High Order Discontinuous Galerkin Methods for Fluid and Solid Mechanics

Per-Olof Persson

Department of Mathematics
University of California, Berkeley


High Order Non-Oscillatory Methods for Wave Propagation, Flow and Transport Problems 2009, Trento, Italy (HONOM 2009)

March 30, 2009
1. Introduction

2. Discretization and Solvers
   - Curved 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 Formulations
   - Applications: Thin structures, viscoplasticity

5. Conclusions
1 Introduction

2 Discretization and Solvers
   - Curved 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 Formulations
   - Applications: Thin structures, viscoplasticity

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: Aeroacoustics and K-H Instability

- Inspired by [Munz et al 03] (linearized)
- Nonlinear behavior: Large scale acoustic wave interacts with small scale flow features, leading to vorticity generation
- High-order accuracy *essential* to capture long-range wave propagations and nonlinear interactions
- $p = 7$ (8th order accuracy), 140-by-28 square elements
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
   - Curved 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 Formulations
   - Applications: Thin structures, viscoplasticity

5. Conclusions
Motivation

- Automatic generation of non-inverted curved elements largely an unresolved problem
- In general this is a global problem, affecting many elements except for simple isotropic 2-D meshes
- In [Persson/Peraire 09], we proposed a non-linear solid mechanics approach, where the mesh is considered an elastic deformable solid
Curved Mesh Generation using Solid Mechanics

- The initial, straight-sided mesh corresponds to undeformed solid
- External forces come from the true boundary data
- Solving for a force equilibrium gives the deformed, curved, boundary conforming mesh
- Bottom-up approach can be used to obtain the boundary data

Reference domain, initial configuration
Equilibrium solution, final curved mesh
Tetrahedral Mesh of Cylindrical Component

- Large curved deformations handled without element inversion
Tetrahedral Mesh of Falcon Aircraft

- Real-world mesh with coarse but realistic elements
- Unstructured Delaunay refinement mesh, with highly curved boundary segments
- Many elements would invert with a local element-wise approach
The Discontinuous Galerkin Method

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

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

- Triangulate domain \( \Omega \) into elements \( \kappa \in T_h \)
- Seek approximate solution \( u_h \) in space of element-wise polynomials:

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

- Multiply by test function \( v_h \in V_h^p \) and integrate over element \( \kappa \):

\[ \int_{\kappa} [(u_h)_t + \nabla \cdot F_i(u_h)] v_h \, dx = 0 \]
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 = \( \mathcal{O}(h^{p+1}) \) for smooth solutions
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 degree $p$
- Block-diagonal mass matrix (no overlap between basis functions)
- Block-wise compact stencil – neighboring elements connected
Viscous Discretization

- General approach for second derivatives:
  - Write as system of first order equations [Arnold et al. 02]:
    \[
    u_t + \nabla \cdot \mathcal{F}_i(u) - \nabla \cdot \mathcal{F}_v(u, \sigma) = 0
    \]
    \[
    \sigma - \nabla u = 0
    \]
  - Discretize using DG, choose appropriate numerical fluxes \(\hat{\sigma}, \hat{u}\)

- Various schemes have been proposed:
  - BR2 [Bassi/Rebay 98]: Different lifting operator for each edge, compact connectivities, similar to Interior Penalty (IP)
  - LDG [Cockburn/Shu 98]: Upwind/Downwind, non-compact
  - CDG [Peraire/Persson 07]:
    Modification of LDG for local dependence – sparse and compact
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.
In the CDG scheme, we split the lifting operators into sums of edge-wise lifting operators $r^e(\phi), l^e(q)$, and set

$$\hat{\sigma} = \{\sigma_h^e\} - C_{11}[u_h] + C_{12}[\sigma_h^e]$$

$$\hat{u} = \{u_h\} - C_{12} \cdot [u_h]$$

where $\sigma_h^e = \kappa \nabla_h u_h + \bar{\sigma}_h^e$, with

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

Since only the lifting operator corresponding to the current edge is used, only neighboring elements are connected.
In primal form, the LDG scheme becomes (ignoring bnd terms):

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

The CDG scheme excludes some terms that are indefinite:

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

Non-compact terms are eliminated but the scheme remains stable
Coercivity and boundedness for the CDG scheme same as for LDG, leading to 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

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

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

- Modification of numerical fluxes in LDG scheme
- Excludes non-compact and indefinite terms
- Provably optimal accuracy $O(h^{p+1})$
- Higher stability/accuracy than LDG/BR2
- Sparser than LDG/BR2/IP
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></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CDG</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>LDG</td>
<td>1</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></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CDG</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>LDG</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>
Implicit Time Integration

- Implicit solvers typically required because of CFL restrictions from viscous effects, low Mach numbers, and adaptive/anisotropic grids
- Jacobian matrices are large even at $p = 2$ or $p = 3$, however:
  - They are required for non-trivial preconditioners
  - They are very expensive to recompute
- Therefore, we consider matrix-based Newton-Krylov solvers
- In [Persson/Peraire, SISC’08], we proposed efficient preconditioners with block-ILU(0) and automatic element ordering
- Distributed parallel versions developed in [Persson ’09]
Dual mesh connectivity, with each entry a large complete graph $A_{ij}$

Off-diagonal blocks actually sparser with CDG, but assume dense for simplicity

Size $N$ of submatrices $A_{ij}$ is often $> 100$

Block-based storage format essential for high performance using BLAS routines

For other structures (e.g. elimination matrices), use block-wise compressed column
Preconditioners for Krylov Methods

- Preconditioning required for fast convergence in Krylov methods
- Standard point-wise Jacobi, ILU, etc, ineffective for DG
- Block Jacobi and Gauss Seidel are generally poor:

\[
\tilde{A}_{ij}^J = \begin{cases} 
A_{ij} & \text{if } i = j, \\
0 & \text{if } i \neq j,
\end{cases}
\]

and

\[
\tilde{A}_{ij}^{GS} = \begin{cases} 
A_{ij} & \text{if } i \leq j, \\
0 & \text{if } i > j.
\end{cases}
\]

- Block-ILU(0) algorithm \(\tilde{A}_{ILU} = \tilde{L}\tilde{U}\) effective for good orderings
- Block-ILU(0) postsmoothing for coarse scale correction
  [Persson/Peraire ’08], cheap, general purpose preconditioner
Properties of Gauss Seidel and ILU preconditioners highly dependent on the ordering of the elements.

For *upwinded scalar convection*, “ordering by lines” gives an optimal upper triangular matrix.

But for viscous or multivariate problems, best ordering not clear.

Matrix-approach: Minimize error in the approximations, rather than using physical observations.
Greedy algorithm for element ordering [Persson/Peraire ’08]:

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 $\implies O(n \log n)$ cost

Increased locality: Consider only neighbors for $j'$
MDF ordering makes block-ILU0 with coarse grid correction almost perfect for convection-diffusion

Good element ordering critical for Navier-Stokes as well:

<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>
Convergence – Model Navier-Stokes Problem

× = No convergence after 1,000 iterations, from [Persson/Peraire ’08]

<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></td>
<td></td>
<td></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 Re=1,000</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>
</tbody>
</table>
Convergence – Model Navier-Stokes Problem

× = No convergence after 1,000 iterations, from [Persson/Peraire ’08]

<table>
<thead>
<tr>
<th>Problem</th>
<th>Parameters</th>
<th>Preconditioner/Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>∆t</td>
<td>M</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></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>∞</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>∞</td>
<td>0.01</td>
</tr>
<tr>
<td>RANS Re=10⁶</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>∞</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>∞</td>
<td>0.01</td>
</tr>
</tbody>
</table>
ILU Parallelization – Domain Decomposition

- In parallel, use partition-wise ILUs with MDF ordering
- Partition using the weights $C_{ij} = \|A_{ii}^{-1}A_{ij}\|_F$
- Essentially a “non-overlapping Schwartz preconditioner with incomplete solutions”
- Approaches Jacobi as # partitions $\rightarrow$ # elements
- Good option for many problems – GMRES iterations cheap compared to matrix creation
Elliptic Wing Problem

- Elliptic wing, $\text{Re} = 2000$, $M = 0.3$, $\text{AoA} = 30^\circ$
- 190,000 tetrahedral elements, $p = 3$, about 19 million DOFs
- Time integration by 2-stage, 3rd order accurate DIRK scheme
- Jacobian storage 36GB, no comparison with serial possible
Almost perfect speedup per compute node with 512 processes
Solution time dominated by assembly for small $\Delta t$
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:

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

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: \( \frac{\partial u}{\partial t} + \nabla \cdot \mathbf{F}(u) = \nabla \cdot (\epsilon \nabla u) \)
- Physical: \( \frac{\partial u}{\partial t} + \nabla \cdot \mathbf{F}(u) = \nabla \cdot \mathbf{F}_v(u;\text{Re, Pr}) \)
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 \(h/p\)

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

\[ C_L = 0.6144 \quad C_D = 0.0104 \]
\[ C_L = 0.6131 \quad C_D = 0.0103 \]
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
   - Curved 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 Formulations
   - Applications: Thin structures, viscoplasticity
5. Conclusions
Methods for Deforming Domains

- Many ALE formulations for unstructured meshes
  - 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 *mapping deformation gradient* $G$ and the *mapping velocity* $v_X$ as

$$G = \nabla_X G$$

$$v_X = \left. \frac{\partial G}{\partial t} \right|_X$$

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]
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

$$\frac{\partial (gg^{-1}U_X)}{\partial t} \bigg|_X - \nabla_X \cdot F_X = 0$$

and

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

- Propagate an Euler vortex on a variable domain with

\[
x(\xi, \eta, t) = \xi + 2.0 \sin(2\pi \xi/20) \sin(\pi \eta/7.5) \sin(1.0 \cdot 2\pi t/t_0)
\]
\[
y(\xi, \eta, t) = \eta + 1.5 \sin(2\pi \xi/20) \sin(\pi \eta/7.5) \sin(2.0 \cdot 2\pi t/t_0)
\]

- 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 $p=1$ to $p=5$. The graph plots $L_2$-error against element size $h$. Each line represents a different polynomial degree, with mapped and unmapped schemes indicated by different markers. The y-axis is logarithmic, ranging from $10^{-8}$ to $10^0$, and the x-axis ranges from $10^{-1}$ to $10^0$. The graph demonstrates the convergence behavior for both mapped and unmapped schemes.]
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: \( \text{Re} = 5000, \, M = 0.2 \)
- Forced vertical motion \( r_z(t) = r_0 \sin \omega t \) (at leading edge)
- Moment equation: \( I \ddot{\theta} + C \theta - S \ddot{r}_z(t) + M_{\text{aero}} = 0 \)
  - \( I \) moment of inertia, \( C \) spring stiffness, \( S = m x_c \) static unbalance
  - \( M_{\text{aero}} \) 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
Outline

1. Introduction
2. Discretization and Solvers
   - Curved 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 Formulations
   - Applications: Thin structures, viscoplasticity
5. Conclusions
Many potential benefits with DG for solid dynamics
- Arbitrary orders of accuracy with tetrahedral elements
- Discontinuous approach highly insensitive to locking
  - Almost incompressible materials
  - Thin structures (beams, shells) with highly stretched elements
- Stabilization for convection
  - High-order accurate ALE
- Block-diagonal mass matrix
  - Efficient explicit methods
Lagrangian Formulation

- Map from reference domain \( V \) to physical domain \( v(t) \)

\[
F = \frac{\partial x}{\partial X}, \quad J = \det F, \quad v(X, t) = \frac{\partial x}{\partial t}, \quad p = \rho_0 v
\]

- Conservation of linear momentum:

\[
\frac{d}{dt} \int_V p \, dV = \int_V \rho_0 b \, dV + \int_{\partial V} t \, dA, \quad t = PN
\]

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

with first Piola-Kirchhoff stress tensor \( P(F) \)
Hyperelastic Neo-Hookean Constitutive Model

- Strain Energy Potential
  \[
  \psi(F) = \psi_{iso}(J^{-1/3}F) + \psi_{vol}(J)
  \]

  \[
  \psi_{iso} = \frac{\mu}{2}[tr(J^{-2/3}F : F) - 3], \quad \psi_{vol} = \frac{1}{2}\kappa(J - 1)^2
  \]

- First Piola-Kirchhoff Stress Tensor
  \[
  P = P_{dev} + P_{vol}; \quad P_{dev} = \frac{\partial \psi_{iso}}{\partial F}, \quad P_{vol} = \frac{\partial \psi_{vol}}{\partial F}
  \]

  \[
  P_{vol} = \frac{d\psi_{vol}(J)}{dJ} \frac{\partial J}{\partial F} = \kappa(J - 1) \quad p = \frac{d\psi_{vol}(J)}{dJ} = \kappa(J - 1)
  \]

  \[
  P_{dev} = \mu J^{-2/3}[F - \frac{1}{3}(F : F)F^{-T}]
  \]
Second-Order Formulation

- Straight-forward second-order formulation in terms of material points $x$ and momentum $p$:

$$\frac{\partial x}{\partial t} = \frac{p}{\rho_0}$$

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

- Computationally inexpensive, first equation essentially an ODE

- Full $p$th order polynomial spaces for $x$ and $p$

- Use CDG scheme for stable discretization of second derivatives

- However, conservative treatment of shocks unclear
First-Order Conservation Law Formulation

- Solve for momentum $p$ and deformation gradient $F$
- Time evolution of the deformation gradient:

$$\frac{\partial F}{\partial t} = \nabla \left( \frac{p}{\rho_0} \right) \quad \text{or} \quad \frac{\partial F}{\partial t} = \nabla \cdot \left( \frac{1}{\rho_0} p \otimes I \right)$$

- Write as system of conservation laws:

$$\mathbf{u}_t + \nabla \cdot \mathbf{F}(\mathbf{u}) = \mathbf{S}(\mathbf{u})$$

with

$$\mathbf{u} = \begin{pmatrix} p \\ F \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} P \\ \frac{1}{\rho_0} p \otimes I \end{pmatrix}, \quad \mathbf{S} = \begin{pmatrix} \rho_0 b \\ 0 \end{pmatrix}$$

- Conservative formulation $\implies$ borrow shock capturing from CFD, jump conditions physically meaningful
- However, computationally 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

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

- The high order elements allow for high stretching
- Discontinuous basis functions highly insensitive to locking
- Model plate problem: Uniform loading, small deformations
- High order convergence rates for $p > 1$
Example: Airbag Inflation

- Undeformed configuration: Two thin sheets, attached at the edges (thickness=$10^{-3}$, diagonal=1.2)
- Discretize with stretched tetrahedra and high-order DG elements
- Interior pressure $p_0$ applied at $t = 0$, quasi-static evolution
Example: Airbag Inflation

- 40-by-40 grid, 16000 tetrahedra, polynomial order $p = 3$
- No specialized membrane models required
- Hard to define convergence, but $\max(\Delta z) \rightarrow$ fixed value
Volumetric Modeling of Coupled Beam/Membrane

- No modeling required to couple beams/membranes
- Only a question of generating the stretched tetrahedral meshes
Outline

1 Introduction

2 Discretization and Solvers
   - Curved 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 Formulations
   - Applications: Thin structures, viscoplasticity

5 Conclusions
Conclusions

Important steps toward practical DG solver for realistic problems:

- Non-linear elasticity for curved mesh generation
- 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 formulations for solid dynamics

Current work: Full monolithic DG fluid-structure formulation, improved numerical schemes and algorithms, extensions for LES/DES, applications in flapping flight, aeroacoustics, instabilities in solids, and supersonic flows