen-structure of the target system. The eigen-structure of the hyperbolic advection-diffusion system (3) is simple enough to enable the construction of the upwind flux[7]. In this paper, however, we consider a simplified construction, which was proposed for the hyperbolic Navier-Stokes scheme in Ref.[8]. In this approach, the numerical flux is defined by the sum of the upwind advection flux Φ^a_{jk} and the upwind hyperbolic-diffusion flux Φ^d_{jk}:

$$\Phi_{jk} = \Phi^a_{jk} + \Phi^d_{jk}, \tag{14}$$

where

$$\Phi^a_{jk} = \frac{1}{2}(\mathbf{H}^a_L + \mathbf{H}^a_R) \cdot \hat{\mathbf{n}}_{jk} - \frac{1}{2}|\mathbf{A}^a_n|(\mathbf{U}_R - \mathbf{U}_L), \tag{15}$$

$$\Phi^d_{jk} = \frac{1}{2}(\mathbf{H}^d_L + \mathbf{H}^d_R) \cdot \hat{\mathbf{n}}_{jk} - \frac{1}{2}|\mathbf{A}^d_n|(\mathbf{U}_R - \mathbf{U}_L). \tag{16}$$

The left and right fluxes and the Jacobians are defined separately for the advection term and the hyperbolic diffusion terms:

$$\mathbf{H}^a_L = [\mathbf{F}^a_L, \mathbf{G}^a_L], \quad \mathbf{H}^a_R = [\mathbf{F}^a_R, \mathbf{G}^a_R], \quad |\mathbf{A}^a_n| = \begin{bmatrix} |a_n| & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{17}$$

$$\mathbf{H}^d_L = \left[\mathbf{F}^d_L, \mathbf{G}^d_L\right], \quad \mathbf{H}^d_R = \left[\mathbf{F}^d_R, \mathbf{G}^d_R\right], \quad |\mathbf{A}^d_n| = \frac{\nu}{L_r} \begin{bmatrix} 1 & 0 & 0 \\ 0 & n_x^2 & n_x n_y \\ 0 & n_x n_y & n_y^2 \end{bmatrix}. \tag{18}$$

Note that the resulting flux can be written as

$$\Phi_{jk} = \frac{1}{2}(\mathbf{H}_L + \mathbf{H}_R) \cdot \hat{\mathbf{n}}_{jk} - \frac{1}{2}\left(|\mathbf{A}^a_n| + |\mathbf{A}^d_n|\right)(\mathbf{U}_R - \mathbf{U}_L), \tag{19}$$

In this way, each absolute Jacobian can be constructed independently for the advective term and the diffusive term. It has been successfully demonstrated for the hyperbolic Navier-Stokes system[8]: the inviscid term and the viscous terms are both hyperbolic and their eigen-structures can be fully analyzed independently.

It should be noted that we do not assume that $|\mathbf{A}_n| = |\mathbf{A}^a_n| + |\mathbf{A}^d_n|$, which is not true. We have simply added a diffusive flux to an advective flux. This is, therefore, an approximation. The impact of the approximation can be studied by comparing it with the numerical flux having the true Jacobian $|\mathbf{A}_n|$, but such a comparison is beyond the scope of the present paper. The objective of this paper is to study the approximate approach, which can be extended to more complex systems. Note that this approximate approach, e.g., simply add a viscous flux to an inviscid flux to construct a Navier-Stokes flux, is widely adapted (although it may not be recognized as approximate) in many discretization methods, including the finite-volume methods and the discontinuous Galerkin methods. Here, in the same way, we have added the upwind hyperbolic diffusion flux, instead of a conventional diffusive flux, to the upwind advective flux.

3.3 Implicit Solver

A steady-state solution can be obtained by marching in time towards the steady state as demonstrated in the previous paper. In this paper, we drop the time derivative term,

$$0 = -\sum_{k\in\{k_j\}} \Phi_{jk} A_{jk} + \mathbf{S}_j V_j, \tag{20}$$

and construct an implicit solver for the global system of steady residual equations. The advantage of $O(1/h)$ acceleration in the steady convergence over traditional methods, which has been observed for explicit time-marching schemes [2, 7, 8, 1], now comes in the iterative convergence of the linear system arising from the implicit formulation as will be demonstrated later.

Consider the global system of residual equations, which consists of rows of the nodal residual (20):

$$0 = \mathbf{Res}(\mathbf{U}_h), \tag{21}$$

where $\mathbf{U}_h$ denotes the global solution vector for which the system is to be solved. We consider the iterative method in the form:

$$\mathbf{U}_h^{n+1} = \mathbf{U}_h^n + \Delta\mathbf{U}_h, \tag{22}$$

where the correction $\Delta\mathbf{U}_h$ is the solution to the following linear system:

$$\frac{\partial \mathbf{Res}}{\partial \mathbf{U}_h}\Delta\mathbf{U}_h = -\mathbf{Res}(\mathbf{U}_h^n). \tag{23}$$

The Jacobian matrix is constructed analytically by differentiating the residual of the first-order scheme and hand-coded. Therefore, the method is Newton's method for the first-order scheme, and a defect correction method for the second and third order schemes, provided the linear system is fully solved. In practice, we do not fully solve but relax the linear system. In this work, we employ, for the sake of simplicity, the sequential block (point-implicit) Gauss-Seidel (GS) relaxation to relax the linear system to a specified tolerance. It is possible to add a pseudo-time term to the left hand side, but it is not used in this work. It is emphasized that the condition number of the Jacobian is $O(1/h)$ for the first-order hyperbolic formulation, not $O(1/h^2)$, which is typical for diffusion schemes, even in the diffusion limit, implying $O(1/h)$ acceleration in iterative convergence over traditional schemes for diffusion dominated problems.

3.4 First-Order Scheme

We construct a first-order scheme by evaluating the left and right states with the nodal solutions:

$$\mathbf{U}_L = \mathbf{U}_j, \qquad \mathbf{U}_R = \mathbf{U}_k, \tag{24}$$

and the numerical flux by the upwind flux (19). The resulting scheme corresponds to Scheme I in Ref.[1]. In this paper, we employ a modified version of Scheme I, called Scheme II, where the gradient variables are used to enable the linear extrapolation of the solution:

$$u_L = u_j + \frac{1}{2}(p_j, q_j)\cdot\Delta\mathbf{l}_{jk}, \qquad u_R = u_k - \frac{1}{2}(p_k, q_k)\cdot\Delta\mathbf{l}_{jk}, \tag{25}$$

where $\Delta\mathbf{l}_{jk} = (\Delta y_{jk}, \Delta y_{jk}) = (x_k - x_j, y_k - y_j)$. The source terms are evaluated by the point integration as indicated in Equation (12). The scheme is compact and first-order accurate for all variables in the diffusion limit as shown in the previous paper[1]. The Jacobian is constructed exactly based on this scheme. In this paper, we demonstrate that the scheme can achieve second-order accuracy in the advection limit. For vanishingly small ν, the hyperbolic diffusion term has no effect on the advection-diffusion equation, but it is still capable of producing accurate gradients, resulting in a second-order accurate scheme for the advective term.

3.5 Second-Order Scheme

For second-order accuracy, we compute the nodal gradients by a linear least-squares (LSQ) method, and evaluate the left and right states by the linear extrapolation from the nodes. Again, we employ Scheme II and

avoid the gradient computation for the solution by using (p, q) as in Equation (25). The gradient computation is required only for p and q to perform the linear extrapolation:

$$p_L = p_j + \frac{1}{2}\nabla p_j \cdot \Delta \mathbf{l}_{jk}, \qquad p_R = p_k - \frac{1}{2}\nabla p_k \cdot \Delta \mathbf{l}_{jk}, \tag{26}$$

$$q_L = q_j + \frac{1}{2}\nabla q_j \cdot \Delta \mathbf{l}_{jk}, \qquad q_R = q_k - \frac{1}{2}\nabla q_k \cdot \Delta \mathbf{l}_{jk}, \tag{27}$$

where ∇p_j is the gradient of p computed by the LSQ method at j, and similarly for ∇q_j . The numerical flux is computed by the upwind flux (19). As in the first-order scheme, the source terms are evaluated by the point integration also for the second-order scheme. This scheme is known to be second-order accurate for both the solution and the gradients in the diffusion limit [1]. In the advection limit, the scheme becomes a third-order advection scheme as shown later.

3.6 Third-Order Scheme

As in the previous work, we consider the third-order edge-based finite-volume scheme discovered by Katz and Sankaran[17]. It is a very special node-centered finite-volume scheme for hyperbolic conservation laws: the second-order node-centered edge-based finite-volume scheme achieves third-order accuracy on triangular grids if the nodal gradients are exact for quadratic functions and the flux is linearly extrapolated to the edge-midpoint in the case of nonlinear fluxes. For linear problems considered in this paper, the flux extrapolation is not necessary. Third-order accuracy is achieved by simply replacing the linear LSQ gradients by quadratic ones in the second-order scheme. The third-order accuracy has been demonstrated for regular as well as irregular triangular grids in Refs.[17, 18, 19]. It is a very economical third-order scheme: third-order accuracy obtained nearly at the cost of the second-order edge-based finite-volume scheme. Nevertheless, its extensions to other types of equations including source terms are not straightforward. As the scheme relies on the second-order error term that vanishes in the steady state, every term in a target equation must be discretized with a compatible second-order error term. For source terms, the use of the point integration is not compatible, and thus third-order accuracy cannot be achieved. An extended Galerkin discretization formula proposed in Ref.[20] is compatible, and also a systematic method to ensure the compatible discretizaton has also been devised in Ref.[6]. See Ref.[1] for more details. For the diffusive term in the original form (1), it is also not straightforward to develop a compatible discretization. Here, we focus on the diffusion term.

In Section 3.6.1, we first illustrate the compatibility problem for the advection-diffusion equation. And then, in Section 3.6.2, we present the hyperbolic construction as a radically simple way to avoid the problem.

3.6.1 Compatibility Problem of Uniform Third-Order Accuracy for Advection-Diffusion

Consider the advection-diffusion equation,

$$\partial_x f + \partial_y g = \nu\,(\partial_{xx} u + \partial_{yy} u), \tag{28}$$

where $(f, g) = (au, bu)$. In the absence of the diffusive term (e.g., $\nu = 0$), the third-order scheme has the following local truncation error at node j on a regular triangular grid[6]:

$$\mathcal{T}_j^{\mathrm{adv}} = C_1\,\partial_{xx}(\partial_x f + \partial_y g) + C_2\,\partial_{xy}(\partial_x f + \partial_y g) + C_3\,\partial_{yy}(\partial_x f + \partial_y g) + O(h^3), \tag{29}$$

where the derivatives are defined at j, h is a typical mesh spacing, and the coefficients, C_1, C_2, and C_3, are geometrical constants of $O(h^2)$. Without loss of generality, we focus on a regular triangular grid composed of isosceles right triangles of spacing h where the truncation error is specifically given by

$$\mathcal{T}_j^{\mathrm{adv}} = \frac{h^2}{12}\left[\partial_{xx}(\partial_x f + \partial_y g) + \partial_{xy}(\partial_x f + \partial_y g) + \partial_{yy}(\partial_x f + \partial_y g)\right] + O(h^3). \tag{30}$$

The second-order error term will vanish because $\partial_x f + \partial_y g = 0$ for the exact solution or equivalently because $\partial_x f + \partial_y g = 0$ in the steady state, and thus the truncation error is upgraded to third-order. Consequently, the discretization error is expected to be third-order. In order to achieve third order accuracy for the advection-diffusion equation, the diffusion scheme must have a second-order error term in the form:

$$\mathcal{T}_j^{\mathrm{diff}} = -\frac{h^2}{12}\left[\partial_{xx}(\nu\,(\partial_{xx} u + \partial_{yy} u)) + \partial_{xy}(\nu\,(\partial_{xx} u + \partial_{yy} u)) + \partial_{yy}(\nu\,(\partial_{xx} u + \partial_{yy} u))\right] + O(h^3), \tag{31}$$

so that

$$\mathcal{T}_j^{\text{adv-diff}} = \mathcal{T}_j^{\text{adv}} + \mathcal{T}_j^{\text{diff}} = \frac{h^2}{12}\left[\partial_{xx} r + \partial_{xy} r + \partial_{yy} r\right] + O(h^3), \tag{32}$$

where

$$r = \partial_x f + \partial_y g - \nu\left(\partial_{xx} u + \partial_{yy} u\right), \tag{33}$$

and thereby the second-order error term vanishes for $r = 0$, i.e., in the steady state. We emphasize that there are two requirements for constructing a uniformly third-order advection-diffusion scheme. First, the diffusion scheme must have a second-order error term that vanishes in the steady state. Second, the second order error must be in the form compatible with that of the advection scheme.

The linear Galerkin scheme (i.e., the continuous P_1 Galerkin scheme), which is equivalent to the three-point central finite-difference scheme on the grid considered here[21], has a second-order error term,

$$\mathcal{T}_j^{\text{diff}} = \frac{\nu h^2}{12}(\partial_{xxxx} u + \partial_{yyyy} u) + O(h^3), \tag{34}$$

which does not vanish in the steady state, leading to

$$\begin{aligned}\mathcal{T}_j^{\text{adv-diff}} &= \mathcal{T}_j^{\text{adv}} + \mathcal{T}_j^{\text{diff}} \\ &= \frac{h^2}{12}\left[\partial_{xx}(\partial_x f + \partial_y g + \nu\partial_{xx} u) + \partial_{xy}(\partial_x f + \partial_y g) + \partial_{yy}(\partial_x f + \partial_y g + \nu\partial_{yy} u)\right] + O(h^3).\end{aligned}$$

Clearly, the second-order error term does not vanish. This scheme is, therefore, second-order accurate, and can be third-order accurate only in the advection limit ($\nu \to 0$). Without exploring various other diffusion schemes or seeking a general guiding principle, we took a third-order version of the linear Galerkin scheme described in Refs.[22, 23], which is obtained by upgrading the element-gradient by a curvature correction. The curvature correction term is computed from the gradients reconstructed at nodes. See Ref.[22] for details. Note that this third-order Galerkin scheme is a corrected linear Galerkin scheme, not a discontinuous Galerkin scheme nor the continuous P_2 Galerkin scheme[22]. This scheme has the following truncation error:

$$\mathcal{T}_j^{\text{diff}} = \frac{h^2}{12}\left[\partial_{xx}(\nu\left(\partial_{xx} u + \partial_{yy} u\right)) + \partial_{xy}(\nu\left(\partial_{xx} u + \partial_{yy} u\right)) + \partial_{yy}(\nu\left(\partial_{xx} u + \partial_{yy} u\right))\right] + O(h^3), \tag{35}$$

and thus it is third-order accurate in the steady state where $\nu\left(\partial_{xx} u + \partial_{yy} u\right) = 0$. However, the sum of the third-order advection scheme and the third-order Galerkin scheme has the following truncation error:

$$\mathcal{T}_j^{\text{adv-diff}} = \mathcal{T}_j^{\text{adv}} + \mathcal{T}_j^{\text{diff}} = \frac{h^2}{12}\left[\partial_{xx} r' + \partial_{xy} r' + \partial_{yy} r'\right] + O(h^3), \tag{36}$$

where

$$r' = \partial_x f + \partial_y g + \nu\left(\partial_{xx} u + \partial_{yy} u\right). \tag{37}$$

We immediately notice that the diffusive term in r' has a wrong sign, and therefore the second-order term does not vanish in the steady state where $r = 0$ but $r' \neq 0$. The scheme is only second-order accurate except in the advection limit or in the diffusion limit, i.e., not uniformly third-order accurate. This is the compatibility problem associated with the second requirement stated above. A similar discussion can be found in Ref.[4], which pertains to second- and high-order residual-distribution schemes.

There is a possibility that the edge-based diffusion scheme in Refs.[16, 24] can also achieve third-order accuracy with a vanishing second-order error term. However, our experience indicates that the scheme requires a cubic fit for the gradient reconstruction. Moreover, it is not immediately clear if the scheme is guaranteed to be compatible with the third-order advection scheme on arbitrary triangular grids. Nevertheless, a recent work reported in Ref.[20] shows that a similar edge-based diffusion scheme yields fourth-order accuracy with a cubic fit and appears to be compatible with the third-order advection scheme. Although encouraging, this type of scheme results in a very large stencil due to the cubic fit, and makes itself less attractive for practical implementations. A systematic construction of the gradient stencils based on high-order elements with a technique to damp high-frequency errors proposed in Ref.[20] may prove to overcome the drawback in practical applications. While the search continues for a compatible and economical third-order diffusion scheme, we show in the next section that the construction of uniformly third-order advection-diffusion schemes is *trivial* in the hyperbolic method. The compatibility problem does not exist because all terms are made hyperbolic and can be discretized in exactly the same way, thus yielding a fully compatible second-order error term.

3.6.2 Hyperbolic Formulation for Uniform Accuracy

Consider a fully hyperbolic formulation:

$$\partial_t \mathbf{U} + \partial_x \mathbf{F} + \partial_y \mathbf{G} = \mathbf{0}, \tag{38}$$

where

$$\mathbf{F} = \mathbf{F}^a + \mathbf{F}^d + \mathbf{F}^s = \begin{bmatrix} au \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -\nu p \\ -u/T_r \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ (y - y_j)\, q/T_r \\ -(x - x_j)\, q/T_r \end{bmatrix}, \tag{39}$$

$$\mathbf{G} = \mathbf{G}^a + \mathbf{G}^d + \mathbf{G}^s = \begin{bmatrix} bu \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -\nu q \\ 0 \\ -u/T_r \end{bmatrix} + \begin{bmatrix} 0 \\ -(y - y_j)\, p/T_r \\ (x - x_j)\, p/T_r \end{bmatrix}, \tag{40}$$

where (x_j, y_j) denotes the location of a node at which the system is discretized. This system is constructed by adding the advective term to the fully hyperbolic diffusion system introduced in Ref.[1]. As described in Ref.[1], the fully hyperbolic diffusion system is constructed by the divergence formulation of the source term [6], and the formulation is equivalent to the original system in the steady state. The main advantage of this particular divergence formulation is that we can avoid the computation of the second derivatives of the source terms, which is required in other techniques to achieve third-order accuracy for equations with source terms. See Ref.[1] for more details. Observe that each part is hyperbolic and can be discretized by the same third-order upwind scheme. The numerical flux is constructed simply as a sum of the upwind advection flux, the upwind hyperbolic-diffusion flux, and the upwind source flux. The discretization is thus written as

$$0 = -\sum_{k \in \{k_j\}} \Phi'_{jk} A_{jk}. \tag{41}$$

Observe that the source term in Equation (20) has been turned into a flux balance and incorporated into the modified flux function, Φ'_{jk}:

$$\Phi'_{jk} = \Phi^a_{jk} + \Phi^d_{jk} + \Phi^s_{jk}, \tag{42}$$

where Φ^a_{jk} and Φ^d_{jk} are the upwind advection and hyperbolic diffusion fluxes given in Equations (15) and (16). The upwind source flux, Φ^s_{jk}, is given by

$$\Phi^s_{jk} = \frac{1}{2}(\mathbf{H}^s_L + \mathbf{H}^s_R) \cdot \hat{\mathbf{n}}_{jk} - \frac{1}{2}|\mathbf{A}^s_n|(\mathbf{U}_R - \mathbf{U}_L), \tag{43}$$

where $\mathbf{H}^s_L = [\mathbf{F}^s_L, \mathbf{G}^s_L]$, $\mathbf{H}^s_R = [\mathbf{F}^s_R, \mathbf{G}^s_R]$, and $\mathbf{A}^s_n = (\partial \mathbf{F}^s/\partial \mathbf{U})n_x + (\partial \mathbf{G}^s/\partial \mathbf{U})n_y$. The absolute Jacobian is evaluated at the midpoint of the edge:

$$\begin{aligned} |\mathbf{A}^s_n| &= R^s_n |\Lambda^s_n| L^s_n \\ &= \begin{bmatrix} 0 & 0 & 0 \\ -\Delta y_{jm}/\Delta s_{jm} & 0 & 0 \\ \Delta x_{jm}/\Delta s_{jm} & 0 & 0 \end{bmatrix} \begin{bmatrix} |\Delta s_{jm}|/T_r & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & -n_y & n_x \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ &= \frac{|\Delta s_{jm}|}{\Delta s_{jm} T_r} \begin{bmatrix} 0 & 0 & 0 \\ 0 & n_y \Delta y_{jm} & -n_x \Delta y_{jm} \\ 0 & -n_y \Delta x_{jm} & n_x \Delta x_{jm} \end{bmatrix}, \end{aligned} \tag{44}$$

where

$$\Delta s_{jm} = \Delta x_{jm} n_x + \Delta y_{jm} n_y, \quad \Delta x_{jm} = x_m - x_j, \quad \Delta y_{jm} = y_m - y_j, \quad x_m = \frac{x_j + x_k}{2}, \quad y_m = \frac{y_j + y_k}{2}. \tag{45}$$

The left and right fluxes are linearly extrapolated from j and k:

$$\mathbf{H}_L^s \cdot \hat{\mathbf{n}}_{jk} = \mathbf{H}_j^s \cdot \hat{\mathbf{n}}_{jk} + \frac{1}{2}\nabla(\mathbf{H}_j^s \cdot \hat{\mathbf{n}}_{jk}) \cdot \Delta\mathbf{l}_{jk} = \frac{1}{2T_r}\begin{bmatrix} 0 \\ \omega_j \Delta y_{jk} \\ -\omega_j \Delta x_{jk} \end{bmatrix}, \tag{46}$$

$$\mathbf{H}_R^s \cdot \hat{\mathbf{n}}_{jk} = \mathbf{H}_k^s \cdot \hat{\mathbf{n}}_{jk} - \frac{1}{2}\nabla(\mathbf{H}_k^s \cdot \hat{\mathbf{n}}_{jk}) \cdot \Delta\mathbf{l}_{jk} = \frac{1}{2T_r}\begin{bmatrix} 0 \\ \Delta y_{jk}(\omega_k - \nabla\omega_k \cdot \Delta\mathbf{l}_{jk}) \\ -\Delta x_{jk}(\omega_k - \nabla\omega_k \cdot \Delta\mathbf{l}_{jk}) \end{bmatrix}, \tag{47}$$

where

$$\omega_j = n_x q_j - n_y p_j, \quad \omega_k = n_x q_k - n_y p_k, \quad \nabla\omega_j = n_x \nabla q_j - n_y \nabla p_j, \quad \nabla\omega_k = n_x \nabla q_k - n_y \nabla p_k. \tag{48}$$

The gradients, ∇p_j and ∇q_j, are computed by a quadratic fit; they are already available from the advective and diffusive parts of the algorithm. It is important to note that the above formula is valid only for the residual at node j. That is, the source flux must be computed separately for the two nodes, j and k. This is because the source fluxes in Equations (39) and (40) must be modified when it is computed for the residual at the node k with (x_j, y_j) replaced by (x_k, y_k), so that the differential system approximates the original system precisely at k [1, 6]. For the node-centered edge-based scheme, the dissipation term becomes identical, but the average flux term is different for j and k. For this reason, the discrete conservation does not hold for the source flux term as it should not (because it is a source term). See Ref.[6] for more details.

The left and right states used to evaluate the numerical fluxes are computed by the same linear extrapolation as in the second-order scheme. For the third-order scheme, however, the gradient computation must be exact for quadratic functions, i.e., a quadratic fit. It requires five neighbors and may extend beyond the edge-connected neighbors in some cases. In the previous paper[1], we selected 10 neighbors by including neighbors of the edge-connected neighbors as necessary, and stored the list of 10 neighbors at each node. Here, we avoid carrying information on the neighbors of the neighbors by implementing the quadratic gradient reconstruction in two steps, where each step is compact, as described in Appendix. This method involves all neighbors of the edge-connected neighbors. The total number of neighbors can be as large as 18, and it can be much more than necessary in many cases. A smart selection of a minimal number of neighbors may be possible, but in this study we employ the two-step method for robustness and simplicity. It is robust as it has far more neighbors than necessary even at boundary nodes, and simple as it can be implemented with the list of edge-connected neighbors only. The latter can be helpful especially in a parallel code, where typically only the edge-connected neighbors are available across partitions.

The resulting scheme is guaranteed to be uniformly third-order accurate because each term is a hyperbolic system and the third-order scheme has already been demonstrated for hyperbolic systems [17, 18, 19]. Here, we just show that the compatibility problem mentioned in the previous section is resolved by the hyperbolic formulation. Expanding the scheme on the regular triangular grid considered in the previous section, we find the truncation error for each equation as

$$\begin{aligned} \mathcal{T}_j^u &= -\frac{\nu h}{6L_r}\left[(\sqrt{2}+\sqrt{5})\partial_x(p-\partial_x u) + \sqrt{2}\,\partial_y(p-\partial_x u) + \sqrt{2}\,\partial_x(q-\partial_y u) + (\sqrt{2}+\sqrt{5})\partial_y(q-\partial_y u)\right] \\ &+ \frac{h^2}{12}\left[\partial_{xx} r + \partial_{xy} r + \partial_{yy} r\right] + O(h^3), \end{aligned} \tag{49}$$

$$\mathcal{T}_j^p = -\frac{h^2}{6T_r}\left[(\partial_{xx}+\partial_{xy})(q-\partial_y u) + \partial_{xx}(p-\partial_x u) + \partial_y(\partial_x q - \partial_y p)\right] + O(h^3), \tag{50}$$

$$\mathcal{T}_j^q = -\frac{h^2}{6T_r}\left[(\partial_{xy}+\partial_{yy})(p-\partial_x u) + \partial_{yy}(q-\partial_y u) - \partial_x(\partial_x q - \partial_y p)\right] + O(h^3), \tag{51}$$

where $r = a\,\partial_x u + b\,\partial_y u - \nu(\partial_x p + \partial_y q)$. Observe that there are first- and second-order error terms but they all vanish in the steady state where $r = 0$, $p - \partial_x u = 0$, $q - \partial_y u = 0$, and $\partial_x q - \partial_y p = 0$. The most critical term is the second-order error term in the first equation, which vanishes in the steady state for the advection-diffusion equation, meaning that the advective and diffusive terms have been discretized in a perfectly compatible manner.

The scheme is, therefore, uniformly third-order accurate for any set of parameters, (a, b) and ν, and also for unstructured triangular grids as already demonstrated for hyperbolic systems in Refs.[17, 18, 19].

We emphasize again that the third-order finite-volume scheme itself has already been studied for hyperbolic systems in Refs.[17, 18, 19]. In the above, we have shown that instead of developing a compatible third-order scheme for diffusion (and source), we can reformulate the diffusion term as a hyperbolic system and directly apply the third-order scheme to the advective and diffusive terms to produce a uniformly third-order accurate advection-diffusion scheme. More importantly, we have shown that the uniform third-order accuracy is achieved simply by adding the upwind hyperbolic diffusion scheme to the upwind advection scheme. A full integration of the advective and diffusive terms, including the source flux, into a single hyperbolic system is not necessary although it is possible.

4 Numerical Results

4.1 Isotropic Grids

We consider the steady advection-diffusion problem in a square domain with the exact solution given by [4]

$$u(x, y) = \cos(2\pi\eta) \exp\left(\frac{-2\pi^2\nu}{1 + \sqrt{1 + 4\pi^2\nu^2}}\,\xi\right), \tag{52}$$

where $\xi = ax + by$, $\eta = bx - ay$, and with the Dirichlet boundary condition. The advection vector is set as $(a, b) = (1.23, 0.12)$ and ν is determined from the parameter Re by

$$\nu = \frac{\sqrt{a^2 + b^2}}{Re}, \tag{53}$$

for $Re = 10^{-6}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3, 10^6$. Numerical results are presented for a series of independently generated eight irregular triangular grids with N nodes, where N =2048, 8192, 18432, 32768, 51200, 73728, 100352, 131072. The coarsest grid is shown in Figure 2. As can be seen, the grid is fully irregular with random number of neighbors and some vanishingly small volumes. The first, second, and third order hyperbolic advection-diffusion schemes, designated as SchemeII(1st), SchemeII(2nd), and SchemeII(3rd), are compared with two traditional schemes. One is the third-order advection scheme with the linear Galerkin scheme, and the other with the third-order Galerkin scheme considered in Section 3.6.1. The former is designated as Galerkin, and the latter as Galerkin(3rd). These schemes are termed 'traditional' because they are scalar schemes directly solving the advection-diffusion equation (1). The implicit iterative method is implemented with the first-order residual Jacobian for the advective part and the exact linear-Galerkin Jacobian for the diffusive part for both traditional schemes. For the Galerkin(3rd) scheme, the Jacobian for the diffusive part is, therefore, approximate. See Table 1 for a summary of discretizations and Jacobians. In all cases, the GS relaxation is terminated when the residual of the linear system is reduced by two orders of magnitude in the L_1 norm. For gradient reconstruction, the unweighted linear LSQ method is used for SchemeII(2nd), and the two-step quadratic LSQ method as in Appendix is used for SchemeII(3rd) and the two traditional schemes. In all cases, the initial solution is set by the exact solution randomly perturbed. Steady convergence is taken to be reached when the residual in the L_1 norm drops by ten orders of magnitude or reaches the machine zero.

Error convergence results are shown in Figures 3 to 10. Only the results for u and p are shown because the results for q is very similar to those for p. Also, the results for $Re = 10^{-3}$ are not shown because they look identical to those for $Re = 10^{-6}$. The legend is shown in the error convergence plot for p only; it applies equally to the plot for u on the left side. For the traditional schemes, p corresponds to the x-component of the quadratic LSQ gradients. First, it is observed that the Galerkin scheme is second-order accurate in the solution and first-order accurate in the gradients (even with the quadratic LSQ fit) except in the advection limit $Re = 10^6$ (Figure 10) where the third-order advection scheme dominates and yields third- and second-order accuracy in the solution and the gradients, respectively. Second, the Galerkin(3rd) scheme gives third-order accuracy in the solution and second-order accuracy in the gradients in the diffusion dominated cases, but the accuracy begins to deteriorate by one order at $Re = 1$ (Figure 6) and starts recovering third-order accuracy at $Re = 10^3$ (Figure 9). As predicted, therefore, the scheme is not uniformly third-order accurate. On the other hand, SchemeII(1st), SchemeII(2nd), and SchemeII(3rd) never encounter such accuracy deterioration. As results show, SchemeII(1st) and SchemeII(2nd) are uniformly first- and second-order accurate, respectively, in both the solution and the gradients. Observe also that as advection dominates ($Re = 10^3, 10^6$), SchemeII(1st)

and SchemeII(2nd) yield one order higher accuracy in the solution, i.e., second-order and third-order accuracy, respectively. The second/third-order accuracy is expected because the gradients are computed with first/second-order accuracy, which is employed in the face-reconstruction for the advection scheme. This way of achieving second/third-order accuracy for advection is interesting and economical: second-order advection scheme with no gradient computations, and third-order advection scheme with linear (not quadratic) LSQ gradients. In effect, the hyperbolic diffusion scheme here plays a single role of providing accurate gradients with the diffusive term kept negligibly small. Finally, observe that SchemeII(3rd) gives third-order accuracy for both the solution and the gradients for all values of Re. It is worth emphasizing that the gradients are third-order accurate even in the advection limit. The third-order scheme is very efficient in terms of the accuracy in the gradients as we shall discuss next.

Iterative convergence results are presented in Figures 11 to 18. The same legend applies as in the error convergence plots, and therefore it is not shown. Again, the results for $Re = 10^{-3}$ are not shown because they look identical to those for $Re = 10^{-6}$. In each figure, three plots are shown: total iterations for convergence, the total number of the GS relaxations, and the total CPU time taken for convergence, all versus $1/h$, where $h = 1/\sqrt{N}$. First, it is observed that the convergence characteristics of the Galerkin(3rd) scheme are nearly identical to those of the Galerkin scheme. This is an unexpected result because the Jacobian is exact for the Galerkin scheme but only approximate for the Galerkin(3rd) scheme. The results imply that the Jacobian of the linear Galekrin scheme behaves like exact for the third-order Galerkin scheme. Second, the number of GS sweeps increases quadratically as the grid gets finer in the diffusion dominated cases while increases linearly in the advection dominated cases. It is consistent with the change in the condition number of the Jacobian matrix from $O(1/h^2)$ in the diffusion limit to $O(1/h)$ in the advection limit. In terms of CPU time, as shown in the right-most plot, the cost of these traditional schemes varies from $O(1/h^4)$ or equivalently $O(N^2)$ in the diffusion limit to $O(1/h^3)$ or $O(N^{1.5})$ in the advection limit. On the other hand, the hyperbolic schemes preserve, *for all values of Re*, $O(N^{1.5})$ convergence in the CPU time. This result is a direct consequence of solving the hyperbolic advection-diffusion system instead of the scalar advection-diffusion equation. In the diffusion limit, the hyperbolic schemes are, therefore, $O(1/h)$ times faster than the traditional schemes. Note that the acceleration factor grows as the grid gets finer (i.e., as $h \to 0$). In the advection limit, both traditional and hyperbolic schemes show $O(N^{1.5})$ convergence in the CPU time. Results for $Re = 10^6$ show that the traditional schemes are (2 or 3 times) faster than the hyperbolic schemes, which is expected because the hyperbolic schemes solve two additional equations. However, it is not immediately clear if the traditional schemes are more efficient because the third-order hyperbolic scheme is capable of delivering third-order accurate gradients. If we focus on the accuracy in the gradients, the third-order hyperbolic scheme is to be compared with a fourth-order scalar advection-diffusion scheme. The cost of the third-order hyperbolic scheme being comparable with that of the second-order scheme, it implies a tremendous potential advantage of the third-order hyperbolic scheme for applications where accurate gradients are sought. Note also that SchemeII(2nd) achieves third-order accuracy in the advection limit with the linear LSQ gradients, not the quadratic LSQ gradients.

Table 2 shows how the number of iterations and GS sweeps vary with Re for the finest grid. For the traditional schemes, the number of GS sweeps increases significantly as diffusion dominates. Also, the the number of iterations increases as advection dominates because of the approximate Jacobian. For the hyperbolic schemes, the number of iterations does not vary significantly, but the number of GS sweeps somewhat increases as diffusion dominates. If desired, it may be possible to reduce the number of GS sweeps by deriving an optimal L_r for the numerical scheme in the diffusion limit as in Ref.[2], not for the differential equations as in Refs.[7]. Yet, we emphasize that the hyperbolic schemes are already an order-of-magnitude more efficient than the traditional schemes in the diffusion limit. See table 3 for the CPU time comparison for the finest grid. It should be noted that this comparison is just for a given grid, and further acceleration is expected for finer grids because the hyperbolic schemes yield $O(1/h)$ acceleration in iterative convergence over conventional schemes.

Finally, we mention that a third-order advection-diffusion scheme developed in Ref.[25] is also capable of producing third-order accurate solution and gradients. In their scheme, the hyperbolic formulation is not used, and the third-order gradients are obtained by an explicit gradient reconstruction. The main and important difference between their scheme and our scheme is that their scheme requires third-order accurate gradients to achieve third-order accuracy for the diffusion term. The same requirement is also found in the third-order scheme of Ref.[20], which again does not use the hyperbolic method, where a cubic fit is used to provide third-order accurate gradients for the viscous terms. In contrast, our scheme requires only second-order accurate gradient reconstruction, which requires only 5 neighbors rather than 9 neighbors in a cubic fit in two dimensions, and achieves third-order accuracy for the solution and the gradients. In three dimensions, a quadratic fit requires 9 neighbors, which can be secured by the neighbors in many cases in tetrahedral grids, while a cubic fit requires

21 neighbors, which requires the neighbors of the neighbors in many cases (or high-order elements as proposed in Ref.[20]). Moreover, our implicit solver is more efficient in terms of iterative convergence, firstly by the consistent Jacobian exact for the first-order scheme (their scheme would perhaps have to rely on inconsistent Jacobians for the diffusive part), and secondly by the reduced condition number, $O(1/h)$ instead of typical $O(1/h^2)$, of the linearized residual system as a consequence of solving the first-order hyperbolic system.

4.2 Derivative Prediction on Irregular Anisotropic Grid

The second test problem focuses on the accuracy of the derivatives computed on the boundary in anisotropic grids. It is a model problem for viscous boundary-layer calculations. An anisotropic grid was generated from an isotropic irregular grid (1089 nodes) by shrinking the y-coordinate. Figure 19 shows the resulting grid. Note that the domain extends to 1.0 in x direction but only to 0.01 in the y direction, leading to the aspect ratio of approximately 100. As a result, the grid is highly skewed, and many conventional diffusion schemes are known to produce oscillatory solutions on such grids [16, 24]. We use the same exact solution as in the previous test case, and set $Re = 10$. Four schemes have been tested for this case: two second-order schemes, Galerkin and SchemeII(2nd); two third-order schemes, Galerkin(3rd) and SchemeII(3rd). The solution u is specified on the boundary for all schemes. For the hyperbolic schemes, the variables p is specified on the boundary, and q is specified on the left, right, and top boundary. On the bottom boundary indicated by the thick red line in Figure 19 (which may be considered as a wall), the variable q, corresponding the y-derivative of u, is computed by the schemes. The residual at a boundary node is closed by the physical flux evaluated at the node. For Galerkin and Galerkin(3rd), the y-derivative is computed from the solution by the quadratic least-squares method. The y-derivative on the bottom boundary is the normal derivative, modeling the viscous stress in viscous boundary-layer computations. Figure 20 shows the results obtained by the second-order schemes. It shows that the normal derivative obtained by Galerkin is very oscillatory. This result is expected by the fact that the Galerkin discretization is not positive on such a grid, and also from the results shown in Refs.[16, 24]. On the other hand, the normal derivative computed by SchemeII(2nd) is highly accurate. The computed derivative values are on top of the curve of the exact derivative. Note that Galerkin uses the third-order advection scheme. The results show that the third-order accuracy in the advective term does not compensate the failure of the Galerkin discretization for this problem. Figure 21 shows the results obtained by third-order schemes. Observe that Galerkin(3rd) gives much better results than Galerkin, but still somewhat oscillatory. SchemeII(3rd), on the other hand, produces a highly accurate normal derivative distribution along the boundary. Figure 22 shows the blow-up of Figure 21, where the high accuracy delivered by SchemeII(3rd) can be clearly seen: the numerically computed derivatives are still on top of the exact curve. These results indicate that the hyperbolic schemes allow highly arbitrary grid generation or adaptation on viscous grids, even through the boundary layer, without degrading the accuracy in the derivative quantities such as the viscous stresses and the heat fluxes.

To verify the order of accuracy and demonstrate the efficiency of the hyperbolic schemes, we performed the computation over a series of grids: 1089, 4225, 16641, 66049, and 103041 nodes. The error convergence results are shown in Figure 23. The results show that the design order of accuracy has been achieved for the hyperbolic schemes at boundary nodes. On the other hand, Galerkin and Galerkin(3rd) show one order lower order accuracy due to the incompatibility of the advection and diffusion schemes. Note that the error is defined as the difference between the predicted q and the exact q on the bottom boundary only; it does not include errors at the interior nodes. As expected from the previous qualitative comparison, we see that the hyperbolic schemes yield significantly lower errors. SchemeII(3rd) gives the lowest level of errors: more than one order of magnitude smaller than Galerkin(3rd). SchemeII is a second-order scheme, but it yields significantly lower errors than Galerkin(3rd). Moreover, the hyperbolic schemes are very efficient. Figure 24 shows the CPU time comparison. As can be clearly seen, although the hyperbolic schemes are slower in the coarsest grid, they get faster than the conventional schemes as the grid is refined. The hyperbolic schemes are intrinsically faster as indicated by the slopes in the figure: $O(1/h^3)$ or equivalently $O(N^{1.5})$ for the hyperbolic schemes, and $O(1/h^4)$ or equivalently $O(N^2)$ for the conventional schemes. These results demonstrate once again that the hyperbolic schemes achieve $O(1/h)$-acceleration in convergence over conventional schemes. It is striking that the hyperbolic schemes are capable of achieving higher-order accuracy as well as higher-order accurate derivatives through the boundary over conventional schemes at a *reduced* cost.

5 Concluding Remarks

We have extended the diffusion schemes developed in the previous paper[1] to the advection-diffusion equation, generating uniformly accurate first, second, and third order advection-diiffusion schemes on unstructured triangular grids. The advective, diffusive, and source terms have been discretized in a unified framework by the method of hyperbolic systems that converts the diffusive and source terms into hyperbolic systems. The developed hyperbolic schemes are node-centered edge-based finite-volume schemes with the upwind flux for all terms. An implicit iterative method has been developed for all schemes based on the exact Jacobian of the first-order scheme. Also, a two-step implementation of the quadratic LSQ gradient reconstruction has been proposed for robustness and simplicity, in which each step is compact, requiring only the list of the edge-connected neighbors.

The developed schemes were compared with two traditional schemes: the third-order advection scheme with the linear Galerkin diffusion scheme, and with a third-order version of the linear Galerkin diffusion scheme[22, 23]. It was shown analytically as well as numerically that the latter scheme cannot be third-order accurate when advection and diffusion are equally important ($Re = 1, 10, 10^2$ in Figures 6, 7, 8) while the former is third-order accurate only in the advection limit. Typically, the accuracy is deteriorated by one order, and in a critical case ($Re = 100$), the deterioration begins to appear on fine grids. We emphasize again that the third-order Galerkin diffusion scheme is a corrected version of the linear Galerkin diffusion scheme as described in Ref.[22], not the discontinuous Galerkin scheme nor the continuous P_2 Galerkin scheme. On the other hand, the developed schemes have been confirmed to be uniformly accurate up to the design (or higher) order accuracy for all values of Re, i.e., $Re = 10^{-6}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3, 10^6$. Specifically, the first-order scheme has been shown to yield first-order accurate solution and gradients, and second-order accurate solution in the advection limit. The second-order scheme has been shown to yield second-order accurate solution and gradients, and third-order accurate solution in the advection limit. The third-order scheme has been shown to yield uniformly third-order accurate solution and gradients.

It was also shown that the first- and second-order schemes become second- and third-order schemes in the advection limit ($Re = 10^6$ in Figure 10). Specifically, the developed first-order scheme yields second-order accurate solutions and first-order accurate gradients in a compact stencil (no gradient reconstruction) with fast Newton-like convergence. The developed second-order scheme yields third-order accurate solutions and second-order accurate gradients with linear LSQ gradients (not quadratic LSQ gradients). These results demonstrate that the upwind hyperbolic diffusion scheme is capable of producing accurate gradients even with a vanishingly small diffusive coefficient. It implies that a compact second-order inviscid scheme may be constructed by incorporating the upwind hyperbolic diffusion scheme to compute accurate gradients. The resulting scheme will not require explicit gradient computations for second-order accuracy, and require only linear LSQ gradients for third-order accuracy.

For iterative convergence, the hyperbolic schemes have been shown to bring $O(1/h)$ acceleration in convergence over traditional schemes except in the advection limit. No such acceleration was observed in the advection limit as expected, and the developed schemes converged somewhat slower than the traditional schemes in the test problem considered. However, there are decisive advantages in the hyperbolic schemes, which overwhelm the slightly slower convergence in the advection limit. First, in the advection limit, the first- and second-order schemes achieve one order higher accuracy in the advection limit as mentioned above. Second, the third-order scheme produces third-order accurate gradients, which typically requires fourth-order schemes.

A particularly important contribution of this paper is the demonstration of the simplified approach: construct an advection-diffusion scheme as a sum of an advection scheme and a hyperbolic diffusion scheme. The approach has been shown to work well in terms of accuracy as well as efficiency for a wide range of parameter Re from the diffusion limit ($Re = 10^{-6}$) to the advection limit ($Re = 10^6$). For the compressible Navier-Stokes equations, it dramatically simplifies the construction of numerical schemes as a viscous scheme can be developed independently from the inviscid scheme. As shown in Ref.[8], the viscous terms can be made a hyperbolic system, its eigen-structure can be fully analyzed, and therefore discretized easily by the upwind scheme or any other scheme suitable for hyperbolic systems. It can then be added to any inviscid scheme to construct a hyperbolic Navier-Stokes scheme.

It has been shown also that the hyperbolic schemes are capable of producing highly accurate derivatives along a boundary on a highly skewed grid. For a test case modeling a viscous boundary layer, the second- and third-order hyperbolic schemes produce highly accurate derivatives normal to the boundary while both second- and third-order Galerkin schemes suffer from oscillations. It demonstrates the potential of the hyperbolic schemes for overcoming the difficulties in predicting viscous and heat fluxes on unstructured grids [26]. Moreover, it was demonstrated that the hyperbolic schemes converge rapidly for this problem as well, giving $O(1/h)$-acceleration

over conventional schemes. It is noteworthy that the hyperbolic schemes are capable of producing higher-order accurate derivatives through the boundary at a significantly *reduced* cost for fine grids.

We emphasize again that the core idea of the hyperbolic method lies in the construction of a first-order hyperbolic system for target differential equations, and therefore it is generally applicable to any second-order partial differential equation as well as any discretization method. Applications to other types of equations are now made particularly easier as we have shown in this paper that numerical schemes can be constructed for each non-hyperbolic term independently by turning it into a hyperbolic system. However, extensions to third- and higher-order derivative terms would require a careful construction of a first-order system to ensure the hyperbolicity. It remains to be demonstrated and left as future work.

It should be noted that the schemes generated based on the hyperbolic method cannot be time accurate with explicit time integration schemes. Time-accurate computations are possible by implicit time integration schemes, including space-time methods. The development of time-accurate hyperbolic schemes based on the backward-difference time integration has been demonstrated in Ref.[14] for a second-order residual-distribution method and in Ref.[27] for a very high-order residual-distribution method. The methodology is immediately applicable to the finite-volume methods as an efficient steady solver is in place and can be used to solve a system of globally coupled equations arising from the implicit time integration scheme. Note finally that the number of extra variables required in the hyperbolic method can be substantially high for complex systems. For the three-dimensional compressible Navier-Stokes equations, 6 viscous stresses and 3 heat fluxes will be required, at least. Consequently, the memory requirement is tripled compared with a conventional Navier-Stokes solver although the factor may be less than three since there are quantities such as grid metrics that do not scale with the number of equations. Note, on the other hand, that the increase in the degrees of freedom is dramatically small compared with other high-order methods, e.g., the discontinuous Galerkin method, especially for the third-order finite-volume scheme (which should be compared with a fourth-order discontinuous Galerkin scheme in terms of the accuracy in the derivatives). Further study is required to determine if the advantages of the hyperbolic method overwhelm any complication arising from a large number of extra variables for the Navier-Stokes equations, and that the hyperbolic method offers a resolution of problems encountered by the current state-of-the-art CFD solvers.

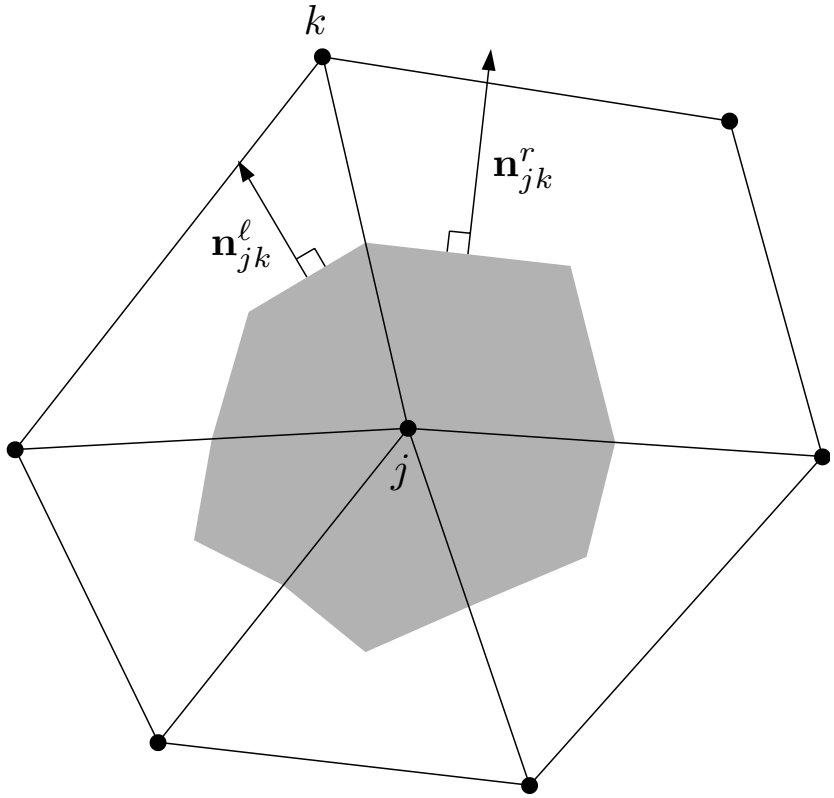

Figure 1: Dual control volume for the node-centered finite-volume method with scaled outward normals associated with an edge, $\{j, k\}$.

Scheme	Discretization		Jacobian	
	Advection	Diffusion	Advection	Diffusion
Galerkin	Third-order upwind	Linear Galerkin	First-order upwind	Exact
Galerkin(3rd)	Third-order upwind	Third-order Galerkin	First-order upwind	Linear Galerkin
SchemeII(1st)	First-order upwind		Exact	
SchemeII(2nd)	Second-order upwind		First-order upwind	
SchemeII(3rd)	Third-order upwind		First-order upwind	

Table 1: Summary of discretizations and Jacobians.

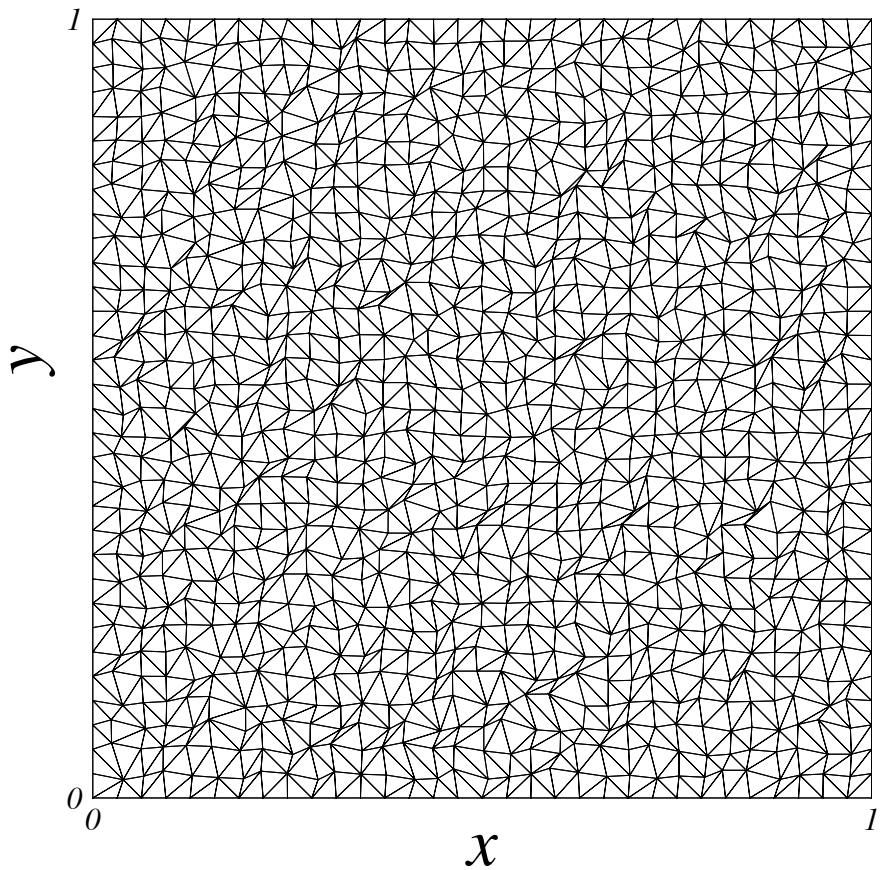

Figure 2: Irregular triangular grid with 2048 nodes.

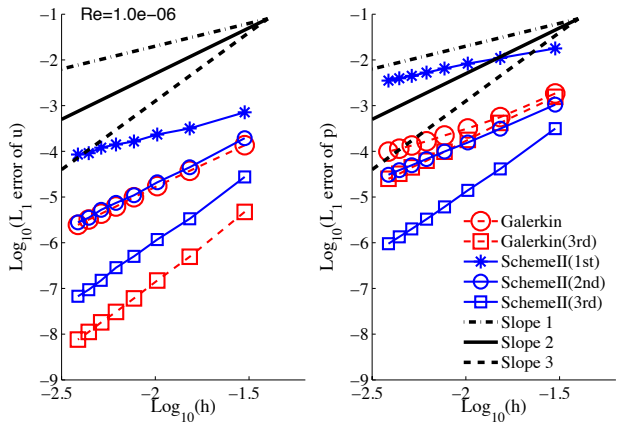

Figure 3: Case $Re = 10^{-6}$

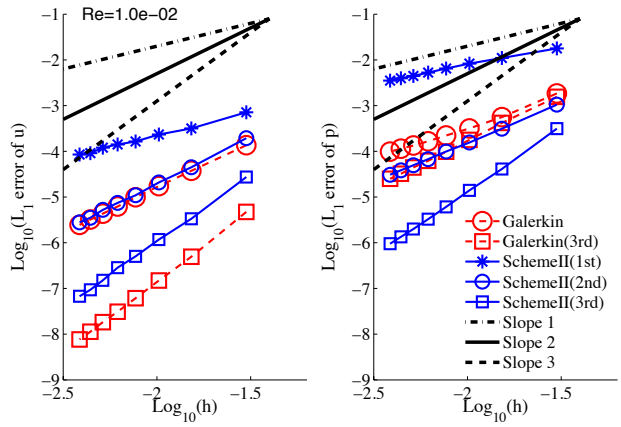

Figure 4: Case $Re = 10^{-2}$

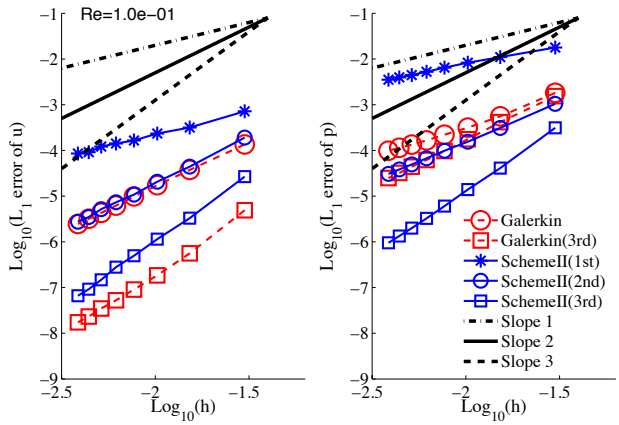

Figure 5: Case $Re = 10^{-1}$

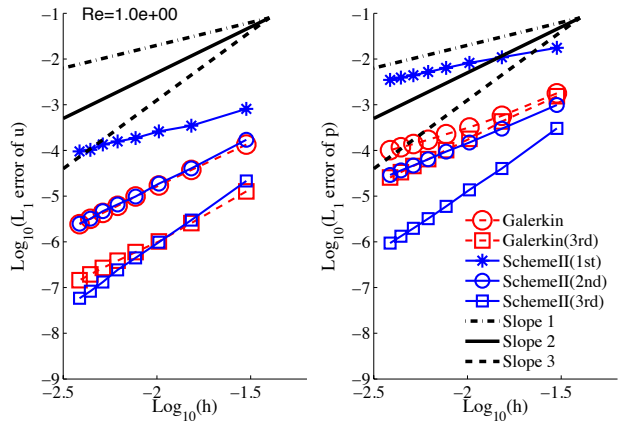

Figure 6: Case $Re = 1$

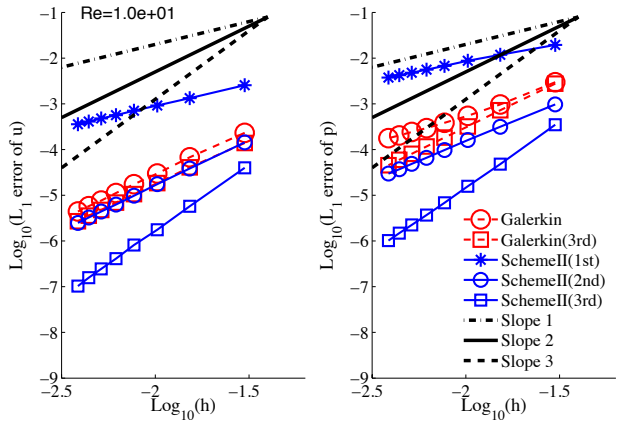

Figure 7: Case $Re = 10$

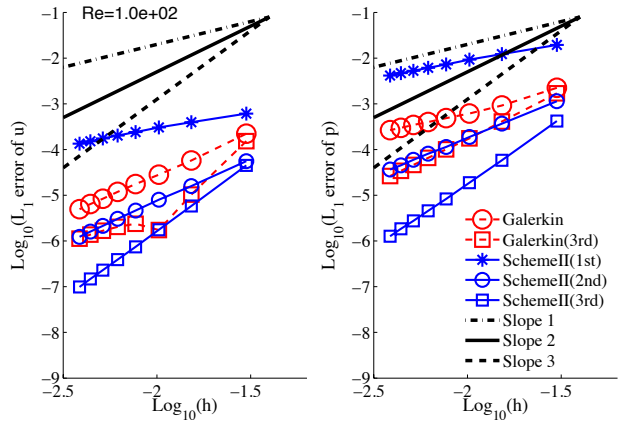

Figure 8: Case $Re = 10^2$

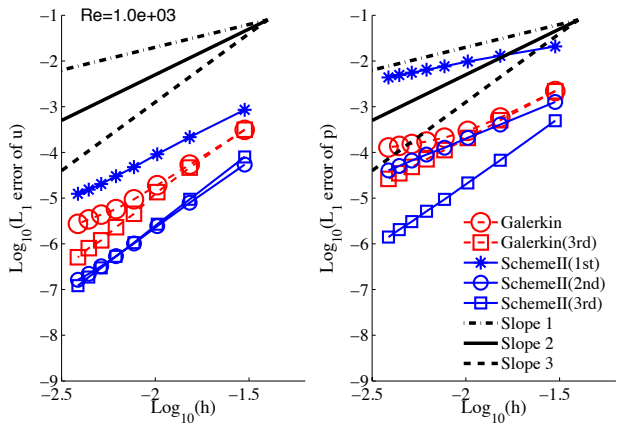

Figure 9: Case $Re = 10^3$

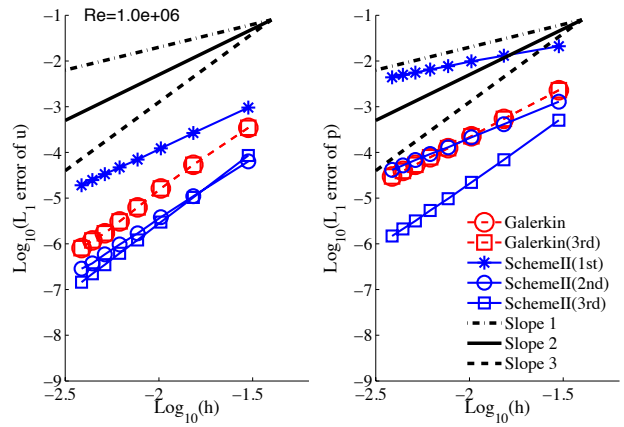

Figure 10: Case $Re = 10^6$

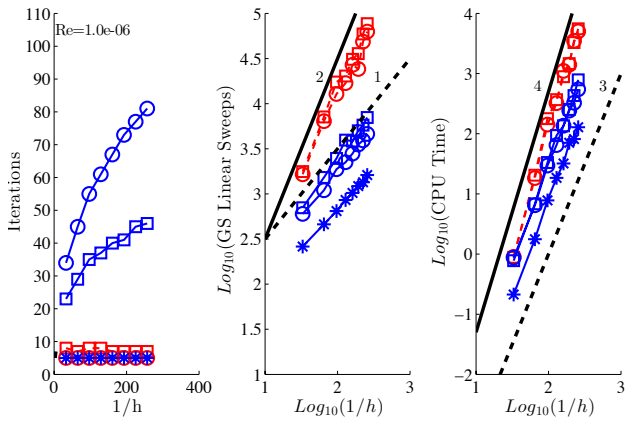

Figure 11: Case $Re = 10^{-6}$. The same legend applies as in Figs.3-10 except for slopes.

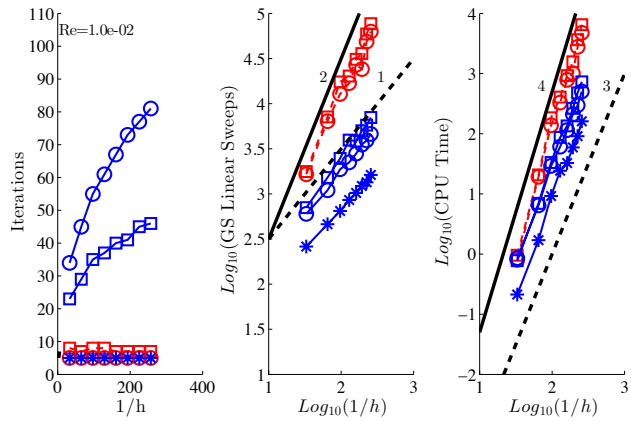

Figure 12: Case $Re = 10^{-2}$. The same legend applies as in Figs.3-10 except for slopes.

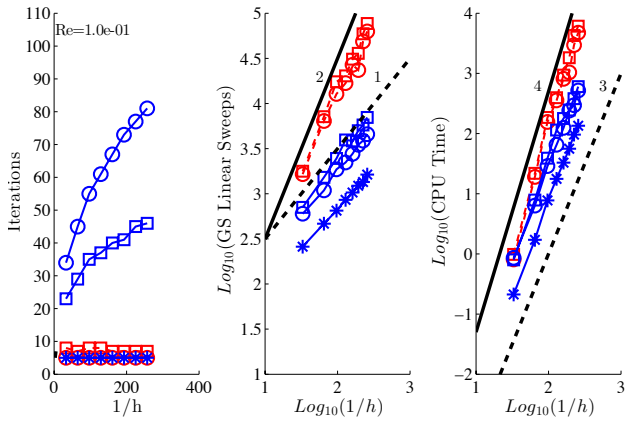

Figure 13: Case $Re = 10^{-1}$.The same legend applies as in Figs.3-10 except for slopes.

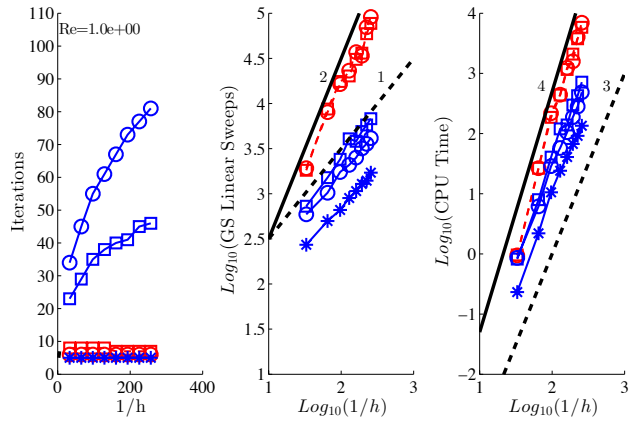

Figure 14: Case $Re = 1$. The same legend applies as in Figs.3-10 except for slopes.

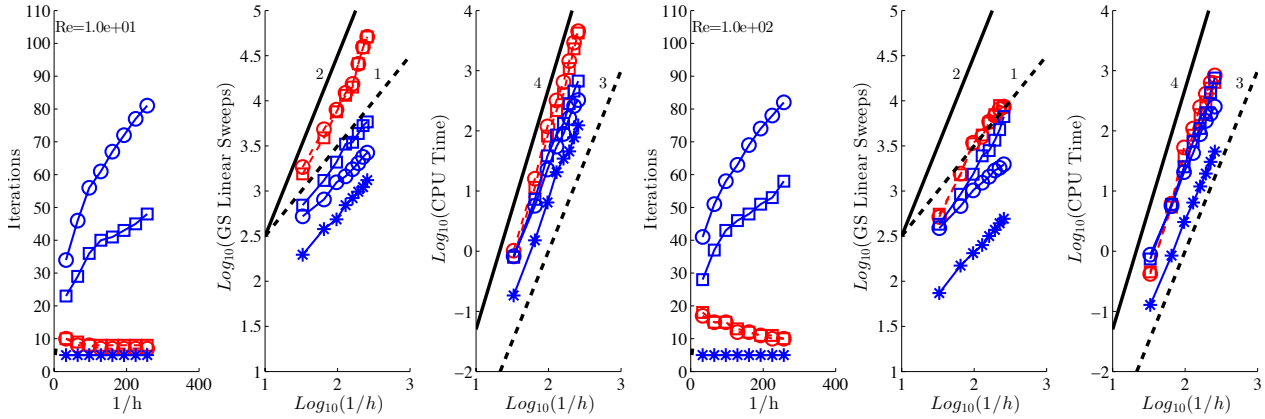

Figure 15: Case $Re = 10$. The same legend applies as in Figs.3-10 except for slopes.

Figure 16: Case $Re = 10^2$. The same legend applies as in Figs.3-10 except for slopes.

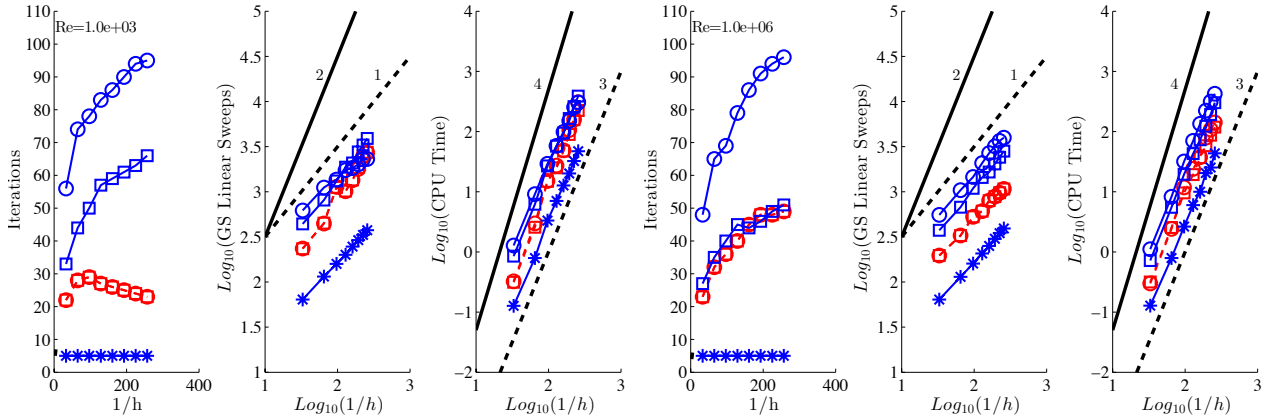

Figure 17: Case $Re = 10^3$. The same legend applies as in Figs.3-10 except for slopes.

Figure 18: Case $Re = 10^6$. The same legend applies as in Figs.3-10 except for slopes.

Scheme	$Re = \sqrt{a^2+b^2}/\nu$								
	10^{-6}	10^{-3}	10^{-2}	10^{-1}	1	10	10^2	10^3	10^6
Galerkin	5(12564)	5(12564)	5(12565)	5(12567)	6(15124)	7(7292)	10(861)	23(118)	49(22)
Galerkin(3rd)	7(10987)	7(10987)	7(10989)	7(11003)	7(11018)	8(6385)	10(848)	23(119)	49(22)
SchemeII(1st)	5(324)	5(324)	5(324)	5(326)	5(343)	5(264)	5(98)	5(75)	5(78)
SchemeII(2nd)	77(58)	77(58)	77(58)	77(57)	77(52)	77(34)	78(25)	93(24)	94(42)
SchemeII(3rd)	46(151)	46(151)	46(151)	46(151)	46(146)	46(122)	58(116)	66(59)	51(56)

Table 2: Total number of iterations in the finest grid case. The number in the parenthesis is the average number of GS-sweeps per iteration required to ensure two orders of magnitude reduction in the residual of the linear system.

Scheme	$Re = \sqrt{a^2+b^2}/\nu$								
	10^{-6}	10^{-3}	10^{-2}	10^{-1}	1	10	10^2	10^3	10^6
Galerkin	5.0e+03	4.8e+03	4.8e+03	4.8e+03	7.0e+03	4.5e+03	8.3e+02	3.1e+02	1.4e+02
Galerkin(3rd)	5.5e+03	5.6e+03	6.5e+03	6.1e+03	5.8e+03	4.2e+03	6.4e+02	2.2e+02	1.2e+02
SchemeII(1st)	1.4e+02	1.3e+02	1.3e+02	1.3e+02	1.4e+02	1.1e+02	4.3e+01	3.9e+01	3.6e+01
SchemeII(2nd)	5.0e+02	4.7e+02	5.2e+02	4.4e+02	4.0e+02	2.9e+02	2.5e+02	2.8e+02	4.2e+02
SchemeII(3rd)	7.2e+02	7.2e+02	7.1e+02	6.4e+02	5.8e+02	5.3e+02	5.9e+02	3.9e+02	3.0e+02

Table 3: Total CPU time for convergence in the finest grid case.

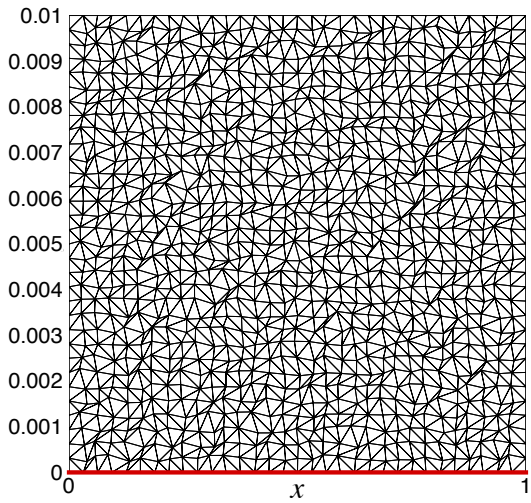

Figure 19: Grid used for the second test problem. 1089 nodes. Aspect ratio of approximately 100 (Note the different scalings in x and y axes). Thick red line indicates the boundary along which the normal derivative is computed by numerical schemes.

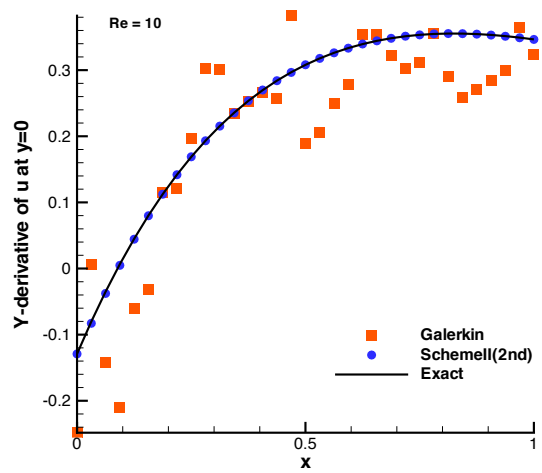

Figure 20: Wall normal derivative, $q = u_y$, distribution along the bottom boundary ($y = 0$) obtained by Galerkin and SchemeII(2nd) on the grid in Figure 19.

Appendix: Two-Step Implementation of Quadratic LSQ Gradient Reconstruction

First, we compute and store the coefficients for the gradient once for a given grid with a set $\{J\}$ of nodes.

1. *Construct temporary data:*

   For each node $j \in \{J\}$, loop over the edge-connected neighbors $k \in \{k_j\}$ and store the edge-vector:

$$\Delta \mathbf{x}_j(k) = \mathbf{x}_k - \mathbf{x}_j, \tag{A.1}$$

   where $\Delta \mathbf{x}_j(k) = (\Delta x_j(k), \Delta y_j(k))$, $\mathbf{x}_k = (x_k, y_k)$, and $\mathbf{x}_j = (x_j, y_j)$. One can skip this step if the data is already available in a code.

2. *Compute gradient coefficients:*

   For each node $j \in \{J\}$, loop over the neighbors of k, $\ell \in \{\ell_k\}$ within the loop over the edge-connected neighbors $k \in \{k_j\}$:

$$\Delta \mathbf{x} = \Delta \mathbf{x}_j(k) + \Delta \mathbf{x}_k(\ell), \tag{A.2}$$

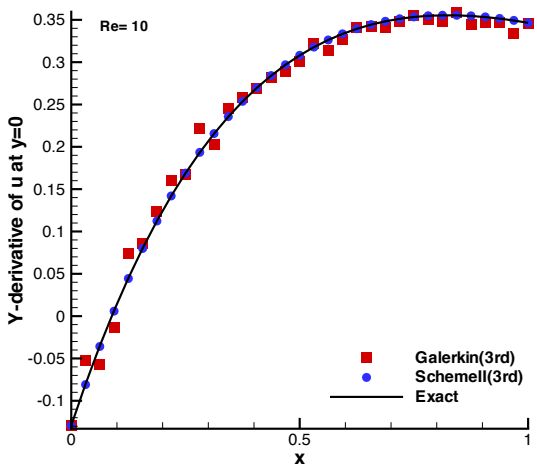

Figure 21: Wall normal derivative, $q = u_y$, distribution along the bottom boundary ($y = 0$) obtained by Galerkin(3rd) and SchemeII(3rd) on the grid in Figure 19.

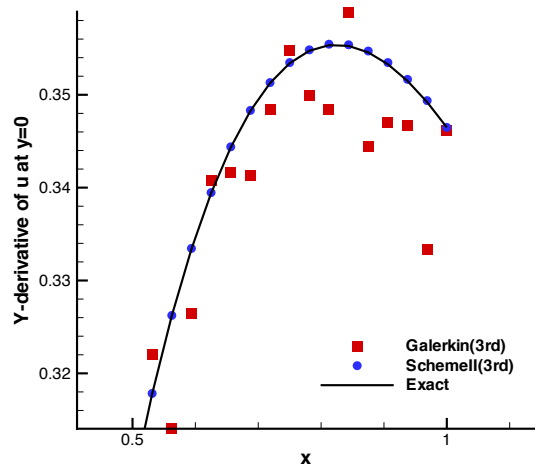

Figure 22: Blow-up of Figure 21.

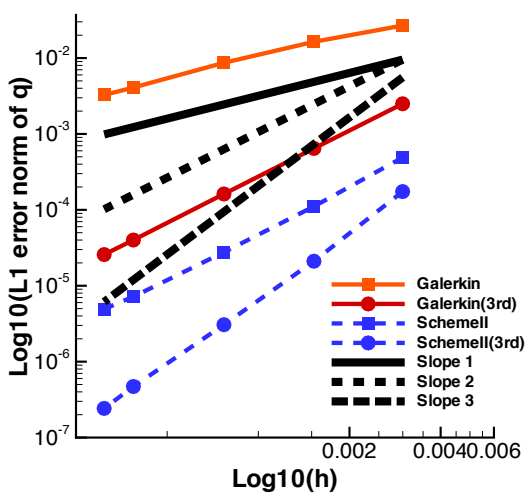

Figure 23: Error convergence of the wall normal derivative, $q = u_y$, distribution along the bottom boundary ($y = 0$).

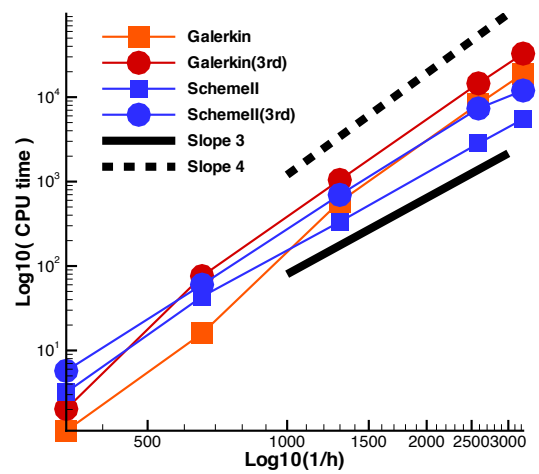

Figure 24: CPU time comparison.

and accumulate each entry of a 5×5 LSQ matrix for a quadratic fit, $\mathbf{A}_{LSQ}$. Note that $\Delta\mathbf{x} = 0$ if the neighbor coincides with the node j. It is not necessary, but we reset $\Delta\mathbf{x} = \Delta\mathbf{x}_j(k)$ in that case to increase the contribution from the edge-connected neighbors. After the loop over $k \in \{k_j\}$, compute the inverse of the LSQ matrix, $\mathbf{A}_{LSQ}^{-1}$. Next, set $i = 0$ and repeat the same double loop. Within the double loop, increment i by $i = i + 1$, and compute the coefficients:

$$\mathbf{c}_{ji} = \mathbf{A}_{LSQ}^{-1}\mathbf{b}, \tag{A.3}$$

where $\mathbf{c}_{ji} = (c_{ji}^x, c_{ji}^y, c_{ji}^{xx}, c_{ji}^{xy}, c_{ji}^{yy})$, $\mathbf{b} = (\Delta x, \Delta y, \Delta x^2/2, \Delta x\Delta y, \Delta y^2/2)$. Save the first two components, c_{ji}^x and c_{ji}^y, at the node j.

The remaining coefficients, c_{ji}^{xx}, c_{ji}^{xy}, and c_{ji}^{yy}, can be used to compute the second derivatives but not required in the third-order scheme for the hyperbolic advection-diffusion system. Note that the method automatically introduces slight weights based on the connectivities, e.g., 2 if a node is processed twice, which happens for the edge-connected neighbor shared by two adjacent elements. We point out that the only data needed are the coefficients c_{ji}^{x} and c_{ji}^{x} for each node, and other data such as Δx_j and Δy_j computed at the first step as well as the LSQ matrices can be deleted at the end of the process.

Having computed and stored the coefficients, we can perform the gradient reconstruction at every residual evaluation in two steps. We outline the procedure for the variable p, but it is equally valid for any variable.

1. *Construct temporary data:* For each node $j \in \{J\}$, loop over the edge-connected neighbors $k \in \{k_j\}$ and store the edge-difference of the variable for which the gradient is sought:

$$\Delta p_j(k) = p_k - p_j. \tag{A.4}$$

2. *Compute the gradient:* For each node $j \in \{J\}$, initialize the gradient, $\nabla p_j = 0$ and the counter, $i = 0$, and loop over the neighbors of k, $\ell \in \{\ell_k\}$ within the loop over the edge-connected neighbors $k \in \{k_j\}$. Within the double loop, increment i by $i = i + 1$ and accumulate the gradient contribution:

$$\nabla p_j = \nabla p_j + \Delta p \begin{bmatrix} c_{ji}^{x} \\ c_{ji}^{y} \end{bmatrix}, \quad \Delta p = \Delta p_j(k) + \Delta p_k(\ell), \tag{A.5}$$

or $\Delta p = \Delta p_j(k)$ in the case the neighbor coincides with j if the reset has been employed in the calculation of the coefficients in Equation (A.2).

Acknowledgments

This work has been funded by the U.S. Army Research Office under the contract/grant number W911NF-12-1-0154 with Dr. Frederick Ferguson as the program manager. Support by Software CRADLE is also greatly acknowledged.

References

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

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

[3] C. Corre and X. Du. A residual-based scheme for computing compressible flows on unstructured grids. *Comput. Fluids*, 38:1338–1347, 2009.

[4] 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.

[5] M. Ricchiuto, N. Villedieu, R. Abgrall, and H. Deconinck. On uniformly high-order accurate residual distribution schemes for advection-diffusion. *J. Comput. Appl. Math.*, 215:547–556, 2008.

[6] H. Nishikawa. Divergence formulation of source term. *J. Comput. Phys.*, 231:6393–6400, 2012.

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

[8] 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.

[9] C. Cattaneo. A form of heat-conduction equations which eliminates the paradox of instantaneous propagation. *Ct. R. Acad. Sci., Paris*, 247:431–433, 1958.

[10] R. B. Lowrie and J. E. Morel. Methods for hyperbolic systems with stiff relaxation. *Int. J. Numer. Meth. Fluids*, 40:413–423, 2002.

[11] H. Gómez, I. Colominas, F. Navarrina, and M. Casteleiro. A discontinuous Galerkin method for a hyperbolic model for convection-diffusion problems in CFD. *Int. J. Numer. Meth. Eng.*, 71:1342–1364, 2007.

[12] F. Navarrina H. Gómez, I. Colominas and M. Casteleiro. A hyperbolic model for convection-diffusion transport problems in CFD: Numerical analysis and applications. *Rev. R. Acad. Cien. Serie A. Mat.*, 102(2):319–334, 2008.

[13] G. I. Montecinos, Lucas O. Mller, and E. F. Toro. Hyperbolic reformulation of a 1D viscoelastic blood flow model and ader finite volume schemes. *J. Comput. Phys.*, 266:101–123, 2014.

[14] A. Mazaheri and H. Nishikawa. First-order hyperbolic system method for time-dependent advection-diffusion problems. *NASA-TM–2014–218175*, March 2014.

[15] A. Mazaheri and H. Nishikawa. Very efficient high-order hyperbolic schemes for time-dependent advection-diffusion problems: Third-, Fourth, and Sixth-Order. *Comput. Fluids*, 2014. submitted.

[16] 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.

[17] A. Katz and V. Sankaran. Mesh quality effects on the accuracy of CFD solutions on unstructured meshes. *J. Comput. Phys.*, 230:7670–7686, 2011.

[18] A. Katz and V. Sankaran. An efficient correction method to obtain a formally third-order accurate flow solver for node-centered unstructured grids. *J. Sci.Comput.*, 51:375–393, 2012.

[19] B. Diskin and J. L. Thomas. Effects of mesh regularity on accuracy of finite-volume schemes. In *Proc. of 50th AIAA Aerospace Sciences Meeting*, AIAA Paper 2012-0609, Nashville, Tennessee, 2012.

[20] B. B. Pincock and A. Katz. High-order flux correction for viscous flows on arbitrary unstructured grids. In *Proc. of 21st AIAA Computational Fluid Dynamics Conference*, AIAA Paper 2011-2566, San Diego, California, June 2013.

[21] B. Diskin, J. L. Thomas, E. J. Nielsen, H. Nishikawa, and J. A. White. Comparison of node-centered and cell-centered unstructured finite-volume discretizations: Viscous fluxes. *AIAA J.*, 48(7):1326–1338, July 2010.

[22] H. Nishikawa. Higher-order discretization of diffusion terms in residual-distribution methods. In H. Deconinck and M. Ricchiuto, editors, *VKI Lecture Series 2006-01: CFD - High Order Discretization Methods*. von Karman Institute for Fluid Dynamics, Belgium, 2006.

[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] H. Nishikawa. Robust and accurate viscous discretization via upwind scheme - I: Basic principle. *Comput. Fluids*, 49(1):62–86, October 2011.

[25] R. Abgrall, D. De Santis, and M. Ricchiuto. High order preserving residual distribution schemes for advection-diffusion scalar problems on arbitrary grids. INRIA Report, N 8157, December 2012.

[26] P. A. Gnoffo. Updates to multi-dimensional flux reconstruction for hypersonic simulations on tetrahedral grids. AIAA Paper 2010-1271, January 2010.

[27] A. Mazaheri and H. Nishikawa. High-order residual-distribution hyperbolic advection-diffusion schemes: 3rd-, 4th-, and 6th-order. In *Proc. of 7th AIAA Theoretical Fluid Mechanics Conference, AIAA Aviation and Aeronautics Forum and Exposition 2014*, AIAA Paper 2014, Atlanta, GA, June 2014.