High Order Discontinuous Galerkin Methods for Conservation Laws

Per-Olof Persson
Massachusetts Institute of Technology

Collaborators: J. Peraire, M. Drela, D. J. Willis, N. C. Nguyen,
E. Israeli, A. Uranga, A. Mohnot, J. P. Whitney

Boston University Aerospace and Mechanical Engineering Seminar

October 19, 2007
Motivation

- Need for higher fidelity CFD prediction
  - DNS/LES/DES applications
  - Accurate RANS for engineering applications (drag prediction, rotor dynamics, fluid/structure interaction, flapping flight)
  - Computational aeroacoustics (direct solution of compressible flow, accurate computation of noise sources)

<table>
<thead>
<tr>
<th></th>
<th>FVM</th>
<th>FDM</th>
<th>FEM</th>
<th>DG</th>
</tr>
</thead>
<tbody>
<tr>
<td>High-order/Low dispersion</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Unstructured meshes</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Stability for conservation laws</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
</tr>
</tbody>
</table>
The Discontinuous Galerkin Method

- DG satisfies three fundamental requirements:
  - High-order/Low dispersion
  - Unstructured meshes
  - Stability for conservation laws

- However, several problems to resolve:
  - High-order geometry representation and mesh generation
  - High CPU/memory requirements (compared to FVM or H-O FDM)
  - Low tolerance to under-resolved features

The challenge is to make DG methods competitive for real-world problems
Example: Aeroacoustics and K-H Instability

- Inspired by calculations of Munz et al [03]) (linearized)

- Nonlinear behavior: Large scale acoustic wave interacts with small scale flow features, leading to vorticity generation

- $p = 7$ (8th order accuracy), 100-by-20 square elements
Example: Flow around Wedge Box

- Simple but representative geometry
- Well-studied aeroacoustics test case
- Low Mach number flow, high order methods required to capture and propagate acoustic waves
Example: Solid Dynamics

- Cast governing equations as system of first order conservation laws
- Account for involutions by appropriate selection of approximating spaces
  - Curl-free deformation tensor
- Use High-Order DG
- Lagrangian non-dissipative Neo-Hookean material
The Discontinuous Galerkin Method

- (Reed/Hill 1973, Lesaint/Raviart 1974, Cockburn/Shu 1989-, etc)
- Consider non-linear hyperbolic system in conservative form:
  \[ \mathbf{u}_t + \nabla \cdot \mathbf{F}_i(\mathbf{u}) = 0 \]
- Triangulate domain \( \Omega \) into elements \( \kappa \in T_h \)
- Seek approximate solution \( \mathbf{u}_h \) in space of element-wise polynomials:
  \[ \mathcal{V}_h^p = \{ \mathbf{v} \in L^2(\Omega) : \mathbf{v}|_\kappa \in P^p(\kappa) \ \forall \kappa \in T_h \} \]
- Multiply by test function \( \mathbf{v}_h \in \mathcal{V}_h^p \) and integrate over element \( \kappa \):
  \[ \int_\kappa [(\mathbf{u}_h)_t + \nabla \cdot \mathbf{F}_i(\mathbf{u}_h)] \mathbf{v}_h \, d\mathbf{x} = 0 \]
The Discontinuous Galerkin Method

- Integrate by parts:

\[ \int_{\kappa} \left[ (u_h)_t \right] v_h \, dx - \int_{\kappa} F_i(u_h) \nabla v_h \, dx + \int_{\partial \kappa} \hat{F}_i(u_h^+, u_h^-, \hat{n}) v_h^+ \, ds = 0 \]

with numerical flux function \( \hat{F}_i(u_L, u_R, \hat{n}) \) for left/right states \( u_L, u_R \) in direction \( \hat{n} \) (Godunov, Roe, Osher, Van Leer, Lax-Friedrichs, etc)

- Global view: Find \( u_h \in V_h^p \) such that this weighted residual is zero for all \( v_h \in V_h^p \)

- Error \( O(h^{p+1}) \) for smooth problems
The DG Method – Observations

- Reduces to the finite volume method for $p = 0$:

$$
(u_h)_t A_\kappa + \int_{\partial \kappa} \hat{F}_i(u_h^+, u_h^-, \hat{n}) \, ds = 0
$$

- Boundary conditions enforced naturally for any polynomial degree $p$

- Mass matrix block-diagonal (no overlap between element basis functions)

- Edge integrals connect neighboring elements – block-wise compact stencil
Nodal Basis and Curved Boundaries

- Nodal basis within each element → sparse element connectivities
- Isoparametric treatment of curved elements (not only at boundaries)
- Important research topic:
  - High order Meshing/Node placement
Viscous Discretization

- General approach for second derivatives:
  - Write as system of first order equations:
    \[
    u_t + \nabla \cdot F_i(u) - \nabla \cdot F_v(u, \sigma) = 0
    \]
    \[
    \sigma - \nabla u = 0
    \]
  - Discretize using DG, choose appropriate numerical fluxes \(\hat{\sigma}, \hat{u}\)

- Various schemes have been proposed:
  - \textit{BR1} [Bassi/Rebay 97]: Averaging, unstable and non-compact
  - \textit{BR2} [Bassi/Rebay 98]: Different lifting operator for each edge, compact connectivities, similar to Interior Penalty (IP)
  - \textit{LDG} [Cockburn/Shu 98]: Upwind/Downwind, non-compact
  - \textit{CDG} [Peraire/Persson 07]:
    Modification of LDG for local dependence – sparse and compact
The Local Discontinuous Galerkin Method

- Consider Poisson problem $-\nabla \cdot (\kappa \nabla u) = f$

- Write as system of first order equations,
  $$\begin{align*}
  -\nabla \cdot \sigma &= f \\
  \sigma &= \kappa \nabla u
  \end{align*}$$

- Use numerical inter-element fluxes
  $$\begin{align*}
  \hat{\sigma} &= \{\sigma_h\} - C_{11}[u_h] + C_{12}[\sigma_h] \\
  \hat{u} &= \{u_h\} - C_{12} \cdot [u_h]
  \end{align*}$$

- Choose a switch which satisfies $S_{K+}^{K-} + S_{K-}^{K+} = 1$ and set
  $$C_{12} = \frac{1}{2}(S_{K+}^{K-} n^+ + S_{K-}^{K+} n^-)$$

- In general, this will introduce non-local couplings
The Compact Discontinuous Galerkin Method

- Modification of LDG fluxes for compact stencil
- Strictly fewer connectivities than LDG and BR2
- More accurate than LDG and BR2
Matrix Representation

- Block matrix representation *fundamental for high performance*
  - Solver algorithms based on blocks
  - Up to 10 times higher performance with optimized BLAS
- Compact stencil $\Rightarrow$ Matrix structure given by mesh connectivities
- Harder to store LDG/BR2/IP efficiently (many use full block storage)

CDG – 2 arrays
LDG – 3 arrays + struct
BR2 / IP – 3 arrays
Preconditioners for Krylov Methods

• Performance of Krylov methods greatly improved by preconditioning

• Various standard methods:
  – Block-diagonal: \( \mathbf{A} \approx \text{blockdiag}(\mathbf{A}) \)
    * Poor in general
  – Block-Incomplete LU: \( \mathbf{A} \approx \tilde{\mathbf{L}}\tilde{\mathbf{U}} \)
    * Good for convection (with the right ordering, more later)
  – \( p \)-multigrid (low-degree) preconditioner
    * Good for diffusion

• Highly efficient combination [Persson/Peraire 06]
  – Use block-ILU(0) as post-smoother for coarse scale correction
  – Combines advantages of ILU and low-degree preconditioners
  – Cheap general purpose preconditioner
Minimum Discarded Fill Ordering

- Performance of ILU(0) highly dependent of ordering
- Greedy algorithm for element ordering [Persson/Peraire 07]:
  At step $j$, if $j'$ is chosen next, we would discard the fill

$$\Delta \tilde{U}_{ik}^{(j,j')} = -\tilde{U}_{ij'} \tilde{U}_{j'j}^{-1} \tilde{U}_{j'k},$$  for neighbors $i \geq j, k \geq j$ of element $j'$

- Choose the $j'$ that minimizes the norm of the discarded fill

$$w^{(j,j')} = \| \Delta \tilde{U}^{(j,j')} \|_F$$

- Some simplifications and a min-heap data structure $\implies \mathcal{O}(n \log n)$ computational cost

- Similar to the Minimum Degree algorithm, but considering the magnitude of the fill instead of just the size
Effect of Ordering on Convection-Diffusion

- Convection-Diffusion model problem, with \((\alpha, \beta) = (1, 2x), \varepsilon \geq 0:\)

\[
\frac{\partial u}{\partial t} + \nabla \cdot \begin{bmatrix} \alpha u \\ \beta u \end{bmatrix} - \nabla \cdot \begin{bmatrix} \varepsilon u_x \\ \varepsilon u_y \end{bmatrix} = 0
\]
Effect of Ordering on Convection-Diffusion

- Reverse Cuthill-McKee vs. Minimum Discarded Fill element ordering
- MDF perfect for convection, but small variations for diffusion
The ILU-p0 Preconditioner - Convection-Diffusion

- Block ILU perfect for convection, multigrid perfect for diffusion
- Block ILU-smoothed multigrid (BILU0-p1) almost perfect for any $\varepsilon$
Effect of Ordering on Navier-Stokes

- Model Navier-Stokes problem for large range of Reynolds numbers
<table>
<thead>
<tr>
<th>Problem</th>
<th>Parameters</th>
<th>Preconditioner/Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Block Jacobi</td>
</tr>
<tr>
<td></td>
<td>$\Delta t$</td>
<td>$M$</td>
</tr>
<tr>
<td>Inviscid</td>
<td>$10^{-3}$</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>$10^{-1}$</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>$\infty$</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>$10^{-3}$</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>$10^{-1}$</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>$\infty$</td>
<td>0.01</td>
</tr>
<tr>
<td>Laminar</td>
<td>$10^{-3}$</td>
<td>0.2</td>
</tr>
<tr>
<td>Re=1,000</td>
<td>$10^{-1}$</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>$\infty$</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>$10^{-3}$</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>$10^{-1}$</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>$\infty$</td>
<td>0.01</td>
</tr>
</tbody>
</table>
# The ILU-p0 Preconditioner – Navier-Stokes

<table>
<thead>
<tr>
<th>Problem</th>
<th>Parameters</th>
<th>Preconditioner/Iterations</th>
<th>Block Jacobi</th>
<th>Block G-S</th>
<th>Block ILU0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\Delta t$</td>
<td>$M$</td>
<td>BJ</td>
<td>BJ-p1</td>
<td>BGS</td>
</tr>
<tr>
<td><strong>Laminar</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Re=20,000</td>
<td>$10^{-3}$</td>
<td>0.2</td>
<td>26</td>
<td>19</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>$10^{-1}$</td>
<td>0.2</td>
<td>456</td>
<td>220</td>
<td>219</td>
</tr>
<tr>
<td></td>
<td>$\infty$</td>
<td>0.2</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>$10^{-3}$</td>
<td>0.01</td>
<td>160</td>
<td>90</td>
<td>61</td>
</tr>
<tr>
<td></td>
<td>$10^{-1}$</td>
<td>0.01</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>$\infty$</td>
<td>0.01</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td><strong>RANS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Re=$10^6$</td>
<td>$10^{-3}$</td>
<td>0.2</td>
<td>76</td>
<td>56</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>$10^{-1}$</td>
<td>0.2</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>$\infty$</td>
<td>0.2</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>$10^{-3}$</td>
<td>0.01</td>
<td>411</td>
<td>231</td>
<td>174</td>
</tr>
<tr>
<td></td>
<td>$10^{-1}$</td>
<td>0.01</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>$\infty$</td>
<td>0.01</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
</tbody>
</table>
**Effect of Ordering on Navier-Stokes**

- Good element ordering critical for Block-ILU/multigrid solver

<table>
<thead>
<tr>
<th>Problem</th>
<th>Element Ordering</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Random</td>
</tr>
<tr>
<td>Inviscid</td>
<td>51</td>
</tr>
<tr>
<td>Laminar, Re=1,000</td>
<td>200</td>
</tr>
<tr>
<td>Laminar, Re=20,000</td>
<td>197</td>
</tr>
<tr>
<td>RANS, Re=10^6</td>
<td>98</td>
</tr>
</tbody>
</table>
With BILU0-p1, convergence is almost independent of Mach number.
Artificial Viscosity for Underresolved Features

- Cannot resolve all flow features (shocks, RANS, singularities)
- Low dissipation makes DG sensitive to underresolution
- Detect by sensors and add artificial viscosity [Persson/Peraire 06, 07]
- Shock capturing with sub-cell resolution, robust solution of Spalart-Alamaras RANS model
Proposed Approach

- Use Artificial Viscosity with a Non-Linear Sensor
- Select Parameters based on Available Resolution
- DG implementation
  - Consistent Treatment of High Order Terms (CDG)
  - Do not exploit nature of Discontinuous Approximation
- Artificial Viscosity Models:
  
  **Shocks**
  - Viscosity Model: Laplacian form $\nabla \cdot (\nu_1 \nabla \vec{U})$
  - Sensor: Density

  **Eddy Viscosity Equation**
  - Viscosity Model: Laplacian form $\nabla \cdot (\nu_2 \nabla \tilde{\nu})$
  - Sensor: $\tilde{\nu}$
• Regularity of solution determined from expansion coefficients decay rate

• Periodic Fourier case:

\[ f(x) = \sum_{k=-\infty}^{\infty} g_k e^{ikx} \]

If \( f(x) \) has \( m \) continuous derivatives \( \rightarrow |g_k| \sim k^{-(m+1)} \)

• For simplices: Use orthonormal Koornwinder basis within each element
Example: RAE2822

- $M = 0.675$, $\alpha = 2.31^\circ$, $Re = 6.5 \cdot 10^6$

\[
p = 2 \quad \text{(constant $h/p$)} \quad \text{and} \quad p = 4
\]

$C_L = 0.6144$, $C_D = 0.0104$

$C_L = 0.6131$, $C_D = 0.0103$
Example: RAE2822

- $M = 0.675, \alpha = 2.31^\circ, \text{Re} = 6.5 \cdot 10^6$

  \[
p = 2 \quad \text{(constant } h/p) \quad p = 4
\]

\[C_L = 0.6144 \quad C_D = 0.0104\]
\[C_L = 0.6131 \quad C_D = 0.0103\]
Example: RAE2822

\[ C_p \]

\[ C_f \]

\[ \frac{x}{c} \]
Example: RAE2822, Transonic

- $M = 0.729$, $Re = 6.5 \cdot 10^6$
Example: RAE2822, Transonic

- $M = 0.729, \text{Re} = 6.5 \cdot 10^6$
Methods for Deforming Domains

- Several attempts using Arbitrary Lagrangian-Eulerian (ALE) formulations
  [Venkatasubban 95], [Lovtev et al 99], [Farhat/Geuzaine 04], [Ahn/Kallinderis 07])
  - Equations discretized on a deforming grid, time-dependent metric
  - At most third order accuracy in space and time demonstrated

- Alternative approach [Visbal/Gaitonde 02] in finite difference setting
  - Map from fixed reference domain to real time-varying domain
  - Needs non-conservative correction to satisfy geometric conservation law and preserve free-stream

- We use a mapping approach, however in a DG setting and with a conservative formulation for free-stream preservation
**ALE Formulation**

- Map from fixed reference domain $V$ to physical deformable domain $v(t)$
- A point $X$ in $V$ is mapped to a point $x(t) = G(X, t)$ in $v(t)$
- Introduce the *mapping deformation gradient* $G$ and the *mapping velocity* $v_X$ as

\[
G = \nabla_X G
\]
\[
v_X = \frac{\partial G}{\partial t} \bigg|_X
\]

and set $g = \det(G)$
Transformed Equations

• The system of conservation laws in the physical domain $v(t)$

$$
\frac{\partial U_x}{\partial t} \bigg|_x + \nabla_x \cdot F_x(U_x, \nabla_x U_x) = 0
$$

can be written in the reference configuration $V$ as

$$
\frac{\partial U_X}{\partial t} \bigg|_X + \nabla_X \cdot F_X(U_X, \nabla_X U_X) = 0
$$

where

$$
U_X = gU_x , \quad F_X = gG^{-1}F_x - U_XG^{-1}v_X
$$

and

$$
\nabla_x U_x = \nabla_X(g^{-1}U_X)G^{-T} = (g^{-1}\nabla_X U_X - U_X \nabla_X (g^{-1}))G^{-T}
$$

• Proof. See [Persson/Peraire/Bonet 07]
Geometric Conservation Law

- A constant solution in $v(t)$ is not necessarily a solution of the discretized equations in $V$, due to inexact integration of the Jacobian $g$

- The time evolution of $g$ is

$$\left. \frac{\partial g}{\partial t} \right|_X - \nabla_X \cdot (gG^{-1}\nu_X) = 0,$$

which in general is non-zero

- Visbal and Gaitonde added corrections to cancel the errors

- Our approach solves instead the convective system

$$\left. \frac{\partial (\bar{g}g^{-1}U_X)}{\partial t} \right|_X - \nabla_X \cdot F_X = 0$$

$$\left. \frac{\partial \bar{g}}{\partial t} \right|_X - \nabla_X \cdot (gG^{-1}\nu_X) = 0$$
Example: Euler Vortex

- Propagate an Euler vortex on a variable domain with

\[ x(\xi, \eta, t) = \xi + 2.0 \sin \left( \frac{2\pi \xi}{20} \right) \sin \left( \frac{\pi \eta}{7.5} \right) \sin \left( 1.0 \cdot \frac{2\pi t}{t_0} \right) \]
\[ y(\xi, \eta, t) = \eta + 1.5 \sin \left( \frac{2\pi \xi}{20} \right) \sin \left( \frac{\pi \eta}{7.5} \right) \sin \left( 2.0 \cdot \frac{2\pi t}{t_0} \right) \]

- Deformed meshes used for visualization only, everything is computed on the reference mesh
Example: Euler Vortex, Convergence

- Optimal order of convergence $O(h^{p+1})$ for mapped scheme
Example: Pitching Airfoil

- HT13 airfoil attached to translating and heaving point by torsional spring
- Fluid properties: $Re = 5000$, $M = 0.2$
- Forced vertical motion $r_z(t) = r_0 \sin \omega t$ (at leading edge)
- Moment equation: $I \ddot{\theta} + C \theta - S \ddot{r}_z(t) + M_{aero} = 0$
  - $I$ moment of inertia, $C$ spring stiffness, $S = m x_c$ static unbalance
  - $M_{aero}$ moment from fluid
Example: Pitching Airfoil/Flapper Design
Example: Heaving and Pitching Foil in Wake

- NACA 0012 foil heaving and pitching in wake of D-section cylinder
- Both oscillate $y(t) = A \sin(2\pi ft)$, foil pitching $\theta = a \sin(2\pi ft + \pi/2)$
- Based on experimental study [Gopalkrishnan et al 94]
Example: Heaving and Pitching Foil in Wake

- Thrust highly dependent on location (or phase)

Drag, Foil position 1

Drag, Foil position 2
Example: Fluid-Structure Interaction

- Consider interaction between fluid and a membrane

- Structural model:
  - Hyperelastic Neo-Hookean membrane formulation
  - Viscous, fluid-like dissipation
  - Continuous Galerkin (FEM) discretization

- Fully time-accurate coupling, since at any time $t$:
  - The membrane shape $x_c(s), y_c(s)$ induces a smooth mapping $x(X, t)$ for the entire domain
  - Forces $f_x(s), f_y(s)$ on membrane from DG solution $u(t)$
Example: Fluid-Structure Interaction

- Exploring structural design for efficient flapping flight

Fluid/membrane simulation

R/C dragonfly
Conclusions

• Important steps toward a practical DG solver for real-world problems

• Efficient viscous discretizations (CDG) and implicit solvers

• Artificial viscosity approach for producing grid independent RANS solutions and sub-cell shock capturing

• RANS is much harder than Euler/Laminar NS

• Optimal accuracy for deformable domains by mapping approach

• Current work: Extensions for LES/DES, structural dynamics, applications in flapping flight, aeroacoustics, and transonic flows