Unsteady CFD Optimization using High-Order Discontinuous Galerkin Finite Element Methods

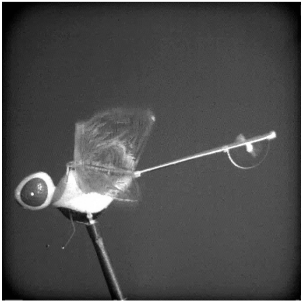
Matthew J. Zahr and Per-Olof Persson

Stanford University and University of California, Berkeley

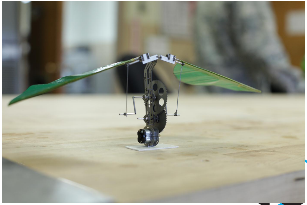
13th US National Congress on Computational Mechanics, San Diego, CA
TS4, Tuesday, July 28, 2015

Optimal Control: Flapping Flight

- Optimal control of a body immersed in a fluid leads to *unsteady PDE-constrained optimization*
- Goal: Determine **kinematics** of the body that minimizes some cost functional subject to **constraints**
  - Steady-state analysis insufficient
- Example: Energetically-optimal flapping at constant thrust
  - Biology
  - Micro Aerial Vehicles



Dragonfly Experiment
(A. Song, Brown U)

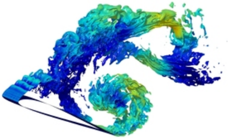


Micro Aerial Vehicle

Shape Optimization: Turbulence

- Shape optimization of a static or moving body in turbulent flow also leads to unsteady PDE-constrained optimization
  - Non-existence of steady-state necessitates unsteady analysis
- Goal: Determine **shape** (and possibly kinematics) that minimizes a cost functional, subject to constraints
- Applications
  - Shape of windmill blade for maximum energy harvesting
  - Maximum lift airfoil

Vertical Windmill



LES Flow past Airfoil

Problem Formulation

Goal: Find the solution of the *unsteady PDE-constrained optimization* problem

$$
\begin{aligned}
\underset{\boldsymbol{U},\ \boldsymbol{\mu}}{\text{minimize}} \quad & \mathcal{J}(\boldsymbol{U}, \boldsymbol{\mu}) \\
\text{subject to} \quad & \boldsymbol{C}(\boldsymbol{U}, \boldsymbol{\mu}) \leq 0 \\
& \frac{\partial \boldsymbol{U}}{\partial t} + \nabla \cdot \boldsymbol{F}(\boldsymbol{U}, \nabla \boldsymbol{U}) = 0 \quad \text{in} \quad v(\boldsymbol{\mu}, t)
\end{aligned}
$$

where

- $\boldsymbol{U}(\boldsymbol{x}, t)$ — PDE solution
- $\boldsymbol{\mu}$ — design/control parameters
- $\mathcal{J}(\boldsymbol{U}, \boldsymbol{\mu}) = \int_{T_0}^{T_f} \int_{\boldsymbol{\Gamma}} j(\boldsymbol{U}, \boldsymbol{\mu}, t)\, dS\, dt$ — objective function
- $\boldsymbol{C}(\boldsymbol{U}, \boldsymbol{\mu}) = \int_{T_0}^{T_f} \int_{\boldsymbol{\Gamma}} \mathbf{c}(\boldsymbol{U}, \boldsymbol{\mu}, t)\, dS\, dt$ — constraints

DOE CSGF

Approach to Unsteady Optimization

- Recast conservation law on deforming domain into one on *fixed*, reference domain (Arbitrary Lagrangian-Euler formulation)
- Globally high-order numerical discretization of transformed equations
  - Spatial Discretization: **Discontinuous Galerkin FEM**
  - Temporal Discretization: **Diagonally-Implicit Runge-Kutta**
  - Solver-consistent discretization of output quantities
- Fully-discrete adjoint method for high-order numerical discretization
- *Gradient-based optimization*

ALE Description of Conservation Law

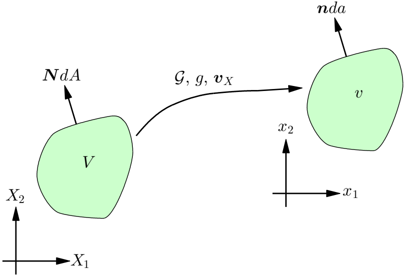
- Map from fixed reference domain V to physical, deformable (parametrized) domain $v(\boldsymbol{\mu}, t)$
- A point $\boldsymbol{X} \in V$ is mapped to $\boldsymbol{x}(\boldsymbol{\mu}, t) = \mathcal{G}(\mathbf{X}, \boldsymbol{\mu}, t) \in v(\boldsymbol{\mu}, t)$

- Introduce transformation

$$\boldsymbol{U}_{\boldsymbol{X}} = g\boldsymbol{U}$$
$$\boldsymbol{F}_{\boldsymbol{X}} = g\boldsymbol{G}^{-1}\boldsymbol{F} - \boldsymbol{U}_{\boldsymbol{X}}\boldsymbol{G}^{-1}\boldsymbol{v}_{\boldsymbol{X}}$$

where

$$\boldsymbol{G} = \nabla_{\boldsymbol{X}}\mathcal{G}, \;\; g = \det \boldsymbol{G}, \;\; \boldsymbol{v}_{\boldsymbol{X}} = \left.\frac{\partial \mathcal{G}}{\partial t}\right|_{\boldsymbol{X}}$$



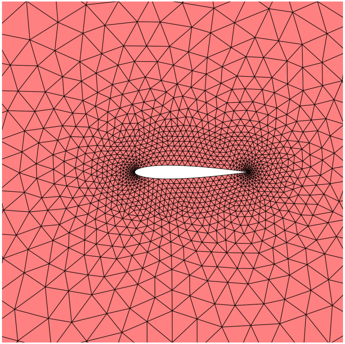
- Transformed conservation law

$$\left.\frac{\partial \boldsymbol{U}_{\boldsymbol{X}}}{\partial t}\right|_{\boldsymbol{X}} + \nabla_{\boldsymbol{X}} \cdot \boldsymbol{F}_{\boldsymbol{X}}(\boldsymbol{U}_{\boldsymbol{X}}, \; \nabla_{\boldsymbol{X}}\boldsymbol{U}_{\boldsymbol{X}}) = 0$$

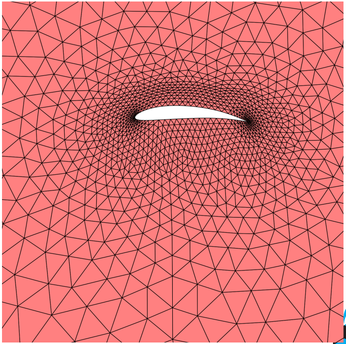
Domain Deformation

- Require mapping $\boldsymbol{x} = \mathcal{G}(\boldsymbol{X}, t)$ to obtain derivatives $\nabla_{\boldsymbol{X}} \mathcal{G}$, $\frac{\partial}{\partial t} \mathcal{G}$
- Shape deformation, via Radial Basis Functions (RBFs), and translational kinematic motion, $\boldsymbol{v}$, applied to reference domain

$$\boldsymbol{X}' = \boldsymbol{X} + \boldsymbol{v} + \sum \boldsymbol{w}_i \Phi(||\boldsymbol{X} - \boldsymbol{c}_i||)$$



Undeformed Mesh



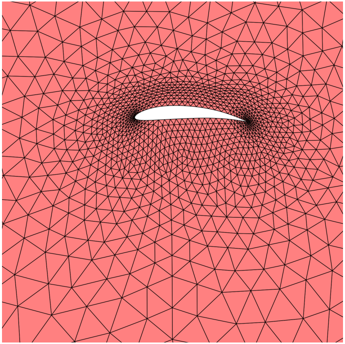
Shape Deformation, Translation

Domain Deformation

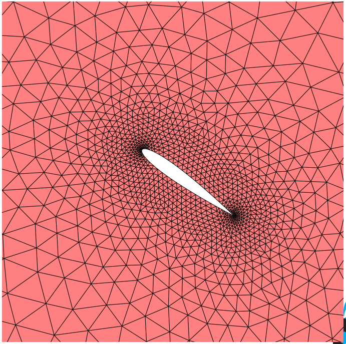
- Require mapping $\boldsymbol{x} = \mathcal{G}(\boldsymbol{X}, t)$ to obtain derivatives $\nabla_{\boldsymbol{X}}\mathcal{G}$, $\frac{\partial}{\partial t}\mathcal{G}$
- Rotational kinematic motion, $\boldsymbol{Q}$, applied via *blending map*

$$\boldsymbol{x} = b(d_R(\boldsymbol{X}))\boldsymbol{X}' + (1 - b(d_R(\boldsymbol{X})))\boldsymbol{Q}\boldsymbol{X}$$

  - $b : \mathbb{R} \rightarrow \mathbb{R}$ is a polynomial on $[0, 1]$ with $(n-1)/2$ vanishing derivatives at $0, 1$
  - $d_R(\boldsymbol{X})$ is signed distance between $\boldsymbol{X}$ and circle of radius R



Shape Deformation, Translation



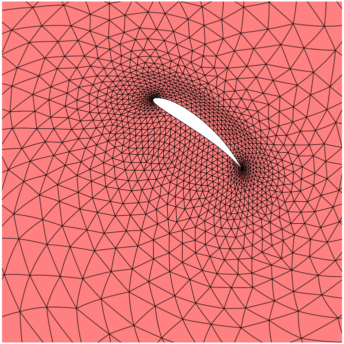
Rigid Rotation of Mesh

Domain Deformation

- Require mapping $\boldsymbol{x} = \mathcal{G}(\boldsymbol{X}, t)$ to obtain derivatives $\nabla_{\boldsymbol{X}} \mathcal{G}$, $\frac{\partial}{\partial t}\mathcal{G}$
- Rotational kinematic motion, $\boldsymbol{Q}$, applied via *blending map*

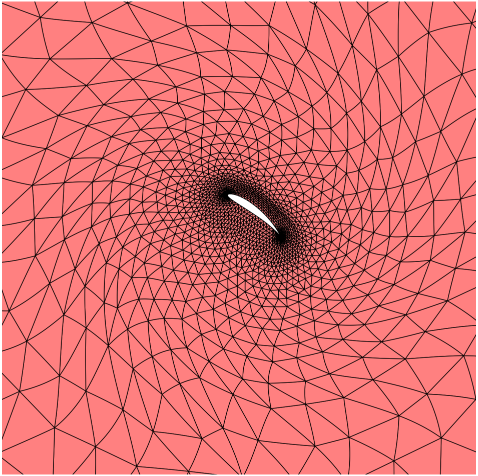
$$\boldsymbol{x} = b(d_R(\boldsymbol{X}))\boldsymbol{X}' + (1 - b(d_R(\boldsymbol{X})))\boldsymbol{Q}\boldsymbol{X}$$

  - $b : \mathbb{R} \rightarrow \mathbb{R}$ is a polynomial on $[0, 1]$ with $(n-1)/2$ vanishing derivatives at $0, 1$
  - $d_R(\boldsymbol{X})$ is signed distance between $\boldsymbol{X}$ and circle of radius R



Blended Mesh

Domain Deformation



Blended Mesh

Spatial Discretization: Discontinuous Galerkin

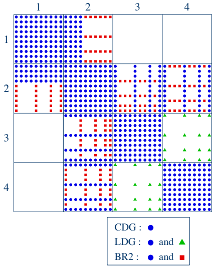
- Re-write conservation law as first-order system

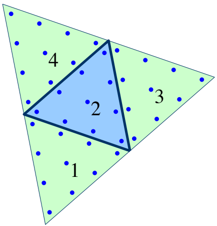
$$\left.\frac{\partial \boldsymbol{U_X}}{\partial t}\right|_{\boldsymbol{X}} + \nabla_{\boldsymbol{X}} \cdot \boldsymbol{F_X}(\boldsymbol{U_X},\ \boldsymbol{Q_X}) = 0$$
$$\boldsymbol{Q_X} - \nabla_{\boldsymbol{X}} \boldsymbol{U_X} = 0$$

- Discretize using DG

- Roe's method for inviscid flux
- Compact DG (CDG) for viscous flux
- *Semi-discrete* equations

$$\mathbb{M}\frac{\partial \boldsymbol{u}}{\partial t} = \boldsymbol{r}(\boldsymbol{u}, \boldsymbol{\mu}, t)$$
$$\boldsymbol{u}(0) = \boldsymbol{u}_0(\boldsymbol{\mu})$$





Stencil for CDG, LDG, and BR2 fluxes

Temporal Discretization: Diagonally-Implicit Runge-Kutta

- Diagonally-Implicit RK (DIRK) are implicit Runge-Kutta schemes defined by lower triangular Butcher tableau $\rightarrow$ **decoupled implicit stages**
- Overcomes issues with high-order BDF and IRK
  - Limited accuracy of A-stable BDF schemes (2nd order)
  - High cost of general implicit RK schemes (coupled stages)

$$\boldsymbol{u}^{(0)} = \boldsymbol{u}_0(\boldsymbol{\mu})$$

$$\boldsymbol{u}^{(n)} = \boldsymbol{u}^{(n-1)} + \sum_{i=1}^{s} b_i \boldsymbol{k}_i^{(n)}$$

$$\boldsymbol{u}_i^{(n)} = \boldsymbol{u}^{(n-1)} + \sum_{j=1}^{i} a_{ij} \boldsymbol{k}_j^{(n)}$$

$$\mathbb{M}\boldsymbol{k}_i^{(n)} = \Delta t_n \boldsymbol{r}\left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_i \Delta t_n\right)$$

$$\begin{array}{c|cccc}
c_1 & a_{11} & & & \\
c_2 & a_{21} & a_{22} & & \\
\vdots & \vdots & \vdots & \ddots & \\
c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\
\hline
 & b_1 & b_2 & \cdots & b_s
\end{array}$$

Butcher Tableau for DIRK scheme

Consistent Discretization of Output Quantities

- Consider any output functional of the form

$$\mathcal{F}(\boldsymbol{U}, \boldsymbol{\mu}) = \int_{T_0}^{T_f} \int_{\boldsymbol{\Gamma}} f(\boldsymbol{U}, \boldsymbol{\mu}, t) \, dS \, dt$$

- Define f_h as the high-order approximation of the spatial integral via the DG shape functions

$$f_h(\boldsymbol{u}(t), \boldsymbol{\mu}, t) = \sum_{\mathcal{T}_e \in \mathcal{T}_{\boldsymbol{\Gamma}}} \sum_{\mathcal{Q}_i \in \mathcal{Q}_{\mathcal{T}_e}} w_i f(\boldsymbol{u}_{ei}(t), \boldsymbol{\mu}, t) \approx \int_{\boldsymbol{\Gamma}} f(\boldsymbol{U}, \boldsymbol{\mu}, t) \, dS$$

- Then, the output functional becomes

$$\mathcal{F}(\boldsymbol{U}, \boldsymbol{\mu}) \approx \mathcal{F}_h(\boldsymbol{u}, \boldsymbol{\mu}) = \int_{T_0}^{T_f} f_h(\boldsymbol{u}(t), \boldsymbol{\mu}, t) \, dt$$

Consistent Discretization of Output Quantities

- Semi-discretized output functional

$$\mathcal{F}_h(\boldsymbol{u}, \boldsymbol{\mu}, t) = \int_{T_0}^{t} f_h(\boldsymbol{u}(t), \boldsymbol{\mu}, t)\, dt$$

- Differentiation w.r.t. time leads to the

$$\dot{\mathcal{F}}_h(\boldsymbol{u}, \boldsymbol{\mu}, t) = f_h(\boldsymbol{u}(t), \boldsymbol{\mu}, t)$$

- Write semi-discretized output functional *and* conservation law as monolithic system

$$\begin{bmatrix} \mathbb{M} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{u}} \\ \dot{\mathcal{F}}_h \end{bmatrix} = \begin{bmatrix} \boldsymbol{r}(\boldsymbol{u}, \boldsymbol{\mu}, t) \\ f_h(\boldsymbol{u}, \boldsymbol{\mu}, t) \end{bmatrix}$$

- Apply DIRK scheme to obtain

$$\boldsymbol{u}^{(n)} = \boldsymbol{u}^{(n-1)} + \sum_{i=1}^{s} b_i \boldsymbol{k}_i^{(n)}$$

$$\mathcal{F}_h^{(n)} = \mathcal{F}_h^{(n-1)} + \sum_{i=1}^{s} b_i f_h\left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_i^{(n-1)}\right)$$

$$\boldsymbol{u}_i^{(n)} = \boldsymbol{u}^{(n-1)} + \sum_{j=1}^{i} a_{ij} \boldsymbol{k}_j^{(n)}$$

$$\mathbb{M}\boldsymbol{k}_i^{(n)} = \Delta t_n \boldsymbol{r}\left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_i^{(n-1)}\right)$$

where $t_i^{(n-1)} = t_{n-1} + c_i \Delta t_n$

- Only interested in *final* time

$$F(\boldsymbol{u}^{(n)}, \boldsymbol{k}_i^{(n)}, \boldsymbol{\mu}) = \mathcal{F}_h^{(N_t)}$$

Adjoint Method

- Consider the *fully-discrete* output functional $F(\boldsymbol{u}^{(n)}, \boldsymbol{k}_i^{(n)}, \boldsymbol{\mu})$ corresponding to the continuous output functional $\mathcal{F}(\boldsymbol{U}, \boldsymbol{\mu}) = \int_{T_0}^{T_f} \int_{\boldsymbol{\Gamma}} f(\boldsymbol{U}, \boldsymbol{\mu}, t) \, dS \, dt$
  - May correspond to either the **objective** function or a **constraint**
- The *total derivative* with respect to the parameters $\boldsymbol{\mu}$, required in the context of gradient-based optimization, takes the form

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial F}{\partial \boldsymbol{\mu}} + \sum_{n=0}^{N_t} \frac{\partial F}{\partial \boldsymbol{u}^{(n)}} \frac{\partial \boldsymbol{u}^{(n)}}{\partial \boldsymbol{\mu}} + \sum_{n=1}^{N_t} \sum_{i=1}^{s} \frac{\partial F}{\partial \boldsymbol{k}_i^{(n)}} \frac{\partial \boldsymbol{k}_i^{(n)}}{\partial \boldsymbol{\mu}}$$

- The sensitivities, $\dfrac{\partial \boldsymbol{u}^{(n)}}{\partial \boldsymbol{\mu}}$ and $\dfrac{\partial \boldsymbol{k}_i^{(n)}}{\partial \boldsymbol{\mu}}$, are expensive to compute, requiring the solution of $n_{\boldsymbol{\mu}}$ linear evolution equations
- Adjoint method: alternative method for computing $\dfrac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}$ requiring one linear evolution evoluation equation for each output functional, F

Overview of Adjoint Derivation

- Define **auxiliary** PDE-constrained optimization problem

$$
\begin{aligned}
\underset{\substack{\boldsymbol{u}^{(0)},\ \ldots,\ \boldsymbol{u}^{(N_t)} \in \mathbb{R}^{N_{\boldsymbol{u}}},\\ \boldsymbol{k}_1^{(1)},\ \ldots,\ \boldsymbol{k}_s^{(N_t)} \in \mathbb{R}^{N_{\boldsymbol{u}}}}}{\text{minimize}} \quad & F(\boldsymbol{u}^{(0)},\ \ldots,\ \boldsymbol{u}^{(N_t)},\ \boldsymbol{k}_1^{(1)},\ \ldots,\ \boldsymbol{k}_s^{(N_t)},\ \bar{\boldsymbol{\mu}}) \\
\text{subject to} \quad & \tilde{\boldsymbol{r}}^{(0)} = \boldsymbol{u}^{(0)} - \boldsymbol{u}_0(\boldsymbol{\mu}) = 0 \\
& \tilde{\boldsymbol{r}}^{(n)} = \boldsymbol{u}^{(n)} - \boldsymbol{u}^{(n-1)} + \sum_{i=1}^{s} b_i \boldsymbol{k}_i^{(n)} = 0 \\
& \boldsymbol{R}_i^{(n)} = \mathbb{M}\boldsymbol{k}_i^{(n)} - \Delta t_n \boldsymbol{r}\left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_i^{(n-1)}\right) = 0
\end{aligned}
$$

- Define **Lagrangian**

$$
\mathcal{L}(\boldsymbol{u}^{(n)},\ \boldsymbol{k}_i^{(n)},\ \boldsymbol{\lambda}^{(n)},\ \boldsymbol{\kappa}_i^{(n)}) = F - {\boldsymbol{\lambda}^{(0)}}^T \tilde{\boldsymbol{r}}^{(0)} - \sum_{n=1}^{N_t} {\boldsymbol{\lambda}^{(n)}}^T \tilde{\boldsymbol{r}}^{(n)} - \sum_{n=1}^{N_t} \sum_{i=1}^{s} {\boldsymbol{\kappa}_i^{(n)}}^T \boldsymbol{R}_i^{(n)}
$$

Fully-Discrete Adjoint Equations

- The solution of the optimization problem is given by the **Karush-Kuhn-Tucker (KKT) sytem**

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{u}^{(n)}} = 0, \quad \frac{\partial \mathcal{L}}{\partial \boldsymbol{k}_i^{(n)}} = 0, \quad \frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}^{(n)}} = 0, \quad \frac{\partial \mathcal{L}}{\partial \boldsymbol{\kappa}_i^{(n)}} = 0$$

- The derivatives w.r.t. the state variables, $\frac{\partial \mathcal{L}}{\partial \boldsymbol{u}^{(n)}} = 0$ and $\frac{\partial \mathcal{L}}{\partial \boldsymbol{k}_i^{(n)}} = 0$, yield the **fully-discrete adjoint equations**

$$\boldsymbol{\lambda}^{(N_t)} = \frac{\partial F}{\partial \boldsymbol{u}^{(N_t)}}^T$$

$$\boldsymbol{\lambda}^{(n-1)} = \boldsymbol{\lambda}^{(n)} + \frac{\partial F}{\partial \boldsymbol{u}^{(n-1)}}^T + \sum_{i=1}^{s} \Delta t_n \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}} \left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_i \Delta t_n \right)^T \boldsymbol{\kappa}_i^{(n)}$$

$$\mathbb{M}^T \boldsymbol{\kappa}_i^{(n)} = \sum_{j=i}^{s} a_{ji} \Delta t_n \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}} \left(\boldsymbol{u}_j^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_j \Delta t_n \right)^T \boldsymbol{\kappa}_j^{(n)}$$

Fully-Discrete Adjoint Equations: Dissection

$$\boldsymbol{\lambda}^{(N_t)} = \frac{\partial F}{\partial \boldsymbol{u}^{(N_t)}}^T$$

$$\boldsymbol{\lambda}^{(n-1)} = \boldsymbol{\lambda}^{(n)} + \frac{\partial F}{\partial \boldsymbol{u}^{(n-1)}}^T + \sum_{i=1}^{s} \Delta t_n \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}} \left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_i \Delta t_n\right)^T \boldsymbol{\kappa}_i^{(n)}$$

$$\mathbb{M}^T \boldsymbol{\kappa}_i^{(n)} = \sum_{j=i}^{s} a_{ji} \Delta t_n \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}} \left(\boldsymbol{u}_j^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_j \Delta t_n\right)^T \boldsymbol{\kappa}_j^{(n)}$$

- **Linear** evolution equations solved **backward** in time
  - Requires solving linear systems of equations with $\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}^T$
  - Accurate solution of linear system required
- Primal state, $\boldsymbol{u}^{(n)}$, and stage, $\boldsymbol{k}_i^{(n)}$, required at each state/stage of dual solve
  - Parallel I/O
- Heavily-dependent on **chosen ouput**
  - $\boldsymbol{\lambda}^{(n)}$ and $\boldsymbol{\kappa}_i^{(n)}$ must be computed for each output functional F

Gradient Reconstruction via Dual Variables

- Equipped with the solution to the primal problem, $\boldsymbol{u}^{(n)}$ and $\boldsymbol{k}_i^{(n)}$, and dual problem, $\boldsymbol{\lambda}^{(n)}$ and $\boldsymbol{\kappa}_i^{(n)}$, the output gradient is reconstructed as

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial F}{\partial \boldsymbol{\mu}} - \boldsymbol{\lambda}^{(0)}{}^T \frac{\partial \boldsymbol{u}_0}{\partial \boldsymbol{\mu}} - \sum_{n=1}^{N_t} \Delta t_n \sum_{i=1}^{s} \boldsymbol{\kappa}_i^{(n)}{}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_i^{(n)})$$

- Independent of sensitivities, $\dfrac{\partial \boldsymbol{u}^{(n)}}{\partial \boldsymbol{\mu}}$ and $\dfrac{\partial \boldsymbol{k}_i^{(n)}}{\partial \boldsymbol{\mu}}$

Isentropic, Compressible Navier-Stokes Equations

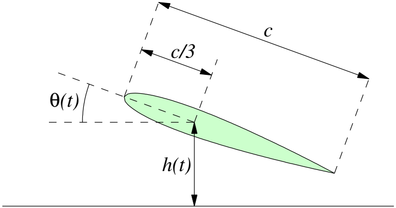
- Proposed globally high-order method holds for arbitrary conservation laws
- Applications in this work focused on compressible Navier-Stokes equations

$$
\begin{aligned}
\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i) &= 0 \\
\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_i}(\rho u_i u_j + p) &= +\frac{\partial \tau_{ij}}{\partial x_j} \quad \text{for } i = 1, 2, 3 \\
\frac{\partial}{\partial t}(\rho E) + \frac{\partial}{\partial x_i}\left(u_j(\rho E + p)\right) &= -\frac{\partial q_j}{\partial x_j} + \frac{\partial}{\partial x_j}(u_j \tau_{ij})
\end{aligned}
$$

- Isentropic assumption (entropy constant) made to reduce dimension of PDE system from $n_{\text{sd}+2}$ to $n_{\text{sd}+1}$

Problem Setup

$$
\begin{aligned}
\underset{h(t),\theta(t)}{\text{maximize}} \quad & \int_0^T \int_{\boldsymbol{\Gamma}} \boldsymbol{f} \cdot \boldsymbol{v} \, dS \, dt \\
\text{subject to} \quad & h(0) = h'(0) = h'(T) = 0, \;\; h(T) = 1 \\
& \theta(0) = \theta'(0) = \theta(T) = \theta'(T) = 0 \\
& \frac{\partial \boldsymbol{U}}{\partial t} + \nabla \cdot \boldsymbol{F}(\boldsymbol{U}, \nabla \boldsymbol{U}) = 0
\end{aligned}
$$

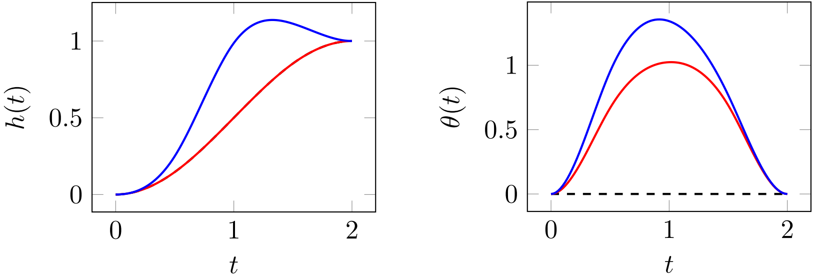


Airfoil schematic, kinematic description

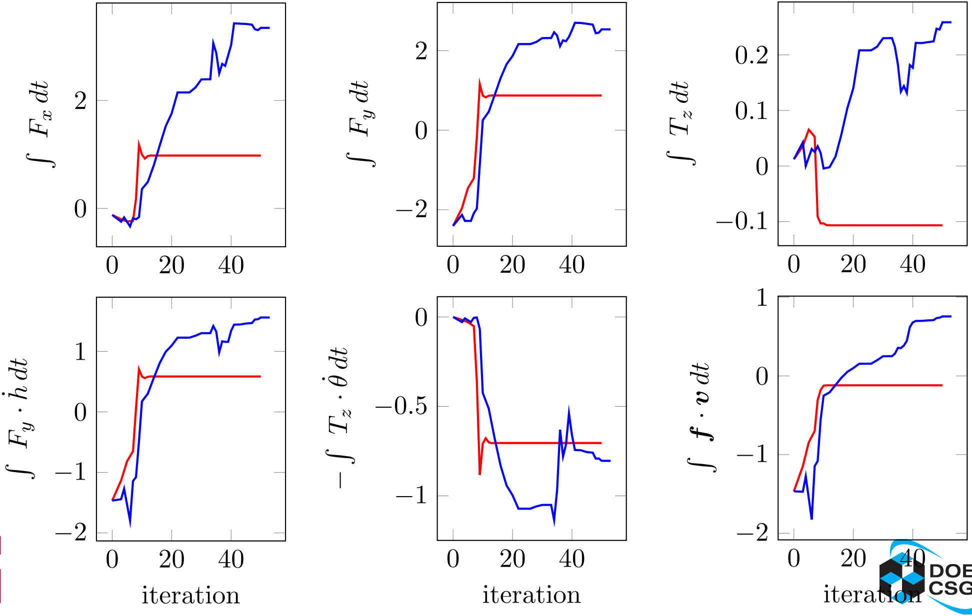
- Non-zero freestream velocity
- $h(t)$, $\theta(t)$ discretized via *clamped cubic splines*
- Knots of cubic splines as optimization parameters, $\boldsymbol{\mu}$
- Black-box optimizer: SNOPT

Optimization Setup

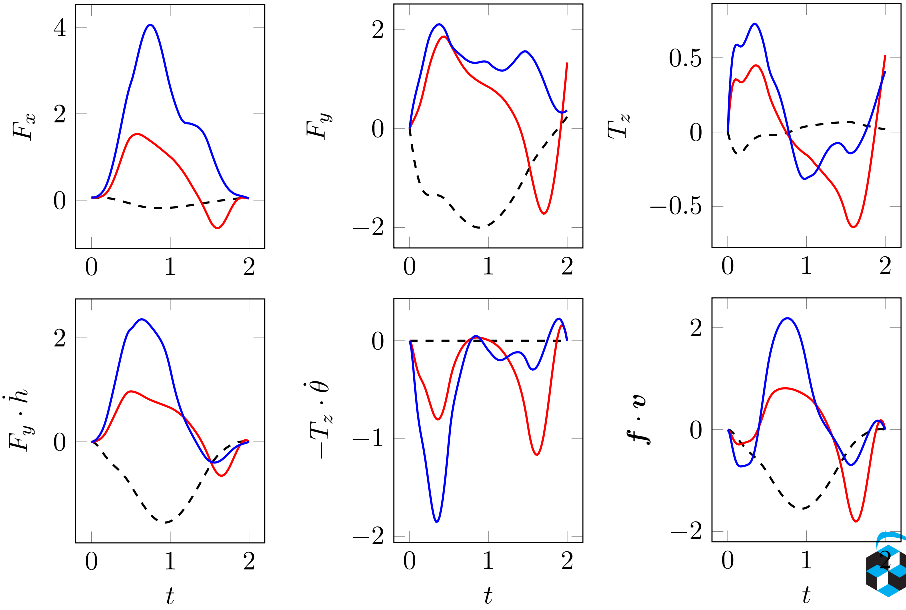
- Initial guess (- - -)
  - $h_0(t) = (1 - \cos(\pi t/T))/2$
  - $\theta_0(t) = 0$
- Optimization 1 (——)
  - $h_0(t) = (1 - \cos(\pi t/T))/2$
  - $\theta(t)$ parametrized (clamped cubic splines)
- Optimization 2 (——)
  - $h(t)$, $\theta(t)$ parametrized (clamped cubic splines)



Optimization Convergence



Output Functional Comparison



Optimization Results: Vorticity Field History

Initial Guess

$h_0(t)$, $\theta^*(t)$

$h^*(t)$, $\theta^*(t)$

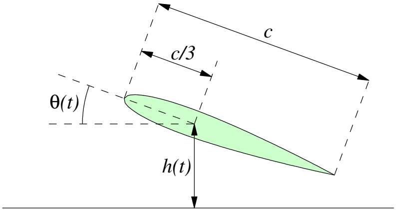
Play/Pause

Summary

	$\int F_x \, dt$	$\int F_y \, dt$	$\int T_z \, dt$	$\int F_y \cdot \dot{h} \, dt$	$-\int T_z \cdot \dot{\theta} \, dt$	$\int \boldsymbol{f} \cdot \boldsymbol{v} \, dt$
(- - -)	-0.121	-2.41	0.0123	-1.47	0.00	-1.47
(——)	0.978	0.872	-0.107	0.585	-0.705	-0.120
(——)	3.34	2.54	2.59	1.56	-0.804	0.756

Problem Setup

$$
\begin{aligned}
\underset{h(t),\theta(t)}{\text{maximize}} \quad & \int_0^T \int_{\boldsymbol{\Gamma}} \boldsymbol{f} \cdot \boldsymbol{v} \, dS \, dt \\
\text{subject to} \quad & -\int_0^T \int_{\boldsymbol{\Gamma}} F_x \, dS \, dt \geq c \\
& h^{(k)}(t) = h^{(k)}(t+T) \\
& \theta^{(k)}(t) = \theta^{(k)}(t+T) \\
& \frac{\partial \boldsymbol{U}}{\partial t} + \nabla \cdot \boldsymbol{F}(\boldsymbol{U}, \nabla \boldsymbol{U}) = 0
\end{aligned}
$$

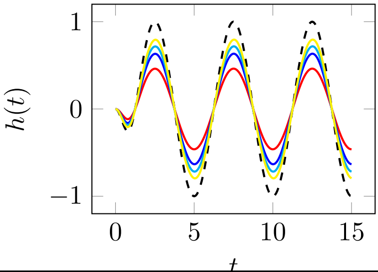


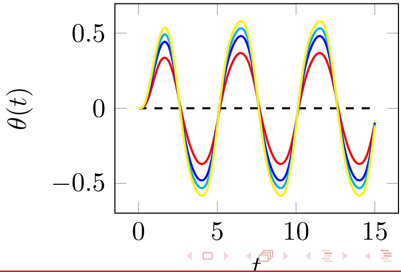
Airfoil schematic, kinematic description

- Non-zero freestream velocity
- $h(t)$, $\theta(t)$ discretized via phase/amplitude of *Fourier modes*
- Knots of cubic splines as optimization parameters, $\boldsymbol{\mu}$
- Black-box optimizer: SNOPT

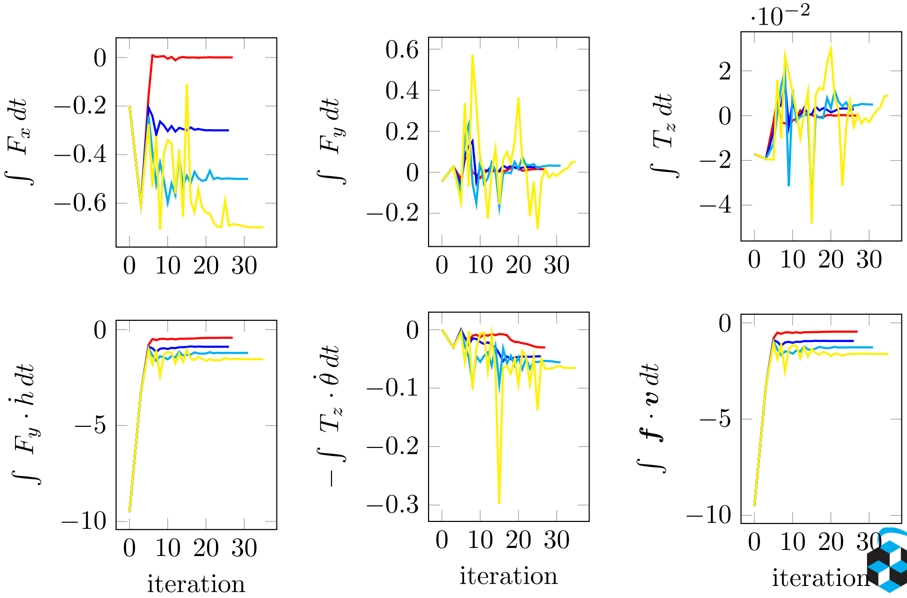
Optimization Setup

- Initial guess (- - -)
  - $h(t) = -\cos(0.4\pi t/T)$
  - $\theta(t) = 0$
- Optimization 1 (red line)
  - $c = 0.0$
  - $h(t)$, $\theta(t)$ parametrized (Fourier)
- Optimization 2 (blue line)
  - $c = 0.3$
  - $h(t)$, $\theta(t)$ parametrized (Fourier)
- Optimization 3 (light blue line)
  - $c = 0.5$
  - $h(t)$, $\theta(t)$ parametrized (Fourier)
- Optimization 4 (yellow line)
  - $c = 0.7$
  - $h(t)$, $\theta(t)$ parametrized (Fourier)

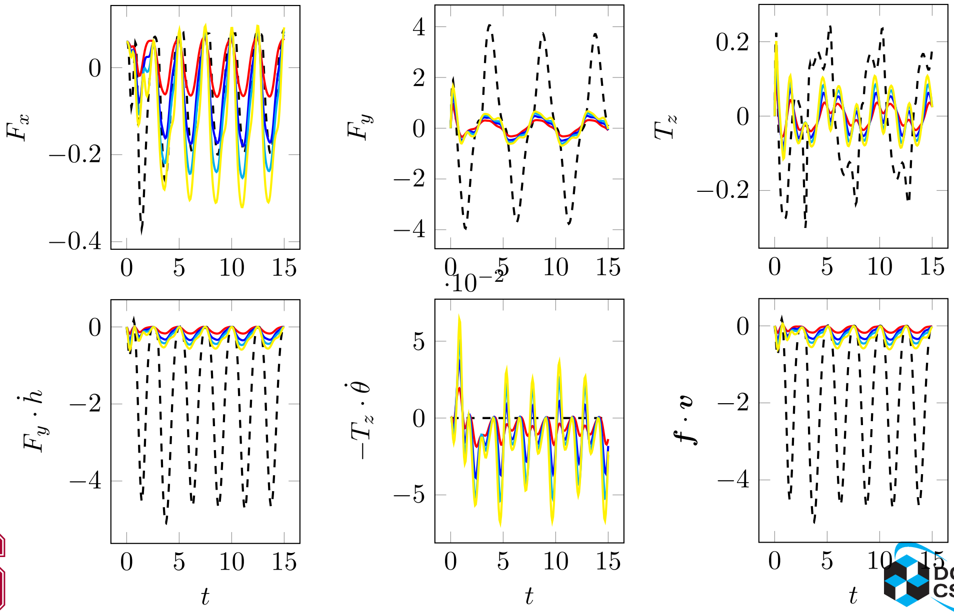




Optimization Convergence



Output Functional Comparison



Optimization Results: Vorticity Field History

Initial Guess (- - -)

$c = 0.0$ (——)

$c = 0.7$ (——)

Play/Pause

Summary

	$\int F_x \, dt$	$\int F_y \, dt$	$\int T_z \, dt$	$\int F_y \cdot \dot{h} \, dt$	$-\int T_z \cdot \dot{\theta} \, dt$	$\int \boldsymbol{f} \cdot \boldsymbol{v} \, dt$
Initial (- - -)	-0.198	-0.0447	-0.0172	-9.51	0.0	-9.51
$c = 0.0$ (—)	0.0	0.0142	0.0	-0.425	-0.0303	-0.455
$c = 0.3$ (—)	-0.3	0.0245	0.00319	-0.894	-0.0459	-0.940
$c = 0.5$ (—)	-0.5	0.0319	0.00501	-1.22	-0.0557	-1.27
$c = 0.7$ (—)	-0.7	0.0510	0.00897	-1.55	-0.0650	-1.61

Shape Optimization

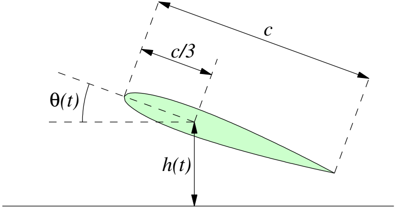
- Recall formula for reconstruction output gradient from primal/dual variables

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial F}{\partial \boldsymbol{\mu}} - \boldsymbol{\lambda}^{(0)^T} \frac{\partial \boldsymbol{u}_0}{\partial \boldsymbol{\mu}} - \sum_{n=1}^{N_t} \Delta t_n \sum_{i=1}^{s} \boldsymbol{\kappa}_i^{(n)^T} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_i^{(n)})$$

- Dependence on sensitivity of initial condition, $\frac{\partial \boldsymbol{u}_0}{\partial \boldsymbol{\mu}}$
  - Non-zero if $\boldsymbol{u}_0(\boldsymbol{\mu})$ is *steady-state* for a $\boldsymbol{\mu}$-parametrized shape
  - $\frac{\partial \boldsymbol{u}_0}{\partial \boldsymbol{\mu}}$ computed via standard sensitivity analysis for steady-state problems OR
  - $\boldsymbol{\lambda}^{(0)^T} \frac{\partial \boldsymbol{u}_0}{\partial \boldsymbol{\mu}}$ computed directly via standard adjoint method for steady-state problems
- This complication is circumvented in this work by chosing a zero freestream $\implies \boldsymbol{u}(\boldsymbol{\mu}) = 0$

Problem Setup

$$\begin{aligned} \underset{\boldsymbol{w}}{\text{maximize}} \quad & \int_0^T \int_{\boldsymbol{\Gamma}} \boldsymbol{f} \cdot \boldsymbol{v} \, dS \, dt \\ \text{subject to} \quad & \frac{\partial \boldsymbol{U}}{\partial t} + \nabla \cdot \boldsymbol{F}(\boldsymbol{U}, \nabla \boldsymbol{U}) = 0 \end{aligned}$$

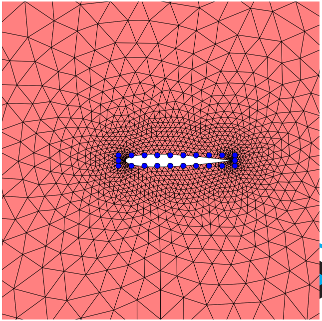


Airfoil schematic, kinematic description

- Radial basis function parametrization

$$\boldsymbol{X}' = \boldsymbol{X} + \boldsymbol{v} + \sum \boldsymbol{w}_i \Phi(||\boldsymbol{X} - \boldsymbol{c}_i||)$$

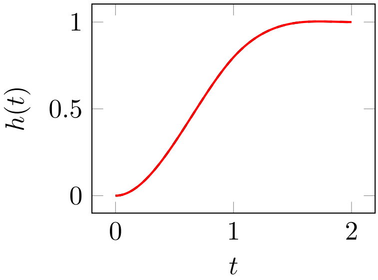
- *Zero* freestream velocity
- $h(t)$, $\theta(t)$ *prescribed*
- Black-box optimizer: SNOPT

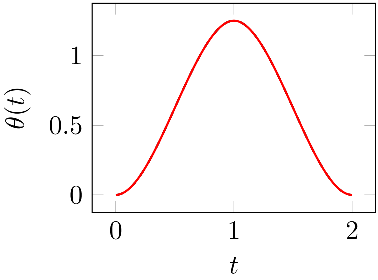


Optimization Setup

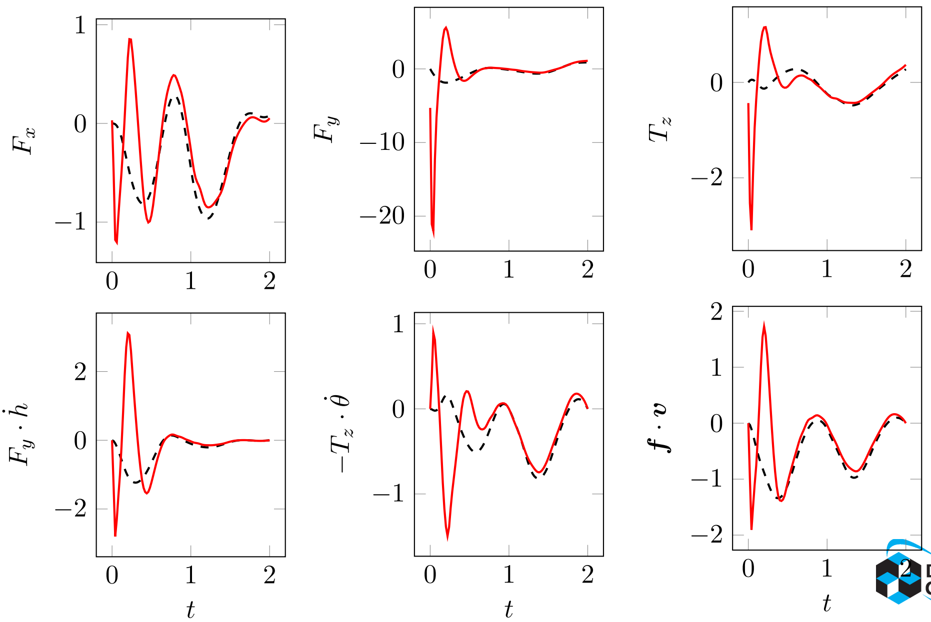
- Initial guess (- - -)
  - $h_0(t)$, $\theta_0(t)$ prescribed
  - $\boldsymbol{w} = \mathbf{0}$

- Optimization 1 (——)
  - $h_0(t)$, $\theta_0(t)$ prescribed
  - $\boldsymbol{w}$ variable





Output Functional Comparison



Optimization Results: Vorticity Field History

Initial Guess (- - -)

Optimal (——)

Play/Pause

Summary

	$\int F_x \, dt$	$\int F_y \, dt$	$\int T_z \, dt$	$\int F_y \cdot \dot{h} \, dt$	$-\int T_z \cdot \dot{\theta} \, dt$	$\int \boldsymbol{f} \cdot \boldsymbol{v} \, dt$
Initial (- - -)	-0.634	-0.727	-0.138	-0.526	-0.484	-1.01
Optimal (——)	-0.461	-0.959	-0.183	-0.145	-0.465	-0.609

Conclusion

- A high-order DG-DIRK discretization of general conservation laws with a mapping-based ALE formulation for deforming domains
- A fully-discrete adjoint method for computing gradients of output functionals and constraints in optimization problems
- Framework demonstrated on the computation of energetically optimal motions of a 2D airfoil in a flow field with constraints
- **Poster**: Unsteady PDE-Constrained Optimization using High-Order DG-FEM

Initial Guess

$h_0(t),\ \theta^*(t)$

$h^*(t),\ \theta^*(t)$

Future Work

- Application of the method to real-world **3D problems**
- Extension of the method to **multiphysics** problems, such as FSI
- Extension of the method to **chaotic** problems, such as LES flows, where care must be taken to ensure the sensitivities are well-defined
- Incorporation of **adaptive model reduction** technology to further reduce the cost of unsteady optimization