High Order Discontinuous Galerkin Methods for Fluid and Solid Mechanics

Per-Olof Persson

Department of Mathematics
Massachusetts Institute of Technology


Applied Math Colloquium
Department of Mathematics, MIT
March 3, 2008
Outline

1. Introduction

2. Discretization and Solvers
   - High-Order Mesh Generation
   - The Discontinuous Galerkin Method
   - The Compact Discontinuous Galerkin (CDG) Method
   - Preconditioning for Newton-Krylov Solvers
   - Stabilization with Artificial Viscosity

3. Deformable Domains
   - Mapping-based ALE Formulation
   - Flapping Flight Applications

4. Large Deformation Solid Dynamics
   - High-Order Lagrangian DG Formulation

5. Conclusions
Motivation

- Need for higher fidelity predictions in computational mechanics
  - 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)
  - Other problems involving wave propagation, multiple scale phenomena, and non-linear interactions
Example: Wave propagation

- Scalar convection equation $u_t + u_x = 0$, fixed resolution
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), 140-by-28 square elements
Unstructured meshes *required for realistic problems*

- Geometric and adaptive flexibility, accurate boundary treatment
- Tetrahedral meshes give robustness and automation
Motivation

Fundamental properties of Discontinuous Galerkin (DG) methods:

<table>
<thead>
<tr>
<th></th>
<th>FVM</th>
<th>FDM</th>
<th>FEM</th>
<th>DG</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) High-order/Low dispersion</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>2) Unstructured meshes</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>3) Stability for conservation laws</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>✓</td>
</tr>
</tbody>
</table>

However, several problems to resolve:

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

The challenge is to make DG competitive for real-world problems
Outline

1 Introduction

2 Discretization and Solvers
   - High-Order Mesh Generation
   - The Discontinuous Galerkin Method
   - The Compact Discontinuous Galerkin (CDG) Method
   - Preconditioning for Newton-Krylov Solvers
   - Stabilization with Artificial Viscosity

3 Deformable Domains
   - Mapping-based ALE Formulation
   - Flapping Flight Applications

4 Large Deformation Solid Dynamics
   - High-Order Lagrangian DG Formulation

5 Conclusions

1. Start with *any* topologically correct initial mesh, for example random node distribution and Delaunay triangulation
2. Move nodes to find force equilibrium in edges
   - Project boundary nodes using *implicit geometry* $\phi(x)$
   - Update element connectivities with Delaunay
DistMesh Applications

- High quality meshes from images and MRI/CT scans
DistMesh Applications

- Shape optimization by combined levelset/finite element method

Structural design
(compliance minimization)

Vibration control
(eigenvalue minimization)

![Graph showing density ρ and λ₁, λ₂ over iterations](image)
High-Order Curved Mesh Generation

- Open research topic: Unstructured curved mesh generation
- High quality meshes make it easier to avoid inversion
- DistMesh approach for automatic curving (ongoing work)
Outline

1 Introduction

2 Discretization and Solvers
   - High-Order Mesh Generation
   - The Discontinuous Galerkin Method
     - The Compact Discontinuous Galerkin (CDG) Method
     - Preconditioning for Newton-Krylov Solvers
     - Stabilization with Artificial Viscosity

3 Deformable Domains
   - Mapping-based ALE Formulation
   - Flapping Flight Applications

4 Large Deformation Solid Dynamics
   - High-Order Lagrangian DG Formulation

5 Conclusions
The Discontinuous Galerkin Method

- (Reed/Hill 1973, Lesaint/Raviart 1974, Cockburn/Shu 1989-, etc)
- Consider non-linear hyperbolic system in conservative form:

\[
{u_t} + \nabla \cdot {\mathcal F_i}(u) = 0
\]

- Triangulate domain \( \Omega \) into elements \( \kappa \in T_h \)
- Seek approximate solution \( u_h \) 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 \( \kappa \):

\[
\int_{\kappa} \left[ (u_h)_t + \nabla \cdot {\mathcal F_i}(u_h) \right] \mathbf v_h \, d\mathbf x = 0
\]
Integrate by parts:

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

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 problem: 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 solutions
The DG Method – Observations

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

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

- Boundary conditions enforced naturally for any degree $p$
- Block-diagonal mass matrix (no overlap between basis functions)
- Block-wise compact stencil – neighboring elements connected

![Mass Matrix](image1)

![Jacobian](image2)
Outline

1 Introduction

2 Discretization and Solvers
   - High-Order Mesh Generation
   - The Discontinuous Galerkin Method
   - The Compact Discontinuous Galerkin (CDG) Method
   - Preconditioning for Newton-Krylov Solvers
   - Stabilization with Artificial Viscosity

3 Deformable Domains
   - Mapping-based ALE Formulation
   - Flapping Flight Applications

4 Large Deformation Solid Dynamics
   - High-Order Lagrangian DG Formulation

5 Conclusions
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 DG Method

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

- Write as system of first order equations,
  \[-\nabla \cdot \sigma = f\]
  \[\sigma = \kappa \nabla u\]

- Use numerical inter-element fluxes

\[
\hat{\sigma} = \{\sigma_h\} - C_{11}[u_h] + C_{12}[\sigma_h] \\
\hat{u} = \{u_h\} - C_{12} \cdot [u_h]
\]

where \(\{\cdot\}, [\cdot]\) denote averaging and difference

- In particular, choosing \(C_{12} = 1\) or \(-1\) depending on a switch for each edge, will upwind/downwind \(\hat{\sigma}, \hat{u}\)
The Local DG Scheme

- Solving for the variables $\sigma_h$ gives

$$\sigma_h = \kappa \nabla_h u_h + \bar{\sigma}_h$$

where

$$\bar{\sigma}_h = \kappa r([u_h]) + \kappa l(C_{12} \cdot [u_h]) + \text{boundary terms}$$

and $r(\phi)$ and $l(q)$ are lifting operators (essentially $L_2$-projections)

- In general, this introduces non-local couplings since the lifting operators involve all element edges
The Compact DG Scheme

In the CDG scheme, we split the lifting operators into sums of edge-wise lifting operators \( r^e(\phi) \), \( l^e(q) \), and set

\[
\hat{\sigma} = \{\sigma^e_h\} - C_{11}[u_h] + C_{12}[\sigma^e_h] \\
\hat{u} = \{u_h\} - C_{12} \cdot [u_h]
\]

where \( \sigma^e_h = \kappa \nabla_h u_h + \bar{\sigma}^e_h \), with

\[
\bar{\sigma}^e_h = \kappa r^e([u_h]) + \kappa l^e(C_{12} \cdot [u_h]) + \text{boundary terms}
\]

Since only the lifting operator corresponding to the current edge is used, only neighboring elements are connected.
Error Estimates

- In primal form, the LDG scheme becomes (ignoring bnd terms):

\[
\int_{\Omega} \kappa (r([u]) + l(C_{12} \cdot [u])) \cdot (r([v]) + l(C_{12} \cdot [v])) \, dx = \\
\sum \sum \int_{\Omega} \kappa (r^e([u]) + l^e(C_{12} \cdot [u])) \cdot (r^f([v]) + l^f(C_{12} \cdot [v])) \, dx
\]

- The CDG scheme excludes some terms that are indefinite:

\[
\sum \int_{\Omega} \kappa (r^e([u]) + l^e(C_{12} \cdot [u])) \cdot (r^e([v]) + l^e(C_{12} \cdot [v])) \, dx = \\
\sum \sum \sum \delta_{ef} \int_{\Omega} \kappa (r^e([u]) + l^e(C_{12} \cdot [u])) \cdot (r^f([v]) + l^f(C_{12} \cdot [v])) \, dx
\]

- Non-compact terms are eliminated but the scheme remains stable
Coercivity and boundedness for the CDG scheme same as for LDG, leading to \textit{a-priori} estimates:

\[
\|\|u - u_h\|\| \leq Ch^p |u|_{p+1,\Omega}
\]

and

\[
\|u - u_h\|_{0,\Omega} \leq Ch^{p+1} |u|_{p+1,\Omega}
\]

with the norm

\[
\|\|v\|\|^2 = \sum_{K \in T_h} |v|_{1,K}^2 + \sum_{e \in E_i} \|r_e([v])\|_{0,\Omega}^2 + \sum_{e \in \partial \Omega_D} \|r_D(v)\|_{0,\Omega}^2
\]

Assumes \(C_{11} = \mathcal{O}(h^{-1})\), but is observed numerically for \(C_{11} = 0\)
The CDG Method – Summary

- Element-wise compact stencil
- Less connectivities than LDG/BR2/IP
- More accurate than LDG and BR2
Switches and Null-space Dimensions

- Unlike the LDG scheme, the CDG scheme appears to be stable for $C_{11} = 0$ and an *inconsistent switch* such as highest element number.

- Simple test [Sherwin et al 05]: Poisson problem, periodic boundary conditions, expected nullspace dimension = 1

### Nullspace dimension

<table>
<thead>
<tr>
<th>Polynomial order $p$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Consistent switch</strong></td>
<td>CDG</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>LDG</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>Natural switch</strong></td>
<td>CDG</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>LDG</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>
ILU and Switch Orientation

- Orientation of lower-triangular blocks important for ILU sparsity
- Take advantage of CDG’s insensitivity to orientation

Switch 1: 
*Same* LU storage

Switch 2: 
*More* LU storage
Switches and Null-space Dimensions

- No additional non-zeros in block-ILU(0) factorization using CDG
- Dense lower-triangular blocks using BR2 / IP

**CDG**

![Stiffness Matrix](image1)

![Block ILU(0)](image2)

640 non-zeros 640 non-zeros
Switches and Null-space Dimensions

- No additional non-zeros in block-ILU(0) factorization using CDG
- Dense lower-triangular blocks using BR2 / IP

**Stiffness Matrix**
- 784 non-zeros

**Block ILU(0)**
- 892 non-zeros
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 $\implies$ Matrix structure given by mesh connectivities
- Hard to store LDG/BR2/IP efficiently

CDG – 2 arrays
LDG – 3 arrays + struct
BR2 / IP – 3 arrays
Outline

1. Introduction

2. Discretization and Solvers
   - High-Order Mesh Generation
   - The Discontinuous Galerkin Method
   - The Compact Discontinuous Galerkin (CDG) Method
   - Preconditioning for Newton-Krylov Solvers
   - Stabilization with Artificial Viscosity

3. Deformable Domains
   - Mapping-based ALE Formulation
   - Flapping Flight Applications

4. Large Deformation Solid Dynamics
   - High-Order Lagrangian DG Formulation

5. Conclusions
Jacobian Sparsity Structure

- Simple model of sparsity pattern: Connectivity of dual mesh, with large complete graphs $A_{ij}$ in each cell
- Sizes of submatrices $A_{ij}$ typically $> 100$ DOFs
- Easy to obtain full BLAS performance! (e.g. in matvec or LU)
Preconditioners for Krylov Methods

- Performance of Krylov methods greatly improved by preconditioning

- Various standard methods for block matrices:
  - Block diagonal: $A \approx \text{blockdiag}(A)$
    - Poor in general
  - Block Incomplete LU: $A \approx \tilde{L}\tilde{U}$
    - 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
For our low-degree dual meshes, it is uncommon that an element $k$ is a neighbor of both element $j$ and $i$ when $j$ is a neighbor of $i$.

This simplifies the Block-ILU(0) algorithm:

$$
\tilde{U} \leftarrow A, \tilde{L} \leftarrow I
$$

for $j = 1$ to $n - 1$

for neighbors $i > j$ of $j$

$$
\tilde{L}_{ij} = \tilde{U}_{ij} \tilde{U}_{jj}^{-1}
$$

$$
\tilde{U}_{ii} \leftarrow \tilde{U}_{ii} - \tilde{L}_{ik} \tilde{U}_{ki}
$$

end for

end for

This means $\tilde{U}_{ij} = A_{ij}$ when $j > i$, or $\tilde{U}$ only differs from $A$ in the diagonal blocks.

Store only diagonal blocks of $\tilde{U}$!
Minimum Discarded Fill Element 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
  \[
  \Delta \tilde{U}_{ik}^{(j,j')} = -\tilde{U}_{ij'}\tilde{U}_{jj'}^{-1}\tilde{U}_{j'k}, \quad \text{for neighbors } i \geq j, k \geq j \text{ of element } j'
  \]
  - Choose the $j'$ that minimizes the norm of the discarded fill
    \[
    w^{(j,j')} = \| \Delta \tilde{U}^{(j,j')} \|_F
    \]
  - Some simplifications, min-heap data structure $\Longrightarrow O(n \log n)$ cost
  - Similar to the Minimum Degree algorithm, but considering the magnitude of the fill instead of just the size
  - Related to the Minimum Discarded Fill algorithm [D’Azevedo, Forsyth, Tang 92]
Compute MDF ordering \( p \) for block-matrix \( A \):

\[
\begin{align*}
B & \leftarrow (\tilde{A}^T)^{-1}A \\
C_{ij} & \leftarrow \|B_{ij}\|_F \\
\text{for } k = 1, \ldots, n \\
\quad & \Delta C \leftarrow 0 \\
\quad & \text{for neighbors } i, j \text{ of element } k, \ i \neq j \\
\quad & \quad \Delta C_{ij} \leftarrow C_{ik}C_{kj} \\
\quad & \text{end for} \\
\quad & w_k \leftarrow \|\Delta C\|_F \\
\text{end for} \\
\text{for } i = 1, \ldots, n \\
\quad & p_i \leftarrow \text{argmin}_j w_j \\
\quad & w_{p_i} \leftarrow \infty \\
\quad & \text{for neighbors } k \text{ of } p_i \text{ not yet numbered} \\
\quad & \quad \text{Recompute } w_k, \text{ only considering} \\
\quad & \quad \text{neighbors not yet numbered} \\
\quad & \text{end for} \\
\text{end for}
\end{align*}
\]

Pre-multiply by block diagonal
Reduce each block to scalar
Compute all weights
Discarded fill matrix
Fill weight
Main loop
Pivot with smallest fill-in
Do not choose \( p_i \) again
Update weights
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 ordering
- MDF makes ILU perfect for convection
- Ordering less important for diffusion, ILU remains poor

![Graph showing GMRES iterations for different ordering methods and convergence for convection and diffusion dominated cases.]
The ILU(0)-p1 Preconditioner

- Combination of block ILU and multigrid [Persson/Peraire 06]
- Coarse scale correction + post-smoothing by $\widetilde{A} = \widetilde{L}\widetilde{U}$:

  0. $A^{(0)} = P^T A P$ Precompute coarse operator, block wise
  1. $b^{(0)} = P^T b$ Restrict residual element/component wise
  2. $A^{(0)} u^{(0)} = b^{(0)}$ Solve coarse scale problem
  3. $u = Pu^{(0)}$ Prolongate solution element/component wise
  4. $u = u + \alpha \widetilde{A}^{-1} (b - Au)$ Apply smoother $\widetilde{A}$ with damping $\alpha$

- Restriction/prolongation operator $P$ block diagonal, based on orthogonal Koornwinder polynomials
- Coarse scale problem solved directly (2-D problems, serial) or iteratively by GMRES or $h$-multigrid
The ILU-p1 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 wide range of Reynolds numbers
The ILU-p1 Preconditioner – Navier-Stokes

- Convergence for various Reynolds numbers, $\Delta t$, and $M$
- $\times$ No convergence after 1,000 iterations

<table>
<thead>
<tr>
<th>Problem</th>
<th>Parameters</th>
<th>Preconditioner/Iterations</th>
</tr>
</thead>
<tbody>
<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-p1 Preconditioner – Navier-Stokes

- Convergence for various Reynolds numbers, $\Delta t$, and $M$
- $\times =$ No convergence after 1,000 iterations

<table>
<thead>
<tr>
<th>Problem</th>
<th>Parameters $\Delta t$ $M$</th>
<th>Preconditioner/Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\Delta t$</td>
<td>$M$</td>
</tr>
<tr>
<td>Laminar</td>
<td>$10^{-3}$</td>
<td>0.2</td>
</tr>
<tr>
<td>Re=20,000</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>
<tr>
<td>RANS</td>
<td>$10^{-3}$</td>
<td>0.2</td>
</tr>
<tr>
<td>Re=10$^6$</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>
**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>
Excellent convergence for Mach number $M \ll 1$ with BILU0-p1
<table>
<thead>
<tr>
<th>Outline</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>5</td>
</tr>
</tbody>
</table>
Artificial Viscosity for Underresolved Features

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

Mach

Sensor
• Regularity of solution determined from the decay rate of expansion coefficients in orthogonal basis

• Example: 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: Expand solution in orthonormal Koornwinder basis:

\[
\begin{align*}
  u &= \sum_{i=1}^{N(p)} u_i \psi_i, \\
  \hat{u} &= \sum_{i=1}^{N(p-1)} u_i \psi_i, \\
  s_e &= \log_{10} \left( \frac{(u - \hat{u}, u - \hat{u})_e}{(u, u)_e} \right)
\end{align*}
\]

• Determine elemental piecewise constant \( \varepsilon_e \)

\[
\varepsilon_e = \begin{cases} 
  0 & \text{if } s_e < s_0 - \kappa \\
  \frac{\varepsilon_0}{2} \left( 1 + \sin \frac{\pi (s_e - s_0)}{2\kappa} \right) & \text{if } s_0 - \kappa \leq s_e \leq s_0 + \kappa \\
  \varepsilon_0 & \text{if } s_e > s_0 + \kappa
\end{cases}
\]

where \( \varepsilon_0 \sim h/p, s_0 \sim 1/p^4 \) and \( \kappa \) empirical.
Euler Equations – Artificial Viscosity Models

- **Laplacian:** \( \frac{\partial u}{\partial t} + \nabla \cdot F(u) = \nabla \cdot (\epsilon \nabla u) \)
- **Physical:** \( \frac{\partial u}{\partial t} + \nabla \cdot F(u) = \nabla \cdot F_v(u; \text{Re}, \text{Pr}) \)

---

**Density**

<table>
<thead>
<tr>
<th>Mach 2</th>
<th>Mach 5</th>
<th>Mach 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laplace</td>
<td>Physical</td>
<td>Physical, ( \kappa = 0 )</td>
</tr>
</tbody>
</table>

---

**Mach**

<table>
<thead>
<tr>
<th>Mach 2</th>
<th>Mach 5</th>
<th>Mach 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>2.5</td>
<td>4.5</td>
<td>6</td>
</tr>
</tbody>
</table>
Example: RAE2822

- Turbulent RANS flow \((M = 0.675, \alpha = 2.31^\circ, \text{Re} = 6.5 \cdot 10^6)\)
- \(p\)-converged solution, fixed resolution \(h/p\)

\[
\begin{align*}
    p & = 2 & (\text{constant } h/p) & p = 4 \\
    C_L & = 0.6144 & C_D & = 0.0104 \\
    C_L & = 0.6131 & C_D & = 0.0103
\end{align*}
\]
Example: RAE2822

- Turbulent RANS flow \( (M = 0.675, \alpha = 2.31^\circ, Re = 6.5 \cdot 10^6) \)
- \( p \)-converged solution, fixed resolution \( h/p \)

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

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

- Highly accurate boundary forces even with coarse meshes
Example: RAE2822, Transonic

- Transonic flow \((M = 0.729, \text{Re} = 6.5 \cdot 10^6)\)
- Sub-cell resolution of shocks

\[ p = 4 \]
Example: RAE2822, Transonic

- Transonic flow \((M = 0.729, Re = 6.5 \cdot 10^6)\)
- Sub-cell resolution of shocks

\[ p = 4 \]
Outline

1 Introduction

2 Discretization and Solvers
   - High-Order Mesh Generation
   - The Discontinuous Galerkin Method
   - The Compact Discontinuous Galerkin (CDG) Method
   - Preconditioning for Newton-Krylov Solvers
   - Stabilization with Artificial Viscosity

3 Deformable Domains
   - Mapping-based ALE Formulation
   - Flapping Flight Applications

4 Large Deformation Solid Dynamics
   - High-Order Lagrangian DG Formulation

5 Conclusions
Methods for Deforming Domains

- Many ALE formulations for unstructured meshes [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 for finite differences [Visbal/Gaitonde 02]
  - Map from fixed reference domain to time-varying physical domain
  - Correction for the Geometric Conservation Law (GCL) with sources

- Our method: Mapping approach in a DG setting, with a conservative formulation for satisfying the GCL
  - Guaranteed stability
  - Arbitrary orders of accuracy in space and time
Map from 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 \textit{mapping deformation gradient} $G$ and the \textit{mapping velocity} $v_X$ as

\[
G = \nabla_X G
\]
\[
v_X = \left. \frac{\partial G}{\partial t} \right|_X
\]

and set $g = \det(G)$
The system of conservation laws in the physical domain $v(t)$

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

can be written in the reference configuration $V$ as

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

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 in $V$, due to inexact integration of the Jacobian $g$.
  - The time evolution of $g$ is
    
    $$ \frac{\partial g}{\partial t} \bigg|_X - \nabla_X \cdot (gG^{-1}v_X) = 0, $$
    
    which in general is non-zero.

- Visbal and Gaitonde added source terms to cancel the errors.

- Our approach solves instead the conservative system
  
  $$ \frac{\partial (\bar{g} g^{-1} U_X)}{\partial t} \bigg|_X - \nabla_X \cdot F_X = 0 $$
  
  $$ \frac{\partial \bar{g}}{\partial t} \bigg|_X - \nabla_X \cdot (gG^{-1}v_X) = 0 $$
Example: Euler Vortex

- Propagate an Euler vortex on a variable domain with

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

- Mapped scheme – Everything is computed on the reference mesh
Example: Euler Vortex, Convergence

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

![Graph showing convergence rates for different polynomial degrees ($p=1, 2, 3, 4, 5$) and element sizes ($h$). The graph compares mapped and unmapped schemes.]
1 Introduction

2 Discretization and Solvers
   - High-Order Mesh Generation
   - The Discontinuous Galerkin Method
   - The Compact Discontinuous Galerkin (CDG) Method
   - Preconditioning for Newton-Krylov Solvers
   - Stabilization with Artificial Viscosity

3 Deformable Domains
   - Mapping-based ALE Formulation
   - Flapping Flight Applications

4 Large Deformation Solid Dynamics
   - High-Order Lagrangian DG Formulation

5 Conclusions
Biologically-Inspired Flapping Flight

- Development of computational tools for studying flapping flight
- Challenging problems: Deforming domains, fluid-structure interaction, transitional flows, etc
Example: Pitching Airfoil

- 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, $S = mx_c$ 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: Locomotion of Free Flapping Body

- Oscillating plate, unconstrained horizontally
  [Vandenberghe et al 04, Alben/Shelley 05]
- Instability breaks symmetry and forces plate into motion
Example: Fluid-Structure Interaction

- Interaction between fluid and a hyperelastic membrane
- Compliancy can alleviate leading edge separation

Experiment (A. Song, Brown U)

Fluid/membrane simulation

Compliant membrane

Rigid flat plate
Example: Dragonfly, Compliant Wings

Experiment (A. Song, Brown U)

Fluid/membrane simulation
1 Introduction
2 Discretization and Solvers
   - High-Order Mesh Generation
   - The Discontinuous Galerkin Method
   - The Compact Discontinuous Galerkin (CDG) Method
   - Preconditioning for Newton-Krylov Solvers
   - Stabilization with Artificial Viscosity
3 Deformable Domains
   - Mapping-based ALE Formulation
   - Flapping Flight Applications
4 Large Deformation Solid Dynamics
   - High-Order Lagrangian DG Formulation
5 Conclusions
Lagrangian DG Formulation for Solid Dynamics

- Governing equations, conservation of linear momentum:
  \[ \frac{\partial p}{\partial t} - \nabla \cdot P = \rho_0 b \]
  with momentum \( p \), first Piola-Kirchhoff stress tensor \( P(F) \), deformation tensor \( F = \partial x/\partial X \)

- Hyperelastic neo-Hookean model for \( P(F) \)

- Many potential benefits:
  - Arbitrary orders of accuracy
  - Discontinuous elements less sensitive to locking
  - DG stabilization for ALE
  - Block-diagonal mass matrix
    \( \Rightarrow \) efficient explicit methods
Lagrangian DG Formulation for Solid Dynamics

- **Formulation 1:** Material points $x$ and momentum $p$:

\[
\begin{align*}
\frac{\partial x}{\partial t} &= \frac{p}{\rho_0} \\
\frac{\partial p}{\partial t} - \nabla \cdot P(F) &= \rho_0 b
\end{align*}
\]

- Cheap, polynomial spaces, CDG scheme for second derivatives
- Non-conservative, treatment of shocks unclear

- **Formulation 2:** Momentum $p$ and deformation gradient $F$:

\[
\begin{align*}
\frac{\partial p}{\partial t} - \nabla \cdot P(F) &= \rho_0 b \\
\frac{\partial F}{\partial t} - \nabla \cdot \left( \frac{p}{\rho_0} \otimes I \right) &= 0
\end{align*}
\]

- Conservative formulation, borrow shock capturing from CFD
- Expensive, needs curl-free spaces for $F$
Oscillating Beam, Energy Conservation

- Energy conservation not explicitly enforced by scheme
- Therefore, it is a good indicator of accuracy

Kinetic, potential, and total energy

\[
p = 1 \\
p = 2 \\
p = 3 \\
p = 4
\]
Volumetric Modeling of Thin Structures

- The high order discretizations allow for highly stretched elements
- Model plate problem, single element across thickness
- Discontinuous basis functions less sensitive to locking
Volumetric Modeling of Thin Structures

- Automatic treatment of coupling between beams/membranes
  - No special models required
  - Only a question of generating the stretched meshes

Thin elastic membrane

Coupling of thin beams/membrane
Important steps toward practical DG solver for realistic problems:

- Efficient viscous discretization (the CDG method)
- General purpose multigrid/block ILU preconditioner
- Robustness by sensors and artificial viscosity, for RANS and shocks
- Optimal accuracy for deformable domains by mapping approach
- High order Lagrangian DG formulation for solid dynamics

Current work: New sparser discretizations, 3-D curved mesh generation, orderings for parallel ILU, algebraic/geometric multigrid for coarse scale problem, extensions for LES/DES, coupling of DG fluid/structure formulations, applications in flapping flight, aeroacoustics, and transonic/supersonic flows