Discontinuous Galerkin Methods for Fluid Flows and Implicit Mesh Generation

Per-Olof Persson
Massachusetts Institute of Technology

Stanford FPC Fluid Mechanics Seminar
April 25, 2006
Background

• Objective: Solve large scale problems in fluid mechanics efficiently and accurately on complex geometries

• Requirements:
  – Low dispersion/high accuracy discretization
  – Methods based on unstructured simplex meshes
  – Robust and fully automated solution process
  – Efficient solvers/simple parallelization
Application Areas

- Computational aeroacoustics
  - Direct solution of the compressible Navier-Stokes equations
  - No modeling of acoustic waves (Lighthill, linearized Euler, etc)
  - High accuracy essential to capture turbulent noise sources, propagate acoustic waves, and to ensure stability at low Mach numbers
  - Complex geometries, e.g. in wind noise prediction for vehicles

- Other DNS/LES/DES problems
  - High order methods a clear benefit in DNS/LES region
  - Need flexible meshes e.g. in the boundary layers

- Drag prediction
  - Solve high Reynolds number transonic flow for various geometries
  - Determine drag coefficient accurately
Numerical Schemes

• Various numerical methods:
  – FD: Only structured grids
  – FVM: Only second order accurate
  – HO FVM: Hard to generalize, boundary conditions difficult
  – FEM/SUPG: Not clear how to stabilize for high orders
  – DG: Unstructured, general, but rarely used for practical problems

• Our goal: Make DG methods competitive for real-world problems
  – New low-memory compact viscous discretization
  – Multigrid/ILU preconditioned Krylov solvers
  – Shock capturing with subcell resolution
  – RANS turbulence modeling by Spalart-Allmaras and $k$-$\omega$ models
The Discontinuous Galerkin Method


- 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:

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

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

\[ \int_{\kappa} \left[ (u_h)_t + \nabla \cdot F_i(u_h) \right] \mathbf{v}_h \, dx = 0 \]
The Discontinuous Galerkin Method

- Integrate by parts:

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

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

- Global view: Find \( u_h \in \mathcal{V}_h^p \) such that this weighted residual is zero for all \( v_h \in \mathcal{V}_h^p \)

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

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

\[
(u_h)_t A_\kappa + \int_{\partial \kappa} \mathcal{H}_i(u^+_h, u^-_h, \hat{n}) \, ds = 0
\]

- 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

![Mass Matrix](image1)

![Jacobian](image2)
Nodal Basis and Curved Boundaries

- Nodal basis within each element $\rightarrow$ sparse element connectivities
- Isoparametric treatment of curved elements (not only at boundaries)

Isotropic mesh

W1 wing (Drag Prediction Workshop)

Anisotropic mesh
Example: Inviscid 2-D Flow

- Lower dissipation with high order schemes → Higher accuracy/DOF

24576 DOFs, $p = 1$

7680 DOFs, $p = 4$
Discretization of Viscous Terms

- Write as system of first order equations:
  \[ u_t + \nabla \cdot F_i(u) - \nabla \cdot F_v(u, q) = 0 \]
  \[ q - \nabla u = 0 \]

- Discretize using DG, various formulations and numerical fluxes:
  - The BR1 scheme (Bassi/Rebay 1997):
    Averaging of fluxes, connections between non-neighboring elements
  - The BR2 scheme (Bassi/Rebay 1998):
    Uses lifting operator, connections only between neighbors
  - The LDG scheme (Cockburn/Shu 1998):
    Upwind/Downwind, sparse connectivities, some wide connections
  - The CDG scheme (Peraire/Persson 2006):
    Modification of LDG for local dependence – sparse and compact
Viscous Discretizations – Stencils

- About 50% fewer DOFs with CDG than BR2 (3-D tetrahedra, $p = 4$)
Time Integration and Iterative Solvers

- Discretization in space using DG gives system of ODEs
  \[
  M \dot{U} = R(U) \text{ with Jacobian } K = dR/dU
  \]

- Implicit time integration requires solving
  \[
  A = \alpha_0 M - \Delta t K
  \]

- Block-Jacobi efficient solver for simple problems (inviscid flow, isotropic)

- In general we need Krylov solvers with sophisticated preconditioning
Multiscale Preconditioner with Block-ILU(0)

- ILU(0): Gaussian elimination, ignoring all new matrix elements (fill-in)
- A remarkably efficient preconditioner: Apply a coarse grid correction followed by post-smoothing with Block-ILU(0) (Persson/Peraire 2006)
- In particular it makes convergence almost independent of Mach number
Example: Viscous Flow

NACA 0012, Re=1000, (Mesh)

Sphere, Re=1000 (Mach & Velocity)

Cylinder, Re=2000 (Entropy)
Shock Capturing with Artificial Viscosity

- Need stabilization for shock problems with $p > 0$
- Common approaches: Slope limiting, filtering – shock width $\delta_s \sim 2h$
- Use artificial viscosity with a non-linear sensor: (Persson/Peraire 2006)
  - Select parameters based on available resolution
  - Consistent treatment of high order terms (CDG)
  - Do not exploit nature of discontinuous approximation
  - Shock width $\delta_s \sim 4h/p$ (subgrid resolution, $p$-convergence)
  - Continuous sensor and viscosity – Newton solver

\[ 4h/p \ll 2h \text{ for high } p \]
Shock Sensor

- Determine regularity of the solution from rate of decay of frequencies

- Periodic Fourier case: \( f(x) = \sum_{k=-\infty}^{\infty} g_k e^{ikx} \)
  If \( f(x) \) has \( m \) continuous derivatives \( \rightarrow |g_k| \sim k^{-(m+1)} \)

- Simplex element \( e \): Expand solution in orthonormal Koornwinder basis \( \psi_i \):
  \[
  u = \sum_{i=1}^{N(p)} u_i \psi_i, \quad \hat{u} = \sum_{i=1}^{N(p-1)} u_i \psi_i, \quad s_e = \log_{10} \left( \frac{(u - \hat{u}, u - \hat{u})_e}{(u, u)_e} \right)
  \]

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

where \( \varepsilon_0 \sim h/p, s_0 \sim 1/p^4 \) and \( \kappa \) empirical
Scalar Results – Burgers’ Equation ($\rho = 10$)

Initial condition $u(x, 0) = \frac{1}{2} + \sin 2\pi x$

T=0.25

T=0.50

Artificial Viscosity
Euler Equations – Artificial Viscosity Models

- Laplacian: \( \frac{\partial u}{\partial t} + \nabla \cdot \mathbf{F}(u) = \nabla \cdot (\epsilon \nabla u) \)

- Physical: \( \frac{\partial u}{\partial t} + \nabla \cdot \mathbf{F}(u) = \nabla \cdot \mathbf{F}_v(u; \text{Re}, \text{Pr}) \)
Transonic – Laplacian model/Density sensor

- NACA 0012, 1.5° attack angle
- $M = 0.8$ and $p = 4$
- Implicit steady-state solution with Newton solver
Supersonic – Laplacian model/Density sensor

- NACA 0012, 1.5° attack angle
- $M = 1.5$ and $p = 4$
- Implicit steady-state solution with Newton solver
Supersonic – Physical model/Entropy sensor

Mach

Entropy

Sensor

Mach 2

Mach 5
RANS Turbulence Modeling

- Recast the Spalart-Allmaras one-equation turbulence model into conservative form: (Persson/Nguyen/Peraire 2006)

\[
\frac{\partial}{\partial t} (\rho \tilde{\nu}) + \frac{\partial}{\partial x_j} (\rho u_j \tilde{\nu}) - \nabla \cdot (f_d \nabla (\rho \tilde{\nu})) = S_R
\]

\[
S_R = c_{b1} \rho \tilde{S} \tilde{\nu} + \frac{c_{b2}}{\sigma} \rho (\nabla \tilde{\nu})^2 - c_{w1} \rho f_w \left( \frac{\tilde{\nu}}{d} \right)^2 - \nabla \rho \cdot \left( \frac{\nu + \tilde{\nu}}{\sigma} \nabla \tilde{\nu} \right)
\]

with modified coefficient

\[
f_d = \left[ (f_{d1})^6 + (f_{d2})^6 \right]^{1/6}
\]

\[
f_{d1} = \frac{\nu + \tilde{\nu}}{\sigma}
\]

\[
f_{d2} = c_d \nu \left( \frac{\Delta_{\text{min}}/p}{0.16 \text{Re}^{-1/7} + d} \right)^2,
\]

- Modification introduced for \( p \geq 1 \) to avoid negative eddy viscosity at end of boundary layer
Turbulent Flow over NACA 0012 Airfoil

- Spalart-Allmaras turbulence modeling, $M = 0.3$, $Re = 1.86 \cdot 10^6$

- $p$-convergence and good agreement with experimental data
• Spalart-Allmaras turbulence modeling, $M = 0.2, \text{Re} = 5 \cdot 10^5, p = 2$

• LES in 2-D not physically meaningful, but comparison shows how URANS smears out complex flow features
Summary DG Methods

- Contributions:
  - Low-cost compact viscous discretization
  - Highly efficient multigrid/ILU preconditioner
  - Shock capturing with subcell resolution using artificial viscosity
  - DG discretization of modified Spalart-Allmaras turbulence model

- Current status:
  - Flexible, lightweight, and highly efficient n-D code fully developed
  - Implicit and explicit solvers, various Krylov solvers and preconditioners
  - Handles large 2-D problems and moderate size 3-D problems in serial

- Current work:
  - Parallelization of code, implementation of LES/DES capabilities, solution of large scale 3-D problems
Mesh Generation

- Given a geometry, determine node points and element connectivity
- Resolve the geometry and high element qualities, but few elements
- Applications: Numerical solution of PDEs (FEM, FVM, BEM), interpolation, computer graphics, visualization
- Popular algorithms: Delaunay refinement, Advancing front, Octree
Geometry Representations

Explicit Geometry

- Parameterized boundaries

\[(x, y) = (x(s), y(s))\]

Implicit Geometry

- Boundaries given by zero level set

\[f(x, y) = 0\]

\[f(x, y) < 0\]

\[f(x, y) > 0\]
Explicit vs Implicit

- Most CAD programs represent geometries explicitly (NURBS and B-rep)
- Most mesh generators require explicit expressions for boundaries (Delaunay refinement, Advancing front)
- Increasing interest for implicit geometries:
  - Easy to implement: Set operations, offsets, blendings, etc
  - Automatic “geometry repair” from resolution of implicit function
  - Extend naturally to higher dimensions
  - Level set methods for evolving interfaces
  - Image based geometry representations (MRI/CT scans)
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 $\phi$
   - Update element connectivities
Internal Forces

For each *interior* node:

\[ \sum_{i} F_i = 0 \]

Repulsive forces depending on edge length \( \ell \) and equilibrium length \( \ell_0 \):

\[ |F| = \begin{cases} 
  k(\ell_0 - \ell) & \text{if } \ell < \ell_0, \\
  0 & \text{if } \ell \geq \ell_0. 
\end{cases} \]

Get expanding mesh by choosing \( \ell_0 \) larger than desired length \( h \).
For each boundary node:
\[ \sum_{i} F_i + R = 0 \]

Reaction force \( R \):
- Orthogonal to boundary
- Keeps node along boundary
Node Movement and Connectivity Updates

- Move nodes \( p \) to find force equilibrium:
  \[
  p_{n+1} = p_n + \Delta t F(p_n)
  \]
- Project boundary nodes to \( \phi(p) = 0 \)
- Elements deform, change connectivity based on element quality or in-circle condition (Delaunay)
Mesh Size Functions

- Need good distribution of mesh element sizes $h(\mathbf{x})$ based on curvatures, feature sizes, and external size constraints.
- Also: $|\nabla h(\mathbf{x})| \leq g$ to limit ratio $G = g + 1$ of neighboring element sizes.
- Optimal gradient limited $h(\mathbf{x})$ steady-state solution to: (Persson 2005)

$$\frac{\partial h}{\partial t} + |\nabla h| = \min(|\nabla h|, g),$$
$$h(t = 0, \mathbf{x}) = h_0(\mathbf{x}).$$
Mesh Size Functions – 3-D Examples

Mesh Size Function $h(x)$

Mesh Based on $h(x)$
Mesh Size Functions – 3-D Examples
Moving Meshes

- Iterative formulation well-suited for geometries with moving boundaries
- Mesh from previous time step good initial condition, a new mesh obtained by a few additional iterations
- Even better if smooth embedding of boundary velocity – move all nodes
- Density control if area changes
Level Sets and Finite Elements

- Level Set Methods superior for interface propagation:
  - Numerical stability and entropy satisfying solutions
  - Topology changes easily handled, in particular in three dimensions

- Unstructured meshes better for the physical problems:
  - Better handling of boundary conditions
  - Natural mesh grading and adaptation

- Proposal: Combine Level Sets and Finite Elements (Persson 2005)
  - Use level sets on background grid for interface
  - Mesh using our iterations, and solve physical problem using FEM
  - Interpolate boundary velocity to background grid
Consider eigenvalue problem

\[-\Delta u = \lambda \rho(x) u, \quad x \in \Omega\]
\[u = 0, \quad x \in \partial\Omega.\]

with

\[\rho(x) = \begin{cases} 
\rho_1 & \text{for } x \notin S \\
\rho_2 & \text{for } x \in S.
\end{cases}\]

Solve the optimization

\[
\min_{S} \lambda_1 \text{ or } \lambda_2 \text{ subject to } \|S\| = K.
\]
Structural Design Improvement

- Linear elastostatics, minimize compliance
  \[ \int_{\partial\Omega} g \cdot u \, ds = \int_{\Omega} A\varepsilon(u) \cdot \varepsilon(u) \, dx \text{ subject to } ||\Omega|| = K. \]

- Level set approaches by Sethian and Wiegmann (Immersed interface method) and Allaire et al (Ersatz material, Young’s modulus \(\varepsilon\))

- Level sets for geometry, unstructured FEM for elasticity (Persson 2005):
Fluid Dynamics with Moving Boundaries

- Incompressible N-S with moving boundaries
- Lagrangian node movement for convective term
- Improve poor mesh elements using new force equilibrium
- Interpolate new velocity field (or $L_2$-projections, ALE, Space-Time, etc)
- Project for incompressibility
Meshing Images

- Images are implicit representations of objects
- Segment image, find distance function/size function, and create mesh
Meshing Images - Medical Applications

- Method extends to three dimensions (MRI/CT scans)
- Tetrahedral volume mesh of Iliac bone (from CT scan)
Summary Implicit Mesh Generation

- New iterative mesh generator for implicit geometries
- Simple algorithm, high element qualities, works in any dimension
- Automatic computation of size functions, PDE-based gradient limiting
- Combine FEM and level set methods for moving interface problems in shape optimization, structural instabilities, fluid dynamics, etc
- Mesh generation for images (in any dimension)
- Future work: Space-time meshes, sliver removal, performance and robustness improvements, quadrilateral/hexahedral meshes, anisotropic gradient limiting, more applications, ...