High-Fidelity Simulation of Flapping Wings
Designed for Energetically Optimal Flight

Per-Olof Persson
Department of Mathematics, University of California, Berkeley
Mathematics Department, Lawrence Berkeley National Laboratory

Berkeley Optimization Day

October 1, 2011
Outline

1 Background
2 Introduction to PDE-Constrained Optimization
   - The Finite Element Method
   - Optimization using the Adjoint Method
3 The Discontinuous Galerkin Method
4 Applications
   - Aerodynamic Shape Optimization
   - Structural Design
   - Optimal Flapping Wing Design
Outline

1 Background

2 Introduction to PDE-Constrained Optimization
   - The Finite Element Method
   - Optimization using the Adjoint Method

3 The Discontinuous Galerkin Method

4 Applications
   - Aerodynamic Shape Optimization
   - Structural Design
   - Optimal Flapping Wing Design
Numerical Simulations and Optimization

- Computational Science and Engineering – Constructing mathematical models and numerical solution techniques to analyze and solve scientific and engineering problems
- Can help reconstruct and understand *known events* (studying natural phenomena, accident investigations, etc)
- Can be used to *predict future situations* → optimization as part of the design process
The Numerical Simulation Process

- Mathematical modeling
  - Define regions, PDE-based models, boundary conditions, etc

- Geometry modeling
  - Describe the regions using Computer Aided Design (CAD)

- Mesh generation
  - Discretize the regions using simple shapes such as triangles and tetrahedra

- Numerical solution
  - Approximate PDEs numerically and solve using efficient algorithms

- Post-processing/Visualization
  - Compute desired quantities, generate images and animations that illustrate the solution

*All components used repeatedly in the optimization process*
Find models based on partial differential equations (PDEs)

Example (Fluid dynamics): The Navier-Stokes equations

$$\rho \left( \frac{\partial u}{\partial t} + u \cdot \nabla u \right) = -\nabla p + \mu \nabla^2 u + f, \quad \nabla \cdot u = 0$$

Example (Elastodynamics): Navier’s equations

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

Example (Electromagnetics): Maxwell’s equations

$$\nabla \cdot E = \frac{\rho}{\varepsilon_0} \quad \nabla \cdot B = 0$$

$$\nabla \times E = -\frac{\partial B}{\partial t} \quad \nabla \times B = \mu_0 J + \mu_0 \varepsilon_0 \frac{\partial E}{\partial t}$$

Also need to specify the physical region, boundary conditions, material parameters, etc
Geometry Modeling

- Generate the physical domains using CAD-based software
Mesh Generation

- Subdivide the domains into simple geometric shapes
- Practical applications require *unstructured meshes*:
  - Complex *geometries* need flexible element topologies
  - Complex *solution fields* need spatially variable resolution
Numerical Solution

- Use numerical techniques to approximate the solution to the PDEs on each element and coupling them together
- Use parallel computers for solving the large, generally nonlinear, algebraic systems of equations that arise
Post-processing / Visualization

- Calculate data such as output quantities, cost functions, etc.
- Generate figures/animations to help understand the results.
Outline

1. Background

2. Introduction to PDE-Constrained Optimization
   - The Finite Element Method
   - Optimization using the Adjoint Method

3. The Discontinuous Galerkin Method

4. Applications
   - Aerodynamic Shape Optimization
   - Structural Design
   - Optimal Flapping Wing Design
The Finite Element Method (FEM)

- Consider Poisson's equation \(-u_{xx} = f\) for \(x \in (0, 1)\), \(u(0) = u(1) = 0\)
- Discretize domain into \textit{elements} (intervals)
- Seek approximate solution in space of piecewise polynomials \(\hat{X}_0\)
- Impose equation weakly: Seek \(\hat{u} \in \hat{X}_0\) such that for all \(v \in \hat{X}_0\):

\[
\int_0^1 (-\hat{u}_{xx} - f)v \, dx = \int_0^1 \hat{u}_x v_x \, dx - \int_0^1 fv \, dx
\]

- Introduce appropriate basis functions for \(\hat{X}_0\)
- Leads to discrete linear system \(Ku = b\), with element-wise local \textit{stiffness matrix} \(K\)
Example: Heat Exchanger

- Consider a gas passing through a series of heat exchangers at constant velocity.
- Gas entry/exit temperature $T_\infty$ given.
- By suitable non-dimensionalization, can be described by a convection-diffusion equation:

\[
u_x = \kappa u_{xx} + f(x; \alpha) \quad \text{for} \quad x \in (0, 1), \quad u(0) = u(1) = 0,
\]

with \( f(x; \alpha) = \sum_i \alpha_i P_i(x) \), heat sources \( \alpha = (\alpha_1, \ldots, \alpha_n) \), and piecewise constant

\[P_i = \begin{cases} 
1 & \text{if } (i - 1)/n < x < i/n \\
0 & \text{otherwise}
\end{cases}\]
The Galerkin approach leads to the FEM formulation:

Find $u_h \in \hat{X}_0$ s.t.

$$
\int_0^1 (\kappa u_{h,x}(x)v_x(x) - u_h(x)v_x(x)) \, dx = \int_0^1 f(x; \alpha)v(x) \, dx, \quad \forall v \in \hat{X}_0
$$

Use space of continuous piecewise linear functions for $\hat{X}_0$

Leads to linear system

$$
r(u, \alpha) = Ku - b(\alpha) = 0
$$

Example: $\kappa = 0.1$, $n_e = 40$

Elements, $\alpha = (1, \ldots, 1)$
**Goal:** Find optimal heat sources $\alpha^\ast$ that generates a desired target temperature distribution $\bar{u}(x)$

- Minimize cost functional

$$I(\alpha) = \frac{1}{2} \int_0^1 (u_h(x; \alpha) - \bar{u}(x))^2 \, dx$$

where $u_h(x; \alpha)$ is the numerical approximation to $u$

- Use simple steepest descent algorithm:

$$\alpha^{r+1} = \alpha^r - \beta \left. \frac{dI}{d\alpha} \right|_{\alpha=\alpha^r}$$

with scalars $\beta$ chosen by a line search method

- Need sensitivities $dI/d\alpha$
The Adjoint Method for Sensitivity Analysis

- Suppose \( r(u, \alpha) = 0 \), and consider calculating the gradient

\[
\frac{dI}{d\alpha} = \frac{\partial I}{\partial \alpha} + \frac{\partial I}{\partial u} \frac{\partial u}{\partial \alpha}
\]

- Use implicit function theorem on the constraint:

\[
\frac{dr}{d\alpha} = 0 \quad \implies \quad \frac{\partial r}{\partial u} \frac{\partial u}{\partial \alpha} = -\frac{\partial r}{\partial \alpha}
\]

- The *direct method* solves \( n = \text{dim}(\alpha) \) linear systems for \( \partial u / \partial \alpha \)

- The *adjoint method* solves one linear system for the adjoint variables \( \psi \) s.t.

\[
\left( \frac{\partial r}{\partial u} \right)^T \psi = \left( \frac{\partial I}{\partial u} \right)
\]

which gives

\[
\frac{dI}{d\alpha} = \frac{\partial I}{\partial \alpha} - \frac{\partial I}{\partial u} \left( \frac{\partial r}{\partial u} \right)^{-1} \left( \frac{\partial r}{\partial \alpha} \right) = \frac{\partial I}{\partial \alpha} - \psi^T \frac{\partial r}{\partial \alpha}
\]
Optimized Heat Exchanger
Outline

1 Background

2 Introduction to PDE-Constrained Optimization
   • The Finite Element Method
   • Optimization using the Adjoint Method

3 The Discontinuous Galerkin Method

4 Applications
   • Aerodynamic Shape Optimization
   • Structural Design
   • Optimal Flapping Wing Design
The Discontinuous Galerkin Method

- 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 Discontinuous Galerkin Method

- 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
Second-order terms, time integration, solvers

- 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
1. Background

2. Introduction to PDE-Constrained Optimization
   - The Finite Element Method
   - Optimization using the Adjoint Method

3. The Discontinuous Galerkin Method

4. Applications
   - Aerodynamic Shape Optimization
   - Structural Design
   - Optimal Flapping Wing Design
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]
Large Eddy Simulation of Flow over Airfoil

- Flow over an SD7003 airfoil at Reynolds number 100,000 and 30° angle of attack
Let design variables $\alpha$ control the domain (e.g. geometric parameters or node locations).

To retain a well-shaped mesh, changes in $\alpha$ typically affect all mesh nodes $X$ by a mapping $x = x(X, \alpha)$.

Compute residual sensitivities by

$$\frac{\partial r}{\partial \alpha} = \frac{\partial r}{\partial x} \frac{\partial x}{\partial \alpha}$$

Many options for cost function, such as:

- Minimize drag force
- Attain a target pressure distribution
Aerodynamic Shape Optimization

- Example: 3D Wing/body optimization [Elliott/Peraire]
- Obtain target pressure distribution
Large Deformation Solid Dynamics

- The dynamic deformation of a rubber-like material
- A hyperelastic neo-Hookean constitutive model
- High-order DG method in space, 3rd order DIRK in time
Topology Optimization – Structural Design

- Linear elastostatics for displacements $u$:
  \[
  \begin{aligned}
  -\text{div}(A\varepsilon(u)) &= 0, \quad \text{in } \Omega \\
  u &= 0, \quad \text{on } \Gamma_D \\
  (A\varepsilon(u)) &= g, \quad \text{on } \Gamma_N.
  \end{aligned}
  \]

- Minimize compliance
  \[
  \int_{\partial \Omega} g \cdot u \, ds = \int_{\Omega} A\varepsilon(u) \cdot \varepsilon(u) \, dx \quad \text{subject to } \|\Omega\| = K.
  \]
Self-adjoint problem – Normal velocity for descent direction: [Murat/Simon], [Sethian/Wiegmann], [Allaire et al]

\[ F(x) = -A e(u) \cdot e(u) \]

Use level sets for geometry, FEM for elasticity [Persson ’05]

Problem regularized by the geometry discretization
Consider eigenvalue problem

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

with

\[\rho(x) = \begin{cases} 
\rho_1 & \text{for } x \not\in S \\
\rho_2 & \text{for } x \in S.
\end{cases}\]

Solve the optimization problem

\[\min_S \lambda_1 \text{ or } \lambda_2 \text{ subject to } \|S\| = K.\]
Solve eigenvalue problem for $i$th eigenmode $\lambda_i, u_i$ using linear finite elements on an unstructured mesh.

Decrease eigenvalue by moving interface in the normal direction by the descent field [Osher/Santosa ’02]

$$F = \frac{\lambda_i (\rho_2 - \rho_1)}{\int_{\Omega} \rho u_i^2 \, dx} u_i^2.$$ 

Use level sets to propagate interface, and DistMesh to generate new unstructured meshes.

Implement the conservation constraint $\|S\| = K$ by solving for a Lagrange multiplier $\nu$ such that the direction field $F + \nu$ conserves mass, using Newton’s method.
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
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

$$G = \nabla_X g$$
$$v_X = \frac{\partial g}{\partial t} \bigg|_X$$

and set $g = \det(G)$

Transform equations to account for the motion

$N dA$ $G$, $g$, $v_X$ $x_2$ $x_1$ $X_2$ $X_1$ $V$ $v$ $nda$
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

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

Details in [Persson/Bonet/Peraire '09], including how to satisfy the so-called Geometric Conservation Law (GCL)
Vertical Axis Wind Turbines

- Experimental design by G. Dahlbacka (LBNL) and collaborators
- 3kW unit, assembled unit (left), numerical simulation (right)
Example: Dragonfly, Compliant Wings

Experiment (A. Song, Brown U)  Fluid/membrane simulation
Bat Simulation – 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”
**Goal:** To design and analyze an effective flapping wing shape for cruising flight

Direct optimization using DG simulations unfeasible:
- Each forward simulation is highly computationally expensive
- Challenging to enforce constraints (time-periodic solutions with prescribed average lift and drag)
- Equations highly nonlinear and sensitive

Instead, we use a *multi-fidelity approach* with a range of simulation tools:
- Run wake-only method, determine optimal flapping kinematics
- Choose a flight speed, determine optimal target wake circulation
- Select wing planform and span-wise camber distribution
- Match wing shed circulation with target wake using DLM
- Evaluate designs using panel and high-order DG N-S methods
Wing geometric properties

Wing is designed to perform similarly to a medium-sized fruit bat, *Cynopterus brachyotis*

- Leading and trailing edges defined using parametrized quadratic lines
- Chord at centerline $c = 0.25$, tip-to-tip span $b = 1$
- Optimal energetics framework: $Ma = 0.0$, Reynolds number $\approx 35,000$
- Navier Stokes: Mach number 0.2, Reynolds number 3,000
- Four simulation cases
Wake Only Aerodynamics Model: Hall et al.

- The wake is the fluid dynamics momentum transfer footprint
- Simple and Fast: Uses only the fixed-in-space wake trace and force requirements
- Goal: To determine the energetically best vorticity distribution for a given wake shape (similar concept as Trefftz Plane)
Wake Only Energetics Model

1) A large collection of possible flapping kinematics are modeled using the wake-only aerodynamics method.

2) The wake-only model results are compiled into a power consumption database/design-space.

3) A flight-force balance is used to determine which wake(s) is the best candidate for the flight condition.
Wake Only Energetics

- Wake geometry for full flapping cycle (two periods shown, starting with downstroke)

- **Idea**: Find a flapping wing that generates the target wake (at each timestep)

- **Approach**: Modified doublet lattice method
Double Lattice Method – FastAero

- Panel Method Potential Flow Solution
  - Efficient for large wing motions
  - Requires surface meshes only
- Inviscid, will not be able to model:
  - Flow separation
  - Leading edge vortices
  - Viscous drag
- Implementation Details
  - Linear basis unstructured panel method
  - Unsteady free vortex particle/sheet wakes
  - Accelerated iterative solution (pFFT, FMM)
Quasi-Inverse Design: Geometry Definition

Automatic wing geometry definition

1) Define the wing planform:
   - Define L.E and T.E
   - Define twist axis: % chord

2) Define wing camber (Hermite Spline):
   - L.E. angle (explicit or function of flow angle)
   - T.E. angle half of L.E. angle

3) Twist local wing sections

4) Rotate wing through flapping angle
Quasi-Inverse Design (DLM)

For each timestep:

1) Solve DLM for current geometry (goal: find buffer-wake strength)

2) Compare buffer-wake strength to target-wake
   - If $\| \mu_{\text{wake.buffer}} - \mu_{\text{wake.target}} \| < tol$: goto next timestep
   - else: Find $d\mu_i/d\alpha_i$, find $\Delta \alpha_i$, mod. wing twist, repeat analysis
Tetrahedral Mesh

- Surfaces triangulated in parametric form using the DistMesh mesh generator [Persson ‘04], and tetrahedral volume mesh generated by a Delaunay refinement code [Peraire ‘98]
- 10,000 nodes and 50,000 tetrahedra in the half-domain
- Resolution focused around wing and in the wake
Deforming Mesh

- Mesh deformation using nonlinear elasticity
- The reference mesh (left) is deformed as the wing surface is moving (right)

Reference Configuration  Physical Configuration
Wing sections have zero camber: Aggressive leading edge angle
LEV formation in first $1/2$ of d/s, followed by significant shedding
Good design in potential flow, poor physical design

N-S lift (green), N-S drag (blue), DLM (dashed)
Navier-Stokes Simulation – Case 2, 10°

- L.E. angle (10°) close to flow alignment – separation mitigated
- Attached flow throughout (small LEV?)
- Strong tip vortices shed during D/S

N-S lift (green), N-S drag (blue), DLM (dashed)
Navier-Stokes Simulation – Case 3, 20°

- L.E. angle (20°) aggressive (greater than flow angle)
- Flow separates on lower/pressure surface of wing, LEV present
- Tip vortices and underside shedding present

N-S lift (green), N-S drag (blue), DLM (dashed)
Navier-Stokes Simulation – Case 4, Dynamic

- L.E. angle aligned locally → mitigate separation, minimal camber
- LEV forms on outboard during later half of D/S
- Wing-shape easily predicted during inviscid design

N-S lift (green), N-S drag (blue), DLM (dashed)
Conclusions and Summary

- Optimization the main motivation for PDE-based simulations
- Efficient gradient calculations using the adjoint method
- High-order Discontinuous Galerkin methods on unstructured meshes for fluids and solids
- A full multi-fidelity approach to design – Wake-only, DLM, and Navier-Stokes

Current work includes:
- New lower-cost discretizations and solvers
- More complex geometries/flight kinematics
- Full fluid-structure coupling