A Reconstructed Discontinuous Galerkin Method Based on a Hierarchical WENO Reconstruction for Compressible Flows on Hybrid Grids

Hong Luo

Department of Mechanical & Aerospace Engineering
North Carolina State University
Raleigh, NC 27695

Presented at
National Institute of Aerospace
May 28, 2013, Hampton, VA
Outline

• Background and Motivation

• Governing Equations

• Hierarchical Reconstructed Discontinuous Galerkin Method
  • Discontinuous Galerkin Method
  • WENO(P₁P₂): WENO reconstruction at P₂
  • HWENO(P₁P₂): WENO reconstruction at P₁

• Numerical Examples

• Concluding Remarks
Background

• Why DG?

  • Several useful mathematical properties with respect to conservation, stability, and convergence.
  • Easy extension to higher-order (>2\textsuperscript{nd}) schemes.
  • Well suited for complex geometries.
  • Easy adaptive strategies, allowing implementation of \textit{hp}-refinement and hanging nodes.
  • \textbf{Compact} and highly parallelizable.
  • Accuracy for low Mach number flows.
Background

• Why not DG?

  • High computing costs (more degrees of freedom)
    → CPU time
    → Storage requirements

  • Treatment of discontinuities (like all other high-order methods)
    → Sensitive to the implementation of limiters
    → Lead to loss of high-order accuracy
Background

To reduce high computing costs of the DG methods, Reconstructed DG (RDG($P_n P_m$)) Schemes were introduced by Dumbser et al.

- $P_n$ indicates that a piecewise polynomial of degree of $n$ is used to represent a DG solution.

- $P_m$ represents a reconstructed polynomial solution of degree of $m$ ($m \geq n$) that is used to compute the fluxes and source terms.

- Provide a unified formulation for both finite volume and DG methods, and contain both classical finite volume and standard DG methods as two special cases of RDG($P_n P_m$) schemes.
## Classification of the RDG(PnPm) Schemes

<table>
<thead>
<tr>
<th>Order of Accuracy</th>
<th>Schemes</th>
</tr>
</thead>
<tbody>
<tr>
<td>O(1)</td>
<td>RDG(P0P0) (DG(P0))</td>
</tr>
<tr>
<td>O(2)</td>
<td>RDG(P0P1) RDG(P1P1) (DG(P1))</td>
</tr>
<tr>
<td>O(3)</td>
<td>RDG(P0P2) RDG(P1P2) RDG(P2P2)</td>
</tr>
<tr>
<td>O(4)</td>
<td>RDG(P0P3) RDG(P1P3) RDG(P2P3) RDG(P3P3)</td>
</tr>
<tr>
<td></td>
<td>:</td>
</tr>
<tr>
<td>O(M+1)</td>
<td>RDG(P0Pm) … RDG(PnPm) … RDG(PmPm) DG</td>
</tr>
</tbody>
</table>

FV: New Class
• Observation: The construction of an accurate and efficient reconstruction operator is crucial to the success of the RDG(PnPm) schemes.

• Objective: Develop a RDG method based on a hierarchical WENO reconstruction: HWENO(P₁P₂), for compressible flows with strong discontinuities on hybrid grids.
  • enhance the accuracy, and therefore reduce the high computational costs of the underlying DG methods
  • avoid the spurious oscillations in the vicinity of strong discontinuities, and therefore maintain the non-linear stability, and naturally linear stability.

• Attempt to address the two weakest links of the DG methods.
Compressible Navier-Stokes Equations

\[
\frac{\partial U(x, t)}{\partial t} + \frac{\partial F_k(U(x, t))}{\partial x_k} = \frac{\partial G_k(U(x, t))}{\partial x_k}
\]

F: inviscid flux vector
G: viscous flux vector
U: conservative variable state vector
DG($p$) Method

\[
\frac{d}{dt} \int_{\Omega_e} U_h B_i d\Omega + \int_{\Gamma_e} F_k(U_h) n_k B_i d\Gamma - \int_{\Omega_e} F_k(U_h) \frac{\partial B_i}{\partial x_k} d\Omega = \int_{\Gamma_e} G_k(U_h) n_k B_i d\Gamma - \int_{\Omega_e} G_k(U_h) \frac{\partial B_i}{\partial x_k} d\Omega, \quad 1 \leq i \leq N
\]

$B_i(x)$: basis functions of the polynomials of degree $p$, $1 \leq i \leq N$.

$N$: dimension of the polynomial space $p$.

$\uparrow$

Discontinuous Galerkin method of degree $p$ DG(P) : $O(h^{p+1})$

$F_k(U_h) n_k = H_k(U_h^L, U_h^R, n_k) \leftarrow$ Numerical Riemann flux function

$G_k(U_h, \frac{\partial U_h}{\partial x_i}) n_k = H_v(U_h^L, U_h^R, \frac{\partial U_h^L}{\partial x_i}, \frac{\partial U_h^R}{\partial x_i}, n)$

The computation of the viscous fluxes has to properly resolve the discontinuities at the interfaces.
Finite Element Basis (Nodal DGM)

\[ U_h = \sum_{i=1}^{N} U_i(t) B_i(x), \]

Bi: Finite element shape functions.

N : Dimension of polynomial space.

The unknowns are the conservative variables at the nodes. Polynomial solutions are dependent on the shape of elements.

May 28, 2013
Taylor Basis (Modal DGM)

\[ U_h = \tilde{U} B_1 + U_x B_2 + U_y B_3 + U_z B_4 \]
\[ + U_{xx} B_5 + U_{yy} B_6 + U_{zz} B_7 + U_{xy} B_8 + U_{xz} B_9 + U_{yz} B_{10} \]

\[ U_x = \frac{\partial U}{\partial x} |_c \Delta x, \quad U_y = \frac{\partial U}{\partial y} |_c \Delta y, \quad U_z = \frac{\partial U}{\partial z} |_c \Delta z, \]

\[ U_{xx} = \frac{\partial^2 U}{\partial x^2} |_c \Delta x^2, \quad U_{yy} = \frac{\partial^2 U}{\partial y^2} |_c \Delta y^2, \quad U_{zz} = \frac{\partial^2 U}{\partial z^2} |_c \Delta z^2, \]

\[ U_{xy} = \frac{\partial^2 U}{\partial x \partial y} |_c \Delta x \Delta y, \quad U_{xz} = \frac{\partial^2 U}{\partial x \partial z} |_c \Delta x \Delta z, \quad U_{yz} = \frac{\partial^2 U}{\partial y \partial z} |_c \Delta y \Delta z. \]

The unknowns are the cell-averaged conservative variables and their derivatives at the center of the cells, regardless of element shapes.
DG(P₁) approximation

\[ N = 4, \quad \mathbf{U}_h = \tilde{\mathbf{U}} B_1 + \mathbf{U}_x B_2 + \mathbf{U}_y B_3 + \mathbf{U}_z B_4 \]

\[ \frac{d}{dt} \int_{\Gamma} \tilde{\mathbf{U}} d\Omega + \int_{\Gamma} \mathbf{F}_k (\mathbf{U}_h) \mathbf{n}_k d\Gamma = 0, \quad i = 1 \]

\[
\begin{align*}
\left( \int B_2 B_2 d\Omega \quad \int B_2 B_3 d\Omega \quad \int B_2 B_4 d\Omega \right) \left( \frac{d\mathbf{U}_x}{dt} \quad \frac{d\mathbf{U}_y}{dt} \quad \frac{d\mathbf{U}_z}{dt} \right) + \\
\left( \int B_3 B_2 d\Omega \quad \int B_3 B_3 d\Omega \quad \int B_3 B_4 d\Omega \right) \left( \frac{d\mathbf{U}_x}{dt} \quad \frac{d\mathbf{U}_y}{dt} \quad \frac{d\mathbf{U}_z}{dt} \right) + \\
\left( \int B_4 B_2 d\Omega \quad \int B_4 B_3 d\Omega \quad \int B_4 B_4 d\Omega \right) \left( \frac{d\mathbf{U}_x}{dt} \quad \frac{d\mathbf{U}_y}{dt} \quad \frac{d\mathbf{U}_z}{dt} \right)
\end{align*}
\]

\[
\int F_k n_k B_2 d\Gamma - \int F_k \frac{\partial B_2}{\partial x_k} d\Omega + \\
\int F_k n_k B_3 d\Gamma - \int F_k \frac{\partial B_3}{\partial x_k} d\Omega + \\
\int F_k n_k B_4 d\Gamma - \int F_k \frac{\partial B_4}{\partial x_k} d\Omega = 0
\]
Features:

- A finite volume code can be easily converted to a DG code.

- Same approximate polynomial solution for any shapes of elements:
  - Can be easily extended and implemented on arbitrary meshes.

- Cell-averaged variables and their derivatives are handily available:
  - Make implementation of WENO reconstruction easy and efficient

- Hierarchic basis
  - Make implementation of $p$-multigrid methods and $p$-refinement easy and efficient

May 28, 2013
• Objective:
  — Reconstruct a quadratic polynomial solution \(P_2\) from the underlying discontinuous linear polynomial DG polynomial solution \(P_1\) based on a WENO reconstruction.
2-exactness Least-squares Reconstruction

- From a linear polynomial DG solution in any cell $i$

$$U_i = \tilde{U}_i + U_{xi}B_2 + U_{yi}B_3 + U_{zi}B_4$$

- Reconstruct a quadratic polynomial solution $U^R$

$$U_i^R = \tilde{U}_i^R + U_{xi}^RB_2 + U_{yi}^RB_3 + U_{zi}^RB_4$$

$$+ U_{xxi}^RB_5 + U_{yyi}^RB_6 + U_{zzi}^RB_7$$

$$+ U_{xyi}^RB_8 + U_{xzi}^RB_9 + U_{yzi}^RB_{10}$$

- 10 degrees of freedom
Requirements for Reconstruction

• Conservation

• Compactness
  • Maintain the compactness of the underlying DG method
  • Necessary for unstructured arbitrary grids
  • Stencils involve only Von Neumann neighborhood (adjacent face-neighboring cells)
Least-squares Reconstruction ($P_1P_2$)

- Requiring conservation and reconstructed first derivatives equal to the ones of the underlying DG solution leads

$$\tilde{U}_i^R = \tilde{U}_i, \quad U_{xi}^R = U_{xi}, \quad U_{yi}^R = U_{yi}, \quad U_{zi}^R = U_{zi}$$

due to the judicious choice of Taylor-basis in the DG formulation

- Six second derivatives only need to be reconstructed
Least-squares Reconstruction ($P_1P_2$)

\[
U_j = \tilde{U}_i + U_{xi} B_2 + U_{yi} B_3 + U_{zi} B_4 \\
+ U_{xxi} B_5 + U_{yyi} B_6 + U_{zzi} B_7 + U_{xyi} B_8 + U_{xzi} B_9 + U_{yzi} B_{10}
\]

\[
\frac{\partial U}{\partial x} \bigg|_j = U_{xi} \frac{1}{\Delta x_i} + U_{xxi} \frac{B_2}{\Delta x_i} + U_{xyi} \frac{B_3}{\Delta x_i} + U_{xzi} \frac{B_4}{\Delta x_i}
\]

\[
\frac{\partial U}{\partial y} \bigg|_j = U_{yi} \frac{1}{\Delta y_i} + U_{yyi} \frac{B_3}{\Delta y_i} + U_{xyi} \frac{B_2}{\Delta y_i} + U_{yzi} \frac{B_4}{\Delta y_i}
\]

\[
\frac{\partial U}{\partial z} \bigg|_j = U_{zi} \frac{1}{\Delta z_i} + U_{zzi} \frac{B_4}{\Delta z_i} + U_{xzi} \frac{B_2}{\Delta z_i} + U_{yzi} \frac{B_3}{\Delta z_i}
\]
Least-squares Reconstruction ($P_1P_2$)

- Similar equations can be written for all face-neighboring cells, which leads to a non-square matrix.

- The number of face-neighboring cells for a tetrahedral cell is 4. As a result, the size of resulting non-square matrix is $16 \times 6$.

- This over-determined linear system of 16 equations for 6 unknowns can be solved in the least-squares sense.
Instability Issues of RDG Method in 3D

- The RDG method suffers from linear instability, similar to RDG(P₀P₁)
  - Instability occurs even for linear equations and in smooth flows.
  - Reconstruction stencils only involve von Neumann neighborhood, i.e., adjacent face-neighboring cells

- To maintain the linear stability
  - Augment stencils in the reconstruction
    → Destroy the compactness of the underlying DGM.
  - Use non-linear stability enforcement to achieve linear stability
    → Limiters in the case of RDG(P₀P₁) method
    → ENO/WENO reconstructions.
Hermite WENO Reconstruction

- On a tetrahedral cell $i$, a convex combination of the least-squares reconstructed 2\textsuperscript{nd} order derivatives at the cell itself and its four face-neighboring cells

$$\frac{\partial^2 U}{\partial x_i \partial x_j} \bigg|_i = \sum_{k=1}^{5} w_k \frac{\partial^2 U}{\partial x_i \partial x_j} \bigg|_k$$

$w_k$: weighting function

$$w_k = \frac{(\varepsilon + o_k)^{-\gamma}}{\sum_{i=1}^{5} (\varepsilon + o_i)^{-\gamma}}$$

$o_k$: oscillation indicator

$$o_k = \left[ \int_{\Omega_i} \left( \frac{\partial^2 U}{\partial x_i \partial x_j} \bigg|_k \right)^2 d\Omega \right]^2$$

$\varepsilon \rightarrow$ a small positive number

$\gamma \rightarrow$ an integer parameter
Hermite WENO Reconstruction

- Central stencil
  - the least-squares reconstructed polynomial at the cell itself

- Biased stencils
  - the least-squares reconstructed polynomials on its four face-neighboring cells
• Observation:
  Although the WENO(P₁P₂) method does not introduce any new oscillatory behavior for the reconstructed curvature terms (second derivatives) due to the WENO reconstruction, it cannot remove inherent oscillations in the underlying DG(P₁) solutions, leading to non-linear instability.

• Objective:
  Reconstruct and modify the linear part (first derivatives) of the resulting quadratic polynomial solution (P₂) in order to ensure non-linear instability for flows with strong discontinuities using WENO reconstruction.
WENO reconstruction at $P_1$ : $\text{HWENO}(P_1 P_2)$

- For a cell $i$, the following four stencils $(i,j1,j2,j3)$, $(i,j1,j2,j4)$, $(i,e,j3,j4)$, and $(i,j2,j3,j4)$, where $j1$, $j2$, $j3$, and $j4$ designate the four adjacent face-neighboring cells of the cell $i$, are chosen to construct a Lagrange polynomial such that

$$U_j = \tilde{U}_i + U_{xi}^R B_2 + U_{yi}^R B_3 + U_{zi}^R B_4$$

$$+ U_{xxi}^R B_5 + U_{yyi}^R B_6 + U_{zzi}^R B_7 + U_{xyi}^R B_8 + U_{xzi}^R B_9 + U_{yzi}^R B_{10}$$
The following four stencils \((i,j1), (i,j2), (i,j3),\) and \((i,j4)\) are chosen to construct a Hermite polynomial such that

\[
\begin{align*}
\frac{\partial U}{\partial x} \bigg|_j &= U^R_{xi} \frac{1}{\Delta x_i} + U^R_{xxi} \frac{B_2}{\Delta x_i} + U^R_{xyi} \frac{B_3}{\Delta x_i} + U^R_{xzi} \frac{B_4}{\Delta x_i} \\
\frac{\partial U}{\partial y} \bigg|_j &= U^R_{yi} \frac{1}{\Delta y_i} + U^R_{yyi} \frac{B_3}{\Delta y_i} + U^R_{xyi} \frac{B_2}{\Delta y_i} + U^R_{yzi} \frac{B_4}{\Delta y_i} \\
\frac{\partial U}{\partial z} \bigg|_j &= U^R_{zi} \frac{1}{\Delta z_i} + U^R_{zzi} \frac{B_4}{\Delta z_i} + U^R_{xzi} \frac{B_2}{\Delta z_i} + U^R_{yzi} \frac{B_3}{\Delta z_i}
\end{align*}
\]
Hermite WENO Reconstruction

- On a tetrahedral cell $i$, a convex combination of these eight reconstructed 1st derivatives and the first derivatives at the cell itself is used to modify the first derivatives

$$\left.\frac{\partial U}{\partial x_i}\right|_i = \sum_{k=1}^{9} w_k \left.\frac{\partial U}{\partial x_i}\right|_k$$

$w_k$: weighting function

$$w_k = \frac{(\varepsilon + o_k)^{-\gamma}}{\sum_{i=1}^{9} (\varepsilon + o_i)^{-\gamma}}$$

$\varepsilon \rightarrow$ a small positive number

$\gamma \rightarrow$ an integer parameter

$w_k$: weighting function

$o_k$: oscillation indicator

$$o_k = \left[ \int_{\Omega_i} \left( \left.\frac{\partial U}{\partial x_i}\right|_k \right)^2 d\Omega \right]^{1/2}$$
Hermite WENO Reconstruction

- Central stencil
  - the gradient from the DG solution itself at cell itself

- Biased stencils
  - the eight reconstructed gradients
### Cost Analysis (Tetrahedral Grids)

<table>
<thead>
<tr>
<th></th>
<th>RDG(P₀P₁)</th>
<th>RDG(P₁P₁)</th>
<th>RDG(P₁P₂)</th>
<th>RDG(P₂P₂)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Number of quadrature points for boundary integrals</strong></td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td><strong>Number of quadrature points for domain integrals</strong></td>
<td>0</td>
<td>4</td>
<td>5</td>
<td>11</td>
</tr>
<tr>
<td><strong>Reconstruction</strong></td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><strong>Order of Accuracy</strong></td>
<td>O(h²)</td>
<td>O(h²)</td>
<td>O(h³)</td>
<td>O(h³)</td>
</tr>
<tr>
<td><strong>Storage for Implicit Diagonal Matrix</strong></td>
<td>25 words Per element</td>
<td>400</td>
<td>400</td>
<td>2500</td>
</tr>
</tbody>
</table>
Design Goal

<table>
<thead>
<tr>
<th>FVFLO</th>
<th>RDGFLO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unstructured triangular/tetrahedral grid</td>
<td>Unstructured arbitrary grid</td>
</tr>
<tr>
<td>Finite volume/finite element formulation</td>
<td>Reconstructed Discontinuous Galerkin formulation</td>
</tr>
<tr>
<td>$h$-adaptation</td>
<td>$hp$-adaptation</td>
</tr>
<tr>
<td>None</td>
<td>Error Estimation Uncertainty Quantification</td>
</tr>
</tbody>
</table>
• Physics
  • Compressible flow for all speeds
  • Inviscid, Laminar, Turbulent (RDG(P₀P₁))
  • Chemically reactive flows (RDG(P₀P₁))

• Numerics
  • Unstructured Hybrid Mesh (tetrahedral, pyramidal, prismatic, and hexahedral)
  • Reconstructed Discontinuous Galerkin Formulation
  • Implicit (SGS, GMRES+LU-SGS)/Explicit (RK) Time Integration
  • \( p \)-multigrid
  • HLLC, LDFSS, AUSM for Inviscid Fluxes
  • BR2, RDG for Viscous Fluxes
  • BGK for Inviscid+Viscous Fluxes

• Parallelization
  • MPI
Numerical Examples

• Strengths of the RDG method
  • Accuracy
  • Robustness
  • Essentially oscillation-free property
Accuracy Demonstration
Convection of a Gaussian and a square wave

The superior dissipation and dispersion property of DG!
Convection of a Gaussian and a square wave

Note the high accuracy and oscillation-free of the RDG!
• Access the order of accuracy of the RDG(P₁P₁), WENO(P₁P₂) and HWENO(P₁P₂) methods for internal flows.
• Entropy production is served as the error measurement.

889 cells  6986 cells  449522 cells
254 pts  1555 pts  81567 pts
171 boundary pts  691 boundary pts  10999 boundary pts
Computed Velocity Contours by HWENO($P_1P_2$)
Convergence Study for different RDG methods

L2-error and order of convergence for the RDG(P₁P₁), WENO(P₁P₂), and HWENO(P₁P₂) methods

<table>
<thead>
<tr>
<th>Length Scale</th>
<th>RDG(P₁P₁)</th>
<th>WENO(P₁P₂)</th>
<th>HWENO(P₁P₂)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L²-error</td>
<td>Order</td>
<td>L²-error</td>
</tr>
<tr>
<td>6.552E-2</td>
<td>2.438E-3</td>
<td></td>
<td>2.183E-3</td>
</tr>
<tr>
<td>3.295E-2</td>
<td>7.356E-4</td>
<td>1.744</td>
<td>2.794E-4</td>
</tr>
<tr>
<td>1.650E-2</td>
<td>1.807e-4</td>
<td>2.032</td>
<td>4.539E-05</td>
</tr>
</tbody>
</table>

Both WENO(P₁P₂) and HWENO(P₁P₂) deliver the designed 3rd order of convergence!!
• Access the order of accuracy of the RDG($P_1P_1$), WENO($P_1P_2$) and HWENO($P_1P_2$) methods for external flows.
• Entropy production is served as the error measurement.

<table>
<thead>
<tr>
<th>Method</th>
<th>535 cells</th>
<th>62426 cells</th>
<th>16467 cells</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>167 points</td>
<td>598 points</td>
<td>3425 points</td>
</tr>
<tr>
<td></td>
<td>124 boundary pts</td>
<td>322 boundary pts</td>
<td>1188 boundary pts</td>
</tr>
</tbody>
</table>
Efficiency Comparison for Different RDG Methods

- L2 norm versus CPU time
- Convergence history versus CPU time (Second)
Convergence Study for different RDG methods

L2-error and order of convergence for the RDG(P₁P₁), WENO(P₁P₂), and HWENO(P₁P₂) methods

<table>
<thead>
<tr>
<th>Length scale</th>
<th>RDG(P₁P₁)</th>
<th>WENO(P₁P₂)</th>
<th>HWENO(P₁P₂)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L²-error</td>
<td>Order</td>
<td>L²-error</td>
</tr>
<tr>
<td>7.760E-2</td>
<td>1.783E-2</td>
<td></td>
<td>1.052E-2</td>
</tr>
<tr>
<td>4.688E-2</td>
<td>5.010E-3</td>
<td>2.519</td>
<td>1.317E-3</td>
</tr>
<tr>
<td>2.476E-2</td>
<td>1.232E-3</td>
<td>2.198</td>
<td>1.978E-4</td>
</tr>
</tbody>
</table>

Both WENO(P₁P₂) and HWENO(P₁P₂) deliver the designed 3rd order of convergence!!
Transonic Flow past an ONERA M6 Wing \((M_\infty = 0.84, \alpha = 3.06^\circ)\)

- Access the accuracy and non-oscillatory property of the HWENO\((P_1P_2)\) method for flows with discontinuities.

**Computed Pressure Contours**

<table>
<thead>
<tr>
<th>Method</th>
<th>Nelem</th>
<th>Npoin</th>
<th>Nboun</th>
</tr>
</thead>
<tbody>
<tr>
<td>WENO((P_0P_1))</td>
<td>593,169</td>
<td>110,282</td>
<td>19,887</td>
</tr>
<tr>
<td>HWENO((P_1P_2))</td>
<td>95,266</td>
<td>18,806</td>
<td>5,287</td>
</tr>
</tbody>
</table>
Computed Pressure Coefficient and Entropy Production Distributions at different spanwise locations

\[ \eta = 0.20 \]
Computed Pressure Coefficient and Entropy Production Distributions at different spanwise locations

\[ \eta = 0.44 \]
Computed Pressure Coefficient and Entropy Production Distributions at different spanwise locations

$\eta=0.80$
Blasius Boundary Layer (M=0.2, Re=100,000)

61x17x1
40 cells on the plate

Eta =1.2
dy= 0.1291E-2

Eta=1.3
dy= 0.155869E-3

Npoin=1800
Nboun=210
62 grid points on the flat plate.
Dy=0.3464E-03 at LE,
Dy=0.82649E-03 at TE.
Computed skin friction coefficients (RDG(P1))

Eta = 1.2
Hybrid

Eta = 1.3
Triangular
Computed skin friction coefficients (RDG(P2))

\[ \eta = 0.95 \]

\[ \eta = 1.2 \]
\[ \delta_y = 0.1291 \times 10^{-2} \]

\[ \eta = 1.3 \]
\[ \delta_y = 0.155869 \times 10^{-3} \]

\( \text{Hybrid} \)

\( \text{Triangular} \)
Computed X-velocity profiles at x=0.2(RDG(P1))

Eta=1.2

Eta=1.3

Hybrid

Triangular

May 28, 2013
Computed X-velocity profiles at x=0.2(RDG(P2))

Eta=1.2

Hybrid

Eta=1.3

Triangular
Computed Y-velocity profiles at $x=0.2$ (RDG(P1))

**Eta=1.2**

**Eta=1.3**

**Hybrid**

**Triangular**
Computed Y-velocity profiles at $x=0.2$(RDG(P2))

$\eta=1.2$

$\eta=1.3$

Hybrid

Triangular
Robustness Demonstration
Water flow in a convergent-divergent nozzle

May 28, 2013
Vapor flow in a convergent-divergent nozzle

Stiffened EOS

No single parameter is changed !!!
No time-derivative preconditioner is required !!!
Access the accuracy for solving low Mach number flow problems.

- Obtained by the RDG($P_0P_1$) on the finest grid
- Obtained by the RDG($P_1P_1$) on the fine grid
- Obtained by the RDG($P_1P_2$) on the fine grid
Comparison of the Computed Velocity Distributions on the Surface of the Sphere

![Graph showing velocity distributions](image)
Parallel Performance
Graph of domain decomposition by METIS
128 partitions and 124,706 elements
Explicit Method                                         Implicit Method

Parallel speedup and efficiency on a single node (up to 16 CPUs)
Explicit Method                                     Implicit Method

Parallel speedup and efficiency on a multiple nodes (up to 8 nodes with 16 CPUs per node)
Applications
• Access the accuracy and non-oscillatory property of the HWENO\(^{(P_1P_2)}\) method for flows with strong discontinuities.

Computed Pressure Contours
(nelem=319,134, npoin=61,075, nboun=14,373)
Computed Pressure Coefficient Distributions at different spanwise locations

η = 0.4077

η = 0.51
• Demonstrate that the HWENO($P_1P_2$) method can be used for computing complicated flows of practical interest.
• Flow condition: $M_\infty=0.85$, $\alpha=2^\circ$

(nelem = 253,577, npoin = 48,851, nboun = 11,802)
Computed Mach Number Contours
Subsonic flow past a SD7003 airfoil (M=0.1, Re=10,000)

23,172 prismatic elements,
2,225 hexhedral elements
text
25,397 grid

Computed vorticity contours
Unsteady Viscous Flow over Tandem Airfoils
$M=0.2$, $Re=10,000$, $\alpha=0$

2,902 hexahedral elements, 4,385 prisms, 10,418 grid points,
• Demonstrate that the WENO(P₁P₂) method can be used for computing vortex flows of practical interest.
• Flow condition: $M_\infty = 0.3$, $Re = 4,000$

(Tetrahedral grid: Nelem = 674,260, Npoin = 120,531, Nboun = 12,991)

Computed Mach Number Contours and streamlines
Concluding Remarks

• A reconstructed discontinuous Galerkin method based on a Hierarchical WENO reconstruction, HWENO(P₁P₂) has been developed for compressible flows at all speeds on hybrid grids.

• The HWENO(P₁P₂) method is able to provide sharp resolution of shock waves essentially without over- and under-shoots for discontinuities and achieve the designed third-order of accuracy for smooth flows.

• RDG methods have the potential to provide a superior alternative to the traditional FV methods, and to become a main choice for the next generation of CFD codes.

• A higher-order RDG-based CFD code will ultimately deliver a more accurate, efficient, robust, and reliable simulation tool with confidence that will enable us to solve flow problems at resolutions never before possible by the current state-of-the-art CFD technology.
Current Work

- Extension of the RDG method for turbulent flows
- Implementation of $hp$-adaptation on hybrid grids
- Port of the RDGFLO code on hybrid CPU/GPU architectures
Thank you