High Order Discontinuous Galerkin Methods
for Fluid and Solid Mechanics

Per-Olof Persson

Department of Mathematics
Massachusetts Institute of Technology

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

Lawrence Livermore National Laboratory
February 4, 2008
Outline

1. Introduction

2. Discretization and Solvers
   - The Discontinuous Galerkin Method
   - High-Order Mesh Generation
   - 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
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*
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
Example: Flow around Elliptic Wing

Unstructured meshes required for realistic problems

- Geometric and adaptive flexibility
- Robustness and automation
Outline

1. Introduction

2. Discretization and Solvers
   - The Discontinuous Galerkin Method
   - High-Order Mesh Generation
   - 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:

\[ \mathcal{V}_h^p = \{ v \in L^2(\Omega) : v|_\kappa \in P^p(\kappa) \ \forall \kappa \in T_h \} \]

- Multiply by test function \( v_h \in \mathcal{V}_h^p \) and integrate over element \( \kappa \):

\[ \int_{\kappa} \left[ (u_h)_t + \nabla \cdot \mathcal{F}_i(u_h) \right] v_h \, dx = 0 \]
The Discontinuous Galerkin Method

- Integrate by parts:

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

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 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 $p = 0$:

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

- 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](image)

![Jacobian](image)
Outline

1. Introduction

2. Discretization and Solvers
   - The Discontinuous Galerkin Method
   - High-Order Mesh Generation
     - 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

- Shape optimization by combined levelset/finite element method

Structural design
(compliance minimization)

Vibration control
(eigenvalue minimization)
DistMesh Applications

- High quality meshes from images and MRI/CT scans
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
   - The Discontinuous Galerkin Method
   - High-Order Mesh Generation
   - 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:

\[
\begin{align*}
    u_t + \nabla \cdot F_i(u) - \nabla \cdot F_v(u, \sigma) &= 0 \\
    \sigma - \nabla u &= 0
\end{align*}
\]

- 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 \(\hat{\sigma}, \hat{u}\)
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.
Element-wise compact stencil

Less connectivities than LDG/BR2/IP

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 $\implies$ Matrix structure given by mesh connectivities
- Hard to store LDG/BR2/IP efficiently

CDG – 2 arrays

LDG – 3 arrays + struct

BR2 / IP – 3 arrays
1. Introduction

2. Discretization and Solvers
   - The Discontinuous Galerkin Method
   - High-Order Mesh Generation
   - 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
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
**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 $\implies O(n \log n)$ 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 ordering
- MDF makes ILU perfect for convection
- Ordering less important for diffusion, ILU remains poor

![Graph showing effect of ordering on GMRES iterations](image)
The ILU(0)-p1 Preconditioner

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

0. $A^{(0)} = P^TAP$  \hspace{1cm} \textit{Precompute coarse operator, block wise}
1. $b^{(0)} = P^Tb$  \hspace{1cm} \textit{Restrict residual element/component wise}
2. $A^{(0)}u^{(0)} = b^{(0)}$  \hspace{1cm} \textit{Solve coarse scale problem}
3. $u = Pu^{(0)}$  \hspace{1cm} \textit{Prolongate solution element/component wise}
4. $u = u + \alpha\tilde{A}^{-1}(b - Au)$  \hspace{1cm} \textit{Apply smoother $\tilde{A}$ 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
- 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
Convergence for various Reynolds numbers, $\Delta t$, and $M$

$\times = \text{No convergence after 1,000 iterations}$

<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 $M$
- $\times = \text{No convergence after 1,000 iterations}$

<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>Laminar Re=20,000</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>RANS Re=10^6</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>
</tbody>
</table>
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
1. Introduction

2. Discretization and Solvers
   - The Discontinuous Galerkin Method
   - High-Order Mesh Generation
   - 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
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
Proposed Approach

- Use artificial viscosity with a non-linear sensor
- Select parameters based on available resolution
- DG implementation
  - Consistent treatment of high order terms (the CDG scheme)
  - Do not exploit nature of discontinuous approximation
- Artificial viscosity models:
  
  **Shocks**
  - Viscosity model: Laplacian form $\nabla \cdot (\nu_1 \nabla U)$
  - Sensor: Density

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

• 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: Orthonormal Koornwinder basis in each element
Example: RAE2822

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

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

\[
C_L = 0.6144 \quad C_D = 0.0104
\]

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

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

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

\[
C_L = 0.6144 \quad C_D = 0.0104 \quad C_L = 0.6131 \quad C_D = 0.0103
\]
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 ($M = 0.729, \text{Re} = 6.5 \cdot 10^6$)
- Sub-cell resolution of shocks

\[ p = 4 \]
Outline

1 Introduction

2 Discretization and Solvers
   - The Discontinuous Galerkin Method
   - High-Order Mesh Generation
   - 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
ALE Formulation

- 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 $v(t)$

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

can be written in the reference configuration $V$ as

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

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
  \[
  \frac{\partial (gg^{-1}U_X)}{\partial t} \bigg|_X - \nabla_X \cdot F_X = 0, \\
  \frac{\partial \bar{g}}{\partial t} \bigg|_X - \nabla_X \cdot (gG^{-1}v_X) = 0
  \]
Example: Euler Vortex

- Propagate an Euler vortex on a variable domain with

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

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

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

1. Introduction
2. Discretization and Solvers
   - The Discontinuous Galerkin Method
   - High-Order Mesh Generation
   - 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: Locomotion of Free Flapping Body

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

- Interaction between fluid and a membrane
- Hyperelastic Neo-Hookean membrane formulation

R/C dragonfly

Fluid/membrane simulation
Outline

1. Introduction
2. Discretization and Solvers
   - The Discontinuous Galerkin Method
   - High-Order Mesh Generation
   - 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
Governing equations, conservation of linear momentum:

\[ \frac{\partial p}{\partial t} - \nabla \cdot P = \rho_0 b \]

with momentum \( p \) and first Piola-Kirchhoff stress tensor \( P(F) \), deformation tensor \( F = \partial x/\partial X \)

Hyperelastic neo-Hookean model for \( P(F) \)
Lagrangian DG Formulation for Solid Dynamics

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

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

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

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

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

  - Conservative formulation, borrow shock capturing from CFD
  - Expensive, needs curl-free spaces for $F$
The high order discretizations allow for highly stretched elements

Model plate problem, single element across thickness
Volumetric Modeling of Thin Structures

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

- 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, extensions for LES/DES, coupling of DG fluid/structure formulations, applications in flapping flight, aeroacoustics, and transonic/supersonic flows