High-Order Numerical Methods for Conservation Laws

Per-Olof Persson

Department of Mathematics, University of California, Berkeley
Mathematics Department, Lawrence Berkeley National Laboratory

UC Berkeley Math Department Colloquium

September 15, 2011
Outline

1. Introduction and Motivation
2. High-Order Discontinuous Galerkin (DG) Methods
   - Application: Large Eddy Simulation of Flow over Airfoil
   - Application: Micromechanical Resonators
3. Reducing the cost: The Line-Based DG Method
4. ALE Formulation for Deforming Domains
   - Application: Vertical Axis Wind Turbines
   - Application: Flapping Flight of Bat
Outline

1 Introduction and Motivation

2 High-Order Discontinuous Galerkin (DG) Methods
   - Application: Large Eddy Simulation of Flow over Airfoil
   - Application: Micromechanical Resonators

3 Reducing the cost: The Line-Based DG Method

4 ALE Formulation for Deforming Domains
   - Application: Vertical Axis Wind Turbines
   - Application: Flapping Flight of Bat
Motivation

- Need for higher fidelity predictions in computational mechanics
  - Turbulent flows, fluid/structure interaction, flapping flight
  - Wave propagation, multiscale phenomena, non-linear interactions
- Widely believed that high-order methods will become the standard in future generations of simulation software
- Challenge: Develop *robust* and *efficient* high-order solvers, that are based on fully unstructured meshes
Why Unstructured Meshes?

- Complex *geometries* need flexible element topologies
- Complex *solution fields* need spatially variable resolution
- Fully automated mesh generators for CAD geometries are based on unstructured simplex elements
- Real-world simulation software dominated by unstructured mesh discretization schemes
Why high-order accurate methods?

- Scalar convection equation $u_t + u_x = 0$
- High-order gives superior performance for equal resolution
Outline

1. Introduction and Motivation

2. High-Order Discontinuous Galerkin (DG) Methods
   - Application: Large Eddy Simulation of Flow over Airfoil
   - Application: Micromechanical Resonators

3. Reducing the cost: The Line-Based DG Method

4. ALE Formulation for Deforming Domains
   - Application: Vertical Axis Wind Turbines
   - Application: Flapping Flight of Bat
The Finite Difference Method (FDM)

- Consider linear convection: \( u_t + u_x = 0 \) for \( x \in [0, 1] \), \( u(0) = u(1) \)
- Approximate \( u_x \) point-wise using difference formulas:

\[
\frac{d}{dx} u(x_n) \approx \frac{u_{n+1} - u_{n-1}}{2\Delta x}
\]

or high-order:

\[
\frac{d}{dx} u(x_n) \approx \frac{u_{n+2} - 8u_{n+1} + 8u_{n-1} - u_{n-1}}{12\Delta x}
\]

or one-sided (e.g. for stability, “upwinding”):

\[
\frac{d}{dx} u(x_n) \approx \frac{3u_n - 16u_{n-1} + 36u_{n-2} - 48u_{n-3} + 25u_{n-4}}{25\Delta x}
\]

- Simple, efficient, flexible
- Needs *structured neighborhood* of nodes – hard to generalize to unstructured grids in 2-D and 3-D
The Finite Element Method (FEM)

- Discretize domain into *elements* (intervals)
- Seek approximate solution in space of piecewise polynomials $X_h$
- Impose equation weakly: Seek $\hat{u} \in \hat{X}$ such that for all $v \in \hat{X}$:

\[ \int_0^1 (\hat{u}_t + \hat{u}_x)v \, dx = \int_0^1 \hat{u}_t v \, dx + \int_0^1 \hat{u}_x v \, dx = \int_0^1 \hat{u}_t v \, dx - \int_0^1 \hat{uv}_x \, dx = 0 \]

- Leads to semi-discrete system $Mu_t + Ku = 0$, with element-wise local $M, K$ matrices
- $M^{-1}$ dense $\implies$ Explicit methods for $u_t = -M^{-1}Ku$ not practical
- Also, unclear how to stabilize by upwinding (but other techniques exist, such as Streamline Upwind Petrov-Galerkin)
The Discontinuous Galerkin Method

- Do not enforce continuity – allow “jumps” between elements

- Galerkin formulation for single element \( \kappa = [0, h] \): For all \( v \in P^p(\kappa) \),

\[
\int_0^h (\hat{u}_t + \hat{u}_x) v \, dx = \int_0^h \hat{u}_t v \, dx + \int_0^h \hat{u}_x v \, dx \\
= \int_0^h \hat{u}_t v \, dx - \int_0^h \hat{u} v_x \, dx + U(u^+, u_p) v(h) - U(u_0, u^-) v(0)
\]

- Numerical flux function \( U(u_R, u_L) \) allows for stabilization by high-order upwinding, e.g. \( U(u_R, u_L) = u_L \)
The Discontinuous Galerkin Method

- The DG formulation leads to linear system of equations:

\[ Mu_t + Ku + \left( -u^- \ 0 \ \ldots \ 0 \ u_p \right)^T = 0 \]

- For example, with \( p = 2 \):

\[
\begin{align*}
\mathbf{u}_t &= -M^{-1}K\mathbf{u} - M^{-1} \left( -u^- \ 0 \ u_2 \right)^T \\
&= \frac{1}{h} \begin{pmatrix}
-6 & -4 & 1 \\
2.5 & 0 & -1 \\
-4 & 4 & -3
\end{pmatrix}
\begin{pmatrix}
u_0 \\
u_1 \\
u_2
\end{pmatrix} + \frac{1}{h} \begin{pmatrix}
9 \\
-1.5 \\
3
\end{pmatrix} u^-
\end{align*}
\]

- Element-wise local FD-type stencil

- Stabilized, “upwinded” through \( u^- \)

- Extends naturally to other PDEs, N-D, unstructured meshes
The DG Method – General case

- Consider non-linear hyperbolic system in conservative form:
  \[ u_t + \nabla \cdot F(u) = 0 \]

- Triangulate domain \( \Omega \) into elements \( \kappa \in T_h \)

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

with numerical flux function \( F(u_L, u_R, \hat{n}) \) for left/right states \( u_L, u_R \) in direction \( \hat{n} \)
The DG Method – General case

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

\[
(u_h)_t A_\kappa + \int_{\partial \kappa} F(u_h^+, u_h^-, \hat{n}) \, ds = 0
\]

- 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 and Jacobian](image)

\( \kappa \quad \partial \kappa \quad n \quad u_L \quad u_R \)
- **CDG fluxes for second order terms** [Peraire/Persson ’08]:
  - Provably optimal accuracy $O(h^{p+1})$
  - Sparsest known scheme (incl LDG/BR2/IP)

- **Implicit time integration by matrix-based Newton-Krylov solvers**
  - L-stable Diagonally Implicit Runge-Kutta (DIRK) methods
  - Block-ILU(0) preconditioners and automatic element ordering [Persson/Peraire ’08]
  - Implicit-Explicit Runge-Kutta schemes for LES-type problems [Persson ’11]
Parallel Solvers

- 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
- Distributed parallel solvers developed in [Persson ’09]
- Parallelization to 1000’s of processes by domain decomposition
- Close to perfect speedup for time accurate simulations
Implementation: The 3DG Software Package

- High-order discretizations on unstructured meshes
- Optimized C++ code with MATLAB and Python interfaces
- Capable of simulating challenging problems:
  - complex real-world geometries
  - transitional flows, multiple scales
  - moving and deforming domains
  - fluid-structure interactions
- General multiphysics framework applicable to a wide range of challenging problems

Thin Structures

Unsteady Flows

Aeroacoustics
Consider the flow over an SD7003 airfoil at Reynolds number 100,000 and $30^\circ$ angle of attack.

LES-type hybrid mesh, with DistMesh for unstructured triangles [Persson '04], each extruded as $6 \times 3$ prisms elements spanwise.

312,000 elements, 31M DOFs, elements curved to align with boundaries using nonlinear elasticity approach [Persson '09].
Governing Equations

- The compressible Navier-Stokes equations:

\[
\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i} (\rho u_i) = 0,
\]

\[
\frac{\partial}{\partial t} (\rho u_i) + \frac{\partial}{\partial x_i} (\rho u_i u_j + p) = + \frac{\partial \tau_{ij}}{\partial x_j} \quad \text{for } i = 1, 2, 3,
\]

\[
\frac{\partial}{\partial t} (\rho E) + \frac{\partial}{\partial x_i} (u_j (\rho E + p)) = - \frac{\partial q_j}{\partial x_j} + \frac{\partial}{\partial x_j} (u_j \tau_{ij}),
\]

with

\[
\tau_{ij} = \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_j} \delta_{ij} \right), \quad q_j = - \frac{\mu}{Pr} \frac{\partial}{\partial x_j} \left( E + \frac{p}{\rho} - \frac{1}{2} u_k u_k \right),
\]

\[
p = (\gamma - 1) \rho \left( E - \frac{1}{2} u_k u_k \right)
\]

- Turbulence modeled by Implicit Large Eddy Simulation (ILES)

[Uranga/Persson/Drela/Peraire ’11]
Cross-section Mesh

- Combine explicit and implicit solvers for turbulent flow problems
- Use Runge-Kutta IMEX schemes, with explicit integration in > 90% of the domain [Persson ’11]
Quality factor predictions for MEMS resonators

- Recent interest in high quality electromechanical resonators
- Applications in wireless communication systems, including cellular handsets, PDA’s, ultra-sensitive radar, etc
- Simulations challenging because of high Q-factors (low damping)
Quality factor predictions for MEMS resonators

- Physical system described by linear elastic isotropic model:

\[ \nabla \cdot \sigma^T + b = \rho \frac{\partial^2 u}{\partial t^2}, \quad \varepsilon = \frac{1}{2} (\nabla u + \nabla u^T), \quad \sigma = 2\mu \varepsilon + \lambda \text{tr}(\varepsilon) \mathbf{1}. \]

- However, the low damping requires extremely accurate numerics.

- Traditional CG-FEM methods are based in frequency domain, but the eigenvalue problems scale very poorly for 3-D problems.

- We propose to use a time-domain approach, with a high-order CDG-type linear elasticity formulation with explicit Runge-Kutta time-stepping [Govindjee/Persson, in review].
Quality factor predictions for MEMS resonators

- We find the resonant modes and corresponding Q-factors from the time-series using *harmonic inversion*
- Drive system by a broad-band Gaussian pulse force

\[ P(t) = A \exp\left(-(t - \alpha w)^2/w^2\right) \]

- Measure resulting displacements, fit to decaying harmonics:

\[ y(t) \approx \sum_k d_k e^{-i\omega_k t} \]
Quality factor predictions for MEMS resonators

- Excellent agreement with eigenvalue calculations and experiments
- Perfect scalability on parallel computers for large 3-D problems

<table>
<thead>
<tr>
<th>Problem</th>
<th>Single Disk</th>
<th>Double Disk</th>
</tr>
</thead>
<tbody>
<tr>
<td># tetrahedra</td>
<td>17,873</td>
<td>38,186</td>
</tr>
<tr>
<td>Polynomial degree</td>
<td>$p = 3$</td>
<td>$p = 3$</td>
</tr>
<tr>
<td>Total DOFs</td>
<td>2.14 million</td>
<td>4.58 million</td>
</tr>
</tbody>
</table>

Time / 1,000 RK steps

<table>
<thead>
<tr>
<th>Nodes, Cores</th>
<th>Time / 1,000 RK steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 nodes, 96 cores</td>
<td>41.2s</td>
</tr>
<tr>
<td>8 nodes, 192 cores</td>
<td>21.6s</td>
</tr>
<tr>
<td>16 nodes, 384 cores</td>
<td>11.5s</td>
</tr>
<tr>
<td>32 nodes, 768 cores</td>
<td>6.4</td>
</tr>
</tbody>
</table>

![Graph showing quality factor predictions](image)
1. Introduction and Motivation

2. High-Order Discontinuous Galerkin (DG) Methods
   - Application: Large Eddy Simulation of Flow over Airfoil
   - Application: Micromechanical Resonators

3. Reducing the cost: The Line-Based DG Method

4. ALE Formulation for Deforming Domains
   - Application: Vertical Axis Wind Turbines
   - Application: Flapping Flight of Bat
Number of non-zeros in Jacobian matrix for nodal DG (\(C\) solution components, \(T\) simplex elements, degree \(p\), dimension \(D\))

- Nodes per element \(S = \binom{p+D}{p} = \mathcal{O}(p^D)\),
- Nodes per face \(s = \binom{p+D-1}{D-1} = \mathcal{O}(p^{D-1})\)

Example: 3-D Navier-Stokes, \(p = 3\), \(T = 100,000\):

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Size</th>
<th>Example storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solution vector</td>
<td>(SCT)</td>
<td>80 MB</td>
</tr>
<tr>
<td>Inviscid Jacobian</td>
<td>((S^2 + (D + 1)s^2)C^2T)</td>
<td>16 GB</td>
</tr>
<tr>
<td>CDG Jacobian</td>
<td>((S^2 + (D + 1)Ss)C^2T)</td>
<td>24 GB</td>
</tr>
<tr>
<td>BR2/IP Jacobian</td>
<td>((S^2 + (D + 1)(S + s)s)C^2T)</td>
<td>32 GB</td>
</tr>
<tr>
<td>Full block Jacobian</td>
<td>((D + 2)S^2C^2T)</td>
<td>40 GB</td>
</tr>
</tbody>
</table>
Observation: A nodal Galerkin approach typically couples all nodes inside an element, which gives a stencil size of $O(p^D)$.

A finite difference approach would apply difference approximations along each coordinate direction, with stencil size $O(Dp)$.

Goal: Study unstructured methods with similar sparse stencils.


Full element connectivity  
Line-based connectivity
Map system of conservation law from $v$ to reference element $V$:

$$
\frac{\partial u}{\partial t} + \nabla \cdot F(u) = 0
$$

$$
J \frac{\partial u}{\partial t} + \nabla x \cdot \tilde{F}(u) = 0
$$

where $\tilde{F} = (\tilde{f}_1, \tilde{f}_2, \tilde{f}_3) = JG^{-1}F$, with $G = \nabla x x$ and $J = \det(G)$.

Consider curve $x_{jk}(\xi) = x(\xi, X_j, X_k)$, find $r_{jk}(X_1) \approx \partial \tilde{f}_1 / \partial X_1$ by a 1-D DG procedure: Find $r_{jk}(\xi) \in P_p([0, 1])^m$ such that

$$
\int_0^1 r_{jk}(\xi) \cdot v(\xi) \, d\xi = \int_0^1 \frac{df_1(u_{jk}(\xi))}{d\xi} \cdot v(\xi) \, d\xi
$$

$$
= \tilde{f}_1(u_{jk}^+(1), u_{jk}(1)) \cdot v(1) - \tilde{f}_1(u_{jk}(0), u_{jk}^-(0)) \cdot v(0) - \int_0^1 \tilde{f}_1(u_{jk}(\xi)) \cdot \frac{dv}{d\xi} \, d\xi
$$
\[
\frac{\partial u}{\partial t} + \nabla \cdot F(u) = 0
\]
\[
J \frac{\partial u}{\partial t} + \nabla x \cdot \tilde{F}(u) = 0
\]

- Note that if \( N = (1, 0, 0) \), then

\[
\tilde{f}_1 = \tilde{F} \cdot N = (JG^{-1}F) \cdot N = F \cdot (JG^{-T}N) = F \cdot n
\]

with the (non-normalized) normal vector \( n = JG^{-T}N \).

- Therefore, the numerical flux with \( N_1^+ = (1, 0, 0) \) can be written

\[
\tilde{f}_1(u_R, u_L) = \tilde{F} \cdot N_1^+(u_R, u_L) = \tilde{F} \cdot n_1^+(u_R, u_L),
\]

which involves the original fluxes \( F \) in the normal direction \( n_1^+ \).

- Use existing approximate Riemann solver as-is
Find $r_{jk}(\xi)$ by standard finite element procedure.

$$u_{jk}(\xi) = \sum_{i=0}^{p} u_{ijk} \phi_i(\xi), \quad r_{jk}(\xi) = \sum_{i=0}^{p} r_{ijk} \phi_i(\xi)$$

Discrete form $M r_{jk} = b$, find $r_{jk}$ by solving $m$ linear systems with $(p + 1)$-by-$(p + 1)$ mass matrix $M$.

Repeat along each direction to obtain semi-discrete formulation:

$$J_{ijk} \frac{du_{ijk}}{dt} + \sum_{n=1}^{3} r_{ijk}^{(n)} = 0$$

Observations:

- All integrals are one-dimensional
- No statement about integration/flux points: Integrals assumed exact
- Numerical fluxes only evaluated point-wise
- CG would have connected neighboring elements globally
Sparsity Patterns

<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>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line-DG connectivities</td>
<td>10</td>
<td>13</td>
<td>16</td>
<td>19</td>
<td>22</td>
<td>25</td>
<td>28</td>
<td>31</td>
<td>34</td>
<td>37</td>
</tr>
<tr>
<td>Nodal-DG connectivities</td>
<td>32</td>
<td>81</td>
<td>160</td>
<td>275</td>
<td>432</td>
<td>637</td>
<td>896</td>
<td>1215</td>
<td>1600</td>
<td>2057</td>
</tr>
</tbody>
</table>

- For $p = 3$ the Line-DG method is 10 times sparser, and for $p = 10$ it is more than 50 times sparser.
- Convergence test for Euler vortex test problem using Line-DG
- Optimal $O(h^{p+1})$ convergence in the infinity norm observed

Coarsest mesh, with degree $p = 7$

Solution (density)
For viscous terms, we use an LDG-type approach. Rewrite as:

\[
\frac{\partial u}{\partial t} + \nabla \cdot F(u, q) = 0, \quad \nabla u = q
\]

and set

\[
\hat{F} \cdot n(u, q, n) = \{\{F(u, q) \cdot n\}\} + C_{11}[u \otimes n] + C_{12}[F(u, q) \cdot n]
\]

\[
\hat{u}(u, q, n) = \{\{u\}\} - C_{12} \cdot [u \otimes n] + C_{22}[F(u, q) \cdot n]
\]

- If \(C_{22} = 0\), \(q\) can be eliminated locally
- If \(C_{12} = S_i \pm n_i^\pm /2\) for some switch function \(S_i^\pm \in \{-1, 1\}\), the scheme gets a simple and sparse upwind/downwind structure:

\[
\hat{F} \cdot n(u_R, q_R, u_L, q_L, n) = C_{11}[u \otimes n] + \begin{cases} 
F(u_R, q_R) \cdot n & \text{if } S = +1 \\
F(u_L, q_L) \cdot n & \text{if } S = -1
\end{cases}
\]

\[
\hat{u}(u_R, q_R, u_L, q_L, n) = C_{22}[F(u, q) \cdot n] + \begin{cases} 
u_L & \text{if } S = +1 \\
u_R & \text{if } S = -1
\end{cases}
\]
Consistency requirements for switch function $S_i^\pm$:

- $S_i^+ = -S_i^-$
- Opposite signs on the two sides of a shared face

Can easily choose $S_i^\pm$ consistently along globally connected lines of elements
Final semi-discrete form for system of equations (including source term):

\[
\frac{\partial u}{\partial t} + \nabla \cdot F(u, \nabla u) = S(u, \nabla u)
\]

becomes

\[
\frac{du_{ijk}}{dt} + \frac{1}{J_{ijk}} \sum_{n=1}^{3} r_{ijk}^{(n)} = S(u_{ijk}, q_{ijk})
\]

\[
\frac{1}{J_{ijk}} \sum_{n=1}^{3} d_{ijk}^{(n)} = q_{ijk}
\]

where both \( r_{ijk}^{(n)} \) and \( d_{ijk}^{(n)} \) in general depend on both \( u \) and \( q \).
Convergence, Poisson’s equation

- Poisson model problem $-\nabla^2 u = f$
- $C_{12} = S_i^\pm n_i^\pm / 2$, $C_{11} = C_{22} = 0$
  (the minimum dissipation LDG method)
- Optimal error $O(h^{p+1})$ in $u$:

<table>
<thead>
<tr>
<th>$p$</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>Rate $u$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.0</td>
<td>3.1</td>
<td>3.9</td>
<td>5.4</td>
<td>6.1</td>
<td>7.1</td>
<td>8.7</td>
<td></td>
</tr>
<tr>
<td>2.1</td>
<td>3.1</td>
<td>4.0</td>
<td>5.4</td>
<td>6.0</td>
<td>7.5</td>
<td>8.5</td>
<td></td>
</tr>
<tr>
<td>2.0</td>
<td>3.0</td>
<td>4.0</td>
<td>5.1</td>
<td>6.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rate $q$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1</td>
<td>2.2</td>
<td>3.2</td>
<td>4.4</td>
<td>5.1</td>
<td>6.4</td>
<td>7.3</td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>2.1</td>
<td>3.0</td>
<td>4.2</td>
<td>5.0</td>
<td>6.4</td>
<td>7.0</td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>2.0</td>
<td>2.9</td>
<td>4.0</td>
<td>5.0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Line-DG – Navier-Stokes

- Flow around SD7003 foil, $Re = 5,000$, degree $p = 8$
- Steady-state solution using Newton-Krylov method
Semi-discrete split form:

\[ \frac{dU}{dt} = R(U, Q), \]
\[ Q = D(U, Q). \]

If \( C_{22} = 0 \), then \( D(U, Q) = D(U) \) and the system can be written:

\[ \frac{dU}{dt} = R(U, D(U)) \]

An implicit solver requires solution of equations of the form

\[ (I - \alpha \Delta t A) \Delta U = \Delta t R(U, D(U)) \]

with

\[ A = \frac{dR}{dU} = \frac{\partial R}{\partial U} + \frac{\partial R}{\partial Q} \frac{\partial D}{\partial U} = K_{11} + K_{12}K_{21}. \]

Problem: \( K_{11}, K_{12}, K_{21} \) sparse but not \( K_{12}K_{21} \)
Special solvers required to retain sparsity:

1. Form Krylov matrix-vector products implicitly without forming $K_{12}K_{21}$:
   \[ Ap = K_{11}p + K_{12}(K_{21}p) \]

2. Precondition using Jacobi, Multigrid, ILU, or ADI-type methods

3. Other approaches: subiterations, local timestepping, implicit-explicit, etc
Unsteady Navier-Stokes around SD7003 airfoil

- Mach 0.2, $Re = 5,000$, AoA = 30°
- $p = 5$, DIRK 3/3 scheme in time
- GMRES + Jacobi preconditioning
- Quasi-Newton method: Re-use Jacobians between iterations and timesteps
- Essentially “explicit performance” for an implicit scheme: Majority of compute time spent in Newton residual evaluations
1 Introduction and Motivation

2 High-Order Discontinuous Galerkin (DG) Methods
   - Application: Large Eddy Simulation of Flow over Airfoil
   - Application: Micromechanical Resonators

3 Reducing the cost: The Line-Based DG Method

4 ALE Formulation for Deforming Domains
   - Application: Vertical Axis Wind Turbines
   - Application: Flapping Flight of Bat
• Use mapping-based ALE formulation for moving domains
  [Visbal/Gaitonde ’02], [Persson/Bonet/Peraire ’09]
• Map from reference domain $V$ to physical deformable domain $v(t)$
• Introduce the *mapping deformation gradient* $G$ and the *mapping velocity* $v_X$ as

\[
G = \nabla_X G
\]

\[
v_X = \frac{\partial G}{\partial t}
\]

and set $g = \det(G)$

• Transform equations to account for the motion
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

\[
\nabla_x U_x = \nabla_X(g^{-1}U_X)G^{-T} = (g^{-1}\nabla_X U_X - U_X \nabla_X(g^{-1}))G^{-T}
\]

Details in [Persson/Bonet/Peraire '09], including how to satisfy the so-called Geometric Conservation Law (GCL)
Recent interest in vertical axis wind turbines (VAWT):
- 2D airfoils, easy to manufacture, supportable at both ends
- Omnidirectional (good in gusty, low wind, e.g. close to ground)
- Lower blade speeds – lower noise and impact
- Can be packed close together in wind farms

Numerical simulations can help overcome remaining challenges:
- Lower theoretical (and practical) efficiency than HAWTs
- Sensitive to design conditions
- Structural problems, fatigue and catastrophic failure

Windterra ECO 1200 1Kw VAWT
Vertical Axis Wind Turbines

- Experimental design by G. Dahlbacka (LBNL) and collaborators
- 3kW unit, CAD design (left) and assembled unit (right)
Preliminary 2D simulation, using vertical symmetry

Solve the Navier-Stokes equations in a rotating frame:

$$\mathcal{G}(X, Y, t) = \begin{bmatrix} \cos \omega t & -\sin \omega t \\ \sin \omega t & \cos \omega t \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix}$$

Hybrid boundary layer/unstructured mesh, element degree $p = 3$
VAWT – Numerical Results

- Simulation with freestream wind speed 12 m/s (horizontally, from the left) and wing tip speed ratio 1.5
- Visualization by $\omega_z$-vorticity in rotating frame

Moment on each blade vs. time
Bio-Inspiration for Flapping Wing MAVs

- Develop high-order accurate simulation capabilities that capture the complex physics in flapping flight.
- Use the computational tools for increased understanding and to design optimized flapping kinematics.
Domain Mapping

- Highly complex wing motion from measured data
- Construct mapping $\mathcal{G}(X, t)$ *numerically* by nonlinear solid mechanics approach [Persson ’09]
- A reference mesh (left) is deformed elastically to smoothly align with the prescribed wing motion (right)
- Grid velocity $v_X = \frac{\partial \mathcal{G}}{\partial t} \bigg|_X$ defined consistently with DIRK scheme
Flapping Bat Flight Simulation

- Visualization of Mach number on isosurface of entropy
- Unphysical separation around simplified animal “body”
Optimal Design of Flapping Wings

- Goal: Automatically generate optimized flapping wing kinematics [Persson/Willis ’11]
- A multifidelity approach, with wake-only, panel, and high-order DG methods
- Example: Flapping wing pair, prescribed camber, solve for optimal wing twist distribution
Conclusions and Summary

- DG and related high-order methods are getting sufficiently mature to handle realistic problems.
- Applications in DNS/LES fluid applications, deforming domains, flapping flight, aeroacoustics, fluid-structure interaction.
- Drastically lower computational cost using line-based approach:
  - Orders of magnitude smaller stencils than nodal DG.
  - Very simple scheme structure, with 1-D integrals and standard point-wise Riemann solvers.
- Current work includes:
  - Development of efficient parallel time-integrators and solvers.
  - Shock capturing and RANS.
  - Extension of schemes to other element types.
  - Applications.