The hybridized DG methods for WS1, WS2, and CS2 test cases

P. Fernandez, N.C. Nguyen and J. Peraire

Aerospace Computational Design Laboratory
Department of Aeronautics and Astronautics, MIT
WS1: Taylor-Green Vortex
Case description

• Flow conditions:
  \[ \text{Re} = \frac{U_0 L}{\nu} = 1,600 \quad \text{and} \quad \infty \]
  \[ M = \frac{U_0}{a_0} = 0.1 \]

• Solver: DIGASO

• Discretization scheme:
  - Space: 3rd-order IEDG
  - Time: 3rd-order DIRK(3,3)

• Quadrature rule: Gauss-Legendre with exact integration of polynomials up to 5th-order

• Number of elements: 64 x 64 x 64 hexes

• Baseline CFL number: 0.375

• Studies presented:
  - Riemann solver
  - SGS model: ILES, Smagorinsky, WALE, Vreman
  - Time-step size

Source: G. Giangaspero et al., HiOCFD3, 2015
Effect of Riemann solver

Time evolution of kinetic energy dissipation rate

Numerical dissipation of kinetic energy

Re = 1600

Re = ∞

Laminar regime  →  Turbulent regime

Subgrid scales appear in the flow
Effect of Riemann solver

Time evolution of kinetic energy dissipation rate

Numerical dissipation of kinetic energy

Re = 1600

Re = ∞

Numerical dissipation 2 orders of magnitude smaller than total dissipation

Subgrid scales appear in the flow

Subgrid scales appear in the flow
Effect of SGS model

- All SGS models produce unphysical dissipation in the laminar regime
- Built-in ILES capability in DG more accurate than explicit SGS models to detect SGS scales and add numerical dissipation only under those conditions
Kinetic energy spectra at $t = 9 \, L / U_0$

ILES with different Riemann solvers

Explicit vs. implicit LES

Energy bump due to over-upwinding

More premature end of inertial range with explicit models
Effect of time-step size

Time evolution of kinetic energy dissipation rate

Numerical generation of entropy

- Vorticity and entropy modes captured with local CFL $\sim 1$. Acoustic modes require CFL $\ll 1$ (recall $M_0 = 0.1$)
- **Optimal LES implementations** likely with local CFL $> 1$: Global CFL $>> 1$ for wall-bounded flows
WS2: Turbulent Channel Flow
Case description

- **Flow conditions:**
  \[ \text{Re}_T = 182 \text{ and } 544 \quad | \quad M = 0.1 \]

- **Channel size:** \( 4\pi\delta \times 2\delta \times 2\pi\delta \)

- **Solver:** DIGASO

- **Discretization scheme:**
  - Space: 3rd- and 5th- order IEDG
  - Time: 3rd-order DIRK(3,3)

- **Quadrature rule:** Gauss-Legendre with exact integration of polynomials up to 5th- and 9th-order, respectively.

- **Number of elements:** 48 x 32 x 40

- **Studies presented:**
  - SGS model
    Riemann solver study is presented in *(Fernandez et al., 2017b)*

Mesh resolution in wall units

<table>
<thead>
<tr>
<th></th>
<th>( \text{Re}_T = 182 )</th>
<th>( \text{Re}_T = 544 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Delta x^+ )</td>
<td>23.8</td>
<td>71.2</td>
</tr>
<tr>
<td>( \Delta y^+_{\text{avg}} )</td>
<td>5.69</td>
<td>17.0</td>
</tr>
<tr>
<td>( \Delta y^+_{\text{wall}} )</td>
<td>0.438</td>
<td>1.310</td>
</tr>
<tr>
<td>( \Delta z^+ )</td>
<td>14.3</td>
<td>42.7</td>
</tr>
<tr>
<td>( \Delta t^+ )</td>
<td>0.290</td>
<td>0.740</td>
</tr>
</tbody>
</table>

Source: P. Blonigan et al., APS DFD Meeting 2016
Effect of SGS model

- ILES more accurate than explicit LES at both Reynolds numbers
- Viscous sublayer: Accurately resolved in all cases (except Smagorinsky-LES)
- Log layer: Accurately resolved in all cases
Effect of SGS model

- **Buffer layer**: Not accurately resolved. All simulations introduce too much “numerical transport of momentum” (not the same as “numerical dissipation”!)

- Ratio “momentum transport-to-dissipation” due to discretization errors and explicit SGS models is larger than the true SGS value
Effect of accuracy order

- 3rd- and 5th-order accurate LES have same number of DOFs and time-step size
- LES predictions improve by increasing accuracy order
- Accurately resolving the large turbulent scales by using a high-order method plays a more important role than that of the subgrid scales

![Graph showing effect of accuracy order](image)
CS2: T106C LTP cascade
Case description

- **Solver:** DIGASO
- **Discretization scheme:**
  - Space: 3rd-order IEDG
  - Time: 3rd-order DIRK(3,3)
- **Number of elements:** 118,680 isoparametric hexes (baseline mesh)
- **Time-step size:** $\Delta t = 5 \cdot 10^{-3} c / v_\infty$ (CFL$_{global}$ = 6.6)
- **LES model:** No model (ILES)
- **Riemann solver:** Lax-Friedrichs-type in *(Fernandez et al., 2017b)*
- **Quadrature rule:** Gauss-Legendre with exact integration of polynomials up to 5th-order
Flow fields on the periodic plane

Mach number

Spanwise vorticity

Time average

Instantaneous
Non-dimensional grid size
Pressure and skin-friction coefficients

- Two types of inflow/outflow boundary conditions are considered
- Small differences in $C_p$ and $C_f$ observed
Analysis of boundary layer instabilities

The nomenclature and details of the post-processing strategy are described in (Fernandez et al., 2017a)
The nomenclature and details of the post-processing strategy are described in (Fernandez et al., 2017a)
Summary

- Implicit LES outperformed explicit SGS models for the transition prediction, wall-free turbulence and wall-bounded turbulence cases considered.

- Minor differences between Riemann solvers: DG methods have an auto-correction mechanism to compensate for overshoots in the Riemann solver.

- DG methods have a built-in implicit LES capability and add numerical dissipation in under-resolved turbulence simulations.

- Built-in implicit SGS model in DG methods is more accurate than explicit models since they add numerical dissipation only when SGS's are present in the flow.

- Higher order are important to capture detailed physics in turbulent flows at higher Reynolds number.

- Optimal LES implementations are likely achieved with local $CFL_{\text{local}} > 1$ ($CFL_{\text{global}} >> 1$ for wall-bounded flows).
Summary

- IEDG is the hybridized DG method of choice: It inherits computational efficiency from EDG and BC robustness from HDG.

- Below ~5th order accuracy, IEDG allows for more efficient implementations than other DG methods.

- Beyond ~5th order accuracy, memory requirements and flop count become prohibitive, and IEDG does not provide any advantages.

- ILU(0) + RAS(1) type preconditioners are extremely efficient for most LES cases considered.

- Accuracy and stability of ILU deteriorates for low cell Peclet numbers (time-step size needs to be reduced more than linearly w.r.t. mesh size): Issues for wall-resolved LES at Re > 500,000.

- Efficiency and scalable nonlinear solvers are required for these methods to be adopted in industrial applications.
Questions?

For additional details:


SciTech talk:

Title: A physics-based shock capturing method for unsteady laminar and turbulent flows
Session: FD-03, CFD for Capturing Flow Discontinuities
Setting: Monday, January 8, 9:30 AM, Room Sun 5

pablof@mit.edu cuongng@mit.edu peraire@mit.edu
Hybridized DG methods
Hybridized DG schemes

• Consider the unsteady compressible Navier-Stokes equations:

\[ \boldsymbol{q} - \nabla \boldsymbol{u} = 0, \quad \text{in } \Omega \times (0, T), \]
\[ \frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot \boldsymbol{F}_{NS}(\boldsymbol{u}, \boldsymbol{q}) = 0, \quad \text{in } \Omega \times (0, T). \]

\[ \boldsymbol{u} = \begin{pmatrix} \rho \\ \rho v_j \\ \rho E \end{pmatrix} \]

• The hybridized DG approach (Nguyen et al., 2015) introduces additional variables \( \hat{\boldsymbol{u}}_h \) on the element faces \( \partial K \). Then:

\[ (\boldsymbol{q}_h, \boldsymbol{u}_h)|_K = f(\hat{\boldsymbol{u}}_h|\partial K), \quad \forall K \in \mathcal{T}_h \]

• This yields a global problem:

\[ \boldsymbol{r}_h(\boldsymbol{u}_h) = \mathbf{0} \quad \rightarrow \quad \boldsymbol{r}_h(\hat{\boldsymbol{u}}_h) = \mathbf{0} \]

Standard DG \quad \text{Hybridized DG}
Hybridized DG schemes

- “Hybridized DG schemes” (Nguyen et al., 2015) are a family of numerical schemes.
- Different choices of the space for $\hat{u}_h$ lead to different schemes. Three examples:

<table>
<thead>
<tr>
<th>Method</th>
<th>Nature of $\hat{u}_h$ space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hybridizable DG (HDG) (Peraire et al., 2010)</td>
<td>Discontinuous across faces</td>
</tr>
<tr>
<td>Embedded DG (EDG) (Peraire et al., 2011)</td>
<td>Continuous across faces</td>
</tr>
</tbody>
</table>
| Interior Embedded DG (IEDG) (Fernandez et al., 2016) | Interior faces: Continuous  
Boundary faces: Discontinuous |
Hybridized DG. Efficiency and accuracy

**Efficiency**

- Non-zero entries in Jacobian of global system:
  \[ \text{NNZ} = N_p N_c^2 \alpha_{\text{NNZ}} \]

<table>
<thead>
<tr>
<th>( \text{NNZ} \equiv \text{Number of non-zeros} )</th>
<th>( N_p \equiv \text{Number of mesh vertices} )</th>
<th>( N_c \equiv \text{Number of components of the PDE} )</th>
</tr>
</thead>
</table>

**Accuracy**

- Optimal accuracy \((p + 1)\) in \( u_h \).
- Superconvergence \((p + 2)\) in \( u_h \) can be inexpensively achieved (HDG only).
- Optimal accuracy \((p + 1)\) in \( q_h \) (HDG only)

**Values of \( \alpha_{\text{NNZ}} \) (tetrahedra)**

<table>
<thead>
<tr>
<th></th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
<th>5th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard DG</td>
<td>480</td>
<td>3,000</td>
<td>12,000</td>
<td>36,750</td>
</tr>
<tr>
<td>HDG</td>
<td>756</td>
<td>3,024</td>
<td>8,400</td>
<td>18,900</td>
</tr>
<tr>
<td>EDG</td>
<td>15</td>
<td>230</td>
<td>1,311</td>
<td>4,410</td>
</tr>
<tr>
<td>IEDG</td>
<td>&lt;15</td>
<td>&lt;230</td>
<td>&lt;1,311</td>
<td>&lt;4,410</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Accuracy order</th>
<th>( u_h )</th>
<th>( q_h )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard DG</td>
<td>( p + 1 )</td>
<td>( p )</td>
</tr>
<tr>
<td>HDG</td>
<td>( p + 1 )</td>
<td>( p + 1 )</td>
</tr>
<tr>
<td>EDG</td>
<td>( p + 1 )</td>
<td>( p )</td>
</tr>
<tr>
<td>IEDG</td>
<td>( p + 1 )</td>
<td>( p )</td>
</tr>
</tbody>
</table>
Parallel implementation and iterative solvers
Parallel iterative solver

• The hybridized DG discretization yields a nonlinear system of equations at every time-step

\[ r_h(\hat{u}_h) = 0 \]

• An efficient and scalable solution procedure is required for these methods to be adopted in industrial applications

• We discuss next on parallel iterative solvers
Parallel iterative solver

Nonlinear system: Newton or quasi-Newton method

- Initial guess $\hat{u}_h^{n,0}$ at time-step $n$ computed with a reduced-basis minimum-residual algorithm:

$$\hat{u}_h^{n,0} = \sum_{j=1}^{s} \alpha_j \hat{u}_h^{n-j}$$

$$\alpha_1, \ldots, \alpha_s = \arg \min_{(\beta_1, \ldots, \beta_s) \in \mathbb{R}^s} \left\| r_h \left( \sum_{j=1}^{s} \beta_j \hat{u}_h^{n-j} \right) \right\|^2$$

Linearization

$$r_h(\hat{u}_h) = 0$$

$$K_h \delta \hat{u}_h = -r_h$$
Parallel iterative solver

Nonlinear system: Newton or quasi-Newton method
- Initial guess \( \hat{u}^{n,0}_h \) at time-step \( n \) computed with a reduced-basis minimum-residual algorithm

Linear system: Restarted GMRES method
- Parallel preconditioner \( M^{-1} \): “Traced node”-based \( \delta \) — overlapping restrictive additive Schwarz (Cai et al., 1999):
  \[
  M^{-1} := \sum_{i=1}^{N} R_i^0 (K_h)_i^{-1} R_i^\delta
  \]
  \( R_i^\beta \equiv \) Restriction operator onto the \( \beta \)-overlap subdomain

\[
(K_h)_i = R_i^\delta K_h R_i^\delta
\]
- Subdomain preconditioner: Block incomplete LU factorization with zero fill-in (BILU0) and MDF reorder (Persson et al., 2008)
  \[
  (K_h)_i^{-1} \approx \tilde{U}_i^{-1} \tilde{L}_i^{-1}
  \]
Parallel iterative solver

**Nonlinear system:** Newton or quasi-Newton method
- Initial guess $\hat{u}_h^{n,0}$ at time-step $n$ computed with a reduced-basis minimum-residual algorithm

**Linear system:** Restarted GMRES method
- Parallel preconditioner $M^{-1}$: “Traced node”-based $\delta$–overlapping restrictive additive Schwarz \cite{Cai1999}
- Subdomain preconditioner: Block incomplete LU factorization with zero fill-in (BILU0) and MDF reorder \cite{Persson2008}
Parallel iterative solver

**Nonlinear system: Newton or quasi-Newton method**
- Initial guess $\hat{u}_h^{n,0}$ at time-step $n$ computed with a reduced-basis minimum-residual algorithm

**Linear system: Restarted GMRES method**
- Parallel preconditioner $M^{-1}$: “Traced node”-based $\delta$ — overlapping restrictive additive Schwarz [Cai et al., 1999]
- Subdomain preconditioner: Block incomplete LU factorization with zero fill-in (BILU0) and MDF reorder [Persson et al., 2008]

**Others:**
- Mixed-precision approach for Newton-GMRES algorithm
- Adaptive quadrature rules
- Stabilized ILU factorization (sILU)
- Minimum Interaction Domain Decomposition (MIDD)
Parallel scalability

- Weak scaling for LES of Ecole Centrale de Lyon compressor cascade
- Low-speed compressor cascade with Reynolds numbers ranging from 200,000 to 400,000
- Numerical discretization:
  - Space: 3rd-order IEDG
  - Time: 3rd-order DIRK(3,3)
- Additional details:
  - Time-step size is kept constant in all runs
  - Computing platform: Titan (OLCF)
  - One MPI rank per physical core
  - GPU computing and hybrid OpenMP/MPI parallelization disabled

![Parallel scalability graph]


