An Introduction to Discontinuous Galerkin Methods for Compressible Flows

David L. Darmofal
Aerospace Computational Design Lab
Massachusetts Institute of Technology

November 5, 2004
Overview

- Motivation: Why develop another CFD algorithm?
- Finite volume methods for hyperbolic conservation laws
- Discontinuous Galerkin (DG) for hyperbolic conservation laws
- DG for elliptic problems
- $p$-multigrid for higher-order DG discretizations
- Conclusions and future work
Motivation for higher order

- State of CFD in applied aerodynamics
  - Finite-volume with at best second order accuracy
  - Questions exist whether current discretizations are capable of achieving desired accuracy levels in practical time
- Decrease computational time and gridding requirements by increasing solution order

\[
\log T = wd\left( -\frac{1}{p} \log E + \log p \right) - \log F + \text{const}
\]

- \( T \) = time to solution
- \( p \) = discretization order
- \( w \) = solution complexity
- \( d \) = dimension of problem
- \( E \) = desired error level (\( E << 1 \))
- \( F \) = computational speed
Project-X Goal

Project X Team Goal:

- To improve the aerothermal design process for complex 3D configurations by significantly reducing the time from geometry to solution at engineering-required accuracy using high-order adaptive methods
Previous Work

- Extensive work on DG for hyperbolic equations
  - Bassi and Rebay (1997)
  - Cockburn and Shu (1998, 2001)
  - Karniadakis et al. (1998, 1999)

- More recently work begun on elliptic equations
  - Cockburn and Shu (1998, 2001)
  - Baumann and Oden (1997)
  - Brezzi et al. (1997)

- Only Bassi and Rebay have published RANS results (1997, 2003)
Integral Form of Hyperbolic Conservation Laws

Apply integral conservation law on triangle 0:

\[
\frac{d}{dt} \int_{A_0} \mathbf{u} \, d\mathbf{x} + \sum_{k=1}^{3} \int_{0k} \mathbf{F}_i(\mathbf{u}) \cdot \mathbf{n} \, ds = 0
\]

For Euler equations:

\[
\mathbf{u} = (\rho, \rho u, \rho v, \rho E)^T
\]

\[
\mathbf{F}_i = (\mathbf{F}_i^x, \mathbf{F}_i^y)^T
\]

\[
\mathbf{F}_i^x = (\rho u, \rho u^2 + p, \rho u v, \rho u H)^T
\]

\[
\mathbf{F}_i^y = (\rho v, \rho u v, \rho v^2 + p, \rho v H)^T
\]
First-order Accurate Finite Volume

In each triangle, assume $u$ is constant.

Apply conservation law on triangle:

$$\frac{d}{dt} u_0 A_0 + \sum_{k=1}^{3} \int_{0k}^{3} \mathcal{H}_i(u_0, u_k, \hat{n}_{0k}) \, ds = 0$$

$\mathcal{H}_i(u_L, u_R, \hat{n}_{LR})$ is flux function that determines inviscid flux in $\hat{n}_{LR}$ direction from left and right states, $u_L$ and $u_R$.

Example flux functions: Godunov, Roe, Osher, Van Leer, Lax-Friedrichs, etc.

This discretization has a solution error which is $O(h)$ where $h$ is mesh size.
In each triangle, reconstruct a linear solution, \( \tilde{u} \), using neighboring averages:

\[
\tilde{u}_0 \equiv u_0 + (x - x_0) \cdot \nabla u_0, \\
\nabla u_0 \equiv \nabla u_0 (u_0, u_1, u_2, u_3).
\]

Apply conservation law on triangle:

\[
\frac{du_0}{dt} A_0 + \sum_{k=1}^{3} \int_{0k} H_i(\tilde{u}_0, \tilde{u}_k, \hat{n}_{0k}) \, ds = 0
\]

On smooth meshes and flows, solution error is \( O(h^2) \).
Pros/Cons of Higher-order Finite Volume

- Increased accuracy on given mesh without additional degrees of freedom

- Difficulty in achieving higher-order on unstructured meshes and near boundaries

- Stabilizing multi-stage methods necessary for local iterative schemes

- Matrix fill-in increased resulting in high-memory requirements
Instability of Local Iterative Methods

Consider steady state problem and define discrete residual for cell $j$,

$$R_j(u) \equiv \sum_{k=1}^{3} \int_{j_k} \mathcal{H}_i(\tilde{u}_j, \tilde{u}_k, \tilde{n}_{jk}) \, ds = 0.$$

A Jacobi iterative method to solve this problem is,

$$u_{j}^{n+1} = u_{j}^{n} - \omega \left( \frac{\partial R_j}{\partial u_j} \right)^{-1} R_j(u).$$

For any finite $\omega$, Jacobi is unstable for higher-order. One solution is a multi-stage method,

$$\hat{u}_j = u_j^n - \hat{\omega} \left( \frac{\partial R_j}{\partial u_j} \right)^{-1} R_j(u^n) \quad \leftarrow \text{Requires two residual evaluations.}$$

$$u_{j}^{n+1} = u_{j}^{n} - \omega \left( \frac{\partial R_j}{\partial u_j} \right)^{-1} R_j(\hat{u})$$
Matrix Fill for Higher-order Finite Volume

First-order

Second-order

Third-order

nz = 115
nz = 355
nz = 601
Discontinuous Polynomial Basis

- Triangulate domain $\Omega$ into non-overlapping elements $\kappa \in T_h$
- Define function space: Element-wise discontinuous polynomials of degree $p$

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

Example of One-Dimensional Bases

$p = 0$ basis

1 DOF/element

$p = 1$ basis

2 DOF/element
DG for Hyperbolic Conservation Laws: Derivation

Start from strong form of governing equations:

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

Look for a solution \( u_h \in V_h^p \).

Multiply governing equation by weight function \( v_h \in V_h^p \) and integrate over element \( \kappa \in T_h \):

\[
\int_\kappa v_h^T [(u_h)_t + \nabla \cdot F_i] \, dx = 0.
\]

Integrate second term by parts (assume interior element):

\[
\int_\kappa v_h^T (u_h)_t \, dx - \int_\kappa \nabla v_h^T \cdot F_i \, dx + \int_{\partial \kappa} v_h^T H_i(u_h^+, u_h^-, \hat{n}) \, ds = 0.
\]
Relationship of DG to other methods

- Recall DG weighted residual (Reed & Hill, 1973):

\[
\int_{\kappa} \mathbf{v}_h^T (\mathbf{u}_h)_t \, d\mathbf{x} - \int_{\kappa} \nabla \mathbf{v}_h^T \cdot \mathbf{F}_i \, d\mathbf{x} + \int_{\partial \kappa} \mathbf{v}_h^+ \mathbf{H}_i (\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n}) \, ds = 0.
\]

- For \( p = 0 \) solution, this reduces to:

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

- Thus, \( p = 0 \) DG is identical to first-order finite volume.

- For \( p > 0 \), DG can be interpreted as a moment method.

- Moment methods for hyperbolic problems were first suggested by Van Leer (1977) and then developed for the Euler equations by Allmaras (1987, 1989) and later Holt (1992).
DG discretization: Global view

- Find \( u_h \in V_h^p \) such that \( \forall v_h \in V_h^p \),

\[
\sum_{\kappa \in T_h} \left\{ \int_{\kappa} v_h^T (u_h)_t \, dx - \int_{\kappa} \nabla v_h^T \cdot F_i \, dx \right\} \\
+ \int_{\Gamma_i} v_h^+ T \mathcal{H}_i(u_h^+, u_h^-, \hat{n}) \, ds + \int_{\partial \Omega} v_h^+ T \mathcal{H}_i^b(u_h^+, u_h^b, \hat{n}) \, ds = 0.
\]

- Boundary conditions enforced weakly through \( \mathcal{H}_i^b(u_h^+, u_h^b, \hat{n}) \)
  where \( u_h^b \) is determined from desired boundary conditions and outgoing characteristics.

- For smooth problems, the error of this scheme is expected to be \( O(h^{p+1}) \).
Pros/Cons of Higher-order DG

- Increased accuracy on given mesh requires additional degrees of freedom
+ Higher-order accuracy not hampered on unstructured meshes nor near boundaries
+ Local iterative methods are stable
+ Matrix fill-in maintains block sparsity of $p = 0$

\[
\int_{\kappa} \mathbf{v}_h^T (\mathbf{u}_h)_t \, dx - \int_{\kappa} \nabla \mathbf{v}_h^T \cdot \mathbf{F}_i \, dx + \int_{\partial\kappa} \mathbf{v}_h^+ \mathcal{H}_i (\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n}) \, ds = 0.
\]
An elemental block Jacobi iterative method to solve this problem is,

\[ u_j^{n+1} = u_j^n - \omega (\partial R_j / \partial u_j)^{-1} R_j(u) \]

where \( \partial R_j / \partial u_j \) is the diagonal block for the element \( j \).

For \( 0 < \omega < 1 \), elemental block Jacobi is stable independent of \( p \).
Matrix Fill for Higher-order DG

First-order ($p = 0$)          Second-order ($p = 1$)          Third-order ($p = 2$)

\[ \text{nz} = 115 \]

\[ \text{nz} = 1035 \]

\[ \text{nz} = 4140 \]
Navier-Stokes Equations

- Navier-Stokes Equations: $u_t + \nabla \cdot F_i(u) - \nabla \cdot F_v(u, \nabla u) = 0$

- $F_v = A_v \nabla u = (F^x_v, F^y_v)$ is the viscous flux vector

$$F^x_v = \begin{pmatrix} 0 \\ \frac{2}{3} \mu (2 \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}) \\ \mu (\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}) \\ \frac{2}{3} \mu (2 \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}) u + \mu (\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}) v + \kappa \frac{\partial T}{\partial x} \end{pmatrix},$$

$$F^y_v = \begin{pmatrix} 0 \\ \mu (\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}) \\ \frac{2}{3} \mu (2 \frac{\partial v}{\partial y} - \frac{\partial u}{\partial x}) \\ \frac{2}{3} \mu (2 \frac{\partial v}{\partial y} - \frac{\partial u}{\partial x}) v + \mu (\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}) u + \kappa \frac{\partial T}{\partial y} \end{pmatrix},$$
Model problem for viscous terms of N-S: 1-D, scalar Poisson’s equation

\[-u_{xx} = f \quad \text{on} \quad [-1, 1]\]

Proceed as for Euler:
- Triangulate domain into non-overlapping elements \( \kappa \in T_h \)
- Define solution and test function space \( \mathcal{V}_h^p \)

Discrete formulation: Find \( u_h \in \mathcal{V}_h^p \) such that \( \forall v_h \in \mathcal{V}_h^p \),

\[
\sum_{\kappa \in T_h} \left\{ - [v_h \widehat{u_x}]_{x_{\kappa-1/2}}^{x_{\kappa+1/2}} + \int_{\kappa} (v_h)_x (u_h)_x \, dx \right\} = \sum_{\kappa \in T_h} \left\{ \int_{\kappa} v_h f \, dx \right\}
\]

Need to define \( \widehat{u_x} \)
DG for Elliptic Operators: First Attempt

- No upwinding mechanism ⇒ choose central flux

\[ \widehat{u}_x = \frac{1}{2} \left( (u_h)_x^L + (u_h)_x^R \right) \]

- Discrete formulation becomes: Find \( u_h \in \mathcal{V}_h \) such that \( \forall v_h \in \mathcal{V}_h \),

\[
\sum_{\kappa \in T_h} \left\{ - \left[ \frac{1}{2} v_h ((u_h)_x^L + (u_h)_x^R) \right]_{x_{\kappa - 1/2}}^{x_{\kappa + 1/2}} + \int_{\kappa} (v_h)_x (u_h)_x \, dx \right\} = \sum_{\kappa \in T_h} \left\{ \int_{\kappa} v_h \, f \, dx \right\}
\]

- PROBLEM: Scheme is inconsistent!
Examine Laplace’s equation with homogeneous Dirichlet BCs

\[-u_{xx} = 0 \text{ on } [-1, 1]\]

\[u(-1) = u(1) = 0\]

Exact solution: \(u(x) = 0\)

If \((u_h)_x = 0\) everywhere, discrete equations satisfied exactly regardless of magnitude of \(u_h\)
First Order System Approach

- Introduce new variable, \( q = u_x \), such that
  \[
  -q_x = f \\
  q - u_x = 0
  \]

- Discrete formulation: Find \( u_h \in \mathcal{V}_h^p \) and \( q_h \in \mathcal{V}_h^p \) such that \( \forall v_h \in \mathcal{V}_h^p \) and \( \forall \tau_h \in \mathcal{V}_h^p \),
  \[
  \sum_{\kappa \in T_h} \left\{ -\left[ v_h \hat{q} \right]_{x_{\kappa-1/2}}^{x_{\kappa+1/2}} + \int_\kappa (v_h)_x q_h \, dx \right\} - \sum_{\kappa \in T_h} \left\{ \int_\kappa v_h f \, dx \right\} = 0
  \]
  \[
  + \sum_{\kappa \in T_h} \left\{ \int_\kappa \tau_h q_h \, dx + \int_\kappa (\tau_h)_x u_h \, dx - \left[ \tau_h \hat{u} \right]_{x_{\kappa-1/2}}^{x_{\kappa+1/2}} \right\} = 0
  \]

- Need to choose \( \hat{q} \) and \( \hat{u} \)
No upwinding mechanism ⇒ choose central fluxes

\[ \hat{u} = \frac{1}{2} (u_h^L + u_h^R); \quad \hat{q} = \frac{1}{2} (q_h^L + q_h^R) \]

Sub-optimal order of accuracy for odd \( p \)

Stencil no longer compact
**BR1 Scheme**

- Define jump, $[\cdot]$, and average, $\{\cdot\}$, operators:
  
  $$
  [s] = s^L - s^R \quad \text{and} \quad \{s\} = \frac{1}{2}(s^L + s^R)
  $$

- Central fluxes become
  
  $$
  \widehat{u} = \{u_h\}; \quad \widehat{q} = \{(u_h)_x\} - \{\delta\}
  $$

- $\delta$ given by following problem: Find $\delta \in \mathcal{V}_h^p$ such that $\forall \tau_h \in \mathcal{V}_h^p$,
  
  $$
  \sum_{\kappa \in T_h} \int_{\kappa} \tau_h \delta \, dx = \sum_n \left[ [u_h] \{\tau_h\} \right]
  $$
BR1 Scheme

- BR1 becomes: Find $u_h \in \mathcal{V}_h^p$ and such that $\forall v_h \in \mathcal{V}_h^p$,

\[
\sum_{\kappa \in T_h} \int_{\kappa} (v_h)_x (u_h)_x \, dx
\]

\[
- \sum_n \left[ \|u_h\| \{(v_h)_x\} + \|v_h\| \{(u_h)_x\} - \{\delta\} \right] = \sum_{\kappa \in T_h} \int_{\kappa} v_h f \, dx
\]

- Stencil extended by $\delta$ dependence on $u_h$
BR2 Scheme

- **Goal:** Eliminate extended stencil
- **Approach:** Modify auxiliary variable, $\delta$, previously defined by:

$$\sum_{\kappa \in T_h} \int_{\kappa} \tau_h \delta \, dx = \sum_n \left[ [u_h] \{\tau_h\} \right]$$

- New variable, $\delta_f$, given by: Find $\delta_f \in \mathcal{V}_h^p$ such that $\forall \tau_h \in \mathcal{V}_h^p$,

$$\int_{\kappa^{L/R}} \tau_h \delta_f^{L/R} \, dx = \left[ [u_h] \{\tau_h\}^{L/R} \right]_{n_f}$$

- New fluxes have same form as before

$$\hat{u} = \{u_h\}; \quad \hat{q} = \{(u_h)_x\} - \eta_f \{\delta_f\}$$
BR2 Scheme

- Replacing \( \{\delta\} \) in BR1 by \( \eta_f \{\delta_f\} \) gives BR2
- For proper choice of \( \eta_f \), can prove optimal order of accuracy
- Stencil is compact
Iterative Solver

- Use work by Fidkowski and Darmofal (2004) on solution of DG discretization of Euler equations

- Nonlinear discrete equations can be written

\[ \mathbf{R}(\mathbf{u}_h) = 0 \]

- Use a preconditioned iterative scheme

\[ \mathbf{u}_h^{n+1} = \mathbf{u}_h^n - \mathbf{P}^{-1} \mathbf{R}(\mathbf{u}_h^n) \]

- Preconditioner
  - Block-element smoothing
    - \( \mathbf{P} = \mathbf{M}_{\text{block}} \Rightarrow \) Block diagonal of the Jacobian
  - Line-element smoothing
    - \( \mathbf{P} = \mathbf{M}_{\text{line}} \Rightarrow \) Block tridiagonal systems from Jacobian
Line Solver

- Motivation: Transport of information in Navier-Stokes equations characterized by convection-diffusion like phenomena
  - Inviscid regions: Information follows characteristic directions set by convection
  - Viscous regions: Diffusion effects can be as strong or stronger than convection

- Procedure:
  - Construct lines of elements based on measure of influence
  - Build and invert $M_{\text{line}}$, which is a set of block tridiagonal systems from the full Jacobian
Example Lines and Performance

![Graph showing line solver iterations vs. ||R||_1](image)

- **Convection-Diffusion Lines**
- **Convection Only Lines**

ACDL Seminar 31/38
$p$-Multigrid: Motivation

- Observation: Smoothers are inefficient at eliminating low frequency error modes on fine level
- $h$-Multigrid
  - Spatially coarse grid used to correct solution on fine grid
  - Grid coarsening is complex on unstructured meshes
- $p$-Multigrid (Ronquist & Patera, Helenbrook et al., Fidkowski & Darmofal)
  - Low order ($p - 1$) approximation used to correct high order ($p$) solution
  - Natural implementation in DG FEM discretization on unstructured meshes
$p$-Multigrid: Full Multigrid

- Full Approximation Scheme (FAS) used
- Line solver used as smoother

Diagram showing the multigrid levels $p=0, p=1, p=2, p=3$.
NACA 0012 Test Case

\( M = 0.5, \ Re = 5000, \ \alpha = 0 \)

Grids are from Swanson at NASA Langley

2112 element mesh  Mach contours
Drag Error Convergence

\[ 10000 \cdot |c_d - c_{d,e}| \]

\( p = 0 \) \hspace{1cm} 0.85

\( p = 1 \) \hspace{1cm} 2.42

\( p = 2 \) \hspace{1cm} 6.88

\( p = 3 \) \hspace{1cm} 3.99

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22

\( p = 3 \) \hspace{1cm} 3.22
CPU Timing

![Graph showing CPU Timing with different curves for p = 1, p = 2, and p = 3. The x-axis represents CPU Time (s) ranging from 10^2 to 10^4, and the y-axis represents 10000*|c_d - c_{d,e}| ranging from 10^{-3} to 10^2. The curves show the relationship between CPU Time and the difference in cost.]
Future Work

- Turbulence modeling (Todd)
- Shocks (Jean-Baptiste & Garrett)
- Adaptation (Chris & Mike)
- Optimization (James)
- Many others

Thanks to the entire Project-X crew... this is their work!