Journal of Computational Physics 321 (2016) 729–754

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Efficient high-order discontinuous Galerkin schemes with first-order hyperbolic advection–diffusion system approach

Alireza Mazaheri [a,*], Hiroaki Nishikawa [b]

[a] NASA Langley Research Center, Hampton, VA 23681, USA
[b] National Institute of Aerospace, Hampton, VA 23666, USA

ARTICLE INFO

Article history:
Received 24 March 2016
Received in revised form 23 May 2016
Accepted 6 June 2016
Available online 9 June 2016

Keywords:
High-order
Discontinuous Galerkin (DG)
DG with hyperbolic first-order system (DG-H)
Advection–diffusion
WENO
Interior Penalty (IP)

ABSTRACT

We propose arbitrary high-order discontinuous Galerkin (DG) schemes that are designed based on a first-order hyperbolic advection–diffusion formulation of the target governing equations. We present, in details, the efficient construction of the proposed high-order schemes (called DG-H), and show that these schemes have the same number of global degrees-of-freedom as comparable conventional high-order DG schemes, produce the same or higher order of accuracy solutions and solution gradients, are exact for exact polynomial functions, and do not need a second-derivative diffusion operator. We demonstrate that the constructed high-order schemes give excellent quality solution and solution gradients on irregular triangular elements. We also construct a Weighted Essentially Non-Oscillatory (WENO) limiter for the proposed DG-H schemes and apply it to discontinuous problems. We also make some accuracy comparisons with conventional DG and interior penalty schemes. A relative qualitative cost analysis is also reported, which indicates that the high-order schemes produce orders of magnitude more accurate results than the low-order schemes for a given CPU time. Furthermore, we show that the proposed DG-H schemes are nearly as efficient as the DG and Interior-Penalty (IP) schemes as these schemes produce results that are relatively at the same error level for approximately a similar CPU time.

Published by Elsevier Inc.

1. Introduction

Designing schemes based on a reformulation of the target governing equations as a first-order hyperbolic advection–diffusion system [1–3] have proven to possess some key features, such as consistent advection and diffusion discretization, elimination of the second-derivative diffusion operator, and $O(1/h)$ speed-up in the number of linear relaxations. The schemes that are designed with the first-order hyperbolic advection–diffusion system have shown [4–8] to produce the same high-order solutions for both the primal variables and their gradients on irregular grids. We have also shown [7,9] that one can design high-order schemes to obtain oscillation-free solution gradients on arbitrary triangular elements. The price to pay for these high-order schemes is the increase of the degrees-of-freedom (DoF) compared to the conventional schemes, that are designed based on the target governing equations. (Note that, throughout this paper, we refer to schemes that are not based on hyperbolic first-order system formulations, as *conventional*.) For example, the DoF for the Navier–Stokes equations jumps, after being reformulated as a first-order hyperbolic advection–diffusion system, from 5 to either 14, 17, or 20 [10], depending on the number of extra gradient variables created with the reformulation strategy. Even though,

* Corresponding author. Tel.: +1 757 864 7013.
E-mail addresses: alireza.mazaheri@nasa.gov (A. Mazaheri), hiro@nianet.org (H. Nishikawa).

http://dx.doi.org/10.1016/j.jcp.2016.06.006
0021-9991/Published by Elsevier Inc.

the quality of these schemes (higher-order in the advective term and the solution gradients and $O(1/h)$ speed-up) may outweigh the additional expense per iteration, much work needs to be done to assess the quality of the solutions per computational time for various problems and across different applications. The recent development of high-order Residual-Distribution (RD), aka fluctuation-splitting, schemes designed based on the first-order hyperbolic advection–diffusion system [4,7] provide good solution gradients on fully irregular triangular elements, while taking the full advantage of the compactness in the RD method. In this method, one can design a Newton solver, with which machine-zero convergence can be obtained within a few residual evaluations. Combining the higher-order gradients with elimination of the extra expense of evaluating the second-derivative diffusion operator in a compact RD stencil may one day be proved to be more economical (for some applications), even with the extra DoF, than the conventional RD schemes.

In some applications, such as large eddy simulations, compact schemes that are of higher order than three are also desired and sometimes necessary [11]. In our previous efforts, we could design up to third-order multi-dimensional Finite-Volume (FV) and RD schemes without compromising the compactness of the stencil. Constructing fourth- and higher-order multi-dimensional FV and RD schemes that are compact appears to be very difficult. This drawback applies to both the conventional FV and RD schemes, as well as those that are designed so far with the first-order hyperbolic advection–diffusion system approach.

In this work, we focus on a discontinuous Galerkin (DG) method, and propose an efficient procedure to construct arbitrary high-order DG schemes for the first-order hyperbolic advection–diffusion system formulations that have the same number of global DoF as that of a conventional DG scheme for a comparable level of accuracy, do not need a second-derivative diffusion operator, and yet give the same high-order accurate solution and solution gradients on irregular elements for general advection–diffusion problems. We refer to these proposed schemes as DG-H schemes. We choose the DG method, because compact arbitrary high-order schemes can be constructed systematically, and significant advances have been made in the development of high-order DG schemes, and as a result, various classes of DG schemes have been developed for both hyperbolic and elliptic problems. These include the method of Bassi and Rebay [12,13], Interior Penalty (IP) methods [14–21], Local DG (LDG) [22–27], Compact DG (CDG) [28,29], Direct DG (DDG) [30,31], Hybridizable DG (HDG) [32–36], and Recovery-DG [37–40] schemes. Our proposed DG-H scheme is not an addition to these powerful classes of the DG schemes, but rather we are simply interested in keeping the benefits of the first-order hyperbolic advection–diffusion system by replacing the second-derivative diffusion operator with a hyperbolic diffusion operator, and constructing efficient arbitrary high-order schemes without the additional DoF resulted by reformulating the target governing equation as a first-order hyperbolic advection–diffusion system, thus obtaining the same DoF over a conventional DG scheme for a comparable level of accuracy. Applying DG to hyperbolic problems has been very successful since it was first introduced in 1973 by Reed and Hill [41] for steady neutron transport equations, and independently in 1977 by Van Leer [42] for time-accurate advection equations. Since then, Cockburn, Shu, and their collaborators have carried out significant developments in the DG methodology for nonlinear hyperbolic conservation laws [43–45,24]. These advances are beneficial and can be carried over to our proposed DG-H schemes, because our scheme is based on a reformulation of the target governing equation as a first-order hyperbolic advection–diffusion system.

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

$$\partial_t u + \partial_x f + \partial_y g = \partial_x(\nu \partial_x u) + \partial_y(\nu \partial_y u) + \tilde{s}(x, y, u), \tag{1}$$

where the diffusion coefficient is $\nu = \nu(u)$, and the advection speeds in the x- and y-direction are defined as $a(u) = \partial f/\partial u$ and $b(u) = \partial g/\partial u$, respectively. Following a similar preconditioned formulation as in Ref. [46], we construct a hyperbolic system as

$$\partial_\tau u + \partial_x f + \partial_y g = \partial_x(\nu p) + \partial_y(\nu q) + s, \tag{2}$$

$$T_r \partial_\tau p = \partial_x u - p, \tag{3}$$

$$T_r \partial_\tau q = \partial_y u - q, \tag{4}$$

where $T_r = L^2/\nu$, $L = 1/2\pi$, τ is the pseudo-time, and s includes $\tilde{s}$ (arising from e.g., chemical reactions or turbulence), and the temporal terms (for time-dependent problems). For example, the source term $\tilde{s}$, for a second-order backward-differencing-formulation for the time discretization, takes the following form

$$s = \tilde{s} - \frac{3}{2}\frac{u}{\Delta t} + \frac{u_0}{\Delta t},$$

where Δt is the physical time step, and u_0 denotes the solution evaluated at a previous physical time iteration (see Refs. [3,4] for more details on time-dependent hyperbolic first-order system simulations). The variables p and q are, in the pseudo steady state, equivalent to the gradients of the primal variable u in the x and y directions, respectively. We remark that this does not imply that the formulation is only valid for steady-state problems as we have shown time-dependent calculations in Refs. [3,4,47]. We also note that the relaxation parameter T_r is a free parameter and can take any value, but we use the optimal value (i.e., L^2/ν) recommended in Ref. [1] to enhance the convergence.

We write the system as a preconditioned conservative equation in a vector form with the preconditioning matrix $\mathbf{P}$ as

$$\mathbf{P}^{-1}\frac{\partial \mathbf{U}}{\partial \tau} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = \mathbf{Q}, \tag{5}$$

where

$$\mathbf{P}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & T_r & 0 \\ 0 & 0 & T_r \end{bmatrix}, \quad \mathbf{U} = \mathbf{U}^a + \mathbf{U}^d = \begin{bmatrix} u \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ p \\ q \end{bmatrix}, \tag{6}$$

$$\mathbf{F} = \begin{bmatrix} f - \nu p \\ -u \\ 0 \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} g - \nu q \\ 0 \\ -u \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} s \\ -p \\ -q \end{bmatrix}, \tag{7}$$

where superscripts a and d refer to the advection and the hyperbolic diffusion terms. If we apply the DG method directly (naively) to this first-order hyperbolic advection–diffusion system, we end up with three times more DoF than the conventional DG schemes that are designed based on the target equation, Eq. (1). For the Navier–Stokes equations in three-dimension, for example, we will have $14/5 = 2.8$, $17/5 = 3.4$, or $20/5 = 4$ times more DoF compared to the conventional DG schemes that are not constructed from the hyperbolic Navier–Stokes equations [10,48].

In this paper, we propose a strategy in constructing arbitrary high-order DG schemes with the first-order hyperbolic advection–diffusion system approach that maintains the same DoF obtained with comparable high-order DG schemes. With this efficient construction, we present new DG schemes with the same property of the conventional schemes plus additional benefits arising from the first-order hyperbolic advection–diffusion formulation (e.g., $O(1/h)$ speed up in linear relaxations with grid refinement), equal order of accuracy for primal and gradients quantities, and an order higher primal accuracy in the inviscid region. The proposed approach can also be directly applied to, for example, the Flux Reconstruction (FR) schemes [49–51]; the construction of this approach will be reported in the future.

The rest of the paper is organized as follows. In Section 2, we review the basics of the DG method, which is used as a basis for construction of our proposed DG-H schemes. The details of formulating high-order DG schemes with the first-order hyperbolic system approach, including our proposed DoF reduction technique, are given in Section 3. Weighted Essentially Non-Oscillatory (WENO) limiter procedure for the proposed DG-H schemes is provided in Section 4. Numerical experiments are conducted in Section 5 followed by some concluding remarks in Section 6.

2. Discontinuous Galerkin (DG) discretization

Consider a triangulation $\mathcal{T}_h$ of the domain Ω in the finite element space of discontinuous functions $V_h = \{\upsilon \in L^\infty(\Omega) : \upsilon|_E \in P^k, \forall E \in \mathcal{T}_h\}$, where P^k denotes the set of polynomials of degree up to k on every irregular triangular element E. Numerical solution $\mathbf{U}_h$ is defined within the element E as a polynomial degree of order k, P^k, for the variable vector $\mathbf{U}$ as:

$$\mathbf{U}_h(x, y, \tau) = \sum_k \varphi_k(x, y)\, \mathbf{U}_k(\tau), \tag{8}$$

where $\varphi_k(x, y) \in P^k$ is the k-th polynomial basis function, and $\mathbf{U}_k$ is the vector of unknowns corresponding to the k-th polynomial basis. For the purpose of our discussion, it is convenient to express the numerical solution in the following form

$$\mathbf{U}_h = \mathbf{B}_k \mathbf{u}_k, \tag{9}$$

where

$$\mathbf{B}_k = \begin{bmatrix} \overbrace{\underbrace{\underbrace{\varphi_1 \mathbf{I}}_{P^0}, \quad \varphi_x \mathbf{I}, \varphi_y \mathbf{I}}_{P^1}, \quad \varphi_{xx}\mathbf{I}, \varphi_{xy}\mathbf{I}, \varphi_{yy}\mathbf{I}}^{P^2}, \quad \varphi_{xxx}\mathbf{I}, \varphi_{xxy}\mathbf{I}, \varphi_{xyy}\mathbf{I}, \varphi_{yyy}\mathbf{I}, \ldots}^{P^3} \end{bmatrix}^{P^k}, \tag{10}$$

$$\mathbf{u}_k = \Bigg[\overbrace{\underbrace{\overline{\mathbf{U}}^T}_{P^0},\ \partial_x \mathbf{U}^T,\ \partial_y \mathbf{U}^T}^{P^1},\ \overbrace{\partial_{xx}\mathbf{U}^T,\ \partial_{xy}\mathbf{U}^T,\ \partial_{yy}\mathbf{U}^T}^{P^2},\ \overbrace{\partial_{xxx}\mathbf{U}^T,\ \partial_{xxy}\mathbf{U}^T,\ \partial_{xyy}\mathbf{U}^T,\ \partial_{yyy}\mathbf{U}^T}^{P^3},\ \overbrace{\dots}^{P^k} \Bigg]^T, \tag{11}$$

where, $\mathbf{I}$ is an $N \times N$ identity matrix, N is the number of equations (e.g., for the hyperbolic advection–diffusion system studied here, $N = 3$), and the superscript T denotes the transpose. Here, we followed Luo et al. [52], and defined the polynomial based on the local Taylor series expansions with the cell averaged value (shown with over-bar) and its derivatives at the cell centroid.

The local Taylor basis functions, $\varphi_m(x, y) \in P^k$, for triangular elements that are used in this study are:

$$\varphi_1 = 1, \qquad \varphi_x = x - x_c, \qquad \varphi_y = y - y_c,$$

$$\varphi_{xx} = \frac{1}{2}\varphi_x^2 - \frac{1}{2A_E}\int_E \varphi_x^2\, d\Omega, \qquad \varphi_{xy} = \varphi_x\varphi_y - \frac{1}{A_E}\int_E \varphi_x\varphi_y\, d\Omega,$$

$$\varphi_{yy} = \frac{1}{2}\varphi_y^2 - \frac{1}{2A_E}\int_E \varphi_y^2\, d\Omega, \qquad \varphi_{xxx} = \frac{1}{6}\varphi_x^3 - \frac{1}{6A_E}\int_E \varphi_x^3\, d\Omega,$$

$$\varphi_{xxy} = \frac{1}{2}\varphi_x^2\varphi_y - \frac{1}{2A_E}\int_E \varphi_x^2\varphi_y\, d\Omega, \qquad \varphi_{xyy} = \frac{1}{2}\varphi_x\varphi_y^2 - \frac{1}{2A_E}\int_E \varphi_x\varphi_y^2\, d\Omega,$$

$$\varphi_{yyy} = \frac{1}{6}\varphi_y^3 - \frac{1}{6A_E}\int_E \varphi_y^3\, d\Omega, \qquad \varphi_{xxxx} = \frac{1}{24}\varphi_x^4 - \frac{1}{24A_E}\int_E \varphi_x^4\, d\Omega,$$

$$\varphi_{xxxy} = \frac{1}{6}\varphi_x^3\varphi_y - \frac{1}{6A_E}\int_E \varphi_x^3\varphi_y\, d\Omega, \qquad \varphi_{xxyy} = \frac{1}{4}\varphi_x^2\varphi_y^2 - \frac{1}{4A_E}\int_E \varphi_x^2\varphi_y^2\, d\Omega,$$

$$\varphi_{xyyy} = \frac{1}{6}\varphi_x\varphi_y^3 - \frac{1}{6A_E}\int_E \varphi_x\varphi_y^3\, d\Omega, \qquad \varphi_{yyyy} = \frac{1}{24}\varphi_y^4 - \frac{1}{24A_E}\int_E \varphi_y^4\, d\Omega,$$

$$\varphi_{xxxxx} = \frac{1}{120}\varphi_x^5 - \frac{1}{120A_E}\int_E \varphi_x^5\, d\Omega, \qquad \varphi_{xxxxy} = \frac{1}{24}\varphi_x^4\varphi_y - \frac{1}{24A_E}\int_E \varphi_x^4\varphi_y\, d\Omega,$$

$$\varphi_{xxxyy} = \frac{1}{12}\varphi_x^3\varphi_y^2 - \frac{1}{12A_E}\int_E \varphi_x^3\varphi_y^2\, d\Omega, \qquad \varphi_{xxyyy} = \frac{1}{12}\varphi_x^2\varphi_y^3 - \frac{1}{12A_E}\int_E \varphi_x^2\varphi_y^3\, d\Omega,$$

$$\varphi_{xyyyy} = \frac{1}{24}\varphi_x\varphi_y^4 - \frac{1}{24A_E}\int_E \varphi_x\varphi_y^4\, d\Omega, \qquad \varphi_{yyyyy} = \frac{1}{120}\varphi_y^5 - \frac{1}{120A_E}\int_E \varphi_y^5\, d\Omega,$$

where A_E is the area of the element E. The volume integrals can be evaluated either exactly (analytically) or using quadrature rules that are exact for polynomial degrees of $2k$. In this study, we use the quadrature rules (see also Sec. 5).

The DG method for solving Eq. (5) is defined by finding a unique solution $\mathbf{U}_h \in V_h$ such that for all the test functions $\upsilon \in V_h$, we have

$$\begin{aligned}
\mathbf{P}^{-1}\int_E \upsilon\partial_\tau \mathbf{U}_h\, d\Omega &= -\int_E \upsilon\left(\partial_x\mathbf{F} + \partial_y\mathbf{G}\right) d\Omega + \int_E \upsilon\mathbf{Q}\, d\Omega, \\
&= -\oint_{\partial\Omega} \upsilon\mathbf{H}\cdot\hat{\mathbf{n}}\, d\Gamma + \int_E \nabla\upsilon\cdot\mathbf{H}\, d\Omega + \int_E \upsilon\mathbf{Q}\, d\Omega, \\
&= -\sum_{e\in\partial\Omega}\int_e \upsilon\mathbf{H}\cdot\hat{\mathbf{n}}\, d\Gamma + \int_E \nabla\upsilon\cdot\mathbf{H}\, d\Omega + \int_E \upsilon\mathbf{Q}\, d\Omega,
\end{aligned}$$

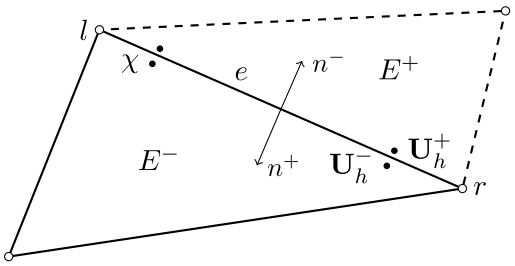


Fig. 1. Schematic of elements E^+ and E^- sharing the edge e along with the corresponding unit outward normals and the bi-valued $\mathbf{U}_h$ on either side of the interface boundary at the quadrature points, χ, shown with black dots (number of quadrature points varies depending on the polynomial order used). Negative superscript refers to local interior element while positive superscript denotes neighboring (exterior) element.

$$\approx -\sum_{e\in\partial\Omega}\sum_{\chi_l} w_l\, \upsilon_l \hat{\mathbf{H}}_l\cdot\hat{\mathbf{n}}\,\Delta s_e + \sum_{\chi_m} w_m\left[\nabla\upsilon_m\cdot\mathbf{H}_m + \upsilon_m\mathbf{Q}_n\right]A_E, \tag{12}$$

where we performed the integration by part once. Here, the flux and source terms are a function of $\mathbf{U}_h$, e refers to an element edge, $\partial\Omega$ denotes the element boundaries, χ_l and χ_m are the quadrature points for the edge and the interior (aka volume) integrals, respectively, w is the quadrature weighting function, Δs_e is the length of each element edge, A_E is the element area, $\mathbf{H} = (\mathbf{F}, \mathbf{G})$, and $\hat{\mathbf{H}}$ is the numerical flux at the interface boundary between the two elements (see Fig. 1), which we will define in Sec. 2.1.

Employing the numerical solution as expressed in Eq. (9), we arrive at the pseudo-time evolution equations for the vector of unknown polynomial coefficients in the element E as

$$\begin{aligned}\mathbf{M}_k\partial_\tau\mathbf{u}_k &= -\int_E \mathbf{B}_k^T\mathbf{P}\left(\partial_x\mathbf{F}+\partial_y\mathbf{G}\right)d\Omega + \int_E \mathbf{B}_k^T\mathbf{P}\mathbf{Q}d\Omega,\\ &= -\oint_{\partial\Omega}\mathbf{B}_k^T\mathbf{P}\mathbf{H}\cdot\hat{\mathbf{n}}\,d\Gamma + \int_E \nabla\mathbf{B}_k^T:\mathbf{P}\mathbf{H}\,d\Omega + \int_E \mathbf{B}_k^T\mathbf{P}\mathbf{Q}d\Omega,\end{aligned} \tag{13}$$

where the mass matrix is defined as

$$\mathbf{M}_k = \int_E \mathbf{B}_k^T\mathbf{B}_k\,d\Omega. \tag{14}$$

Remark on pseudo-time integration: In the time-dependent hyperbolic advection–diffusion formulation, the physical time, t, as shown earlier is included in the source term (see Refs. [3,4,47] for more details), and integrated the same way as the interior integrals with the same quadrature rules used for other source terms. We are then left with the pseudo-time integration on the left hand side, which we evaluate it with a volume quadrature rule that is exact for polynomial order of degree $2k$. This results in an evaluation of a mass matrix as given in Eq. (14).

2.1. Numerical flux

The flux at the element interface, $\hat{\mathbf{H}}$, is bi-valued (i.e., discontinuous) and therefore, needs to be approximated. The numerical flux also plays a significant role in coupling the DoF of each element with the neighboring elements. Here, we use the local Lax–Friedrichs (LLxF) scheme, which we briefly describe them next in the framework of the first-order hyperbolic advection–diffusion system formulation.

2.2. Local Lax–Friedrichs (LLxF) flux

We define the LLxF flux for the hyperbolic advection–diffusion system as

$$\hat{\mathbf{H}} = \frac{1}{2}\left[\left[\mathbf{H}(\mathbf{U}^+)+\mathbf{H}(\mathbf{U}^-)\right] - \mathbf{P}^{-1}(\alpha^a+\alpha^d)(\mathbf{U}^{a^+}-\mathbf{U}^{a^-})\hat{\mathbf{n}} - \mathbf{P}^{-1}\alpha^d(\mathbf{U}^{d^+}-\mathbf{U}^{d^-})\hat{\mathbf{n}}\right] \tag{15}$$

where the negative and positive superscripts denote the local interior and its neighboring (exterior) elements, respectively, and $\hat{\mathbf{n}} = (\hat{n}_x, \hat{n}_y)$ is the outward unit normal vector to the edge (or face in 3D) of the interior element. Note that the numerical flux is given for an interior element and the numerical flux for the neighboring (exterior) element is simply $-\hat{\mathbf{H}}$, and therefore, the numerical fluxes are conservative. Here, we define the advection and the hyperbolic diffusion spectral radii α^a and α^d locally at each quadrature point as

$$\alpha^a = max(|\lambda^a(\hat{u})|), \quad \alpha^d = max(|\lambda^d(\hat{u})|), \quad \hat{u} = \frac{1}{2}(u^+ + u^-). \tag{16}$$

Separating the dissipation term into advective and hyperbolic diffusion terms allows us to directly apply the proposed schemes to more complex governing equations, such as Navier–Stokes, where the advective and the hyperbolic diffusive eigen-structures [10] are available.

3. Proposed DG schemes for the first-order hyperbolic system (DG-H)

In this section, we describe, in details, the construction of high-order cell-centered DG-H schemes for arbitrary polynomial orders on irregular triangular elements. We start the discussion by first introducing some notations.

We construct a DG method with the first-order hyperbolic advection–diffusion system approach, and a reduced number of polynomial coefficients, such that the numerical solution can be expressed as

$$\mathbf{U}_h = \mathbf{C}_k \mathbf{V}_k \tag{17}$$

where subscript k denotes the polynomial order approximation, $\mathbf{C}_k$ is a modified basis matrix, and $\mathbf{V}_k$ is a reduced vector of unknown polynomial coefficients. We will define the constructions of $\mathbf{C}_k$, $\mathbf{V}_k$, and their relations to the original basis functions, $\mathbf{B}_k$, and the vector of unknown polynomial coefficients, $\mathbf{u}_k$, which are given in Eq. (9), shortly in this section. At the end of this process, we apply the Galerkin discretization by multiplying the hyperbolic advection–diffusion system by the modified basis functions, $\mathbf{C}_k$, and perform an integration by part to arrive at

$$\mathbf{M}_k \partial_\tau \mathbf{V}_k = -\oint_{\partial\Omega} \mathbf{C}_k^T \mathbf{P}\mathbf{H}\cdot\hat{\mathbf{n}}\, d\Gamma + \int_E \nabla \mathbf{C}_k^T : \mathbf{P}\mathbf{H}\, d\Omega + \int_E \mathbf{C}_k^T \mathbf{P}\mathbf{Q}\, d\Omega, \tag{18}$$

where the mass matrix is defined as

$$\mathbf{M}_k = \int_E \mathbf{C}_k^T \mathbf{C}_k\, d\Omega. \tag{19}$$

Equation (18) describes the pseudo-time evolution equations for the reduced vector of unknown polynomial coefficients, $\mathbf{V}_k$, for the element E.

The modified basis functions, $\mathbf{C}_k$, and the reduced number of unknown polynomial coefficients, $\mathbf{V}_k$, are described next for polynomial orders of zero, one, two, and $k \geq 3$. In essence, we are seeking a new finite-element space based on the information available from the auxiliary variables, which are the gradients of the primal variable. A similar approach is used in defining a new finite element space, for example, for Maxwell [53] and ideal MHD [54,55] equations, where a local divergence-free polynomial vector is used to approximate the gradients of the magnetic field. This resulted in a reduced DoF by about a factor of two for two-dimensional problems. In our approach, we reduce the DoF by seeking a new finite element space by utilizing the information that is already available from the polynomial space of the auxiliary variables. Imposing divergence-free polynomial space for the gradient variables could further reduce the DoF of the proposed DG-H schemes; this is left for future investigations. For our discussion, we believe it is beneficial to explicitly spell out the proposed new finite element bases starting with the polynomial order of zero, which is done next followed by the polynomial orders of one, two and $k \geq 3$. We then give a simple step-by-step instruction to define a new finite element spaces for a general polynomial order of degree k.

3.1. DG-H scheme with a polynomial order of zero (P^0)

The polynomial order of zero, P^0, is an exceptional case and no reduction in the number of unknown polynomial coefficients can be performed. Therefore, the proposed DG-H(P^0) scheme is constructed straightforwardly by applying the Galerkin discretization to the hyperbolic advection–diffusion system as described here.

We start the process by first expressing the numerical solution as a polynomial degree of order zero (i.e., $k = 0$)

$$\mathbf{U}_h = \varphi_1 \mathbf{I} = \mathbf{B}_0\, \mathbf{u}_0, \tag{20}$$

where

$$\mathbf{B}_0 = \begin{bmatrix} \varphi_1 & 0 & 0 \\ 0 & \varphi_1 & 0 \\ 0 & 0 & \varphi_1 \end{bmatrix}, \quad \mathbf{u}_0 = \overline{\mathbf{U}} = \begin{bmatrix} \overline{u} \\ \overline{p} \\ \overline{q} \end{bmatrix}. \tag{21}$$

Therefore, we have one DoF for each variable for the total of three DoF per element:

$$u_h = \varphi_1\, \overline{u},$$
$$p_h = \varphi_1\, \overline{p},$$
$$q_h = \varphi_1\, \overline{q}.$$

For the P^0 polynomial, $\overline{u}$, $\overline{p}$, and $\overline{q}$ are the minimum required number of unknowns and thus, $\mathbf{B}_0 = \mathbf{C}_0$, $\mathbf{u}_0 = \mathbf{V}_0$, and the numerical solution can be expressed as

$$\mathbf{U}_h = \mathbf{B}_0\,\mathbf{u}_0 = \mathbf{C}_0\,\mathbf{V}_0. \tag{22}$$

We then directly apply Eq. (18) and solve for the three DoF per element. As we will show later, the constructed P^0 scheme is first-order for both solution and solution gradients on irregular elements, making this scheme ideal for consistent p-multigrid.

We can further upgrade the polynomial representation of u_h by one order with the available polynomial coefficients as

$$\begin{aligned} u_h &= \varphi_1\,\overline{u} + \varphi_x\,u_x + \varphi_y\,u_y, \\ p_h &= \varphi_1\,\overline{p}, \\ q_h &= \varphi_1\,\overline{q}, \end{aligned} \tag{23}$$

where u_x and u_y are the gradients of u at the cell centroid and therefore, we can evaluate them with the polynomial representations of p_h and q_h as

$$\begin{aligned} u_x &\leftarrow p_h(x_c, y_c) = \overline{p} \\ u_y &\leftarrow q_h(x_c, y_c) = \overline{q}. \end{aligned}$$

Thus, we arrive at the following upgraded polynomial

$$\begin{aligned} u_h &= \varphi_1\,\overline{u} + \varphi_x\,\overline{p} + \varphi_y\,\overline{q}, \\ p_h &= \varphi_1\,\overline{p}, \\ q_h &= \varphi_1\,\overline{q}. \end{aligned} \tag{24}$$

We can use the upgraded polynomial only in the flux evaluation and/or incorporate it, as well, in the modified basis functions matrix, $\mathbf{C}_0$, as given below:

$$\mathbf{C}_0 = \begin{bmatrix} \varphi_1 & \varphi_x & \varphi_y \\ 0 & \varphi_1 & 0 \\ 0 & 0 & \varphi_1 \end{bmatrix}. \tag{25}$$

The incorporation of the upgraded polynomial u_h in the $\mathbf{C}_0$ matrix is optional and will not affect the order of accuracy of the scheme for diffusion problems. This upgrade, which does not increase the DoF, makes the DG-H(P^0) scheme exact for exact linear solution, and increases the order of accuracy of the primal variable to second-order in the advection-limit ($\nu \to 0$), while the solution gradients remain at first-order.

We remark that employing polynomial approximations of the auxiliary variables p and q for the evaluation of the first derivatives of the primal variable u does not imply that the constructed scheme is only applicable to steady-state calculations. We realize that the evolution equations are in pseudo-time, and as we mentioned earlier the source term of our hyperbolic first-order system contains the physical-time discretization. It is also important to note that we always seek a pseudo-steady state solution in both steady-state and time-dependent calculations. Thus, utilizing, for example, p_h for approximating u_x is perfectly valid.

3.2. DG-H scheme with a linear polynomial (P^1)

We start the construction of the P^1 scheme by expressing the numerical solution with a linear polynomial approximation

$$\begin{aligned} u_h &= \varphi_1\overline{u} + \varphi_x u_x + \varphi_y u_y, \\ p_h &= \varphi_1\overline{p} + \varphi_x p_x + \varphi_y p_y, \\ q_h &= \varphi_1\overline{q} + \varphi_x q_x + \varphi_y q_y, \end{aligned}$$

which can be written in a vector form as

$$\mathbf{U}_h = \mathbf{B}_1\,\mathbf{u}_1, \tag{26}$$

where

$$\mathbf{B}_1 = \left[\varphi_1\mathbf{I},\ \varphi_x\mathbf{I},\ \varphi_y\mathbf{I}\right] = \begin{bmatrix} \varphi_1 & 0 & 0 & \varphi_x & 0 & 0 & \varphi_y & 0 & 0 \\ 0 & \varphi_1 & 0 & 0 & \varphi_x & 0 & 0 & \varphi_y & 0 \\ 0 & 0 & \varphi_1 & 0 & 0 & \varphi_x & 0 & 0 & \varphi_y \end{bmatrix}, \tag{27}$$

$$\mathbf{u}_1 = \left[\overline{\mathbf{U}},\ \mathbf{U}_x,\ \mathbf{U}_y\right]^T = \left[\overline{u},\ \overline{p},\ \overline{q},\ u_x,\ p_x,\ q_x,\ u_y,\ p_y,\ q_y\right]^T. \tag{28}$$

We now introduce a unified notation for the q_x and p_y as

$$\mathrm{v}_{xy} \equiv q_x = p_y, \tag{29}$$

and expressed the cell-centered u_x and u_y values as

$$u_x \leftarrow p_h(x_c, y_c) = \overline{p} + \cancelto{0}{\varphi_x^c}\, p_x + \cancelto{0}{\varphi_y^c}\, p_y = \overline{p},$$
$$u_y \leftarrow q_h(x_c, y_c) = \overline{q} + \cancelto{0}{\varphi_x^c}\, q_x + \cancelto{0}{\varphi_y^c}\, q_y = \overline{q},$$

where sub/superscript c denote evaluation at the cell centroid. That is, the polynomial approximation can be written in a modified form as

$$u_h = \varphi_1 \overline{u} + \varphi_x \overline{p} + \varphi_y \overline{q},$$
$$p_h = \varphi_1 \overline{p} + \varphi_x p_x + \varphi_y \mathrm{v}_{xy},$$
$$q_h = \varphi_1 \overline{q} + \varphi_x \mathrm{v}_{xy} + \varphi_y q_y,$$

or in a vector form as

$$\mathbf{U}_h = \mathbf{B}_1\, \tilde{\mathbf{u}}_1 = \mathbf{B}_1 \mathbf{Z}_1 \mathbf{V}_1 = \tilde{\mathbf{C}}_1 \mathbf{V}_1, \tag{30}$$

where

$$\tilde{\mathbf{u}}_1 = \left[\overline{u},\ \overline{p},\ \overline{q},\ \overline{p},\ p_x,\ \mathrm{v}_{xy},\ \overline{q},\ \mathrm{v}_{xy},\ q_y\right]^T, \tag{31}$$
$$\mathbf{V}_1 = \left[\overline{u},\ \overline{p},\ \overline{q},\ p_x,\ \mathrm{v}_{xy},\ q_y\right]^T, \tag{32}$$

$$\mathbf{Z}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{33}$$

Note that $\mathbf{u}_1$ represents the unknown polynomial coefficients resulted from the direct DG construction, while $\mathbf{V}_1$ is a vector of reduced unknown polynomial coefficients.

Similar to the proposed P^0 scheme, we can improve the polynomial order of u_h by one degree using the available independent coefficients and arrive at

$$u_h = \varphi_1\, \overline{u} + \varphi_x\, \overline{p} + \varphi_y\, \overline{q} + \varphi_{xx}\, p_x + \varphi_{xy}\, \mathrm{v}_{xy} + \varphi_{yy}\, q_y, \tag{34}$$
$$p_h = \varphi_1\, \overline{p} + \varphi_x\, p_x + \varphi_y\, \mathrm{v}_{xy}, \tag{35}$$
$$q_h = \varphi_1\, \overline{q} + \varphi_x\, \mathrm{v}_{xy} + \varphi_y\, q_y, \tag{36}$$

where the quadratic terms of the polynomial representation of u_h can either be used only for the flux evaluations or also be incorporated into the basis functions to arrive at the following polynomial approximations

$$\mathbf{U}_h = \tilde{\mathbf{B}}_1\, \tilde{\mathbf{u}}_1 = \tilde{\mathbf{B}}_1 \mathbf{Z}_1 \mathbf{V}_1 = \mathbf{C}_1 \mathbf{V}_1, \tag{37}$$

where

$$\tilde{\mathbf{B}}_1 = \begin{bmatrix} \varphi_1 & 0 & 0 & \varphi_x & \varphi_{xx} & \omega_1\varphi_{xy} & \varphi_y & (1-\omega_1)\varphi_{xy} & \varphi_{yy} \\ 0 & \varphi_1 & 0 & 0 & \varphi_x & 0 & 0 & \varphi_y & 0 \\ 0 & 0 & \varphi_1 & 0 & 0 & \varphi_x & 0 & 0 & \varphi_y \end{bmatrix}. \tag{38}$$

The presence of the ω_1 constant, which can take any value, is the result of the equality given in Eq. (29), and has no effect in the final answer because it will be disappeared in the modified basis functions as shown bellow:

$$\mathbf{C}_1 = \tilde{\mathbf{B}}_1 \mathbf{Z}_1 = \begin{bmatrix} \varphi_1 & \varphi_x & \varphi_y & \varphi_{xx} & \omega_1\varphi_{xy} + (1-\omega_1)\varphi_{xy} & \varphi_{yy} \\ 0 & \varphi_1 & 0 & \varphi_x & \varphi_y & 0 \\ 0 & 0 & \varphi_1 & 0 & \varphi_x & \varphi_y \end{bmatrix}$$
$$= \begin{bmatrix} \varphi_1 & \varphi_x & \varphi_y & \varphi_{xx} & \varphi_{xy} & \varphi_{yy} \\ 0 & \varphi_1 & 0 & \varphi_x & \varphi_y & 0 \\ 0 & 0 & \varphi_1 & 0 & \varphi_x & \varphi_y \end{bmatrix} = \mathbf{B}_1 \mathbf{Z}_1 + \mathbf{r}_u \mathbf{c}_1^T, \tag{39}$$

where $\mathbf{r}_u$ is a vector indicating which variable is given the extra terms and $\mathbf{c}_1^T$ is a vector containing the extra basis functions:

$$\mathbf{r}_u = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{c}_1^T = [0, 0, 0, \varphi_{xx}, \varphi_{xy}, \varphi_{yy}].$$

With the introduction of the modified basis function, $\mathbf{C}_1$, and the reduced vector of unknown polynomial coefficients, $\mathbf{V}_1$, we simply apply the Galerkin discretization as given in Eq. (18).

The proposed DG-H(P^1) scheme is second-order for both solution and solution gradients. Furthermore, the proposed P^1 scheme is exact for exact quadratic solution, and in the advection-limit, the order of accuracy of the solution u is third-order, while the gradients remain at second-order.

3.3. DG-H scheme with a quadratic polynomial (P^2)

The procedure in constructing the P^2 scheme starts with a quadratic polynomial approximation of the numerical solution as

$$u_h = \varphi_1 \overline{u} + \varphi_x u_x + \varphi_y u_y + \varphi_{xx} u_{xx} + \varphi_{xy} u_{xy} + \varphi_{yy} u_{yy}$$
$$p_h = \varphi_1 \overline{p} + \varphi_x p_x + \varphi_y p_y + \varphi_{xx} p_{xx} + \varphi_{xy} p_{xy} + \varphi_{yy} p_{yy},$$
$$q_h = \varphi_1 \overline{q} + \varphi_x q_x + \varphi_y q_y + \varphi_{xx} q_{xx} + \varphi_{xy} q_{xy} + \varphi_{yy} q_{yy},$$

which can be written in a vector form as

$$\mathbf{U}_h = \mathbf{B}_2 \, \mathbf{u}_2, \tag{40}$$

where

$$\mathbf{B}_2 = \left[\varphi_1 \mathbf{I},\ \varphi_x \mathbf{I},\ \varphi_y \mathbf{I},\ \varphi_{xx} \mathbf{I},\ \varphi_{xy} \mathbf{I},\ \varphi_{yy} \mathbf{I}\right],$$
$$= \begin{bmatrix} \varphi_1 & 0 & 0 & \varphi_x & 0 & 0 & \varphi_y & 0 & 0 & \varphi_{xx} & 0 & 0 & \varphi_{xy} & 0 & 0 & \varphi_{yy} & 0 & 0 \\ 0 & \varphi_1 & 0 & 0 & \varphi_x & 0 & 0 & \varphi_y & 0 & 0 & \varphi_{xx} & 0 & 0 & \varphi_{xy} & 0 & 0 & \varphi_{yy} & 0 \\ 0 & 0 & \varphi_1 & 0 & 0 & \varphi_x & 0 & 0 & \varphi_y & 0 & 0 & \varphi_{xx} & 0 & 0 & \varphi_{xy} & 0 & 0 & \varphi_{yy} \end{bmatrix}, \tag{41}$$

$$\mathbf{u}_2 = \left[\overline{\mathbf{U}},\ \mathbf{U}_x,\ \mathbf{U}_y,\ \mathbf{U}_{xx},\ \mathbf{U}_{xy},\ \mathbf{U}_{yy}\right]^T,$$
$$= \left[\overline{u},\ \overline{p},\ \overline{q},\ u_x,\ p_x,\ q_x,\ u_y,\ p_y,\ q_y,\ u_{xx},\ p_{xx},\ q_{xx},\ u_{xy},\ p_{xy},\ q_{xy},\ u_{yy},\ p_{yy},\ q_{yy}\right]^T. \tag{42}$$

Similar to the proposed P^1 scheme, we can unify some of the polynomial coefficients as

$$u_{xx} \equiv p_x,$$
$$\mathrm{v}_{xy} \equiv q_x = p_y,$$
$$\mathrm{v}_{xxy} \equiv q_{xx} = p_{xy},$$
$$\mathrm{v}_{xyy} \equiv q_{xy} = p_{yy},$$
$$u_{yy} \equiv q_y,$$

and replace the cell centered u_x and u_y with the following expressions

$$u_x \leftarrow p_h(x_c, y_c) = \overline{p} + \cancelto{0}{\varphi_x^c}\, p_x + \cancelto{0}{\varphi_y^c}\, p_y + \varphi_{xx}^c p_{xx} + \varphi_{xy}^c \mathrm{v}_{xxy} + \varphi_{yy}^c \mathrm{v}_{xyy} = \overline{p} + \varphi_{xx}^c p_{xx} + \varphi_{xy}^c \mathrm{v}_{xxy} + \varphi_{yy}^c \mathrm{v}_{xyy},$$
$$u_y \leftarrow q_h(x_c, y_c) = \overline{q} + \cancelto{0}{\varphi_x^c}\, q_x + \cancelto{0}{\varphi_y^c}\, q_y + \varphi_{xx}^c \mathrm{v}_{xxy} + \varphi_{xy}^c \mathrm{v}_{xyy} + \varphi_{yy}^c q_{yy} = \overline{q} + \varphi_{xx}^c \mathrm{v}_{xxy} + \varphi_{xy}^c \mathrm{v}_{xyy} + \varphi_{yy}^c q_{yy},$$

where, again, sub/superscript c denotes the evaluation of the basis functions at the cell centroid. This leads to the following quadratic polynomial approximation for the numerical solution:

$$u_h = \varphi_1 \overline{u} + \varphi_x(\overline{p} + \varphi^c_{xx} p_{xx} + \varphi^c_{xy} \mathrm{v}_{xxy} + \varphi^c_{yy} \mathrm{v}_{xyy}) + \varphi_y(\overline{q} + \varphi^c_{xx} \mathrm{v}_{xxy} + \varphi^c_{xy} \mathrm{v}_{xyy} + \varphi^c_{yy} q_{yy})$$
$$+ \varphi_{xx} p_x + \varphi_{xy} \mathrm{v}_{xy} + \varphi_{yy} q_y,$$
$$p_h = \varphi_1 \overline{p} + \varphi_x p_x + \varphi_y \mathrm{v}_{xy} + \varphi_{xx} p_{xx} + \varphi_{xy} \mathrm{v}_{xxy} + \varphi_{yy} \mathrm{v}_{xyy},$$
$$q_h = \varphi_1 \overline{q} + \varphi_x \mathrm{v}_{xy} + \varphi_y q_y + \varphi_{xx} \mathrm{v}_{xxy} + \varphi_{xy} \mathrm{v}_{xyy} + \varphi_{yy} q_{yy}.$$

Here, it is important to closely examine the polynomial representation of the primal variable, u_h, which we rewrite it as

$$u_h = \varphi_1 \overline{u} + \varphi_x \overline{p} + \varphi_y \overline{q} + \varphi_{xx} p_x + \varphi_{xy} \mathrm{v}_{xy} + \varphi_{yy} q_y$$
$$+ \varphi_x \varphi^c_{xx} p_{xx} + \varphi_x \varphi^c_{xy} \mathrm{v}_{xxy} + \varphi_x \varphi^c_{yy} \mathrm{v}_{xyy} + \varphi_y \varphi^c_{xx} \mathrm{v}_{xxy} + \varphi_y \varphi^c_{xy} \mathrm{v}_{xyy} + \varphi_y \varphi^c_{yy} q_{yy}.$$

We immediately note that this polynomial is no longer a valid polynomial approximation, because it corresponds to the following basis functions

$$\varphi_1,\ \varphi_x,\ \varphi_y,\ \varphi_{xx},\ \varphi_{xy},\ \varphi_{yy},\ \varphi_x\varphi^c_{xx},\ \varphi_x\varphi^c_{xy},\ \varphi_x\varphi^c_{yy},\ \varphi_y\varphi^c_{xx},\ \varphi_y\varphi^c_{xy},\ \varphi_y\varphi^c_{yy},$$

which are not completely independent (e.g., φ_x and $\varphi_x\varphi^c_{xx}$ are not independent) and therefore, leads to an underdetermined problem. For instance, the Galerkin discretization of the advection equation $\partial_x u = 0$ with a polynomial $u_h = \varphi_1 a + \varphi_x b + \varphi_x c$ will result in two independent discrete equations for three coefficients (a, b, c). Consequently, the coefficients cannot be determined uniquely, thus will lead to inconsistency. For instance, some coefficients could be obtained accurately while other coefficients are not accurate at all; e.g., a and b could be accurate but c could be inaccurate. In the P^2 scheme we have thus far, this, for example, could lead to accurate $\overline{p}$ and $\overline{q}$ predictions and very inaccurate $p_{xx}, \mathrm{v}_{xxy}, \mathrm{v}_{xyy}$, and/or q_{yy} results. In a more general situation (or a worse case), there is no guarantee that any of the variables could be predicted accurately. To remedy this issue, we upgrade the u_h by one degree. This is possible without a reconstruction or increase of DoF because p_{xx}, v_{xxy}, v_{xyy}, and q_{yy} correspond to the cubic terms for the cubic polynomial u. With the unified notation for the p_{xx} and u_{xxx}, i.e.,

$$p_{xx} \equiv u_{xxx},$$

the upgraded polynomial approximation for the P^2 case reduces to

$$u_h = \varphi_1 \overline{u} + \varphi_x \overline{p} + \varphi_y \overline{q} + \varphi_{xx} p_x + \varphi_{xy} \mathrm{v}_{xy} + \varphi_{yy} q_y$$
$$+ (\varphi_x \varphi^c_{xx} + \varphi_{xxx}) p_{xx} + (\varphi_x \varphi^c_{xy} + \varphi_y \varphi^c_{xx} + \varphi_{xxy}) \mathrm{v}_{xxy} + (\varphi_x \varphi^c_{yy} + \varphi_y \varphi^c_{xy} + \varphi_{xyy}) \mathrm{v}_{xyy} + (\varphi_y \varphi^c_{yy} + \varphi_{yyy}) q_{yy},$$
$$p_h = \varphi_1 \overline{p} + \varphi_x p_x + \varphi_y \mathrm{v}_{xy} + \varphi_{xx} p_{xx} + \varphi_{xy} \mathrm{v}_{xxy} + \varphi_{yy} \mathrm{v}_{xyy},$$
$$q_h = \varphi_1 \overline{q} + \varphi_x \mathrm{v}_{xy} + \varphi_y q_y + \varphi_{xx} \mathrm{v}_{xxy} + \varphi_{xy} \mathrm{v}_{xyy} + \varphi_{yy} q_{yy},$$

which corresponds to the following independent modified basis functions:

$$\varphi_1,\ \varphi_x,\ \varphi_y,\ \varphi_{xx},\ \varphi_{xy},\ \varphi_{yy},\ (\varphi_x\varphi^c_{xx} + \varphi_{xxx}),\ (\varphi_x\varphi^c_{xy} + \varphi_y\varphi^c_{xx} + \varphi_{xxy}),\ (\varphi_x\varphi^c_{yy} + \varphi_y\varphi^c_{xy} + \varphi_{xyy}),\ (\varphi_y\varphi^c_{yy} + \varphi_y\varphi^c_{yy}).$$

Thus, the one degree polynomial upgrade for u_h is no longer an option but required for our proposed DG-H(P^2) scheme. In fact, this procedure is required for any DG-H(P^k) scheme with $k \geq 2$.

Expressing the polynomial approximation in a vector form, we arrive at

$$\mathbf{U}_h = \tilde{\mathbf{B}}_2 \tilde{\mathbf{u}}_2 = \tilde{\mathbf{B}}_2 \mathbf{Z}_2 \mathbf{V}_2 = \mathbf{C}_2 \mathbf{V}_2, \tag{43}$$

where

$$\tilde{\mathbf{u}}_2 = [\overline{u},\ \overline{p},\ \overline{q},\ (\overline{p} + \varphi^c_{xx} p_{xx} + \varphi^c_{xy} \mathrm{v}_{xxy} + \varphi^c_{yy} \mathrm{v}_{xyy}),\ p_x,\ \mathrm{v}_{xy},\ (\overline{q} + \varphi^c_{xx} \mathrm{v}_{xxy} + \varphi^c_{xy} \mathrm{v}_{xyy} + \varphi^c_{yy} q_{yy}),$$
$$\mathrm{v}_{xy},\ q_y,\ p_x,\ p_{xx},\ \mathrm{v}_{xxy},\ \mathrm{v}_{xy},\ \mathrm{v}_{xxy},\ \mathrm{v}_{xyy},\ q_y,\ \mathrm{v}_{xyy},\ q_{yy}]^T, \tag{44}$$

and

$$\mathbf{V}_2 = \left[\overline{u},\ \overline{p},\ \overline{q},\ p_x,\ \mathrm{v}_{xy},\ q_y,\ p_{xx},\ \mathrm{v}_{xxy},\ \mathrm{v}_{xyy},\ q_{yy}\right]^T, \tag{45}$$

$$\tilde{\mathbf{B}}_2 = \begin{bmatrix} \varphi_1 & 0 & 0 \\ 0 & \varphi_1 & 0 \\ 0 & 0 & \varphi_1 \\ \varphi_x & 0 & 0 \\ 0 & \varphi_x & 0 \\ 0 & 0 & \varphi_x \\ \varphi_y & 0 & 0 \\ 0 & \varphi_y & 0 \\ 0 & 0 & \varphi_y \\ \varphi_{xx} & 0 & 0 \\ \varphi_{xxx} & \varphi_{xx} & 0 \\ \omega_1\varphi_{xxy} & 0 & \varphi_{xx} \\ \varphi_{xy} & 0 & 0 \\ (1-\omega_1)\varphi_{xxy} & \varphi_{xy} & 0 \\ \omega_2\varphi_{xyy} & 0 & \varphi_{xy} \\ \varphi_{yy} & 0 & 0 \\ (1-\omega_2)\varphi_{xyy} & \varphi_{yy} & 0 \\ \varphi_{yyy} & 0 & \varphi_{yy} \end{bmatrix}^T, \quad \mathbf{Z}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \varphi^c_{xx} & \varphi^c_{xy} & \varphi^c_{yy} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & \varphi^c_{xx} & \varphi^c_{xy} & \varphi^c_{yy} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{46}$$

where, again, ω_1 and ω_2, which can take any value, appear because $q_{xx} = p_{xy}$ and $q_{xy} = p_{yy}$, and have no effect on the result as they will be canceled in the modified basis function, $\mathbf{C}_2 = \tilde{\mathbf{B}}_2\mathbf{Z}_2$, which is the one used in the Galerkin discretization. Similar to the proposed P^1 scheme, we can also expressed the resulting polynomial approximation using the original basis functions, $\mathbf{B}$, as

$$\mathbf{U}_h = \mathbf{B}_2\,\tilde{\mathbf{u}}_2 + \mathbf{r}_u\mathbf{c}_2^T\mathbf{V}_2 = (\mathbf{B}_2\mathbf{Z}_2 + \mathbf{r}_u\mathbf{c}_2^T)\mathbf{V}_2, \tag{47}$$

where, again, $\mathbf{r}_u$ is a vector indicating which variable is given the extra terms and $\mathbf{c}_2^T$ is a vector containing the extra basis functions:

$$\mathbf{r}_u = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{c}_2^T = [0, 0, 0, 0, 0, 0, \varphi_{xxx}, \varphi_{xxy}, \varphi_{xyy}, \varphi_{yyy}].$$

Note that Eqs. (43) and (47) are identical. With the formulations of the modified basis function, $\mathbf{C}_2$, and the reduced vector of unknown polynomial coefficients, $\mathbf{V}_2$, we apply the Galerkin discretization, Eq. (18), and solve for the pseudo-time evolution equations.

Our proposed P^2 scheme is third-order for both the solution and the solution gradients, and is exact for exact cubic functions. In addition, the scheme becomes fourth-order in the advection-limit for the primal variables. For general advection–diffusion problems, the proposed scheme is equivalent to the conventional DG(P^3) scheme for the same order of accuracy of the solution gradients and the total number of DoF.

3.4. DG-H scheme with a kth-order polynomial (P^k)

In a general case, starting with the reformulation of the target governing equation as a first-order hyperbolic advection–diffusion system, we construct the proposed high-order DG-H scheme for an arbitrary polynomial order of k with the following simple steps:

- define a numerical solution for the independent variables of the first-order hyperbolic advection–diffusion system with a kth-order polynomial approximation. For the hyperbolic advection–diffusion system studied here, we arrive at

  $$\mathbf{U}_h = [u_h, p_h, q_h]^T \in P^k,$$

  where P^k denotes a vector space polynomial of degree k that is spanned by the Taylor basis functions;
- unify the coefficients with the process described in the previous sections (e.g., define v_{xy} as a unified notation for q_x and p_y, and so forth);
- upgrade the polynomial approximation for the primal variables by one-degree with the available independent polynomial coefficients. Note that this step is only required for the polynomial approximations of order $k \geq 2$. This step, however, is optional for the proposed P^0 and P^1 schemes. Regardless of the chosen polynomial order k, this step results in a $(k+2)$-order accurate solutions for the primal variables in the advection-limit (high-Reynolds number). Thus, we highly recommend this upgrade for all the polynomial orders as it does not increase the DoF and yet increases the order of accuracy of the primal variables for inviscid problems;

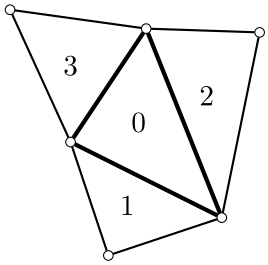


Fig. 2. The WENO stencil.

- express the numerical solution vector $\mathbf{U}_h$ in a vector form as

$$\mathbf{U}_h = \mathbf{B}_k \tilde{\mathbf{u}}_k,$$

  where $\tilde{\mathbf{u}}_k$ is a vector of unknown polynomial coefficients, and $\mathbf{B}_k$ is the basis functions;
- define the reduced vector of unknown polynomial coefficients $\mathbf{V}_k$ by identifying the independent polynomial coefficients.
- express the numerical solution with the modified basis functions as

$$\mathbf{U}_h = \tilde{\mathbf{B}}_k \tilde{\mathbf{u}}_k = \tilde{\mathbf{B}}_k \mathbf{Z}_k \mathbf{V}_k = \mathbf{C}_k \mathbf{V}_k,$$

  where $\mathbf{C}_k = \tilde{\mathbf{B}}_k \mathbf{Z}_k = \mathbf{B}_k \mathbf{Z}_k + \mathbf{r}_u \mathbf{c}_k^T$ is the modified basis functions, and $\mathbf{Z}_k$ may be regarded as an operator that transforms the $\tilde{\mathbf{u}}_k$ to the reduced vector of unknown polynomial coefficients $\mathbf{V}_k$. Note $\mathbf{r}_u$ is a vector indicating the primal variables are given the extra terms and $\mathbf{c}_k^T$ is a vector containing the extra basis functions needed to upgrade the polynomial approximation for the primal variables by one-degree;
- apply the Galerkin discretization, Eq. (18).

4. WENO limiting procedure for the proposed DG-H schemes

Here we simply follow the compact WENO reconstruction procedure outlined by Zhu et al. [56,57] for triangular elements, and apply it to the proposed DG-H schemes. The main advantages of this WENO limiter over other proposed WENO limiters [58,59] are its compactness, depending only on the information from the target cell and its immediate neighbors, and the ability to reconstruct the entire polynomial (point values and all the moments) in one shot. For completeness and better clarity, we briefly describe the procedure here in the context of our DG-H schemes.

We start the WENO limiting procedure by replacing the cell averaged values of the neighbors of the target cells, on which the WENO reconstruction is being performed, with the cell averaged values of the target cell. This process, which ensures that the reconstructed polynomial maintains the original cell averaged values, can be expressed as

$$\tilde{\mathbf{U}}_h^{(i)} = \mathbf{U}_h^{(i)} - \overline{\mathbf{U}}^{(i)} + \overline{\mathbf{U}}^{(0)}, \quad i = 1, 2, 3 \tag{48}$$

where the superscript (i) denotes the local cell number as depicted in Fig. 2, and the over-line corresponds to the cell averaged values on the target cell, $E^{(0)}$; i.e.,

$$\overline{\mathbf{U}}^{(i)} = \frac{1}{A_{E^{(0)}}} \int_{E^{(0)}} \mathbf{U}_h^{(i)} d\Omega$$

where $A_{E^{(0)}}$ is the area of the target cell. We then create a new polynomial for the target cell by a convex combination of the modified polynomials obtained in the previous step on the WENO stencil (see Fig. 2), and arrive at

$$\mathbf{U}_h^{(0)^{new}} = \sum_{i=0}^{3} \mathrm{W}_i \tilde{\mathbf{U}}_h^{(i)}, \tag{49}$$

where W_i are the normalized nonlinear weights defined as [60–62,57,63]

$$\mathrm{W}_i = \frac{\overline{\mathrm{W}}_i}{\sum_{j=0}^{3} \overline{\mathrm{W}}_j}, \quad \overline{\mathrm{W}}_i = \frac{\gamma_i}{(1 + \beta_i^2)^r}, \tag{50}$$

and γ_i are the linear weights (such that $\sum_{i=0}^{3} \gamma_i = 1$),

$$\gamma_i = \begin{cases} 0.001, & i \neq 0 \text{ (neighboring cell)} \\ 1 - \sum_{\substack{j \\ j \neq 0}} \gamma_j, & i = 0 \text{ (target cell)} \end{cases}, \tag{51}$$

which is based on the recommendations made by Zhong and Shu [63] and Dumbser and Käser [64]. Here, the smoothness indicator, β_i, [60] is defined for each component of the $\tilde{\mathbf{U}}_h$ vector. For example, the smoothness indicator for the primal variable $\tilde{u}_h$ may be expressed as

$$\beta_i = \sum_{|l|=1}^{k} A_{E^{(0)}}^{|l|-1} \int_{E^{(0)}} \left(\frac{\partial^{|l|}}{\partial x^{l_1} \partial y^{l_2}} \tilde{u}_h^{(i)} \right)^2 d\Omega, \quad l = (l_1, l_2), \quad |l| = l_1 + l_2, \tag{52}$$

and similarly for other variables. The r parameter in the denominator of the Eq. (50) is an integer; the higher the value of r, the more rapid the transition is from smooth to discontinuous regions, hence the better non-oscillatory performance as well as accuracy in smooth regions, but the less robust to steady state convergence (which relies on the smoothness of the WENO weights) [65]. In our simulations, we use $r = 2$ but observed no noticeable difference in solutions with either $r = 1$ or $r = 2$ for the problems studied here.

Note that for better efficiency the WENO limiter is usually applied only to the *trouble cells* [66] (that is the cells that contain oscillatory solutions), but can also be applied to smooth regions without the loss of accuracy.

In summary, and in the context of our proposed DG-H schemes, we repeat the following steps starting with a polynomial $\mathbf{U}_h = \mathbf{C}_k \mathbf{V}_k$ until convergence is obtained:

- solve the DG-H scheme to get a candidate polynomial for the next nonlinear iteration;
- apply the WENO limiter of Zhong and Shu [63] (to all or only to the trouble cells), and compute a new polynomial approximation, $\mathbf{U}_h^{(0)^{new}}$, using Eq. (49). In this work, we apply the WENO limiter to all the cells;
- compute the moments (or polynomial coefficients) of the reconstructed polynomial as

$$\mathbf{V}_k = \mathbf{M}_k^{-1} \int_{E^{(0)}} \mathbf{C}_k^T \mathbf{U}_h^{(0)^{new}} d\Omega.$$

5. Numerical experiments

In this section, we present some numerical experiments to demonstrate the performance of the proposed DG-H schemes. Here, for clarity and without loss of any ambiguity, we drop the subscript k corresponding to the polynomial order.

We obtain all solutions with Newton iterations by constructing an implicit solver as

$$\mathbf{M}\, \partial_\tau \mathbf{V} = \mathbf{Res}(\mathbf{V}), \tag{53}$$

where $\mathbf{V}$ is a global vector of unknown polynomial coefficients. We then solve Eq. (53) for the pseudo steady state:

$$\left(\frac{\mathbf{M}}{\Delta\tau} \mathbf{I} - \frac{\partial \mathbf{Res}}{\partial \mathbf{V}} \right) \Delta \mathbf{V} = \mathbf{Res}(\mathbf{V}),$$

where $\Delta\tau$ is a pseudo-time step taken as infinity for problems with no discontinuity. For problems containing discontinuity, we usually start with $\Delta\tau = O(1)$ and gradually increase it with Newton iterations to $\Delta\tau = 10^7$. Note that the physical time derivatives are discretized and included in $\mathbf{Res}(\mathbf{V})$ (see Ref. [3]). Here, we only report steady state solutions. We compute $\Delta\mathbf{V}$ by solving the above system of linearized equations with the exact residual Jacobians evaluated by an Automatic Differentiation algorithm using the operator overloading technique through the chain rules [4,7], and then we update the solution by Newton's method $\mathbf{V} + \Delta\mathbf{V}$. We solved the system of linearized equations both with MUMPS direct sparse linear solver [67,68], and iteratively using Gauss–Seidel algorithm. For two-dimensional problems considered here, we found that the direct linear solver is more efficient than the Gauss–Seidel iterative solver with three order reductions in the linear residuals.

We perform the numerical experiments on fixed irregular unstructured triangular grids. We generate these irregular grids in two steps; 1) we randomly modify the orientations of the triangular edges of a uniform grid, 2) we then apply 40% random perturbations to both the x and y coordinates of the nodes. For example, the new x nodal coordinates are taken to be $x \pm 40\%(\text{random number} - 1/2)\Delta x$, where Δx is the distance between the node's immediate right and left neighbors along the x-direction. We repeat a similar procedure for the y nodal coordinates. Fig. 3 shows a representative of irregular unstructured triangular grids. In our experiments, we do not perform grid adaptation nor do we use specialized grids for any of the test cases presented here.

For all the computational results, we use the local Lax–Friedrichs (LLxF) numerical flux and the quadrature rules that are exact for polynomial orders of $(2k+1)$ and $2k$, respectively, for edge and interior integrals [44]. These, along with the corresponding DoF per element are tabulated in Table 1.

All the presented results for the DG-H schemes are based on the numerical approximation given in Eq. (17). That is, the DG-H(P^k) schemes correspond to P^{k+1} polynomial approximation for the primal variable u, while the gradients of u (i.e., p and q) are approximated with a polynomial degree of order k.

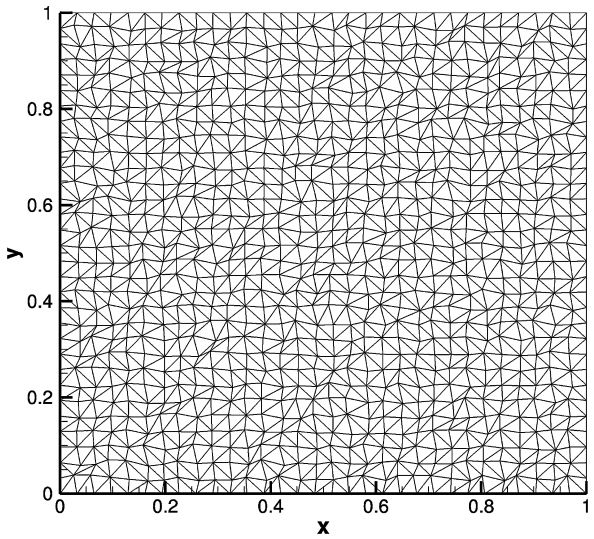


Fig. 3. Representative of an irregular unstructured triangular grid.

Table 1
Number of quadrature points used for the proposed DG-H and the conventional DG schemes on irregular triangular elements.

Polynomial (P^k)	Edge ($2k+1$)		Interior ($2k$)		DoF per element	
	DG-H	DG	DG-H	DG	DG-H	DG
P^0	2	2	3	3	3	1
P^1	2	2	3	3	6	3
P^2	3	3	7	7	10	6
P^3	4	4	13	13	15	10
P^4	5	5	19	19	21	15

5.1. Accuracy verifications

Consider the steady advection–diffusion equation

$$a\partial_x u + b\partial_y u = \nu\left(\partial_{xx}u + \partial_{yy}u\right), \quad \nu = \begin{cases} 1 \text{ or } 10^{-8}, & \text{for DG-H} \\ 1 \text{ or } 10^{-8}, & \text{for DG} \end{cases},$$

with a Dirichlet boundary condition in a unit square domain. The exact solution has an exponential form [69]

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

where A and C are arbitrary constants, $\xi = ax + by$, and $\eta = bx - ay$. We take $a = 2$, $b = 1$, $A = 2$, and $C = -0.009$ (see Ref. [7]). For the vanishingly small viscosity coefficient (advection problem), we compare the results with conventional DG schemes. These are shown in Fig. 4 (Note: the comparison plots are on the same scale). In the advection limit, we obtain $(k+2)$-order of accuracy for the primal, and $(k+1)$-order of accuracy for the solutions gradients with the DG-H schemes. For the DG schemes on irregular elements, we achieve $(k+1)$-order accurate for primal and (k)-order accurate solution gradients. Thus, the DG-H(P^k) schemes are equivalent to DG(P^{k+1}) schemes for the same order of accuracy for both solution and solution gradients and the same numbers of DoF. We again note that the DG-H(P^k) schemes use the polynomial approximation of order k that is implicitly upgraded (without any reconstruction) to a polynomial degree of order $(k+1)$ for the primal variable u. Thus both DG-H(P^k) and DG(P^{k+1}) give $(k+2)$-order accurate solution and $(k+1)$-order accurate solution gradients, and they both have identical DoF.

The accuracy results for the advection–diffusion problem with $\nu = 1$ are shown in Fig. 5, where the results are compared with interior penalty (IP) schemes (see Appendix A for the IP formulation used in this work). The $(k+1)$-order of accuracy is achieved for both solution and solution gradients with the proposed DG-H schemes. The results obtained with the IP schemes are also provided in this figure. For irregular triangular elements, the IP schemes achieve $(k+1)$-order accurate solution if k is an odd number. For even k, we observed $(k+1/2)$-order accurate for the solution variables u. For any polynomial approximation of order of k (i.e., both odd and even), the IP schemes give kth-order accurate solution gradients for advection, diffusion, and advection–diffusion problems. Note that a formal proof of $(k+1/2)$ accurate solution for general random grid is given in Refs. [70,71]. However, $(k+1)$-order accurate solution is often observed in practice.

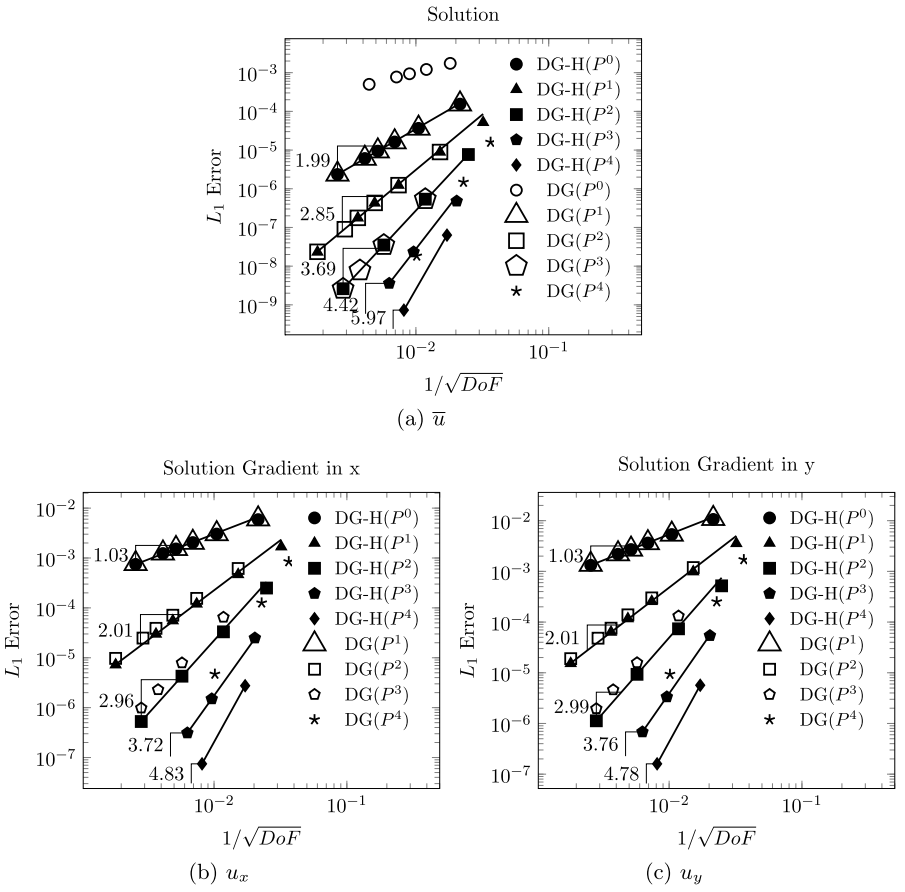


Fig. 4. Accuracy verification for the proposed DG-H schemes compared with the conventional DG schemes on irregular triangular grids for advection equations.

Table 2
Order of accuracy comparisons between the proposed DG-H and high-order DG and IP (P^k) schemes on irregular triangular elements.

Problem	DG-H order of accuracy		DG order of accuracy		IP order of accuracy	
	Solution	Gradients	Solution	Gradients	Solution	Gradients
Advection	$k+2$	$k+1$	$k+1$	k	$k+1$	k
Diffusion	$k+1$	$k+1$	–	–	$k+1$[a]	k
Advection–diffusion	$k+1$	$k+1$	–	–	$k+1$[a]	k

[a] For k even we observed $k+1/2$; see Fig. 5.

We remark that both the DG-H and the IP schemes produce excellent quality solution and solution gradients on irregular triangular elements. We also note that the DG-H schemes produce equal order of accurate solution and solution gradients for general advection–diffusion problems, without requiring a second-order diffusion operator. Equal order of accuracy for solution and solution gradients could potentially be beneficial in, for example, multigrid applications, which we will investigate in future.

In a general term, we can summarize the accuracy features of the DG-H and DG schemes as tabulated in Table 2. These results are based on the results presented in Figs. 4 and 5. Order of accuracy of the advection and diffusion terms as well as the solution gradients obtained by the DG-H and the DG schemes are also compiled with in terms of DoF. These are given in Table 3, which shows that the same order of accuracy is achieved without increasing DoF, for the advective term and the gradients. Obtaining high-order accurate gradients is an objective of this work.

We would also like to remark on the relative computational costs of the proposed DG-H schemes in comparison with the high-order DG schemes. All computations are performed on a Macbook Pro with a 2.3 GHz Intel Core i7 chip and a 16 GB 1600 MHz DDR3 ram. We did not attempt any code optimization. It is also necessary to note that our code is a research code, and thus the reported CPU times is only qualitative. Figs. 6 and 7 show the relative errors of the solution and solution gradients versus the CPU time for various polynomial orders for DG-H, DG, and IP schemes. In general, both figures show

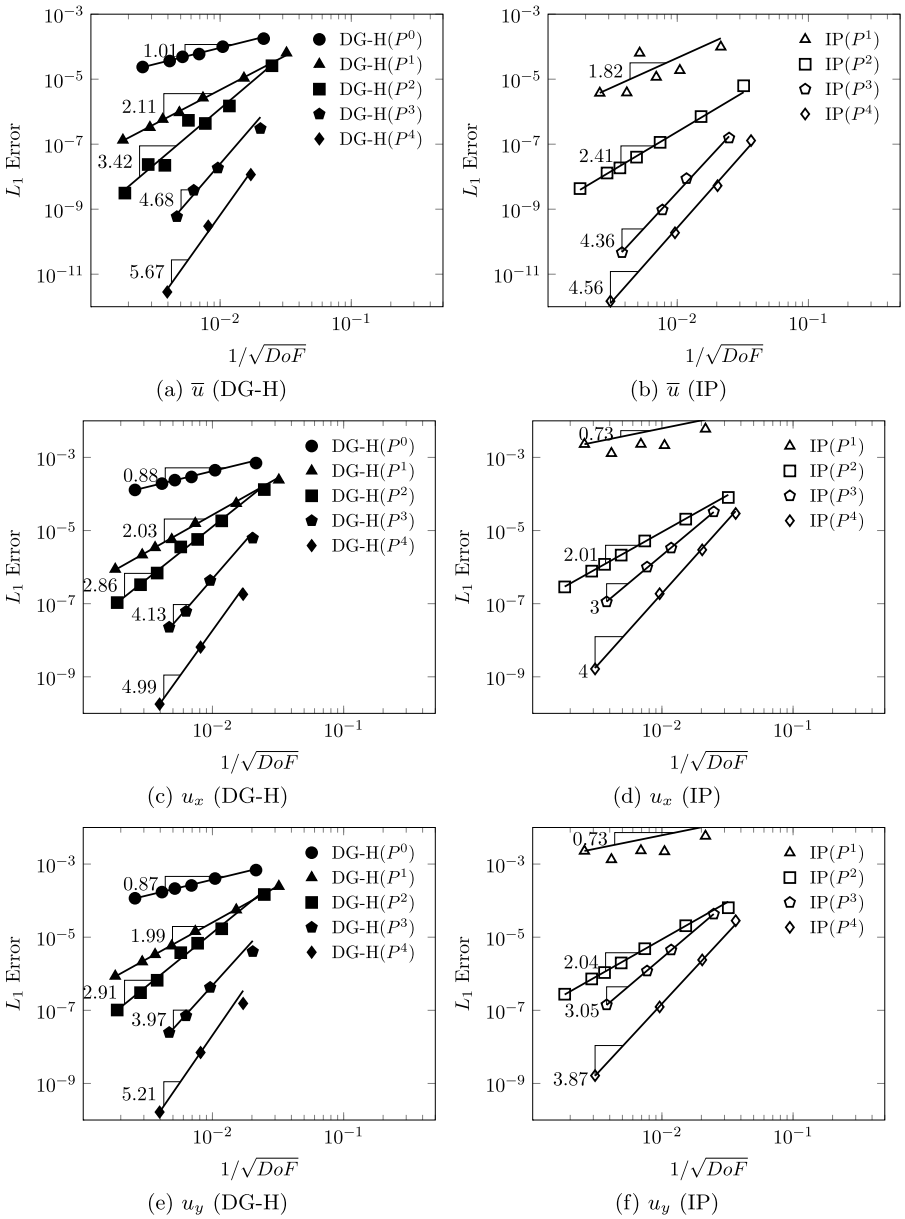


Fig. 5. Accuracy verification for the proposed DG-H schemes on irregular triangular grids for advection–diffusion equations compared with the interior penalty (IP) scheme.

Table 3
Order of accuracy of the advective term, the diffusion term, and the gradients for the proposed DG-H and the conventional DG schemes with an identical DoF.

DoF	DG-H			DG/IP		
	Advection	Diffusion	Gradients	Advection	Diffusion	Gradients
3	2	1	1	2	2	1
6	3	2	2	3	3	2
10	4	3	3	4	4	3
15	5	4	4	5	5	4
21	6	5	5	6	6	5

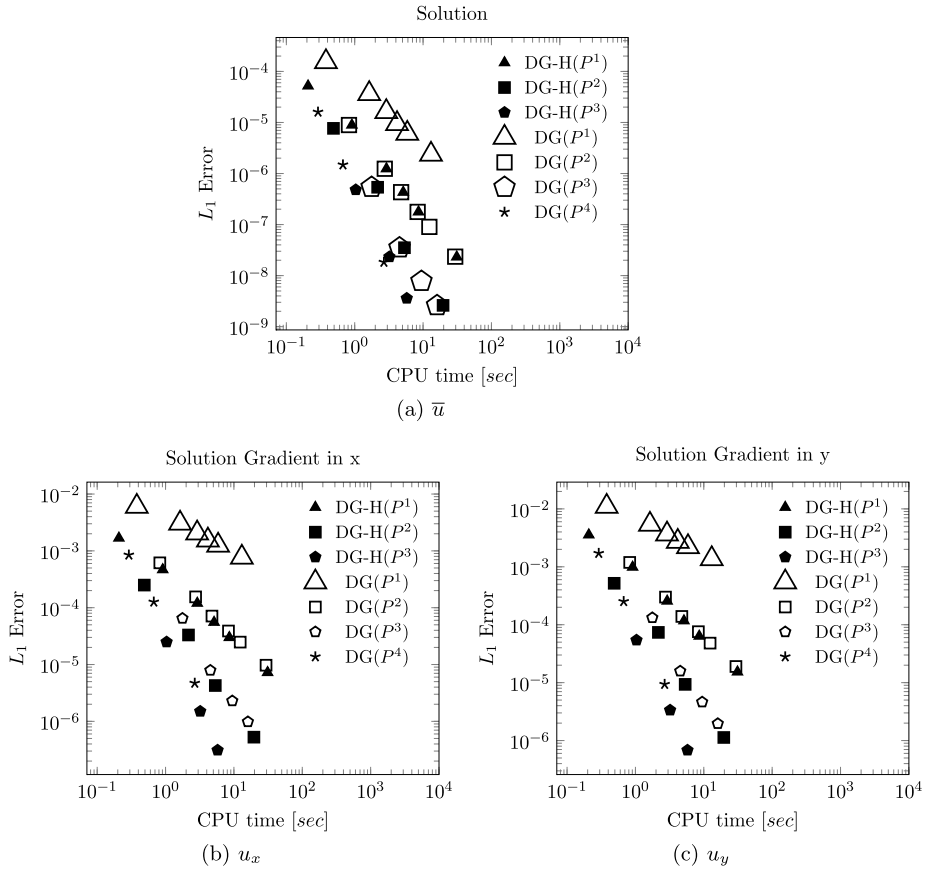


Fig. 6. Relative error vs. CPU time for solution and solution gradients computed with the high-order DG-H and DG schemes.

that for a given CPU time, higher order schemes produce results that are orders of magnitude more accurate than the ones predicted with the lower order schemes and therefore, high-order schemes are more efficient than the low-order schemes. The results also indicate that the proposed DG-H schemes cost nearly the same as the DG schemes to achieve relatively the same error. In comparison with IP schemes, both the DG-H and the IP schemes produce relatively the same error for the solution gradients for a given CPU time. However, it appears that the high-order IP schemes are slightly more efficient than the proposed DG-H schemes, but a more comprehensive analysis for more complex simulations (e.g., Navier–Stokes) is needed to better characterize these two schemes. Thus, the reported CPU times are not completely conclusive. It is also important to conduct such analysis with a code that is optimized for each of the presented schemes; we performed no optimization for any of the presented schemes in our study. We plan to report on such a comprehensive cost analysis for the Navier–Stokes equations in future studies.

5.2. Burgers equation

Consider the steady Burgers equation

$$\partial_x\left(\frac{u^2}{2}\right) + \partial_y u = \nu\left(\partial_{xx}u + \partial_{yy}u\right), \quad \nu = \begin{cases} 10^{-8}, & \text{for DG-H} \\ 0, & \text{for DG} \end{cases},$$

with $u(x,0) = 2x - 1.5$ as a boundary condition. Fig. 8 shows the contour plots of the solution computed by the proposed DG-H and the conventional DG schemes with and without the WENO limiter. Both the proposed DG-H and the DG schemes produce good solutions with and without the WENO limiter. We observed no noticeable difference between the solutions obtained with and without the WENO limiter. This is similar to the observation provided in Ref. [57] that the DG schemes can compute, without any limiter, solutions that are smooth or contains not strong-enough shocks. It appears that the Burgers equation do not produce a strong-enough shock and therefore, it is not surprising that the schemes without WENO limiter perform equally well.

Solution gradients across the compression fans are discontinuous. As shown in Fig. 9, both DG-H and DG schemes of all polynomial orders accurately predict the discontinuous solution gradients on the irregular triangular grids. These results are

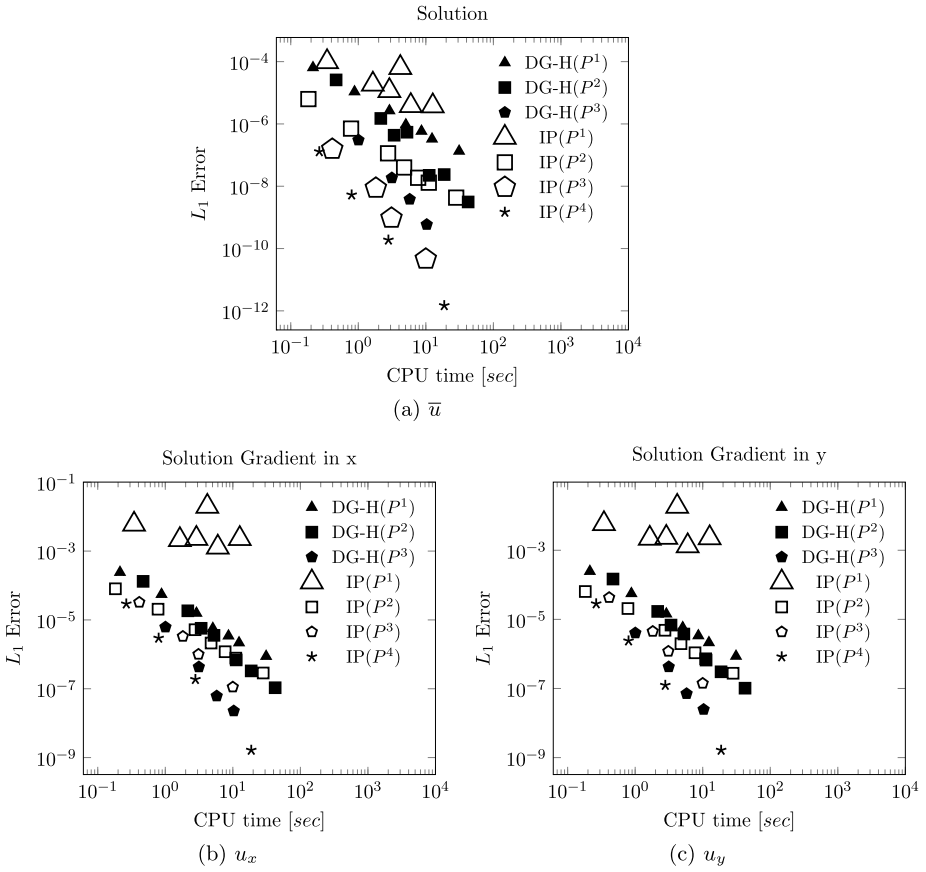


Fig. 7. Relative error vs. CPU time for solution and solution gradients computed with the high-order DG-H and IP schemes.

particularly remarkable compared with the high-order RD schemes proposed in Refs. [7,9], which despite its attractiveness and ability to predict discontinuous solutions accurately, predicted less accurate discontinuous solution gradients.

We also observed that sufficient reduction in the residual of the first moments (i.e., u_x and u_y) are required for both the DG-H and the DG schemes to obtain accurate solution gradients. This is demonstrated in Fig. 10.

It is also interesting to note that predicting good solution gradients does not necessarily require high-order DG schemes. The first-order solution gradients obtained with the DG-H(P^0) and the DG(P^1) on a 60×60 irregular triangular grid, which is shown in Fig. 11, are also very good.

5.3. Buckley–Leverett equation

Consider a two-dimensional non-convex Buckley–Leverett equation,

$$\partial_x \left(\frac{4u^2}{4u^2 + (1-u)^2} \right) + \partial_y u = \nu \left(\partial_{xx} u + \partial_{yy} u \right), \quad \nu = \begin{cases} 10^{-8}, & \text{for DG-H} \\ 0, & \text{for DG} \end{cases},$$

in $(x, y) \in [-1, 1] \times [0, 1]$ with $u(x, 0) = 1$ for $-\frac{1}{2} \leq x \leq 0$ and $u(x, 0) = 0$ otherwise. The exact solution to this equation is a mixture of shock, rarefaction, and contact discontinuity. The three-dimensional contour plot of the solution obtained with the fifth-order DG-H schemes, as shown in Fig. 12, illustrates the composite wave structure of this equation, which may be regarded as a model equation for two-phase flow. As pointed out by Qui et al. [72], some high-order methods may fail to converge to the correct entropy solution for this problem.

The contour plots of the high-order solution obtained with the proposed DG-H and the conventional DG schemes are presented in Fig. 13. The high-order solutions with and without the WENO limiter are nearly the same and have no noticeable under/overshoots. Again, this is because the shocks generated by the Buckley–Leverett equation, similar to the Burgers equation test case, are not strong-enough to admit solution oscillations for either the DG-H or the DG schemes. To generate strong-enough shocks, we will revisit similar comparisons (in future studies) when we apply these schemes to the Navier–Stokes equations.

The predicted solutions are compared with the exact solution at $y = 0.4$ [63,73]. As shown in Fig. 14, both the DG-H and the DG schemes of various orders give a correct entropy solution, and are very accurate in predicting the composite struc-

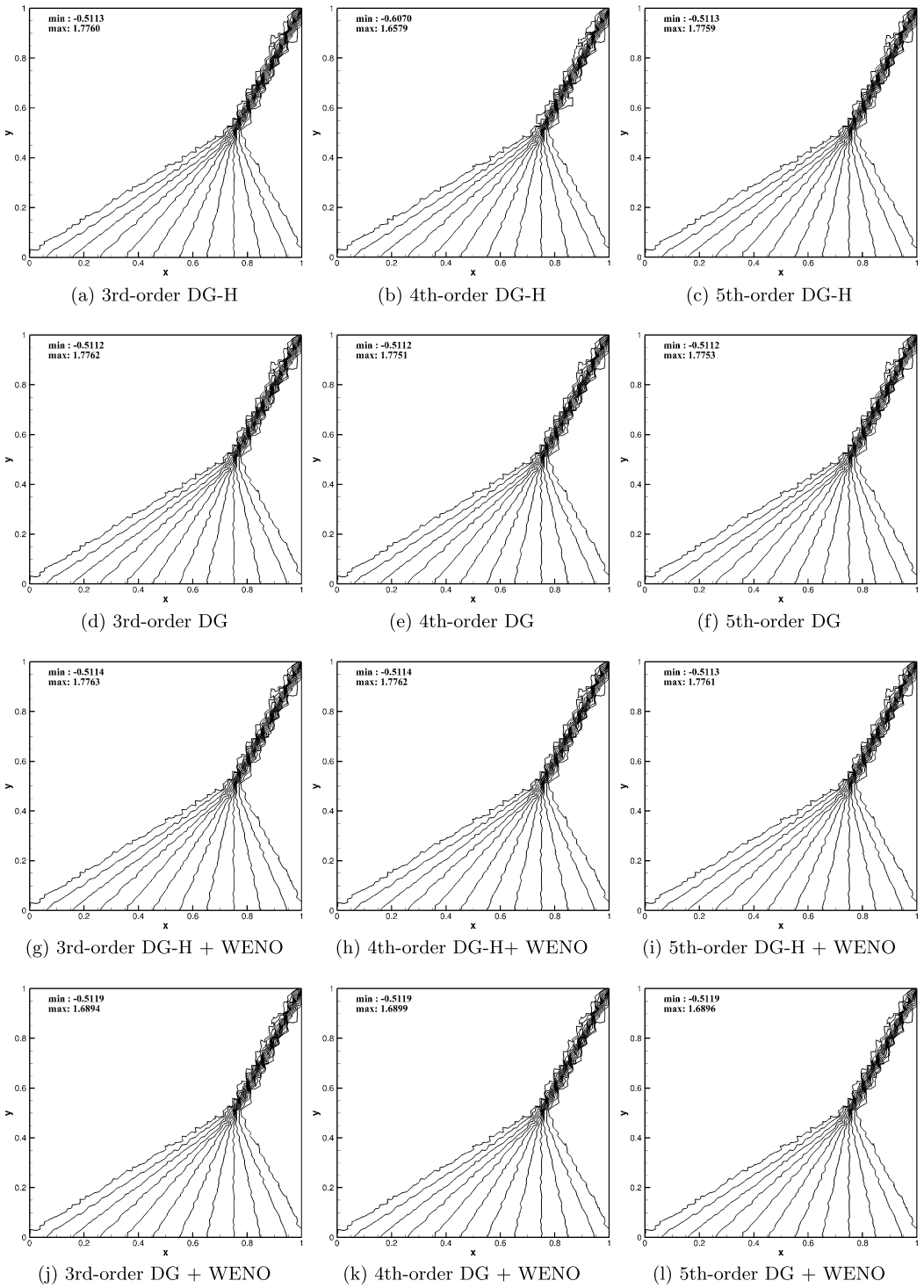
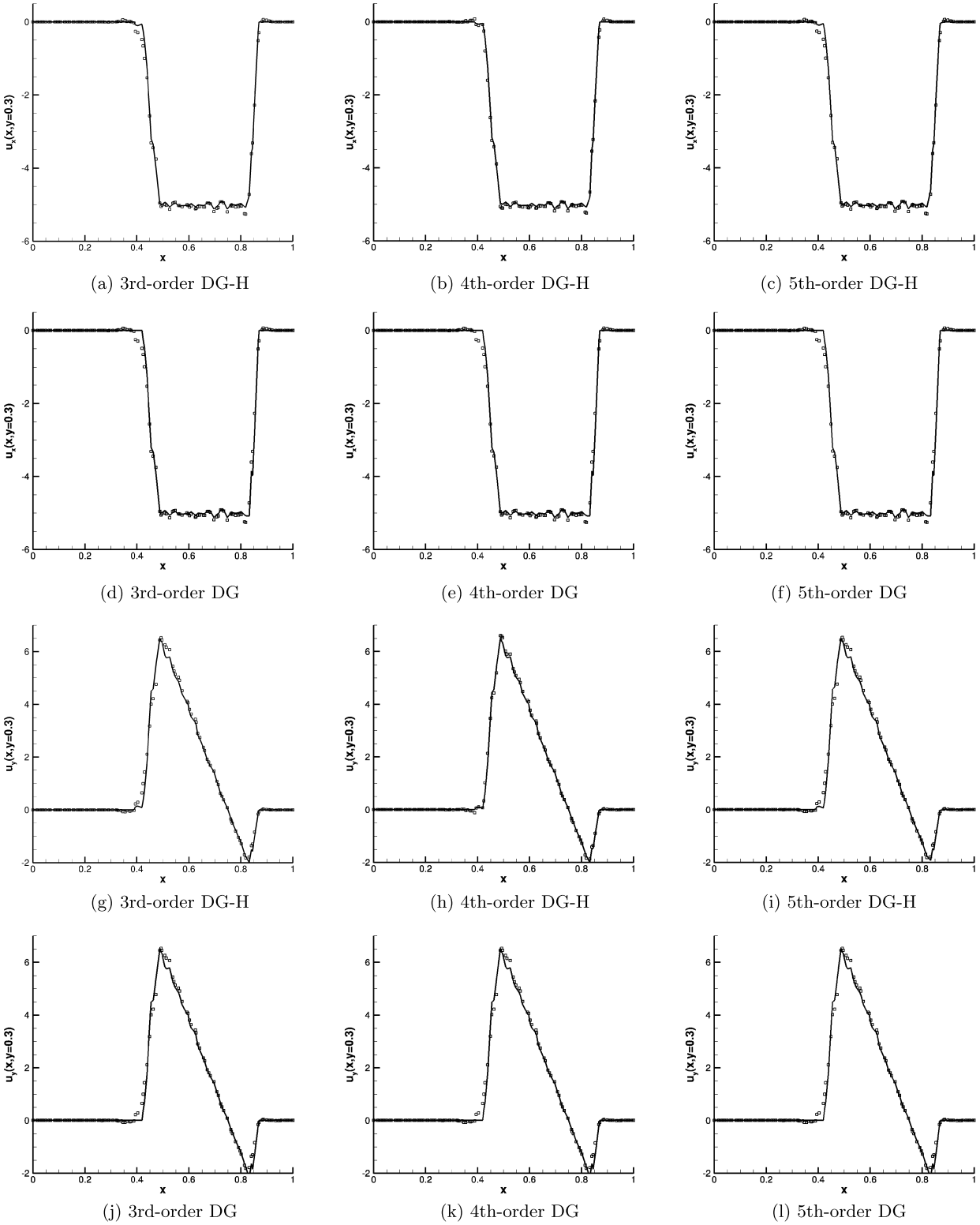


Fig. 8. Burgers test case: Contour plots of the solution u using 60×60 irregular triangular elements.

tures. We also observed no noticeable improvements in the solutions between the third-, fourth-, and fifth-order schemes. Note that the solutions are compared on the same given grid without post-processing the solutions with a high-order visualization [74].



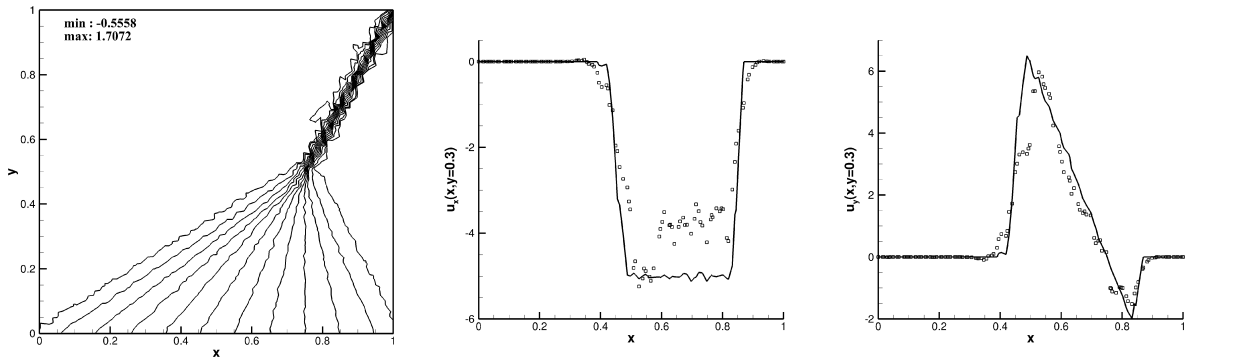
(a) 3rd-order DG-H (b) 4th-order DG-H (c) 5th-order DG-H

(d) 3rd-order DG (e) 4th-order DG (f) 5th-order DG

(g) 3rd-order DG-H (h) 4th-order DG-H (i) 5th-order DG-H

(j) 3rd-order DG (k) 4th-order DG (l) 5th-order DG

Fig. 9. Burgers test case: Discontinuous solution gradients u_x and u_y at $y = 0.3$ with 60×60 irregular triangular elements. The solid lines are the reference values and the symbols are the solution gradients at the cell center.



(a) L_1 of u_x and u_y Residuals $= 10^{-7}$ (b) L_1 of u_x and u_y Residuals $= 10^{-7}$ (c) L_1 of u_x and u_y Residuals $= 10^{-7}$

(d) L_1 of u_x and u_y Residuals $= 10^{-9}$ (e) L_1 of u_x and u_y Residuals $= 10^{-9}$ (f) L_1 of u_x and u_y Residuals $= 10^{-9}$

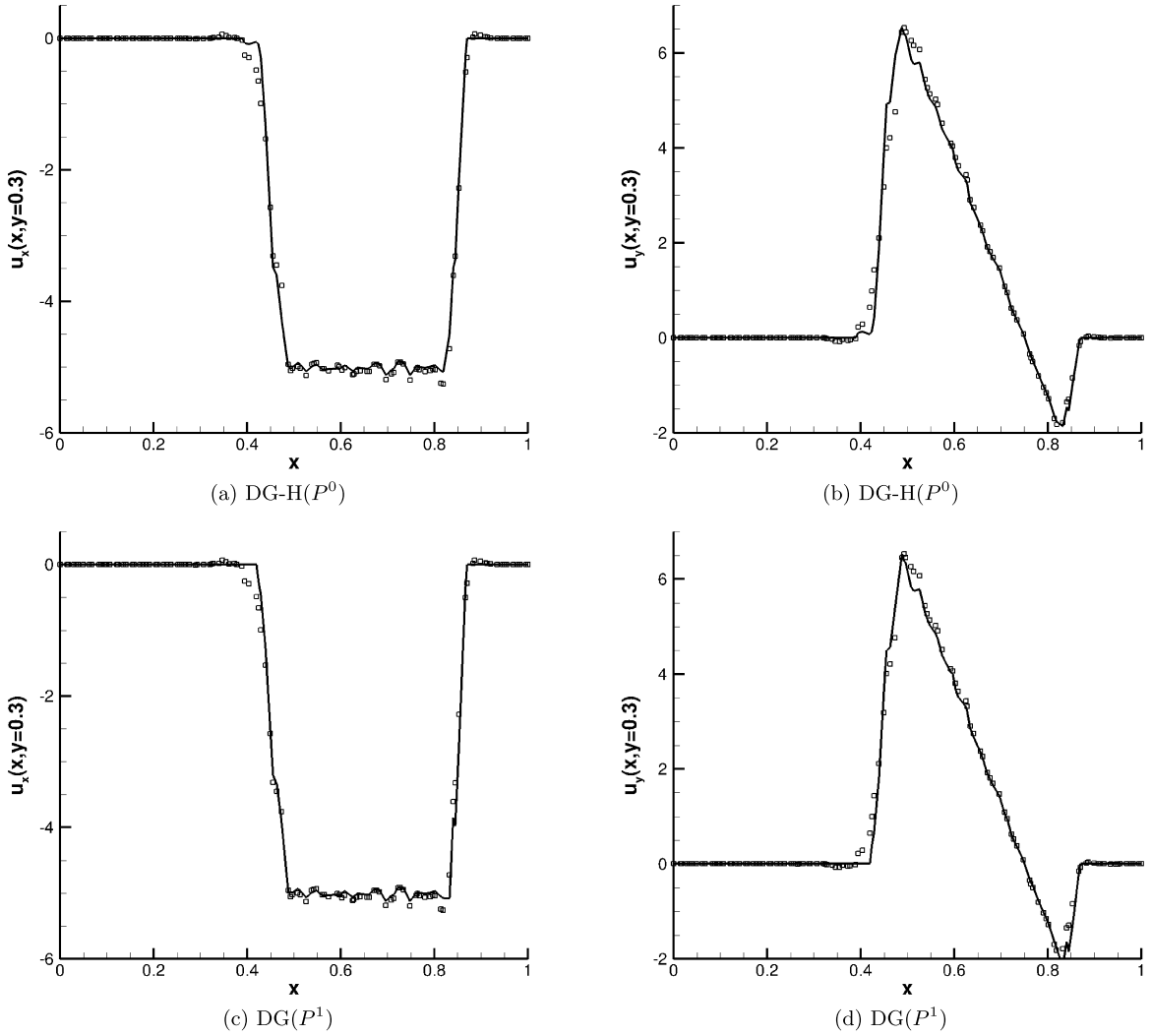
Fig. 10. Burgers test case: Importance of the residual convergences of the first moments in accurately predicting solution gradients. Shown are the results obtained with the third-order DG-H scheme. The solid lines are the reference values and the symbols are the solution gradients at the cell center.

6. Concluding remarks

We proposed arbitrary high-order Discontinuous Galerkin (DG) schemes with the first-order hyperbolic advection–diffusion formulation approach, called DG-H. A construction strategy is introduced that results in the same degrees-of-freedom as the comparable conventional high-order DG schemes. We showed that the constructed DG-H schemes give equal order of accuracy for both the primal and its gradients on irregular triangular grids. A few numerical examples are provided to demonstrate the capability of the DG-H schemes in comparison with the conventional Taylor bases DG schemes (LLxF flux). We also verified the order of accuracy of the DG-H schemes on irregular triangular elements. For diffusion problems, the DG-H schemes of polynomial order of k give $(k+1)$-order accurate solution and solution gradients on irregular triangular elements. For advection dominated problems, the proposed DG-H schemes give $(k+2)$-order of accuracy for the primal variable, and $(k+1)$-order accurate gradients on irregular triangular elements. We also showed that the DG-H, DG and IP schemes produce excellent quality solution and solution gradients on irregular triangular elements. A relative qualitative cost analysis between the proposed DG-H, DG and IP schemes revealed that for a given CPU time, the higher the polynomial order the lower the predicted relative errors and therefore, the high-order schemes are more economical than lower order schemes. We also showed that the proposed DG-H schemes are nearly as efficient as the DG and Interior-Penalty (IP) schemes as these schemes produce results that are relatively at the same error level for approximately a similar CPU time. A more comprehensive cost analysis is needed to qualitatively characterize the performance of the proposed DG-H schemes against DG and IP schemes for more complex simulations with the Navier–Stokes equations.

Acknowledgements

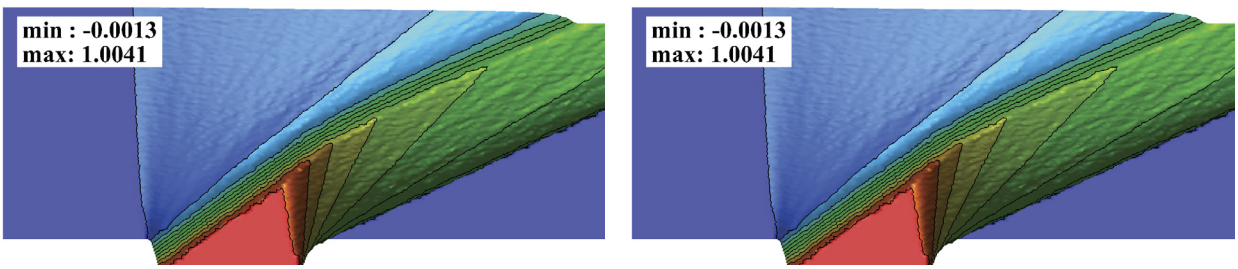
This work is supported internally by the Center Chief Technology Office of NASA Langley Research Center through the Center Innovation Fund (CIF) project. The first author would like to thank Prof. Chi-Wang Shu (Brown University) for helpful discussions, and Dr. Xinghui Zhong (University of Utah) for providing the exact solution to the Buckley–Leverett test problem. We are also grateful to the anonymous reviewers for their constructive comments and help in improving the quality of this manuscript.



(a) DG-H(P^0) (b) DG-H(P^0)

(c) DG(P^1) (d) DG(P^1)

Fig. 11. Burgers test case: Predicting good solution gradients does not necessarily require high-order schemes. Shown are the first-order solution gradients predicted with the second-order DG-H and DG schemes using 60×60 irregular triangular elements. The solid lines are the reference values and the symbols are the solution gradients at the cell center.



(a) 5th-order DG-H (b) 5th-order DG-H + WENO

Fig. 12. Buckley–Leverett test case: three-dimensional contour plot of the composite wave structure obtained with the fifth-order DG-H scheme with and without the WENO limiter using 160×120 irregular triangular elements in $(x, y) \in [-1, 1] \times [0, 1]$.

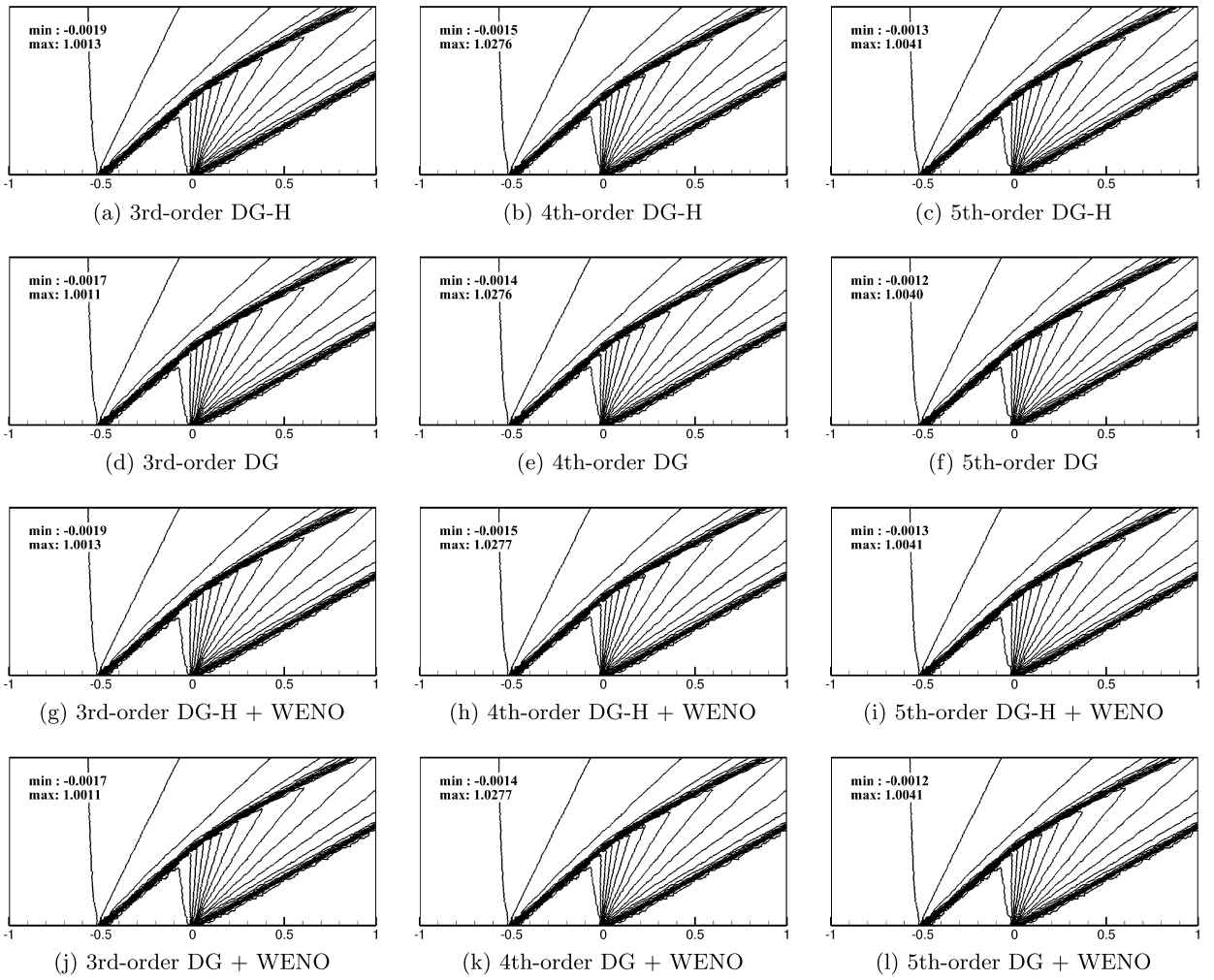


Fig. 13. Buckley–Leverett test case: contour plots of the high-order solution computed with the high-order DG-H and DG schemes with and without the WENO limiter using 160×120 irregular triangular elements in $(x, y) \in [-1, 1] \times [0, 1]$.

Appendix A. Second-derivative diffusion operator

Here, we describe Interior Penalty (IP) schemes for solving nonlinear elliptic equations. Consider the following two-dimensional nonlinear advection–diffusion equation

$$\partial_t u + \partial_x f + \partial_y g = \partial_x(\nu \partial_x u) + \partial_y(\nu \partial_y u), \tag{A.1}$$

where the diffusion coefficient is $\nu = \nu(u)$, and the advection speeds in the x- and y-direction are defined as $a(u) = \partial f/\partial u$ and $b(u) = \partial g/\partial u$, respectively. We write the equation as

$$\partial_t u + \partial_x f + \partial_y g = \partial_x(\nu p) + \partial_y(\nu q), \tag{A.2}$$

$$p - \partial_x u = 0, \tag{A.3}$$

$$q - \partial_y u = 0, \tag{A.4}$$

or in a vector form as

$$\partial_t u + \partial_x f + \partial_y g = \nabla \cdot (\nu \mathbf{D}), \tag{A.5}$$

$$\mathbf{D} = \nabla u. \tag{A.6}$$

We obtain the weak formulation by multiplying the equation with the test functions υ, and defining $\mathbf{H} = (f, g)$ to arrive at

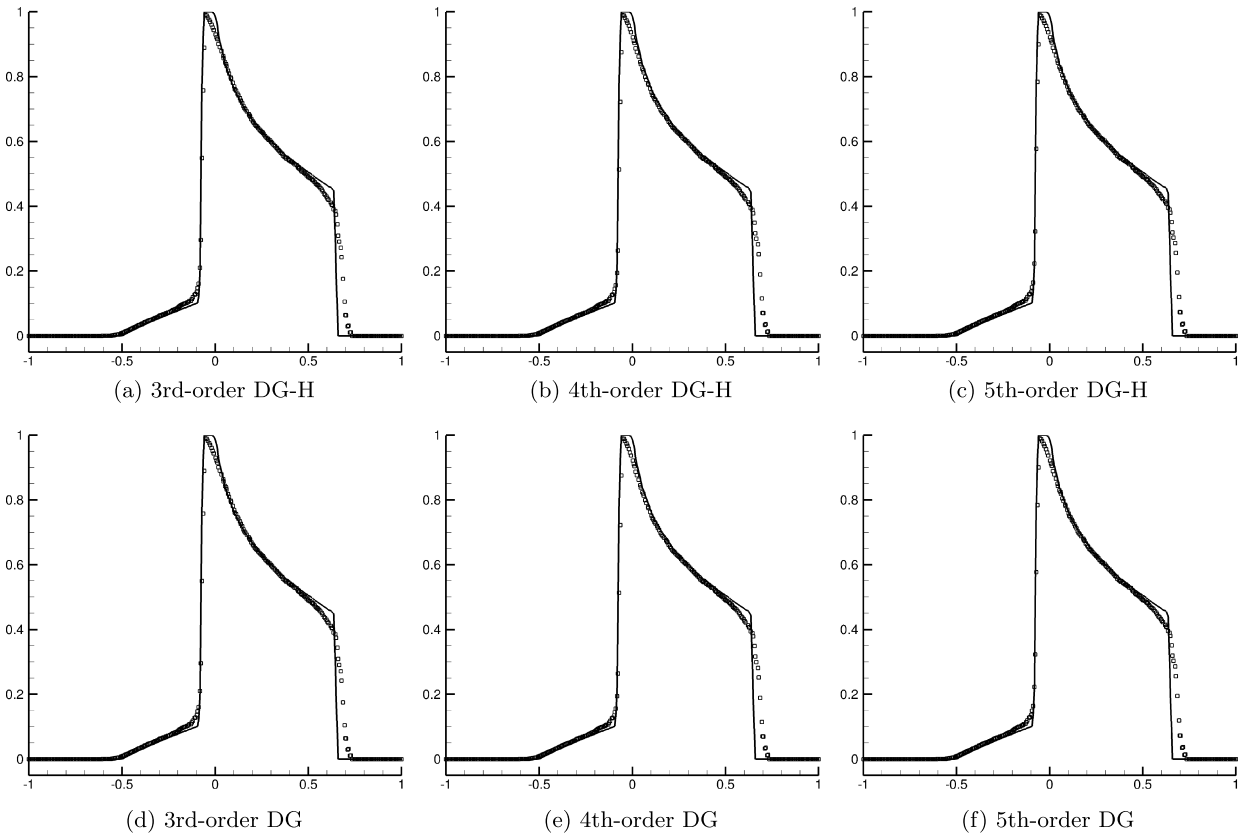


Fig. 14. Buckley–Leverett test case: solution $u(x, y)$ at $y = 0.4$ obtained with the high-order DG-H and DG schemes on 120×80 irregular triangular elements in $(x, y) \in [-1, 1] \times [0, 1]$. Solid lines are the reference solutions, while the symbols are the numerical values.

$$
\begin{aligned}
\int_E \upsilon \partial_t u_h \, d\Omega &= -\int_E \upsilon \left(\partial_x f + \partial_y g\right) d\Omega + \int_E \upsilon \nabla \cdot (\nu \mathbf{D}) \, d\Omega, \\
&= -\oint_{\partial\Omega} \upsilon \mathbf{H} \cdot \hat{\mathbf{n}} \, d\Gamma + \int_E \nabla \upsilon \cdot \mathbf{H} \, d\Omega + \underline{\oint_{\partial\Omega} \upsilon (\nu \mathbf{D}) \cdot \hat{\mathbf{n}} \, d\Gamma - \int_E \nabla \upsilon \cdot (\nu \mathbf{D}) \, d\Omega}, \\
&= -\sum_{e \in \partial\Omega} \int_e \upsilon \mathbf{H} \cdot \hat{\mathbf{n}} \, d\Gamma + \int_E \nabla \upsilon \cdot \mathbf{H} \, d\Omega + \underline{\sum_{e \in \partial\Omega} \int_e \upsilon (\nu \mathbf{D}) \cdot \hat{\mathbf{n}} \, d\Gamma - \int_E \nabla \upsilon \cdot (\nu \mathbf{D}) \, d\Omega}, \\
&\approx -\sum_{e \in \partial\Omega} \sum_{\chi_l} w_l \, \upsilon_l \, (\hat{\mathbf{H}}_l - \underline{(\nu \hat{\mathbf{D}})_l}) \cdot \hat{\mathbf{n}} \, \Delta s_e + \sum_{\chi_m} w_m \left[\nabla \upsilon_m \cdot (\mathbf{H}_m - \underline{(\nu \nabla_h u_h)_m}\right] A_E, \qquad \text{(A.7)} \\
&= -\sum_{e \in \partial\Omega} \sum_{\chi_l} w_l \left[\upsilon_l \, (\hat{\mathbf{H}}_l - \underline{(\nu \hat{\mathbf{D}})_l}) + \underline{(\nu \hat{u})_l \nabla \upsilon_l}\right] \cdot \hat{\mathbf{n}} \, \Delta s_e \\
&+ \sum_{\chi_m} w_m \left[\nabla \upsilon_m \cdot \mathbf{H}_m + \underline{u_m \nabla \upsilon_m \cdot \nabla \nu_m + u_m \Delta \upsilon_m}\right] A_E, \qquad \text{(A.8)}
\end{aligned}
$$

where the second integration by part is performed based on the relation $\nabla \upsilon \cdot (\nu \nabla u) = \nabla \cdot (\nu u \nabla \upsilon) - u \nabla \upsilon \cdot \nabla \nu - u \Delta \upsilon$, where Δ denotes Laplacian operator. For clarity, we added an underline to the terms associated with diffusion. Note that, it is often easier, for implementation, to work with Eq. (A.7) than Eq. (A.8).

We evaluate the numerical flux $\hat{\mathbf{H}}$ with the LLxF formulation given in Eq. (15), and employ the IP approach to evaluate $\hat{u}$ and the diffusion term $\hat{\mathbf{D}}$ as

$$
\hat{u} = \frac{1}{2}(u_h^+ + u_h^-), \quad \hat{\mathbf{D}} = \frac{1}{2}\left(\nabla_h u_h^+ + \nabla_h u_h^-\right) - \eta (u^+ - u^-) \hat{\mathbf{n}},
$$

where u_h is represented with a polynomial approximation of order k. The IP coefficient, η, is a grid dependent term, which we use the explicit expression derived and introduced by Shahbazi [75] as

$$\eta = \begin{cases} \frac{(k+1)(k+d)}{2d} \max(\frac{l^+}{A^+}, \frac{l^-}{A^-}), & \text{interior edges} \\ \frac{(k+1)(k+d)}{d} \frac{l^-}{A^-}, & \text{boundary edges} \end{cases},$$

where $d = 2$ for two-dimensional elements, l is the perimeter (or area in 3D), and A is the area (or volume in 3D) of the element. The IP scheme reduces to the original scheme of Bassi and Rebay [12] with $\eta = 0$.

References

[1] H. Nishikawa, A first-order system approach for diffusion equation. I: Second order residual distribution schemes, J. Comput. Phys. 227 (2007) 315–352.
[2] H. Nishikawa, A first-order system approach for diffusion equation. II: Unification of advection and diffusion, J. Comput. Phys. 229 (2010) 3989–4016.
[3] A. Mazaheri, H. Nishikawa, First-order hyperbolic system method for time-dependent advection–diffusion problems, Technical Report TM–2014–218175, NASA, March 2014.
[4] A. Mazaheri, H. Nishikawa, Very efficient high-order hyperbolic schemes for time-dependent advection–diffusion problems: third-, fourth-, and sixth-order, Comput. Fluids 102 (2014) 131–147.
[5] H. Nishikawa, First-, second-, and third-order finite-volume schemes for diffusion, J. Comput. Phys. 256 (2014) 791–805.
[6] H. Nishikawa, First, second, and third order finite-volume schemes for advection–diffusion, J. Comput. Phys. 273 (2014) 287–309.
[7] A. Mazaheri, H. Nishikawa, Improved second-order hyperbolic residual-distribution scheme and its extension to third-order on arbitrary triangular grids, J. Comput. Phys. 300 (2015) 455–491.
[8] H. Nishikawa, Philip L. Roe, Third-order active-flux scheme for advection diffusion: hyperbolic diffusion, boundary condition, and Newton solver, Comput. Fluids 125 (2016) 71–81.
[9] A. Mazaheri, H. Nishikawa, High-order shock-capturing hyperbolic residual-distribution schemes on irregular triangular grids, Comput. Fluids 131 (2016) 29–44.
[10] H. Nishikawa, Alternative formulations for first-, second-, and third-order hyperbolic Navier–Stokes schemes, in: 22nd AIAA Computational Fluid Dynamics Conference, Dallas, TX, 2015, AIAA Paper 2015-2451.
[11] D. Drikakis, M. Hahn, A. Mosedale, B. Thornber, Large eddy simulation using high-resolution and high-order methods, Philos. Trans. R. Soc. A 367 (2009) 2985–2997.
[12] F. Bassi, S. Rebay, A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations, J. Comput. Phys. 131 (1997) 267–279.
[13] F. Bassi, S. Rebay, G. Mariotti, S. Pedinotti, M. Savini, A high-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flows, in: Proceedings of 2nd European Conference on Turbomachinery, Fluid Dynamics and Thermodynamics, Technologisch Instituut, Antwerpen, Belgium, 1997, pp. 99–108.
[14] J.R. Douglas, T. Dupont, Interior penalty procedures for elliptic and parabolic Galerkin methods, in: Lecture Notes in Phys., vol. 58, Springer-Verlag, Berlin, 1976.
[15] G.A. Baker, Finite element method for elliptic equations using nonconforming elements, Math. Comput. 31 (1977) 45–59.
[16] D. Arnold, An interior penalty finite element method with discontinuous elements, PhD thesis, The University of Chicago, Chicago, IL, 1979.
[17] D. Arnold, An interior penalty finite element method with discontinuous elements, SIAM J. Numer. Anal. 19 (1982) 742–760.
[18] G.A. Baker, W.N. Jureidini, O.A. Karakashian, Piecewise solenoidal vector fields and the Stokes problem, SIAM J. Numer. Anal. 27 (1990) 1466–1485.
[19] T. Rusten, P.S. Vassilevski, R. Winther, Interior penalty preconditioners for mixed finite element approximations of elliptic problems, Math. Comput. 65 (1996) 447–466.
[20] B. Rivieře, M.F. Wheeler, V. Girault, Improved energy estimates for interior penalty, constrained and discontinuous Galerkin methods for elliptic problems I, Comput. Geosci. 3 (1999) 337–360.
[21] D.N. Arnold, F. Brezzi, B. Cockburn, Unified analysis of discontinuous Galerkin methods for elliptic problems, SIAM J. Numer. Anal. 39 (2002) 1749–1779.
[22] B. Cockburn, C.-W. Shu, The local discontinuous Galerkin method for time-dependent convection–diffusion systems, SIAM J. Numer. Anal. 35 (1998) 2440–2463.
[23] P. Castillo, B. Cockburn, I. Perugia, D. Schötzau, An a priori error analysis of the local discontinuous Galerkin method for elliptic problems, SIAM J. Numer. Anal. 38 (2000) 1676–1706.
[24] B. Cockburn, C. Dawson, Approximation of the velocity by coupling discontinuous Galerkin and mixed finite element methods for flow problems, Comput. Geosci. 6 (2002) 505–522.
[25] B. Cockburn, G. Kanschat, D. Schötzau, A locally conservative LDG method for the incompressible Navier–Stokes equations, Math. Comput. 74 (2005) 1067–1085.
[26] P. Castillo, A review of the local discontinuous Galerkin (LDG) method applied to elliptic problems, Appl. Numer. Math. 56 (2006) 1307–1313.
[27] Y. Xu, C.-W. Shu, Local discontinuous Galerkin methods for high-order time-dependent partial differential equations, J. Comput. Phys. 7 (2010) 1–46.
[28] J. Peraire, P.-O. Persson, The compact discontinuous Galerkin (CDG) method for elliptic problems, SIAM J. Sci. Comput. 30 (2008) 1806–1824.
[29] S. Brdar, A. Dedner, R. Klökorn, Compact and stable discontinuous Galerkin methods for convection–diffusion problems, SIAM J. Sci. Comput. 34 (2012) A263–A282.
[30] H. Liu, J. Yan, The direct discontinuous Galerkin (DDG) methods for diffusion problems, SIAM J. Numer. Anal. 47 (2009) 475–698.
[31] H. Liu, J. Yan, The direct discontinuous Galerkin (DDG) methods for diffusion with interface corrections, Commun. Comput. Phys. 8 (2010) 541–564.
[32] B. Cockburn, J. Gopalakrishnan, R. Lazarov, Unified hybridization of discontinuous Galerkin, mixed and continuous Galerkin methods for second order elliptic problems, SIAM J. Numer. Anal. 47 (2009) 1319–1365.
[33] N.C. Nguyen, J. Peraire, B. Cockburn, An implicit high-order hybridizable discontinuous Galerkin method for linear convection-diffusion equations, J. Comput. Phys. 228 (2009) 3232–3254.
[34] N.C. Nguyen, J. Peraire, B. Cockburn, An implicit high-order hybridizable discontinuous Galerkin method for nonlinear convection–diffusion equations, J. Comput. Phys. 228 (2009) 8841–8855.
[35] N.C. Nguyen, J. Peraire, B. Cockburn, A hybridizable discontinuous Galerkin method for Stokes flow, Comput. Methods Appl. Mech. Eng. 199 (2010) 582–597.
[36] B. Cockburn, The hybridizable discontinuous Galerkin methods, in: International Congress of Mathematicians, Hyderabad, India, 2010.
[37] B. van Leer, S. Nomura, Discontinuous Galerkin for diffusion, in: 17th AIAA Computational Fluid Dynamics Conference, Toronto, Ontario, Canada, 2005, AIAA Paper 2005-5108.
[38] B. van Leer, M. Lo, M. van Raalte, A discontinuous Galerkin method for diffusion based on recovery, in: 18th AIAA Computational Fluid Dynamics Conference, Miami, FL, 2007, AIAA Paper 2007-4083.
[39] B. van Leer, M. Lo, Unification of discontinuous Galerkin methods for advection and diffusion, in: 47th AIAA Aerospace Sciences Meeting and Exhibit, Orlando, FL, 2009, AIAA Paper 2009-400.
[40] Kwok Ho Marcus Lo, A space–time discontinuous Galerkin method for Navier–Stokes with recovery, PhD thesis, The University of Michigan, MI, 2011.

[41] W.H. Reed, T.R. Hill, Triangular mesh methods for the neutron transport equation, Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.
[42] B. van Leer, Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection, J. Comput. Phys. 23 (1977) 276–299.
[43] B. Cockburn, C.-W. Shu, TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws II. General framework, Math. Comput. 52 (1989) 411–435.
[44] B. Cockburn, C.-W. Shu, The Runge–Kutta discontinuous Galerkin method for conservation laws V, J. Comput. Phys. 141 (1998) 199–224.
[45] B. Cockburn, C.-W. Shu, Runge–Kutta discontinuous Galerkin methods for convection-dominated problems, J. Comput. Phys. 16 (2001) 173–261.
[46] 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, Honolulu, Hawaii, 2011, AIAA Paper 2011-3043.
[47] A. Mazaheri, M. Ricchiuto, H. Nishikawa, A first-order hyperbolic system approach for dispersion, J. Comput. Phys. 321 (2016) 593–605.
[48] Y. Nakashima, N. Watanabe, H. Nishikawa, Hyperbolic Navier–Stokes solver for three-dimensional flows, in: 54th AIAA Aerospace Sciences Meeting, San Diego, CA, 2016, AIAA Paper 2016-1101.
[49] H.T. Huynh, A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods, in: 18th AIAA Computational Fluid Dynamics Conference, Miami, FL, 2007, AIAA Paper 2007-4079.
[50] H.T. Huynh, A reconstruction approach to high-order schemes including discontinuous Galerkin for diffusion, in: 47th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, Orlando, FL, 2009, AIAA Paper 2009-403.
[51] D. De Grazia, G. Mengaldo, D. Moxey, P.E. Vincent, S. Sherwin, Connections between the discontinuous Galerkin method and high-order flux reconstruction schemes, Int. J. Numer. Methods Fluids 75 (2014) 860–877.
[52] H. Luo, L. Luo, R. Nourgaliev, V.A. Mousseau, N. Dinh, A reconstructed discontinuous Galerkin method for the compressible Navier–Stokes equations on arbitrary grids, J. Comput. Phys. 229 (2010) 6961–6978.
[53] B. Cockburn, F. Li, C.-W. Shu, Locally divergence-free discontinuous Galerkin methods for the Maxwell equations, J. Comput. Phys. 194 (2004) 588–610.
[54] F. Li, C.-W. Shu, Locally divergence-free discontinuous Galerkin methods for MHD equations, J. Sci. Comput. 22 (2005) 413–442.
[55] S. Yakovlev, L. Xu, F. Li, Locally divergence-free discontinuous Galerkin methods for ideal MHD equations, J. Comput. Sci. 4 (2013) 80–91.
[56] J. Zhu, J. Qui, C.-W. Shu, M. Dumbser, Runge–Kutta discontinuous Galerkin method using WENO limiters II: unstructured meshes, J. Comput. Phys. 227 (2008) 4330–4353.
[57] J. Zhu, X. Zhong, C.-W. Shu, J. Qiu, Runge–Kutta discontinuous Galerkin method using a new type of WENO limiters on unstructured meshes, J. Comput. Phys. 248 (2013) 200–220.
[58] J. Qui, C.-W. Shu, One the construction, comparison, and local characteristic decomposition for high-order central WENO schemes, J. Comput. Phys. 183 (2002) 187–209.
[59] J. Qui, C.-W. Shu, Hermite WENO schemes and their application as limiters for Runge–Kutta discontinuous Galerkin method: one-dimensional case, J. Comput. Phys. 193 (2003) 115–135.
[60] G. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1996) 202–228.
[61] C. Hu, C.-W. Shu, Weighted essentially non-oscillatory schemes on triangular meshes, J. Comput. Phys. 150 (1999) 97–127.
[62] M. Käser, A. Iske, ADER schemes on adaptive triangular meshes for scalar conservation laws, J. Comput. Phys. 205 (2005) 486–508.
[63] X. Zhong, C.-W. Shu, A simple weighted essentially nonoscillatory limiter for Runge–Kutta discontinuous Galerkin method, J. Comput. Phys. 232 (2013) 397–415.
[64] M. Dumbser, M. Käser, Arbitrary high-order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems, J. Comput. Phys. 221 (2007) 693–723.
[65] Chi-Wang Shu, Personal communication, Feb. 2016.
[66] L. Krivodonova, J. Xin, J.-F. Remacle, N. Chevaugeon, J.E. Flaherty, Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws, Appl. Numer. Math. 48 (2004) 323–338.
[67] P.R. Amestoy, I.S. Duff, J. Koster, J.-Y. L'Excellent, A fully asynchronous multifrontal solver using distributed dynamic scheduling, SIAM J. Matrix Anal. Appl. 23 (2001) 15–41.
[68] P.R. Amestoy, A. Guermouche, J.-Y. L'Excellent, S. Pralet, Hybrid scheduling for the parallel solution of linear systems, Parallel Comput. 32 (2006) 136–156.
[69] Katate Masatsuka, I do Like CFD, vol. 1, second edition, Self Published, 2013.
[70] C. Johnson, J. Pikäranta, An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation, Math. Comput. 46 (1986) 1–26.
[71] G. Richter, An optimal-order error estimate for the discontinuous Galerkin method, Math. Comput. 50 (1988) 75–88.
[72] J. Qui, M. Dumbser, C.-W. Shu, The discontinuous Galerkin method with Lax–Wendroff type time discretizations, Comput. Methods Appl. Mech. Eng. 194 (2000) 4528–4543.
[73] Xinghui Zhong, Personal communication, Feb. 2016.
[74] David Walfisch, Visualization for high-order discontinuous Galerkin CFD results, Master's thesis, Massachusetts Institute of Technology, MA, 2007.
[75] K. Shahbazi, An explicit expression for the penalty parameter of the interior penalty method, J. Comput. Phys. 205 (2005) 401–407.