A review of flux reconstruction or correction procedure via reconstruction method for the Navier-Stokes equations

Z.J. WANG* and H.T. HUYNH**

*University of Kansas
Department of Aerospace Engineering, Lawrence, KS 66045, U.S.A.
E-mail: zjw@ku.edu
** NASA Glenn Research Center,
Cleveland, OH 44135, U.S.A.

Received 31 August 2015

Abstract
Computational Fluid Dynamics (CFD) methods that are third-order accurate or higher are generally considered high-order methods in the aerospace community. These methods have the potential of yielding accurate CFD solutions in a more efficient manner than first or second-order ones. High-order methods include the popular discontinuous Galerkin (DG) as well as the spectral volume (SV), spectral difference (SD), and PnPm or hybrid DG/finite volume methods. Recently, the flux reconstruction (FR) or correction procedure via reconstruction (CPR) method provides a unifying framework for existing methods including the above mentioned DG, SV, SD as well as new ones with various accuracy and stability properties. This paper provides a review of recent FR/CPR developments together with sample applications and pacing items.

Key words: High-order methods, Unstructured meshes, Conservation laws, Flux reconstruction, Correction procedure via reconstruction, Navier-Stokes equations

1. Introduction
Computational fluid dynamics (CFD) has, in the past two decades, become an indispensable design tool in aerospace, mechanical, and many other engineering disciplines. Production CFD codes typically solve the Reynolds averaged Navier-Stokes equations using a first or second-order numerical method together with a turbulence model. These codes are capable of providing simulations with engineering accuracy (~5% solution error) for a variety of flows. However, for vortex dominated problems such as flows with large separations, existing production CFD tools often result in poor predictions. Such problems call for high-order accurate numerical methods and large eddy simulations (LES).

High-order accurate methods, which include the discontinuous Galerkin (Reed & Hill, 1973, Cockburn & Shu, 1998, Bassi & Rebay, 1997a, 1997b, Cockburn, Karniadakis, and Shu, 2000), spectral difference (Liu, Vinokur, and Wang, 2006, Kopriwa and Kollas, 1996), spectral volume (Wang, Zhang, and Liu, 2004), and PnPm or hybrid DG/finite volume (Dumbser et al, 2008 and Zhang et al, 2012) methods, have been developed for many decades on both structured and unstructured meshes. Reviews on this subject can be found in, e.g., (Cockburn, Karniadakis, and Shu, 2000, Ekaterinaris 2005, and Wang, 2007). There has been a concerted effort by experts worldwide to evaluate the performance of these methods through a series of workshops, namely, the International Workshops on High-Order CFD Methods (2012, 2013, 2015, and upcoming 2016). The history and motivation for these workshops were given in (Wang et al. 2013).

In this paper, we review recent developments of a high-order method derived via the differential formulation called flux reconstruction or correction procedure via reconstruction (FR/CPR). The method, introduced by the second author in (Huynh, 2007), provides a framework, which unifies popular high-order methods such as the discontinuous Galerkin (DG), spectral difference (SD), and spectral volume (SV). The FR/CPR method was reviewed in (Huynh, Wang and Vincent, 2014). Here, we focus on more recent progresses.

FR/CPR formulation for 2D and 3D Navier-Stokes equations has been completed for mixed elements. Successful 3D implementations include (Haga, Gao, and Wang, 2010, 2011, Lu, Dawes, and Yuan, 2012, Vermeire, Cagnone, and

This paper is organized as follows. The basic concept of the FR/CPR method for 1D conservation laws is presented in Section 2. Section 3 discusses extensions of the FR/CPR method to multiple dimensions. Section 4 describes several recent developments, and pacing items. Finally, concluding remarks are given in Section 5.

2. FR/CPR methods for 1D conservation laws

Consider the conservation law

\[ u_t + f_x = 0, \]  

with initial condition \( u(x,0) = u_0(x) \) 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 computational domain be divided into \( N \) (possibly non-uniform) cells or elements \( E_j, \ j = 1, 2, \ldots, N \). Denote the center of \( E_j \) by \( x_j \), its two end points by \( x_{j-1/2} \) and \( x_{j+1/2} \), and its width by \( h_j \). With \( \xi \) varying on \( I = [-1,1] \) and \( x \) on \( E_j \), the linear function mapping \( I \) onto \( E_j \) and its inverse are

\[ x(\xi) = x_j + \xi h_j / 2 \quad \text{and} \quad \xi(x) = 2(x - x_j) / h_j. \]

The conservation law is transformed from the physical domain to the standard element

\[ u_t + f_x \xi_x = u_t + \frac{2}{h_j} f_\xi = 0. \]  

The exact solution \( u \) is approximated by piecewise polynomials of degree \( K-1 \) with no continuity requirement at element interfaces. On element \( j \), the polynomial is denoted \( u_j(\xi) \). The main task in the FR/CPR method is to reconstruct a flux polynomial \( F_j(\xi) \), which is one degree higher, i.e., degree \( K \), so that the governing equation is solved like a finite difference scheme

\[ \frac{\partial u_j}{\partial t} + \frac{2}{h_j} \frac{\partial F_j}{\partial \xi} = 0. \]  

Although the numerical solution is discontinuous across element interfaces, the flux polynomial is required to be continuous so that (a) its derivative does not involve a Dirac delta function and (b) to ensure conservation. An efficient way to solve (2.3) is to use a nodal formulation using a set of solution points defined on \( I \), which are typically the Gauss or Lobatto points denoted by \( \xi_k, \ k = 1, \ldots, K \). Let the Lagrange basis functions for \( \xi_k \) be \( l_k(\xi), \ k = 1, \ldots, K \). Then the solution polynomial on element \( j \) can be written as

\[ u_j(\xi) = \sum_{k=1}^{K} u_{j,k} l_k(\xi) \]  

where \( u_{j,k} \) is the solution at solution point \( k \). Eq. (2.3) can be replaced by the following nodal formulation

\[ \frac{\partial u_j}{\partial t} + \frac{2}{h_j} \frac{\partial F_j}{\partial \xi} = 0. \]
\[
\frac{du_{j,k}}{dt} + \frac{2}{k_j} \frac{\partial F_j(\xi)}{\partial \xi} = 0. \tag{2.5}
\]

We march in time by, say, a Runge-Kutta method (Shu, 1988). With \( f_{j,k} = f(u_{j,k}) \), let \( f_j(\xi) \) be the polynomial of degree \( K - 1 \) interpolating \( f_{j,k}, k = 1, \ldots, K \),

\[
f_j(\xi) = \sum_{k=1}^{K} f_{j,k} l_k(\xi). \tag{2.6}
\]

The flux polynomials \( \{f_j\} \) form a function, which is generally discontinuous across cell interfaces and \( f_j \) is called the discontinuous flux function. The derivative of a discontinuous function involves the Dirac delta function. To avoid this problem and also to achieve conservation, we construct a continuous flux function \( F_j(\xi) \), which approximates the discontinuous function in some sense. At each interface, the unique flux common to the two adjacent cells is taken to be the Riemann flux found in a finite volume method, i.e.,

\[
f_{j+1/2,\text{com}} = f_{\text{Riem}}(u_{i+1/2,L}, u_{i+1/2,R}),
\]

where \( u_{i+1/2,L}, u_{i+1/2,R} \) are the solutions just to the left and right of the interface \( x_{i+1/2} \).

\[
u_{j+1/2,L} = u_j(1) \quad \text{and} \quad u_{j+1/2,R} = u_{j+1}(-1). \tag{2.7}
\]

The continuous flux function is obtained by adding a correction to the discontinuous one

\[
F_j(\xi) = f_j(\xi) + [f_{j-1/2,\text{com}} - f_j(-1)] g_{LB}(\xi) + [f_{j+1/2,\text{com}} - f_j(1)] g_{RB}(\xi) \tag{2.8}
\]

where \( g_{LB}(\xi), g_{RB}(\xi) \) are degree \( K \) correction polynomials satisfying

\[
g_{LB}(-1) = 1, \quad g_{LB}(1) = 0, \tag{2.9}
\]

\[
g_{RB}(-1) = 0, \quad g_{RB}(1) = 1. \tag{2.10}
\]

Note that the purpose of the correction polynomials is to ensure that the continuous flux polynomial takes on the values of the Riemann flux at the two end points, i.e.,

\[
F_j(-1) - f_j(-1) = f_{j-1/2,\text{com}} - f_j(-1) \quad \text{and} \quad F_j(1) - f_j(1) = f_{j+1/2,\text{com}} - f_j(1). \tag{2.11}
\]

We would like the continuous flux polynomial to approximate the discontinuous one. In other words, we like the correction polynomials to approximate zero.

Next, the derivative of \( F_j \) at the solution point \( \xi_k \) is

\[
(F_j)_{j,k} = \left( f_j \right)_{j,k} + \left[ f_{j-1/2,\text{com}} - f_j(-1) \right] g_{LB}'(\xi_k) + \left[ f_{j+1/2,\text{com}} - f_j(1) \right] g_{RB}'(\xi_k). \tag{2.12}
\]

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

\[
(f_j)_{j,k} = \sum_{i=1}^{K} f_{j,i} \frac{dl_i(\xi_k)}{d\xi}. \tag{2.13}
\]

Instead of differentiating \( f_j \) as above, we can also use the chain rule

\[
(f_j)_{j,k} = \frac{\partial f}{\partial u}(u_{j,k})(u_{\xi})_{j,k}. \tag{2.14}
\]
Wang and Gao (2009) found that for the Euler equations, the chain rule yields more accurate solutions. The final equation then becomes

$$\frac{du_{j,k}}{dt} + \frac{2}{\Delta t} \left[ \left( f_{j} \right)_{j,k} + \left[ f_{j} \right]_{L} \alpha_{L,k} + \left[ f_{j} \right]_{R} \alpha_{R,k} \right] = 0,$$

(2.15)

where $\left[ f_{j} \right]_{L} = [f_{j-1/2, \text{com}} - f_{j}(-1)]$ and $\left[ f_{j} \right]_{R} = [f_{j+1/2, \text{com}} - f_{j}(1)]$ are the flux jumps at the left and right interfaces, respectively, and $\alpha_{L,k} = g_{LB}'(\xi_k)$ and $\alpha_{R,k} = g_{RB}'(\xi_k)$ are constants.

We summarize the FR/CPR algorithm below.

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 $\left( f_{j} \right)_{j,k}$ of the discontinuous flux function by (2.13). Alternatively, the chain rule (2.14) can be employed.

3. At the two interfaces of $E_j$, get the corrections $f_{j-1/2, \text{com}} - f_{j}(-1)$ and $f_{j+1/2, \text{com}} - f_{j}(1)$. At the solution points, evaluate $\left( F_{j} \right)_{j,k}$ by (2.12).

4. March in time by, say, a Runge-Kutta method.

This completes the algorithm.

Various corrections functions are included in (Huynh, 2007). Some new schemes are discovered by the Jameson group at Stanford, e.g., (Vincent, Castonguay, Jameson, 2011a and 2011b, and Vincent et al., 2015).

The Diffusion Equation. On $(-\infty, \infty)$, consider the diffusion equation,

$$u_t = u_{xx},$$

(2.16)

with initial condition

$$u(x,0) = u_0(x)$$

(2.17)

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.

The first task is to estimate $u_x$ at the solution points $x_{j,k}$. Since the function $\{u_j\}$ is discontinuous across the interfaces, to estimate $u_x$, we first reconstruct a piecewise polynomial function $\{u^C_j\}$ of degree $K$, which is continuous across the cell interfaces, and approximates $u_j$ (the superscript ‘C’ stands for ‘continuous’ or ‘corrected’). The derivative approximation $\left( u^C_j \right)_{x}(x_{j,k})$ accounts for the data interaction (whereas $\left( u_j \right)_{x}$ does not).

In order for $\{u^C_j\}$ to be continuous at the interfaces, $u^C_j$ and $u^C_{j+1}$ 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 $u_L$ and $u_R$ given by (2.7),

$$u_{\text{com}} = u_{j+1/2, \text{com}} = (u_L + u_R)/2.$$ 

(2.18)
The above formula was employed by Bassi and Rebay (1997a, 2000). A more general formula is the weighted average, with $0 \leq \kappa \leq 1$,

$$u_{\text{com}} = u_{j+1/2, \text{com}} = \kappa u_L + (1 - \kappa)u_R.$$  \hfill (2.19)

For $\kappa = 0$ or $\kappa = 1$, we have the one-sided formula used in the local DG or LDG (Cockburn and Shu 1998) as well as the compact DG or CDG methods (Peraire and Persson 2008).

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

$$u_j^c(\xi) = u_j(\xi) + [u_{j+1/2, \text{com}} - u_j(-1)] g_{\text{LB}}(\xi) + [u_{j+1/2, \text{com}} - u_j(1)] g_{\text{RB}}(\xi).$$  \hfill (2.20)

The derivative is

$$\left( u_j^c \right)_x(\xi) = \left( u_j \right)_x(\xi) + [u_{j-1/2, \text{com}} - u_j(-1)] g_{\text{LB}}'(\xi) + [u_{j+1/2, \text{com}} - u_j(1)] g_{\text{RB}}'(\xi).$$  \hfill (2.21)

And the derivative $\left( u_j^c \right)_x$ follows. See Fig. 1(a).

![Fig. 1. Centered-type common derivative: (a) using a four-cell stencil and (b) using a two-cell stencil via (2.24). Here, the solution polynomials are linear, and the correction function $g_{DG}$ is quadratic.](image)

At each interface, in formula (2.19) for the common value, with $0 \leq \kappa \leq 1$, the weight for $u_L$ is $\kappa$ and that for $u_R$ 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 $\left( u_j^c \right)_x$, the resulting $(u_x)_{j+1/2, \text{com}}$ has a stencil of four cells, from $j - 1$ to $j + 2$ (see Fig. 1(a)). Since the calculation of $u_{xx}$ in cell $j$ employs $(u_x)_{j-1/2, \text{com}}$ and $(u_x)_{j+1/2, \text{com}}$, 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. To this end, correcting for the right boundary of cell $j$, set

$$u_j^{\text{RB}}(\xi) = u_j(\xi) + [u_{j+1/2, \text{com}} - u_j(1)] g_{\text{RB}}(\xi)$$  \hfill (2.22)

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

Then \( u_{j+1}^{LB} \) corrects for the left boundary, \( u_{j+1}^{LB}(-1) = u_{j+1/2,\text{com}} \), 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_{x})_{j+1/2,\text{com}} &= (1 - \kappa) \frac{2}{h_{j+1}} \left\{ (u_{j})_{\xi}(1) + \left[ u_{j+1/2,\text{com}} - u_{j}(1) \right] g_{RB}'(1) \right\} \\
&+ \kappa \frac{2}{h_{j+1}} \left\{ (u_{j+1})_{\xi}(-1) + \left[ u_{j+1/2,\text{com}} - u_{j+1}(-1) \right] g_{LB}'(-1) \right\}.
\end{align*}

(2.24)

See Fig. 1(b). Note the dependence only on data on \( E_{j} \) and \( E_{j+1} \).

With the corrected derivative given by (2.24) and the common derivative above, we can obtain the corrected second derivative estimates.

The above procedure yields the FR/CPR versions of the BR2 scheme if \( \kappa = 1/2 \) (Bassi and Rebay, 2000) and the LDG (Cockburn and Shu 1998) or CDG schemes (Peraire and Persson 2008) if \( \kappa = 0 \) or \( \kappa = 1 \).

3. Extension to multiple dimensions

The extension of the FR/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 FR/CPR formulation is applied on the standard element in each coordinate direction.

Consider the 2D conservation law

\[ u_{t} + f_{x} + g_{y} \equiv u_{t} + \nabla \cdot \vec{f} = 0 \]  

(3.1)

Denote \( \vec{r} = (x, y) \) the coordinates of the physical domain, and \( \vec{\xi} = (\xi, \eta) \) the coordinates of the standard element and \( \vec{f} = (f, g) \). To achieve an efficient implementation, elements in the physical domain are transferred into the standard element. For example, the standard triangle is

\[ T = \{ \vec{\xi} = (\xi, \eta) | (\xi, \eta) \geq 0; \xi + \eta \leq 1 \}, \]

Figure 2. Transformation of physical elements to standard elements

\[ \begin{array}{c}
(0, 1) \\
(1, 0) \\
(1, 1) \\
(-1, -1)
\end{array} \]

\[ \begin{array}{c}
\vec{\xi} \quad \vec{\eta} \\
\vec{\xi} \quad \vec{\eta} \\
\vec{\xi} \quad \vec{\eta}
\end{array} \]

\[ \begin{array}{c}
f = 1 \\
f = 2 \\
f = 3
\end{array} \]
and the standard quadrilateral is

\[ Q = \left\{ (\xi, \eta) \mid -1 \leq (\xi, \eta) \leq 1 \right\} . \]

Let the transformation be written as

\[ \vec{r} = \sum_i M_i(\vec{\xi}) \vec{r}_i , \]

(3.2)

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

\[ \frac{\partial \vec{u}}{\partial t} + \frac{\partial \vec{f}}{\partial \xi} + \frac{\partial \vec{g}}{\partial \eta} = \vec{0} , \]

(3.3)

where \( \vec{u} = J u \cdot \vec{r} , \quad \vec{f} = J f(\xi, f + \eta, g) , \quad \vec{g} = J(\eta, f + \eta, g) \), and \( J \) is the Jacobian matrix of the transformation, i.e.,

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

For a quadrilateral element of index \( j \) two indices \((k,m)\) are used to denote the solution point, and \( \vec{u}_{j,k,m} \) denotes the degrees of freedom (DOFs). The CPR formulation is then

\[ \frac{\partial \vec{u}_{j,k,m}}{\partial t} + \frac{\partial \vec{f}_{j,k,m}}{\partial \xi} + \frac{\partial \vec{g}_{j,k,m}}{\partial \eta} = 0 , \]

(3.4)

\[ + \frac{\vec{f}_{\text{com}}(-1,\eta_{k,m}) - \vec{f}_j(-1,\eta_{k,m})}{2} \frac{\alpha_{k,j}}{2} + \frac{\vec{f}_{\text{com}}(1,\eta_{k,m}) - \vec{f}_j(1,\eta_{k,m})}{2} \frac{\alpha_{k,j}}{2} \]

\[ + \frac{\vec{g}_{\text{com}}(\xi_{k,m},-1) - \vec{g}_j(\xi_{k,m},-1)}{2} \frac{\alpha_{k,j}}{2} + \frac{\vec{g}_{\text{com}}(\xi_{k,m},1) - \vec{g}_j(\xi_{k,m},1)}{2} \frac{\alpha_{k,j}}{2} = 0 , \]

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) (Wang & Gao, 2009). As it turns out, the final form is very similar to (3.4). 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 Figure 3. The CPR formulation for a linear triangle can be rewritten as

\[ \frac{\partial u_{j,k,m}}{\partial t} + \frac{\partial f(u)}{\partial x_{j,k}} + \frac{\partial g(u)}{\partial y_{j,k}} + \frac{1}{V_f} \sum_{j \neq k} \sum_{f} \alpha_{k,j,f} [f^n]_{j,f} S_f = 0 . \]

(3.5)

Figure 3. Possible solution points (squares) and flux points (circles) for a degree 2 element
where \( |V_j| \) is the area of the triangle, and \( S_f \) is the length of side \( f \), and \( \left[ f^* \right] \) is the normal flux difference between the common Riemann flux and the internal normal flux. For a high-order triangular element, we use the transformed equation on the standard triangle, i.e.,

\[
\frac{\partial \tilde{u}_{j,k}}{\partial t} + \frac{\partial \tilde{\nu}(\tilde{u}_j)}{\partial \xi_{j,k}} + \frac{\partial \tilde{g}(\tilde{u}_j)}{\partial \eta_{j,k}} + \frac{1}{|V_j|} \sum_{f \in \gamma_{j,k}} \sum_i \alpha_{j,f,i} [\tilde{f}^*]_{j,j} S^j_f = 0, \tag{3.6}
\]

where \( \tilde{f}^* = (\tilde{f}, \tilde{g}) \cdot \tilde{n} \). Let \( \tilde{S}_\xi = |\tilde{V}\xi|, \tilde{S}_\eta = |\tilde{V}\eta| \), which physically represent the “area vector” of constant \( \xi \) and \( \eta \) lines in the physical domain, and obviously \( \tilde{f} = \tilde{f} \cdot \tilde{S}_\xi \) and \( \tilde{g} = \tilde{f} \cdot \tilde{S}_\eta \). For high-order elements, \( |J| \) is a high-order polynomial. There will be accuracy loss if we assume \( \tilde{u} \) is a polynomial in the standard element (Yu, Wang and Liu, 2014). In addition, Equations (3.4) and (3.6) may have difficulty preserving a constant freestream (Abe, et al 2015). Instead, we employ the following identity

\[
\frac{1}{|J|} \left( \frac{\partial \tilde{f}}{\partial \xi} + \frac{\partial \tilde{g}}{\partial \eta} \right) = \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y},
\]

to transform both back to the physical domain

\[
\frac{\partial u_{j,k,m}}{\partial t} + \frac{\partial f_{j,k,m}}{\partial x} + \frac{\partial g_{j,k,m}}{\partial y} + \frac{1}{|J_{j,k,m}|} \left\{ \right. \\
\left[ \tilde{f}_c(\xi_{j,0,m}) - \tilde{f}_c(-1,\eta_{j,k,m}) \frac{\alpha_{j,k,m}}{2} + \tilde{f}_c(1,\eta_{j,k,m}) - \tilde{f}_c(-1,\eta_{j,k,m}) \frac{\alpha_{j,k,m}}{2} \right. \\
\left. \left[ \tilde{g}_c(\xi_{j,0,m}) - \tilde{g}_c(-1,\eta_{j,k,m}) \frac{\alpha_{j,k,m}}{2} + \tilde{g}_c(1,\eta_{j,k,m}) - \tilde{g}_c(-1,\eta_{j,k,m}) \frac{\alpha_{j,k,m}}{2} \right] \right\} = 0, \tag{3.7}
\]

and

\[
\frac{\partial u_{j,k}}{\partial t} + \frac{\partial f_{j,k}}{\partial x} + \frac{\partial g_{j,k}}{\partial y} + \frac{1}{|J_{j,k}|} \sum_{f \in \gamma_{j,k}} \sum_i \alpha_{j,f,i} [f^*]_{j,j} S^j_f = 0, \tag{3.8}
\]

For the standard triangle, \( |V_j|=1/2 \). For face 1, \( n^j_1 = (0,-1), S^j_1 = 1 \), and

\[
[f^*]_{j,j} S^j_1 = -[\tilde{g}_c - \tilde{g}(\tilde{u}_j)]_{j,j} = (f^*_c - g^*(u_j))_{j,j} [\tilde{S}_{n1j}] = [f^*]_{j,j} [\tilde{S}_{n1j}], \tag{3.9}
\]

where superscript \( \xi \) means that the variables or operations are evaluated on the standard element. Similar formulas can be obtained for the other 2 faces. Finally, Eq. (3.8) can be further expressed as

\[
\frac{\partial u_{j,k}}{\partial t} + \frac{\partial f_{j,k}}{\partial x} + \frac{\partial g_{j,k}}{\partial y} + \frac{2}{|J_{j,k}|} \sum_{f \in \gamma_{j,k}} \sum_i \alpha_{j,f,i} [f^*]_{j,j} S^j_f = 0, \tag{3.10}
\]

where \( S_{1j} = [\tilde{S}_{n1j}], S_{2j} = [\tilde{S}_x + \tilde{S}_{n1j}], S_{3j} = [\tilde{S}_x]_{j,j} \). The extension to 3D elements follows a similar path, and can be found in (Haga, Gao, Wang 2011 and Wang, 2011).

4. Other developments and pacing items

In order for high-order methods including the FR/CPR to be routinely employed in engineering design, several
obstacles need to be overcome. These obstacles or pacing items have been identified and discussed at the International Workshops on High-Order CFD Methods 2012, 2013, and 2015 (which also provide an excellent venue for researchers and users to assess the status of high-order methods). Among the pacing items, we discuss high-order meshes, error estimates and hp-adaptations, and issues concerning large eddy simulations below.

4.1 High-order mesh generation

Meshes with straight edges or linear meshes work well for standard second-order methods. For high-order discontinuous Galerkin methods, however, it was first found by Bassi and Rebay (1997b) that at solid walls, linear meshes result in a significant loss of accuracy whereas high-order meshes (i.e., with curved edges in 2D described by polynomials) work well. Another reason high-order meshes are needed is that meshes for high-order methods are considerably coarser than those for second order ones; as a result, they require curved edges in 2D and curved surfaces in 3D for accurate description of solid walls. Currently, no commercial mesh generators are capable of generating high-order meshes. The reason is the limited market as well as the difficulties in generating high-order meshes. However, for relatively simple geometries, gmsh (Geuzaine and Remacle, 2009) is quite adequate.

High-order mesh generation poses two new challenges. First, it is more difficult to generate coarse meshes for a complex geometry as automated mesh generation algorithms can break down when generating coarse surface meshes at regions with high curvature. Second, generating highly clustered viscous meshes near a curved wall is daunting as interior mesh lines can cross the curved boundary, or intersect each other. Curved interior elements are necessary to remove such crossings.

Several approaches have been used to overcome some of these difficulties. Many groups generated fine multi-block structured meshes first; these fine meshes are then merged to produce high-order quadratic or quartic quadrilateral and hexahedral meshes (Wang, 2014). Although this approach has had some success, it is time consuming to generate structured meshes for complex geometries; therefore, the approach does not provide a long term solution. Another approach is to generate a relatively coarse linear mesh using a commercial mesh generator. Then the elements with curved wall boundary are made high order using curved edges and surfaces, as shown in Fig. 4. Finally, the interior elements are also curved based on solid mechanics by using radial basis functions to avoid grid lines crossing into each other, thus guaranteeing the positivity of the Jacobian of the geometric transformation. In case the geometry is not available, a reconstruction technique is used to rebuild high-order surfaces (Jiao & Wang, 2011) for solid walls before the interior surfaces and volume meshes are curved.

To facilitate communications between a high-order mesh generator and a high-order flow solver, we need a standard format to store hybrid meshes containing high-order elements as well as the usual types of elements including triangular, quadrilateral, tetrahedral, hexahedral, pyramidal and prismatic. Until recently, Gmsh was the only format capable of supporting high-order elements; therefore, it was employed in the 1st International Workshop on High-Order CFD Methods. Recently, the widely used CFD standard called CGNS was successfully extended to handle cubic and quartic elements. Mesh standards for high-order elements play a critical role since there are so many more variations in defining high-order elements.

![Linear mesh](image1) ![High-order mesh](image2)

Figure 4. Illustration of linear and high-order meshes

[DOI: 10.1299/mer.15-00475] © 2016 The Japan Society of Mechanical Engineers
To deal with curved boundaries, a high-order meshing tool (Ims, Duan, Wang, 2015) has been developed by the University of Kansas with funding from NASA and the Air Force Office of Scientific Research to convert linear meshes into high-order ones using a surface reconstruction algorithm. The input and output format is CGNS. The key steps in the conversion are:

- Read the linear mesh in CGNS format.
- Select patches to perform high-order reconstructions.
- Detect sharp edges, which are considered critical features.
- Reconstruct high order polynomial representations for the selected patches. High-order reconstructions are not performed across sharp edges.
- Curve volume elements where necessary to avoid negative Jacobians.
- Save the final mesh in CGNS format.

Figure 5 displays the GUI for the mesh conversion tool.

Figure 5. The graphic user interface for the linear to high-order mesh conversion tool

4.2 Error Estimate and hp-Adaptations

Mesh adaptation (or h-adaptation) has been demonstrated in academia and government research labs for low order methods as a very effective way to reduce simulation cost and improve solution accuracy, especially for unstructured meshes. With the development of high-order methods, a high-order solver can incorporate 1st to 6th or even higher order schemes in the same simulation. For a problem with smooth geometry and smooth solution, higher order (higher $p$) is often more effective. For problems with both smooth and non-smooth regions, an hp-adaptive approach is usually more optimal. Typically, one would use a first- or second-order scheme with fine elements near a shock, and a 4th or 6th order scheme with coarse elements in the smooth flow regions. The order-adaptation or $p$-adaptation adds another dimension into the simulation process. Performing $hp$-adaptations potentially results in higher payoffs than having $h$ or $p$-adaptation alone. The decision on where to perform $h$ or $p$-adaptation is not an easy one: in many cases, it is difficult to distinguish a smooth flow with large-gradient from a discontinuity.

There have been significant progress in the past decades on error estimates and $hp$-adaptations, and we refer interested readers to the recent review by Fidkowski and Darmofal (2011) for more details. Below, we provide a brief overview of adjoint-based error estimate and adaptation applied to the FR/CPR method.

Let $Q_h$ denote the numerical solution, $J_h(Q_h)$ the scalar engineering output of interest (such as the lift or drag coefficient). The output adjoint $\psi_h$ has the same dimension as $Q_h$, and is the sensitivity to an infinitesimal residual perturbation, which can be computed based on the discrete adjoint equation

$$\left( \frac{\partial R_h}{\partial Q_h} \right)^T \psi_h = \left( \frac{\partial J_h}{\partial Q_h} \right)^T = 0.$$  \hspace{1cm} (4.1)
The adjoint has become a very powerful tool in error estimate and adaptations. Its application to high-order methods appears to speed up CFD simulations significantly, sometimes by orders of magnitude.

The first obstacle is to develop an adjoint-consistent formulation for the nodal FR/CPR formulation. This task was overcome by Shi & Wang, in 2013a. Next, we present a demonstration case from (Shi and Wang, 2015a, 2015b). The problem is a laminar flow over a NACA0012 airfoil with freestream $M = 0.5$ and angle of attack of 1 degree, a benchmark case from the 1st International Workshop on High-Order CFD Methods. The Reynolds number based on the chord length of the airfoil is $Re = 5000$. The initial mesh is a coarse quartic mesh shown in Fig. 6, together with the Mach distribution. The drag and lift coefficients are the outputs of interest. Adjoint-based h-adaptations with a 4th order FR/CPR scheme are driven by the output-based error indicator. The adaptive results are compared with those of uniform refinements. Figure 7 displays the final meshes using lift and drag adjoints, and the Mach contours on the final adapted mesh. Note that regions near the stagnation streamlines and in the boundary layer were targeted for refinements. The trailing edge was also refined repeatedly to reduce the effect of the singularity. Figure 8 shows the CD error vs mesh size and non-dimensional total CPU time, with uniform and adaptive mesh refinements. With h-adaptation, a much higher effective convergence rate was achieved as shown in the figure. It was found that the adjoint based h-adaptation approach can reduce the computational cost in terms of the number of DOFs and work units by orders of magnitude.
4.3 Issues on large eddy simulations

It is widely accepted by the CFD community that high-order methods can significantly improve the accuracy of large eddy simulations (LES) of turbulent flows due to their much higher resolution than standard second order methods. On the other hand, it is also widely believed that pure LES for high Reynolds number flow problems (Re > 1.e6) is too expensive in the foreseeable future because the accurate computation of wall shear stresses requires a fine mesh resolution in all three directions. To illustrate this point, we take a sample calculation of turbulent channel flow from (Zhu, et al, 2015) at Reτ = 640 with a pure LES without any subgrid scale stress model. In case 1, the mesh resolution in the wall normal, spanwise and streamwise direction are y+ = 12, z+ = 30, and x+ = 60, whereas in case 2, the mesh resolution in the wall normal direction remains the same, and the number of points double in the other two directions resulting in z+ = 15, and x+ = 30. The computed average u’ - y’ profile is displayed in Fig. 9. Note that even if the normal mesh resolution remains the same, large errors in the velocity profile are produced when the mesh resolution in the wall-tangential directions is not sufficient, resulting in large errors in skin friction prediction.

Figure 9 also illustrates the need for wall modeling for high Reynolds number problems. Wall models are used to compute an accurate wall shear stress when the mesh resolution in the wall normal and tangential directions is not sufficient. Wall modeling not only reduces the number of mesh points in the wall normal direction, but also in the wall-tangential directions, thus making hybrid LES-RANS types of computations orders of magnitude faster than a pure LES approach.

Another important issue in LES is the subgrid scale (SGS) stress model, which closes the filtered Navier-Stokes equations. SGS models are used to compute the SGS stress based on the filtered solutions, and the earlistone, the static Smagorinsky model (Smagorinsky, 1963), is based on the idea of eddy viscosity. A significant improvement to the static model is the dynamic model developed by Germano et al in 1991. In (Li, Wang, 2015), it was found that when employed with the FR/CPR method, both the static and dynamic models produced results which were worse than those with no SGS model, i.e., implicit LES (ILES), for both the turbulent channel flow and isotropic turbulence decay.

In order to understand why SGS models provide no improvement, a study of various models with the 1D Burgers’ equation using the FR/CPR method was performed (Li and Wang, 2015) with the following findings:

- The static or dynamic models produce SGS stresses with very little correlation with the true SGS stress.
- The scale-similarity model of Bardina (Bardina, Ferziger, Reynolds, 1981) captures the phase of the SGS stress well.
- The results with SGS models are again worse than those by ILES (i.e., with no SGS model) for this simple example.

Finally, in addition to the three pacing items discussed above, progress in shock capturing, and efficient and scalable time marching algorithms for extreme-scale computers are also needed. Since these two topics were reviewed in a recent article (Wang, 2014), we will not elaborate here.
5. Conclusions

In this review article, we outlined the basic ideas of the FR/CPR formulation, and highlighted several recent developments and pacing items. It was demonstrated by many teams of researchers that the FR/CPR method has the potential to provide accurate numerical solutions for the Navier-Stokes equations, especially with large eddy simulations for turbulent flows. However, obstacles in high-order mesh generation, solution-based hp-adaptations, scalable time marching algorithms on extreme scale computers, and wall modeling for large eddy simulations need to be overcome for the method to become a powerful design tool in aerospace and mechanical industries.

Acknowledgements

The first author gratefully acknowledges support over many years from AFOSR, NASA, ARO, DOE, NSF, ONR, Michigan State University, Iowa State University, and the University of Kansas.

References


CGNS System - the standard for numerical analysis data (http://www.cgns.org).


Vermeire, B.C., and Nadarajah, S., Adaptive IMEX Time-Stepping for ILES Using the Correction Procedure via Reconstruction Scheme, AIAA-2013-2687.


