ACCEPTED MANUSCRIPT

Journal of Computational Physics
https://doi.org/10.1016/j.jcp.2018.04.058

Reconstructed Discontinuous Galerkin Methods for Linear Advection-Diffusion Equations Based on First-Order Hyperbolic System

Jialin Lou *, Lingquan Li †, and Hong Luo ‡
Department of Mechanical and Aerospace Engineering, North Carolina State University, Raleigh, NC 27695, USA
and
Hiroaki Nishikawa §
National Institute of Aerospace, Hampton, VA 23666, USA

May 4, 2018

Abstract

Newly developed reconstructed Discontinuous Galerkin (rDG) methods are presented for solving linear advection-diffusion equations on hybrid unstructured grids based on a first-order hyperbolic system (FOHS) formulation. Benefiting from both FOHS and rDG methods, the developed hyperbolic rDG methods are reliable, accurate, efficient, and robust, achieving higher orders of accuracy than conventional DG methods for the same number of degrees-of-freedom. Superior accuracy is achieved by reconstruction of higher-order terms in the solution polynomial via gradient variables introduced to form a hyperbolic diffusion system and least-squares/variational reconstruction. Unsteady capability is demonstrated by an L-stable implicit time-integration scheme. A number of advection-diffusion test cases with a wide range of Reynolds numbers, including boundary layer type problems and unsteady cases, are presented to assess accuracy and performance of the newly developed hyperbolic rDG methods. Numerical experiments demonstrate that the hyperbolic rDG methods are able to achieve the designed optimal order of accuracy for both solutions and their derivatives on regular, irregular, and heterogeneous grids, indicating that the developed hyperbolic rDG methods provide an attractive and probably an even superior alternative for solving the linear advection-diffusion equations.

1 Introduction

Nowadays, the discontinuous Galerkin (DG) methods, originally developed for solving the steady neutron transport [1] and unsteady advection problems [2], have shown increasing attention in science and engineering filed for solving conservation laws. They are widely used in computational fluid dynamics (CFD), computational acoustics, and computational magneto-hydrodynamics. By combining the advantages of the finite element (FE) and finite volume (FV), DG methods, one can achieve high order accuracy while retaining the compactness of the stencil. Meanwhile, DG methods are especially suitable for hyperbolic-type systems of equations in terms of solution accuracy [3, 4, 5, 6, 7], treatment of non-conforming meshes [8], and implementation of the hp-adaptivity [9]. However, the DG methods have a number of their own weaknesses. In particular, how to reduce the computing costs for the DG methods, and how to discretize and efficiently solve elliptic/parabolic equations remain two unresolved and challenging issues in the DG methods.

In order to reduce both computational costs and storage requirements of DG methods, a new family of reconstructed DG methods, termed $\mathrm{P}_n\mathrm{P}_m$ schemes, referred to as rDG($\mathrm{P}_n\mathrm{P}_m$) in this paper, was introduced by Dumbser et al. [10, 11, 12]. Here, P_n indicates that a piecewise polynomial of degree of n is used to represent an underlying DG solution, and P_m represents a reconstructed polynomial solution of degree of m ($m \geq n$) that is used to compute the fluxes and source terms. Note that the rDG($\mathrm{P}_n\mathrm{P}_m$) schemes provide a unified

*Ph.D Student
†Ph.D Student
‡Professor
§Associate Research Fellow

ACCEPTED MANUSCRIPT

formulation for both FV and DG methods, and contain both classical FV and standard DG methods as two special cases of rDG(P_nP_m) schemes. Obviously, the construction of an accurate and efficient reconstruction operator is crucial to the success of the rDG(PnPm) schemes. In Dumbser's work [10, 11, 12], a higher order polynomial solution is reconstructed using a L2 projection, requiring it indistinguishable from the underlying DG solutions in the contributing cells in the weak sense. The resultant over-determined system is then solved using a least-squares method that guarantees exact conservation, not only of the cell averages but also of all higher order moments in the reconstructed cell itself, such as slopes and curvatures. However, this conservative least-squares reconstruction approach is computationally expensive, as the L2 projection, i.e., the operation of integration, is required to obtain the resulting over-determined system. Furthermore, the reconstruction might be problematic for a boundary cell, where the number of the face-neighboring cells might be not enough to provide the necessary information to recover a polynomial solution of a desired order. Fortunately, the projection-based reconstruction is not the only way to obtain a polynomial solution of higher order from the underlying discontinuous Galerkin solutions. In a reconstructed DG method using a Taylor basis developed by Luo et al. [13, 14, 15, 16] for the solution of the compressible Euler and Navier–Stokes equations on arbitrary grids, a higher order polynomial solution is reconstructed by use of a strong interpolation, requiring point values and derivatives to be interpolated on the face-neighboring cells. The resulting over-determined linear system of equations is then solved in the least-squares sense. This reconstruction scheme only involves von Neumann neighborhood, and thus is compact, simple, robust, and flexible. Like the projection-based reconstruction, the strong reconstruction scheme guarantees exact conservation, not only of the cell averages but also of their slopes due to a judicious choice of the Taylor basis. The latest hierarchical WENO-based rDG(P_nP_m) schemes [17, 18] are designed not only to reduce the high computing costs associated with DG methods, but also to avoid spurious oscillations in the vicinity of strong discontinuities.

Indeed, DG methods are natural choices for solving hyperbolic systems, such as the compressible Euler equations. However, when it comes to elliptic or parabolic equations, such as the compressible Navier-Stokes equations, the DG formulation is far less certain and advantageous. Approaches made to resolve this issue could be found in the literature [5, 14, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28]. Those methods have introduced in some way the influence of the discontinuities in order to define correct and consistent diffusive fluxes. Unfortunately, all these methods seem to require substantially more computational effort than the classical continuous finite element methods, which are naturally more suited for the discretization of elliptic problems. There is also an approach where a scalar diffusion scheme is derived from a hyperbolic diffusion formulation [29, 30]. It has been extended to higher-order in the context of the residual-distribution method [31], but has not been extended in the DG methods beyond second-order.

Over the last several years, an alternative approach to viscous discretizations, which reformulates the viscous terms as a first-order hyperbolic system (FOHS), was developed by Nishikawa [32, 33, 34, 35, 36], Nishikawa and Roe [37], Nakashima et al. [38], Liu and Nishikawa [39], Mazaheri and Nishikawa [40], Montecinos and Toro [41], Montecinos et al. [42], Toro and Montecinos [43], and Ahn et al. [44]. Note that the approaches in the references [41, 42, 43], present explicit ADER schemes for hyperbolic-diffusion systems with L_r, a free parameter defined as relaxation length, of $\mathcal{O}(h)$ rather than $\mathcal{O}(1)$. In their approach L_r (or T_r, another parameter as relaxation time) needs to depend on the mesh size in order to preserve the designed order of accuracy with explicit time stepping. Thus, their approach is different from the hyperbolic approach we present and discuss here. In the FOHS formulation, by including derivative quantities as additional variables, the equations are first formulated as a first order system (FOS). Then, it is rendered to be hyperbolic, which is the distinguished feature of the FOHS method from other FOS methods, by adding pseudo time derivatives to the first-order system. It thus generates a system of pseudo-time evolution equations for the solution and the derivatives in the partial differential equation (PDE) level, not in the discretization level as in DG methods. The hyperbolic reformulation in the PDE level would allow a dramatic simplification in the discretization as the well-established methods can be directly applied to the viscous terms. Moreover, the presented hyperbolic approach is not targeted at addressing stiff source terms already present in the original formulations. The approach introduces source terms which are not stiff for typical hyper-Re problems of small ν since $T_r = \mathcal{O}(1/\nu)$, in contrast to the hyperbolic approach used in Toro's work [43] where $T_r = \mathcal{O}(\nu)$. The FOHS method is especially attractive in the context of the DG methods since it allows the use of inviscid algorithms for the viscous terms and thus greatly simplifies the discretization of the compressible Navier-Stokes equations. Moreover, the FOHS method yields a numerical scheme that can achieve the same order of accuracy in the solution and its derivatives on irregular grids and high-quality noise-free gradients on such grids. This is a very important feature for unstructured-grid viscous simulations, where target quantities are derivatives, e.g., viscous stresses and heat fluxes.

A challenge in combining the DG method and the FOHS method lies in a very large number of discrete

ACCEPTED MANUSCRIPT

unknowns arising from both methods. For a scalar equation in two dimensions, the FOHS method introduces two derivatives as additional variables, and a DG(P_1) method introduces three degrees of freedom (DoFs) for each variable (solution, and two derivatives), resulting in the total of nine degrees of freedom. In 2015, the fourth author noticed that these degrees of freedom can be significantly reduced by unifying inter-related high-order moments of the derivative variables and extending the idea of Scheme-II [34] to replace high-order moments of a solution polynomial by the derivative variables. He has shown that the total number of degrees of freedom can be reduced from nine to six while the order of polynomial is upgraded to quadratic for the solution variable. The resulting approximation is comparable to a conventional DG(P_2) method. Therefore, if compared with a one-order higher conventional DG method, the FOHS method requires virtually no increase in the degrees of freedom. The method extends systematically to higher order of accuracy: P_k hyperbolic DG method gives comparable accuracy as DG(P_{k+1}) method for the same number of degrees of freedom. Later, the method was presented formally in Ref. [45], focusing on advection-dominated problems. However, the specific method described in Ref. [45] is not yet an attractive approach for practical applications. First, it has one-order-lower accuracy in the diffusion term than a conventional DG method (see Table 3 in Ref. [45]), thus leading to lower order accuracy, for example, in boundary layer calculations. Second, since a direct solver is employed for solving the linear system in the Newton method, convergence acceleration by the elimination of second derivatives, which is one of the advantages of the hyperbolic method, is not achieved. Therefore, this approach is, although more efficient than a straightforward DG discretization of the FOHS, actually less efficient than conventional DG methods, not fully taking advantage of the hyperbolic method. More importantly, the method does not contribute to reducing the cost of the DG method. In this study, we explore the combination of the FOHS method and the rDG method in order to reduce the cost of the DG method towards affordable high-order unstructured-grid methods for practical applications.

Another difficulty would arise when it comes to unsteady problems. Typically, implicit-time stepping schemes are employed in the hyperbolic method, and all previous developments rely on the backward difference formulas (BDF) [37, 46]. The first- and second-order BDF formulas are unconditionally stable (L-stable), and thus suitable for practical applications. However, higher-order (≥ 3) backward-difference formulas are only conditionally stable. It is highly desirable to develop unconditionally-stable high-order hyperbolic schemes for unsteady problems. Also, the high-order BDF method is not self-starting, requiring several lower order BDF methods at the starting stages. Furthermore, the time step would need to be fixed unless some further modification is made, like the variable time step BDF methods [37]. To overcome these difficulties, we consider an explicit first stage, single diagonal coefficient, diagonally implicit Runge-Kutta time integration scheme (ESDIRK) [47] and demonstrate the unsteady capability of the developed hyperbolic schemes. Compared with BDF methods, implicit Runge-Kutta (IRK) methods are A-stable and L-stable for arbitrary order in time. Also, variable time step sizes can be easily applied. Moreover, ESDIRK schemes are self-starting, i.e., one does not need to set up different temporal schemes at the beginning. Although ESIRK schemes would be more computationally expensive than the BDF counterpart for the same time step size, the cost can be reduced by taking a larger time step without encountering instability and thus to maintain the design order of accuracy.

The objective of the effort discussed in the present work is to develop high-order hyperbolic rDG methods for solving linear advection-diffusion equations based on the FOHS formulation, termed hyperbolic rDG methods in this paper. An efficient and consistent construction is presented, which provides higher-order accuracy in both the solution and the derivatives than conventional DG methods for the same number of degrees of freedom. Also, different gradient reconstruction methods, which are used to increase the polynomial order in the rDG methods, are explored, including hybrid least-squares (LS) [48] and variational reconstruction (VR) [49]. The hyperbolic rDG method is a general framework, including finite-volume methods and the schemes in Ref. [45] as special cases. In this study, we consider the upwind hyperbolic-diffusion flux to exploit the maximum potential of the hyperbolic methods. A number of linear advection-diffusion problems are presented, including boundary-layer type problems and pure diffusion problems which are not reported in Ref. [45], indicating the hyperbolic rDG method is a cost-effective high-order scheme, and has the potential to ultimately be applied to the incompressible and compressible Navier-Stokes equations on fully irregular, adaptive, anisotropic, unstructured grids.

The outline of the rest of this paper is organized as follows. A FOHS formulation for advection-diffusion equations is described in Section 2. The rDG methods for solving the hyperbolic diffusion equations are presented in Section 3. Extensive numerical experiments are reported in Section 4. Concluding remarks and a plan of future work are given in Section 5.

ACCEPTED MANUSCRIPT

2 FOHS formulation for Advection-Diffusion Equations

Consider the following model linear advection-diffusion equation in 2D.

$$\frac{\partial \varphi}{\partial t} + a\frac{\partial \varphi}{\partial x} + b\frac{\partial \varphi}{\partial y} = \nu\left(\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2}\right) + f(x,y), \tag{1}$$

where φ denotes a scalar function that can be referred to as the primary solution variable (or the velocity potential), (a, b) is a constant advection vector, ν is a positive diffusion coefficient, and $f(x, y)$ is the source term. In order to reformulate this equation into a first-order hyperbolic advection diffusion system, derivatives of the primary solution variable φ would be needed as additional variables. Therefore, the gradient variables (or the velocity vector) $\mathbf{v}$ is defined as

$$\mathbf{v} = \nabla\varphi = [\mathrm{v}_x, \quad \mathrm{v}_y]^T, \tag{2}$$

where v_x and v_y are the components of the gradient variables.

By adding pseudo time derivatives with respect to all variables, the following first-order hyperbolic system for this advection-diffusion equation can be formulated.

$$\begin{cases} \dfrac{\partial \varphi}{\partial \tau} + \dfrac{\partial \varphi}{\partial t} + a\dfrac{\partial \varphi}{\partial x} + b\dfrac{\partial \varphi}{\partial y} = \nu\left(\dfrac{\partial \mathrm{v}_x}{\partial x} + \dfrac{\partial \mathrm{v}_y}{\partial y}\right) + f(x,y), \\ \dfrac{\partial \mathrm{v}_x}{\partial \tau} = \dfrac{1}{T_r}\left(\dfrac{\partial \varphi}{\partial x} - \mathrm{v}_x\right), \\ \dfrac{\partial \mathrm{v}_y}{\partial \tau} = \dfrac{1}{T_r}\left(\dfrac{\partial \varphi}{\partial y} - \mathrm{v}_y\right), \end{cases} \tag{3}$$

where t and τ are understood as the physical time and the pseudo time respectively. Clearly, the gradient variables would relax to the solution derivatives in the steady state of pseudo time, leading to the consistent gradients at any instant of physical time. Here, T_r is a free parameter, named as relaxation time. Note that the system is equivalent to the original advection-diffusion equation in the steady state for any nonzero T_r, but T_r needs to be positive for the system to be hyperbolic. For steady problems, the system without the physical time derivative can be solved by marching in the pseudo time to yield a steady solution to the original equation. As for unsteady problems, the physical time derivative is discretized by the ESDIRK (Explicit first stage, Single Diagonal coefficient, diagonally Implicit Runge-Kutta) method, while the pseudo time may be discretized by BDF1 to march in τ towards a pseudo steady state. The details would be discussed later.

At this point, it is noticed that the FOHS formulation has introduced two extra variables, v_x and v_y, and a numerical scheme would involve two additional equations compared with a scheme applied to the original scalar equation, i.e., Eq.(1). Seemingly, the FOHS formulation requires more computational efforts than a conventional scheme. In FV methods, the number of discrete equations would increase by the number of derivatives for each variable, e.g., fifteen extra equations in the case of the three-dimensional Navier-Stokes equations [36, 38]. Note that this is equivalent to P_1 DG methods, which introduce fifteen derivatives in the discretization level: both Hyperbolic-Navier-Stokes (HNS) FV and P_1 DG methods require twenty discrete unknowns. However, the HNS FV method is not necessarily more expensive than a conventional FV method because the resulting schemes achieve $\mathcal{O}(1/h)$ speed-up in iterative convergence by the elimination of a typical $\mathcal{O}(1/h^2)$ diffusion stiffness, and also yield one-order higher order accuracy in the solution gradients and and in the advective/inviscid approximation as demonstrated in Refs. [34, 35, 36, 50, 51]. In DG methods, these extra variables and their moments can be used to build/replace the high-order moments in the primary solution polynomial; thus the order of polynomial of the primary solution variable is always P_0. As a result, it leads to a scheme achieving a comparable level of accuracy as a conventional DG scheme for the same number of degrees of freedom as mentioned in Introduction. Nevertheless, the resulting scheme has one-order-lower order of accuracy in the diffusion term, and most importantly does not contribute to reducing the cost of the DG methods [45]. The objective of the present study is to demonstrate that the FOHS method combined with the rDG method can break this barrier and generate schemes truly more efficient than conventional DG methods.

The FOHS can be written in the vector form as

$$\frac{\partial \mathbf{U}}{\partial \tau} + \mathbf{T}\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_x}{\partial x} + \frac{\partial \mathbf{F}_y}{\partial y} = \mathbf{S}, \tag{4}$$

ACCEPTED MANUSCRIPT

where

$$\mathbf{U} = \begin{pmatrix} \varphi \\ \mathrm{v}_x \\ \mathrm{v}_y \end{pmatrix}, \quad \mathbf{T} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{F}_x = \begin{pmatrix} a\varphi - \nu \mathrm{v}_x \\ -\varphi/T_r \\ 0 \end{pmatrix}, \quad \mathbf{F}_y = \begin{pmatrix} b\varphi - \nu \mathrm{v}_y \\ 0 \\ -\varphi/T_r \end{pmatrix}, \quad \mathbf{S} = \begin{pmatrix} f(x,y) \\ -\mathrm{v}_x/T_r \\ -\mathrm{v}_y/T_r \end{pmatrix}. \tag{5}$$

In this paper, we consider the advection term and the diffusive term separately.

$$\mathbf{F}_x = \mathbf{F}_x^a + \mathbf{F}_x^d = \begin{pmatrix} a\varphi \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -\nu \mathrm{v}_x \\ -\varphi/T_r \\ 0 \end{pmatrix}, \quad \mathbf{F}_y = \mathbf{F}_y^a + \mathbf{F}_y^d = \begin{pmatrix} b\varphi \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -\nu \mathrm{v}_y \\ 0 \\ -\varphi/T_r \end{pmatrix}. \tag{6}$$

Consider the Jacobian of the flux projected along $\mathbf{n} = (n_x, \quad n_y)$,

$$\mathbf{A}_n = \frac{\partial \mathbf{F}_x}{\partial \mathbf{U}} n_x + \frac{\partial \mathbf{F}_y}{\partial \mathbf{U}} n_y = \mathbf{A}_n^a + \mathbf{A}_n^d, \tag{7}$$

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

$$\mathbf{A}_n^a = \frac{\partial \mathbf{F}_x^a}{\partial \mathbf{U}} n_x + \frac{\partial \mathbf{F}_y^a}{\partial \mathbf{U}} n_y = \begin{pmatrix} a_n & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \mathbf{A}_n^d = \frac{\partial \mathbf{F}_x^d}{\partial \mathbf{U}} n_x + \frac{\partial \mathbf{F}_y^d}{\partial \mathbf{U}} n_y = \begin{pmatrix} 0 & -\nu n_x & -\nu n_y \\ -n_x/T_r & 0 & 0 \\ -n_y/T_r & 0 & 0 \end{pmatrix}, \tag{8}$$

and

$$a_n = a n_x + b n_y. \tag{9}$$

The only non-zero eigenvalue of advective Jacobian is a_n, while the diffusive Jacobian has the following eigenvalues

$$\lambda_1 = \sqrt{\frac{\nu}{T_r}}, \quad \lambda_2 = -\sqrt{\frac{\nu}{T_r}}, \quad \lambda_3 = 0. \tag{10}$$

The first two nonzero eigenvalues indicate that the system describes a wave propagating isotropically if we only consider the diffusive part. The third eigenvalue corresponds to the inconsistency damping mode [32]. The relaxation time T_r does not affect the steady solution, and thus can be defined solely for the purpose of accelerating the convergence to the steady state. For simplicity, T_r is defined as

$$T_r = \frac{L_r^2}{\nu}, \quad L_r = \frac{1}{\max(\mathrm{Re}, 2\pi)}, \quad \mathrm{Re} = \frac{\sqrt{a^2+b^2}}{\nu}. \tag{11}$$

Note that we include Reynolds (or Péclet) number information in the relaxation length scale, so that the developed method could deliver the designed order of accuracy with fast convergence when it comes to narrow boundary layer type problem. See Ref. [52] for details.

This study considers linear equations only. Extensions to nonlinear diffusion equations require additional considerations in the construction of numerical schemes, and will be addressed in a subsequent paper. Diffusion equations with a tensor coefficient have already been discussed in our previous paper [53], and will not be discussed in this paper.

3 Hyperbolic Reconstructed Discontinuous Galerkin Methods

3.1 Reconstructed Discontinuous Galerkin Methods

The FOHS of equations, i.e. Eq. (4), can be discretized using a discontinuous Galerkin finite element formulation. We assume that the domain Ω is subdivided into a collection of non-overlapping arbitrary elements (or cells) $\{\Omega_i\}$, and then introduce the following broken Sobolev space V_h^n

$$V_h^n = \left\{ v_h \in \left[L^2(\Omega)\right]^k : v_h|_{\Omega_i} \in \left[V_n^k\right] \forall \Omega_i \in \Omega \right\}, \tag{12}$$

which consists of discontinuous vector polynomial functions of degree n, and where k is the dimension of the unknown vector and V_n is the space of all polynomials of degree $\leq n$. To formulate the discontinuous Galerkin method, we introduce the following weak formulation, which is obtained by multiplying Eq. (4) by a test

ACCEPTED MANUSCRIPT

function $\mathbf{W}_h$, integrating over an element Ω_e, and then performing an integration by parts: find $\mathbf{U}_h \in V_h^p$ such as

$$\frac{\partial}{\partial \tau}\int_{\Omega_i} \mathbf{W}_h \mathbf{U}_h d\Omega + \frac{\partial}{\partial t}\int_{\Omega_i} \mathbf{W}_h \mathbf{T} \mathbf{U}_h d\Omega + \int_{\Gamma_i} \mathbf{W}_h \mathbf{F}_k \mathbf{n}_k d\Gamma - \int_{\Omega_i} \frac{\partial \mathbf{W}_h}{\partial x_k} \mathbf{F}_k d\Omega = \int_{\Omega_i} \mathbf{W}_h \mathbf{S} d\Omega, \quad \forall \mathbf{W}_h \in V_h^n, \tag{13}$$

where $\mathbf{U}_h$ and $\mathbf{W}_h$ are represented by piecewise polynomial functions of degrees p, which are discontinuous between the cell interfaces, and $\mathbf{n}_k$ the unit outward normal vector to the Γ_i: the boundary of Ω_i. The standard DG solution $\mathbf{U}_h$ within the element Ω_i can be expressed as

$$\mathbf{U}_h(x, y, t, \tau) = \mathbf{C}(x, y)\mathbf{V}(t, \tau), \tag{14}$$

where $\mathbf{C}$ is a basis matrix, and $\mathbf{V}$ is a vector of unknown polynomial coefficients. A further discussion about $\mathbf{C}$ and $\mathbf{V}$ will be given later in this section.

If we set the test function $\mathbf{W}_h$ as the transpose of the basis matrix $\mathbf{C}$, then the following equivalent system would be arrived.

$$\frac{\partial}{\partial \tau}\int_{\Omega_i} \mathbf{C}^T \mathbf{C} \mathbf{V} d\Omega + \frac{\partial}{\partial t}\int_{\Omega_i} \mathbf{C}^T \mathbf{T} \mathbf{C} \mathbf{V} d\Omega + \int_{\Gamma_i} \mathbf{C}^T \mathbf{F}_k \mathbf{n}_k d\Gamma - \int_{\Omega_i} \frac{\partial \mathbf{C}^T}{\partial x_k} \mathbf{F}_k d\Omega = \int_{\Omega_i} \mathbf{C}^T \mathbf{S} d\Omega. \tag{15}$$

Since the numerical solution $\mathbf{U}_h$ is discontinuous between element interfaces, the interface fluxes are not uniquely defined. This scheme is called the discontinuous Galerkin method of degree n, or in short notation DG(P_n) method. By simply increasing the degree n of the polynomials, the DG methods of corresponding higher order are obtained.

Compared with reconstructed FV methods, the DG methods would require more degrees of freedom, additional domain integration, and more Gauss quadrature points for the boundary integration, which leads to more computational costs and storage requirements. Inspired by the reconstructed DG methods from Dumbser et al. in the frame of $\mathrm{P}_n\mathrm{P}_m$ scheme [10, 11, 12], termed rDG($\mathrm{P}_n\mathrm{P}_m$) in this paper, least-squares based and variational reconstruction based rDG methods are designed to achieve high order of accuracy while reducing the computational cost. In fact, a unified formulation would be provided by rDG method for both FV and DG methods. The standard FV and DG methods would be nothing but special cases in rDG framework, thus allowing for a direct efficiency comparison. For rDG($\mathrm{P}_n\mathrm{P}_m$) method with $m > n$, a higher-order reconstructed numerical solution is constructed over an element Ω_i:

$$\mathbf{U}_h^R(x, y, t, \tau) = \mathbf{C}^R(x, y)\mathbf{V}^R(t, \tau), \tag{16}$$

where the superscript R indicates reconstructed polynomials, and higher-order derivatives (higher than n-th and up to m-th) are reconstructed from the underlying P_n polynomial. This higher-order numerical solution $\mathbf{U}_h^R$ is used for flux and source term computations in order to raise the order of accuracy. There are three approaches to the reconstruction. One is a least-squares reconstruction method, and another is a variational reconstruction method. The variational reconstruction generates a globally coupled system of equations for gradients by minimizing jumps in the solution and derivatives at element interfaces [49]. The resulting linear system is iteratively solved along with the solution iteration, and therefore the cost is comparable to a least-squares reconstruction. See Ref. [49] for details. The method based on the least-squares/variational reconstruction is expressed by rDG($\mathrm{P}_n\mathrm{P}_m$). The third approach, which is unique in the FOHS formulation considered here, is to directly use the gradient variables and their moments to evaluate the higher-order derivatives in the primary solution polynomial. Or equivalently, this approach can be thought of as defining the primary solution as P_m, and use the higher-order moments to represent the gradient variables in the FOHS formulation. This is the key idea to effectively reducing the number of discrete unknowns despite the increase in the variables in the FOHS formulation. The method based on this approach is expressed by DG($\mathrm{P}_0\mathrm{P}_m$). Naturally, rDG($\mathrm{P}_n\mathrm{P}_m$) and DG($\mathrm{P}_0\mathrm{P}_m$) can be combined to generate efficient schemes as we will discuss later.

By moving the third and fourth terms to the right-hand-side (r.h.s.) in Eq. (15), we will arrive at

$$\mathbf{M}_\tau \frac{\partial \mathbf{V}}{\partial \tau} + \mathbf{M}_t \frac{\partial \mathbf{V}}{\partial t} = \mathbf{R}(\mathbf{U}_h^R), \tag{17}$$

where $\mathbf{M}_\tau$ and $\mathbf{M}_t$ are the mass matrices defined as,

$$\mathbf{M}_\tau = \int_{\Omega_i} \mathbf{C}^T \mathbf{C} d\Omega, \tag{18}$$

ACCEPTED MANUSCRIPT

$$\mathbf{M}_t = \int_{\Omega_i} \mathbf{C}^T \mathbf{T} \mathbf{C} d\Omega, \tag{19}$$

and $\mathbf{R}$ is the residual vector, defined as

$$\mathbf{R} = \int_{\Omega_i} \frac{\partial \mathbf{C}^T}{\partial x_k} \mathbf{F}_k(\mathbf{U}_h^R) + \mathbf{C}^T \mathbf{S}(\mathbf{U}_h^R) d\Omega - \int_{\Gamma_i} \mathbf{C}^T \mathbf{F}_k(\mathbf{U}_h^R) \mathbf{n}_k d\Gamma. \tag{20}$$

3.2 Hyperbolic Reconstructed Discontinuous Galerkin Methods

Based on different rDG methods, some effective discretization hyperbolic rDG methods will be presented to deal with the derived FOHS. The format $\mathbf{A} + \mathbf{B}$ is used to indicate the discretization method for the system, where $\mathbf{A}$ refers to the discretization method for φ and $\mathbf{B}$ refers to the discretization method for its derivatives. Different choices and combinations for $\mathbf{A}$ and $\mathbf{B}$ are compared in the authors' previous work [53]. To minimize the memory and storage cost of the developed methods, one can apply DG(P_n) or rDG($\mathrm{P}_n\mathrm{P}_m$) methods only on the gradient variables. On the other hand, a higher order of polynomial for φ can be constructed with only one degree of freedom. Therefore, in this paper, we focus on DG($\mathrm{P}_0\mathrm{P}_{n+1}$)+DG($\mathrm{P}_n$) and DG($\mathrm{P}_0\mathrm{P}_{m+1}$)+rDG($\mathrm{P}_n\mathrm{P}_m$) methods.

In the implementation of the DG methods in this paper, modal-basis DG methods are adopted. The numerical polynomial solutions are represented using a Taylor series expansion at the cell center and normalized to improve the conditioning of the system matrix: e.g., in two dimensions,

$$\begin{aligned} \varphi_h = \overline{\varphi} + \varphi_x^c \Delta x B_2 + \varphi_y^c \Delta y B_3 + \varphi_{xx}^c \Delta x^2 B_4 + \varphi_{yy}^c \Delta y^2 B_5 + \varphi_{xy}^c \Delta x \Delta y B_6 \\ + \varphi_{xxx}^c \Delta x^3 B_7 + \varphi_{yyy}^c \Delta y^3 B_8 + \varphi_{xxy}^c \Delta x^2 \Delta y B_9 + \varphi_{xyy}^c \Delta x \Delta y^2 B_{10} + \cdots, \end{aligned} \tag{21}$$

where the $\overline{\varphi}$ represents the cell-averaged quantity of φ, and the superscript c stands for the central values. The basis functions are given as follows

$$\begin{gathered} B_1 = 1, B_2 = \frac{x - x_c}{\Delta x}, B_3 = \frac{y - y_c}{\Delta y}, \\ B_4 = \frac{1}{2}\left(B_2^2 - \frac{1}{\Omega_i}\int_{\Omega_i} B_2^2 d\Omega\right), B_5 = \frac{1}{2}\left(B_3^2 - \frac{1}{\Omega_i}\int_{\Omega_i} B_3^2 d\Omega\right), B_6 = B_2 B_3 - \frac{1}{\Omega_i}\int_{\Omega_i} B_2 B_3 d\Omega, \\ B_7 = \frac{1}{6}\left(B_2^3 - \frac{1}{\Omega_i}\int_{\Omega_i} B_2^3 d\Omega\right), B_8 = \frac{1}{6}\left(B_3^3 - \frac{1}{\Omega_i}\int_{\Omega_i} B_3^3 d\Omega\right), \\ B_9 = \frac{1}{2}\left(B_2^2 B_3 - \frac{1}{\Omega_i}\int_{\Omega_i} B_2^2 B_3 d\Omega\right), B_{10} = \frac{1}{2}\left(B_2 B_3^2 - \frac{1}{\Omega_i}\int_{\Omega_i} B_2 B_3^2 d\Omega\right). \end{gathered} \tag{22}$$

Here, we have

$$\Delta x = 0.5(x_{\max} - x_{\min}), \quad \Delta y = 0.5(y_{\max} - y_{\min}), \tag{23}$$

where $x_{\max}$, $x_{\min}$, $y_{\max}$, and $y_{\min}$ are used to represent the maximum and minimum coordinates values of the vertexes of the cell.

As we mentioned in the previous work [53], one can choose the unknown vector $\mathbf{V}$ to make the resultant scheme have the same number of degrees of freedom as conventional DG methods for a comparable level of accuracy. As a matter of fact, if a Petrov-Galerkin formulation with a simplified basis function matrix is implemented, one would end up with the same hyperbolic rDG methods we used for diffusion equation [53]. On the other hand, if a consistent Galerkin formulation is used, one can make all variables coupled and thus to have better stability properties for advection-diffusion problems. This is the approach taken in this study. Below, some examples of the unknown vector $\mathbf{V}$ and the basis matrix $\mathbf{C}$ under Galerkin formulation are shown for better illustration. The first three examples are the hyperbolic DG schemes, and the last two are the hyperbolic rDG schemes, which are more efficient.

- DG($\mathrm{P}_0\mathrm{P}_1$)+DG(P_0)

$$\mathbf{V} = \begin{pmatrix} \overline{\varphi} \\ \overline{\varphi_x}\Delta x \\ \overline{\varphi_y}\Delta y \end{pmatrix}, \tag{24}$$

ACCEPTED MANUSCRIPT

$$\mathbf{C} = \begin{pmatrix} B_1 & B_2 & B_3 \\ 0 & B_1\Delta x^{-1} & 0 \\ 0 & 0 & B_1\Delta y^{-1} \end{pmatrix}. \tag{25}$$

As we can see here, the basis matrix $\mathbf{C}$ has included the connection between φ and its derivatives, leading to a coupled system. Note that the derivatives of φ are expressed by the gradient variables in the FOHS, whereas conventional P_1 DG methods determine them as solutions to discrete equations derived by the weak formulation. This particular construction corresponds to a less-efficient variant in Ref. [45]: it has the same number of degrees of freedom as a conventional second-order P_1 DG method, but is only first-order accurate for diffusion.

- DG(P_0P_2)+DG(P_1)

$$\mathbf{V} = [\overline{\varphi}, \overline{\varphi_x}\Delta x, \overline{\varphi_y}\Delta y, \varphi^c_{xx}\Delta x^2, \varphi^c_{yy}\Delta y^2, \varphi^c_{xy}\Delta x\Delta y]^T, \tag{26}$$

$$\mathbf{C} = \begin{pmatrix} B_1 & B_2 & B_3 & B_4 & B_5 & B_6 \\ 0 & B_1\Delta x^{-1} & 0 & B_2\Delta x^{-1} & 0 & B_3\Delta x^{-1} \\ 0 & 0 & B_1\Delta y^{-1} & 0 & B_3\Delta y^{-1} & B_2\Delta y^{-1} \end{pmatrix}. \tag{27}$$

Compared with authors' previous work [53], the degrees of freedom for DG(P_0P_2)+DG(P_1) has been reduced from 7 to 6 by replacing the redundant cross term with a unified unknown. Hence, this method has the same number of degrees of freedom as P_2 conventional DG methods. This scheme, again, corresponds to a less-efficient variant in Ref. [45]: it has the same number of degrees of freedom as a conventional third-order P_2 DG method, but is only second-order accurate for diffusion.

- DG(P_0P_3)+DG(P_2)

$$\mathbf{V} = [\overline{\varphi}, \overline{\varphi_x}\Delta x, \overline{\varphi_y}\Delta y, \varphi^c_{xx}\Delta x^2, \varphi^c_{yy}\Delta y^2, \varphi^c_{xy}\Delta x\Delta y, \varphi^c_{xxx}\Delta x^3, \varphi^c_{yyy}\Delta y^3, \varphi^c_{xxy}\Delta x^2\Delta y, \varphi^c_{xyy}\Delta x\Delta y^2]^T, \tag{28}$$

$$\mathbf{C} = \begin{pmatrix} B_1 & 0 & 0 \\ B_2 & B_1\Delta x^{-1} & 0 \\ B_3 & 0 & B_1\Delta y^{-1} \\ B_4 & B_2\Delta x^{-1} & 0 \\ B_5 & 0 & B_3\Delta y^{-1} \\ B_6 & B_3\Delta x^{-1} & B_2\Delta y^{-1} \\ B_7 + B_2B_4^c & B_4\Delta x^{-1} & 0 \\ B_8 + B_3B_5^c & 0 & B_5\Delta y^{-1} \\ B_9 + B_2B_6^c + B_3B_4^c & B_6\Delta x^{-1} & B_4\Delta y^{-1} \\ B_{10} + B_2B_5^c + B_3B_6^c & B_5\Delta x^{-1} & B_6\Delta y^{-1} \end{pmatrix}^T. \tag{29}$$

The complexity in the basis matrix is due to the fact that the average values and cell centers values are not equal. This can be derived using Taylor expansion. One can find a similar procedure in Ref. [45]. Note that it has the same number of degrees of freedom as a conventional fourth-order P_3 DG method, but is only third-order accurate for diffusion.

- DG(P_0P_2)+rDG(P_0P_1) - Hyperbolic rDG

$$\mathbf{V} = \begin{pmatrix} \overline{\varphi} \\ \overline{\varphi_x}\Delta x \\ \overline{\varphi_y}\Delta y \end{pmatrix}, \tag{30}$$

$$\mathbf{C} = \begin{pmatrix} B_1 & B_2 & B_3 \\ 0 & B_1\Delta x^{-1} & 0 \\ 0 & 0 & B_1\Delta y^{-1} \end{pmatrix}. \tag{31}$$

$$\mathbf{V}^R = [\overline{\varphi}, \overline{\varphi_x}\Delta x, \overline{\varphi_y}\Delta y, \varphi^{c,R}_{xx}\Delta x^2, \varphi^{c,R}_{yy}\Delta y^2, \varphi^{c,R}_{xy}\Delta x\Delta y]^T, \tag{32}$$

$$\mathbf{C}^R = \begin{pmatrix} B_1 & B_2 & B_3 & B_4 & B_5 & B_6 \\ 0 & B_1\Delta x^{-1} & 0 & B_2\Delta x^{-1} & 0 & B_3\Delta x^{-1} \\ 0 & 0 & B_1\Delta y^{-1} & 0 & B_3\Delta y^{-1} & B_2\Delta y^{-1} \end{pmatrix}. \tag{33}$$

This is a more efficient hyperbolic rDG method. It is based on the same $\mathbf{C}$ and $\mathbf{V}$ as DG(P_0P_1)+DG(P_0), and thus has the same mass matrices. However, a higher order polynomial $\mathbf{U}_h^R$ is used for computing the flux and

ACCEPTED MANUSCRIPT

source term, and thus it yields a more accurate solution. The higher order terms, i.e., $\varphi_{xx}^{c,R}\Delta x^2, \varphi_{yy}^{c,R}\Delta y^2, \varphi_{xy}^{c,R}\Delta x\Delta y$ are computed from $\overline{\varphi_x}$ and $\overline{\varphi_y}$ by using reconstruction schemes. For example, in this study, a hybrid least-squares scheme (LS) [48] and a variational reconstruction scheme (VR) [49] have been implemented to obtain higher moments. This scheme is compared with a conventional P_1 DG method in terms of the degrees of freedom, and achieves second-order accuracy for both the solution $\overline{\varphi}$, and the derivatives $\overline{\varphi_x}$ and $\overline{\varphi_y}$, and third-order accuracy for $\overline{\varphi}$ in the advection limit or on regular (or mildly-distorted irregular) grids. This is the class of hyperbolic-rDG schemes that reduces the cost of the DG methods; it is one of the main target schemes in the present work.

- DG(P_0P_3)+rDG(P_0P_2) - Hyperbolic rDG

$$\mathbf{V} = \begin{pmatrix} \overline{\varphi} \\ \overline{\varphi_x}\Delta x \\ \overline{\varphi_y}\Delta y \end{pmatrix}, \tag{34}$$

$$\mathbf{C} = \begin{pmatrix} B_1 & B_2 & B_3 \\ 0 & B_1\Delta x^{-1} & 0 \\ 0 & 0 & B_1\Delta y^{-1} \end{pmatrix}. \tag{35}$$

$$\mathbf{V}^R = [\overline{\varphi}, \overline{\varphi_x}\Delta x, \overline{\varphi_y}\Delta y, \varphi_{xx}^{c,R}\Delta x^2, \varphi_{yy}^{c,R}\Delta y^2, \varphi_{xy}^{c,R}\Delta x\Delta y, \varphi_{xxx}^{c,R}\Delta x^3, \varphi_{yyy}^{c,R}\Delta y^3, \varphi_{xxy}^{c,R}\Delta x^2\Delta y, \varphi_{xyy}^{c,R}\Delta x\Delta y^2]^T, \tag{36}$$

$$\mathbf{C}^R = \begin{pmatrix} B_1 & 0 & 0 \\ B_2 & B_1\Delta x^{-1} & 0 \\ B_3 & 0 & B_1\Delta y^{-1} \\ B_4 & B_2\Delta x^{-1} & 0 \\ B_5 & 0 & B_3\Delta y^{-1} \\ B_6 & B_3\Delta x^{-1} & B_2\Delta y^{-1} \\ B_7 + B_2B_4^c & B_4\Delta x^{-1} & 0 \\ B_8 + B_3B_5^c & 0 & B_5\Delta y^{-1} \\ B_9 + B_2B_6^c + B_3B_4^c & B_6\Delta x^{-1} & B_4\Delta y^{-1} \\ B_{10} + B_2B_5^c + B_3B_6^c & B_5\Delta x^{-1} & B_6\Delta y^{-1} \end{pmatrix}^T. \tag{37}$$

This is an even more efficient hyperbolic-rDG scheme. It is still based on the same $\mathbf{C}$ and $\mathbf{V}$ with three degrees of freedom, but achieves third-order accuracy and fourth-order in the advection limit. The higher order terms, i.e., $\varphi_{xx}^{c,R}\Delta x^2, \varphi_{yy}^{c,R}\Delta y^2, \varphi_{xy}^{c,R}\Delta x\Delta y, \varphi_{xxx}^{c,R}, \varphi_{yyy}^{c,R}, \varphi_{xxy}^{c,R}, \varphi_{xyy}^{c,R}$ are computed from $\overline{\varphi_x}$ and $\overline{\varphi_y}$ by using reconstruction schemes. In this case, a quadratic reconstruction is required, which can be more efficiently performed by the variational method than the least-squares method. The DG(P_0P_3)+rDG(P_0P_2) scheme is much more efficient than a conventional P_2 DG method that requires six degrees of freedom to achieve third-order accuracy in the solution and only second-order accuracy in the derivatives. This scheme, therefore, greatly reduces the cost of the DG method, and as will be shown later, it is demonstrated to be a robust and accurate scheme.

Table 1 summarizes the types of schemes considered in this study. Expected order of accuracy is indicated separately for the advection term, the diffusion term, and the solution gradient. It shows clearly that the hyperbolic DG schemes of the class DG(P_0P_{k+1})+DG(P_k) [45] is one-order-lower accurate in the diffusion term than the DG(P_k) scheme of the same DoFs. On the other hand, the hyperbolic rDG schemes of the class DG(P_0P_{k+1})+rDG(P_k) are more accurate than the DG scheme of the same DoFs. Moreover, the hyperbolic rDG scheme is even more accurate than a conventional DG with a larger number of DoFs: e.g., compare DG(P_0P_3)+rDG(P_0P_2) with DG(P_2). As will be demonstrated later, the hyperbolic rDG schemes achieve super convergence in some cases, and thus can be much more efficient than expected.

ACCEPTED MANUSCRIPT

Table 1: Comparison of DG schemes and hyperbolic-DG/rDG schemes for expected order of accuracy.

Scheme	DoFs	Advection	Diffusion	Gradient
DG(P_1)	3	2nd	2nd	1st
DG(P_0P_1)+DG(P_0)	3	2nd	1st	1st
DG(P_0P_2)+rDG(P_0P_1)	3	3rd	2nd	2nd
DG(P_0P_3)+rDG(P_0P_2)	3	4th	3rd	3rd
DG(P_2)	6	3rd	3rd	2nd
DG(P_0P_2)+DG(P_1)	6	3rd	2nd	2nd
DG(P_0P_3)+rDG(P_1P_2)	6	4th	3rd	3rd
DG(P_3)	10	4th	4th	3rd
DG(P_0P_3)+DG(P_2)	10	4th	3rd	3rd

3.3 Numerical Flux

Classically, the conventional DG would need two numerical flux schemes to solve the advection-diffusion equation. While DG methods are naturally developed for hyperbolic equations, the diffusive flux are not that straightforward or efficient. However, with the FOHS, the rDG method can use well-established methods for hyperbolic systems. In this paper, following Refs. [35], the simplest upwind method is applied for the numerical flux across the interface:

$$\mathbf{F}_{ij} = \frac{1}{2}(\mathbf{F}_L + \mathbf{F}_R) \cdot \mathbf{n}_{ij} - \frac{1}{2}(|\mathbf{A}_n^a| + |\mathbf{A}_n^d|)(\mathbf{U}_R - \mathbf{U}_L). \tag{38}$$

where the subscripts L and R indicate the face values of the polynomials in the cell i and j (i.e., the interior and exterior values), respectively, $\mathbf{n}_{ij} = (n_x, n_y)$ is the unit directed area vector, and $|\mathbf{A}^a|$ and $|\mathbf{A}^d|$ would be

$$|\mathbf{A}_n^a| = \begin{pmatrix} |a_n| & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad |\mathbf{A}_n^d| = \frac{\nu}{L_r} \begin{pmatrix} 1 & 0 & 0 \\ 0 & n_x^2 & n_x n_y \\ 0 & n_x n_y & n_y^2 \end{pmatrix}. \tag{39}$$

Note that, the absolute Jacobian is constructed independently for both advection and diffusion terms. Here we are not assuming $|\mathbf{A}_n| = |\mathbf{A}_n^a| + |\mathbf{A}_n^d|$, which is not true. What we did here is an approximation, which would allow us to avoid the analysis for the eigen-structure for the whole system. As a matter of fact, for this simple advection-diffusion equation, the eigen-structure for the whole system is still analyzable with some extra effort [33]. However, when it comes to complex conservation laws, such as the Navier-Stokes equations, one can only rely on the approximation approach at present. The simplified approach used here has been successfully demonstrated for the Navier-Stokes system [36, 38, 39, 46, 54].

3.4 Boundary Condition

Boundary conditions are enforced weakly through the numerical flux in a similar manner as in the previous work [37, 53]. For all test problems, the Dirichlet condition is considered, and therefore only the solution φ is given on boundaries. At a boundary face, $\mathbf{n}_{ij}$ is taken to be outward, and thus $\mathbf{U}_R$ is considered as a boundary state. The boundary condition is incorporated into the boundary state as

$$\mathbf{U}_R = (\varphi_b, \mathrm{v}_n n_x + \partial_s \varphi_b t_x, \mathrm{v}_n n_y + \partial_s \varphi_b t_y), \tag{40}$$

where φ_b is the value given as a boundary condition, and $\partial_s \varphi_b$ is the tangential derivative that can be obtained from the given boundary condition, v_n is the face-normal projection of $(\mathrm{v}_x, \mathrm{v}_y)$ evaluated at the left (interior) state $\mathbf{U}_L$, $\mathbf{n}_{ij} = (n_x, n_y)$, and (t_x, t_y) denotes a unit tangent vector of the boundary face. Note that s is taken to be positive in the counterclockwise direction along a boundary, and the tangent vector is also taken in the same direction. In the case of a unit square domain, the boundary state becomes

$$\mathbf{U}_R = (\varphi_b, \mathrm{v}_x, \partial_y \varphi_b), \tag{41}$$

ACCEPTED MANUSCRIPT

at the left and right boundaries, and

$$\mathbf{U}_R = (\varphi_b, \partial_x \varphi_b, \mathrm{v}_y), \tag{42}$$

at the top and bottom boundaries.

Note that the normal component v_n may be specified in place of φ_b in the case of Neumann problems. As discussed in [32], the hyperbolic diffusion system has one wave going out of the domain, and therefore one quantity should be left unspecified, which corresponds to the normal derivative v_n in the Dirichlet case (or φ in the Neumann case). Or it may be argued that since the hyperbolic diffusion system is equivalent to the original diffusion equation in the pseudo steady state, the boundary condition should also be the same as the original problem. The tangential derivative can be specified since φ is known in the Dirichlet case, but it is not necessary; the results are very similar with and without specifying $\partial_s \varphi_b$.

3.5 Steady Solver

For a steady problem, one can drop off the term associated with the physical time. A steady solution can be obtained by an implicit solver with the local pseudo time step defined at a cell i as

$$\Delta\tau = \mathrm{CFL} \frac{2\Omega_i}{\sum_{k\in\{k_i\}} \left(|a_n|_k + \nu/L_r + \Omega_k/T_r\right)}, \tag{43}$$

where $\{k_i\}$ is a set of neighbor cells of the cell i. The solution is updated as

$$\mathbf{V}^{k+1} = \mathbf{V}^k + \Delta\mathbf{V}, \tag{44}$$

where k is the iteration counter, and the correction $\Delta\mathbf{V}$ is obtained by solving the linearized system for Eq. (17), which is given at the cell i as

$$\left(\frac{\mathbf{M}_\tau}{\Delta\tau} - \frac{\partial \mathbf{R}}{\partial \mathbf{V}}\right) \Delta\mathbf{V} = \mathbf{R}(\mathbf{V}^k), \tag{45}$$

where $\mathbf{R}(\mathbf{V}^k)$ denotes the residual $\mathbf{R}(\mathbf{U}_h^R)$ evaluated at the k-th iteration. This is equivalent to performing the pseudo-time integration towards the steady state by the BDF1 scheme. Note that the Jacobian matrix $\frac{\partial \mathbf{R}}{\partial \mathbf{V}}$ is constant and needs to be computed only once at the beginning of the computation for linear equations considered in this study. The most widely used methods to solve this linear system are iterative solution methods and approximate factorization methods. In this study, GMRES+LU-SGS and GCR+SGS(n_r) is used, where LU-SGS/SGS(n_r) serves as the preconditioner, where n_r is the number of relaxations. For the hyperbolic rDG schemes, the Jacobian matrix $\frac{\partial \mathbf{R}}{\partial \mathbf{V}}$ is constructed by ignoring the reconstructed high-order moments. Therefore, the solver is not Newton's method even when $\Delta\tau \to \infty$.

3.6 Unsteady Scheme

As for unsteady problems, we employ the ESDIRK scheme [47] for the physical time integration:

$$\begin{aligned}
&\text{(i)} \quad \mathbf{V}^{(1)} = \mathbf{V}^n, \\
&\text{(ii)} \quad \text{For } i = 2, \ldots, m, \\
&\qquad \mathbf{M}_\tau \frac{\partial \mathbf{V}^{(i)}}{\partial \tau} + \frac{\mathbf{M}_t}{\Delta t}\left(\mathbf{V}^{(i)} - \mathbf{V}^n\right) = \sum_{j=1}^{i} a_{ij} \mathbf{R}(\mathbf{V}^{(j)}), \\
&\text{(iii)} \quad \mathbf{V}^{n+1} = \mathbf{V}^{(m)},
\end{aligned} \tag{46}$$

where Δt is a physical time step, and a_{ij} are the Butcher coefficients of the scheme. The Butcher table for the third-order ESDIRK3 scheme ($m = 4$) employed is shown in Table 2. Due to the fact that $a_{11} = 0$, the first stage is explicit. At the i-th stage ($i \geq 2$), if one defines the source term associated with it as

$$\mathbf{Q} = \frac{\mathbf{M}_t}{\Delta t} \mathbf{V}^n + \sum_{j=1}^{i-1} a_{ij} \mathbf{R}(\mathbf{V}^{(j)}), \tag{47}$$

ACCEPTED MANUSCRIPT

the equation for the intermediate stages can be rewritten as

$$\mathbf{M}_\tau \frac{\partial \mathbf{V}^{(i)}}{\partial \tau} = a_{ii}\mathbf{R}(\mathbf{V}^{(i)}) + \mathbf{Q} - \frac{\mathbf{M}_t}{\Delta t}\mathbf{V}^{(i)}. \tag{48}$$

This system needs to be solved at each stage for the pseudo steady state. It is solved by the steady solver described in the previous section, i.e.,

$$\mathbf{V}^{(i),k+1} = \mathbf{V}^{(i),k} + \Delta\mathbf{V}, \tag{49}$$

where

$$\left(\frac{\mathbf{M}_\tau}{\Delta\tau} + \frac{\mathbf{M}_t}{\Delta t} - a_{ii}\frac{\partial \mathbf{R}}{\partial \mathbf{V}}\right)\Delta\mathbf{V} = a_{ii}\mathbf{R}(\mathbf{V}^{(i),k}) + \mathbf{Q} - \frac{\mathbf{M}_t}{\Delta t}\mathbf{V}^{(i),k}. \tag{50}$$

Table 2: Butcher tableau for the third-order ESDIRK scheme [47].

$c_1 = 0$	$a_{11} = 0$	0	0	0
$c_2 = \frac{1767732205903}{2027836641118}$	$a_{21} = \frac{1767732205903}{4055673282236}$	$a_{22} = a_{44}$	0	0
$c_3 = \frac{3}{5}$	$a_{31} = \frac{274623878719}{10658868560708}$	$a_{32} = \frac{-640167445237}{6845629431997}$	$a_{33} = a_{44}$	0
$c_4 = 1$	$a_{41} = b_1$	$a_{42} = b_2$	$a_{43} = b_3$	$a_{44} = \frac{1767732205903}{4055673282236}$
V^{n+1}	$b_1 = \frac{1471266399579}{7840856788654}$	$b_2 = \frac{-4482444167858}{7529755066697}$	$b_3 = \frac{112661239266428}{11593286722821}$	b_4

It is noted that the main focus of the present study for unsteady problems is to demonstrate the unsteady capability of the developed hyperbolic rDG schemes. The ESDIRK schemes are highly desirable for practical applications, being unconditionally stable and thus allowing arbitrary size of time step without introducing low-order errors. The use of a lower-order time integration scheme such as the BDF2 method may be allowed, but the time step needs to be small enough in order not to waste high-order spatial accuracy. To avoid such a possibility, the time-integration scheme must be stable and high-order, and the ESDIRK schemes are highly suitable to meet the requirement. In this study, we focus on the third-order ESDIRK scheme with a small time step and demonstrate that the developed hyperbolic rDG schemes can be used to solve unsteady problems with the ESDIRK scheme. A detailed study on comparison with other schemes, efficiency assessment, and effects of time step size, is left as future work.

4 Numerical Examples

For all computations, the relaxation time T_r is defined as in Equation (11), which is independent of mesh spacing. In Toro [43], T_r is carefully defined to be proportional to ν and also to a power of the mesh spacing to preserve design accuracy. Such a careful definition is required because they directly solve the hyperbolic formulation, Equation (3), with $\tau = t$ for time-dependent problems, which is equivalent to the original equation, Equation (1), only in the limit $T_r \to 0$. In contrast, we solve the original equation with implicit time-stepping schemes, where the hyperbolic formulation is used only for the spatial discretization and the steady solver used at each physical time step. This approach is valid because the hyperbolic formulation is equivalent to the original equation for any T_r in the pseudo steady state. Therefore, T_r only affects the dissipative character of the numerical flux and the nature of iterative steady convergence. The formula in Equation 11 has been chosen to avoid vanishing dissipation for large Re [52] and accelerate iterative convergence [32, 33, 34]. In our approach, the equivalence to the original equation is established not by a small relaxation time as in Toro [43], but by solving the hyperbolic formulation to a pseudo steady state at sufficient levels of residual convergence. It is particularly noteworthy that T_r used here has a completely opposite dependence on ν, i.e., $T_r = O(1/\nu)$, in contrast to Toro [43] where $T_r = O(\nu)$. For small ν, as typical in high-Re problems, the latter leads to a hyperbolic system with stiff source terms. Also, their T_r must depend on the mesh size as well as the order of accuracy [43], whereas our T_r independent of the mesh size and remains the same for any order of accuracy.

ACCEPTED MANUSCRIPT

4.1 1D Boundary layer problem

In the first test case, we consider the following 1D problem

$$\frac{\partial \varphi}{\partial t} + a\frac{\partial \varphi}{\partial x} = \nu \frac{\partial^2 \varphi}{\partial x^2} + f(x), \quad 0 \le x \le 1, \tag{51}$$

with

$$\varphi(0) = \varphi(1) = 1, \tag{52}$$

and the source term $f(x)$ is given as

$$f(x) = \frac{\pi}{\mathrm{Re}}\left(a\cos(\pi x) + \pi\nu\sin(\pi x)\right), \quad \mathrm{Re} = \frac{a}{\nu}. \tag{53}$$

The exact steady solution to the problem is

$$\varphi(x) = \frac{\exp(-\mathrm{Re} - \exp(x\mathrm{Re} - \mathrm{Re}))}{\exp(-\mathrm{Re}) - 1} + \frac{1}{\mathrm{Re}}\sin(\pi x). \tag{54}$$

The exact solution can be regarded as a function of Reynolds number. In the diffusion limit, it would be a smooth sine curve, while developing a very narrow boundary layer near $x = 1$ if advection limit is approached. Based on the FOHS formulation, we apply the developed hyperbolic rDG methods to solve the following equivalent system.

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = \mathbf{S}, \tag{55}$$

where

$$\mathbf{U} = \begin{pmatrix} \varphi \\ \mathrm{v}_x \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} a\varphi - \nu \mathrm{v}_x \\ -\varphi/T_r \end{pmatrix}, \quad \mathbf{S} = \begin{pmatrix} f(x) \\ -\mathrm{v}_x/T_r \end{pmatrix}. \tag{56}$$

In order to capture the boundary layer, one would need enough resolution in the layer. Thus, numerical experiments are carried out with non-uniform grids generated from a uniform grid by the following mapping

$$x_i = \frac{1 - \exp(-\alpha\xi_i)}{1 - \alpha}, \quad \xi_i = \frac{i-1}{Nelem}. \tag{57}$$

For high Reynolds number case, or in other words, in the advection limit, one would need to increase α to ensure the convergence. In this paper, we set $a = 1$, and all numerical results were obtained for a wide range of the Reynolds numbers, $\mathrm{Re} = 10^k$, where $k = -8, 0, 8$ with varying ν. And the corresponding α is set to be $4.5, 4.5, 22.5$. And the number of the elements is set to be 32, 64, 128, and 256 for all Reynolds numbers.

Several hyperbolic rDG methods are applied here. For cases with smaller Reynolds numbers (Re= 1 and Re= 10^{-8}), all presented method achieve deigned or even higher order of accuracy. See Table 3. However, for high Reynolds number case, a very strong boundary layer arises near $x = 1$, and some hyperbolic methods, including those in Ref. [45], become unstable. These results are somewhat consistent with those in Ref. [52]. The presented methods are in the same family of the Scheme II in Ref. [52], which directly uses the gradient variables to construct higher order polynomials for the primary solution variable. Ref. [52] shows that this efficient construction introduces an artificial negative diffusion coefficient for a high-Reynolds-number boundary-layer-type problem, resulting accuracy and convergence problems [52]. However, the first-order scheme, and our target schemes based on rDG($\mathrm{P}_0\mathrm{P}_k$)with $k = 1, 2$ do not suffer such a problem and are found to be stable, delivering their design order of accuracy. Unstable schemes may be remedied by extending techniques suggested in Ref. [52], e.g., DG(P_n)+DG(P_n) or rDG($\mathrm{P}_n\mathrm{P}_m$)+rDG($\mathrm{P}_n\mathrm{P}_m$) [53], the upwind flux based on the unified eigen-structure, or indirect gradient reconstructions called Scheme IQ in the Ref. [52]. These approaches should be explored in future work.

ACCEPTED MANUSCRIPT

Table 3: Order of accuracy with different Re.

Scheme [DoFs]	Advection $\nu = 10^{-8}, \mathrm{Re} = 10^{8}$		Advection-Diffusion $\nu = 1, \mathrm{Re} = 1$		Diffusion $\nu = 10^{8}, \mathrm{Re} = 10^{-8}$	
	φ	v_x	φ	v_x	φ	v_x
DG(P_0P_1)+DG(P_0) [2]	0.91	0.98	1.00	1.00	1.01	1.00
DG(P_0P_2)+DG(P_1) [3]	-	-	2.00	2.00	2.00	2.00
DG(P_0P_3)+DG(P_2) [4]	-	-	4.01	3.00	3.96	3.00
DG(P_0P_2)+rDG_LS(P_0P_1) [2]	2.00	2.14	1.97	2.05	1.97	2.04
DG(P_0P_3)+rDG_LS(P_1P_2) [3]	-	-	3.12	2.68	3.11	2.69
DG(P_0P_2)+rDG_VR(P_0P_1) [2]	2.00	1.90	2.00	2.05	2.00	2.05
DG(P_0P_3)+rDG_VR(P_0P_2) [2]	3.96	3.02	3.99	3.69	4.06	3.66
DG(P_0P_3)+rDG_VR(P_1P_2) [3]	-	-	3.89	3.06	3.80	3.04

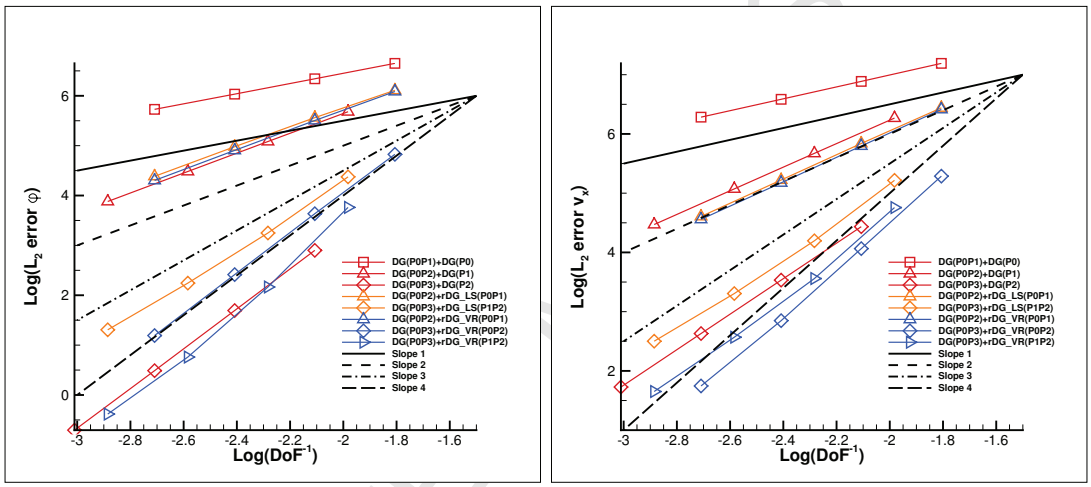

Figure 1: Grid refinement study for Re= 10^{-8}.

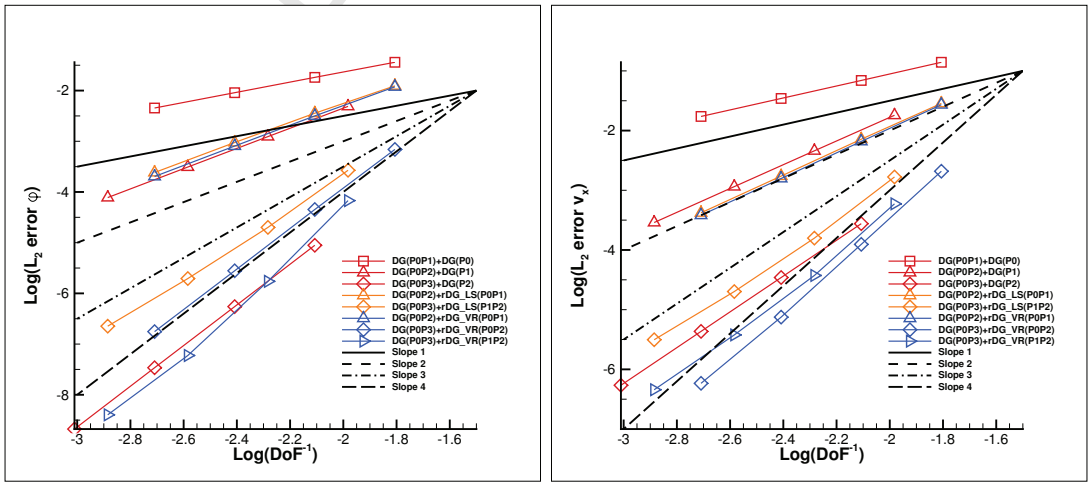

Figure 2: Grid refinement study for Re= 1.

ACCEPTED MANUSCRIPT

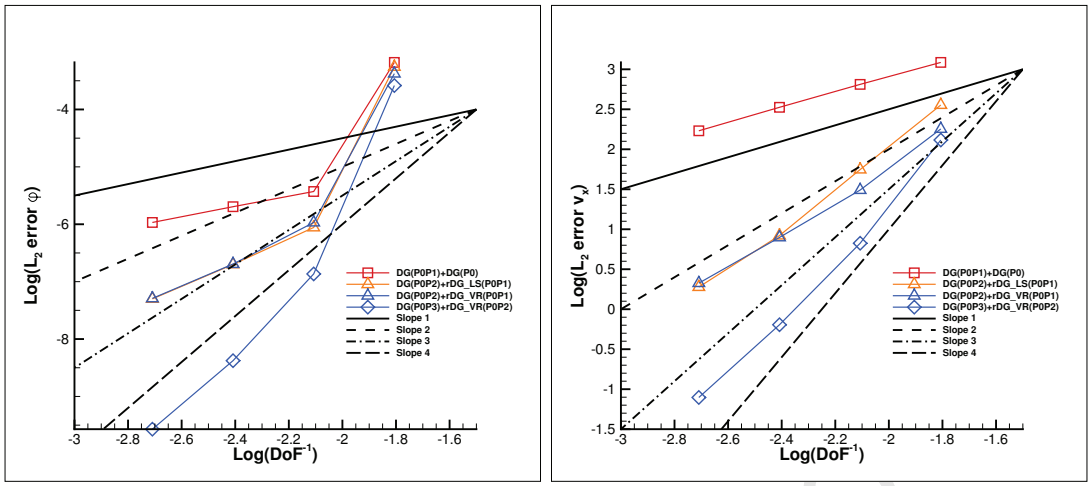

Figure 3: Grid refinement study for Re= 10^8.

4.2 2D steady advection diffusion problem

A steady model advection diffusion problem in a unit square is considered in this section, i.e.,

$$\frac{\partial \varphi}{\partial t} + a\frac{\partial \varphi}{\partial x} + b\frac{\partial \varphi}{\partial y} = \nu\left(\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2}\right), \tag{58}$$

with the exact solution given by

$$\varphi(x,y) = C\cos(A\pi\eta)\exp\left(\frac{1-\sqrt{1+4A^2\pi^2\nu^2}}{2\nu}\xi\right), \tag{59}$$

and

$$\xi = ax + by, \quad \eta = bx - ay. \tag{60}$$

$(a,b) = (2,1)$, $A = 2$, $C = -0.009$ with $\nu = 10^{-8}, 10^0, 10^8$ are set in this test case. Three types of meshes: regular, irregular, and heterogeneous as shown in Figure 4, are considered to test the performance of the hyperbolic rDG methods not only on regular but also on highly distorted grids. A grid refinement study is conducted to assess the accuracy and convergence of the hyperbolic rDG methods for these three types of meshes. In each type of mesh, the advection limit case ($\nu = 10^{-8}$), the advection-diffusion case ($\nu = 1$), and the diffusion limit case ($\nu = 10^8$) are investigated. The results obtained by different rDG methods are presented in Tables 4 to 6 and in Figures 5 to 13. In addition, an efficiency comparison study is carried out for the advection-diffusion case on the irregular mesh. Note that solutions on two more finer grids (the finest mesh contains $131,072$ triangles and $66,049$ points) are computed for a better illustration of the efficiency assessment. The L2 error obtained by different hyperbolic rDG methods versus CPU time is shown in Figures 14, where the slope indicates the efficiency of the presented schemes.

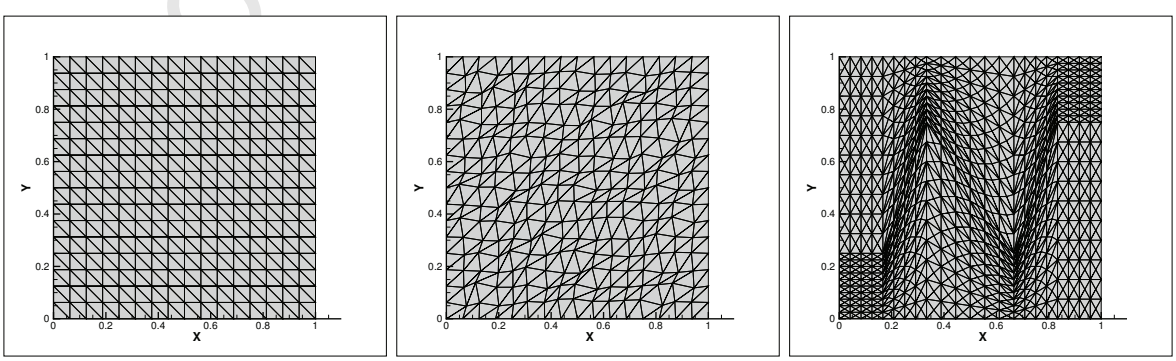

Figure 4: The sample mesh of each type, i.e., 17×17 regular grid (left), 17×17 irregular grid (middle), and 23×21 heterogeneous grid (right).

ACCEPTED MANUSCRIPT

Table 4: Order of accuracy on regular grids with different ν.

Scheme [DoFs]	Advection $\nu = 10^{-8}, \mathrm{Re} = \sqrt{5} \times 10^{8}$		Advection-Diffusion $\nu = 1, \mathrm{Re} = \sqrt{5}$		Diffusion $\nu = 10^{8}, \mathrm{Re} = \sqrt{5} \times 10^{-8}$	
	φ	v_x	φ	v_x	φ	v_x
DG(P_0P_1)+DG(P_0) [3]	2.11	0 .99	1.26	1.00	1.95	1.00
DG(P_0P_2)+DG(P_1) [6]	3.02	2.01	2.03	1.62	2.05	1.62
DG(P_0P_3)+DG(P_2) [10]	3.97	2.97	3.65	2.99	3.59	2.97
DG(P_0P_2)+rDG_LS(P_0P_1) [3]	3.18	2.01	-	-	-	-
DG(P_0P_3)+rDG_LS(P_1P_2) [6]	4.14	3.22	3.69	2.82	3.70	2.82
DG(P_0P_2)+rDG_VR(P_0P_1) [3]	3.11	2.00	2.88	2.17	2.88	2.18
DG(P_0P_3)+rDG_VR(P_0P_2) [3]	4.49	3.28	3.04	2.90	2.99	2.81
DG(P_0P_3)+rDG_VR(P_1P_2) [6]	4.30	3.06	3.74	3.12	3.76	3.11

Table 5: Order of accuracy on irregular grids with different ν.

Scheme [DoFs]	Advection $\nu = 10^{-8}, \mathrm{Re} = \sqrt{5} \times 10^{8}$		Advection-Diffusion $\nu = 1, \mathrm{Re} = \sqrt{5}$		Diffusion $\nu = 10^{8}, \mathrm{Re} = \sqrt{5} \times 10^{-8}$	
	φ	v_x	φ	v_x	φ	v_x
DG(P_0P_1)+DG(P_0) [3]	1.93	0.99	1.26	0.92	1.92	0.89
DG(P_0P_2)+DG(P_1) [6]	2.74	1.91	2.38	1.73	2.39	1.73
DG(P_0P_3)+DG(P_2) [10]	3.97	2.97	2.99	2.72	2.96	2.75
DG(P_0P_2)+rDG_LS(P_0P_1) [3]	2.80	1.92	-	-	-	-
DG(P_0P_3)+rDG_LS(P_1P_2) [6]	4.18	3.19	3.61	2.67	3.61	2.70
DG(P_0P_2)+rDG_VR(P_0P_1) [3]	2.74	1.93	2.78	2.21	2.81	2.22
DG(P_0P_3)+rDG_VR(P_0P_2) [3]	3.77	2.97	3.02	2.62	3.01	2.78
DG(P_0P_3)+rDG_VR(P_1P_2) [6]	3.87	3.01	3.97	3.23	3.94	3.22

Overall, the hyperbolic rDG methods are able to deliver the designed or higher order of accuracy for most of the cases. Note that there exists no boundary layer in this problem, and thus all schemes converged without any problem in the advection limit. However, we do observe that DG(P_0P_2)+rDG_LS(P_0P_1) being unstable for non-advection limit case. It appears that this issue can be fixed by either adding more cells in the LS stencil or applying limiters. On the other hand, DG(P_0P_3)+rDG_LS(P_1P_2), DG(P_0P_3)+rDG_VR(P_0P_2), and DG(P_0P_3)+rDG_VR(P_1P_2) are able to deliver fourth-order accuracy in φ and third-order accuracy in gradients in all the cases very effectively. Variational reconstruction is based on a global stencil with compact data structure, resolving the stability issue of the LS reconstruction and thus making the extension to higher order reconstruction simple and straightforward. Furthermore, the variational reconstruction can be performed without taking into account boundary conditions. One can observe from the efficiency plots that for the same error, 10^{-7} in primary variable φ or 10^{-5} in the gradients, the presented rDG methods are more efficient than their DG counterparts and the DG(P_0P_3)+rDG_VR(P_1P_2) is the most efficient method as expected. The high efficiency of the VR based rDG methods is attributed to the fact that only one SGS relaxation sweep is used to solve the global linear system in the variational reconstruction at each time step and the convergence is only achieved at the steady state. The numerical results indicate that the presented hyperbolic rDG schemes with the variational reconstruction are attractive and worth further investigation.

ACCEPTED MANUSCRIPT

Table 6: Order of accuracy on heterogeneous grids with different ν.

Scheme [DoFs]	Advection $\nu = 10^{-8}, \mathrm{Re} = \sqrt{5} \times 10^{8}$		Advection-Diffusion $\nu = 1, \mathrm{Re} = \sqrt{5}$		Diffusion $\nu = 10^{8}, \mathrm{Re} = \sqrt{5} \times 10^{-8}$	
	φ	v_x	φ	v_x	φ	v_x
DG(P_0P_1)+DG(P_0) [3]	2.11	0.95	1.17	1.00	1.98	1.00
DG(P_0P_2)+DG(P_1) [6]	3.15	2.07	2.60	1.92	2.60	1.92
DG(P_0P_3)+DG(P_2) [10]	4.08	3.04	3.12	2.76	3.10	2.76
DG(P_0P_2)+rDG_LS(P_0P_1) [3]	3.05	2.06	-	-	-	-
DG(P_0P_3)+rDG_LS(P_1P_2) [6]	4.17	3.11	3.87	2.99	4.00	3.10
DG(P_0P_2)+rDG_VR(P_0P_1) [3]	3.05	2.06	2.38	2.01	2.39	2.21
DG(P_0P_3)+rDG_VR(P_0P_2) [3]	4.51	3.27	3.57	3.09	3.53	3.04
DG(P_0P_3)+rDG_VR(P_1P_2) [6]	4.23	3.08	3.89	3.11	3.86	2.99

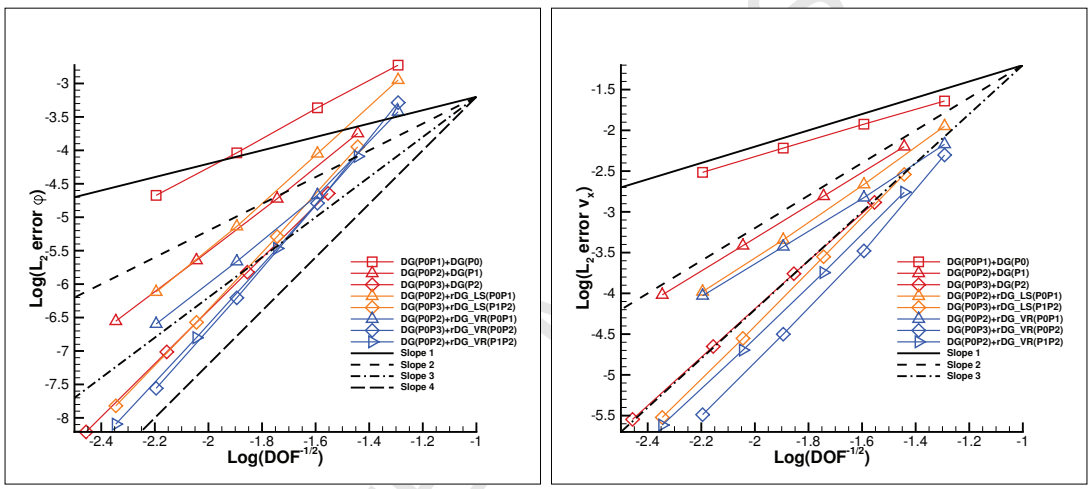

Figure 5: Grid refinement study on regular grids with $\nu = 10^{-8}, \mathrm{Re} = \sqrt{5} \times 10^{8}$.

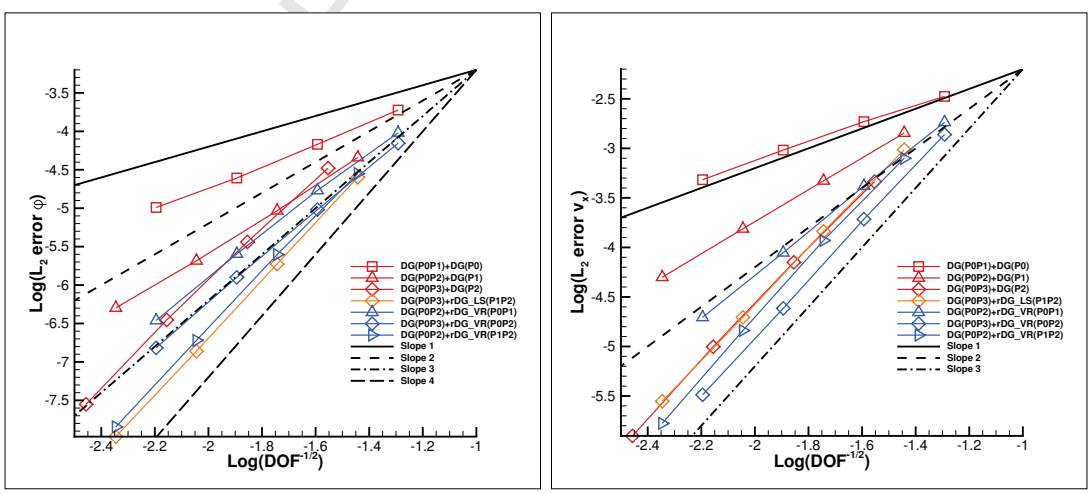

Figure 6: Grid refinement study on regular grids with $\nu = 1, \mathrm{Re} = \sqrt{5}$.

ACCEPTED MANUSCRIPT

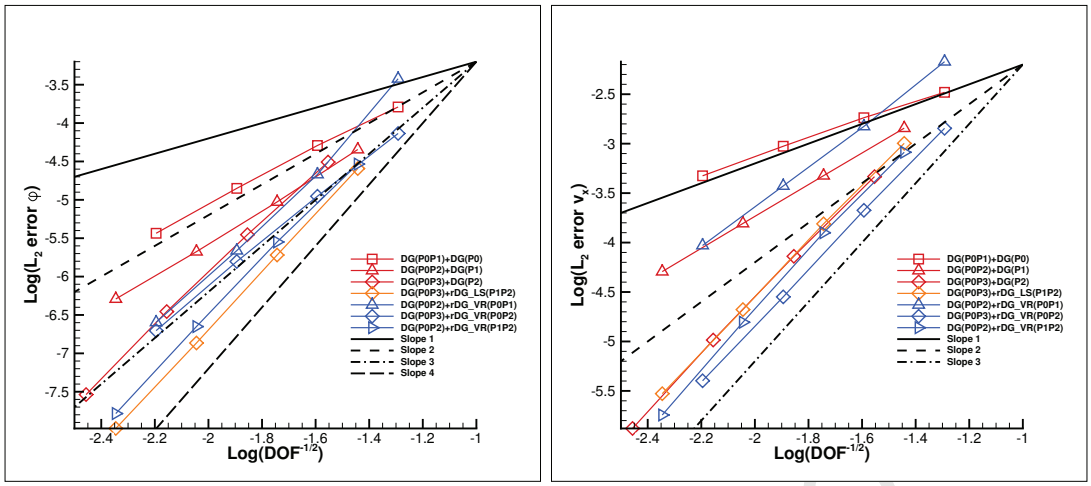

Figure 7: Grid refinement study on regular grids with $\nu = 10^8, \mathrm{Re} = \sqrt{5} \times 10^{-8}$.

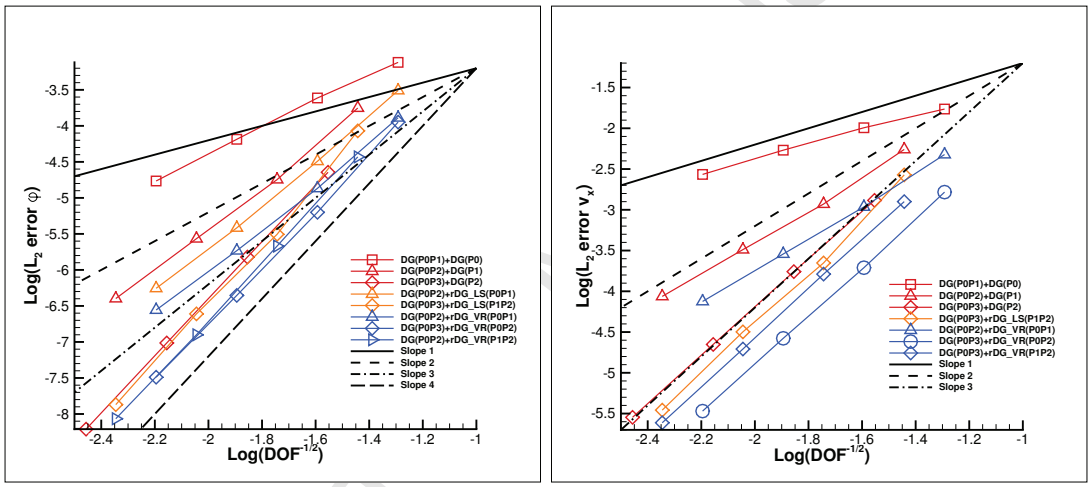

Figure 8: Grid refinement study on irregular grids with $\nu = 10^{-8}, \mathrm{Re} = \sqrt{5} \times 10^8$.

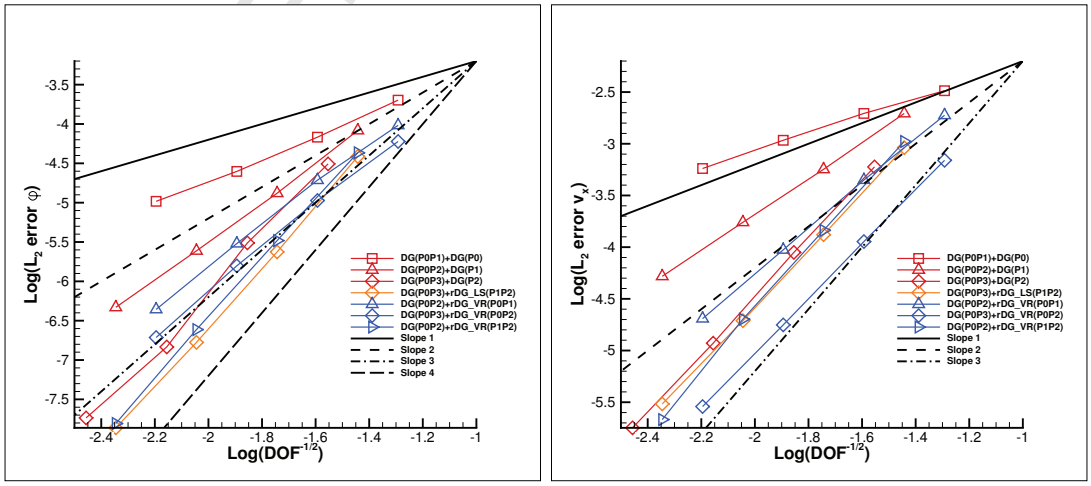

Figure 9: Grid refinement study on irregular grids with $\nu = 1, \mathrm{Re} = \sqrt{5}$.

ACCEPTED MANUSCRIPT

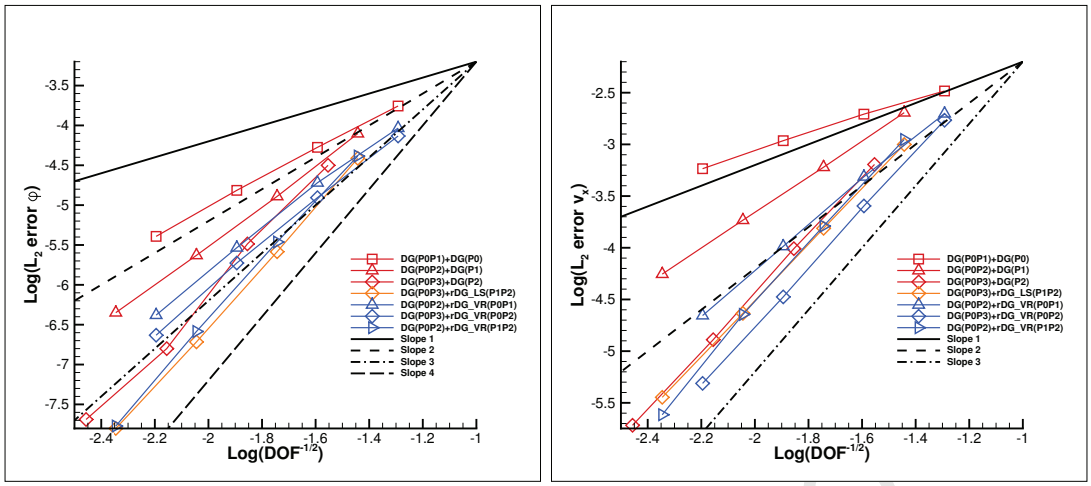

Figure 10: Grid refinement study on irregular grids with $\nu = 10^8, \mathrm{Re} = \sqrt{5} \times 10^{-8}$.

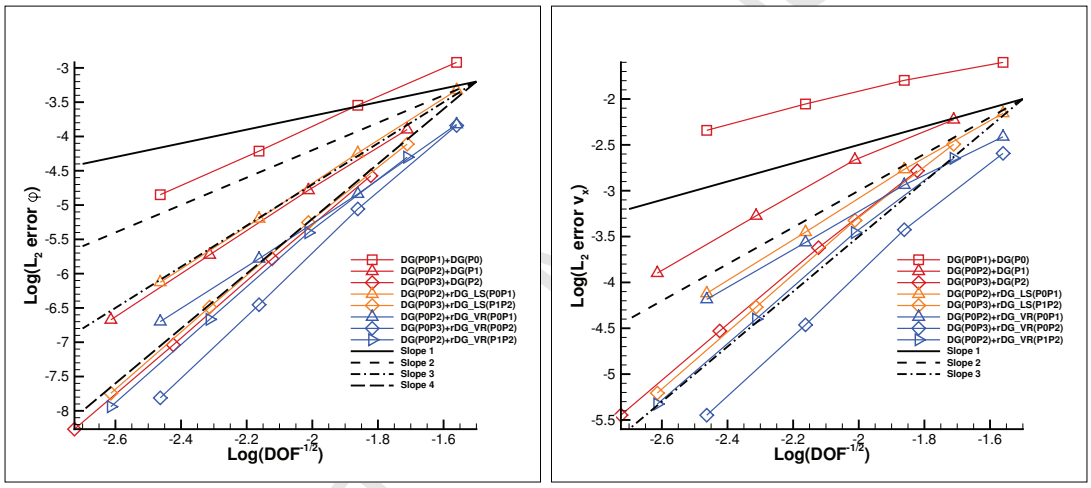

Figure 11: Grid refinement study on heterogeneous grids with $\nu = 10^{-8}, \mathrm{Re} = \sqrt{5} \times 10^8$.

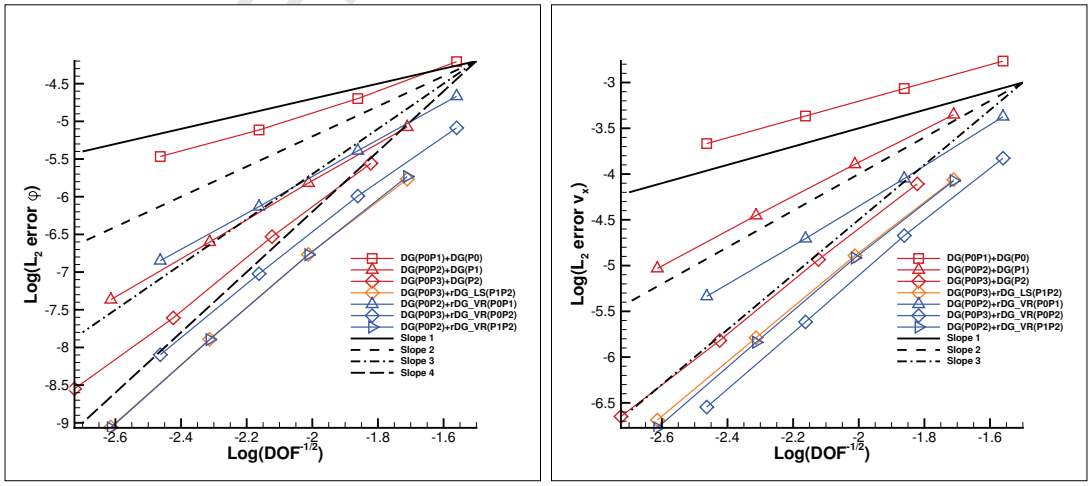

Figure 12: Grid refinement study on heterogeneous grids with $\nu = 1, \mathrm{Re} = \sqrt{5}$.

ACCEPTED MANUSCRIPT

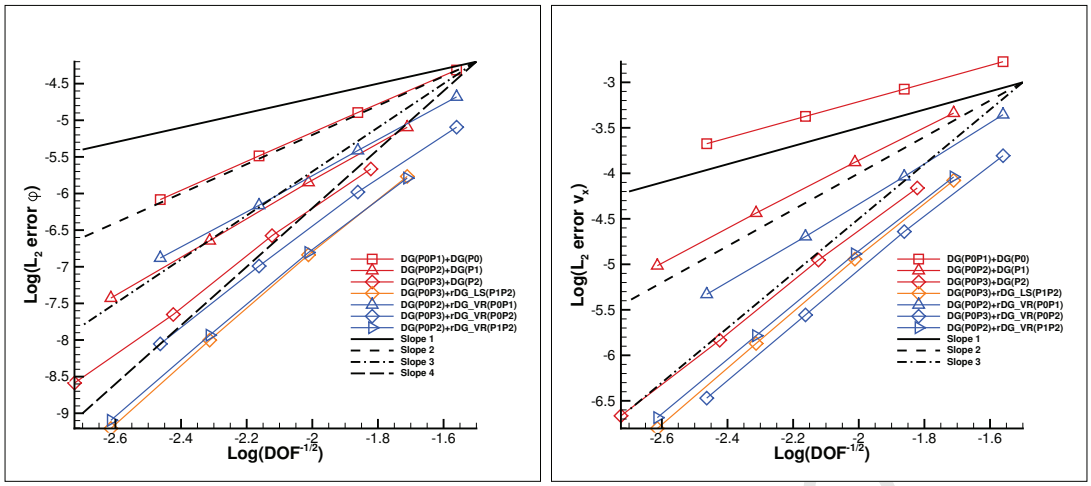

Figure 13: Grid refinement study on heterogeneous grids with $\nu = 10^8, \mathrm{Re} = \sqrt{5} \times 10^{-8}$.

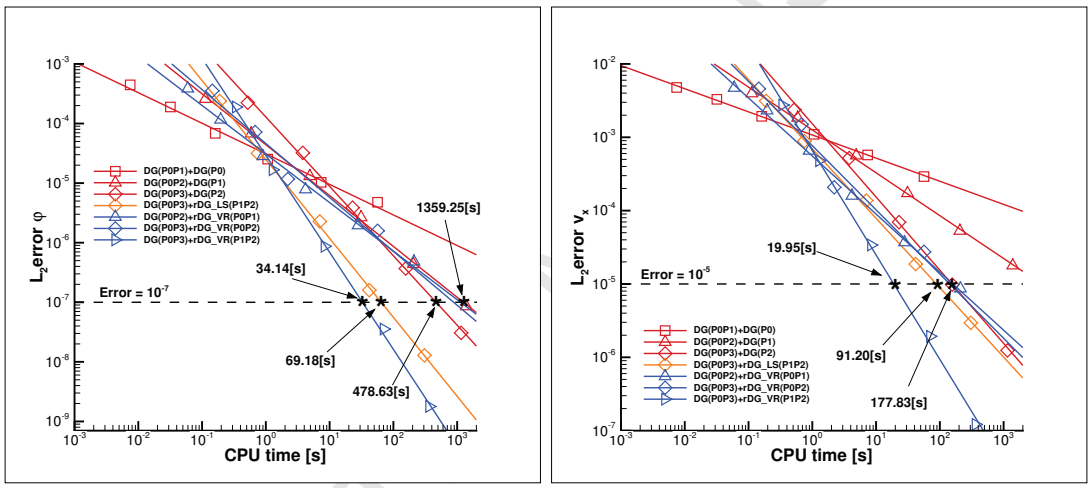

Figure 14: Efficiency plots of the hyperbolic rDG methods: Error versus CPU time for a sequence of 6 irregular grids with $\nu = 1, \mathrm{Re} = \sqrt{5}$.

4.3 1D unsteady advection diffusion problem

Consider the following exact solution to the 1D unsteady advection diffusion problem,

$$\varphi(x,t) = \frac{1}{\sqrt{4t+1}} \exp\left(-\frac{(x-at-x_0)^2}{\nu(4t+1)}\right), \quad 0 \le x \le 2. \tag{61}$$

where

$$a = 10^4, \quad \nu = 0.01, \quad x_0 = 0.5. \tag{62}$$

The parameters are chosen to yield an advection dominant problem. The initial Gaussian bump will travel with a constant velocity with a small diffusion effect. In this paper, a grid refinement test has been carried out to verify the spatial order of accuracy in this unsteady case. A small physical time step $\Delta t = 10^{-9}$ has been set for all the grids with a fixed final time $t_{\mathrm{end}} = 10^{-4}$. The grids are uniform with $nelem = 32, 64, 128$ and 256. Dirichlet boundary conditions has been applied on both ends. All the presented methods use the same degrees of freedom, 2, which is equivalent to a conventional P_1 DG method. The numerical results are shown in Figure 15. As we can see, all the presented hyperbolic rDG methods can deliver the design or higher order of accuracy for the unsteady problem.

Secondly, a comparison between the developed schemes and a conventional DG (Direct DG in this study), is shown in Figure 16. Periodic boundary conditions are enforced with $\Delta t = 10^{-9}$, $t_{\mathrm{end}} = 10^{-3}$ on a uniform mesh

ACCEPTED MANUSCRIPT

($nelem = 32$). Clearly, the presented hyperbolic rDG methods can outperform the conventional counterpart, better resolving the peak of the Gaussian profile.

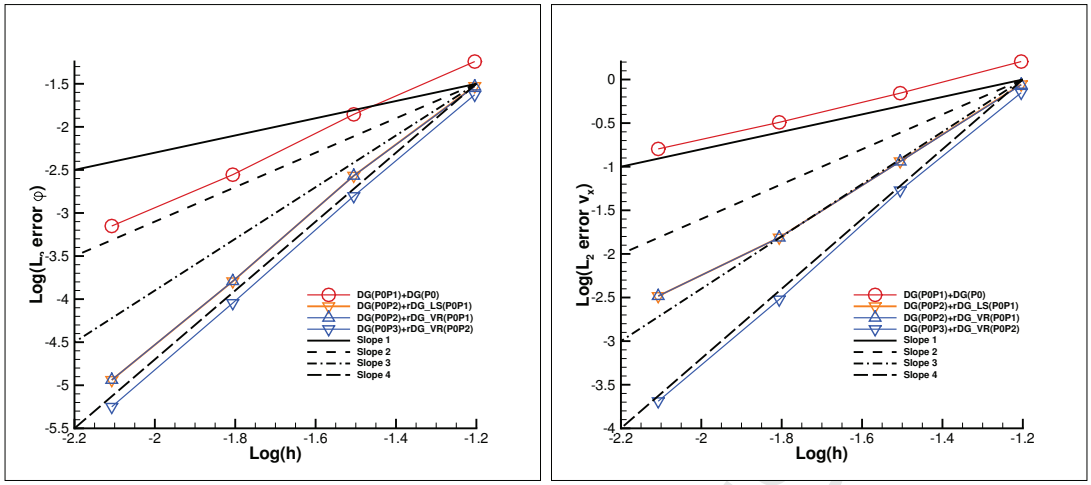

Figure 15: Grid refinement study on regular grids for 1D unsteady case.

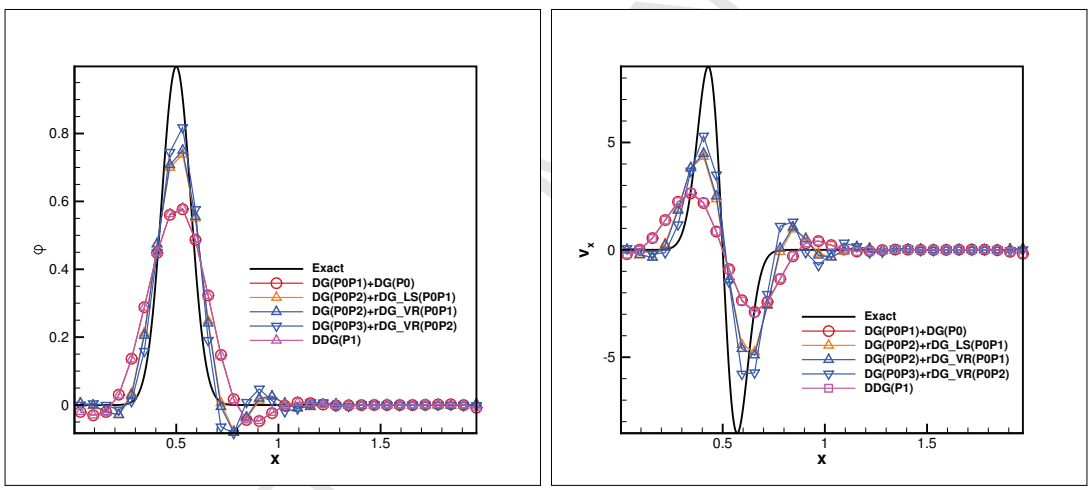

Figure 16: Grid refinement study on regular grids for 1D unsteady case.

4.4 2D unsteady advection diffusion problem

In this case, a 2D unsteady case are considered. The analytical solution is given as

$$\varphi(x,y,t) = \frac{1}{4t+1}\exp\left(-\frac{(x-at-x_0)^2+(y-bt-y_0)^2}{\nu(4t+1)}\right), \quad (x,y)\in[0,2]\times[0,2], \tag{63}$$

where

$$x_0 = y_0 = 1.0, \quad a = b = 10^{-5}, \quad \nu = 0.01. \tag{64}$$

The regular and irregular grids of the 2D steady case are applied here with a scaling factor of 2. The physical time step is set as $\Delta t = 10^{-3}$ with $t_{end} = 1$. Dirichlet boundary conditions are applied on all the boundary faces. Here, several hyperbolic rDG schemes are presented here, including DG(P_0P_4)+rDG_VR(P_1P_3). The numerical results are shown in Table 7 and Figures 17 to 18. All the presented schemes are shown to provide the design or higher order of accuracy in this unsteady case.

ACCEPTED MANUSCRIPT

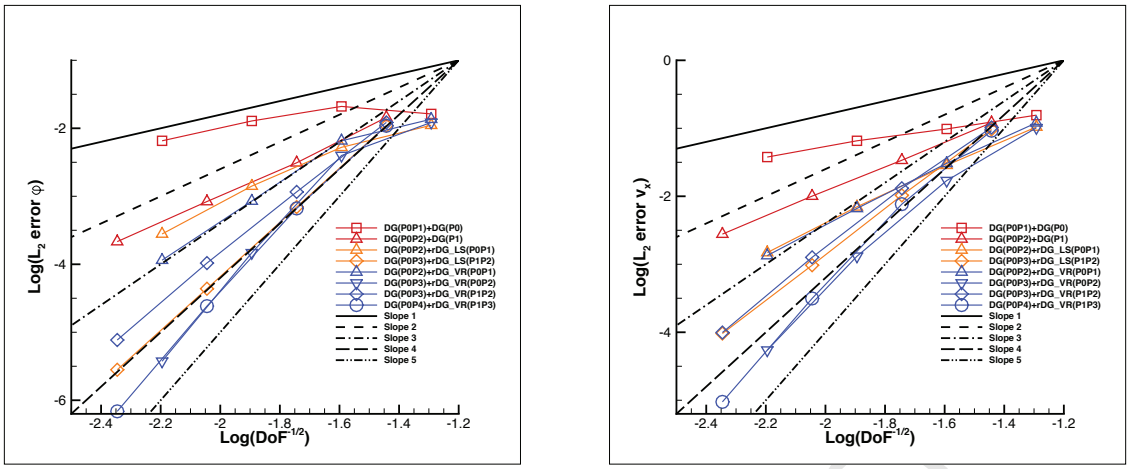

Figure 17: Grid refinement study on regular grids for 2D unsteady case.

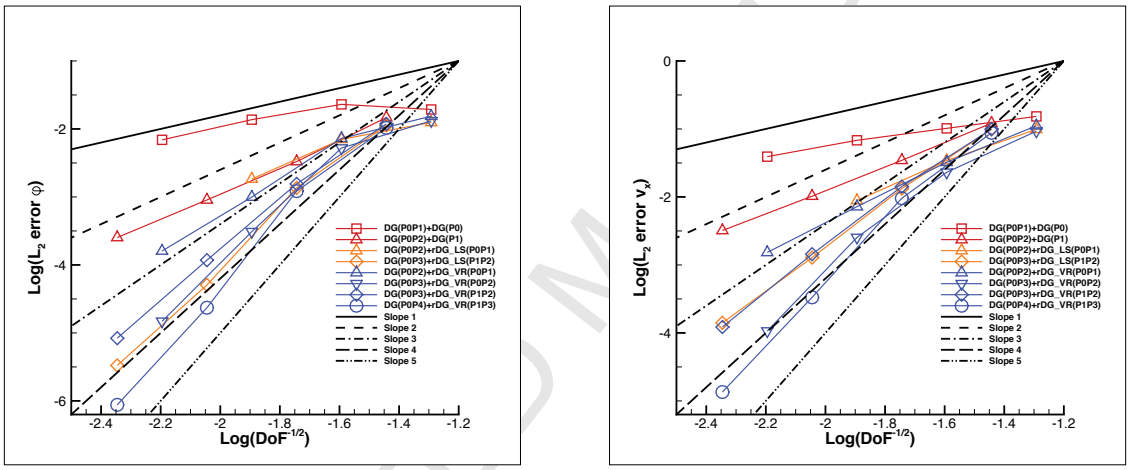

Figure 18: Grid refinement study on irregular grids for 2D unsteady case.

Table 7: Order of accuracy on 2D unsteady advection diffusion problem.

Scheme [DoFs]	Regular		Irregular	
	φ	v_x	φ	v_x
DG(P_0P_1)+DG(P_0) [3]	0.97	0.79	0.99	0.80
DG(P_0P_2)+DG(P_1) [6]	1.96	1.86	1.84	1.68
DG(P_0P_2)+rDG_LS(P_0P_1) [3]	2.34	2.24	1.94	1.95
DG(P_0P_3)+rDG_LS(P_1P_2) [6]	3.96	3.34	3.93	3.18
DG(P_0P_2)+rDG_VR(P_0P_1) [3]	2.89	2.30	2.64	2.24
DG(P_0P_3)+rDG_VR(P_0P_2) [3]	5.30	4.57	4.38	4.57
DG(P_0P_3)+rDG_VR(P_1P_2) [6]	3.76	3.66	3.81	3.56
DG(P_0P_4)+rDG_VR(P_1P_3) [6]	5.14	5.06	4.74	4.62

5 Conclusions and Outlook

High-order reconstructed discontinuous Galerkin (rDG) methods based on a first-order hyperbolic system (FOHS) for advection-diffusion equations have been developed and presented. The FOHS formulation allows a

ACCEPTED MANUSCRIPT

straightforward DG discretization for diffusion. Additional gradient variables introduced to form a hyperbolic system are used to reduce the total number of degrees of freedom such that a P_k hyperbolic DG scheme is almost equivalent to a conventional P_{k+1} DG scheme in terms of accuracy. The cost is further reduced by the rDG method, where the highest order terms in the polynomials are obtained by gradient reconstruction methods. The study shows that the variational gradient reconstruction method yields more stable and accurate results than the least-squares method does. The resulting hyperbolic rDG schemes have been demonstrated for both steady and unsteady advection-diffusion problems, giving the same order of accuracy in both the solution and its derivatives on regular, irregular, and heterogeneous grids. The steady problem is solved efficiently by an implicit solver. The unsteady problem is solved by the ESDIRK scheme, where the steady solver is used to solve the system of unsteady residual equations at each stage. The numerical examples showed in the paper illustrate the capability and the potential of the developed methods, indicating that the hyperbolic-rDG methods provide attractive alternatives to solve both steady and unsteady advection-diffusion equations. Future work would be focused on extending the hyperbolic rDG method to Navier-Stokes equation on fully 3D unstructured grids.

Acknowledgments

This work has been funded by the U.S. Army Research Office under the contract/grant number W911NF-16-1-0108 with Dr. Matthew Munson as the program manager.

References

[1] W.H. Reed and T.R. Hill. Triangular Mesh Methods for the Neutron Transport Equation. *Los Alamos Scientific Laboratory Report*, LA-UR–73-479, 1973.

[2] B. Van Leer. Towards the ultimate conservative difference scheme. IV. a new approach to numerical convection. *Journal of Computational Physics*, 23:276–299, 1977.

[3] B. Cockburn, G. Karniadakis, and Shu C. W. The Development of Discontinuous Galerkin Method. *Discontinuous Galerkin Methods, Theory, Computation, and Applications. Edited by B. Cockburn, G. E. Karniadakis, and C. W. Shu. Lecture Notes in Computational Science and Engineering*, 11:5–50, 2000.

[4] F. Bassi and S. Rebay. High-Order Accurate Discontinuous Finite Element Solution of the 2D Euler Equations. *Journal of Computational Physics*, 138:251–285, 1997.

[5] F. Bassi and S. Rebay. A High-Order Accurate Discontinuous Finite Element Method for the Numerical Solution of the Compressible Navier-Stokes Equations. *Journal of Computational Physics*, 131(2):267–279, 1997.

[6] T. C. Warburton and G. E. Karniadakis. A Discontinuous Galerkin Method for the Viscous MHD Equations. *Journal of Computational Physics*, 152:608–641, 1999.

[7] P. Rasetarinera and M. Y. Hussaini. An Efficient Implicit Discontinuous Spectral Galerkin Method. *Journal of Computational Physics*, 172:718–738, 2001.

[8] H. Luo, J. D. Baum, and R. Löhner. A Discontinuous Galerkin Method Using Taylor Basis for Compressible Flows on Arbitrary Grids. *Journal of Computational Physics*, 227(20):8875–8893, 2008.

[9] J. J. Sudirham, J. J. W. Van Der Vegt, and R. M. J. Van Damme. Space–time discontinuous galerkin method for advection–diffusion problems on time-dependent domains. *Applied numerical mathematics*, 56(12):1491–1518, 2006.

[10] M. Dumbser, M. Kaser, V. A. Titarev, and E. F. Toro. Quadrature-free non-oscillatory finite volume schemes on unstructured meshes for nonlinear hyperbolic systems. *Journal of Computational Physics*, 226(1):204–243, 2007.

[11] M. Dumbser and O. Zanotti. A unified framework for the construction of one-step finite volume and discontinuous Galerkin schemes on unstructured meshes. *Journal of Computational Physics*, 227(18):8209–8253, 2008.

ACCEPTED MANUSCRIPT

[12] M. Dumbser. Arbitrary High Order PNPM Schemes on Unstructured Meshes for the Compressible Navier-Stokes Equations. *Computers & Fluids*, 39(1):60–76, 2010.

[13] H. Luo, L. Luo, and R. Nourgaliev. A Reconstructed Discontinuous Galerkin Method for the Compressible Euler Equations on Arbitrary Grids. *Communication in Computational Physics*, 12(5):1495–1519, 2012.

[14] H. Luo, L. Luo, R. Nourgaliev, V. Mousseau, and N. Dinh. A Reconstructed Discontinuous Galerkin Method for the Compressible Navier-STokes Equations on Arbitrary Grids. *Journal of Computational Physics*, 229(19):6961–6978, 2010.

[15] H. Luo, Y. Xia, and R. Nourgaliev. A Class of Reconstructed Discontinuous Galerkin Methods in Computational Fluid Dynamics. *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C2011)*, Brazil, 2011.

[16] H. Luo, L. Luo, A Ali, R. Nourgaliev, and C. Cai. A Parallel, Reconstructed Discontinuous Galerkin Method for the Compressible Flows on Arbitrary Grids. *Communication in Computational Physics*, 9(2):363–389, 2011.

[17] H. Luo, Y. Xia, S. Li, and R. Nourgaliev. A Hermite WENO Reconstruction-Based Discontinuous Galerkin Method for the Euler Equations on Tetrahedral grids. *Journal of Computational Physics*, 231(16):5489–5503, 2012.

[18] H. Luo, Y. Xia, S. Spiegel, R. Nourgaliev, and Z. Jiang. A reconstructed discontinuous Galerkin method based on a hierarchical WENO reconstruction for compressible flows on tetrahedral grids. *Journal of Computational Physics*, 236:477–492, 2013.

[19] F. Bassi and S. Rebay. GMRES discontinuous Galerkin solution of the Compressible Navier-Stokes Equations. *Discontinuous Galerkin Methods, Theory, Computation, and Applications. Edited by B. Cockburn, G. E. Karniadakis, and C. W. Shu. Lecture Notes in Computational Science and Engineering*, 11:197–208, 2000.

[20] F. Bassi and S. Rebay. A High Order Discontinuous Galerkin Method for Compressible Turbulent Flow. *Discontinuous Galerkin Methods, Theory, Computation, and Applications. Edited by B. Cockburn, G. E. Karniadakis, and C. W. Shu. Lecture Notes in Computational Science and Engineering*, 11, 2000.

[21] B. Cockburn and C.W. Shu. The Local Discontinuous Galerkin Method for Time-dependent Convection-Diffusion System. *SIAM, Journal of Numerical Analysis*, 35(6):2440–2463, 1998.

[22] C. E. Baumann and J. T. Oden. A Discontinuous *hp* Finite Element Method for the Euler and Navier-Stokes Equations. *International Journal for Numerical Methods in Fluids*, 31(1):79–95, 1999.

[23] J. Peraire and P. O. Persson. The Compact Discontinuous Galerkin Method for Elliptic Problems. *SIAM Journal on Scientific Computing*, 30(4):1806–1824, 2008.

[24] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM journal on numerical analysis*, 39(5):1749–1779, 2002.

[25] G. Gassner, F. Lorcher, and C. D. Munz. A contribution to the construction of diffusion fluxes for finite volume and discontinuous Galerkin schemes. *Journal of Computational Physics*, 224(2):1049–1063, 2007.

[26] B. van Leer and S. Nomura. Discontinuous Galerkin Method for Diffusion. *AIAA Paper*, 2005-5108, 2005.

[27] B. van Leer and M. Lo. A Discontinuous Galerkin Method for Diffusion based on Recovery. *AIAA Paper*, 2007-4083, 2007.

[28] M. Raalte and B. van Leer. Bilinear forms for the recovery-based discontinuous Galerkin method for diffusion. *Communication in Computational Physics*, 5(2-4):683–693, 2009.

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

[30] H. Nishikawa. Two ways to extend diffusion schemes to navier-stokes schemes: Gradient formula or upwind flux. In *20th AIAA Computational Fluid Dynamics Conference*, American Institute of Aeronautics and Astronautics, Reston, VA, 2011. submitted for publication.

ACCEPTED MANUSCRIPT

[31] R. Abgrall, Dante De Santis, and Mario Ricchiuto. High-order preserving residual distribution schemes for advection-diffusion scalar problems on arbitrary grids. *SIAM Journal on Scientific Computing*, 36(3):A955–A983, 2014.

[32] H. Nishikawa. A first-order system approach for diffusion equation. I: Second-order residual-distribution schemes. *Journal of Computational Physics*, 227(1):315–352, 2007.

[33] H. Nishikawa. A first-order system approach for diffusion equation. II: Unification of advection and diffusion. *Journal of Computational Physics*, 229(11):3989–4016, 2010.

[34] H. Nishikawa. First-, second-, and third-order finite-volume schemes for diffusion. *Journal of Computational Physics*, 256:791–805, 2014.

[35] H. Nishikawa. First, second, and third order finite-volume schemes for advection–diffusion. *Journal of Computational Physics*, 273:287–309, 2014.

[36] H. Nishikawa. Alternative formulations for first-, second-, and third-order hyperbolic Navier-Stokes schemes. In *Proc. of 22nd AIAA Computational Fluid Dynamics Conference*, AIAA Paper 2015-2451, Dallas, TX, 2015.

[37] H. Nishikawa and P. Roe. Third-order active-flux scheme for advection diffusion: Hyperbolic diffusion, boundary condition, and newton solver. *Computers & Fluids*, 125:71–81, 2016.

[38] Y. Nakashima, N. Watanabe, and H. Nishikawa. Hyperbolic Navier-Stokes solver for three-dimensional flows. In *54th AIAA Aerospace Sciences Meeting*, AIAA Paper 2016-1101, San Diego, CA, 2016.

[39] Y. Liu and H. Nishikawa. Third-order inviscid and second-order hyperbolic Navier-Stokes solvers for three-dimensional inviscid and viscous flows. In *46th AIAA Fluid Dynamics Conference*, AIAA Paper 2016-3969, Washington, D.C., 2016.

[40] A. Mazaheri and H. Nishikawa. Improved second-order hyperbolic residual-distribution scheme and its extension to third-order on arbitrary triangular grids. *Journal of Computational Physics*, 300:455–491, July 2015.

[41] G. Montecinos and E. Toro. Reformulations for general advection–diffusion–reaction equations and locally implicit ader schemes. *Journal of Computational Physics*, 275:415–442, 2014.

[42] G. Montecinos, L. Müller, and E. Toro. Hyperbolic reformulation of a 1d viscoelastic blood flow model and ader finite volume schemes. *Journal of Computational Physics*, 266:101–123, 2014.

[43] E. Toro and G. Montecinos. Advection-diffusion-reaction equations: hyperbolization and high-order ader discretizations. *SIAM Journal on Scientific Computing*, 36(5):A2423–A2457, 2014.

[44] E. Lee, H. Ahn, and H. Luo. Cell-centered high-order hyperbolic finite volume method for diffusion equation on unstructured grids. *Journal of Computational Physics*, 355:464–491, 2018.

[45] A. Mazaheri and H. Nishikawa. Efficient high-order discontinuous Galerkin schemes with first-order hyperbolic advection-diffusion system approach. *Journal of Computational Physics*, 321:729–754, September 2016.

[46] y. Liu and H. Nishikawa. Third-order inviscid and second-order hyperbolic navier-stokes solvers for three-dimensional unsteady inviscid and viscous flows. In *55th AIAA Aerospace Sciences Meeting*, page 0738, 2017.

[47] H. Bijl, M. Carpenter, V. Vatsa, and C. Kennedy. Implicit time integration schemes for the unsteady compressible navier–stokes equations: laminar flow. *Journal of Computational Physics*, 179(1):313–329, 2002.

[48] J. Cheng, T. Liu, and H. Luo. A hybrid reconstructed discontinuous galerkin method for compressible flows on arbitrary grids. *Computers & Fluids*, 139:68–79, 2016.

[49] Q. Wang, Y. Ren, J. Pan, and W. Li. Compact high order finite volume method on unstructured grids iii: Variational reconstruction. *Journal of Computational Physics*, 337:1–26, 2017.

ACCEPTED MANUSCRIPT

[50] 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.

[51] Y. Liu and H. Nishikawa. Third-order edge-based hyperbolic Navier-Stokes scheme for three-dimensional viscous flows. In *23rd AIAA Computational Fluid Dynamics Conference*, AIAA Paper 2017-3443, Denver, Colorado, 2017.

[52] H. Nishikawa and Y. Liu. Hyperbolic advection-diffusion schemes for high-Reynolds-number boundary-layer problems. *Journal of Computational Physics*, 352:23–51, 2018.

[53] J. Lou, X. Liu, H. Luo, and H. Nishikawa. Reconstructed discontinuous Galerkin methods for hyperbolic diffusion equations on unstructured grids. In *55th AIAA Aerospace Sciences Meeting*, AIAA Paper 2017-0310, Grapevine, Texas, 2017.

[54] 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.

ACCEPTED MANUSCRIPT