Shock Capturing via Limiting for High-Order Methods including Discontinuous Galerkin

H. T. Huynh
NASA Glenn Research Center
Cleveland, OH 44135, USA
huynh@grc.nasa.gov

Abstract

High-order methods, such as discontinuous Galerkin (DG), spectral, and flux reconstruction (FR), are prone to generating unwanted oscillations near shocks and discontinuities. Conventional limiting techniques, while effective in suppressing oscillations near shocks, often compromise accuracy near extrema, where the solution is only first-order accurate. This paper introduces a novel limiting technique for these high-order schemes, aimed at effectively managing shocks while preserving accuracy. The key idea is to expand the standard monotonicity limits to provide "room" near smooth extrema, ensuring that limiting has no effect and thus preserving accuracy. Near a discontinuity, these expanded limits effectively reduce to the original monotonicity limits, suppressing oscillations. Additional motivation is drawn from a formula for the derivative of Radau polynomials, which depicts the behavior of oscillations resulting from discontinuities. This behavior leads to a simplification by applying the limits to the sum of magnitudes of all modes, linear and higher degree. Unlike typical approaches, which rely on successful detection to activate limiting, our limiter depends continuously on the data, thereby avoiding potential issues if detection fails. To reduce computing time, efficient criteria for detecting smooth regions where limiting is unnecessary are presented. Combined with detection, the continuous dependence on the data is lost, but the method is more economical. A notable characteristic of the entire process is its simplicity in both concept and implementation. Numerical tests for advection and Euler equations are conducted to demonstrate the effectiveness of the proposed method.

1 Introduction

As is well-known, popular high-order methods such as discontinuous Galerkin (DG), spectral, and flux reconstruction (FR), tend to produce oscillations near shocks and discontinuities. These oscillations cause a loss of accuracy and can result in negative pressure and/or density, leading to code break down.

To mitigate these oscillations, nonlinear stabilization techniques, often referred to as shock-capturing methods, have been developed. They can be broadly categorized into one of four groups: limiting (Van Leer 1977, Cockburn et al. 1990, Hoteit et al. 2004, Krivodonova et al. 2004, Krivodonova 2007, Yang and Wang 2009, Park et al. 2010, Park and Kim 2012, 2014, 2016, Moe et al. 2015, You and Kim 2018, Lu et al. 2019), filtering (Hesthaven and Warburton 2008, Asthana et al. 2015), artificial viscosity (Persson and Peraire 2006, Park et al. 2014), or method modification (Dumbser et al. 2015), each with a substantial body of literature (see the references given therein). These approaches differ in accuracy, robustness, computational cost, and difficulty of extensions. More recent works with promising results include (Dzanica et al. 2022, You et al. 2023, Kim et al. 2024).

Among shock capturing methods, we narrow our focus to limiting type, which is arguably the most popular. For standard piecewise linear schemes, the limiting approach imposes constraints on the calculated slopes to prevent excessive steepness, based on Van Leer's (1977) concept of preserving monotonicity: when the data are monotone, the limited slopes yield a monotone solution. The main drawback of limiting is that it causes a loss of accuracy near smooth extrema where the solution is

somewhat flattened and only first-order accurate. Such a reduction in accuracy undermines the inherent high-order nature of the methods employed here.

Concerning higher order (quadratic and higher) methods, the key idea for limiting is the following. First, a solution is obtained using the base scheme. Next, a detection method is employed, marking a cell as "troubled" if, for example, high frequency or high modal components are significant, indicating the presence of oscillations and the necessity for limiting. For such cells, a constraint or a nonlinear reconstruction is applied to reduce steepness and suppress oscillations in the solution. Conversely, if the cell is deemed "good"—that is, not "troubled"—then the base solution is not altered.

Such DG-type limiters rely heavily on the detection technique, and detection can fail: occasionally, a good cell can be marked troubled, and vice versa. If a good cell in smooth regions is marked troubled, limiting may cause a loss of accuracy, and if a troubled cell is marked good, oscillations may take place. The detection methods must also balance between either marking too many cells as troubled resulting in more computing time or marking too few resulting in oscillations. Some of these methods also rely on user specified parameters which must be recalibrated for different data.

In this paper, we introduce a limiting technique designed to mitigate oscillations near shocks while maintaining accuracy for high-order methods such as DG, spectral, and FR. This technique extends the method proposed in (Huynh 1995) and (Suresh and Huynh 1997) for traditional interpolation-type schemes. The key idea is to expand the standard monotonicity limits to provide "room" so that near smooth extrema, limiting has no effect, thereby preserving accuracy. Near a discontinuity, the expanded limits effectively reduce to the original monotonicity limits, suppressing oscillations. Our limiter depends continuously on the data, in contrast to typical limiting methods, which rely heavily on the success of detection to activate or deactivate limiting. The algorithm is further simplified by two observations: (a) near a discontinuity, the magnitudes of the high order modes in the solution become excessively large, but their signs are consistent with the direction of the jump, thus limiting only needs to reduce their magnitudes; and (b) the limiting bounds can be applied to the sum of magnitudes of all modes, linear and higher order. Thus, our limiting shares the reduction of higher-order modes with filtering techniques. The key difference is that our approach can reduce even the linear mode and/or nullify all higher-order modes. These large reductions, however, occur only near a discontinuity, unlike in standard filtering, where a moderate reduction is applied everywhere.

To reduce computing time, we also present efficient criteria for identifying smooth regions where limiting is unnecessary. The main criterion among the three for our detection relies on comparing the jumps at the interfaces to the quadratic content within the cell. The resulting algorithm typically marks one to three cells near a shock as troubled. For a contact discontinuity, after it is dissipated over a few cells, the detection can result in no troubled cells, indicating that the data is sufficiently smooth to be managed by the base method with no limiting. Together with detection, the combined method no longer depends continuously on the data, but it is more economical. A notable characteristic of the entire process is its simplicity in both concept and implementation.

Our approach, along with its motivations and associated discussions, is strongly influenced by the behavior of the DG method at a discontinuity. To illustrate this behavior, we need to formulate the DG method for advection from an interpolation viewpoint using the FR approach. In essence, projection in the DG formulation is equivalent to interpolation at the Radau points. Here, the oscillations resulting from a discontinuity are given by the derivative of the Radau polynomial. While Radau quadratures are well known, Radau polynomials are not widely recognized. These polynomials, which approximate a unit jump at a cell interface, are defined by averaging the Legendre polynomials. Hence, providing a description of DG in the FR framework, along with discussions on Radau polynomials and DG solutions at a discontinuity, becomes necessary and is detailed below.

The paper is essentially self-contained and organized as follows. Section 2 presents the preliminaries including the Radau polynomials and their derivatives. Section 3 describes the DG method in the FR framework for advection. Section 4 shows the behavior of the DG solution for a jump discontinuity. Our limiting and detection methods are presented in Section 5. Numerical examples are provided in Section 6. Finally, conclusions and discussions can be found in Section 7.

2 Preliminaries

For any nonnegative integer m, let $\boldsymbol{P}_m$ be the space of polynomials of degree m or less.

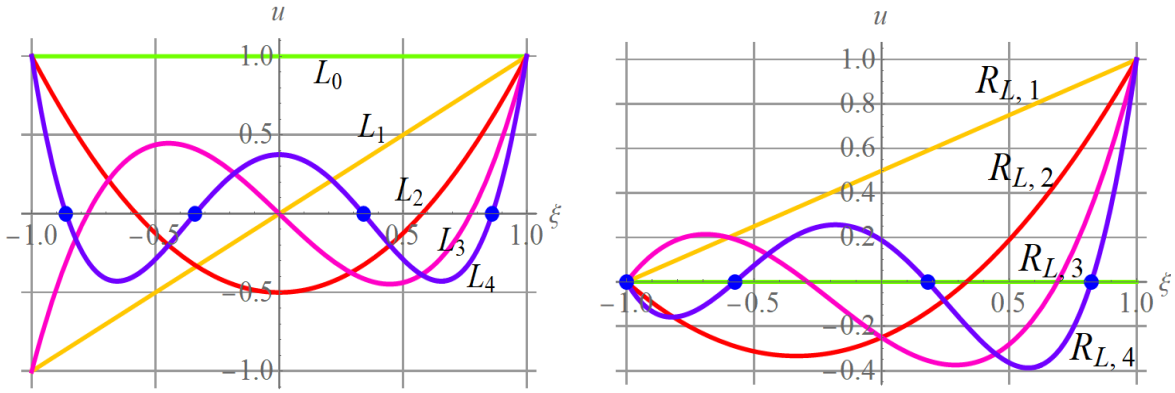
For any two functions v and w on $I = [-1, 1]$, denote $(v, w) = (v, w)_I = \int_{-1}^{1} v(\xi) w(\xi)\, d\xi$.

We need the Legendre polynomials to define the Radau polynomials. Let the Legendre polynomial L_k on I be defined as the unique polynomial of degree k that satisfies $L_k(1) = 1$ and L_k is orthogonal to $\boldsymbol{P}_{k-1}$ (or $L_k \perp \boldsymbol{P}_{k-1}$), i.e., $(L_k, \xi^m) = \int_{-1}^{1} L_k(\xi)\xi^m\, d\xi = 0$, for $m = 0, 1, \ldots, k-1$. The first few Legendre polynomials are

$$L_0 = 1, \quad L_1 = \xi, \quad L_2 = \frac{3\xi^2 - 1}{2}, \quad L_3 = \frac{5\xi^3 - 3\xi}{2}, \text{ and } L_4 = \frac{35\xi^4 - 30\xi^2 + 3}{8}. \tag{2.1}$$

The zeros of L_k are the k Gauss points.

Fig. 2.1(a) shows the graphs of the above Legendre polynomials; the dots, which represent the zeros of L_4, are the 4 Gauss points.



(a) Legendre polynomials (b) Left Radau polynomials

Fig. 2.1: (a) The graphs of the Legendre polynomials L_k, $k = 0, \ldots, 4$; the dots represent the 4 Gauss points. (b) The graphs of the left Radau polynomials $R_{L,k}$, $k = 1, \ldots, 4$; the dots represent the 4 left Radau points. The left Radau polynomials approximate a unit jump (up) at the right boundary.

The Radau quadrature formulas are well-known, but the Radau polynomials are not. For any $k \geq 1$, let the left and right *Radau polynomials* of degree k be defined respectively by

$$R_{L,k} = \frac{1}{2}(L_k + L_{k-1}), \quad \text{and} \quad R_{R,k} = \frac{(-1)^k}{2}(L_k - L_{k-1}). \tag{2.2a,b}$$

Note that $R_{L,k}$ and $R_{R,k}$ are reflections of each other: $R_{R,k}(\xi) = R_{L,k}(-\xi)$. At the two boundaries, for the left Radau polynomial,

$$R_{L,k}(-1) = 0, \quad \text{and} \quad R_{L,k}(1) = 1. \tag{2.3a,b}$$

For the right Radau polynomial,

$$R_{R,k}(-1) = 1, \quad \text{and} \quad R_{R,k}(1) = 0. \tag{2.4a,b}$$

The first few $R_{L,k}$'s are

$$R_{L,1} = (1 + \xi)/2, \quad R_{L,2} = (-1 + 2\xi + 3\xi^2)/4,$$

$$R_{L,3} = (-1 - 3\xi + 3\xi^2 + 5\xi^3)/4, \quad \text{and} \quad R_{L,4} = (3 - 12\xi - 30\xi^2 + 20\xi^3 + 35\xi^4)/16.$$

Fig. 2.1(b) shows the graphs of these left Radau polynomials. The dots corresponding to the zeros of $R_{L,4}$ represent the 4 left Radau points, which include $\xi = -1$.

The corresponding $R_{R,k}$ can easily be obtained by $R_{R,k}(\xi) = R_{L,k}(-\xi)$.

Since both L_k and L_{k-1} are orthogonal to $\boldsymbol{P}_{k-2}$, both Radau polynomials also have this property:

$$R_{L,k} \perp \boldsymbol{P}_{k-2}, \quad \text{and} \quad R_{R,k} \perp \boldsymbol{P}_{k-2}\,. \tag{2.5a,b}$$

Let the step-down function S_D and step-up function S_U on I be defined respectively by

$$S_D(\xi) = \begin{cases} 1 & \text{for} \quad \xi = -1 \\ 0 & \text{for} \ -1 < \xi \le 1 \end{cases} \quad \text{and} \quad S_U(\xi) = \begin{cases} 0 & \text{for} \ -1 \le \xi < 1 \\ 1 & \text{for} \ \ \xi = 1. \end{cases} \tag{2.6a,b}$$

On I, S_D represents a unit jump (down) at $\xi = -1$, and S_U represents a unit jump (up) at $\xi = 1$.

The right Radau polynomial $R_{R,k}$ can be considered as an approximation to the step-down function S_D since $R_{R,k}(-1) = 1$, $R_{R,k}(1) = 0$, and $R_{R,k}$ approximates 0 on $(-1, 1]$ in the sense that $R_{R,k} \perp \boldsymbol{P}_{k-2}$, i.e., $\left(R_{R,k}, \xi^m\right) = 0$ for $m = 0, 1, \dots, k-2$. In other words, except for the two conditions at the boundaries, namely (2.4a,b), $R_{R,k}$ utilizes the rest of the conditions ($k-1$ of them) to approximate 0 in the sense of orthogonality or projection: $R_{R,k} \perp \boldsymbol{P}_{k-2}$.

Similarly, the left Radau polynomial $R_{L,k}$ approximates S_U.

The above discussion implies that the left and right Radau polynomials can be employed to approximate the jumps at the cell interfaces for the DG method as will be discussed below.

The zeros of $R_{L,k}$ are the k left Radau points, and the zeros of $R_{R,k}$ are the k right Radau points.

Next, we present simple formulas for ${R_{L,k+1}}'$ and ${R_{R,k+1}}'$, which approximate the derivative of a jump at the boundaries, and show the behavior of the DG solution at a discontinuity:

$${R_{L,k+1}}' = \frac{1}{2}\sum_{m=0}^{k} (2m+1)L_m = \frac{1}{2}\{L_0 + 3L_1 + 5L_2 + 7L_3 + \cdots + (2k+1)L_k\}, \tag{2.7a}$$

and

$${R_{R,k+1}}' = \frac{1}{2}\sum_{m=0}^{k} (-1)^{m+1}(2m+1)L_m = \frac{1}{2}\{-L_0 + 3L_1 + \cdots + (-1)^{k+1}(2k+1)L_k\}. \tag{2.7b}$$

Note the pattern $1, 3, 5, \dots, 2k+1$. For the proof of (2.7), see Appendix A. The above imply

$${R_{L,k+1}}' = {R_{L,k}}' + \frac{2k+1}{2}L_k, \quad \text{and} \quad {R_{R,k+1}}' = {R_{R,k}}' + (-1)^{k+1}\frac{2k+1}{2}L_k. \tag{2.8a,b}$$

The values at the boundaries, employed later, are also straightforward:

$${R_{L,k+1}}'(1) = -{R_{R,k+1}}'(-1) = \frac{1}{2}\{1 + 3 + 5 - \cdots + (2k+1)\} = \frac{(k+1)^2}{2}, \tag{2.9}$$

and

$${R_{L,k+1}}'(-1) = -{R_{R,k+1}}'(1) = \frac{1}{2}\{1 - 3 + 5 + \cdots + (-1)^k(2k+1)\} = (-1)^k\,\frac{k+1}{2}. \tag{2.10}$$

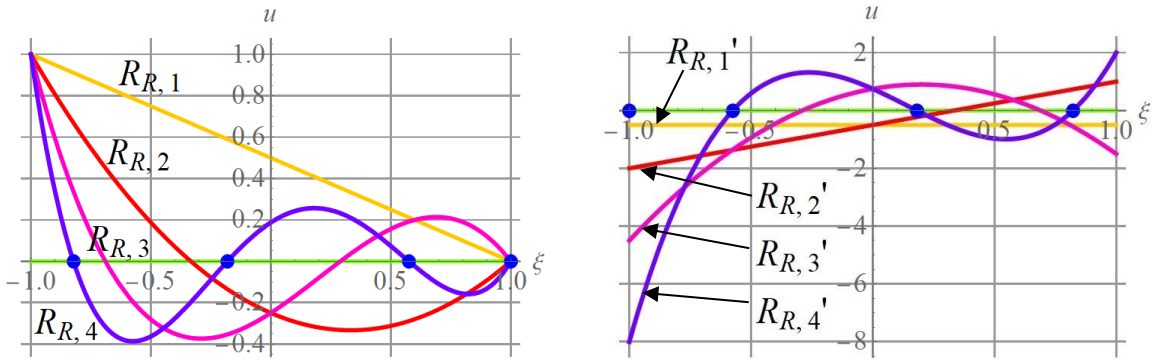
Fig. 2.2 shows the plots of (a) the first few right Radau polynomials and (b) their derivatives. In (b), consistent with (2.9), the values ${R_{R,k+1}}'(-1)$, for $k = 0, \dots, 3$, respectively, are: $-1/2$, $-4/2$, $-9/2$, and $-16/2$; and consistent with (2.10), the values ${R_{R,k+1}}'(1)$ are $-1/2$, 1, $-3/2$, and 2.

Denote the $k+1$ left and right Radau points in increasing order by $\xi_{L,m}$ and $\xi_{R,m}$ respectively, $m = 1, \dots, k+1$. Then, while $R_{R,k+1}$ vanishes at the $k+1$ *right* Radau points, its derivative ${R_{R,k+1}}'$ vanishes at the k interior points of the $k+1$ *left* Radau points: for $l = 2, \dots, k+1$,

$${R_{R,k+1}}'(\xi_{L,l}) = 0. \tag{2.11}$$

The exception is at the left boundary $\xi = -1$ where ${R_{R,k+1}}'(-1) = -(k+1)^2/2$ by (2.9). The proof of (2.11) can be found in Appendix B.

A similar statement holds for the left Radau polynomial with appropriate changes.



(a) Right Radau polynomials (b) Derivatives of Right Radau polynomials

Fig. 2.2: (a) Graphs of the right Radau polynomials $R_{R,k}$, $k = 1, \ldots, 4$, alongside dots corresponding to the zeros of $R_{R,4}$ representing the 4 right Radau points; (b) graphs of $R_{R,k}{}'$, note the different scales. Consistent with (2.11), for each k, the $k-1$ zeros of $R_{R,k}{}'$ together with $\xi = -1$ form the k left Radau points represented by the dots in (b) for $k = 4$.

3. Discontinuous Galerkin method for advection via interpolation

For advection, by casting the DG method in the flux reconstruction (FR) framework, projection, the key idea in the DG formulation, turns out to be equivalent to interpolation at the Radau points.

Consider the advection equation

$$u_t + a u_x = 0 \tag{3.1}$$

with initial condition

$$u(x,0) = u_0(x) \tag{3.2}$$

where t is time, x space, and a the advection speed, a positive constant (we can set $a = 1$). By assuming that u_0 is periodic, boundary conditions are trivial and thus omitted. The exact solution at time t is obtained by shifting the data curve to the right a distance at: $u_{\text{exact}}(x,t) = u_0(x - at)$.

Let the domain of calculation Ω be divided into (possibly non-uniform) cells E_j, $j = 1, 2, \ldots$ Denote the center of E_j by x_j and its width by h_j. Set $h = \max(h_j)$, $j = 1, 2, \ldots$ Using the reference frame, with ξ varying on $I = [-1, 1]$ and x on E_j, the linear function mapping I onto E_j and its inverse are

$$x(\xi) = x_j + \xi \frac{h_j}{2}, \quad \text{and} \quad \xi(x) = \frac{2}{h_j}(x - x_j). \tag{3.3}$$

Thus, $dx = h_j d\xi/2$. A function $r_j(x)$ on E_j results in a function on I denoted by, for simplicity of notation, $r_j(\xi)$ with $r_j(\xi) = r_j\big(x(\xi)\big)$. By the chain rule, $dr_j/dx = (2/h_j)(dr_j/d\xi)$.

Let p be a nonnegative integer. At time t, let the solution $u(x,t)$ be approximated on each cell E_j by a polynomial of degree p in x denoted by $u_j(x,t)$. For each fixed t, the collection of all polynomials $\{u_j(.,t)\}$ as j varies forms a function denoted by u_h, which is generally discontinuous across cell interfaces. In the reference description for E_j, the solution $u_j(\xi,t)$ can be expressed in *modal form* as

$$u_j(\xi,t) = \sum_{k=0}^{p} u_{j,k}(t)\, L_k(\xi). \tag{3.4}$$

For each k, $0 \le k \le p$, the value $u_{j,k}$ is called the k-th mode (or k-th moment) of the solution in cell j, and u_j of degree p is determined by the $p+1$ modes $u_{j,k}$.

Assume that the data $u_{j,k}(t^n) = u_{j,k}$ are known for all cells. We wish to calculate $(u_h)_x$ or equivalently $(u_h)_\xi$. The initial condition $u_0(x)$ can be discretized by using the values at the $p+1$ Gauss points in each cell. The resulting nodal data can easily be transformed into modal form (3.4).

As is routine, to involve data interaction among cells, at each interface $j+1/2$, we define a value common for the two adjacent cells j and $j+1$ by upwinding: using the reference frame,

$$u_{j+1/2}^{\text{upw}} = u_j(1). \tag{3.5}$$

The DG method can be cast as follows (for the proof, see Appendix C). In each cell j, we reconstruct the solution by a polynomial of degree $p+1$, one degree higher than that of u_j, denoted by U_j, and defined by $p+2$ conditions. At the two interfaces, U_j takes on the common (upwind) values,

$$U_j(-1) = u_{j-1/2}^{\text{upw}} = u_{j-1}(1) \quad \text{and} \quad U_j(1) = u_{j+1/2}^{\text{upw}} = u_j(1). \tag{3.6a,b}$$

For the remaining p conditions, we require that $U_j(\xi)$ approximates $u_j(\xi)$ as closely as possible through projection:

$$(U_j - u_j) \perp \boldsymbol{P}_{p-1}. \tag{3.7}$$

The degree $p+1$ for U_j serves the purpose that $(U_j)_\xi$, which yields $(u_h)_\xi$, matches the degree of u_j.

Focussing on cell j, at the left interface, we have the following jump denoted by J_L:

$$U_j(-1) - u_j(-1) = u_{j-1/2}^{\text{upw}} - u_j(-1) = u_{j-1}(1) - u_j(-1) = J_L\,. \tag{3.8}$$

(A more complete notation would be $J_{j,L}$.) On the other hand, at the right interface, $J_R = 0$,

$$U_j(1) - u_j(1) = u_{j+1/2}^{\text{upw}} - u_j(1) = J_R = 0. \tag{3.9}$$

Eqs. (3.7)–(3.9) provide $p+2$ conditions defining $U_j - u_j$. The $p+1$ conditions of (3.7) and (3.9) for $U_j - u_j$ are identical to those for $R_{R,p+1}$, namely, (2.5b) and (2.4b) with $k = p+1$. Thus, by multiplying $R_{R,p+1}$ by the constant J_L, we can also match condition (3.8) at the left boundary, i.e.,

$$U_j - u_j = J_L R_{R,p+1}. \tag{3.10}$$

Equivalently,

$$U_j = u_j + J_L R_{R,p+1}. \tag{3.11}$$

Hence, by adding the "correction" $J_L R_{R,p+1}$ to u_j, we obtain U_j that matches the common values at the interfaces, namely, $U_j(-1) = u_{j-1/2}^{\text{upw}}$ and $U_j(1) = u_{j+1/2}^{\text{upw}}$, and $(U_j)_\xi$ yields the desired $(u_h)_\xi$.

In short, $(U_j)_\xi$ yields the desired $(u_h)_\xi$ by the DG method for the following reasons. The DG solution is of degree p, therefore, U_j needs to be of degree $p+1$. The DG formulation involves integration by parts to enforce the upwind value $u_{j-1/2}^{\text{upw}}$ and $u_{j+1/2}^{\text{upw}}$ at the two interfaces, consistent with U_j matching these values by (3.8) and (3.9). To define U_j, we need p additional conditions. These are provided by—using the key DG idea—requiring that U_j approximates u_j as closely as possible via projection: $(U_j - u_j) \perp \boldsymbol{P}_{p-1}$ as in (3.7).

The polynomial u_j of degree p can be defined by its values at the $p+1$ right Radau points, and U_j can defined by the $p+2$ conditions that it interpolates u_j at these Radau points and $U_j(-1) = u_{j-1/2}^{\text{upw}}$.

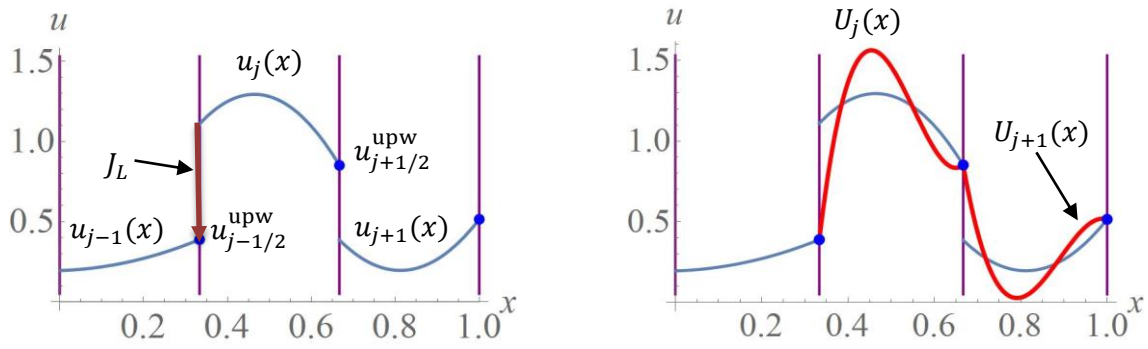
Fig. 3.1 depict (a) piecewise quadratic data, and (b) the cubic polynomials U_j and U_{j+1}, which interpolate u_j and u_{j+1}, respectively, at the 3 right Radau points; $(U_j)_\xi$ yields the DG spatial derivative.

Thus, for advection, loosely put, projection in the DG method is equivalent to interpolation at the **right** Radau points (between U_j and u_j).

On the other hand, the **left** Radau points play an important role in the interpolation between $(U_j)_\xi$ and $(u_j)_\xi$. At the left boundary $\xi = -1$,

$$(U_j)_\xi(-1) = (u_j)_\xi(-1) - J_L \frac{(k+1)^2}{2}.$$

However, at the p interior points of the $p+1$ left Radau points, by (2.11), $(U_j)_\xi$ equals $(u_j)_\xi$. Thus, at the left Radau points, the "correction" to $(u_j)_\xi$, in the form $(U_j)_\xi$, is lumped to the left boundary. This fact also relates to the sign changes of the oscillations across the left Radau points in Fig. 4.1(b) later.



(a) Piecewise quadratic data (b) Cubic polynomials U_j and U_{j+1}

Fig. 3.1: Piecewise quadratic DG method in the FR framework for advection: (a) piecewise quadratic data; (b) cubic polynomial $U_j(x)$, which matches the upwind value at the left interface and interpolates u_j at the 3 right Radau points; $(U_j)_x$ yields $(u_h)_x$ for the DG method.

In passing, if there is also a jump at the right boundary of cell j, (e.g., the speed a changes sign in the cell, or the centered common value is used instead of the upwind value in (3.5)), then to match the common values at both boundaries, (3.11) must be modified to $U_j = u_j + J_L R_{R,p+1} + J_R R_{L,p+1}$.

With U_j by (3.11), the advection equation can then be discretized in semi-discrete form as

$$\frac{\partial}{\partial t} u_j(\xi, t) = -\frac{2}{h_j} a (U_j)_\xi. \tag{3.12}$$

This completes the description of the DG method in the FR framework.

For convenience, the DG method of degree p is abbreviated to DGp. We also denote u_ξ by u'.

The time stepping discussed below corresponds to an Euler forward method of size $\Delta\tau$ (by one stage of the Runge-Kutta method) with a CFL number of $\sigma = a\Delta\tau/h_j$. For convenience, it is referred to as a time step corresponding to σ. The DG solution $\tilde{u}_j$ under such a time step is given by, using the above,

$$\tilde{u}_j = u_j - 2\sigma U_j'. \tag{3.13}$$

(The wave travels a distance 2σ in the reference frame.) Hence, with U_j by (3.11),

$$\tilde{u}_j = u_j - 2\sigma u_j' - 2\sigma J_L R_{R,p+1}{}'. \tag{3.14}$$

Therefore, by (2.7b), the DGp solution is

$$\tilde{u}_j = u_j - 2\sigma u_j' + \sigma J_L \left[L_0 - 3L_1 + 5L_2 + \cdots + (-1)^p (2p+1) L_p\right]. \tag{3.15}$$

In (3.14) and the above, $u_j - 2\sigma u_j'$ represents the advection of u_j with no interaction, while $-2\sigma J_L R_{R,p+1}{}'$ and $\sigma J_L[\ldots]$ represent the effect of the interaction among cells due to the jump J_L.

The next discussion on accuracy is essential for the limiting procedure. At $\xi = 1$, since $L_k(1) = 1$,

$$u_j(1) = \sum_{k=0}^{p} u_{j,k} = u_{j,0} + \cdots + u_{j,p}\,. \tag{3.16}$$

At smooth regions, the k-th mode $u_{j,k}$ is of size $O(h^k)$, decreasing like h^k for small h. Each successive $u_{j,k}$ contributes to an estimate $\sum_{m=0}^{k} u_{j,m}$ for $u(x_{j+1/2})$ accurate to order k, i.e., with an error $O(h^{k+1})$.

In particular, the p-th mode $u_{j,p}$ is $O(h^p)$, and $u_j(1)$ above approximates $u(x_{j+1/2})$ with an error $O(h^{p+1})$. A similar statement holds for $u_j(-1)$. Therefore, at all interfaces in smooth regions,

$$\left|u_{j+1}(-1) - u_j(1)\right| = O(h^{p+1}). \tag{3.17}$$

Thus, if cell j is in a smooth region, both J_L and J_R are $O(h^{p+1})$, and (3.15) results in small corrections for all modes in the solution.

At a discontinuity, however, the jump J_L is $O(1)$, and the term $-2\sigma J_L {R_{R,p+1}}'$ in (3.15) leads to *higher modes of successively larger magnitudes*, resulting in oscillations.

4. Behavior of solutions at a discontinuity

Oscillatory DG solutions for step data.

The following example demonstrates the nature of the oscillations generated by the DG method at a discontinuity. Let the domain $\Omega = [0, 1]$ be divided into 3 uniform cells denoted by $j-1$, j, and $j+1$. With ξ on I, consider the case of a step-down data where $u_{j-1}(\xi) = 1$, and $u_j(\xi) = u_{j+1}(\xi) = 0$ as shown in Fig. 4.1(a). Thus, the data are monotonically decreasing. Under a time step corresponding to σ, since $u_j = 0$, and $J_L = 1$, by (3.14),

$$\tilde{u}_j = -2\sigma {R_{R,p+1}}'. \tag{4.1}$$

Or, by (2.7b),

$$\tilde{u}_j = \sigma\big[L_0 - 3L_1 + 5L_2 + \cdots + (-1)^p(2p+1)L_p\big]. \tag{4.2}$$

Note that the magnitude of the mode increases like $2k+1$ as the mode order k increases.

Since $L_k(-1) = (-1)^k$, the above yields the following solution value at the left interface:

$$\tilde{u}_j(-1) = \sigma\{1 + 3 + 5 + \cdots + (2p+1)\} = \sigma(p+1)^2. \tag{4.3}$$

At the right interface,

$$\tilde{u}_j(1) = \sigma\{1 - 3 + 5 + \cdots + (-1)^p(2p+1)\} = \sigma(-1)^p(p+1). \tag{4.4}$$

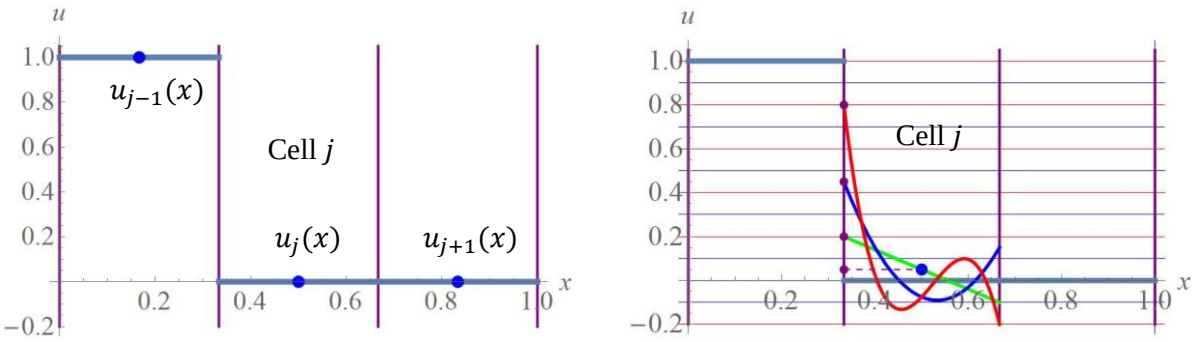
Thus, for the step-down data, as p increases, at the left interface, $\tilde{u}_j(-1)$ increases as $\sigma(p+1)^2$, and at the right interface, $\tilde{u}_j(1)$ has alternating signs and its magnitude increases as $\sigma(p+1)$.

Denote the DG solution of degree p under a time step corresponding to σ by $[\tilde{u}_j]^{(p)}$. It follows from (4.2) for our step-down example that the DGp solutions satisfy the recurrence relation

$$[\tilde{u}_j]^{(p)} = [\tilde{u}_j]^{(p-1)} + \sigma(-1)^p(2p+1)L_p. \tag{4.5}$$

Fig. 4.1 shows (a) the step data and (b) the cubic DG solution (red curve) under a time step with $\sigma = 0.05$. The cubic solution equals $\sum_{m=0}^{3} u_{j,m}L_m$ where, for $m = 0, \ldots, 3$, respectively, the modes are $u_{j,m} = \sigma$, -3σ, 5σ, and -7σ. For $0 \le k \le 2$, each $\sum_{m=0}^{k} u_{j,m}L_m$ is identical to the DGk solution, whose plots are also shown. For $m = 0, \ldots, 3$, the modal values $u_{j,m}\, L_m(-1)$, are σ, 3σ, 5σ, and 7σ respectively; they result in solution values $[\tilde{u}_j]^{(m)}(-1)$ of σ, 4σ, 9σ, and 16σ represented by the dots on the vertical line $x = 1/3$. Thus, the modes $u_{j,m}$'s have the "correct" sign in the sense that they result

in solution values increasing toward 1 at the interface corresponding to the jump (this observation holds for small σ; a large σ can result in values larger than 1). At the right boundary, $[\tilde{u}_j]^{(m)}(1)$ equals σ, -2σ, 3σ, and -4σ. Consistent with (2.11) for ${R_{R,p+1}}'$, the DG solutions also cross the x-axis at the corresponding interior left Radau points.



(a) Step-down data (b) Corresponding cubic DG solution

Fig. 4.1: (a) Step data and (b) cubic DG solution under a time step with $\sigma = 0.05$ (red curve). The cubic solution equals $\sum_{m=0}^{3} u_{j,m} L_m$ where $u_{j,m} = \sigma, -3\sigma, 5\sigma, -7\sigma$ for $m = 0, \ldots, 3$, or $u_{j,m} = (-1)^m(2m+1)\sigma$. Each $\sum_{m=0}^{k} u_{j,m} L_m$, $0 \le k \le 2$, is identical to the DGk solution.

For our step-down example, at the left interface, by (4.3), all solution modes contribute to increasing the value toward 1. That is, they exhibit the "correct" sign (a wrong sign would lead to a decrease in value). Thus, for each mode, we can limit only its magnitude and leave its sign unchanged.

At the right interface, however, by (4.4), $\tilde{u}_j(1)$ has alternating signs with magnitude increasing as $\sigma(p+1)$. Thus, near a discontinuity, to assure that the solution lies within the range [0,1], for modes $u_{j,2}$ and above, zeroing them out is a reasonable option.

The filtering approach suppresses oscillations by reducing the magnitudes of the high modes everywhere. Various functions have been devised for this purpose (Hesthaven and Warburton 2008). However, determining an appropriate reduction for each mode remains challenging. Additionally, a significant drawback is that reducing mode sizes at smooth regions causes a loss of accuracy.

Our approach is different from standard filtering: for the above example, our limiting amounts to *reducing even the linear mode and zeroing out all higher modes*. Since our limiting has effect only at a few cells near discontinuities and has nonexistent or negligible effect at smooth regions, accuracy remains intact while oscillations are suppressed.

Given the oscillatory DG solutions, the question is what should the solution be for the step-down example above? Since the step data exhibits two essential properties—monotonicity and values within the range [0, 1]—it is reasonable to require the solution to also possess these properties.

Monotone solutions for step data.

The following is the FR generalization (Huynh 2007, 2009, Huynh et al. 2014) of the DG method: for advection, set

$$U_j = u_j + J_L g \tag{4.6}$$

where g, called a "correction function", is of degree $p+1$ and determined by $p+2$ conditions. It must approximate the step-down function in some sense. At the two boundaries, g is required to satisfy

$$g(-1) = 1, \; g(1) = 0, \tag{4.7}$$

with p additional conditions to be prescribed. Here, condition $g(-1) = 1$ serves to match the upwind value at the left boundary, while condition $g(1) = 0$ serves to leave the value $u_j(1)$ unchanged.

If $g = R_{R,p+1}$ in (4.6), the result is the DGp method. The $p+1$ zeros of this g are the right Radau points. To obtain a monotone g, we can push all the zeros to $\xi = 1$ by requiring that $\xi = 1$ is a zero of multiplicity $p+1$. Equivalently, for $1 \le k \le p$, all derivatives to degree k vanish: $g^{(k)}(1) = 0$. Thus,

$$g = \left(\frac{1-\xi}{2}\right)^{p+1}. \tag{4.8}$$

Consequently,

$$g' = -\frac{1}{2}(p+1)\left(\frac{1-\xi}{2}\right)^{p}. \tag{4.9}$$

With U_j by (4.6) and g by (4.8), the FR solution after a time step corresponding to σ is given by

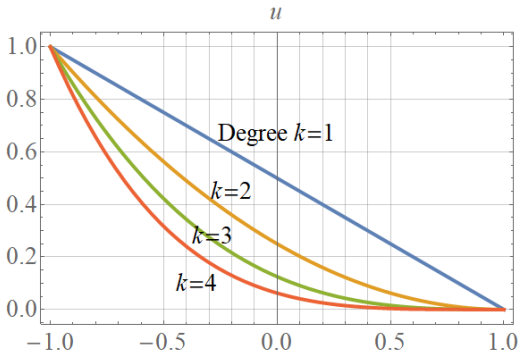
$$\tilde{u}_j = u_j - 2\sigma u_j' + \sigma J_L\left[(p+1)\left(\frac{1-\xi}{2}\right)^{p}\right]. \tag{4.10}$$

Hence, for our example of step-down data,

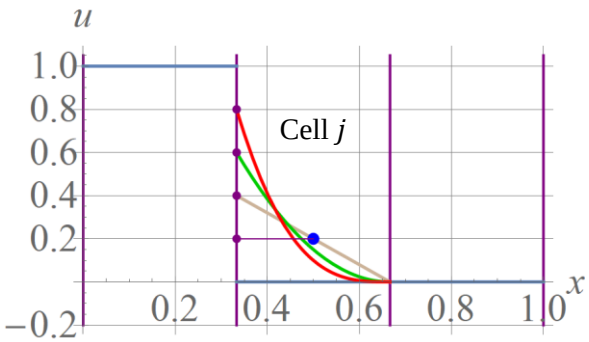
$$\tilde{u}_j = \sigma(p+1)\left(\frac{1-\xi}{2}\right)^{p}. \tag{4.11}$$

For $p = 0$, $\tilde{u}_j = \sigma$, a constant function. For $p = 0, 1, 2, \ldots$, at the left boundary, the solution values are $\sigma, 2\sigma, 3\sigma, \ldots$ as opposed to $\sigma, 4\sigma, 9\sigma, \ldots$ of the DG methods, i.e., $\sigma(p+1)$ as opposed to $\sigma(p+1)^2$. At the right boundary, for all $p \ge 1$, $\tilde{u}_j(1) = 0$ and, for $1 \le k \le p-1$, $\tilde{u}_j^{(k)}(1) = 0$.

Fig. 4.2 shows (a) the graphs of monotone correction functions (4.8) of degree $k = p+1$, where $p = 0, \ldots, 3$, and (b) the corresponding FR solutions (4.11) for the step-down data, which are monotonically decreasing in the cell.



(a) Monotone correction functions



(b) Corresponding monotone FR solutions

Fig. 4.2: (a) Graphs of monotone correction functions (4.8) of degree $k = p+1$ where $p = 0, \ldots, 3$, and (b) the corresponding monotone FR solutions (curves) of degree up to 3 with time step $\sigma = 0.2$.

Regarding the monotone correction function g of (4.8), it can be shown via Fourier analysis that with $p = 1$, the resulting linear FR scheme is stable. However, for $p > 1$, the schemes are unstable.

It appears reasonable to set the goal that for the step-down example above, limiting converts the oscillatory DGp solution in Fig. 4.1(b) into the monotone solution (4.11) shown in Fig. 4.2(b). Such methods remain to be explored.

Here, we focus on limiting methods associated with the linear case $p = 1$ in (4.11) and apply them to higher-order modes. The main reason is that even with linear functions, a discontinuity can still be sharply resolved (in 2 to 4 cells). Additionally, our focus is on simple methods for the moment.

5. Limiting

As is well-known and discussed above, second and higher-order methods generate oscillations near discontinuities. One way to mitigate these oscillations is to limit the data so that it is not too steep. However, limiting can lead to a loss of accuracy near smooth extrema. Below, we present a limiting method capable of suppressing oscillations near discontinuities while preserving accuracy near extrema.

Our limiting procedure can be viewed as an extension of the techniques introduced in (Huynh 1995) for piecewise linear and (Suresh and Huynh 1997) for higher order methods. It turns out that with DG-type methods of degree $p \geq 1$, we can differentiate a discontinuity from a smooth extremum by a three-cell stencil, as opposed to the five-cell stencil required in finite-difference or finite-volume schemes. Hence, the limiting procedure here utilizes a three-cell stencil, like typical limiting processes.

We employ the standard 3-stage Runge-Kutta (RK) time-stepping method. The solution after each stage is given by (3.13): $\tilde{u}_j = u_j - 2\sigma U_j'$. Using the ODE notation k_i for stage i, $\tilde{u}_j = u_j + \sigma(k_i)_j$. In the presence of discontinuities, the solution $\tilde{u}_j$ might be oscillatory. To suppress oscillations, we apply the limiting process below to $\tilde{u}_j$ resulting in the limited solution $\tilde{u}_j^{\mathrm{LTD}}$. Then, since k_i will be used in later stages, it must be updated by $(k_i)_j = (\tilde{u}_j^{\mathrm{LTD}} - u_j)/\sigma$.

As is routine, we limit the k-th mode, $1 \leq k \leq p$, but not the zero-th mode. The cell average quantities $u_{j,0}$'s are fundamental for the scheme to be conservative and should not be altered.

Limiting methods in the literature often constrain $u_{j,k}$ for $k = p, \ldots, 1$ in that order, i.e., the highest mode is constrained first. Here, we employ a reverse order, namely, $k = 1, \ldots, p$, i.e., we limit the linear mode $u_{j,1}$ first. The reason is that in smooth regions, limiting has nonexistent or minimal effect, so the mode order is irrelevant. Near a discontinuity, as shown in Fig. 4.1(b), high modes amplify with increasing mode order. If limiting already effects the linear mode, then the large higher-order modes are all suspects, and even zeroing them out seems reasonable.

To prepare, for any real numbers x and y, let $\mathrm{minmod}(x, y)$ be the median of x, y, and 0, or

$$\mathrm{minmod}(x, y) = \frac{1}{2}[\mathrm{sign}(x) + \mathrm{sign}(y)]\,\min(|x|, |y|). \tag{5.1}$$

That is, $\mathrm{minmod}(x, y)$ equals 0 if x and y are of opposite sign and equals the value with smaller modulus if x and y are of the same sign. Denote the smallest interval containing numbers $x_1, \ldots, x_m$ by

$$I[x_1, \ldots, x_m] = [\min(x_1, \ldots, x_m), \max(x_1, \ldots, x_m)].$$

For any numbers α, β, and y, the constraint that y lies in the interval $I[\alpha, \beta]$ can be enforced by

$$\mathrm{median}(y, \alpha, \beta).$$

We can express the above in minmod form, e.g., using α as pivot,

$$\mathrm{median}(y, \alpha, \beta) = \alpha + \mathrm{median}(y - \alpha, 0, \beta - \alpha) = \alpha + \mathrm{minmod}(y - \alpha, \beta - \alpha). \tag{5.2}$$

Monotonicity bounds

Consider first the linear case. Focusing on cell j, using the average value $u_{j,0}$ as a pivot, to preserve monotonicity, at interface $x_{j+1/2}$, we require $u_{j,1}$ to lie between 0 and $u_{j+1,0} - u_{j,0}$, or $u_{j,1} \in I[0, u_{j+1,0} - u_{j,0}]$. Similarly, at interface $x_{j-1/2}$, we require $-u_{j,1}$ to lie between 0 and $u_{j-1,0} - u_{j,0}$, or equivalently, $u_{j,1} \in I[0, u_{j,0} - u_{j-1,0}]$; see Fig. 5.1. To combine these conditions, set

$$u^{\mathrm{mm}} = \mathrm{minmod}\big[u_{j+1,0} - u_{j,0}, u_{j,0} - u_{j-1,0}\big]. \tag{5.3a}$$

The constraint that $u_{j,1}$ must be between 0 and u^{mm} yields MUSCL (Monotone Upstream-Centered Schemes for Conservation Laws) limiting (Van Leer 1977). It is enforced by

$$u_{j,1} \leftarrow \mathrm{minmod}\big[u_{j,1}, u^{\mathrm{mm}}\big]. \tag{5.3b}$$

In smooth regions of the solution with a nonzero slope, the above constraint provides ample "room" so that the limited $u_{j,1}$ resulting from (5.3b) remains identical to the original $u_{j,1}$, meaning limiting has no effect; see Fig. 5.1(b). However, a significant drawback arises near extrema, where u^{mm} can become 0 or very small, leading to $u_{j,1} = 0$ or very small after limiting, thus reducing accuracy to first order. We will discuss methods to expand the range $I[0, u_{j+1,0} - u_{j,0}]$ to alleviate this issue later.

Eqs. (5.3a,b) represent Van Leer's MUSCL constraint (1977): "… in order to preserve the monotonicity of a sequence of mesh averages, *the linear function* (12) *must not take values outside the range spanned by the neighboring mesh averages*." Here, "*the linear function* (12)" corresponds to a linear solution in cell j. An error in the DG literature misidentifies (5.3a,b) as the total variation diminishing (TVD) constraint and erroneously assigns essentially the minmod slope (5.3a) as the final MUSCL slope or final $2u_{j,1}$, resulting in an overly dissipative MUSCL outcome (e.g., p. 150, Hesthaven and Warburton 2008).

As depicted in Fig. 4.1(b), the modes exhibit the "correct" signs. Thus, we can simplify the above constraint by limiting only the magnitudes, which also facilitates the limiting of higher-order modes. Note that reducing the modal magnitude is also employed in the filtering approach; however, the extent of reduction and the underlying reasons differ completely between limiting and filtering.

For simplicity of notation, we sometimes drop the subscript j, e.g., we use the notation u_L^{Lim} where the superscript "Lim" stands for "Limit", subscript "L" for "left", and subscript j is implied. All quantities defined in (5.4)–(5.7) are positive (≥ 0). Set

$$u_L^{\mathrm{Lim}} = \left|u_{j-1,0} - u_{j,0}\right|, \quad u_R^{\mathrm{Lim}} = \left|u_{j+1,0} - u_{j,0}\right|. \tag{5.4}$$

and

$$u^{\mathrm{Lim}} = \min\left(u_L^{\mathrm{Lim}}, u_R^{\mathrm{Lim}}\right). \tag{5.5}$$

Denote u^{Lim} by u_1^{Lim} where the subscript 1 represents "linear" (u_k^{Lim} will be defined later). We require

$$\left|u_{j,1}\right| \leq u_1^{\mathrm{Lim}}. \tag{5.6}$$

The above constraint is enforced by replacing $u_{j,1}$ by the limited quantity $u_{j,1}^{\mathrm{Ltd}}$,

$$u_{j,1}^{\mathrm{Ltd}} = \mathrm{sign}\left(u_{j,1}\right) \min\left(\left|u_{j,1}\right|, u_1^{\mathrm{Lim}}\right). \tag{5.7}$$

Fig. 5.1(a) shows u_L^{Lim}, u_R^{Lim}, $u^{\mathrm{Lim}} = u_1^{\mathrm{Lim}} = u_R^{\mathrm{Lim}}$, the original linear data (thin blue line), and the limited slope corresponding to $u_{j,1}^{\mathrm{Ltd}}$ (thick purple line). Here, for ease of observation, $\sigma = 0.2$. Fig. 5.1(b) shows the ample "room" provided by u^{Lim} at smooth regions with a nonzero slope.

Constraints like (5.6) and (5.7) are typically employed to limit slope quantities. Higher-order modes are then limited by employing neighboring higher-order modes in a similar manner.

Our approach is different: we use $u^{\mathrm{Lim}} = u_1^{\mathrm{Lim}}$ to constrain all modes by requiring that

$$\sum_{k=1}^{p} \left|u_{j,k}\right| \leq u^{\mathrm{Lim}}. \tag{5.8}$$

With $u_1^{\mathrm{Lim}} = u^{\mathrm{Lim}}$ by (5.5) and $u_{j,1}^{\mathrm{Ltd}}$ by (5.7), we define $u_2^{\mathrm{Lim}} = u_1^{\mathrm{Lim}} - \left|u_{j,1}^{\mathrm{Ltd}}\right|$, which is, loosely put, the remainder of the limit. We then require $\left|u_{j,2}\right| \leq u_2^{\mathrm{Lim}}$, and the process continues. In other words, to limit the k-th mode for $2 \leq k \leq p$, with the limited quantity $u_{j,k-1}^{\mathrm{Ltd}}$, let the remainder of the limit be

$$u_k^{\mathrm{Lim}} = u_{k-1}^{\mathrm{Lim}} - \left|u_{j,k-1}^{\mathrm{Ltd}}\right|. \tag{5.9}$$

We require

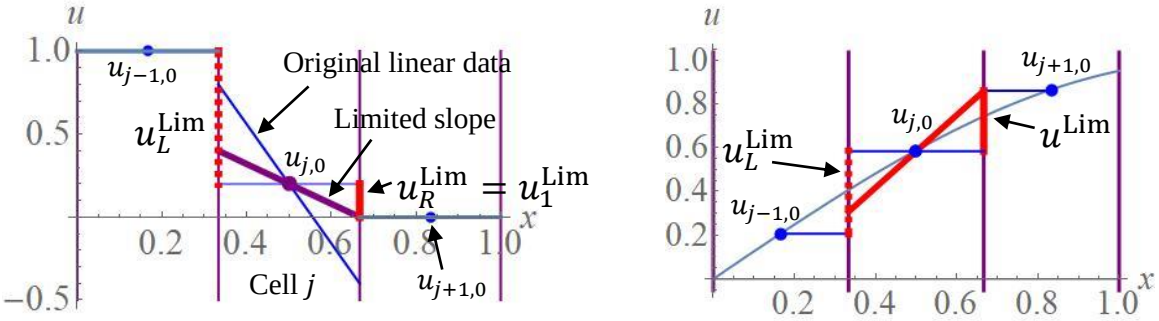
$$\left|u_{j,k}\right| \leq u_k^{\mathrm{Lim}}. \tag{5.10}$$

The above constraint is satisfied by replacing $u_{j,k}$ by the limited quantity $u_{j,k}^{\text{Ltd}}$:

$$u_{j,k}^{\text{Ltd}} = \text{sign}(u_{j,k}) \min(|u_{j,k}|, u_k^{\text{Lim}}). \tag{5.11}$$

This completes the limiting procedure except that we still need to enlarge u^{Lim} to provide "room" near extrema as will be discussed below.

Note that for any $k \geq 1$, if $|u_{j,k}| \geq u_k^{\text{Lim}}$, then $|u_{j,k}^{\text{Ltd}}| = u_k^{\text{Lim}}$, and the remaining limit $u_{k+1}^{\text{Lim}} = u_k^{\text{Lim}} - |u_{j,k}^{\text{Ltd}}|$ is zero, resulting in the vanishing of all $k+1$ and higher modes after limiting.



(a) Limiting alters $u_{j,1}$ (b) Limiting provides ample "room"

Fig. 5.1: (a) Quantities u_L^{Lim} (dashed red vertical line), u_R^{Lim} (thick red vertical line), $u_1^{\text{Lim}} = u_R^{\text{Lim}}$, the original linear data (thin blue line), and the limited linear data (thick purple line); (b) at a smooth region with a nonzero slope, u^{Lim} (red thick vertical line) provides ample "room" so that limiting has no effect.

For the step-down example in Fig. 4.1(b), the limiting process reduces the cubic solution to a limited linear solution: all modes of order $k \geq 2$ vanish after limiting. At first glance, the lack of higher-order modes might suggest that condition (5.8) dissipates a discontinuity. However, as illustrated later in Fig. 6.2, this limit yields a relatively sharp transition at a discontinuity (spanning from 2 to 4 cells).

At smooth regions with a nonzero slope (away from extrema), for a uniform mesh, up to an error of $O(h^2)$, we have $(u_{j,0} - u_{j-1,0}) \approx (u_{j+1,0} - u_{j,0}) \approx 2u_{j,1}$. Therefore, the limit $u_1^{\text{Lim}} \approx 2u_{j,1}$ provides ample "room" so that limiting has no effect on all $u_{j,k}$, $k \geq 1$. This fact is depicted in Fig. 5.1(b).

At a smooth extremum, however, u^{Lim} can become 0, causing a loss of accuracy as shown in Fig. 5.2(a).

Limiting near smooth extrema

To address this loss of accuracy, we enlarge u^{Lim} in a manner that allows more "room" near a smooth extremum so that limiting has no effect, and thus, accuracy is preserved. At a discontinuity, the enlarged limit essentially reduces to the original limit u^{Lim} in (5.5). In fact, for the case of the step-down data in Fig. 4.1(b) above, the enlarged limit is identical to the original limit.

We deal with real numbers below. For cell j, consider the quadratic defined by the first three modes:

$$q_j(\xi) = u_{j,0} + u_{j,1}L_1(\xi) + u_{j,2}L_2(\xi) = u_{j,0} + u_{j,1}\xi + u_{j,2}(3\xi^2 - 1)/2,$$

where the expressions for L_1 and L_2 are given in (2.1). Therefore,

$$q_j(-1) = u_{j,0} - u_{j,1} + u_{j,2}, \quad q_j(0) = u_{j,0} - u_{j,2}/2, \quad \text{and} \quad q_j(1) = u_{j,0} + u_{j,1} + u_{j,2}. \tag{5.12}$$

We can now enlarge u_R^{Lim} for cell j. At interface $x_{j+1/2}$, using the neighboring cell averages, set

$$v_{11} = \tfrac{1}{2}(u_{j,0} + u_{j+1,0}). \tag{5.13}$$

For cell j, consider the quadratic q_j in the local frame $I = [-1, 1]$. The line joining $(-1, q_j(-1))$ and $(0, q_j(0))$ intersects the right boundary $\xi = 1$ at $v_{12} = 2q_j(0) - q_j(-1)$. By (5.12),

$$v_{12} = u_{j,0} + u_{j,1} - 2u_{j,2}. \tag{5.14}$$

Note that whereas $q_j(1)$ in (5.12) has the term $u_{j,2}$, the above contains the term $-2u_{j,2}$, which serves to provide the extra room so that limiting has minimal effect in smooth regions. By reflection, for cell $j + 1$, the line joining $(1, q_{j+1}(1))$ and $(0, q_{j+1}(0)$ intersects the line $\xi = -1$, i.e., the left cell boundary, at $v_{13} = 2q_{j+1}(0) - q_{j+1}(1)$. By using (5.12) with appropriate change in indices,

$$v_{13} = u_{j+1,0} - u_{j+1,1} - 2u_{j+1,2}. \tag{5.15}$$

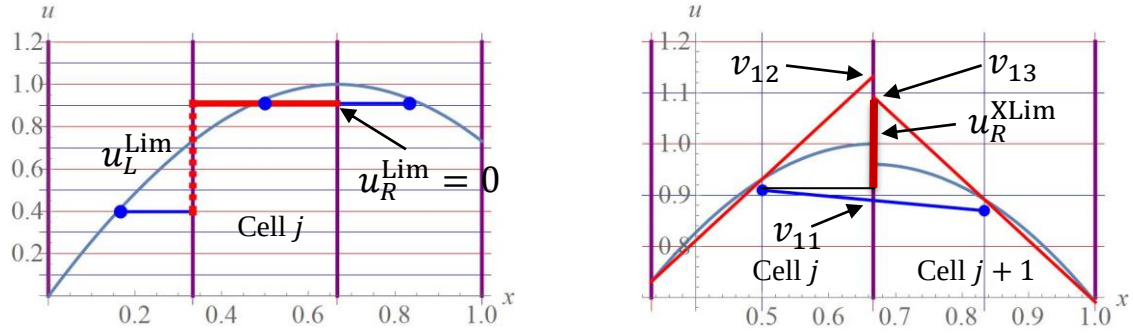
At interface $j + 1/2$, denote by v_{14} the median of v_{11}, v_{12}, and v_{13}:

$$v_{14} = v_{11} + \text{minmod}(v_{12} - v_{11}, v_{13} - v_{11}). \tag{5.16}$$

To the right of cell j, we enlarge the range $I[u_{j,0}, u_{j+1,0}]$ to $I[u_{j,0}, u_{j+1,0}, v_{14}]$. To simplify the process, we replace u_R^{Lim} by the extended limit u_R^{XLim} where, using $u_{j,0}$ as the pivot for cell j,

$$u_R^{\text{XLim}} = \text{Max}(|u_{j+1,0} - u_{j,0}|, |v_{14} - u_{j,0}|). \tag{5.17}$$

Note that near a smooth extremum with a nonzero curvature, i.e., $u_{j,2} \neq 0$, the quantities v_{12} and v_{13} are close to each other and the median v_{14} used in u_R^{XLim} provides room so that limiting has no effect as can be seen in Fig. 5.2(b). The use of minmod function for v_{14} also assures that for the case of a step-down data in Fig. 4.1(b), the extended limit u_R^{XLim} reduces to the original limit u_R^{Lim}.



(a) Monotonicity limit u_R^{Lim} near a maximum (b) Extended limit u_R^{XLim} near a maximum

Fig. 5.2: (a) u_R^{Lim} reduces to 0, and limiting causes a loss of accuracy; (b) near a smooth maximum with a nonzero curvature, the extended limit u_R^{XLim} provides ample room so limiting has no effect.

The extension for the left limit is similar. At the interface $x_{j-1/2}$, set

$$v_{21} = \frac{1}{2}\left(u_{j,0} + u_{j-1,0}\right). \tag{5.18}$$

The line joining $(1, q_j(1))$ and $(0, q_j(0))$ intersects the left boundary at $2q_j(0) - q_j(1) = v_{22}$, where

$$v_{22} = u_{j,0} - u_{j,1} - 2u_{j,2}. \tag{5.19}$$

For cell $j - 1$, the line joining $(-1, q_{j-1}(-1))$ and $(0, q_{j-1}(0))$ intersects the right boundary $x_{j-1/2}$ at $2q_{j-1}(0) - q_{j-1}(-1) = v_{23}$, where

$$v_{23} = u_{j-1,0} + u_{j-1,1} - 2u_{j-1,2}. \tag{5.20}$$

Set

$$v_{24} = v_{21} + \text{minmod}(v_{22} - v_{21}, v_{23} - v_{21}). \tag{5.21}$$

Thus, the extended limit u_L^{XLim} is given by

$$u_L^{\text{XLim}} = \text{Max}(|u_{j-1,0} - u_{j,0}|, |v_{24} - u_{j,0}|). \tag{5.22}$$

Combining both sides, like (5.5), the extended limit is

$$u^{\text{XLim}} = \min\left(u_L^{\text{XLim}}, u_R^{\text{XLim}}\right). \tag{5.23}$$

The rest of the algorithm for all modes is identical to (5.5)–(5.11) with u^{Lim} replaced by u^{XLim}. This completes the limiting process.

We summarize the limiting algorithm below.

Let the solution after one stage of Runge-Kutta (i.e., after an Euler forward step) be denoted by $\{u_{j,k}\}$, which might be oscillatory. For each index (or at cell) j, set $v_{11}, \ldots, v_{14}$, and u_R^{XLim} by (5.13)–(5.17), and set $v_{21}, \ldots, v_{24}$, and u_L^{XLim} by (5.18)–(5.22). Set u^{XLim} by (5.23). Then with $u_1^{\text{Lim}} = u^{\text{XLim}}$, for k from 1 to p, set

$$u_{j,k}^{\text{Ltd}} = \text{sign}\left(u_{j,k}\right) \min\left(\left|u_{j,k}\right|, u_k^{\text{Lim}}\right). \tag{5.24}$$

and

$$u_{k+1}^{\text{Lim}} = u_k^{\text{Lim}} - \left|u_{j,k}^{\text{Ltd}}\right|. \tag{5.25}$$

Finally, for the standard 3-stage RK method, with the ODE notation k_i for stage i, and with $u_{j,k}$ replaced by $u_{j,k}^{\text{Ltd}}$, we update $(k_i)_j$ accordingly. This completes the algorithm.

The above limiting algorithm contains no "if" or "max" statements; furthermore, "sign" is exclusively combined with "min" statements. Clearly, its outcome depends continuously on the data.

For coding purpose, the algorithm can be shortened, e.g., (5.13)–(5.16) can be combined into the quantity $v_{14} - u_{j,0}$ in (5.17), resulting in

$$v_{14} - u_{j,0} = 0.5\left(u_{j+1,0} - u_{j,0}\right) + v_{15}$$

where

$$v_{15} = \text{minmod}\left(u_{j,1} - 2u_{j,2} + 0.5u_{j,0} - 0.5u_{j+1,0}\,,\ -u_{j+1,1} - 2u_{j+1,2} + 0.5u_{j+1,0} - 0.5u_{j,0}\right).$$

Note that from a filtering point of view, our limiting technique corresponds to occasionally filtering even the first mode $u_{j,1}$ and nullifying all higher modes (near a discontinuity). The technique has the advantage of providing ample room so that limiting has nonexistent or minimal effect at smooth regions, and it only takes effect at a few cells near a discontinuity.

Detecting "good" cells at smooth regions where limiting has no effect

While the above algorithm is relatively straightforward, for fluid flow equations, the costs add up. Therefore, as is commonly practiced in the literature albeit for different reasons, it is prudent to detect "good" cells at smooth regions where limiting does not alter the data and therefore is not needed. We present here a detection method with three criteria. For each cell, if any one of the criteria is met, the cell is flagged as good, and no limiting is required. The criteria are effective, as only a few cells near a discontinuity are flagged "troubled", indicating the need for limiting. However, even at those cells, limiting may still leave the data unaltered. Typically, each discontinuity has four or fewer troubled cells.

The first criterion relates to the jumps or gaps at the two interfaces of the cell. At smooth regions, by (3.17), this gap is a small quantity of order $O(h^{p+1})$. Denote the quadratic content for cell j by

$$c_j^Q = \left|u_{j,1}\right| + \left|u_{j,2}\right|. \tag{5.26}$$

We do not include higher modes because at a discontinuity, these modes can be very large and may result in false detection in (5.27) below. The cell j is considered to be good if the gaps at the two interfaces are both smaller than $c_j^Q/5$:

$$\text{Max}\big[\,|u_{j-1}(1)-u_j(-1)|, |u_{j+1}(-1)-u_j(1)|\,\big] \le \frac{1}{5}c_j^Q + 10^{-3}. \tag{5.27}$$

Here, we allow a noise of level 10^{-3}. The factor $1/5$ works well in the numerical examples below. For fluid flow equations, we apply detection only to density and not to momentum or energy. Therefore, when prudence is required, this factor can be reduced, for example, to $1/10$.

Note that if the gaps are small, then the correction to the solution, namely $-2\sigma J_L R_{R,p+1}{}'$ in (3.14) or (3.15), is relatively small compared to the quadratic content.

The second criterion relates to the second modes, or equivalently, the curvature terms (Huynh 1995), at the cell and its two neighbors being close to each other. Cell j is considered to be good if

$$\frac{4}{5} \le \frac{u_{j-1,2}}{u_{j,2}} \le \frac{5}{4} \quad \text{and} \quad \frac{4}{5} \le \frac{u_{j+1,2}}{u_{j,2}} \le \frac{5}{4}. \tag{5.28}$$

For coding, denote $c = 4/5$. (Here, $c < 1$, and the closer c is to 1, the more stringent the criterion.) The left expression above can be stated as $u_{j-1,2}$ lies between $cu_{j,2}$ and $c^{-1}u_{j,2}$, which can be written as

$$\big(u_{j-1,2} - cu_{j,2}\big)\big(u_{j-1,2} - c^{-1}u_{j,2}\big) \le 0. \tag{5.29}$$

Thus, (5.28) can be expressed as, with a noise level of 10^{-5} (due to the product terms),

$$\text{Max}\big[\big(u_{j-1,2} - cu_{j,2}\big)\big(u_{j-1,2} - c^{-1}u_{j,2}\big), \big(u_{j+1,2} - cu_{j,2}\big)\big(u_{j+1,2} - c^{-1}u_{j,2}\big)\ \big] \le 10^{-5}. \tag{5.30}$$

Finally, the third criterion relates to the slopes. Cell j is considered good if $|u_{j,1}|$ is well within u^{Lim}:

$$|u_{j,1}| \le \frac{3}{4}\,u^{\text{Lim}} + 10^{-4}. \tag{5.31}$$

If any of the criteria (5.27), (5.30), or (5.31) is met, the cell is flagged as good; otherwise, the cell is troubled, and limiting is necessary. This completes the detection algorithm.

Typically, at each discontinuity, the detection method above yields four or fewer troubled cells, and sometimes even zero troubled cells after the discontinuity has dissipated.

Concerning noise levels, a tighter (smaller) level results in more troubled cells.

6. Numerical Results

The following numerical tests demonstrate the effectiveness of our limiting technique.

In all figures, the dots represent the cell average solutions, the blue curves the polynomial solutions, and the red curves the initial or exact solution. The standard 3-stage RK method is employed for time stepping; for solutions of degree $p = 2, 3, 4$, the corresponding CFL limits are approximately 0.209, 0.130, and 0.089, respectively. The CFL number σ is generally chosen close to the limit.

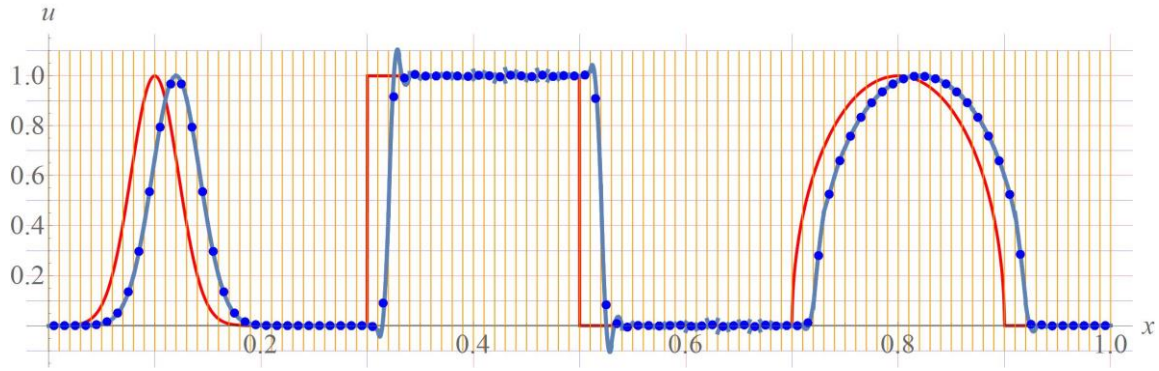
Advection

The first test involves advection on the domain $[0, 1]$ with 100 cells and periodic boundary conditions. The initial condition $u_0(x)$ consists of three waves, each of half width 0.1, including a Gaussian, a rectangular, and a semi-ellipse wave:

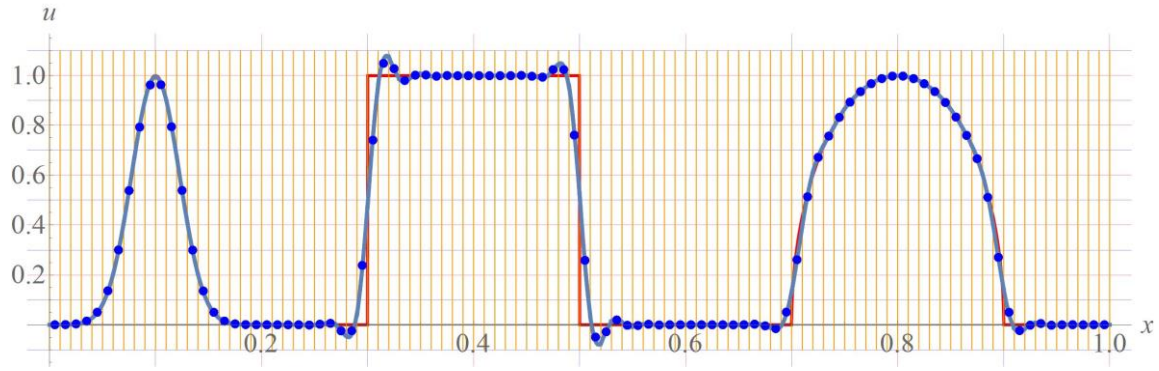
$$u_0(x) = \begin{cases} e^{-10\left(\frac{x-0.1}{0.1}\right)^2}, & 0 \le x \le 0.2 \\ 1, & 0.3 \le x \le 0.5 \\ 1 - \left(\dfrac{x-0.8}{0.1}\right)^2, & 0.7 \le x \le 0.9 \\ 0, & \text{otherwise.} \end{cases}$$

Fig. 6.1 shows the DG2 solutions with no limiting and $\sigma = 0.2$. In Fig 6.1(a), the solution after 5 time steps (the wave travels a distance 1 cell width) is noisy at the rectangular wave, and a little noise appears for the semi-ellipse wave. In Fig 6.1(b), after 10 time steps (2 cell widths), some of the noise is dissipated due to the numerical dissipation of the DG method. In Fig 6.1(c), after 500 time steps (1 period), the solution reaches a steady level of oscillations.

(a) 5 time steps (distance travelled: 1 cell width)



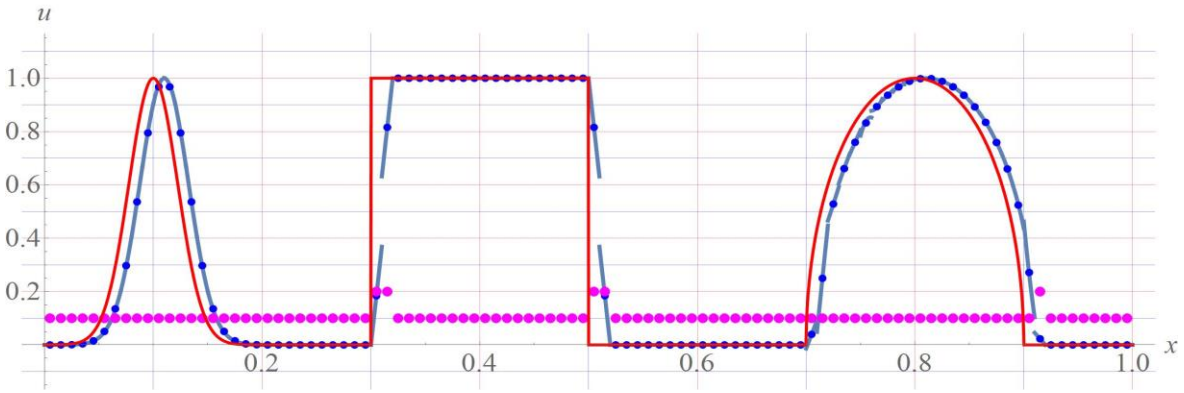
(b) 10 time steps (distance travelled: 2 cell widths)



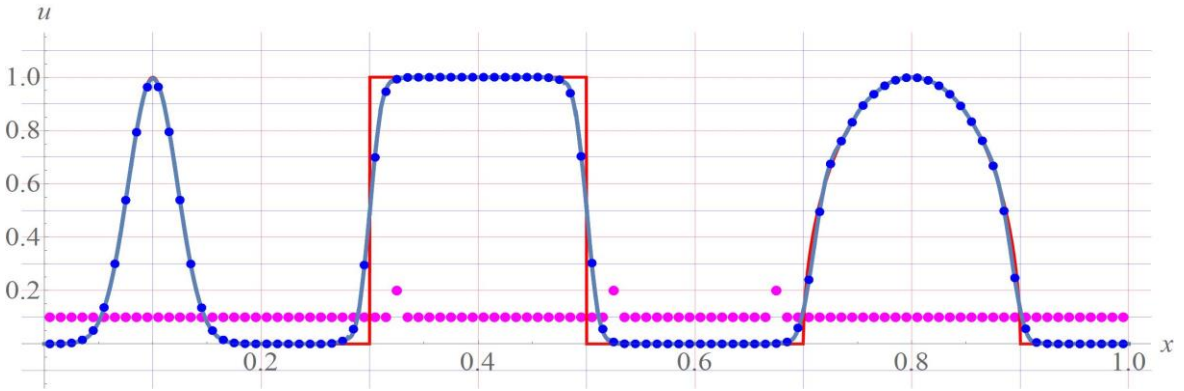
(c) 500 time steps (distance travelled: 1 period)

Fig. 6.1. DG2 solutions with no limiting and CFL number $\sigma = 0.2$. The blue dots represent the cell average solutions, the blue curves the polynomial solutions, and the red curves the exact solution.

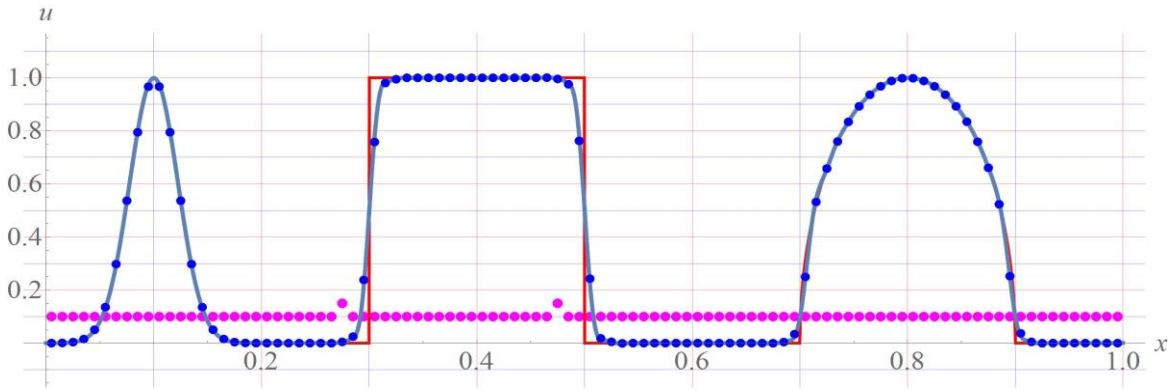
Fig. 6.2 shows the DGp solutions with limiting for $p = 2, 3, 4$. Fig. 6.2(a) depicts the DG2 solution for $\sigma = 0.2$ after 5 time steps (1 cell width); similar to the unlimited case in Fig. 6.1(a), a little noise appears for the semi-ellipse wave. Fig. 6.2(b) shows the DG2 solution with $\sigma = 0.2$ after 500 time steps (1 period). Fig. 6.2(c) represents the DG3 solution for $\sigma = 0.125$ after 800 time steps (1 period); for this case, each jump is resolved by only 2 cells. Note the improvement over the DG2 solution in Fig. 6.2(b). Fig. 6.2(d) depicts the DG4 solution with $\sigma = 1/12$ after 1200 time steps (1 period); here, the solution is very similar to the DG3 solution in Fig. 6.2(c). Note the Gaussian, including its maximum, is well resolved in all cases demonstrating that our limiting preserves accuracy.



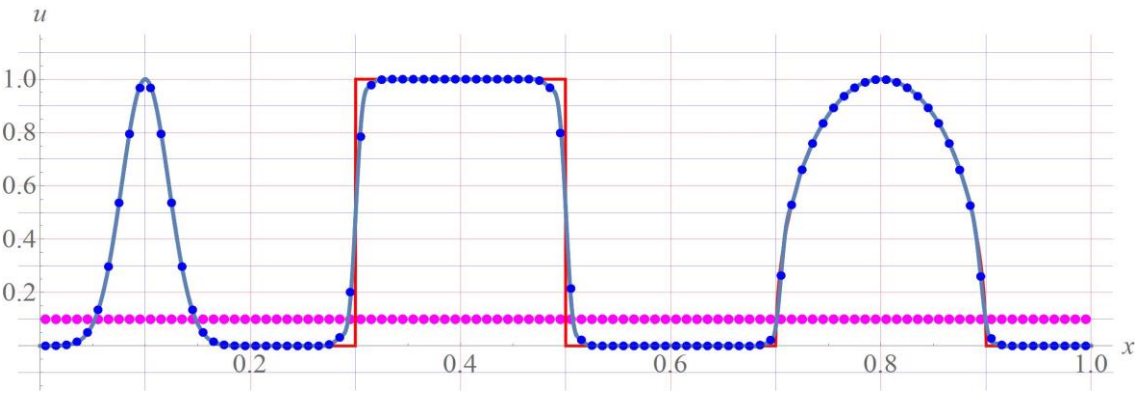
(a) DG2 solution with limiting and $\sigma = 0.2$ after 5 time steps (distance travelled: 1 cell width)



(b) DG2 solution with limiting and $\sigma = 0.2$ after 500 time steps (distance travelled: 1 period)



(c) DG3 solution with limiting and $\sigma = 0.125$ after 800 time steps (distance travelled: 1 period)



(d) DG4 with limiting and $\sigma = \frac{1}{12} \approx 0.08333$ after 1200 time steps (distance travelled: 1 period)

Fig. 6.2. DGp solutions with limiting for $p = 2, 3,$ and 4. The blue dots represent the cell average solutions, the blue curves the polynomial solutions, and the red curves the exact solution. The magenta dots with a value greater than 0.1 identify troubled cells where limiting is required.

Concerning the detection method, the (magenta) dots at level $u = 0.1$ represent good cells where limiting is not needed, and the dots at level $u > 0.1$ represent troubled cells where limiting is required. In the case of troubled cells, a dot at level $u = 0.15$, as in Fig. 6.2(c), means limiting has no effect even though the cell was marked as troubled. For Fig. 6.2(a), it appears that the cell just to the right of $x = 0.7$ should be flagged as troubled due to the infinite slope at the foot of the wave. However, since the second modes $u_{j,2}$ at the cell and its two neighbors are small, this cell is flagged as good due to the noise level of 10^{-5} allowed in criterion (5.29) for detection. If we use a noise level of 10^{-6}, this cell would be flagged as troubled. Also note that there are no troubled cells in Fig. 6.2(d); here, the data has become smooth enough and the polynomial solution is of a high enough order that the gaps at the interfaces are small, and all cells are good.

Unlike the linear case of advection where it is possible to have no troubled cells near a discontinuity, for the nonlinear case of a shock, the presence of troubled cells is nearly certain; this is due to the self-sharpening mechanism at shocks resulting from the converging characteristics (for advection, characteristics are parallel).

To illustrate that our detection method works well even with coarse meshes, we apply it to the $\sin 2\pi x$ periodic initial data on the domain $[0, 1]$. The first case employs DG2 with 3 cells, $\sigma = 0.15$, and 20 time steps, i.e., the wave travels 1 period. With such a coarse mesh, the detection still results in no troubled cells, and limiting or no limiting yield identical solutions. If we use just 2 cells, then limiting takes effect, and the solution is damped to essentially zero; the method considered this data as noise and damps it out. The second case employs DG3 with only 2 cells, $\sigma = 0.1$, and 20 time steps, or 1 period. For this case, the detection yields no troubled cells, and limiting has no effect.

Euler Equations in One-Spatial Dimension

The next set of tests deals with the Euler equations in one-spatial dimension (1D). Here, as is typical, limiting is applied to the (conservative) characteristic variables of troubled cells.

The next test, used by Sod (1978), is the Riemann problem with the initial data

$$(\rho_L, u_L, p_L) = (1, 0, 1) \quad \text{and} \quad (\rho_R, u_R, p_R) = (0.125, 0, 0.1)$$

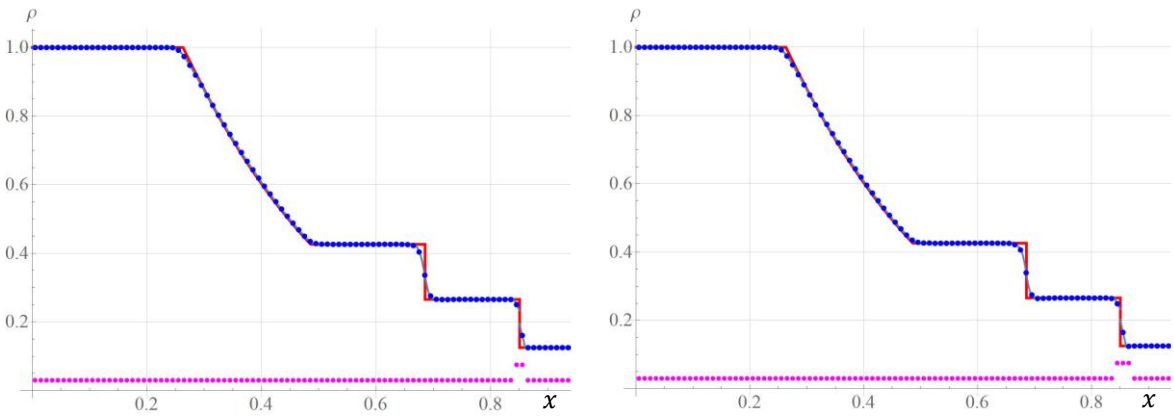
where the subscript L represents the condition for $0 \le x \le 0.5$, and R for $0.5 < x \le 1$. The 1D Euler equations are solved on the domain $[0, 1]$ with 100 cells. The final time is 0.2. Boundary conditions are by simple extrapolation from the interior.

Fig. 6.3 shows the DG solutions with limiting for (a) $p = 2$ after 200 time steps, each of size 0.001, and (b) $p = 3$ after 320 time steps, each of size 0.000625. Note that there are no oscillations, and discontinuities are well resolved.

The third test, due to Shu and Osher (1989), has several extrema in the smooth regions. It is used to test accuracy of methods near extrema in the presence of a shock. On the interval $-5 \le x \le 5$, a shock propagating at Mach 3 interacts with sine waves in density as described by the initial condition:

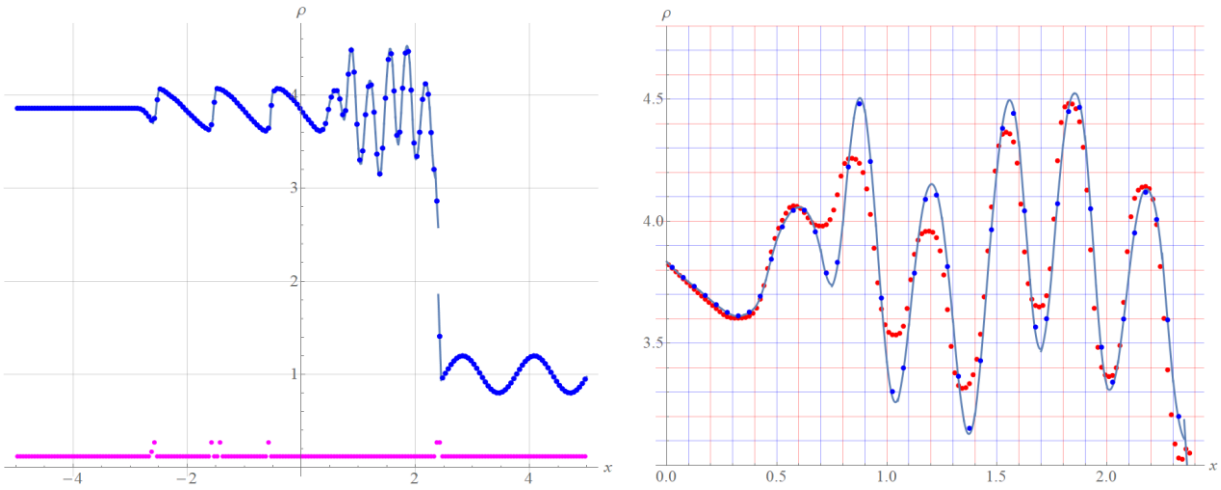
$$(\rho, u, p) = \begin{cases} (3.857, 2.629, 10.333), & -5 \le x \le -4 \\ (1 + 0.2 \sin 5x\,, 0, 1), & \text{otherwise.} \end{cases}$$

Fig. 6.4 shows the solution with 200 cells for $p = 2$ with a time step size of 0.002093 after 860 time steps. Figure 6.4(a) portrays the solution across the entire domain. At each discontinuity, only 2 or 1 cell is flagged as troubled. Figure 6.4(b) provides a close-up view of the solution on the domain $[0, 2.4]$. For comparison, the standard second-order solution using the Van Albada limiter with the same number of degrees of freedom, meaning 600 cells, are shown as red dots. Note that on this domain, there are no troubled cells, and the extrema are well resolved by the DG2 solution even after the shock passes through.



(a) DG2 solution (b) DG3 solution

Fig. 6.3: Solution densities. (a) DG2 solution after 200 time steps, each of size 0.001; there are 2 troubled cells at the shock and zero troubled cells at the contact discontinuity. (b) DG3 solution after 320 time steps, each of size 0.000625; there are 3 troubled cells at the shock and zero troubled cells at the contact discontinuity. The blue dots represent the cell average solutions, the blue curves the polynomial solutions, and the red curves the exact solution. The magenta dots with a value greater than 0.04 represent troubled cells where limiting is required.



(a) DG2 solution on whole domain (b) DG2 solution (in blue) on domain $[0, 2.4]$

Fig. 6.4: Solution densities with 200 cells, $p = 2$, time step size of 0.002093, after 860 time steps. (a) Solution on the whole domain; only 2 or 1 troubled cells are detected at each discontinuity. (b) Close-up view of solution on the domain $[0, 2.4]$; for comparison, the standard second-order solution using the Van Albada limiter with 600 cells are represented by red dots.

7. Conclusions and Discussion

In conclusion, we introduced a new high-order limiting technique capable of managing shocks while preserving accuracy in smooth regions without relying critically on a detection method. Initially, we described the DG method within the FR framework, where the solution is formulated via interpolation rather than standard projection, with the Radau polynomials and their derivatives playing a crucial role. We illustrated the behavior of the DG solution at jump discontinuities, which partially motivates our approach. Subsequently, we presented our limiting technique, where the key innovation lies in expanding the standard monotonicity limits to provide ample "room" near smooth extrema, ensuring that limiting has no effect. Near a discontinuity, these expanded limits essentially reduce to the original

limits, suppressing oscillations. The algorithm is simplified by applying the limiting bounds to the sum of magnitudes of all modes, linear and higher order, successively. We also presented a detection method identifying regions where limiting is unnecessary to reduce computing time. Numerical examples were provided to demonstrate the effectiveness of the new limiter.

Additional tests, as well as 2D and 3D applications, particularly the extension to unstructured (triangular) meshes, remain to be explored.

Acknowledgments

This research was sponsored by NASA's Transformational Tools and Technologies (TTT) Project of the Transformative Aeronautics Concepts Program under the Aeronautics Research Mission Directorate.

Appendices

A. Proof of formulas (2.7a,b) for the derivatives of Radau polynomials.

The derivative of the Legendre polynomial is given by (see, e.g., Wikipedia):

$$L_k{}' = (2(k-1)+1)L_{k-1} + (2(k-3)+1)L_{k-3} + (2(k-5)+1)L_{k-5} + \cdots$$

where the last term is L_0 if k is odd and $3L_1$ if k is even. That is, for odd k, or $k = 2m+1$,

$$L_{2m+1}{}' = L_0 + 5L_2 + 9L_4 \ldots + (2(2m)+1)L_{2m}.$$

For even k, or $k = 2m$,

$$L_{2m}{}' = 3L_1 + 7L_3 + 11L_5 \ldots + (2(2m-1)+1)L_{2m-1}.$$

The above two equations imply (2.7a, b), thus completing the proof.

B. Proof of formula (2.11): the derivative of the *right* Radau polynomial, $R_{R,k+1}{}'$, vanishes at the k interior points of the $k+1$ *left* Radau points.

Indeed, denote the $k+1$ left Radau points by $\xi_{L,m}$, $m = 1, \ldots, k+1$, where $\xi_{L,1} = -1$. For each m, let $\phi_{L,m}$ be the corresponding Lagrange polynomial (degree k) that takes on value 1 at $\xi_{L,m}$ and value 0 at all other $\xi_{L,l}$'s. Then for $m \geq 2$, $\phi_{L,m}(-1) = 0$. Thus, again for $m \geq 2$, using integration by parts, since the boundary terms vanish due to $R_{R,k+1}(1) = 0$ and $\phi_{L,m}(-1) = 0$,

$$\left(R_{R,k+1}{}', \phi_{L,m}\right) = -\left(R_{R,k+1}, \phi_{L,m}{}'\right).$$

Note that $\phi_{L,m}{}'$ is of degree $k-1$. Recall that by (2.5b) with k replaced by $k+1$, $R_{R,k+1} \perp \boldsymbol{P}_{k-1}$. Therefore, $\left(R_{R,k+1}, \phi_{L,m}{}'\right) = 0$. Thus, for $m \geq 2$, by the above, $\left(R_{R,k+1}{}', \phi_{L,m}\right) = 0$. Next, by applying the left Radau quadrature to $\left(R_{R,k+1}{}', \phi_{L,m}\right)$, and since $\phi_{L,m}\left(\xi_{L,l}\right) = 0$ for all $l \neq m$, we obtain, for $m \geq 2$,

$$R_{R,k+1}{}'\left(\xi_{L,m}\right)\phi_{L,m}\left(\xi_{L,m}\right) = R_{R,k+1}{}'\left(\xi_{L,m}\right) = 0.$$

This completes the proof.

C. Proof that for advection, the DG method can be cast in interpolation form (3.11).

Indeed, focusing on cell $E = E_j$, let ϕ be a test function, i.e., a polynomial of degree p (independent of t). The advection equation leads to the following requirement via projection,

$$\frac{\partial}{\partial t}(u_h, \phi)_E + ((u_h)_x, \phi)_E = 0.$$

The above involves no interaction among cells. To account for interaction, we use integration by parts,

$$\frac{\partial}{\partial t}(u_h, \phi)_E + (u_h\phi)_{\partial E} - (u_h, \phi_x)_E = 0.$$

For u_h in the boundary term $(u_h\phi)_{\partial E}$, we employ the common values $u^{\text{upw}}_{j+1/2}$ and $u^{\text{upw}}_{j-1/2}$ defined in (3.5). Thus, on cell j, the DG solution $u_h = u_j$ is required to satisfy, for any degree p polynomial ϕ,

$$\frac{\partial}{\partial t}(u_h, \phi)_E + u^{\text{upw}}_{j+1/2}\phi(1) - u^{\text{upw}}_{j-1/2}\,\phi(-1) - (u_h, \phi_x)_E = 0. \tag{A.1}$$

Integrate by parts again, but this time, use the values just inside cell j for the boundary terms,

$$\frac{\partial}{\partial t}(u_h, \phi)_E + ((u_h)_x, \phi)_E + (u^{\text{upw}}_{j+1/2} - u_j(1))\phi(1) - (u^{\text{upw}}_{j-1/2} - u_j(-1))\phi(-1) = 0.$$

For advection, by (3.5), $u^{\text{upw}}_{j+1/2} = u_j(1)$, and by (3.8), $(u^{\text{upw}}_{j-1/2} - u_j(-1)) = J_L$. The above implies

$$\frac{\partial}{\partial t}(u_h, \phi)_E + ((u_h)_x, \phi)_E - J_L\phi(-1) = 0. \tag{A.2}$$

Here, the term $-J_L\phi(-1)$ accounts for data interaction among neighboring cells.

Next, we wish to eliminate ϕ, i.e., we wish to express $\phi(-1)$ in the form (γ, ϕ) for some γ of degree p. To this end, let $\mathbb{R}$ be the real line. The mapping B from $\boldsymbol{P}_p$ to $\mathbb{R}$ defined by, $B(\phi) = \phi(-1)$ for any ϕ of degree p, is a linear functional on $\boldsymbol{P}_p$. Thus, there exists γ in $\boldsymbol{P}_p$ such that $B(\phi) = (\gamma, \phi)$. That is,

$$(\gamma, \phi) = \phi(-1).$$

In fact, $\gamma = -{R_{R,p+1}}'$. To prove this, using integration by parts,

$$\left({R_{R,p+1}}', \phi\right) = R_{R,p+1}(1)\phi(1) - R_{R,p+1}(-1)\phi(-1) - \left(R_{R,p+1}, \phi'\right).$$

Next, we use (2.5b), namely, $R_{R,p+1} \perp \boldsymbol{P}_{p-1}$. Since ϕ' is of degree $p-1$, $\left(R_{R,p+1}, \phi'\right) = 0$. Next, $R_{R,p+1}(1) = 0$ and $R_{R,p+1}(-1) = 1$. Therefore, the above implies

$$\left({R_{R,p+1}}', \phi\right) = -\phi(-1),$$

Thus, by the above, (A.2) implies

$$\frac{\partial}{\partial t}(u_h, \phi)_E + ((u_h)_x, \phi)_E + J_L\left({R_{R,p+1}}', \phi\right) = 0.$$

That is,

$$\frac{\partial}{\partial t}(u_h, \phi)_E + \left((u_h + J_L R_{R,p+1})_x, \phi\right)_E = 0.$$

Therefore, with $U_j = u_j + J_L R_{R,p+1}$, the DG method leads to $(u_h)_x = \left(U_j\right)_x$. This completes the proof.

References

[1] K. Asthana, M.R. López-Morales, and A. Jameson, "Non-linear stabilization of high-order Flux Reconstruction schemes via Fourier-spectral filtering," J. Comput. Phys., 303, pp. 269–294, 2015.

[2] B. Cockburn, S. Hou, and C.-W. Shu, "The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV. The multidimensional case," Mathematics of Computation, vol. 54, no. 190, pp. 545–581, 1990.

[3] T. Dzanica , W. Trojak, and F. D. Witherden, "Utilizing time-reversibility for shock capturing in nonlinear hyperbolic conservation laws," arXiv:2110.03653v2 [math.NA] 18 Apr 2022.

[4] M. Dumbser, O. Zanotti, R. Loubère, and S. Diot, "A posteriori subcell limiting of the discontinuous Galerkin finite element method for hyperbolic conservation laws," J. Comput. Phys., 278, 1, pp. 47-75, 2014.

[5] H. Hoteit, P. Ackerer, R. Mos´e, J. Erhel, and B. Philippe, “New two-dimensional slope limiters for discontinuous Galerkin methods on arbitrary meshes,” International Journal for Numerical Methods in Engineering, vol. 61, no. 14, pp. 2566–2593, 2004.
[6] J. S. Hesthaven and T. Warburton, “Nodal discontinuous Galerkin methods: algorithms, analysis, and applications,” Vol. 54, Texts in Applied Mathematics, Springer New York, 2008.
[7] H. T. Huynh, “Accurate upwind methods for the Euler equations,” SIAM J. Numer. Anal. 32, 1565, 1995.
[8] H. T. Huynh, “A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods,” AIAA Paper 2007-4079.
[9] H.T. Huynh, “A reconstruction approach to high-order schemes including discontinuous Galerkin for diffusion,” AIAA Paper 2009-403.
[10] H.T. Huynh, Z.J. Wang, and P.E. Vincent, “High-order methods for computational fluid dynamics: A brief review of compact differential formulations on unstructured grids,” Computers and Fluids, Vol 98, pp. 209-220, 2014.
[11] L. Krivodonova, “Limiters for high-order discontinuous Galerkin methods,” J. Comput. Phys., vol. 226, no. 1, pp. 879–896, 2007.
[12] L. Krivodonova, J. Xin, J.-F. Remacle, N. Chevaugeon, and J. E. Flaherty, “Shock detection and limiting with discontinuous galerkin methods for hyperbolic conservation laws,” Applied Numerical Mathematics 48 (3-4) 323–338, 2004.
[13] J. Kim , Hojun You , and C. Kim, “Shock-capturing PID Controller for high-order methods with data-driven gain optimization,” J. Comput. Phys., 508, 113015, 2024.
[14] Q. Lu, G. Liu, P. Ming, and Z.J. Wang, “A parameter-free gradient-based limiter for the FR/CPR method on mixed unstructured meshes,” AIAA 2019-3210.
[15] S.A. Moe, J.A. Rossmanith, and D.C. Seal, “A simple and effective high-order shock-capturing limiter for discontinuous Galerkin methods,” arXiv:1507.03024 [math.NA], 2015.
[16] J.S. Park, C. Kim, “Multi-dimensional limiting process for finite volume methods on unstructured grids,” Computers & Fluids, 65, 8–24, 2012.
[17] J.S. Park and C. Kim, “Higher-order multi-dimensional limiting strategy for discontinuous galerkin methods in compressible inviscid and viscous flows,” Computers & Fluids 96, 377–396, 2014.
[18] J.S. Park and C. Kim, “Hierarchical multi-dimensional limiting strategy for correction procedure via reconstruction,” J. Comput. Phys., 308, 57–80, 2016.
[19] J. S. Park, S.-H. Yoon, and C. Kim, “Multi-dimensional limiting process for hyperbolic conservation laws on unstructured grids,” J. Comput. Phys., 229 (3) 788–812, 2010.
[20] J.S. Park, M. Yu, C. Kim, and Z.J. Wang “Comparative study of shock-capturing methods for high-order CPR: MLP and artificial viscosity,” ICCFD8-2014-0067, 2014.
[21] Per-Olof Persson and Jaime Peraire, “Sub-cell shock capturing for discontinuous Galerkin methods,” AIAA Paper 2006-112.
[22] C.-.W. Shu and S. Osher, “Efficient implementation of essentially non-oscillatory shock-capturing schemes, II,” J. Comput. Phys., 83, pp. 32-78, 1989.
[23] G.A, Sod, “A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws,” J. Comput. Phys., 27(1):1–31, 1978.
[24] A. Suresh and H. T. Huynh, “Accurate Monotonicity-Preserving Schemes with Runge–Kutta Time Stepping,” J. Comp. Phys., 136, pp 83-99, 1997.
[25] B. Van Leer, ‘Towards the ultimate conservative difference scheme: IV. A New Approach to Numerical Convection,” J. Comput. Phys., 23, pp. 276–299, 1977.
[26] P. Woodward and P. Colella, “The numerical simulation of two-dimensional fluid flow with strong shocks,” J. Comput. Phys., 54, pp. 115–173, 1984.
[27] M. Yang and Z.J. Wang, “A Parameter-Free Generalized Moment Limiter for High-Order Methods on Unstructured Grids,” Adv. Appl. Math. Mech., Vol. 1, No. 4, pp. 451-480, 2009.
[28] H. You, J. Kim, and C. Kim, Deneb: An open-source high-performance multi-physical flow solver based on high-order DRM-DG method, Computer Physics Communications 286, 108672, 2023.
[29] H. You and C. Kim, “High-order multi-dimensional limiting strategy with subcell resolution I. Two-dimensional mixed meshes,” J. Comput. Phys., 375, 1005–1032, 2018.