High-order methods for computational fluid dynamics: A brief review of compact differential formulations on unstructured grids

H.T. Huynh, Z.J. Wang, P.E. Vincent

1. Introduction

In the field of Computational Fluid Dynamics (CFD), low-order methods are generally robust and reliable; as a result, they are routinely employed in practical calculations. For the same computing cost, high-order methods can provide considerably more accurate solutions, but they are more complicated and less robust. The need to improve and develop new high-order methods with favorable properties has attracted the interest of many researchers as evidenced by the recently held First (2012) and Second (2013) International Workshops on High-Order CFD Methods (http://zjwang.com/hiocfd.html and http://www.dlr.de/as/desktopdefault.aspx/tabid-8170/13999_read-35550/).

The Discontinuous Galerkin (DG) method is currently among the most widely used high-order numerical methods for solving the compressible Navier–Stokes equations on unstructured meshes. It was introduced for the neutron transport equation by Reed and Hill [45], analyzed by LaSaint and Raviart [33] and developed and made popular for fluid dynamics equations by Cockburn, Shu, Bassi, Rebay, and others (see e.g., [11,3,4,13,14,48], and the references therein). Efficient DG schemes storing data at nodal points known as nodal DG methods were presented by Hesthaven and Warburton [21].

Alternative approaches to high-order accuracy employing the differential form (as opposed to traditional DG which employs the integral form) have been proposed. Kopriva and collaborators pioneered this approach with the staggered-grid spectral method on quadrilateral meshes [32,30,31,27]. The approach was extended to triangular meshes by Liu et al. [39] and named Spectral Difference (SD). Solutions for a wide range of problems by these methods were provided by Wang et al. [58] and Liang et al. [36,37]. Another class of schemes called Spectral Volume (SV) presented by Wang et al. [59] is based on the idea of subdividing each cell into subcells or control volumes in a structured manner. A review of these as well as other types of high-order schemes can be found in [53].

Recently, an approach to high-order accuracy with the advantage of simplicity and economy called Flux Reconstruction (FR) was introduced by Huynh [23,24,26,54]. For the case of one spatial dimension (1D), the approach amounts to evaluating the derivative of a discontinuous piecewise polynomial function by employing its straightforward derivative estimate together with a correction, which accounts for the jumps at the interfaces. The FR framework unifies several existing schemes: with appropriate choices of correction terms, it recovers DG, SD, as well as SV, and the FR versions...
are generally simpler and more economical than the original versions. In addition, the approach results in numerous new methods that are stable and super accurate, i.e., more accurate than expected (also known as super convergent). The FR framework was applied to ordinary differential equations in [25], and the result is the Radau IIA collocation method.

Whereas extension of 1D FR to a quadrilateral mesh is straightforward via tensor product, that for unstructured triangular meshes is nontrivial since tensor product no longer applies. Here, Wang and Gao [56] showed that the derivative correction can be extended without the reconstruction concept. The resulting method was applied to solve the 2D Euler equations and named LCP (lifting collocation penalty). Extensions to the 2D Navier–Stokes equations on meshes of mixed elements were carried out by Gao and Wang [15] and Gao et al. [17]. Extensions to the 3D Euler and Navier–Stokes equations on mixed meshes were presented by Haga et al. [18,19] and Wang et al. [57] and to dynamic meshes by Yu et al. [64]. It was later shown by Huynh [26] that the reconstruction concept applies to triangles as well. Due to the tight connection between FR and LCP, the involved authors combined the names and call them the CPR method (Correction Procedure via Reconstruction). PnPm-CPR schemes for the Navier–Stokes equations were studied by Shi et al. [47]. A modification to assure that the resulting method is conservative was presented by Gao and Wang [46]. Comparisons for various types of benchmark problems were presented by Li and Wang [34].

Reconstruction. PnPm-CPR schemes for the Navier–Stokes equations are motivated by the desire to capture the solution accurately and efficiently. In the case of FR methods, the reconstruction is carried out through a projection onto the space of polynomials of degree $m$.

In this paper, we present a brief review of recent developments for the FR/CPR schemes as well as some key pacing items. Basic concepts of the DG and FR methods and their relations for a simple integration problem are presented in Section 2. Section 3 deals with conservation laws including the diffusion equation. Section 4 discusses 2D and 3D extensions of FR methods. Stability proofs are sketched in Section 5. Section 6 contains representative numerical results. Additional recent and ongoing research is described in Section 7. Some pacing items and areas for future research are discussed in Section 8. Finally, conclusions and discussion are presented in Section 9.

2. A simple integration problem

We present the key ideas of both the integral and differential formulations and their relations using the following simple integration problem. On $I = [−1, 1]$, solve

$$u'(x) = f(x), \quad u(−1) = u_0.$$  \hfill (2.1)

The exact solution is

$$u_{exact}(x) = u_0 + \int_{−1}^{x} f(\xi) d\xi.$$  \hfill (2.2)

The standard DG formulation is concise and results in a matrix equation discussed later, but it does not convey some key properties of the solution. Here, we will show that the DG solution $u_0$ of degree $K − 1$ can be obtained as follows. First, project $f$ onto the space of polynomials of degree $K − 1$ resulting in $f_h$; next, set $U_h(x) = u_0 + \int_{−1}^{x} f_h(\xi) d\xi$; thus, $U_h$ is of degree $K$. The solution $u_0$ is determined by the values of $U_h$ at the $K$ right Radau points (more on these points later).

2.1. Examples

The proof for the above DG solution will be carried out after the following two examples, which convey the behavior of solutions. For the first example, find the linear DG solution of

$$u'(x) = 2x, \quad u(−1) = 1.$$  \hfill (2.3)

Here, the cubic projection of $f$ is $f_h = 4x^3$; next, $U_h(x) = u_{exact}(x) = x^4$. The cubic DG solution is determined by the values of $U_h$ at the four right Radau points shown in Fig. 2.1(b).

In both examples, $U_h$ provides an approximation one degree higher than $u_0$, and the values for both $U_h$ and $u_0$ at the $K$ right Radau points are the same. For this integration problem, since the ’wind direction’ is unique, from left to right due to $u_0$, the polynomial $U_h$ is well defined. For the case of the Euler equations, however, the ’wind direction’ is not unique and the definition of $U_h$ becomes unclear. As will be shown, the FR technique of constructing Runge–Kutta scheme by Vermeire et al. [50].

Next, we discuss some recent additional contributions to the development of FR/CPR. Suppressing oscillations near shocks via localized artificial diffusivity or LAD was carried out by Miyaji [43] and Haga et al. [20]. In the latter, FR methods for body-fitted Cartesian unstructured grids were also developed. Extensions of a particularly simple FR/CPR scheme called $g_2$ to the Navier–Stokes equations on moving and deforming domains were presented by Liang et al. [35]. A comparison of computational efficiencies of SD and CPR methods were carried out in Liang et al. [35]. Applications to turbulent internal flows for turbomachinery and mitigation of aliasing errors were discussed by Lu et al. [41,40]. An interface element approach dealing with non-conforming polynomials together with p-adaptation for viscous compressible flow simulations was elaborated by Cagnone and Nadarajah [67]. In addition, an implicit large eddy simulation (ILES) solver was developed for CPR scheme using a third-order singly diagonal implicit Runge–Kutta scheme by Vermeire et al. [50].

2.2. Preparations

The projection operator and various special polynomials play an important role. Therefore, they are briefly reviewed. With any non-negative integer $m$, let $P_m$ be the space of polynomials of degree $m$.
or less. For any two functions \(v\) and \(w\) on \(I = [-1, 1]\), denote their inner product by

\[
(v, w) = \int_{-1}^{1} v(x)w(x)dx.
\]

The \(L^2\) norm of \(v\) is

\[
\|v\| = \left( \int_{-1}^{1} (v(x))^2dx \right)^{1/2}.
\]

A polynomial \(v\) is orthogonal to \(P_m\) on \(I\) if, for each \(l\), \(0 \leq l \leq m\),

\[
(v, x^l) = \int_{-1}^{1} v(x)x^l dx = 0.
\]

The criterion of orthogonality to \(P_m\) provides \(m + 1\) conditions.

2.3. Legendre polynomials

For each integer \(m \geq 0\), the Legendre polynomial \(P_m\) on \(I\) is defined as the unique polynomial of degree \(m\) that is orthogonal to \(P_{m-1}\) and \(P_m(1) = 1\). The Legendre polynomials are given by a recurrence formula (e.g. [22]):

\[
P_0(x) = 1, \quad P_1(x) = x,
\]

and, for \(k \geq 2\),

\[
P_k(x) = \frac{2k-1}{k} x P_{k-1}(x) - \frac{k-1}{k} P_{k-2}(x).
\]

The first few Legendre polynomial's are

\[
P_0(x) = 1, \quad P_1(x) = x, \quad P_2(x) = \frac{1}{2} (3x^2 - 1), \quad P_3(x) = \frac{1}{2} (5x^3 - 3x), \quad P_4(x) = \frac{1}{8} (35x^4 - 30x^2 + 3).
\]

They are plotted in Fig. 2.2(a).

Useful properties of the Legendre polynomials are listed below. If \(k > m\), then \(P_k\) is orthogonal to \(P_m\). Next, \(P_k\) is an even function (involving only even powers of \(x\)) for even \(k\), and an odd function for odd \(k\). For all \(k\),

\[
P_k(-1) = (-1)^k, \quad P_k(1) = 1.
\]

In addition,

\[
\|P_k\|_2 = \left( P_k, P_k \right) = \frac{2}{2k + 1}.
\]

For later use, the derivative values at the end points are

\[
P_k'(1) = \frac{2(k+1)}{2}.
\]

The zeros of \(P_k\) are the \(k\) Gauss points on \([-1,1]\).

Denote the projection operator onto \(P_m\) by \(P_{m}\). Then for any function \(w\),

\[
Pr_m(w) = \sum_{k=0}^{m} \frac{\left( w, P_k \right)}{\|P_k\|^2} P_k.
\]

2.4. Radau polynomials

The right Radau polynomial of degree \(k \geq 1\) is defined by

\[
R_{k}(x) = \left( \frac{-1}{2} \right)^k (P_k - P_{k-1}).
\]

The zeros of \(R_{k}\) are the \(k\) right Radau points on \([-1,1]\). The first few right Radau polynomials are plotted in Fig. 2.2(b). The above definition implies that \(R_{k}\) is orthogonal to \(P_{k-2}\) since both \(P_k\) and \(P_{k-1}\) have this property. In addition,

\[
R_{k}(-1) = 1 \quad \text{and} \quad R_{k}(1) = 0.
\]

Note that \(R_{k,0}\), which is of degree \(k\), is defined by the above two conditions together with the \(k-1\) conditions that it is orthogonal to \(P_{k-2}\). We can think of \(R_{k}\) as a polynomial approximating the step down function \(s\) defined by \(s(-1) = 1\) and \(s(x) = 0\) for \(-1 < x \leq 1\). At \(x = \pm 1\),

\[
R_{k,0}(x) = \begin{cases} 0 & \text{if } x = -1, \\ \frac{1}{2} (P_k - P_{k-1}) & \text{if } x = 1. \end{cases}
\]

---

**Fig. 2.1.** (a) Linear DG solution for problem (2.2) and (b) cubic DG solution for problem (2.3).

**Fig. 2.2.** (a) Legendre polynomials and (b) right Radau polynomials.
We note in passing that the above expression for \( R'_{k;k}(-1) \) is related to the time step size restriction (or CFL condition) proportional to \( 1/k^2 \).

### 2.5. Lobatto polynomials

The Lobatto polynomial of degree \( k \geq 2 \), is defined by

\[
\mathcal{L}_k = P_k - P_{k-2}.
\]

(2.9)

The zeros of the Lobatto polynomial of degree \( k \) are the \( k \) Lobatto points; they include the two boundaries \( \pm 1 \). The first few Lobatto polynomials are shown in Fig. 2.3(a).

### 2.6. Lagrange polynomials

Let \( k \) be an integer \( \geq 1 \). For the nodal methods (e.g., [21]), denote the nodes or solution points on \( I \) by \( \tilde{z}_k \), \( k = 1, \ldots, K \), which are typically the Gauss or Lobatto points. Let \( l_k \) be the associated Lagrange polynomials, i.e., for each \( k \), \( l_k \) is of degree \( K - 1 \) taking on value 1 at \( \tilde{z}_k \) and 0 at all other \( \tilde{z}_m \).

\[
l_k(\zeta) = \prod_{m=1, m \neq k}^{K} \frac{\zeta - \tilde{z}_m}{\tilde{z}_k - \tilde{z}_m}.
\]

(2.10)

Lagrange basis functions for the case of four Lobatto points are shown in Fig. 2.3(b).

### 2.7. DG formulation

Let the basis functions \( \phi_k \) for \( P_{K-1} \) be either the Lagrange polynomials \( l_k \) (nodal DG) or the Legendre polynomials (modal DG) \( P_{K-1}, k = 1, \ldots, K \). The solution is approximated by

\[
u_h = \sum_{k=1}^{K} u_k \phi_k.
\]

(2.11)

Let \( \phi \) be a test function, which is in \( P_{K-1} \); typically, \( \phi \) is one of the basis functions \( \phi_k \), \( k = 1, \ldots, K \). Then the weak form for (2.1) or \( u' = f \) is the following requirement

\[
(u', \phi) = (f, \phi).
\]

(2.12)

Note that the above does not involve the condition \( u(-1) = u_1 \) (the equation involving this condition is (2.16) or (2.17) later).

Let \( f_h \) be the projection of \( f \) onto \( P_{K-1}; f_h = \operatorname{Pr}_{P_{K-1}}(f) \) via (2.5) with \( m = K - 1 \). Since \( \phi \) varies on \( P_{K-1} \), we have \( (f, \phi) = (f_h, \phi) \), i.e., we can project first, then apply the inner product. The above implies

\[
(u', \phi) = (f_h, \phi).
\]

(2.13)

Going back to (2.1), the solution of \( u' = f_h \) and \( u(-1) = u_1 \) is

\[
U(x) = u_1 + \int_{-1}^{x} f_h(\zeta) d\zeta.
\]

(2.14)

Clearly, \( U \) is of degree \( K \) since \( f_h \) is of degree \( K - 1 \). But the solution we are seeking is \( u_h \) of degree \( K - 1 \). The question is: how do we obtain \( u_h \) from \( U \)? Or, to put it differently, how does the DG solution \( u_h \) relate to \( U \)?

To answer the above question, we first define the DG solution. For the DG method, \( u_h(-1) \) is allowed to be and is generally different from \( u_r \). Applying integration by parts to the left hand side of (2.13) and with \( u \) replaced by \( u_h \),

\[
[u_h \phi]_{-1}^1 - (u_h, \phi') = (f_h, \phi).
\]

(2.15)

To involve the initial condition, we replace \( u_h(-1) \) in the boundary term above by \( u_1 \). Thus, the DG solution \( u_h \) is required to satisfy, for all \( \phi \) in \( P_{K-1} \),

\[
u_h(1)\phi(1) - u_h\phi(-1) - (u_h, \phi') = (f_h, \phi).
\]

(2.16)

Applying integration by parts again to \( (u_h, \phi') \),

\[-[u_h - u_h(-1)]\phi(-1) + (u_h', \phi) = (f_h, \phi).
\]

(2.17)

This time, for the boundary term, \( u_h(-1) \) is employed. The above, the result of integrating by parts twice, is the ‘strong form’ whereas (2.16) is the ‘weak form’ [21]. The solution can be obtained by solving either (2.16) or (2.17) where \( u_h \) is of the form (2.11), and \( \phi \) is replaced by \( \phi_k \), \( k = 1, \ldots, K \); the result is a matrix equation for the \( K \) unknowns \( u_k \).

### 2.8. Flux Reconstruction formulation

Our goal is to eliminate the test function in (2.17) so that the integral formulation results in a differential one. To this end, we raise the following question: can we find a polynomial \( g_{LR} = g \) on \( I \) which possesses the property that for any \( \phi \) of degree \( K - 1 \) or less,

\[-\phi(-1) = (g', \phi).
\]

(2.18)

If such a polynomial exists, we can combine (\( u_h, \phi \)) in (2.17) with \( (g', \phi) \) and eliminate the test function. From (2.17), since \( f_h \) is of degree \( K - 1 \), we require \( g \) to also be of degree \( K - 1 \); as a result, \( g \) is of degree \( K \). Applying integration by parts to the right hand side of (2.18), we have

\[-\phi(-1) = g(1)\phi(1) - g(-1)\phi(-1) - (g, \phi').
\]

(2.19)

The above holds if \( g \) satisfies

\[g(-1) = 1, \quad g(1) = 0,
\]

(2.20)

and, for all \( \phi \) in \( P_{K-1} \),

\[\text{Fig. 2.3. (a) Lobatto polynomials and (b) Lagrange polynomials for the four Lobatto points.}\]
\[(g, \phi') = 0. \tag{2.21}\]

Since \(f\) is of degree \(K - 1\), \(\phi'\) is of degree \(K - 2\); moreover, \(\phi'\) spans \(P_{K-2}\) as \(\phi\) spans \(P_{K-1}\). The above condition implies that \(g\) is orthogonal to \(P_{K-2}\), i.e., for any polynomial \(\phi\) of degree \(K - 2\),
\[(g, \phi) = 0. \tag{2.22}\]

Orthogonality to \(P_{K-2}\) provides \(K - 1\) conditions; \((2.20)\) provides the other two. Loosely put, the condition \((g, -1) = 1\) deals with the jump at the left boundary whereas the above together with \(g(1) = 0\) implies that away from the left boundary, \(g\) approximates the zero function. By the discussion after \((2.7)\), \(g\) is the right Radau polynomial,
\[g = R_{k\times}. \tag{2.23}\]

Thus, the answer to the question posed for \((2.18)\) is positive. Note that \(g = \text{grk}\), which corrects for the jump at the left boundary, is the right Radau polynomial (vanishing at \(x = 1\)).

We now return to our task of eliminating \(\phi\). With \(g\) as above, \((g', \phi) = -\phi(1)\). Thus, by \((2.17)\),
\[|u_k - u_0(-1)| + (u_k, \phi) = (f_k, \phi). \tag{2.24}\]

What is crucial here is that \(\phi\) can be eliminated: since \(g', u_0\), and \(f_0\) are of degree \(K - 1\) or less, and \(\phi\) spans \(P_{K-1}\), the above implies
\[(|u_k - u_0(-1)| + u_0) = f_0. \tag{2.25}\]

Therefore, \(|u_k - u_0(-1)| + u_0\) is a polynomial of degree \(K\) whose value at the left boundary is
\[|u_k - u_0(-1)| + u_0(-1) = u_k. \tag{2.26}\]

The above two conditions imply, with \(U\) defined by \((2.14)\),
\[|u_k - u_0(-1)| + u_0 = U. \tag{2.27}\]

Next, denoting the \(K\) right Radau points by \(r_k, k = 1, \ldots, K\). Since \(g\) vanishes at these points, \(u_k(r_k) = U(r_k)\).
\[u_k(r_k) = U(r_k). \tag{2.28}\]

Thus, with \(U\) of degree \(K\) defined by \((2.14)\), the DG solution \(u_0\) of degree \(K - 1\) is defined by the values \(U\) at the \(K\) right Radau points.

The key idea of FR is the following. Given \(u_0\) which does not match the boundary condition \(u_0\), we can add the correction term \(|u_k - u_0(-1)|\) to \(u_0\) so that the resulting \(U\) takes on the left boundary value \(u_0\) when it traverses the property of "best possible" approximation to \(u_0\). The function \(U\) is defined as the "continuous flux function" as opposed to \(u_0\), which is discontinuous across the cell boundaries. It turns out that for conservation laws, there are numerous ways to define \(g\) so that the resulting scheme is stable.

For the ordinary differential equation (ODE) \(u'(x) = f(x, u, u') = 0\), there is only one "wind direction", i.e., \(x\) moves forward. The above argument shows that for such a case, it is sensible to use the \(K\) right Radau points as collocation or quadrature points, which implies making use of \(U\). The resulting scheme is identical to an implicit collocation Runge–Kutta scheme called Radau IIA [25].

3. FR/CPR methods for the one-dimensional case

3.1. Conservation laws

Consider the conservation law
\[u_t + f_x = 0 \tag{3.1}\]

with initial condition \(u(x, 0) = u_{\text{in}}(x)\) and the flux \(f\) depends on \(u\). The solution \(u\) is assumed to be periodic or of compact support so that boundary conditions are trivial.

Let the domain of calculation be divided into (possibly nonuniform) cells or elements \(E_i, j = 1, 2, \ldots\) Denote the center of \(E_i\) by \(x_i\) and its width by \(h_i\). With \(z\) varying on \(l = [-1, 1]\) and \(x\) on \(E_i\), the linear function mapping \(l\) onto \(E_i\) and its inverse are
\[x(z) = x_i + \frac{z}{2} h_i, \quad \text{and} \quad \xi(x) = 2(x - x_i)/h_i. \tag{3.2}\]

In addition, denote the nodes or solution points on \(l\), which are typically the Gauss or Lobatto points, by \(\xi_k, k = 1, \ldots, K\). Then, they relate to the nodes on \(E_i\) by
\[x_{ik} = x(\xi_k) = x_i + \frac{\xi_k h_i}{2}. \tag{3.2}\]

The global derivative, e.g., \(f_n\), can be obtained from the local one via the chain rule \(\frac{d}{dt} = \frac{\partial}{\partial x}\).

Whereas the FR approach can be formulated in modal form, for simplicity, we discuss only the nodal form. On \(E_i\), with the Lagrange basis functions \(\phi_{ik}(x) = \varphi_{ik}(\xi), k = 1, \ldots, K\), the solution be approximated by a polynomial of degree \(K - 1\) denoted by \(u_i\)
\[u_{ik} = \sum_{k=1}^{K} u_{ik} \phi_{ik}. \tag{3.3}\]
Here, \(u_{ik}\) depends on \(t\) and \(\phi_{ik}\) on \(x\).

At time level \(n\), suppose \(u_{ik}\) are known for all \(j\) and \(k\). We wish to calculate \(du_{ik}/dt = du_{ik}/dt\) at \(t = t^n\), i.e., to calculate \((f_{ik})_{k,j}\). Then, we march in time by, say, a Runge–Kutta method.

With \(f_{ik} = f(u_{ik})\), let \(f_{ik}\) be the polynomial of degree \(K - 1\) interpolating \(f_{ik}\), \(k = 1, \ldots, K\).
\[f_{ik} = \sum_{k=1}^{K} u_{ik} \phi_{ik}. \tag{3.4}\]

The flux polynomials \(f_{ik}\) form a function, which is generally discontinuous across cell interfaces and \(f_i\) is called the discontinuous flux function. Note that \((f_{ik})\) involves no interaction of data among cells.

To account for interaction, we construct a continuous flux function, which approximates the discontinuous function in some sense, and then calculate its derivative. The continuous flux function will be obtained by adding a correction to the discontinuous one. As a result, we still need the derivative of the discontinuous function. At each \(\xi_k, k = 1, \ldots, K\), it is easy to derive the derivative matrix \([\partial f]/[\partial u]\) where
\[f_{ij} = (f_{ik})_{j,k}. \tag{3.5}\]

Instead of differentiating \(f_{ik}\) as above, we can use the chain rule
\[f_{ij} = (f_{ik}(u_{ik}))/k. \tag{3.6}\]

Wang and Gao [56] found that for the Euler equations, the chain rule yields more accurate solutions. It is, however, generally nonconservative for nonlinear problems.

At each interface \(x_{j+1/2}\), set
\[u_i = u_{i+1/2}\] and \(u_k = u_{j+1/2}\). \tag{3.7}\]

From these two values, we can obtain a common flux (shared by the two adjacent cells) denoted by \(f_{i+1/2,\text{com}}\). For advection problems, the common flux is typically an upwind flux; for diffusion problems, however, it is usually a centered quantity. This flux is often called the 'numerical flux' by the DG community.

Next, we reconstruct the flux by a continuous function \(F\) such that on each cell \(E_i, F\) is a polynomial denoted by \(F_i\) approximating the discontinuous flux function \(f_i\). To assure continuity across cells, \(F_i\) is required to take on the common flux values at the two interfaces:
\[F_i(x_{j+1/2}) = F_{i+1/2,\text{com}} \quad \text{and} \quad F_i(x_{j-1/2}) = F_{j-1/2,\text{com}}. \tag{3.8}\]

In addition, \(F_i\) is required to be of degree \(K\) so that its derivative is of degree \(K - 1\), matching that of \(u_{ik}\). Switching to the local description,
\[ F_j(-1) - f_j(-1) = f_{j-1/2, \mathrm{com}} - f_j(-1) \quad \text{and} \quad F_j(1) - f_j(1) = f_{j+1/2, \mathrm{com}} - f_j(1). \]  

(3.9)

Therefore, \( F_j - f_j \) takes on the above prescribed left and right correction values, which is of degree \( K_1 \), and approximates the zero function in some sense.

We now separate the prescription of the correction at the left interface from that of the right. On \( I \), let \( g_{LB} \) be the correction function for the left boundary defined by

\[ g_{LB}(-1) = 1, \quad g_{LB}(1) = 0, \]  

(3.10)

and \( g_{LB} \) is a polynomial of degree \( K \) approximating the zero function in some sense. Let \( g_{RB} \) be the correction function for the right boundary defined by reflection

\[ g_{RB}(\xi) = g_{LB}(-\xi). \]  

(3.11)

For the left interface \( x_{j-1/2} \), the polynomial

\[ f_j(-1) + f_{j-1/2, \mathrm{com}} - f_j(-1) g_{LB}(\xi). \]  

(3.12)

provides a correction to \( f_j(\xi) \) by changing the flux value at this interface from \( f_j(-1) \) to \( f_{j-1/2, \mathrm{com}} \) while leaving the value at the right interface unchanged, namely \( f_j(1) \). Next, the polynomial

\[ F_j(\xi) = f_j(\xi) + f_{j-1/2, \mathrm{com}} - f_j(-1) g_{LB}(\xi) + f_{j-1/2, \mathrm{com}} - f_j(1) g_{RB}(\xi) \]  

(3.13)

provides corrections to both interfaces: \( F_j(-1) = f_{j-1/2, \mathrm{com}} \) and \( F_j(1) = f_{j+1/2, \mathrm{com}} \). Thus, \( F_j \) is of degree \( K \); takes on the two common flux values, and approximates \( f_j \) in the same sense that \( g_{LB} \) and \( g_{RB} \) approximate the zero function. The derivative of \( F_j \) at the solution point \( \xi \) is

\[ (F_j)_\xi = (f_j)_\xi + f_{j-1/2, \mathrm{com}} - f_j(-1) g_{LB}(\xi) + f_{j-1/2, \mathrm{com}} - f_j(1) g_{RB}(\xi) \]  

(3.14)

The derivative \( (F_j)_\xi \) follows. The solution \( u_{j,k} \) can then be updated via, say, a Runge–Kutta method.

What is crucial in (3.14) is that at each solution point, the derivative \( (F_j)_\xi \) of the continuous flux function is obtained by correcting the derivative \( (f_j)_\xi \) of the discontinuous flux function. The correction amount is straightforward once the values \( g_{LB}(\xi_k) \) and \( g_{RB}(\xi_k) \) are known. These derivative values, in turn, can easily be derived once \( g_{LB} \) and \( g_{RB} \) are defined on \( I \).

We summarize the FR/CPR algorithm below.

### 3.2. Algorithm

At time level \( n \), suppose \( u_{j,k} \) are known for all \( j \) and \( k \).

1. At each interface \( j+1/2 \), if the left and right values of \( u \) are not available, calculate them; then estimate and store the common (upwind) fluxes at all interfaces.
2. In the cell \( j \), for \( k = 1, \ldots, K \), evaluate \( f_{j,k} = f(u_{j,k}) \); then obtain \( (f_{j,k})_\xi \) of the discontinuous flux function by (3.5). Alternatively, the chain rule (3.6) can be employed.
3. At the two interfaces of \( E_j \), get the corrections \( f_{j-1/2, \mathrm{com}} - f_j(-1) \) and \( f_{j+1/2, \mathrm{com}} - f_j(1) \). At the solution points, evaluate \( (F_{j,k})_\xi \) by (3.14) and then \( (F_{j,k})_\xi \).
4. March in time by, say, a Runge–Kutta method. This completes the algorithm.

### 3.3. Correction functions

Next, we discuss various choices for the correction function \( g = g_{LB} \). For more details, see [23,24]. Recall that \( g \) is determined by (3.10) together with the \( K-1 \) conditions. These different choices for \( g \) result in the DG, SD, SV, as well as new schemes. Thus, the FR approach provides a unifying framework for these methods.

The first choice for \( g \) requires that it is orthogonal to \( P_{K-2} \), which means it is the right Radau polynomial as in (2.23), and the resulting scheme is identical to DG,

\[ g = g_{DG} = R_{K-1}. \]  

(3.15)

The condition of orthogonality to \( P_{K-2} \) can be relaxed. It was verified via Fourier analysis [23,24] that if \( g \) is orthogonal to \( P_{K-3} \) the resulting scheme is stable (the converse is not true, however). Since both \( R_{K-1} \) and \( R_{K-2} \) are orthogonal to \( P_{K-3} \), such a correction function can be written as,

\[ g = 2R_{K-1} + (1-x)R_{K-1}. \]  

(3.16)

where, \( 0 \leq x \leq 1 \) and \( x \) remains to be determined.

The second choice for \( g \), denoted by \( g_2 \) or \( g_{Lump,lo} \) for ‘lumping for Lobatto points’, is defined as follows. Since a steeper correction function tends to result in a scheme with a smaller CFL limit, to make \( g \) less steep, the extra condition is obtained by pushing one of the zeros to the right boundary, i.e., \( \xi = 1 \) is a zero of multiplicity two. After some algebra,

\[ g_2 = g_{Lump,lo} = \frac{K-1}{2K-1} R_{K-1} + \frac{K}{2K-1} R_{K-2}. \]  

(3.17)

The function \( g_2 \) has the following remarkable property. Among the \( K \) Lobatto points, \( g_2 \) vanishes at \( K-1 \) of them; the exception is the left boundary.

The final choice here for \( g \) requires that \( g \) vanishes at the \( K-1 \) Gauss points:

\[ g_{Ga} = \frac{K}{2K-1} R_{K-1} + \frac{K-1}{2K-1} R_{K-2}. \]  

(3.18)

Whereas the original staggered-grid and SD schemes are mildly unstable, the above provides a modification using the \( K-1 \) Gauss points as flux points. The resulting scheme is stable for all \( K \).

It can be shown by calculations using Fourier analysis that the scheme using \( g_{DG} \) is stable and accurate to order \( 2K-1 \), and the schemes using \( g_2 \) and \( g_{Ga} \) are stable and accurate to order \( 2K-2 \). In general, if \( g \) is orthogonal to \( P_m \), the resulting scheme is accurate to order \( K+1+m \). If \( g \) is orthogonal to \( P_{K-3} \), i.e., \( g \) is of the form (3.16), the resulting scheme is Fourier stable. Again, for more details, see [23,24]. Stability proofs for FR schemes will be discussed in Section 5.

Also note that the steepest slope of \( g \), which often takes place at the left boundary, relates to the time step size limit. For example, for DG, as in (2.8), \( g' = R_{K-1}(-1) = -k/2 \); when an explicit Runge–Kutta method is employed, it is well known that the time step size limit for the DG scheme of degree \( K-1 \) is roughly proportional to \( 1/k^2 \).

### 3.4. The diffusion equation

On \((-\infty, \infty)\), consider the diffusion equation,

\[ u_t = u_{xx}, \]  

(3.19)

with initial condition

\[ u(x, 0) = u_0(x) \]  

(3.20)

As in the case of (3.1), assume that the data \( u_{j,k} \) are known at time level \( n \). We wish to evaluate the second derivative in a manner which takes into account the data interaction among cells. For simplicity and efficiency, the stencil of the scheme is required to remain compact in the sense that the second derivative evaluation in a cell involves the data of only that cell and the two immediate neighbors.
3.5. Common values and corrected derivative estimates

The first task is to estimate \( u_t \) at the solution points \( x_{j,k} \). Since the function \( \{ u \} \) is discontinuous across the interfaces, to estimate \( u_t \), we first reconstruct \( u \) by a piecewise polynomial function \( \{ u_j^C \} \), which is continuous across the cell interfaces, and on each \( E_{x_j} \), is of degree \( K \) and approximates \( u_t \) (the superscript ‘C’ stands for ‘continuous’ or ‘corrected’). The derivative approximation \( \{ u_j^C \} \) accounts for the data interaction.

In order for \( \{ u_j^C \} \) to be continuous at the interfaces, \( u_j^C \) and \( u_{j+1}^C \) must take on the same value at \( x_{j+1/2} \). Thus, at each interface, we need to define a common interface value (or common value). Here, for a diffusion problem, we use a centered-type quantity: with \( u_t \) and \( u_0 \) given by (3.7),

\[
\begin{align*}
u_{com} &= u_{j+1/2, com} = (u_t + u_R)/2.
\end{align*}
\]

(3.21)

The above formula was employed by Bassi and Rebay [3,5]. A more general formula is the weighted average, with \( 0 \leq \kappa \leq 1 \),

\[
\begin{align*}
u_{com} &= u_{j+1/2, com} = K u_t + (1 - K) u_R.
\end{align*}
\]

(3.22)

For \( \kappa = 0 \) or \( \kappa = 1 \), we have the one-sided formula used in the local DG or LDG [12] as well as the compact DG or CDG methods [44].

Next, we require \( u_j^C(X) \) to take on the common values \( u_{j-1/2, com} \) at \( x_{j-1/2} \) and \( u_{j+1/2, com} \) at \( x_{j+1/2} \), to be of degree \( K \), and to approximate \( u_j(X) \). That is, in the local coordinate,

\[
\begin{align*}
u_j^C(\zeta) &= u_j(\zeta) + [u_{j-1/2, com} - u_j(1)] g_{LB}(\zeta) \\
&\quad + [u_{j+1/2, com} - u_j(1)] g_{RB}(\zeta).
\end{align*}
\]

(3.23)

The derivative is

\[
\begin{align*}
\left( u_j^C \right)_\zeta &= \left( u_j \right)_\zeta + [u_{j-1/2, com} - u_j(1)] g_{LB,1}(\zeta) \\
&\quad + [u_{j+1/2, com} - u_j(1)] g_{RB,1}(\zeta).
\end{align*}
\]

(3.24)

And the derivative \( \left( u_j^C \right)_x \) follows. See Fig. 3.1(a).

3.6. Common derivative and corrected second derivative estimates

At each interface, in formula (3.22) for the common value, with \( 0 \leq \kappa \leq 1 \), the weight for \( u_t \) is \( \kappa \) and that for \( u_0 \) is \( 1 - \kappa \). To define the common derivative value, we switch the two weights. Loosely put, this switch makes the method unbiased. If we apply the weighted average to \( \{ u_j^C \} \), the resulting \( \{ u_{j,1/2, com} \} \) has a stencil of four cells, from \( j - 1 \) to \( j + 2 \) (see Fig. 3.1(a)). Since the calculation of \( u_t \) in cell \( j \) employs \( \{ u_{j-1/2, com} \} \) and \( \{ u_{j+1/2, com} \} \), the corresponding scheme has a five-cell stencil.

We now define a common derivative at \( j + 1/2 \) that involves only the data in the two adjacent cells. A scheme with such a compact stencil is desirable since it is easy to code, the boundary conditions involved are simple, and the resulting implicit version has a sparse and generally invertible matrix. To this end, correcting for the right boundary of cell \( j \), set

\[
\begin{align*}
u_{j+1/2}^{LB}(\zeta) &= u_j(\zeta) + [u_{j+1/2, com} - u_j(1)] g_{RB}(\zeta)
\end{align*}
\]

(3.25)

i.e., \( g_{RB} \) corrects for the right boundary, namely \( u_{j+1/2, com} \), while leaving the value at the left boundary unchanged, namely \( u_j(1) \). Next, correcting for the left boundary of cell \( j + 1 \), set

\[
\begin{align*}
u_{j+1/2}^{LB}(\zeta) &= u_{j+1}(\zeta) + [u_{j+1/2, com} - u_{j+1}(1)] g_{LB}(\zeta)
\end{align*}
\]

(3.26)

Then \( g_{LB} \) corrects for the left boundary, namely \( u_{j+1}(1) \), while leaving the value at the right boundary unchanged, namely \( u_j(1) \).

Finally, for the common derivative at \( j + 1/2 \), set

\[
\begin{align*}
u_{j+1/2, com} &= (1 - \kappa) \frac{2}{h_j} \left\{ (u_j)_1(1) + [u_{j+1/2, com} - u_j(1)] g_{RB}(1) \right\} \\
&\quad + \kappa \frac{2}{h_{j+1}} \left\{ (u_{j+1})(1) + [u_{j+1/2, com} - u_{j+1}(1)] g_{LB}(1) \right\}.
\end{align*}
\]

(3.27)

See Fig. 3.1(b). Note the dependence only on \( u_{j+1/2, com} \) and the data on \( E_j \) and \( E_{j+1} \).

With the corrected derivative given by (3.24) and the common derivative above, we can obtain the corrected second derivative estimates.

The above procedure yields the CPR versions of the BR2 scheme if \( \kappa = 1/2 \) [5] and the LDG [12] or CDG schemes [44] if \( \kappa = 0 \) or \( \kappa = 1 \). Comparing the centered versus one-sided common values with \( s_{loc} \) as correction function (i.e., BR2 versus CDG), the former is of order \( 2K - 1 \); the latter, order \( 2K \); the former, however, has the advantage that its CFL limit is more than two times larger than the latter via Fourier analysis [24]. Since super-convergence (or super-accuracy) does not hold for the general case of nonlinear equations, the scheme using centered common values appear to have a slight edge. For additional comparisons, see [44].

4. Two/three-dimensional extension

The extension of the CPR formulation to quadrilateral and hexahedral elements is straightforward. The basic idea is to first transform the governing equations from a physical element to the reference or standard element. Then, the 1D CPR formulation is applied on the standard element in each coordinate direction.

Consider the 2D conservation law

\[
\begin{align*}
u_t + \phi_x + \gamma_y &= 0.
\end{align*}
\]

(4.1)

Denote \( \bar{r} = (x, y) \) the coordinates of the physical domain, and \( \tilde{r} = (\xi, \eta) \) the coordinates of the standard element. Let the transformation be written as

\[
\begin{align*}
\bar{r} &= \sum_j \tilde{M}_j \tilde{r}_j.
\end{align*}
\]

(4.2)
where \( \vec{r} \) are the physical coordinates used to define an element, and \( M(\vec{\zeta}) \) is the shape function. The transformed equation takes the following form

\[
\vec{u}_t + J \frac{\partial \vec{u}}{\partial \vec{x}} + \vec{g}_n = 0
\]  

(4.3)

where \( \vec{u} = |\vec{u}| \vec{u} \), \( \vec{f} = |\vec{f}| (\vec{\zeta}_j \vec{f} + \vec{\zeta}_g \vec{g}) \), \( \vec{g} = |\vec{g}| (\eta_j \vec{f} + \eta_g \vec{g}) \) and \( J \) is the Jacobian matrix of the transformation, i.e.,

\[
J = \frac{\partial \vec{r}}{\partial \vec{x}} = \begin{bmatrix} x_\zeta & x_\eta \\ y_\zeta & y_\eta \end{bmatrix}
\]

For a quadrilateral element of index \( j \) (not related to \( j \) of (4.2)), two indices \((k, m)\) are used to denote the solution point, and \( u_{j,k,m} \) denotes the degrees of freedom (DOFs). The CPR formulation is then

\[
\frac{\partial u_{j,k,m}}{\partial t} + \frac{\partial f_{j,k,m}}{\partial x} + \frac{\partial g_{j,k,m}}{\partial y} + \left( \int_{\Omega_{k,m}} (\eta_{j,k,m} - 1) \, d\Omega \right) \frac{\partial u_{j,k,m}}{\partial x} + \left( \int_{\Omega_{k,m}} (\eta_{j,k,m} - 1) \, d\Omega \right) \frac{\partial u_{j,k,m}}{\partial y} = 0,
\]  

(4.4)

where the constants \( g_{j,k} \) and \( g_{j,m} \) are the derivatives of the correction functions, and are also called correction coefficients.

The extension to simplex and other types of elements is not as straightforward since the correction functions are not readily available. The first extension to triangular elements was based on the so-called lifting collocation penalty approach (LCP) [56]. As it turns out, the final form is very similar to (4.4). The details of the derivation are omitted and we summarize the basic formulation here. Define two sets of points, solution points and flux points as shown in Fig. 4.1. The CPR formulation can be rewritten as

\[
\frac{\partial u_{j,k}}{\partial t} + \frac{\partial f_{j,k}}{\partial x} + \frac{\partial g_{j,k}}{\partial y} + \frac{1}{|\Omega|} \sum_{f \in \Omega} \sum_{|f|} 2 \vec{e}_{f,j} [f^m]_{f,j} S_f = 0,
\]  

(4.5)

where \(|\Omega|\) is the area of the triangle, and \( S_f \) is the length of side \( f \), and \([f^m] \) is the normal flux difference between the common Riemann flux and the internal flux. The extension to 3D elements follows a similar path, and can be found in [19].

5. Stability of Flux Reconstruction methods

5.1. Energy stability for linear advection problems in 1D

Jameson [28] recently proved that a particular SD scheme (recovered via FR) is energy-stable for linear advection problems in 1D. Vincent et al. [51] subsequently extended this result, and identified a family of stable FR schemes for linear advection problems in 1D (for all orders of accuracy). Specifically, it was proven that if the left and right flux correction functions are defined as follows

\[
g_{j,k} = \frac{(-1)^k}{2} \left[ P_k - \left( \frac{\eta_k P_{k+1} + P_{k-1}}{1 + \eta_k} \right) \right],
\]  

(5.1)

\[
g_{j,k} = \frac{1}{2} \left[ P_k + \left( \frac{\eta_k P_{k+1} + P_{k-1}}{1 + \eta_k} \right) \right],
\]  

(5.2)

where \( k = K - 1 \) is the degree of the solution polynomial within each element, \( P_{k-1}, P_k \), and \( P_{k+1} \) are Legendre polynomials of the denoted degree (normalized such that \( |P_k(\pm1)| = 1 \) for all \( k \)), and \( \eta_k = c(2k + 1)(a_k k)^2 \).

With

\[
a_k = \frac{(2k)!}{2^{2k}(k!)^2},
\]

\[
c < \infty,
\]

then a broken Sobolev type norm of the approximate solution is guaranteed to be non-increasing, and thus bounded. Consequently, by equivalence of norms in the finite-dimensional solution space, any norm of the solution is guaranteed to remain bounded, and thus the method is guaranteed to be stable.

The resulting one parameter family of FR schemes, defined in terms of the free parameter \( c \), have been referred to as Energy Stable Flux Reconstruction (ESFR) schemes or Vincent–Castonguay–Jameson–Huynh (VCJH) schemes. It can be noted that judicious choice of the parameter \( c \) leads to recovery of various known FR schemes. Specifically, if \( c = 0 \), then a nodal DG scheme is recovered, if

\[
c = \frac{2k}{(2k + 1)(k + 1)(a_k k)^2},
\]

then a particular SD scheme is recovered (the scheme is, in fact, the particular SD scheme that Huynh showed to be Fourier stable, and Fourier stable, and Jameson [28] proved to be energy stable), and if

\[
c = \frac{2(k + 1)}{(2k + 1)(a_k k)^2},
\]

then the \( g \) FR scheme is recovered.

5.2. Energy stability for linear advection problems on 2D triangular grids

VCJH schemes for linear advection problems in 1D have been extended by Castonguay et al. [9] to treat linear advection problems on 2D triangular grids. As in the 1D case, a one-parameter family of correction functions were identified that guarantee a particular norm of the solution is non-increasing. However, unlike in the 1D case, an explicit correction for these correction functions was not presented (instead the divergence of each correction function was defined implicitly via a matrix system). Interestingly, the one-parameter family of schemes did not appear to include a SD scheme as a special case—despite the fact that Balan, May, and Schöberl [2] were able to identify stable SD schemes on triangular grids for several orders or accuracy.

5.3. Energy stability for linear advection–diffusion problems

Recently, Williams et al. [61], Castonguay et al. [10], Williams et al. [62], Williams [60], and Williams and Jameson [63] extended VCJH schemes for linear advection problems to develop a range of
VCJH schemes for linear advection–diffusion problems on grids of one, two and three dimensional simplex elements. Their approach involves use of VCJH correction functions to construct a $C^0$ continuous polynomial representation of the solution, in addition to a representation of the flux that has $C^0$ continuous interface normal components. Development of an energy-stable treatment for diffusive terms is important, since it is a prerequisite for effective solution of the Navier–Stokes equations.

5.4. Non-linear stability

Jameson et al. [29] showed that FR methods can be afflicted by an aliasing driven instability if the flux function is non-linear. Such instabilities are a consequence of aliasing errors (that occur when a polynomial representation of the non-linear flux is constructed via a collocation projection at the solution points). Jameson et al. [29] also demonstrated that the location of the solution points plays a critical role in determining the extent of any aliasing driven instabilities. Specifically, they suggest that the solution points should be located at the abscissa of a strong quadrature rule in order to minimize aliasing driven instabilities. This finding is supported by the numerical experiments of 8, who used the FR approach to solve the Euler equations on 2D triangular grids. They found that if the solution points were located at the so-called alpha-optimized points of Castonguay et al. [21] then the simulations blew up. However, if the solution points were located at the abscissa of a high-strength quadrature rule derived by Hesthaven and Warburton [49], then the simulations remained stable (see Fig. 5.1).

5.5. Von Neumann analysis

Energy-based stability proofs are powerful since they apply for all orders of accuracy and on non-uniform grids. However, they do not offer insight into all the stability properties of a numerical scheme. Huynh [23] and Vincent, Castonguay and Jameson [52] presented comprehensive von Neumann analyses of FR methods in order to elucidate further stability properties of the schemes. Their results indicate that the form of the flux correction function has a significant impact on the CFL stability limit associated with a given FR scheme. In the context of 1D VCJH schemes for linear advection, it has been shown that increasing the free parameter $c$ (from zero) can increase the CFL limit by over a factor of two in certain cases (at the cost of a reduction in the overall accuracy of the scheme).

5.6. Remaining stability problems

The above developments are significant in terms of understanding fundamental stability properties of FR schemes. However, there remain various stability issues that need addressing. Firstly, whilst empirical evidence suggests that the 1D stability proofs of Vincent et al. [51] and Castonguay et al. [10] extend to tensor product elements (quadrilaterals and hexahedra), there exists no mathematical proof, and it remains an open question as to whether multi-dimensional tensor product formulations based on 1D VCJH schemes are in fact linearly stable. Moreover, as yet VCJH schemes for advection–diffusion problems have not been extended to prismatic and pyramid shaped elements, both of which are widely used to create unstructured meshes of complex 3D geometries. Finally, robust strategies for reducing/controlling aliasing driven instabilities in multiple element types need to be developed.

6. Representative numerical examples

In this section, two numerical examples are shown to demonstrate the capability of the CPR formulation.

6.1. Direct numerical simulation of the Taylor–Green vortex at $Re = 1600$

This is a benchmark case (C3.5) of the 1st International Workshop on High-Order CFD Methods (Wang et al. 2013), and was designed to evaluate numerical methods in accurately capturing the evolution of a smooth flow to a turbulent flow. A very high-resolution simulation with a spectral method is used as the “analytical solution”. In the present simulations, p2, p3 and p4 CPR schemes were used with different mesh sizes ranging from $64^3$ to $96^3$. Table 6.1 summarizes all the cases with the time steps used in the simulations. The 3rd-order SSP (strong-stability preserving) Runge–Kutta scheme was used for time integration. The computed energy dissipation rate and enstrophy history are displayed in Fig. 6.1. The obvious trend is that the higher-order schemes perform better than the lower order schemes. For example, the p3 scheme with $64^3$ cells performs better than the p2 scheme with $96^3$ elements, and is less expensive.

6.2. Computations of bio-inspired vortex-dominated flows

This case was performed by Yu et al. [64], and re-produced here with permission. Flows over a rectangular flapping wing, as shown

![Fig. 5.1](image-url)
in Fig. 6.2, are studied here. The wing can undergo a flapping or combined flapping-pitching motion. A remeshing technique is then used to deform the mesh at each time step.

In this study, the Strouhal number ($St$) of the finite-span flapping wing was selected to be well within the optimal range usually used by flying insects, birds, and fish (i.e., $0.2 < St < 0.4$). The Mach number of the free stream is set to be 0.05 to mimic incompressible flow. The Reynolds number ($Re$) based on the free stream velocity and the maximum chord length is 1200. The reduced frequency of the flapping motion is 3.5, and the Strouhal number of the wingtip is 0.38. The space discretization accuracy for the simulation is of third order, and the time integration is performed with the explicit third order TVD Runge–Kutta method.

The flapping alone and the combined flapping-pitching were studied. The computed iso-surfaces of the Q-criterion at different phase angles for both motions are shown in Fig. 6.3. The histories of the thrust coefficient from both motions are presented in Fig. 6.4. Note that the thrust from the combined motion is more than an order of magnitude higher than that from flapping alone. In fact, the averaged thrust from the combined motion is 27 times as large as that from the flapping motion.

### 7. Additional recent research on FR/CPR methods

In this section, we briefly discuss additional recent contributions to the development of FR/CPR.

Suppressing oscillations near shocks via localized artificial diffusivity or LAD was carried out by Miyaji [43] and Haga et al. [20]. Initial results are encouraging; however, dealing with shocks requires further research and tests, especially for the case of multiple dimensions.

Concerning SD-type schemes, extensions of a particularly simple FR/CPR scheme called $g_2$ to the Navier–Stokes equations on moving and deforming domains were presented by Liang et al. [35,38]. They demonstrated that the $g_2$ scheme is (up to 40%) faster and easier to implement than the SD method. Comparison of computational efficiencies for various types of schemes versus CPR was carried out by Yu and Wang [65]. Their results show that CPR schemes are the most efficient.

An interface element approach dealing with non-conforming polynomials together with p-adaptation for viscous compressible flow simulations was elaborated by Cagnone and Nadarajah [6] and Cagnone et al. [7]. In addition, an implicit large eddy simulation (ILES) solver was developed for CPR schemes using a third-order singly diagonal implicit Runge–Kutta scheme by Vermeire et al. [50]. Preliminary work on FR methods for body-fitted Cartesian unstructured grids was presented in Haga et al. [20]. A PnPm-CPR method for the Navier–Stokes equations was studied by Shi et al. [47]. Here, among the main findings in applying PnPm to CPR was that the accuracy gain in multiple dimensions with a compact reconstruction stencil is rather limited.

It is well-known that hp-adaptation is beneficial for high-order methods [55]. Adjoint-based error estimation and hp-adaptation for CPR was studied by Shi and Wang [46]. Their results show significant savings compared to the uniform h or p refinement.

Progress toward applying FR/CPR methods to solve practical internal flow problems has been made by Lu et al. [41,40], and Lu and Dawes [42]. These authors have devised a parallel solver on hybrid unstructured meshes including tetrahedra, prisms, pyramids and hexahedra for turbulent subsonic/transonic flows.

### 8. Pacing items

Pacing items are similar to those mentioned by the committee for the International Workshop on High-Order CFD Methods [55]. Some of the key items are discussed below.
8.1. High-order mesh generation

In order to achieve high-order accuracy, curved geometries need to be represented with high-order polynomials. The generation of unstructured, highly-clustered viscous meshes near high-order boundaries requires further research to improve robustness. The main difficulty is that cells near the curved geometries can overlap each other.

8.2. Capturing shocks

The two main approaches are local artificial dissipation (LAD) and limiting. The former involves user specified parameters, and latter often causes convergence to stall. A third approach is via h-p mesh adaptation where, near shocks, the mesh is refined and the method switches to the first-order upwind scheme; here, deciding when to switch is not trivial. An optimal method should capture shocks with high resolution, preserve accuracy at smooth regions of the flow, and be convergent when needed.

8.3. Time stepping

Much research is needed on how to handle the stiffness generated by highly-anisotropic meshes near walls for viscous flows. Low storage and efficient iterative solution methods for both steady and unsteady flow problems are active area of research. Time stepping methods are derived typically by mathematics where stability and accuracy are the main focus. Can time-stepping methods be derived in combination with physics to deal with the stiffness issue?

8.4. Mesh adaptation

There are numerous works on this topic, however, a simple, efficient, and robust 3D adaptation method requires further research.

9. Conclusions and discussion

In conclusion, we presented a brief review of recent developments for the FR/CPR schemes. Basic description and stability of the approach were discussed. Representative numerical examples were shown. Some key pacing items were mentioned. The FR/CPR framework appears to be promising and capable of contributing toward the goal of faster and higher fidelity CFD capabilities for more accurate flow field predictions.

Acknowledgments

The first author was supported by the Aeronautical Sciences Project of NASA. He wishes to thank Drs. James DeBonis and Seth Spiegel for their thorough reviews and numerous valuable suggestions. The second author has been supported by AFOSR under Grant FA95501210286 and NASA under Grant NNX12AK04A.