High-Order Methods for Optimization and Control of Conservation Laws on Deforming Domains

Matthew J. Zahr
Stanford University

Collaborators: Per-Olof Persson (UCB), Charbel Farhat (Stanford U)

Applied Mathematics Seminar, UC Berkeley
Wednesday, September 30, 2015

Questions ...

- How to flap a symmetric, 2D body such that the time-averaged thrust is identically 0?
- Among the kinematically-admissible, zero-thrust flapping motions, which requires the least energy to perform?

Play/Pause

Questions ...

- How to flap a symmetric, 2D body such that the time-averaged thrust is identically 0?
- Among the kinematically-admissible, zero-thrust flapping motions, which requires the least energy to perform?

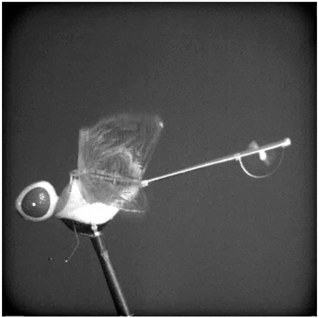
Energy = -9.51	Energy = -0.455	Energy = -1.61
Thrust = 0.198	Thrust = 0.0	Thrust = 0.7



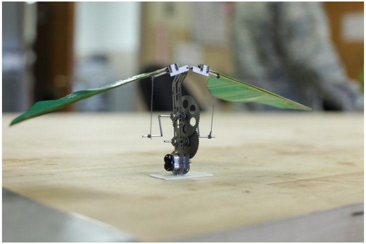
Play/Pause

Harder Question

- What is the energetically-optimal flapping motion of these systems?
- Constraints
  - time-average thrust = 0
  - time-average lift = weight of body and payload
  - stability constraints
  - structural constraints



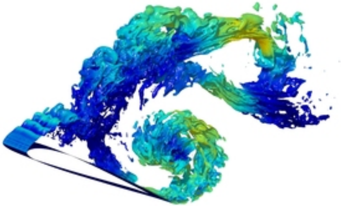
Dragonfly Experiment
(A. Song, Brown U)



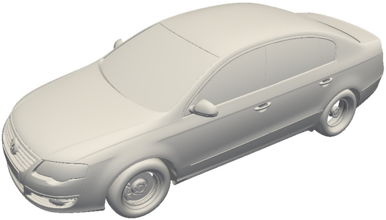
Micro Aerial Vehicle

Harder Question

- What is the optimal shape for each system?



LES Flow past Airfoil



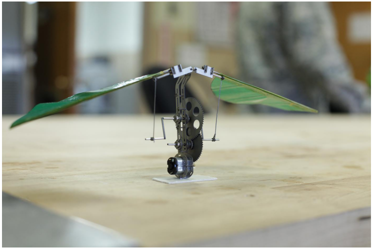
Passat

Vertical Windmill

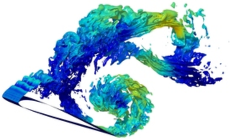
Time-Dependent PDE-Constrained Optimization

- Optimization of systems that are inherently dynamic or without a steady-state solution
- **Optimal control** of body immersed in a fluid
  - Goal: Determine **kinematics** that minimizes a cost functional, subject to constraints
- **Shape optimization** of body in turbulent flow
  - Goal: Determine **shape** that minimizes a cost functional, subject to constraints

Vertical Windmill



Micro Aerial Vehicle



LES Flow past Airfoil

Problem Formulation

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

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

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

ALE Description of Conservation Law

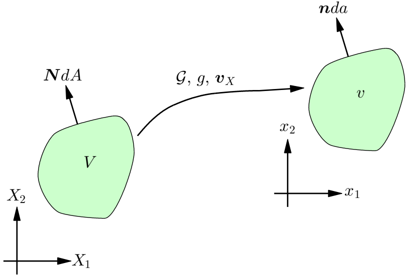
- Introduce map from fixed reference domain V to physical, deformable (parametrized) domain $v(\boldsymbol{\mu}, t)$
- A point $\boldsymbol{X} \in V$ is mapped to $\boldsymbol{x}(\boldsymbol{\mu}, t) = \mathcal{G}(\mathbf{X}, \boldsymbol{\mu}, t) \in v(\boldsymbol{\mu}, t)$

- Introduce transformation

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

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

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

Spatial Discretization: Discontinuous Galerkin

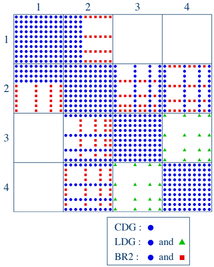
- Re-write conservation law as first-order system

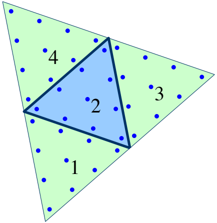
$$\left.\frac{\partial \boldsymbol{U}_{\boldsymbol{X}}}{\partial t}\right|_{\boldsymbol{X}} + \nabla_{\boldsymbol{X}} \cdot \boldsymbol{F}_{\boldsymbol{X}}(\boldsymbol{U}_{\boldsymbol{X}},\ \boldsymbol{Q}_{\boldsymbol{X}}) = 0$$
$$\boldsymbol{Q}_{\boldsymbol{X}} - \nabla_{\boldsymbol{X}} \boldsymbol{U}_{\boldsymbol{X}}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

- 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 → **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

Globally High-Order Discretization

- Fully-Discrete Conservation Law

$$
\begin{aligned}
\boldsymbol{u}^{(0)} &= \boldsymbol{u}_0(\boldsymbol{\mu}) \\
\boldsymbol{u}^{(n)} &= \boldsymbol{u}^{(n-1)} + \sum_{i=1}^{s} b_i \boldsymbol{k}_i^{(n)} \\
\boldsymbol{u}_i^{(n)} &= \boldsymbol{u}^{(n-1)} + \sum_{j=1}^{i} a_{ij} \boldsymbol{k}_j^{(n)} \\
\mathbb{M}\boldsymbol{k}_i^{(n)} &= \Delta t_n \boldsymbol{r}\left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_i \Delta t_n\right)
\end{aligned}
$$

- Fully-Discrete Output Functional

$$
F(\boldsymbol{u}^{(0)}, \dots, \boldsymbol{u}^{(N_t)}, \boldsymbol{k}_1^{(1)}, \dots, \boldsymbol{k}_s^{(N_t)}, \boldsymbol{\mu})
$$

Adjoint Method for PDE-Constrained Optimization

- *Continuous* PDE-constrained optimization problem

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

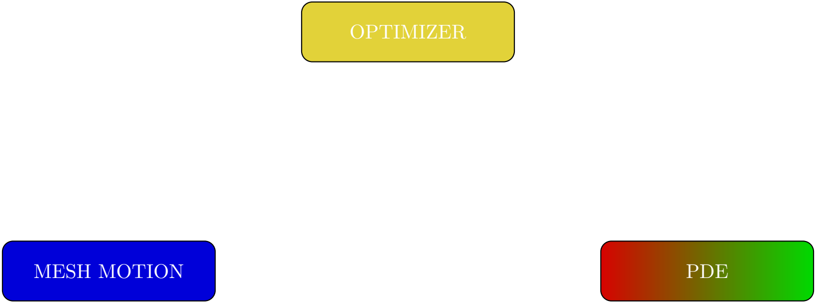
- Replace conservation law and functional with fully-discrete high-order numerical approximation

Adjoint Method for PDE-Constrained Optimization

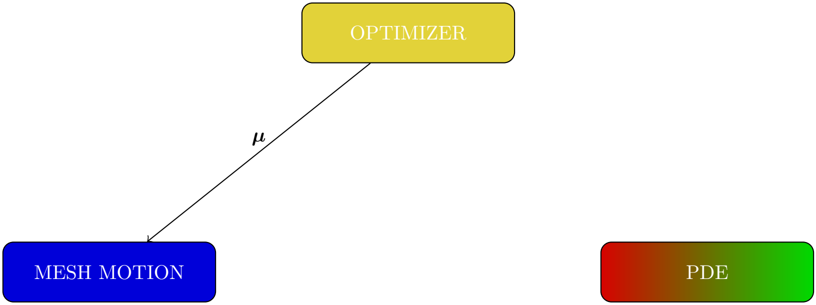
- *Fully-discrete* PDE-constrained optimization problem

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

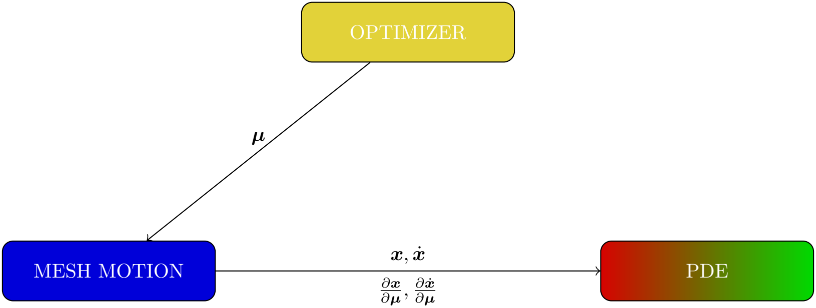
Generalized Reduced-Gradient Approach



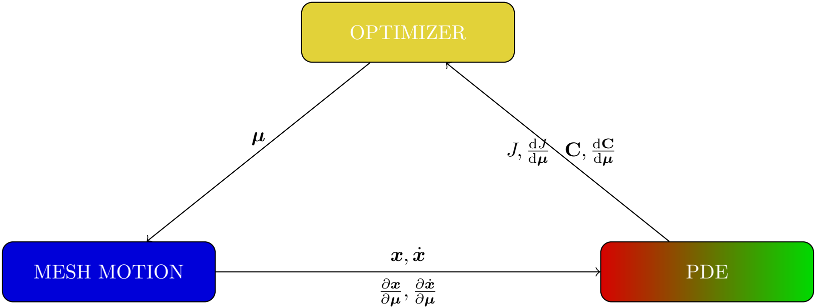
Generalized Reduced-Gradient Approach



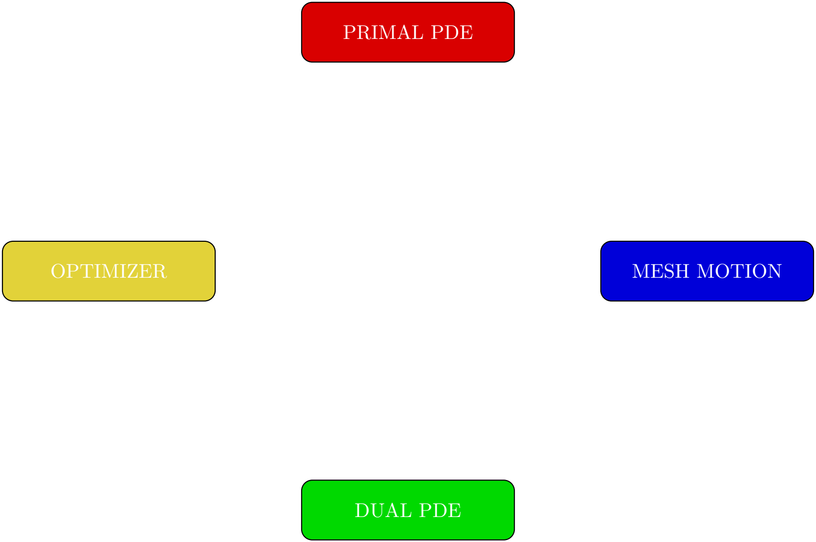
Generalized Reduced-Gradient Approach



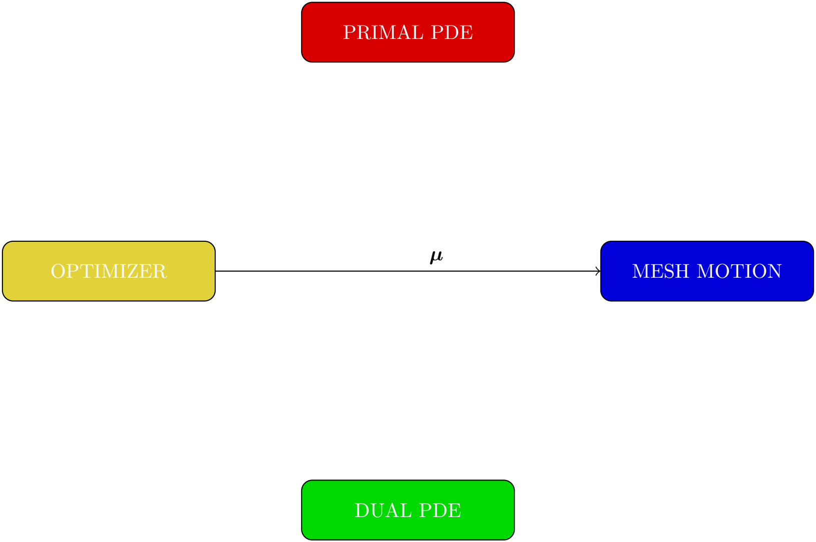
Generalized Reduced-Gradient Approach



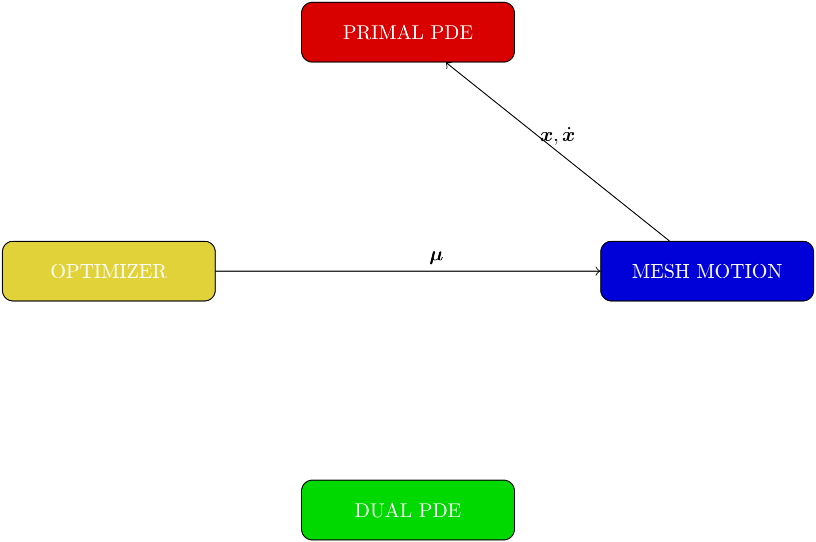
Generalized Reduced-Gradient Approach



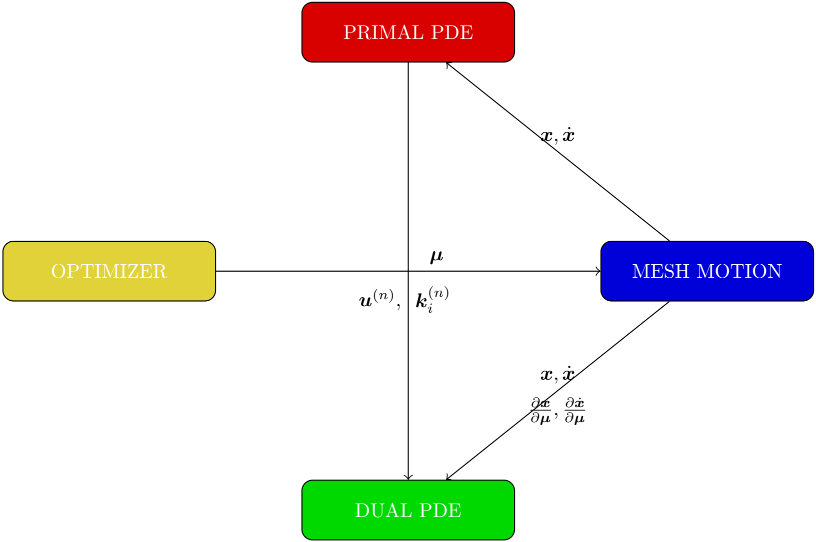
Generalized Reduced-Gradient Approach



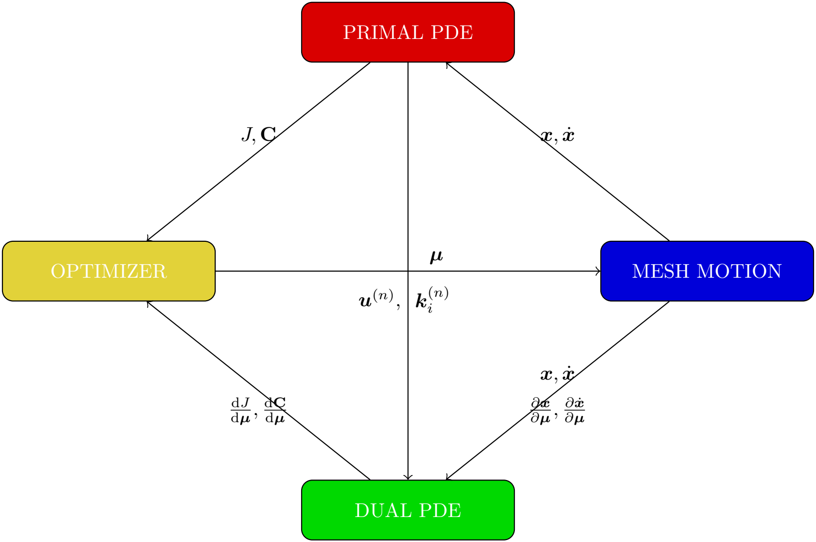
Generalized Reduced-Gradient Approach



Generalized Reduced-Gradient Approach



Generalized Reduced-Gradient Approach



Adjoint Method

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

Overview of Adjoint Derivation

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

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

- 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**

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

- 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**

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

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

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

Fully-Discrete Adjoint Equations: Dissection

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

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

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

- **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 $\dfrac{\partial \boldsymbol{k}_i^{(n)}}{\partial \boldsymbol{\mu}}$

Shape Optimization

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

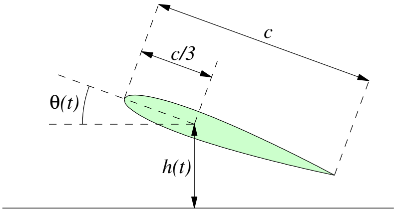
$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial F}{\partial \boldsymbol{\mu}} - \boldsymbol{\lambda}^{(0)\,T} \frac{\partial \boldsymbol{u}_0}{\partial \boldsymbol{\mu}} - \sum_{n=1}^{N_t} \Delta t_n \sum_{i=1}^{s} \boldsymbol{\kappa}_i^{(n)\,T} \frac{\partial \boldsymbol{r}}{\partial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

- 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 choosing a zero freestream $\implies \boldsymbol{u}_0(\boldsymbol{\mu}) = 0$

Problem Setup

$$\underset{\boldsymbol{w}}{\text{maximize}} \quad \int_0^T \int_{\boldsymbol{\Gamma}} \boldsymbol{f} \cdot \boldsymbol{v} \, dS \, dt$$

$$\text{subject to} \quad \frac{\partial \boldsymbol{U}}{\partial t} + \nabla \cdot \boldsymbol{F}(\boldsymbol{U}, \nabla \boldsymbol{U}) = 0$$

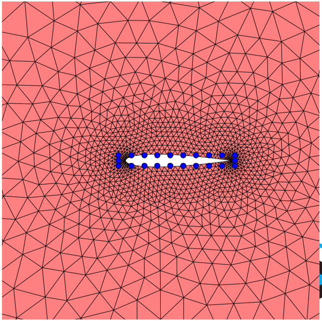


Airfoil schematic, kinematic description

- Radial basis function parametrization

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

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



Optimization Results: Vorticity Field History

- 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

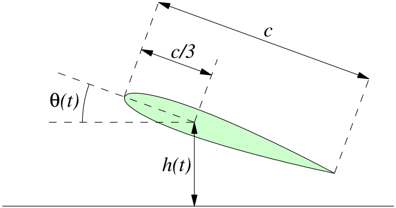
Play/Pause

Optimization 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

Problem Setup

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

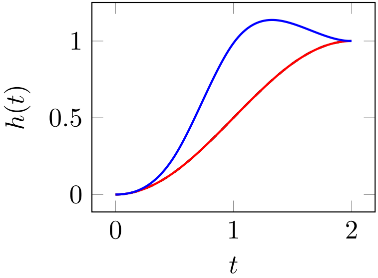


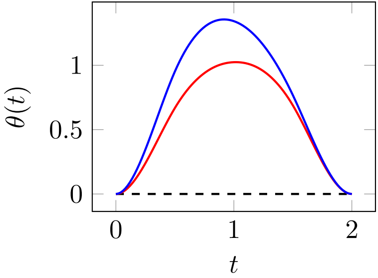
Airfoil schematic, kinematic description

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





Optimization Results: Vorticity Field History

Energy $= -1.47$	Energy $= -0.120$	Energy $= 0.756$
(- - -)	(——)	(——)

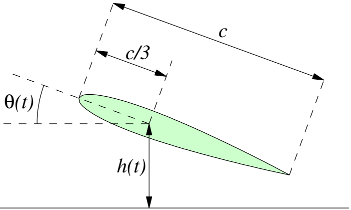
Play/Pause

Optimization 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

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

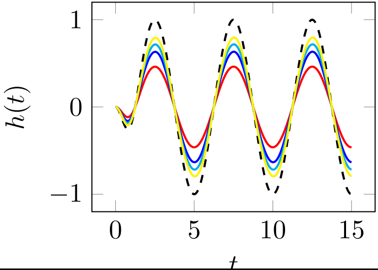


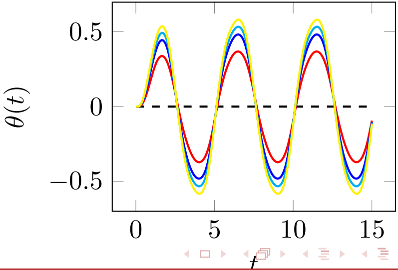
Airfoil schematic, kinematic description

- 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 Results: Vorticity Field History

Energy = -9.51 Thrust = 0.198	Energy = -0.455 Thrust = 0.0	Energy = -1.61 Thrust = 0.7
		
(- - -)	(—)	(—)

Play/Pause

Optimization 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
$c = 0.0$ (red)	0.0	0.0142	0.0	-0.425	-0.0303	-0.455
$c = 0.3$ (blue)	-0.3	0.0245	0.00319	-0.894	-0.0459	-0.940
$c = 0.5$ (light blue)	-0.5	0.0319	0.00501	-1.22	-0.0557	-1.27
$c = 0.7$ (yellow)	-0.7	0.0510	0.00897	-1.55	-0.0650	-1.61

Time-Periodic Solutions

- To properly optimize a cyclic, or periodic problem, need to simulate a **representative** period
- Necessary to avoid transients that will impact output functionals
- **Task**: Find initial condition, $\boldsymbol{u}_0$, such that flow is periodic, i.e. $\boldsymbol{u}^{(N_t)} = \boldsymbol{u}_0$

(- - -)

(——)

(——)

Play/Pause

Fully-Discrete Time-Periodic Solution

- Recall fully-discrete conservation law

$$\boldsymbol{u}^{(0)} = \boldsymbol{u}_0(\boldsymbol{\mu})$$
$$\boldsymbol{u}^{(n)} = \boldsymbol{u}^{(n-1)} + \sum_{i=1}^{s} b_i \boldsymbol{k}_i^{(n)}$$
$$\boldsymbol{u}_i^{(n)} = \boldsymbol{u}^{(n-1)} + \sum_{j=1}^{i} a_{ij} \boldsymbol{k}_j^{(n)}$$
$$\mathbb{M}\boldsymbol{k}_i^{(n)} = \Delta t_n \boldsymbol{r}\left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_{n-1} + c_i \Delta t_n\right)$$

- Time-periodicity is defined as

$$\boldsymbol{u}^{(N_t)}(\boldsymbol{u}_0) = \boldsymbol{u}_0$$

Shooting Method for Time-Periodic Solutions

- Apply Newton's method to solve nonlinear system of equations

$$\boldsymbol{R}(\boldsymbol{u}_0) = \boldsymbol{u}^{(N_t)}(\boldsymbol{u}_0) - \boldsymbol{u}_0 = 0$$

- Nonlinear iteration defined as

$$\boldsymbol{u}_0 \leftarrow \boldsymbol{u}_0 - \boldsymbol{J}(\boldsymbol{u}_0)^{-1}\boldsymbol{R}(\boldsymbol{u}_0)$$

where $\boldsymbol{J}(\boldsymbol{u}_0) = \dfrac{\partial \boldsymbol{u}^{(N_t)}}{\partial \boldsymbol{u}_0} - \boldsymbol{I}$

- $\dfrac{\partial \boldsymbol{u}^{(N_t)}}{\partial \boldsymbol{u}_0}$ is a **large, dense** matrix and expensive to construct
- Krylov method to solve $\boldsymbol{J}(\boldsymbol{u}_0)^{-1}\boldsymbol{R}(\boldsymbol{u}_0)$ only requires matrix-vector products

$$\boldsymbol{J}(\boldsymbol{u}_0)\boldsymbol{v} = \frac{\partial \boldsymbol{u}^{(N_t)}}{\partial \boldsymbol{u}_0}\boldsymbol{v} - \boldsymbol{v}$$

Fully-Discrete Sensitivity Method

- Direct differentiation of fully-discrete conservation law, and multiplication by $\boldsymbol{v}$, leads to the fully-discrete sensitivity equations

$$\frac{\partial \boldsymbol{u}^{(0)}}{\partial \boldsymbol{u}_0}\boldsymbol{v} = \boldsymbol{v}$$

$$\frac{\partial \boldsymbol{u}^{(n)}}{\partial \boldsymbol{u}_0}\boldsymbol{v} = \frac{\partial \boldsymbol{u}^{(n-1)}}{\partial \boldsymbol{u}_0}\boldsymbol{v} + \sum_{i=1}^{s} b_i \frac{\partial \boldsymbol{k}_i^{(n)}}{\partial \boldsymbol{u}_0}\boldsymbol{v}$$

$$\mathbb{M}\frac{\partial \boldsymbol{k}_i^{(n)}}{\partial \boldsymbol{u}_0}\boldsymbol{v} = \Delta t_n \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}\left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_i^{(n-1)}\right)\left[\frac{\partial \boldsymbol{u}^{(n-1)}}{\partial \boldsymbol{u}_0}\boldsymbol{v} + \sum_{j=1}^{i} a_{ij}\frac{\partial \boldsymbol{k}_j^{(n)}}{\partial \boldsymbol{u}_0}\boldsymbol{v}\right]$$

- Sensitivity variables: $\dfrac{\partial \boldsymbol{u}^{(n)}}{\partial \boldsymbol{u}_0}\boldsymbol{v}$, and $\dfrac{\partial \boldsymbol{k}_i^{(n)}}{\partial \boldsymbol{u}_0}\boldsymbol{v}$

Fully-Discrete Sensitivity Equations: Dissection

$$\frac{\partial \boldsymbol{u}^{(0)}}{\partial \boldsymbol{u}_0}\boldsymbol{v} = \boldsymbol{v}$$

$$\frac{\partial \boldsymbol{u}^{(n)}}{\partial \boldsymbol{u}_0}\boldsymbol{v} = \frac{\partial \boldsymbol{u}^{(n-1)}}{\partial \boldsymbol{u}_0}\boldsymbol{v} + \sum_{i=1}^{s} b_i \frac{\partial \boldsymbol{k}_i^{(n)}}{\partial \boldsymbol{u}_0}\boldsymbol{v}$$

$$\mathbb{M}\frac{\partial \boldsymbol{k}_i^{(n)}}{\partial \boldsymbol{u}_0}\boldsymbol{v} = \Delta t_n \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}\left(\boldsymbol{u}_i^{(n)},\ \boldsymbol{\mu},\ t_i^{(n-1)}\right)\left[\frac{\partial \boldsymbol{u}^{(n-1)}}{\partial \boldsymbol{u}_0}\boldsymbol{v} + \sum_{j=1}^{i} a_{ij}\frac{\partial \boldsymbol{k}_j^{(n)}}{\partial \boldsymbol{u}_0}\boldsymbol{v}\right]$$

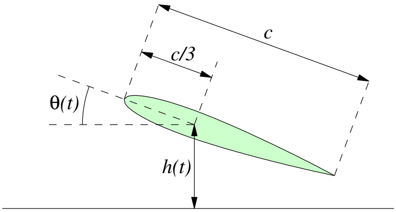
- **Linear** evolution equations solved **forward** in time for each $\boldsymbol{v}$
  - Requires solving linear systems of equations with $\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}$
  - Accurate solution of linear system required
- Primal state, $\boldsymbol{u}^{(n)}$, and stage, $\boldsymbol{k}_i^{(n)}$, required at each state/stage of sensitivity solve

Problem Setup

Find $\boldsymbol{u}_0$ such that

$$\boldsymbol{R}(\boldsymbol{u}_0) = \boldsymbol{u}^{(N_t)}(\boldsymbol{u}_0) - \boldsymbol{u}_0 = 0$$

and $\boldsymbol{u}^{(N_t)}$ is the solution of the fully-discrete conservation law at the final time step, N_t



Airfoil schematic, kinematic description

- $h(t)$, $\theta(t)$ prescribed
- Nonlinear solvers
  - Fixed-point iteration
  - Newton-Raphson method
- Linear solver: Unpreconditioned GMRES
- Fully-discrete sensitivity method used to compute $\dfrac{\partial \boldsymbol{u}^{(N_t)}}{\partial \boldsymbol{u}_0}\boldsymbol{v}$

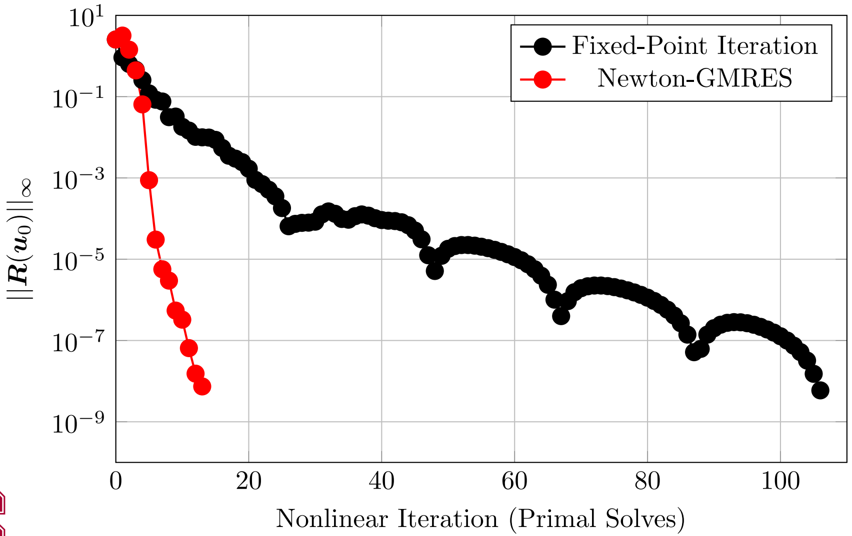
Time-Periodic Flow: Flapping Foil

Initial Guess for Newton-Krylov
Steady-State Flow

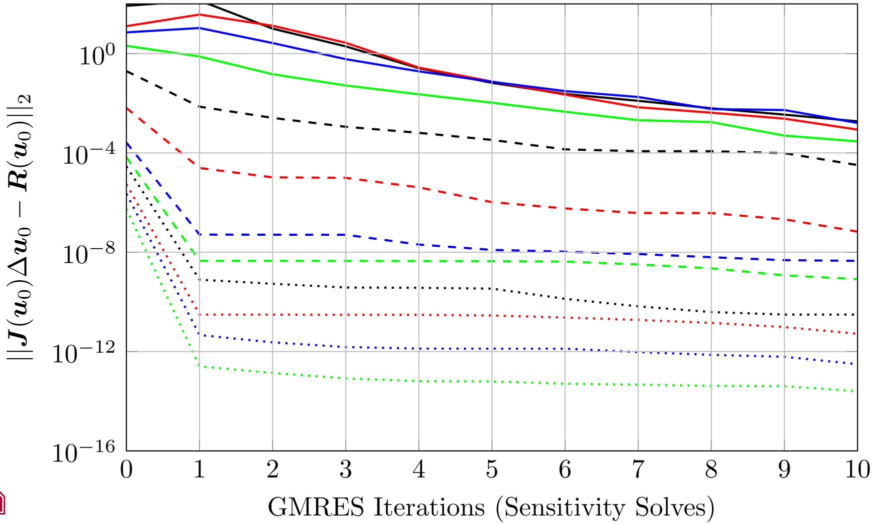
Solution of Newton-Krylov

Play/Pause

Nonlinear Solver Convergence



Linear Solver Convergence



Periodic PDE-Constrained Optimization

Recall *fully-discrete* PDE-constrained optimization problem

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

Periodic PDE-Constrained Optimization

Slight modification leads to fully-discrete periodic PDE-constrained optimization problem

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

Adjoint Method for Periodic PDE-Constrained Optimization

- Following identical procedure as for non-periodic case, the adjoint equations corresponding to the periodic conservation law are

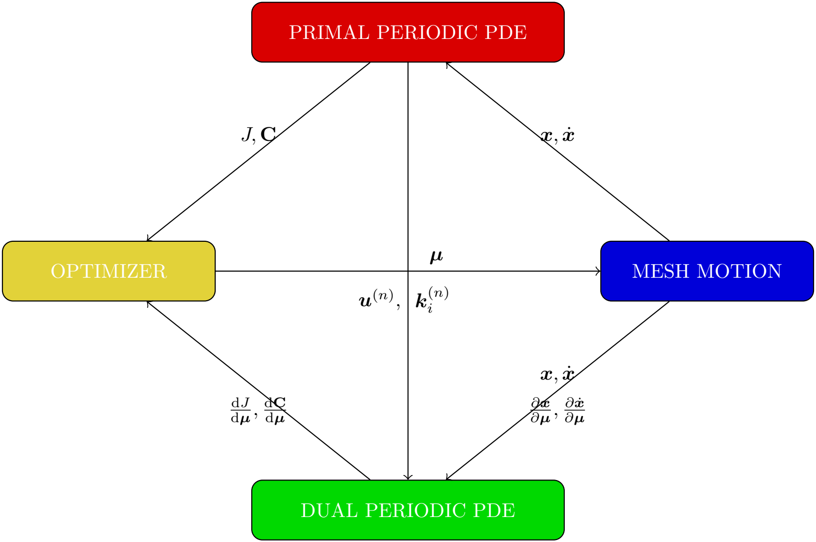
$$\boldsymbol{\lambda}^{(N_t)} = \boldsymbol{\lambda}^{(0)} + \frac{\partial F}{\partial \boldsymbol{u}^{(N_t)}}^T$$

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

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

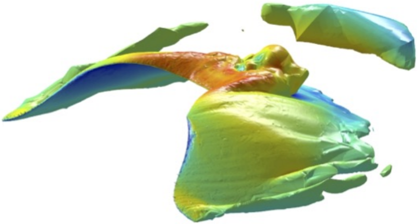
- Dual problem is also periodic
- Solve *linear, periodic* problem using Krylov shooting method

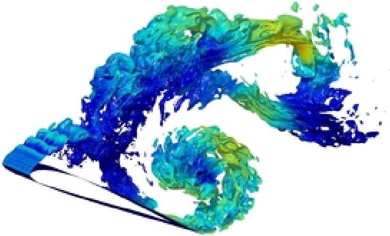
Generalized Reduced-Gradient Approach



Overview

- Utilized high-order DG-DIRK discretization of general conservation laws with a mapping-based ALE formulation for deforming domains
- Introduced fully-discrete adjoint method for computing gradients of output functionals
    - Framework demonstrated on the computation of energetically-optimal motions of a 2D airfoil in a flow field with constraints
- Introduced fully-discrete sensitivity equations and used Newton-Krylov shooting method to compute time-periodic flows
- **Next steps**: periodic optimization, 3D, multiphysics, model reduction



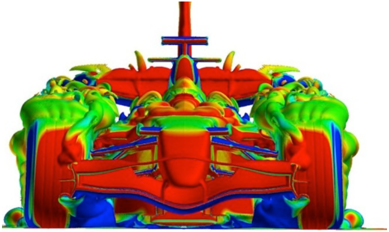


Problem Formulation

Goal: Rapidly solve PDE-constrained optimization problems of the form

$$\begin{array}{ll} \underset{\boldsymbol{u}\in\mathbb{R}^{N_{\boldsymbol{u}}},\ \boldsymbol{\mu}\in\mathbb{R}^{n_{\boldsymbol{\mu}}}}{\text{minimize}} & f(\boldsymbol{u},\boldsymbol{\mu}) \\ \text{subject to} & \boldsymbol{R}(\boldsymbol{u},\boldsymbol{\mu}) = 0 \end{array}$$

where $\boldsymbol{R} : \mathbb{R}^{N_{\boldsymbol{u}}} \times \mathbb{R}^{n_{\boldsymbol{\mu}}} \rightarrow \mathbb{R}^{N_{\boldsymbol{u}}}$ is the discretized (steady, nonlinear) PDE, $\boldsymbol{u}$ is the PDE state vector, $\boldsymbol{\mu}$ is the vector of parameters, and $N_{\boldsymbol{u}}$ is **assumed to be very large**.



Reduced-Order Model

- Model Order Reduction (MOR) assumption: *state vector lies in low-dimensional subspace*

$$\boldsymbol{u} \approx \boldsymbol{u}_r = \boldsymbol{\Phi}\boldsymbol{y} \qquad \Longrightarrow \qquad \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}} \approx \frac{\partial \boldsymbol{u}_r}{\partial \boldsymbol{\mu}} = \boldsymbol{\Phi}\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{\mu}}$$

  where $\boldsymbol{y} \in \mathbb{R}^{n_{\boldsymbol{y}}}$ are the reduced coordinates of $\boldsymbol{u}_r$ in the basis $\boldsymbol{\Phi} \in \mathbb{R}^{N_{\boldsymbol{u}} \times n_{\boldsymbol{y}}}$, and $n_{\boldsymbol{y}} \ll N_{\boldsymbol{u}}$
- Substitute assumption into High-Dimensional Model (HDM), $\boldsymbol{R}(\boldsymbol{u}, \boldsymbol{\mu}) = 0$

$$\boldsymbol{R}(\boldsymbol{\Phi}\boldsymbol{y}, \boldsymbol{\mu}) \approx 0$$

- Require projection of residual in low-dimensional *left subspace*, with basis $\boldsymbol{\Psi} \in \mathbb{R}^{N_{\boldsymbol{u}} \times n_{\boldsymbol{y}}}$ to be zero

$$\boxed{\boldsymbol{R}_r(\boldsymbol{y}, \boldsymbol{\mu}) = \boldsymbol{\Psi}^T \boldsymbol{R}(\boldsymbol{\Phi}\boldsymbol{y}, \boldsymbol{\mu}) = 0}$$

Definition of $\boldsymbol{\Phi}$: Proper Orthogonal Decomposition

- Perfect basis should satisify:

$$\{\boldsymbol{u}(\boldsymbol{\mu})\} \bigcup \left\{\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu})\right\} \subseteq \text{range } \boldsymbol{\Phi}$$

State-Sensitivity POD

- Collect state and sensitivity snapshots by sampling HDM

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{u}(\boldsymbol{\mu}_1) & \boldsymbol{u}(\boldsymbol{\mu}_2) & \cdots & \boldsymbol{u}(\boldsymbol{\mu}_n) \end{bmatrix}$$
$$\boldsymbol{Y} = \begin{bmatrix} \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}_1) & \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}_2) & \cdots & \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}_n) \end{bmatrix}$$

- Use Proper Orthogonal Decomposition to generate reduced bases from each *individually*, and concatenate to get ROB

$$\boldsymbol{\Phi} = \begin{bmatrix} \text{POD}(\boldsymbol{X}) & \text{POD}(\boldsymbol{Y}) \end{bmatrix}$$

Definition of $\mathbf{\Psi}$: Minimum-Residual ROM

- Recall ROM governing equation: $\boldsymbol{R}_r(\boldsymbol{y},\ \boldsymbol{\mu}) \equiv \boldsymbol{\Psi}^T \boldsymbol{R}(\boldsymbol{\Phi}\boldsymbol{y},\ \boldsymbol{\mu}) = 0$
- Standard options for choice of left basis $\boldsymbol{\Psi}$
  - Galerkin: $\boldsymbol{\Psi} = \boldsymbol{\Phi}$
  - Least-Squares Petrov-Galerkin (LSPG): $\boldsymbol{\Psi} = \dfrac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}}\boldsymbol{\Phi}$

Minimum-Residual Property

A ROM possesses the minimum-residual property if $\boldsymbol{R}_r(\boldsymbol{y},\ \boldsymbol{\mu}) = 0$ is equivalent to the optimality condition of

$$\underset{\boldsymbol{y}\in\mathbb{R}^n}{\text{minimize}} \quad ||\boldsymbol{R}(\boldsymbol{\Phi}\boldsymbol{y},\ \boldsymbol{\mu})||_{\Theta}$$

for $\Theta \succ 0$

- LSPG possesses minimum-residual property
- Implications
  - Recover exact solution when basis not compressed
  - Monotonic improvement of solution as basis size increases

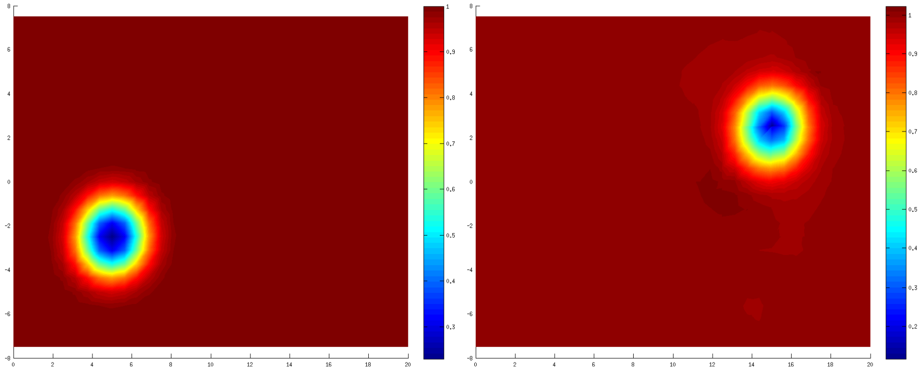
Nonlinear ROM Bottleneck

- Nonlinear ROM residual and Jacobian:

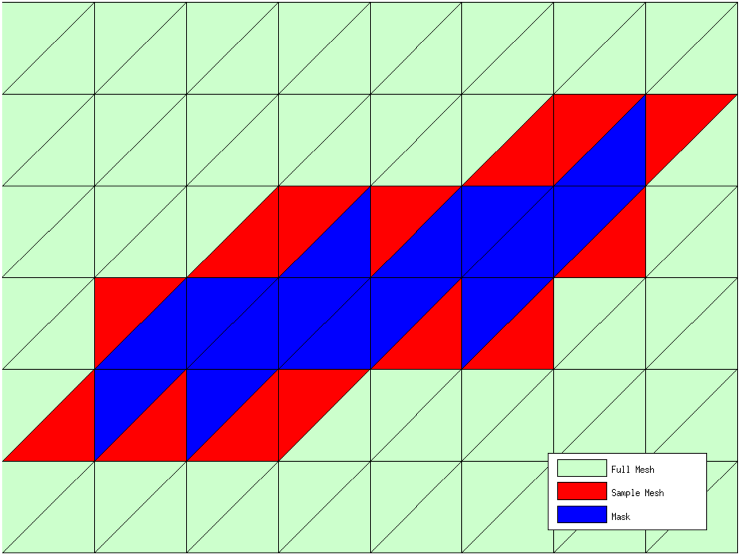
$$\boldsymbol{R}_r(\boldsymbol{y},\ \boldsymbol{\mu}) = \boldsymbol{\Psi}^T \boldsymbol{R}(\boldsymbol{\Phi}\boldsymbol{y},\ \boldsymbol{\mu})$$
$$\frac{\partial \boldsymbol{R}_r}{\partial \boldsymbol{y}}(\boldsymbol{y},\ \boldsymbol{\mu}) = \boldsymbol{\Psi}^T \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}}(\boldsymbol{\Phi}\boldsymbol{y},\ \boldsymbol{\mu})\boldsymbol{\Phi}$$

  - Large-scale quantity vs. Small-scale quantity
- To avoid large-scale operations to evaluate residual/Jacobian, introduce **gappy** approximation
  - Only requires evaluation of subset of rows of $\boldsymbol{R}$ and $\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}}$
  - In turn only requires subset of rows of $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$

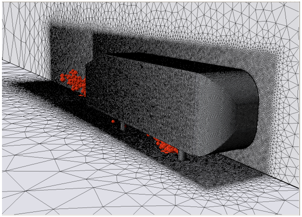
Gappy Approximation: Euler Vortex



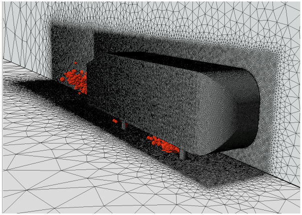
Gappy Approximation: Euler Vortex



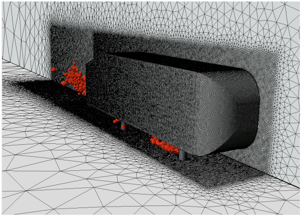
Gappy Approximation: Ahmed Body



(a) 253 sample nodes



(b) 378 sample nodes

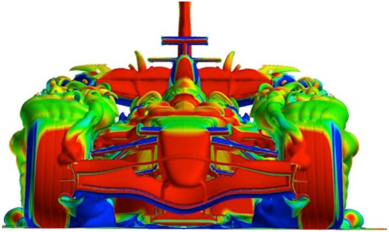


(c) 505 sample nodes

ROM-Constrained Optimization

Replace PDE-constrained optimization problem with ROM-constrained optimization problem

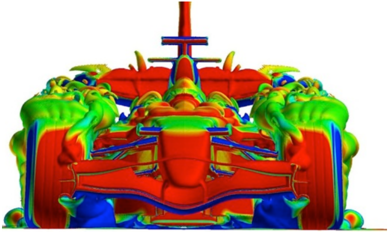
$$\begin{aligned} \underset{\boldsymbol{u}\in\mathbb{R}^{N_{\boldsymbol{u}}},\ \boldsymbol{\mu}\in\mathbb{R}^{n_{\boldsymbol{\mu}}}}{\text{minimize}} \quad & f(\boldsymbol{u}, \boldsymbol{\mu}) \\ \text{subject to} \quad & \boldsymbol{R}(\boldsymbol{u}, \boldsymbol{\mu}) = 0 \end{aligned}$$



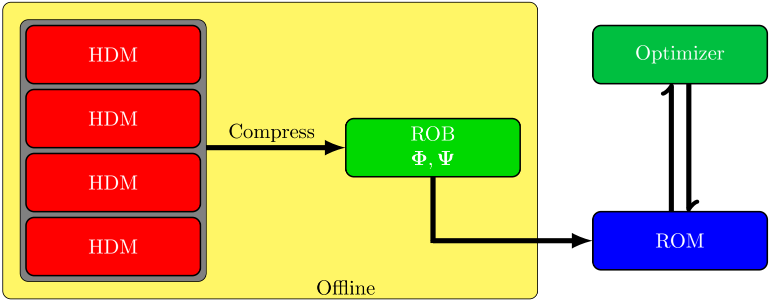
ROM-Constrained Optimization

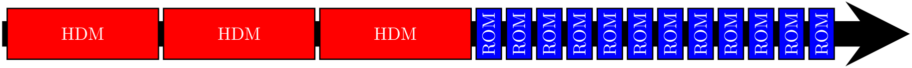
Replace PDE-constrained optimization problem with ROM-constrained optimization problem

$$\begin{aligned}
\underset{\boldsymbol{y}\in\mathbb{R}^{N_{\boldsymbol{y}}},\ \boldsymbol{\mu}\in\mathbb{R}^{n_{\boldsymbol{\mu}}}}{\text{minimize}} \quad & f(\boldsymbol{\Phi y}, \boldsymbol{\mu}) \\
\text{subject to} \quad & \boldsymbol{\Psi}^T \boldsymbol{R}(\boldsymbol{\Phi y}, \boldsymbol{\mu}) = 0
\end{aligned}$$

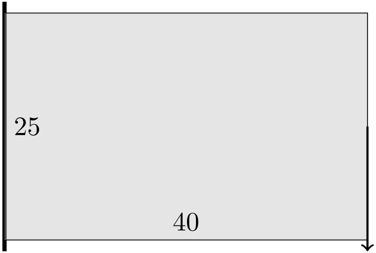


Offline/Online Approach to ROM-Constrained Optimization

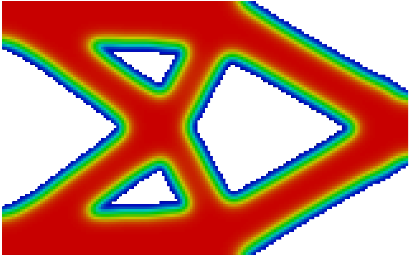




Numerical Experiment: Topology Optimization

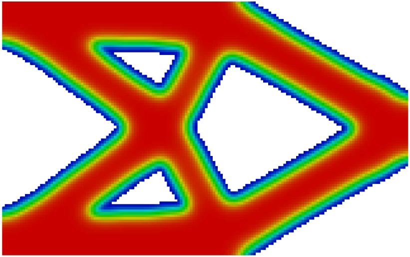


Cantilever Schematic

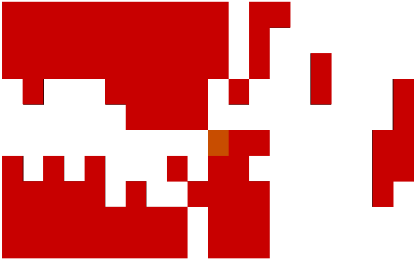


Optimal Solution

Numerical Experiment: Topology Optimization



Optimal Solution (HDM)

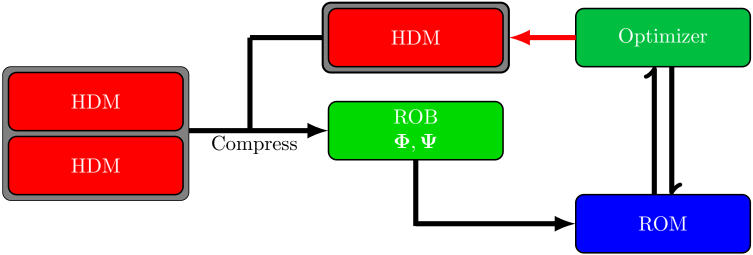


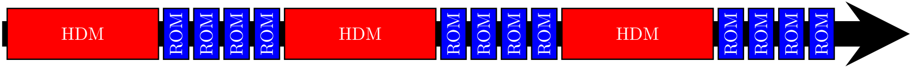
Optimal Solution (ROM)

HDM Solution	**ROB** Construction	**Greedy** Algorithm	**ROM** Optimization
2.84×10^3 s	5.48×10^4 s	1.67×10^5 s	30 s
1.26%	24.36%	74.37%	0.01%

HDM Optimization: 1.97×10^4 s

Adaptive Approach to ROM-Constrained Optimization





Nonlinear Trust-Region Framework

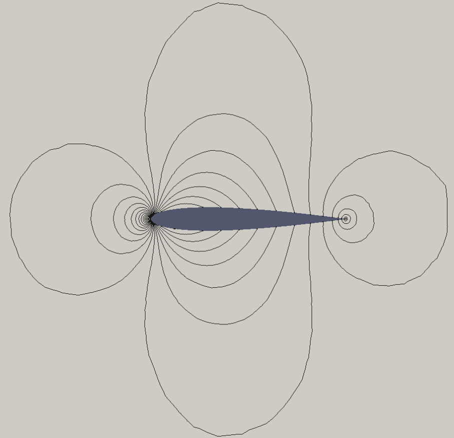
Adaptive Approach to ROM-Constrained Optimization

- Collect snapshots from HDM at *sparse sampling* of the parameter space
  - Initial condition for optimization problem
- Build ROB $\boldsymbol{\Phi}$ from sparse training
- Solve optimization problem

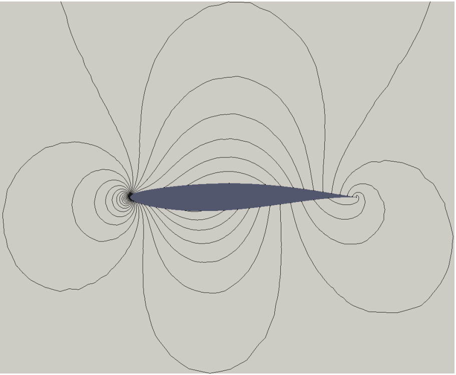
$$\begin{aligned} \underset{\boldsymbol{y}\in\mathbb{R}^n,\ \boldsymbol{\mu}\in\mathbb{R}^p}{\text{minimize}} \quad & f(\boldsymbol{\Phi}\boldsymbol{y}, \boldsymbol{\mu}) \\ \text{subject to} \quad & \boldsymbol{\Psi}^T \boldsymbol{R}(\boldsymbol{\Phi}\boldsymbol{y}, \boldsymbol{\mu}) = 0 \\ & \frac{1}{2}||\boldsymbol{R}(\boldsymbol{\Phi}\boldsymbol{y}, \boldsymbol{\mu})||_2^2 \leq \epsilon \end{aligned}$$

- Use solution of above problem to enrich training and repeat until convergence

Compressible, Inviscid Airfoil Inverse Design



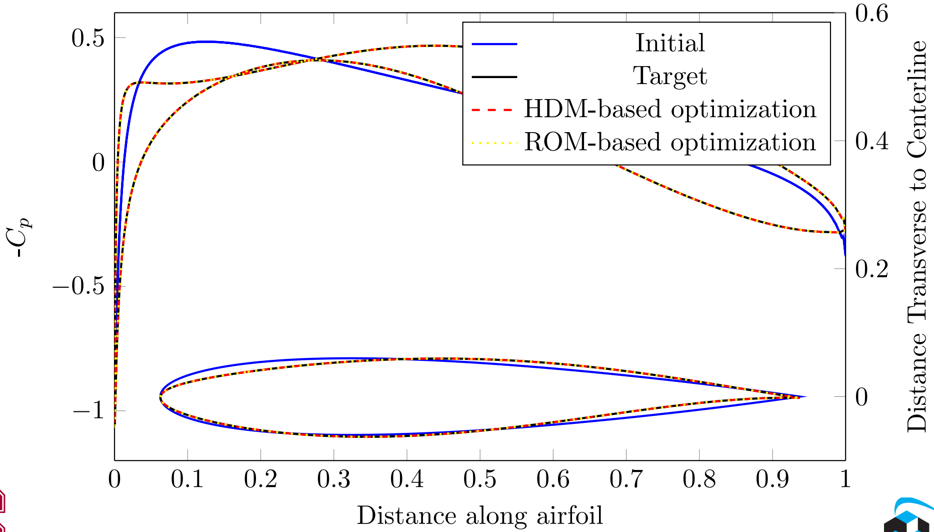
(a) NACA0012: Pressure field ($M_\infty = 0.5$, $\alpha = 0.0°$)



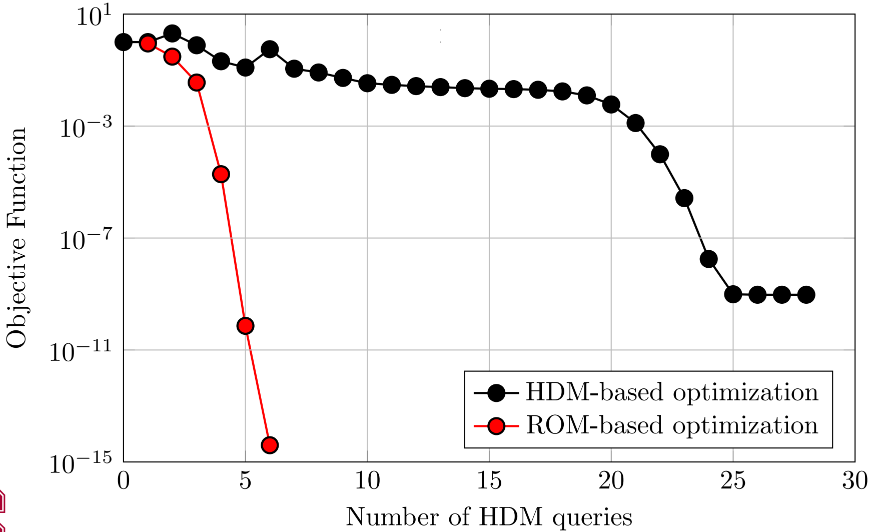
(b) RAE2822: Pressure field ($M_\infty = 0.5$, $\alpha = 0.0°$)

- Pressure discrepancy minimization (Euler equations)
  - Initial Configuration: NACA0012
  - Target Configuration: RAE2822

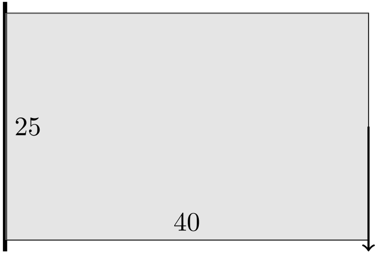
Optimization Results: How Close?



Optimization Results: How Fast?



Problem Setup

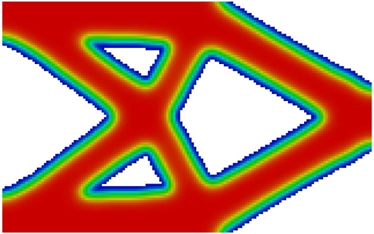


- 16000 8-node brick elements, 77760 dofs
- Total Lagrangian form, finite strain, StVK
- St. Venant-Kirchhoff material
- Sparse Cholesky linear solver (CHOLMOD)
- Newton-Raphson nonlinear solver
- Minimum compliance optimization problem

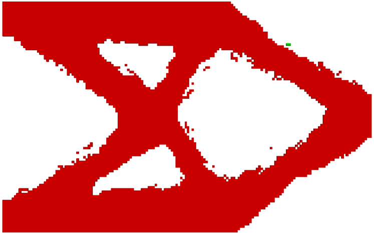
$$\begin{aligned} \underset{\mathbf{u}\in\mathbb{R}^{n_\mathbf{u}},\ \boldsymbol{\mu}\in\mathbb{R}^{n_{\boldsymbol{\mu}}}}{\text{minimize}} \quad & \mathbf{f}_{\text{ext}}{}^T\mathbf{u} \\ \text{subject to} \quad & V(\boldsymbol{\mu}) \leq \frac{1}{2}V_0 \\ & \mathbf{r}(\mathbf{u},\ \boldsymbol{\mu}) = 0 \end{aligned}$$

- Gradient computations: Adjoint method
- Optimizer: SNOPT
- Maximum ROM size: $k_\mathbf{u} \leq 5$

Optimal Solution Comparison



HDM



CTRPOD + $\boldsymbol{\Phi}_\mu$ adaptivity

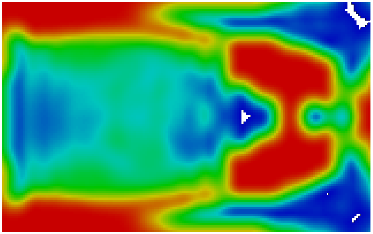
HDM Solution	**HDM** Gradient	**HDM** Optimization
7458s (450)	4018s (411)	8284s

HDM
Elapsed time = 19761s

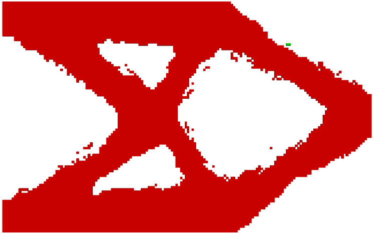
HDM Solution	**HDM** Gradient	**ROB** Construction	**ROM** Optimization
1049s (64)	88s (9)	727s (56)	39s (3676)

CTRPOD + $\boldsymbol{\Phi}_\mu$ adaptivity
Elapsed time = 2197s, Speedup ≈ 9x

Solution after 64 HDM Evaluations



HDM

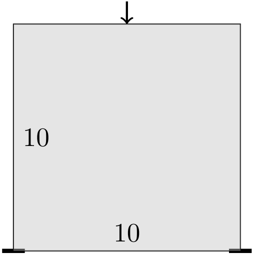


CTRPOD + Φ_μ adaptivity

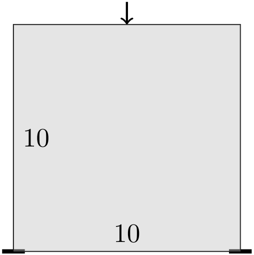
- CTRPOD + Φ_μ adaptivity: superior approximation to optimal solution than HDM approach after fixed number of HDM solves (64)
- Reasonable option to *warm-start* HDM topology optimization

CTRPOD + Φ_μ adaptivity

Problem Setup



(a) XY view



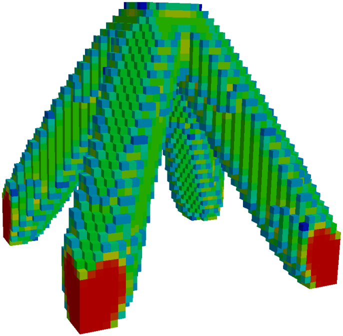
(b) XZ view

- 64000 8-node brick elements, 206715 dofs
- Total Lagrangian formulation, finite strain
- St. Venant-Kirchhoff material
- Jacobi-Preconditioned Conjugate Gradient
- Newton-Raphson nonlinear solver
- Minimum compliance optimization problem

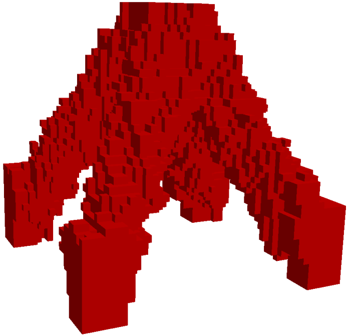
$$\begin{aligned} \underset{\mathbf{u}\in\mathbb{R}^{n_{\mathbf{u}}},\ \boldsymbol{\mu}\in\mathbb{R}^{n_{\boldsymbol{\mu}}}}{\text{minimize}} \quad & \mathbf{f}_{\text{ext}}{}^T\mathbf{u} \\ \text{subject to} \quad & V(\boldsymbol{\mu}) \leq 0.15 \cdot V_0 \\ & \mathbf{r}(\mathbf{u},\ \boldsymbol{\mu}) = 0 \end{aligned}$$

- Gradient computations: Adjoint method
- Optimizer: SNOPT
- Maximum ROM size: $k_{\mathbf{u}} \leq 5$

Optimal Solution Comparison



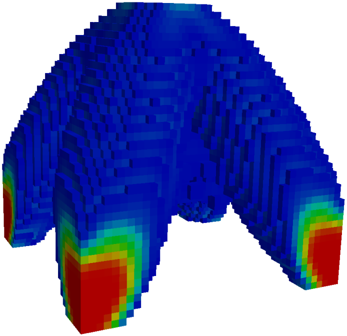
HDM



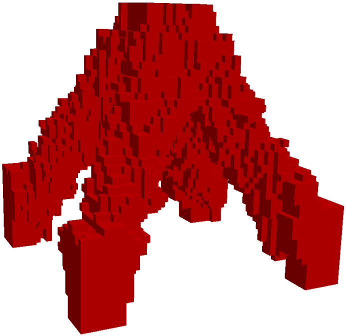
CTRPOD + Φ_μ adaptivity

- **HDM**, elapsed time = 179176s
- **CTRPOD+Φ_μ** adaptivity, elapsed time = 15208s
- Speedup $\approx 12\times$

Solution after 68 HDM Evaluations



HDM

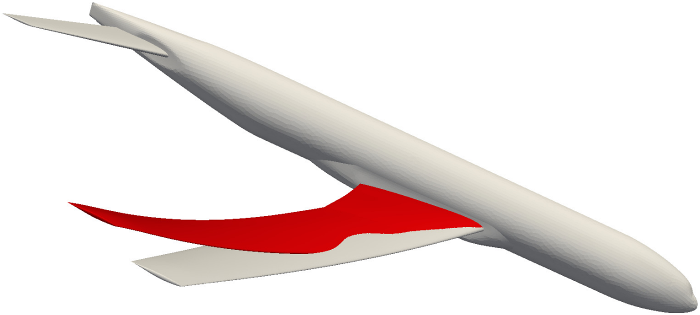


CTRPOD + Φ_μ adaptivity

- CTRPOD + Φ_μ adaptivity: superior approximation to optimal solution than HDM approach after fixed number of HDM solves (68)
- Reasonable option to *warm-start* HDM topology optimization

Conclusion

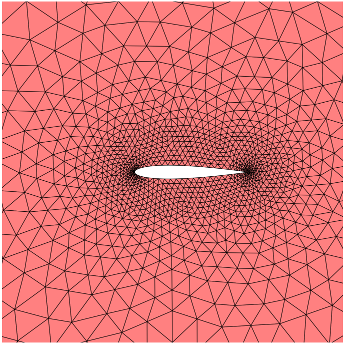
- Introduced adaptive, nonlinear trust-region framework for accelerating PDE-constrained optimization problems
- Demonstrated approach on canonicals problem from computational mechanics, including nonlinear topology optimization and aerodynamic shape optimization
  - Up to an order of magnitude improvement over standard approach to PDE-constrained optimization
- **Next steps**: incorporate hyperreduction, 3D, extension to unsteady



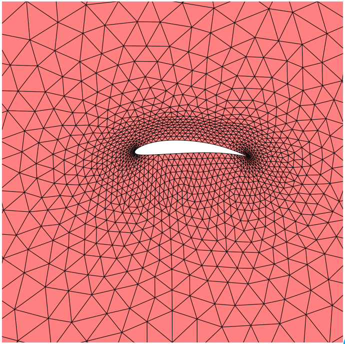
Domain Deformation

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

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



Undeformed Mesh

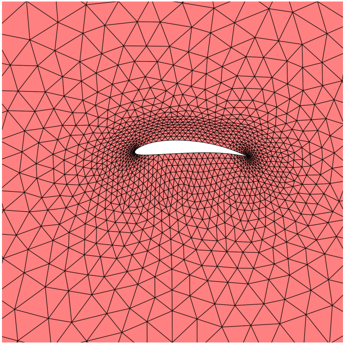


Shape Deformation

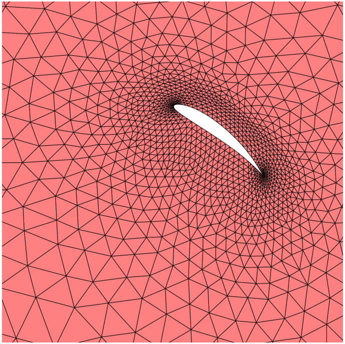
Domain Deformation

- Rigid body translation, $\boldsymbol{v}$, and rotation, $\boldsymbol{Q}$, applied to deformed configuration

$$\boldsymbol{X}'' = \boldsymbol{v} + \boldsymbol{Q}\boldsymbol{X}'$$



Shape Deformation



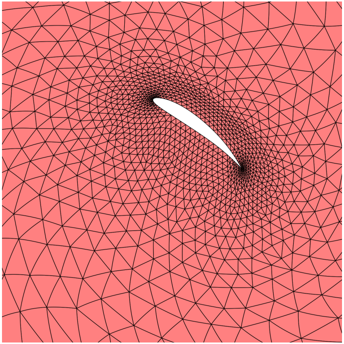
Shape Deformation, Rigid Motion

Domain Deformation

- Spatial blending between deformation with and without rigid body motion to avoid large velocities at far-field

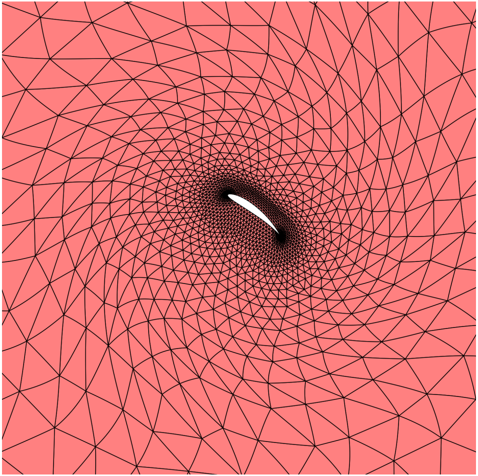
$$\boldsymbol{x} = b(\boldsymbol{X})\boldsymbol{X}' + (1 - b(\boldsymbol{X}))\boldsymbol{X}''$$

  - $b : \mathbb{R}^{n_{sd}} \to \mathbb{R}$ is a function that smoothly transitions from 0 inside a circle of radius R_1 to 1 outside circle of radius R_2



Blended Mesh

Domain Deformation



Blended Mesh

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

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

- Differentiation w.r.t. time leads to

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

- 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

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