10.2514/6.2014-3221

AIAA AVIATION Forum
16-20 June 2014, Atlanta, GA
7th AIAA Theoretical Fluid Mechanics Conference

AIAA-2014-3221

44th AIAA Fluid Dynamics Conference, 16–20 June 2014, Atlanta, GA

Analysis of Improved Advection Schemes for Discontinuous Galerkin Methods

Loc Khieu* and Eric Johnsen†
The University of Michigan, Ann Arbor, MI 48109, USA

Abstract

The Recovery-based Discontinuous Galerkin (RDG) discretization has been shown to be the most accurate among contemporary DG schemes for diffusion on a Cartesian grid. It achieves the order of accuracy $3p+2$ for even p and $3p+1$ for odd p, where p is the order of the polynomial basis. However, the overall performance of a Navier–Stokes simulation, using RDG for viscous terms, is limited to $2p+1$ due to the DG discretization for advection. We describe two different approaches to improve the accuracy of the DG discretization for advection. The first option is able to reach a maximal order of $4p+3$, albeit utilizing an enlarged computational stencil. The second one attains $3p+1$ without enlarging the stencil. It is also computationally cheaper owing to the lower-order reconstruction.

I. Introduction

Discontinuous Galerkin (DG) methods combine advantages of finite volume (FV) and finite element (FE) approaches into a single framework. The Galerkin formulation, borrowed from the FE methodology, produces a high-order accurate and compact-stencil scheme with several advantages over high-order FV and finite difference (FD) methods, such as their portability to complex geometries, scalability in parallel architecture and relatively simple extension to high order. The discontinuous nature of the basis functions in each cell introduces means for adding numerical dissipation in the convective terms, such that, unlike traditional FE schemes, discontinuous solutions can be captured in a stable fashion. The DG method was originally introduced for solving the steady-state neutron transport equation on triangular meshes,[1] and later analyzed.[2–5] The introduction of the Runge-Kutta DG (RKDG) method by Cockburn and Shu[6–9] made this approach appealing for time-dependent convection-dominated problems. The RKDG method was shown to be well-suited to handle shocks through the use of slope limiters in one and two dimensions, on rectangular and triangular meshes. Similar to FV schemes, DG methods use Riemann solvers to determine the inter-cell flux and thus introduce the appropriate amount of dissipation at the discontinuities.

However, the main difficulty preventing a simple extension to solving diffusive terms (i.e., second-order derivatives or differences), such as in the Navier-Stokes equations, is the fact that derivatives are undefined at cell interfaces due to the discontinuity in the basis functions between neighboring cells. Two main approaches have been followed in the past: one in which gradient

[*] Postdoctoral Fellow, Aerospace Engineering Department, AIAA Member. Email: khieuhl@umich.edu

[†] Assistant Professor, Mechanical Engineering Department, AIAA Member. Email: ejohnsen@umich.edu

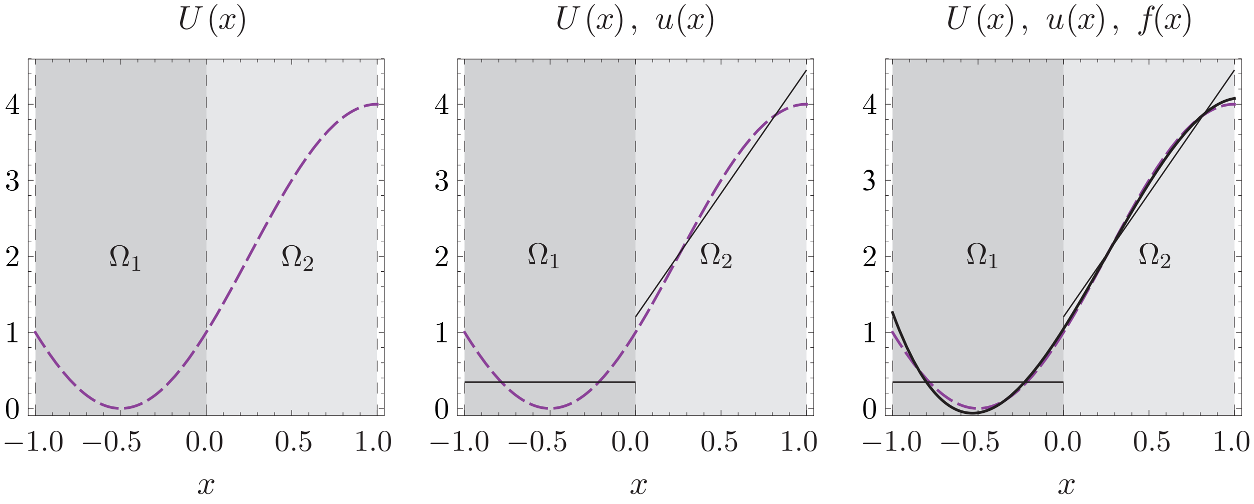


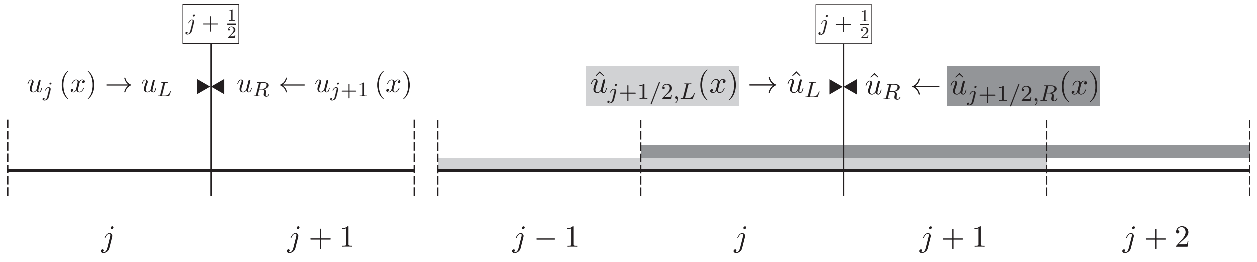
Figure 1: Recovery in one dimension for a piecewise-linear discretization ($p = 1$) on two adjacent intervals $(-1, 0)$ and $(0, 1)$. Shown are, from left to right, the original sin initial value $U(x)$ (*dashed*), its piecewise-linear projection $u(x)$ (*thin* and *solid*), together with $U(x)$, and the cubic recovered function $f(x)$ (*thick* and *solid*) together with $u(x)$ and $U(x)$. All three distributions yield the same value when taking their inner product with either test function on either interval, making them indistinguishable in the weak sense.

information is provided and which is stabilized by interior penalty terms or artificial diffusion with adjustable parameters; and another more recent based on the idea of recovery of the underlying function over neighboring cells. With the former, interior penalty methods have been used for elliptic and parabolic problems.[10–12] Another class of methods treat the second-order partial differential equations as a system of first-order equations.[13–16] Such methods have been used to solve for diffusive terms, but there is no consistent methodology and adjustable parameters are required.

In recovery,[17–21] the discontinuity at the interface is removed in the weak sense by a local polynomial required to satisfy moments with the original solution in the elements that span the interface. A typical one-dimensional example is shown in Figure (1). The smooth recovered function is used to compute the necessary function values and derivatives at the interface. The resulting recovery-based discontinuous Galerkin method (RDG) is proved to be stable.[17] Utilizing a p-order elemental tensor-product polynomial basis, RDG has been demonstrated to achieve the order $3p+2$ or $3p + 1$ for p even or odd, respectively,[17] on a Cartesian grid. Huynh[22] shows that RDG is the most accurate, as well as have the most favorable stability restriction, in a comparison study of all contemporary DG schemes for diffusion. The result is robust: it holds in any number of dimensions, for linear as well as nonlinear equations, with or without mixed derivatives. Lo[18,23] was the first to show the optimal accuracy for a 2-D diffusion-shear operator and for the 2-D Navier-Stokes terms; Johnsen et al.[24–26] showed it for 3-D turbulence calculations.

We are interested in utilizing RDG in DNS simulations, e.g., compressible isotropic turbulence and Taylor–Green vortex that were presented in 24. Owing to the simplicity in computational domains, a Cartesian grid is sufficient, plus it brings out the best in RDG. However, the current overall accuracy of the simulations is seriously impaired by the DG discretization for the advection terms due to its lower order of accuracy ($2p + 1$ comparing to $3p + 2/3p + 1$ of RDG). To benefit fully from RDG, improvement to the advection discretization has to be made.

This paper is organized as follows. In Section II, we present our concepts of improving advection discretization. Methodology of Von Neumann analysis and obtained results are shown in Section III. Our conclusions are presented in Section IV, together with further developments.


$j+\frac{1}{2}$
$u_j(x) \rightarrow u_L$
$u_R \leftarrow u_{j+1}(x)$
j
$j+1$
$j+\frac{1}{2}$
$\hat{u}_{j+1/2,L}(x) \rightarrow \hat{u}_L$
$\hat{u}_R \leftarrow \hat{u}_{j+1/2,R}(x)$
$j-1$
j
$j+1$
$j+2$

(a) Compact stencil

(b) Enlarged stencil

Figure 2: Stencils of flux computation at an arbitrary interface.

II. Concepts of Improving DG Discretization for Advection

We start out by enhancing performance of DG discretization for the one-dimensional linear scalar advection equation

$$\partial_t u + \partial_x F(u) = 0, \tag{1}$$

where $F(u) = au$ and advection speed a is assumed to be positive without loss of generality. The flux at an arbitrary interface $j+1/2$ is calculated by upwinding from the left and right values of u at that interface, u_L and u_R respectively. In the basic DG scheme for advection, they are calculated from the numerical representations $u_j(x)$ and $u_{j+1}(x)$ as shown schematically in Figure (2a). The resulted scheme achieves the order of accuracy $2p+1$. The overall computational stencil to evolve solution in cell j contains three cells: $j-1$, j and $j+1$.

To make improvement upon the accuracy *within* the upwinding framework, we have to raise the polynomial order of the numerical representations from which u_L and u_R are calculated. The higher-order representations $\hat{u}_{j+1/2,L}(x)$ and $\hat{u}_{j+1/2,R}(x)$ will be reconstructed from available data in the vicinity of the interested interface. This is the standard technique employed in high-resolution FV methods, such as MUSCL.[27] This technique has been used successfully in DG+MUSCL[26] to gain one extra order of accuracy quickly and cheaply. Below, we will present two different *systematic* approaches to attain our goal of having comparable accuracy with that of RDG: *cell-centered* and *interface-centered* reconstructions.

II.A. Cell-centered reconstruction

In this approach, the higher-order polynomial representation $\hat{u}_{j+1/2,L}(x)$ is reconstructed using data from three cells $j-1$, j and $j+1$. Similarly, the reconstruction of $\hat{u}_{j+1/2,R}(x)$ uses data from three cells j, $j+1$ and $j+2$. Then the required interface values u_L and u_R are calculated from those as illustrated in Figure (2b). Repeating the same procedure at interface $j-1/2$ will show that indeed $\hat{u}_{j-1/2,R}(x)$ is identical to $\hat{u}_{j+1/2,L}(x)$. Thus, there is one enhanced representation $\hat{u}_j$ that is *unique* to cell j; the reconstruction is therefore called *cell-centered*.

The reconstructed $\hat{u}$ is required to be indistinguishable from the original u_{j-1}, u_j and u_{j+1} in the weak sense, that is,

$$\int_{x_{j-3/2}}^{x_{j-1/2}} (v_k)_{j-1}\, \hat{u}_j \, dx = \int_{x_{j-3/2}}^{x_{j-1/2}} (v_k)_{j-1}\, u_{j-1}\, dx, \quad k = 0, ..., K_{j-1}, \tag{2}$$

$$\int_{x_{j-1/2}}^{x_{j+1/2}} (v_k)_{j}\, \hat{u}_j \, dx = \int_{x_{j-1/2}}^{x_{j+1/2}} (v_k)_{j}\, u_{j}\, dx, \quad k = 0, ..., p, \tag{3}$$

$$\int_{x_{j+1/2}}^{x_{j+3/2}} (v_k)_{j+1}\, \hat{u}_j \, dx = \int_{x_{j+1/2}}^{x_{j+3/2}} (v_k)_{j+1}\, u_{j+1}\, dx, \quad k = 0, ..., K_{j+1}. \tag{4}$$

To satisfy these equations, $\hat{u}$ must contain $K_{j-1} + p + K_{j+1} + 3$ degrees of freedom to form a polynomial of degree $\hat{p} = K_{j-1} + p + K_{j+1} + 2$, defined on the domain $\left[x_{j-3/2}, x_{j+3/2}\right]$. Basis functions $(v_k)_{j-1}$, $(v_k)_j$ and $(v_k)_{j+1}$ define specific moments of the original solutions (u_{j-1}, u_j, u_{j+1}) that $\hat{u}$ has to preserve on their corresponding cells.

In theory, values of K_{j-1} and K_{j+1} can be different that leads to a *bias* reconstruction. In the context of linear scalar advection, one can argue for using upwind-bias reconstruction by utilizing more data from the left cell than from the right one ($K_{j-1} > K_{j+1}$). For a system of advection equations, however, the wave structure at the interface is more complex with both left- and right-traveling waves. It is thus better not to favor any direction at this reconstruction step; from now on we take $K_{j-1} = K_{j+1} = K$. The reader may find this particular scheme familiar; it is indeed the $P_N P_M$ method invented by Dumbser.[28–30]

Based on the number of moments from cells $j-1$ and $j+1$ to be preserved in $\hat{u}_j$, we have considered two options for the reconstruction.

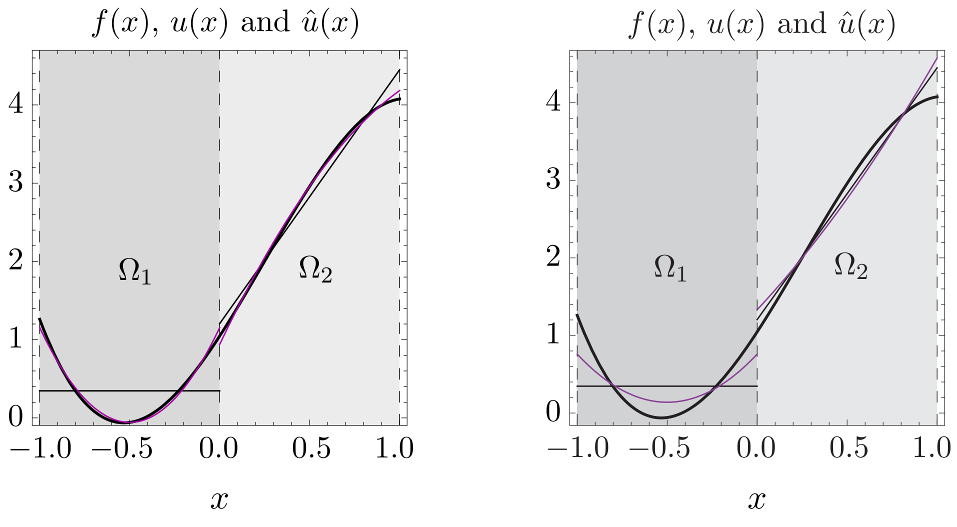
1. *Full reconstruction.* All $p+1$ moments from each neighbor is preserved, i.e. $K = p$, making $\hat{u}_j$ a polynomial of degree $3p+2$. We anticipate that the order-of-accuracy will increase to $4p+3$, comparing to $2p+1$ of the basic scheme; the difference is precisely the amount of additional information taken from the neighbors. In the terminology of the $P_N P_M$ method, this option corresponds to the $P_p P_{3p+2}$ scheme. This option is called `ccf`# which denotes cell-centered full reconstruction at $p = \#$. Interchangeably, the term `ccf` might be used to indicate this scheme in a generic case.

2. *Partial reconstruction.* The previous option is an overkill as the accuracy of RDG for diffusion is only $3p+2/3p+1$ for even/odd value of p. We expect to match the accuracy of RDG more closely by using only a subset of available neighbor information. Specifically, we choose to preserve $\frac{p+2}{2}$ or $\frac{p+1}{2}$ moments (for p even or odd, respectively) from each neighbor cell to achieve the accuracy of $3p+3/3p+2$, making $\hat{u}_j$ a polynomial of degree $2p+2/2p+1$. In the $P_N P_M$ terminology, this option corresponds to $N = p$ and $M = 2p+2/2p+1$. However, this is not strictly a $P_N P_M$ as formulated by Dumbser et al.,[28–30] in which $\hat{u}_j$ is required to always preserve all available moments of u_{j-1}, u_j, and u_{j+1} on their respective domains. This generates $3p+3$ conditions while, by construction, $\hat{u}_j^{p'}$ has only $2p+3$ degrees-of-freedom for even p and $2p+2$ for odd p, leading to a overdetermined system that has to be solved by least-square. Our approach is therefore simpler and sufficiently accurate still. By the same token, this option is denoted `ccp`# to indicate cell-centered partial reconstruction at $p = \#$, or `ccp` for the generic case.

One apparent disadvantage of the cell-centered reconstruction is the enlarged five-cell overall stencil, even though the stencil stays unchanged regardless of p value. It is highly preferable to retain the original compact 3-cell stencil while improving the discretization. We will present our ideas for this matter in the next subsection.

II.B. Interface-centered reconstruction

To keep the overall stencil unchanged from three cells, the calculation for flux at interface $j+1/2$ should involve only data from its direct neighbors as shown in Figure (2a). The reconstruction of $\hat{u}_{j+1/2,L}(x)$ and $\hat{u}_{j+1/2,R}(x)$ therefore uses information only from cells j and $j+1$. Similarly, that of $\hat{u}_{j-1/2,L}(x)$ and $\hat{u}_{j-1/2,R}(x)$ contains information only form cells $j-1$ and j. Now defined on the domain of cell j, there are two different enhanced representations, $\hat{u}_{j-1/2,R}$ and $\hat{u}_{j+1/2,L}$, that strictly associates with interfaces $j-1/2$ and $j+1/2$, respectively. In other words, they are *centered* at their corresponding interfaces, thus the name *interface-centered reconstruction* is used.



(a) Down Projection (`icd1[−1]`)

(b) Binary Reconstruction (`icb1[0]`)

Figure 3: Illustration of the down-projection and binary reconstructions for $p = 1$. The recovered function f (solid **thick** black line) and the piecewise-linear representations u (solid *thin* black line) are carried over from Figure (1). Note that $\hat{u}(x)$ (solid *thin* magenta line) from both down-projection and binary reconstructions are discontinuous across $x = 0$ interface as expected.

Our first idea is to reuse the recovered function $f(x)$ that is already calculated for the diffusion discretization. As shown in Figure (1), f is centered at the interface of interested and its reconstruction involves only two direct neighbor cells. However, to directly use $f_{j+2/2}$ for flux calculation at interface $j+1/2$ is essentially similar to central discretization, which has undesired stability property for advection problem.[31] To create $\hat{u}_{j+1/2,L}$ and $\hat{u}_{j+1/2,R}$ that are discontinuous at $x_{j+1/2}$, f is down projected to two lower-order polynomials defined on cells j and $j+1$. Recall that f is a polynomial of degree $2p+1$; it can be down projected to polynomials of degrees $\hat{p}$, where $p+1 \leq \hat{p} \leq 2p$.

$$\int_{x_{j-1/2}}^{x_{j+1/2}} (v_k)_j \, \hat{u}_{j+1/2,L} dx = \int_{x_{j-1/2}}^{x_{j+1/2}} (v_k)_j \, f_{j+1/2} dx, \quad k = 0, ..., \hat{p}, \tag{5}$$

$$\int_{x_{j+1/2}}^{x_{j+3/2}} (v_k)_{j+1} \, \hat{u}_{j+1/2,R} \, dx = \int_{x_{j+1/2}}^{x_{j+3/2}} (v_k)_{j+1} \, f_{j+1/2} \, dx, \quad k = 0, ..., \hat{p}. \tag{6}$$

The recovered function f is not down projected to polynomials of degree p because it results in u_j and u_{j+1}, which form the basic scheme.

Resulted schemes will be called `icd`$\#_1$`[`$-\#_2$`]` standing for interface-centered down projection at $p = \#_1$; its generic name is `icd`. The number between the square brackets, $\#_2$, denotes the reduction amount in polynomial degree of $u_{j+1/2,L/R}$ comparing to f. For example, scheme `icd1[−1]` has f down projected to polynomials of one-degree less, i.e., $u_{j+1/2,L/R}$ are $2p^{\text{th}}$-order polynomials. Figure (3a) illustrates two enhanced representations, $\hat{u}_{0,L}$ and $\hat{u}_{0,R}$, associated with interface $x = 0$ of scheme `icd1[−1]`, for $p = 1$ discretization; the recovered function f and the piecewise-linear representations are carried over from example in Figure (1).

We expect that the central scheme resulted from direct usage of f achieve the order

$$(2p+1) + (p+1) = 3p+2.$$

The order of accuracy of an arbitrary `icd`$\#_1$`[`$-\#_2$`]` scheme will then be $3p+2-\#_2$, and the only viable scheme is `icd`$\#_1$`[−1]`. It matches RDG scheme for odd p but achieves one-order lower when p is even.

The second idea of compact reconstruction is inspired by the recovery procedure in RDG, with a small modification. The reconstruction for $\hat{u}_{j+1/2,L}$ will now bias towards the left neighbor, cell j, indicating it might preserve more moments of u_j than those of u_{j+1},

$$\int_{x_{j-1/2}}^{x_{j+1/2}} (v_k)_j \, \hat{u}_{j+1/2,L} dx = \int_{x_{j-1/2}}^{x_{j+1/2}} (v_k)_j \, u_j dx, \quad k = 0, ..., p, \tag{7}$$

$$\int_{x_{j+1/2}}^{x_{j+3/2}} (v_k)_{j+1} \, \hat{u}_{j+1/2,L} \, dx = \int_{x_{j+1/2}}^{x_{j+3/2}} (v_k)_{j+1} \, u_{j+1} \, dx, \quad k \in \mathcal{K}, \tag{8}$$

where $\mathcal{K}$ is only a subset of $\{0, ..., p\}$, which are indexes of all moments of the original numerical representations. Similarly, the reconstruction for $\hat{u}_{j+1/2,R}$ biases towards u_{j+1} of the right neighbor,

$$\int_{x_{j-1/2}}^{x_{j+1/2}} (v_k)_j \, \hat{u}_{j+1/2,R} dx = \int_{x_{j-1/2}}^{x_{j+1/2}} (v_k)_j \, u_j dx, \quad k \in \mathcal{K}, \tag{9}$$

$$\int_{x_{j+1/2}}^{x_{j+3/2}} (v_k)_{j+1} \, \hat{u}_{j+1/2,R} \, dx = \int_{x_{j+1/2}}^{x_{j+3/2}} (v_k)_{j+1} \, u_{j+1} \, dx, \quad k = 0, ..., p, \tag{10}$$

The same subset $\mathcal{K}$ will be used in both Eqs. (8) and (9). There are at most p elements in $\mathcal{K}$ and they are not required to be sequential. For example, $\mathcal{K} = \{0, 2\}$ indicates that $\hat{u}_{j+1/2,L}$ and $\hat{u}_{j+1/2,R}$ will only preserve the 0^{th}-order and 2^{nd}-order moments of u_{j+1} and u_j respectively. All schemes of this family will be called `icb`$\#_1$`[`$\#_2$`]` denoting interface-centered binary reconstruction at $p = \#_1$, and $\#_2$ are all elements of subset $\mathcal{K}$; again, `icb` is used for a generic case. The enhanced representations from scheme `icb1[0]` ($p = 1$) is displayed in Figure (3b).

In terms of order of accuracy, we expect that the binary-reconstruction schemes will achieve a maximal order of $3p+1$ corresponding to $\mathcal{K}$ has p members. There are $p+1$ possibilities to form distinct and orderless subsets of this type from the set $\{0, 1, ..., p\}$, leading to $p+1$ flavors of a $3p+1$-order binary-reconstruction scheme. When its number of members in $\mathcal{K}$ is less than p, the order of accuracy reduces accordingly.

All possible down-projection and binary-reconstruction schemes for $p \in [1, 3]$ are shown in Table 1. In the next section, we use Von Neumann analysis to investigate proposed discretization schemes for $p = 1$ and 2.

III. Von Neumann Analysis

III.A. Methodology

We perform the Fourier analysis for the basic DG scheme ($p = 1$) for the scalar linear advection equation (1). The DG update equations have the following general form:

$$d_t \int_{x_{j-1/2}}^{x_{j+1/2}} v_j u_j \, dx = -\int_{x_{j-1/2}}^{x_{j+1/2}} v_j \; a \, \partial_x u_j \, dx$$

$$\Rightarrow d_t \int_{x_{j-1/2}}^{x_{j+1/2}} v_j u_j \, dx = -a \left[v_j \left(x_{j+1/2} \right) u_{j+1/2} - v_j \left(x_{j-1/2} \right) u_{j-1/2} \right] + a \int_{x_{j-1/2}}^{x_{j+1/2}} u_j \, \partial_x v_j \, dx. \tag{11}$$

Solution in an arbitrary cell j is represented by a polynomial of degree one,

$$u_j(x) = \sum_{\alpha=0}^{1} a_j^{(\alpha)} \phi_j^{(\alpha)}(x) = a_j^{(0)} \phi_j^{(0)} + a_j^{(1)} \phi_j^{(1)} = a_j^{(0)} + a_j^{(1)} (2\xi - 1), \tag{12}$$

where the local spatial variable $\xi \in [0, 1]$ relates to the global coordinate by

$$\xi = \frac{x - x_j}{\Delta x_j} - \frac{1}{2};$$

Table 1: Down-projection and binary-reconstruction schemes.

p	Down Projection		Binary Reconstruction		
	$\hat{p}$	Scheme names	$\hat{p}$	$\mathcal{K}$	Scheme names
1	2	icd1[−1]	2	$\{0\}$	icb1[0]
				$\{1\}$	icb1[1]
2	4	icd2[−1]	4	$\{0,1\}$	icb2[0, 1]
				$\{0,2\}$	icb2[0, 2]
				$\{1,2\}$	icb2[1, 2]
	3	icd2[−2]	3	$\{0\}$	icb2[0]
				$\{1\}$	icb2[1]
				$\{2\}$	icb2[2]
3	6	icd3[−1]	6	$\{0,1,2\}$	icb3[0, 1, 2]
				$\{0,1,3\}$	icb3[0, 1, 3]
				$\{0,2,3\}$	icb3[0, 2, 3]
				$\{1,2,3\}$	icb3[1, 2, 3]
	5	icd3[−2]	5	$\{0,1\}$	icb3[0, 1]
				$\{0,2\}$	icb3[0, 2]
				$\{0,3\}$	icb3[0, 3]
				$\{1,2\}$	icb3[1, 2]
				$\{1,3\}$	icb3[1, 3]
				$\{2,3\}$	icb3[2, 3]
	4	icd3[−3]	4	$\{0\}$	icb3[0]
				$\{1\}$	icb3[1]
				$\{2\}$	icb3[2]
				$\{3\}$	icb3[3]

subscript j in Δx_j will be subsequently omitted because the analysis is performed on a uniform grid.

The upwind values at the left and right interfaces are:

$$u_{j-1/2} = u_{j-1}\left(x_{j-1/2}\right) = a_{j-1}^{(0)} + a_{j-1}^{(1)},$$
$$u_{j+1/2} = u_{j}\left(x_{j+1/2}\right) = a_{j}^{(0)} + a_{j}^{(1)}.$$

Substituting those to Eqn. (11) together with $v_j^{(1)} = 1$ and $v_j^{(2)} = 2\xi - 1$, we obtain the following

$$\frac{\Delta x}{a}\frac{\partial}{\partial t}\begin{bmatrix} a_j^{(0)} \\ a_j^{(1)} \end{bmatrix} = \underbrace{\begin{bmatrix} -1+T^{-1} & -1+T^{-1} \\ 3\left(1-T^{-1}\right) & -3\left(1+T^{-1}\right) \end{bmatrix}}_{M(T)} \begin{bmatrix} a_j^{(0)} \\ a_j^{(1)} \end{bmatrix}; \tag{13}$$

here T represents forward translation by one cell, $Ta_j^{(\alpha)} = a_{j+1}^{(\alpha)}$ and $T^{-1}a_j^{(\alpha)} = a_{j-1}^{(\alpha)}$. We determine the Fourier transform of the matrix operator $M(T)$ by substituting T for its Fourier transform $e^{i\beta}$,

$$\hat{M}(\beta) = \begin{bmatrix} -1 + e^{-i\beta} & -1 + e^{-i\beta} \\ 3\left(1 - e^{-i\beta}\right) & -3\left(1 + e^{-i\beta}\right) \end{bmatrix}; \tag{14}$$

the Taylor-series expansions of its two eigenvalues are

$$\lambda_1 = -6 + 3\, i\beta + O\left(\beta^2\right), \tag{15}$$

$$\lambda_2 = -i\beta - \frac{i\beta^4}{72} + O\left(\beta^5\right). \tag{16}$$

Matrix $\hat{M}(\beta)$ is an approximation of the exact differential operator in Fourier mode, which is

$$\lambda_{\text{exact}} = -i\,\beta.$$

Eqn. (16) shows that it is represented by λ_2 with an accuracy of third order. This eigenvalue is thus called "good" or "consistent" eigenvalue (see[19]), denoted by λ_{con}.

In analyzing the improved schemes presented in Section II, the form of the update equations is still Eqn. (11). The interface upwind values are instead calculated as

$$u_{j-1/2} = \hat{u}_{j-1/2,L}\left(x_{j-1/2}\right),$$
$$u_{j+1/2} = \hat{u}_{j+1/2,L}\left(x_{j+1/2}\right);$$

the integral term stays unchanged. Results we obtain will be presented in the next section.

III.B. Results and discussion

Information contained in the "consistent" eigenvalues of $\hat{M}(\beta)$ for $p = 1$ and $p = 2$ are listed in Tables 2 and 3, respectively. They confirm values of the order of accuracy that were predicted earlier in Section II. At least the context of a scalar linear advection equation,

- `ccf` is $(4p + 3)^{th}$ order;
- `ccp` achieve the order $3p + 3$ or $3p + 2$ for p even or odd, respectively (note: this scheme is scalable);
- both `icd` and `icb` families of schemes have the highest order of $3p + 1$.

The loci of all $p + 1$ eigenvalues of each scheme are shown in Figs. 4 and 5.

At $p = 1$, all schemes would have no stability issue except for the down-projection enhancement. Its eigenvalues lie entirely on the positive-real half plane; to the best of our knowledge, there is no implicit or explicit time-integration scheme that can stabilize that scheme.

The cell-centered full-reconstruction and partial-reconstruction schemes also have their eigenvalues stayed strictly on the negative-real half plane at $p = 2$, as shown in Figure (5).

At first glance, Figure (5) indicates that schemes `icd2[−1]` of the `icd` family might be useful. Scheme `icd2[−2]`, besides having suboptimal order of accuracy, its eigenvalues show that it is also unstable. Figure (6), however, shows that eigenvalues of `icd2[−1]` does not lie strictly on the negative-real half plane either. Their incursion into the positive side is small enough to be covered by an explicit SSP Runge–Kutta 3-stage (RK3) or 4-stage (RK4) time-integration schemes. For a very small value of CFL number, however, neither of them is able to stabilize `icd2[−1]` completely. In the vicinity of the origin, eigenvalues of RK3 and RK4 stay tangent to the imaginary axis, therefore the "spikes" in the locus of `icd2[−1]` will eventually lie outside. A positive value for the lower

Table 2: Results from Von Neumann analysis for p=1

Scheme	ooa[a]	Taylor-series expansion of λ_{con}
basic[b]	3	$-i\beta - \dfrac{\beta^4}{72} + O\left(\beta^5\right)$
`ccf1`	7	$-i\beta - \dfrac{\beta^8}{19\,600} + O\left(\beta^9\right)$
`ccp1`	5	$-i\beta - \dfrac{11\beta^6}{7\,200} + O\left(\beta^7\right)$
`icd1[−1]`	4	$-i\beta + \dfrac{i\beta^5}{180} + \dfrac{\beta^6}{108} + O\left(\beta^7\right)$
`icb1[0]`	4	$-i\beta + \dfrac{i\beta^5}{180} - \dfrac{\beta^6}{288} + O\left(\beta^7\right)$
`icb1[1]`	4	$-i\beta + \dfrac{i\beta^5}{180} - \dfrac{\beta^6}{432} + O\left(\beta^7\right)$

[a] Order of accuracy

[b] Basic DG discretization for advection

limit of CFL number show that the scheme will eventually become unstable if the advection part is updated *explicitly* in an advection–diffusion simulation. Due to this limited stability and the significant variation in behavior of member schemes, `icd` does not seem to be a viable option.

For $p = 2$ case, all three `icb2[0]`/`icb2[1]`/`icb2[2]` schemes are stable but their accuracy is suboptimal. The most accurate members of this family (`icb2[0,1]`/`icb2[0,2]`/`icb2[1,2]`), however, do have the aforementioned positive lower-limit for CFL number. Figure (6) shows that they have similar positive-side incursions to that of `icd2[−1]`, albeit several times smaller. Owing to this improvement and the consistent variations between lower-order and higher-order members regarding stability property, the `icb` family is worth further exploration.

IV. Conclusions and future developments

We have presented two different approaches to improving accuracy of the DG method for advection using upwind flux. Our goal is to match the accuracy level of the RDG method, which will be used to discretize the viscous terms of the Navier–Stokes equations.

In the first approach named *cell-centered reconstruction*, a unique enhanced representation is reconstructed for each computational cell. The computation uses data from its two left and right neighbor cells. Therefore, the stencil for flux calculation is enlarged from 2 cells of the basic DG discretization to 4 cells, leading to a unfavorable five-cell overall stencil. The order of accuracy for *full reconstruction* option (`ccf`) is as high as $4p + 3$, and that of the less-accurate version *partial reconstruction* (`ccp`) is $3p+3$ or $3p+2$ for p even or odd. The former version is the familiar $P_N P_M$ method, but the latter is slightly different. Von Neumann analysis indicate that both options have good stability property.

Our desire for a compact overall stencil leads to the invention of *interface-centered reconstruction*

Table 3: Results from Von Neumann analysis for p=2

Scheme	ooa[a]	Taylor-series expansion of λ_{con}
basic[b]	5	$-i\beta - \dfrac{\beta^6}{7\,200} + O\left(\beta^7\right)$
`ccf2`	11	$-i\beta - \dfrac{\beta^{12}}{426\,888\,000} + O\left(\beta^{13}\right)$
`ccp2`	9	$-i\beta - \dfrac{103\,\beta^{10}}{457\,228\,800} + O\left(\beta^{11}\right)$
`icd2[−1]`	7	$-i\beta + \dfrac{\beta^8}{793\,800} + O\left(\beta^9\right)$
`icd2[−2]`	6	$-i\beta - \dfrac{i\beta^7}{34\,650} + O\left(\beta^8\right)$
`icb2[0,1]`	7	$-i\beta + \dfrac{\beta^8}{120\,960} + O\left(\beta^9\right)$
`icb2[0,2]`	7	$-i\beta + \dfrac{13\,\beta^8}{1\,058\,400} + O\left(\beta^9\right)$
`icb2[1,2]`	7	$-i\beta + \dfrac{\beta^8}{66\,150} + O\left(\beta^9\right)$
`icb2[0]`	6	$-i\beta + \dfrac{17\,i\beta^7}{403\,200} + O\left(\beta^8\right)$
`icb2[1]`	6	$-i\beta + \dfrac{29\,i\beta^7}{604\,800} + O\left(\beta^8\right)$
`icb2[2]`	6	$-i\beta + \dfrac{i\beta^7}{16\,800} + O\left(\beta^8\right)$

[a] Order of accuracy

[b] Basic DG discretization for advection

approach. Now there are multiple higher-order surrogates for the original numerical representation in a cell, as many as number of interfaces. These enhance representations are generally not the same. Each of the surrogates centers at the associated interface and it is reconstructed using information only from the two cells abutting that interface. The flux-calculation stencil remains two cells and the overall stencil contains 3 cells. This approach leads to two distinct families of schemes: the *down-projection reconstruction* denoted by `icd`, and the *binary reconstruction* denoted by `icb`. The most accurate members in both families achieve the order $3p+1$. They both have unfavorable stability property; the situation of the `icd` family is more severe and we deem it is not worthwhile to explore it further.

For future developments, we will explore options to stabilize the most accurate `icb` schemes by,

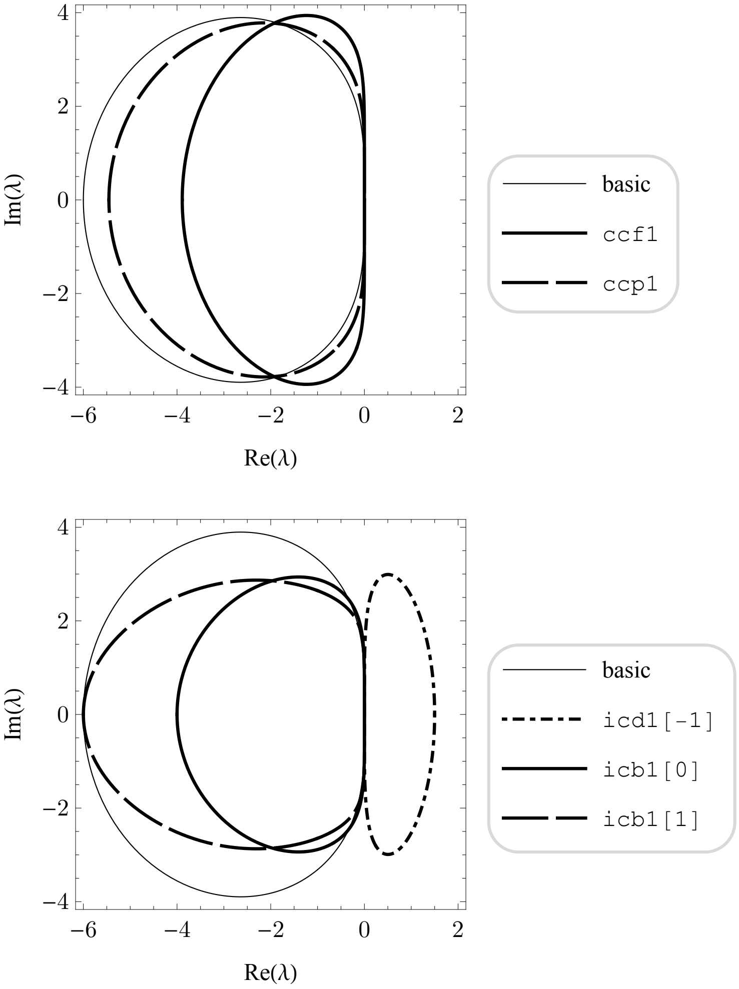


Figure 4: Eigenvalues for $p = 1$.

e.g., artificial dissipation, while still maintain the order $3p+1$. Another direction is to investigate issues of accuracy and stability when simulating nonlinear equations such as the Euler system.

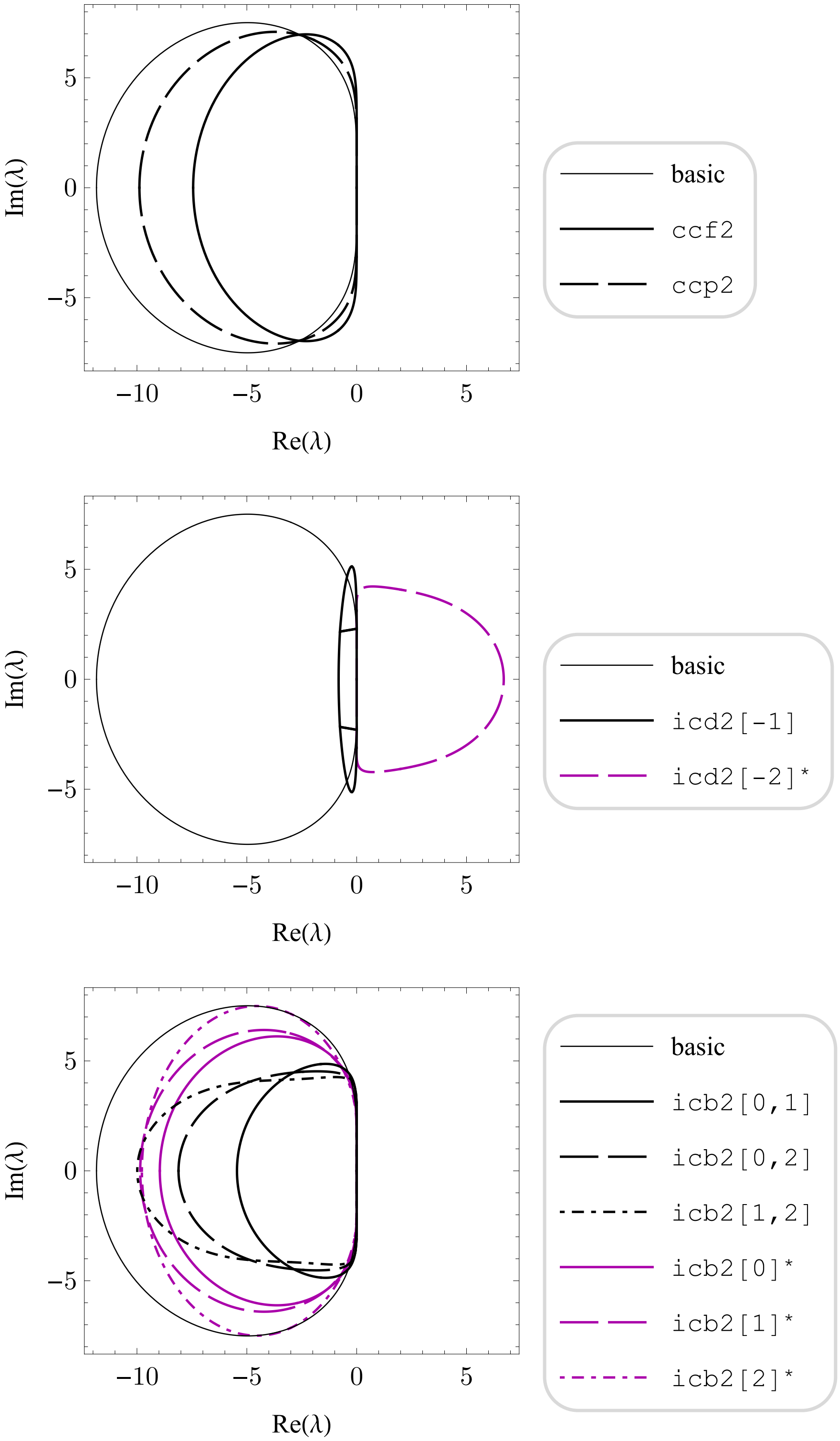


Figure 5: Eigenvalues for $p = 2$. Note that all starred ($\star$) schemes have sub-optimal order of accuracy, see Table 3.

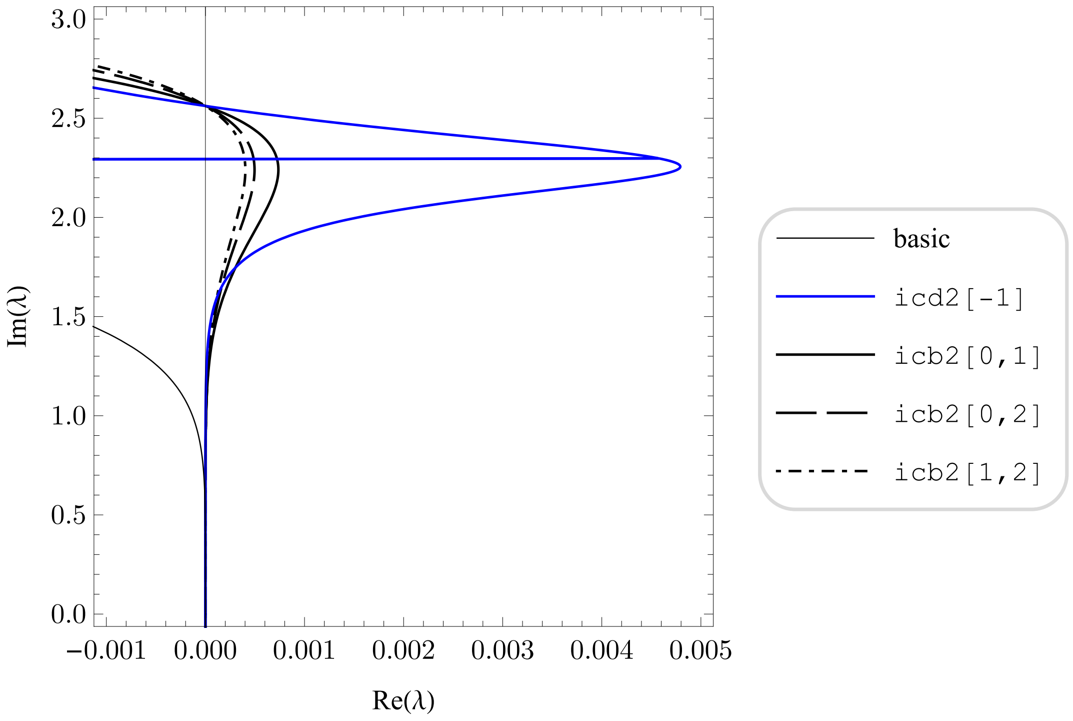


Figure 6: Eigenvalues of few schemes in the vicinity of the imaginary axis, at $p = 2$.

References

[1] Reed, W. and Hill, T., “Triangular mesh methods for the neutron transport equation,” Tech. Rep. LA-UR–73-479, Los Alamos Scientific Laboratory, 1973.

[2] Johnson, C. and Pitkäranta, J., “An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation,” *Mathematics of Computation*, Vol. 46, No. 173, 1986, pp. 1–26.

[3] LeSaint, P. and Raviart, P.-A., “On a finite element method for solving the neutron transport equation,” *Mathematical aspects of finite elements in partial differential equations*, edited by C. De Boor, Academic Press, 1974, p. 89.

[4] Peterson, T. E., “A note on the convergence of the discontinuous Galerkin method for a scalar hyperbolic equation,” *SIAM Journal on Numerical Analysis*, Vol. 28, No. 1, 1991, pp. 133–140.

[5] Richter, G. R., “An optimal-order error estimate for the discontinuous Galerkin method,” *Mathematics of Computation*, Vol. 50, No. 181, 1988, pp. 75–88.

[6] Cockburn, B. and Shu, C.-W., “TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. II. General framework,” *Mathematics of Computation*, Vol. 52, No. 186, 1989, pp. 411–435.

[7] Cockburn, B., Lin, S.-Y., and Shu, C.-W., “TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems,” *Journal of Computational Physics*, Vol. 84, No. 1, 1989, pp. 90–113.

[8] Cockburn, B., Hou, S., and Shu, C.-W., “The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV. The multidimensional case,” *Mathematics of Computation*, Vol. 54, No. 190, 1990, pp. 545–581.

[9] Cockburn, B. and Shu, C.-W., “The RungeKutta discontinuous Galerkin method for conservation laws V: multidimensional systems,” *Journal of Computational Physics*, Vol. 141, No. 2, 1998, pp. 199–224.

[10] Arnold, D. N., Brezzi, F., Cockburn, B., and Marini, L. D., “Unified analysis of discontinuous Galerkin methods for elliptic problems,” *SIAM Journal on Numerical Analysis*, Vol. 39, No. 5, 2002, pp. 1749–1779.

[11] Brezzi, F., Manzini, G., Marini, D., Pietra, P., and Russo, A., “Discontinuous Galerkin approximations for elliptic problems,” *Numerical Methods for Partial Differential Equations*, Vol. 16, No. 4, 2000, pp. 365–378.

[12] Douglas, J. and Dupont, T., “Interior penalty procedures for elliptic and parabolic Galerkin methods,” *Computing methods in applied sciences*, Springer, 1976, pp. 207–216.

[13] Bassi, F. and Rebay, S., "A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations," *Journal of Computational Physics*, Vol. 131, No. 2, 1997, pp. 267–279.

[14] Bassi, F., Rebay, S., Mariotti, G., Pedinotti, S., and Savini, M., "A high-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flows," *Proceedings of 2nd European Conference on Turbomachinery, Fluid Dynamics and Thermodynamics*, Technologisch Instituut, Antwerpen, Belgium, 1997, pp. 99–108.

[15] Cockburn, B. and Shu, C. W., "The local discontinuous Galerkin method for time-dependent convection-diffusion systems," *SIAM Journal on Numerical Analysis*, Vol. 35, No. 6, 1998, pp. 2440–2463.

[16] Peraire, J. and Persson, P. O., "The compact discontinuous Galerkin (CDG) method for elliptic problems," *Siam Journal on Scientific Computing*, Vol. 30, No. 4, 2008, pp. 1806–1824.

[17] Lo, M. and Van Leer, B., "Analysis and Implementation of Recovery-Based Discontinuous Galerkin for Diffusion," 22-25 June 2009, AIAA 2009-3786.

[18] Lo, M. and Van Leer, B., "Recovery-Based Discontinuous Galerkin for Navier-Stokes Viscous Terms," 27-30 June 2011 2011, AIAA 2011-3406.

[19] Van Leer, B. and Nomura, S., "Discontinuous Galerkin for Diffusion," 2005, AIAA-2005-5108.

[20] Van Leer, B., Lo, M., and Van Raalte, M., "A Discontinuous Galerkin Method for Diffusion Based on Recovery," 25–28 June 2007, AIAA 2007-4083.

[21] Van Leer, B. and Lo, M., "Unification of Discontinuous Galerkin Methods for Advection and Diffusion," 5-8 January 2009 2009, AIAA 2009-400.

[22] Huynh, H. T., "A Reconstruction Approach to High-Order Schemes Including Discontinuous Galerkin for Diffusion," *47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition*, Orlando, Florida, US, 2009, AIAA 2009-0403.

[23] Lo, K. H. M., *A space-time discontinuous Galerkin method for Navier-Stokes with recovery*, Ph.D. thesis, 2011.

[24] Johnsen, E., Varadan, S., and Van Leer, B., "A Three-Dimensional Recovery-Based Discontinuous Galerkin Method for Turbulence Simulations," *51st AIAA Aerospace Sciences Meeting*, Grapevine, Texas, 7–10 January 2013, AIAA 2013-0515.

[25] Johnsen, E., Varadan, S., and Nair, A., "Recovery Discontinuous Galerkin Method for Compressible Turbulence," *21st AIAA Computational Fluid Dynamics Conference*, American Institute of Aeronautics and Astronautics, 24–27 June, AIAA 2013-3066.

[26] Johnsen, E., Khieu, L., and Nair, A., "A Simple Method to Improve the Accuracy of Advection in Discontinuous Galerkin Methods for Navier-Stokes Simulations," *52nd Aerospace Science Meeting*, American Institute of Aeronautics and Astronautics, 13–17 January 2014, AIAA 2014-1276.

[27] Van Leer, B., "Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method," *Journal of Computational Physics*, Vol. 32, No. 1, 1979, pp. 101–136.

[28] Dumbser, M., *Arbitrary high order schemes for the solution of hyperbolic conservation laws in complex domains*, Shaker, 2005.

[29] Dumbser, M. and Munz, C. D., "Arbitrary high order discontinuous Galerkin schemes," *Numerical methods for hyperbolic and kinetic problems*, edited by S. Cordier, T. Goudon, M. Gutnic, and E. Sonnendrücker, Vol. 7 of *IRMA Lectures in Mathematics and Theoretical Physics*, EMS Publishing House, 2005, pp. 295–333.

[30] Dumbser, M., Balsara, D. S., Toro, E. F., and Munz, C. D., "A unified framework for the construction of one-step finite volume and discontinuous Galerkin schemes on unstructured meshes," *Journal of Computational Physics*, Vol. 227, No. 18, 2008, pp. 8209–8253.

[31] Hirsch, C., *Numerical Computation of Internal and External Flows*, Vol. 1: Fundamental of Computational Fluid Dynamics, Butterworth-Heinemann, 2nd ed., 2007.