Adjoint-Based Optimization of Time-Dependent Fluid-Structure Systems using a High-Order Discontinuous Galerkin Discretization

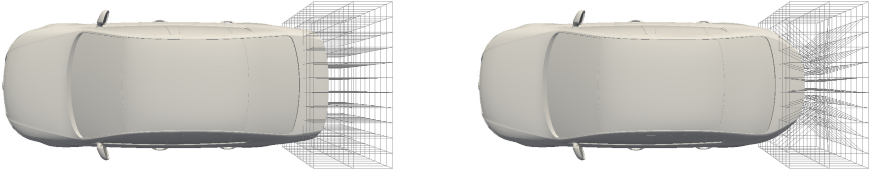
Matthew J. Zahr† and Per-Olof Persson

ECCOMAS High-Order Nonlinear Numerical Methods (HONOM)
University of Stuttgart, Stuttgart, Germany
March 27-31, 2017

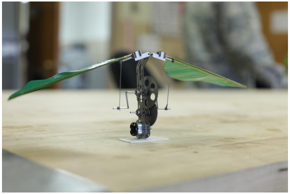
† Luis W. Alvarez Postdoctoral Fellow
Department of Mathematics
Lawrence Berkeley National Laboratory
University of California, Berkeley

PDE optimization is ubiquitous in science and engineering

Design: Find system that optimizes performance metric, satisfies constraints



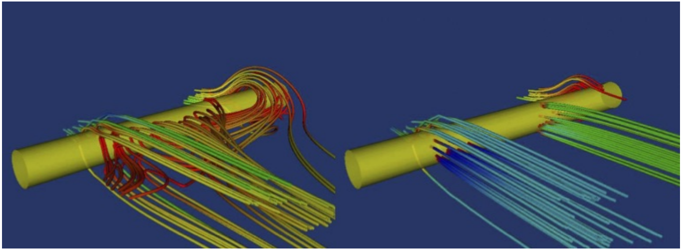
Aerodynamic shape design of automobile



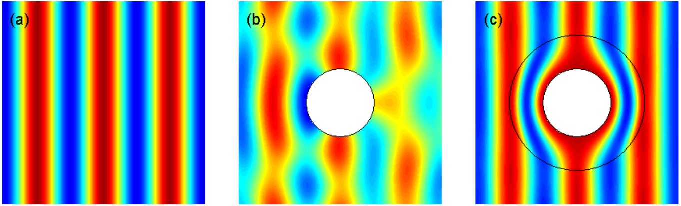
Optimal flapping motion of micro aerial vehicle

PDE optimization is ubiquitous in science and engineering

Control: Drive system to a desired state



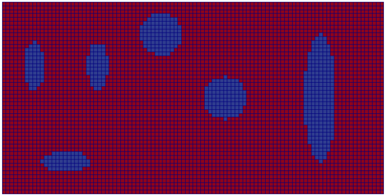
Boundary flow control

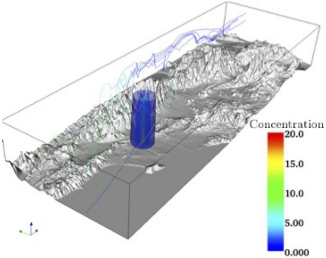


Metamaterial cloaking – electromagnetic invisibility

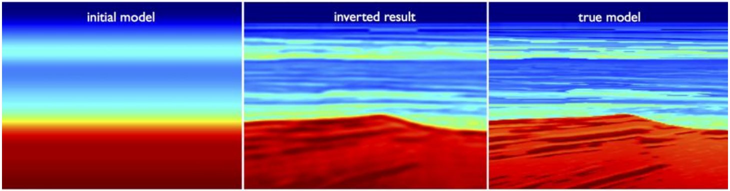
PDE optimization is ubiquitous in science and engineering

Inverse problems: Infer the problem setup given solution observations





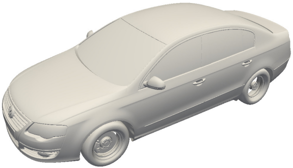
Left: Material inversion – find inclusions from acoustic, structural measurements
Right: Source inversion – find source of airborne contaminant from downstream measurements



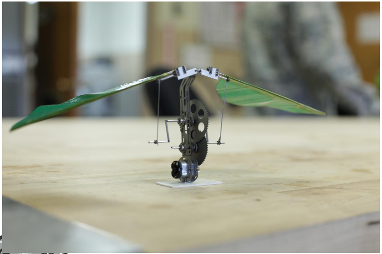
Full waveform inversion – estimate subsurface of Earth's crust from acoustic

Time-dependent PDE-constrained optimization

- Introduction of **fully discrete adjoint method** emanating from **high-order** discretization of governing equations
- **Time-periodicity** constraints
- Extension to high-order partitioned solver for **fluid-structure** interaction
- Applications: flapping flight, energy harvesting

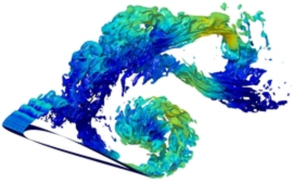


Volkswagen Passat



Micro Aerial Vehicle

Vertical Windmill



LES Flow past Airfoil

Unsteady PDE-constrained optimization 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

Nested approach to PDE-constrained optimization

PDE optimization requires repeated queries to primal and dual PDE

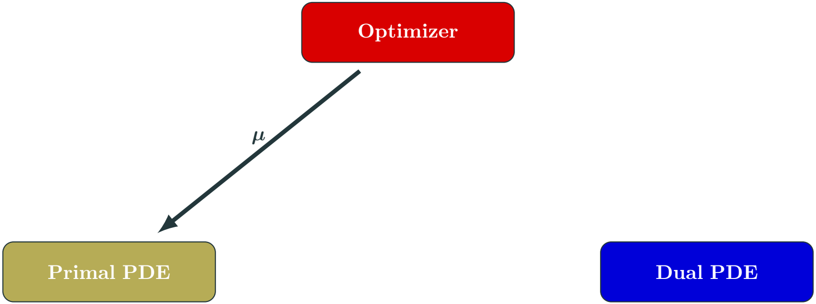
Optimizer

Primal PDE

Dual PDE

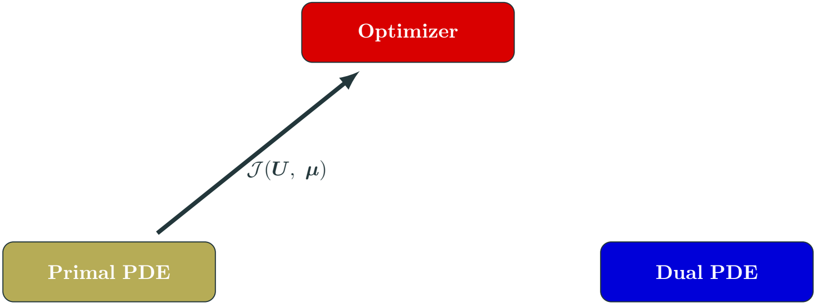
Nested approach to PDE-constrained optimization

PDE optimization requires repeated queries to primal and dual PDE



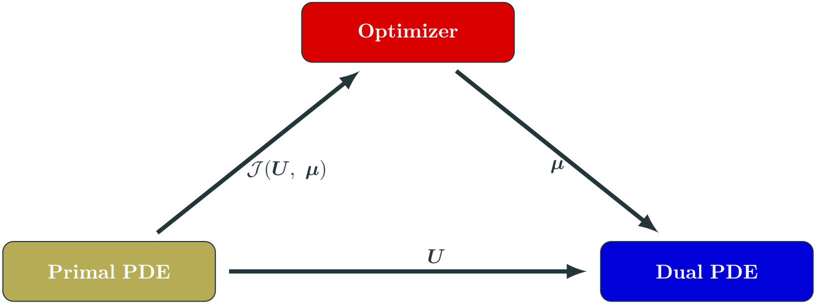
Nested approach to PDE-constrained optimization

PDE optimization requires repeated queries to primal and dual PDE



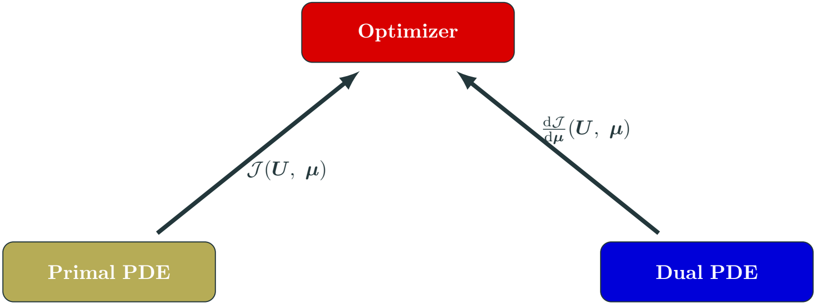
Nested approach to PDE-constrained optimization

PDE optimization requires repeated queries to primal and dual PDE



Nested approach to PDE-constrained optimization

PDE optimization requires repeated queries to primal and dual PDE



High-order discretization of PDE-constrained optimization

- *Continuous* PDE-constrained optimization problem

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

- *Fully discrete* PDE-constrained optimization problem

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

Highlights of globally high-order discretization

- **Arbitrary Lagrangian-Eulerian** formulation: Map, $\mathcal{G}(\cdot, \boldsymbol{\mu}, t)$, from physical $v(\boldsymbol{\mu}, t)$ to reference V

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

- **Space discretization**: discontinuous Galerkin

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

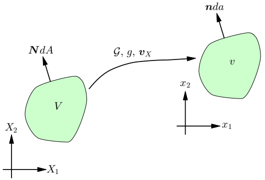
- **Time discretization**: diagonally implicit RK

$$\boldsymbol{u}_n = \boldsymbol{u}_{n-1} + \sum_{i=1}^{s} b_i \boldsymbol{k}_{n,i}$$

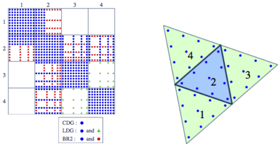
$$\boldsymbol{M}\boldsymbol{k}_{n,i} = \Delta t_n \boldsymbol{r}\left(\boldsymbol{u}_{n,i},\ \boldsymbol{\mu},\ t_{n,i}\right)$$

- **Quantity of interest**: solver-consistency

$$F(\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{N_t}, \boldsymbol{k}_{1,1}, \ldots, \boldsymbol{k}_{N_t,s})$$



Mapping-Based ALE



DG Discretization

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

Butcher Tableau for DIRK

Adjoint method to efficiently compute gradients of QoI

- Consider the *fully discrete* output functional $F(\boldsymbol{u}_n, \boldsymbol{k}_{n,i}, \boldsymbol{\mu})$
  - 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

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial F}{\partial \boldsymbol{\mu}} + \sum_{n=0}^{N_t} \frac{\partial F}{\partial \boldsymbol{u}_n} \frac{\partial \boldsymbol{u}_n}{\partial \boldsymbol{\mu}} + \sum_{n=1}^{N_t} \sum_{i=1}^{s} \frac{\partial F}{\partial \boldsymbol{k}_{n,i}} \frac{\partial \boldsymbol{k}_{n,i}}{\partial \boldsymbol{\mu}}$$

- The sensitivities, $\frac{\partial \boldsymbol{u}_n}{\partial \boldsymbol{\mu}}$ and $\frac{\partial \boldsymbol{k}_{n,i}}{\partial \boldsymbol{\mu}}$, are expensive to compute, requiring the solution of $n_{\boldsymbol{\mu}}$ linear evolution equations
- **Adjoint method**: alternative method for computing $\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}}$ that require one linear evolution equation for each quantity of interest, F

Adjoint equation derivation: outline

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

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

- Define **Lagrangian**

$$
\mathcal{L}(\boldsymbol{u}_n,\ \boldsymbol{k}_{n,i},\ \boldsymbol{\lambda}_n,\ \boldsymbol{\kappa}_{n,i}) = F - {\boldsymbol{\lambda}_0}^T \boldsymbol{R}_0 - \sum_{n=1}^{N_t} {\boldsymbol{\lambda}_n}^T \boldsymbol{R}_n - \sum_{n=1}^{N_t}\sum_{i=1}^{s} {\boldsymbol{\kappa}_{n,i}}^T \boldsymbol{R}_{n,i}
$$

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

$$
\frac{\partial \mathcal{L}}{\partial \boldsymbol{u}_n} = 0, \quad \frac{\partial \mathcal{L}}{\partial \boldsymbol{k}_{n,i}} = 0, \quad \frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}_n} = 0, \quad \frac{\partial \mathcal{L}}{\partial \boldsymbol{\kappa}_{n,i}} = 0
$$

Dissection of fully discrete adjoint equations

- **Linear** evolution equations solved **backward** in time
- **Primal** state/stage, $\boldsymbol{u}_{n,i}$ required at each state/stage of dual problem
- Heavily dependent on **chosen ouput**

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

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

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

- Gradient reconstruction via dual variables

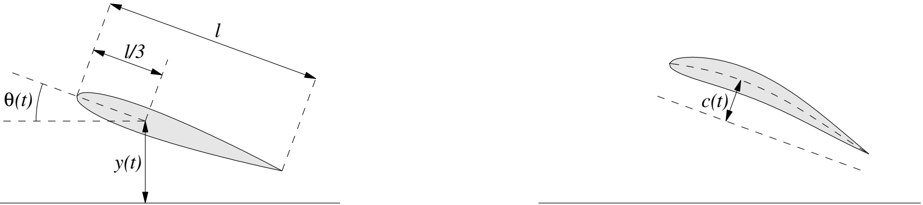
$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial F}{\partial \boldsymbol{\mu}} + \boldsymbol{\lambda}_0{}^T \frac{\partial \boldsymbol{g}}{\partial \boldsymbol{\mu}}(\boldsymbol{\mu}) + \sum_{n=1}^{N_t} \Delta t_n \sum_{i=1}^{s} \boldsymbol{\kappa}_{n,i}{}^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\mu}}\left(\boldsymbol{u}_{n,i},\ \boldsymbol{\mu},\ t_{n,i}\right)$$

[Zahr and Persson, 2016]

Energetically optimal flapping under x-impulse constraint

$$
\begin{aligned}
&\underset{\boldsymbol{\mu}}{\text{minimize}} \quad && -\int_{2T}^{3T}\int_{\boldsymbol{\Gamma}} \boldsymbol{f}\cdot\boldsymbol{v}\,dS\,dt \\
&\text{subject to} \quad && \int_{2T}^{3T}\int_{\boldsymbol{\Gamma}} \boldsymbol{f}\cdot\boldsymbol{e}_1\,dS\,dt = q \\
& && \boldsymbol{U}(\boldsymbol{x},0) = \boldsymbol{g}(\boldsymbol{x}) \\
& && \frac{\partial \boldsymbol{U}}{\partial t} + \nabla\cdot\boldsymbol{F}(\boldsymbol{U},\nabla\boldsymbol{U}) = 0
\end{aligned}
$$

- Isentropic, compressible, Navier-Stokes
- Re $=1000$, M $=0.2$
- $y(t)$, $\theta(t)$, $c(t)$ parametrized via periodic cubic splines
- Black-box optimizer: SNOPT



Airfoil schematic, kinematic description

Optimal control - fixed shape

Fixed shape, optimal Rigid Body Motion (RBM), varied x-impulse

Energy $=$ 9.4096
x-impulse $=$ -0.1766

Initial Guess

Energy $=$ 0.45695
x-impulse $=$ 0.000

Optimal RBM
$J_x = 0.0$

Energy $=$ 4.9475
x-impulse $=$ -2.500

Optimal RBM
$J_x = -2.5$

Optimal control, time-morphed geometry

Optimal Rigid Body Motion (RBM) and Time-Morphed Geometry (TMG), varied x-impulse

Initial Guess	Optimal RBM/TMG	Optimal RBM/TMG
Energy $=$ 9.4096	Energy $=$ 0.45027	Energy $=$ 4.6182
x-impulse $=$ -0.1766	x-impulse $=$ 0.000	x-impulse $=$ -2.500
	$J_x = 0.0$	$J_x = -2.5$

Optimal control, time-morphed geometry

Optimal Rigid Body Motion (RBM) and Time-Morphed Geometry (TMG),
x-impulse $= -2.5$

Initial Guess	Optimal RBM	Optimal RBM/TMG
Energy $=$ 9.4096	Energy $=$ 4.9476	Energy $=$ 4.6182
x-impulse $=$ -0.1766	x-impulse $=$ -2.500	x-impulse $=$ -2.500

Initial Guess

Optimal RBM
$J_x = -2.5$

Optimal RBM/TMG
$J_x = -2.5$

Energetically optimal flapping in three-dimensions

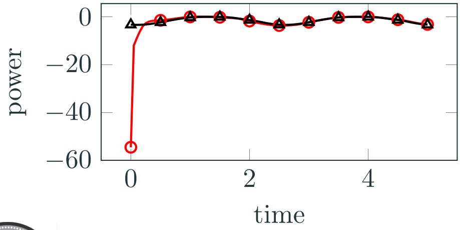
Energy $= 1.4459\text{e-}01$	Energy $= 3.1378\text{e-}01$
Thrust $= -1.1192\text{e-}01$	Thrust $= 0.0000\text{e+}00$

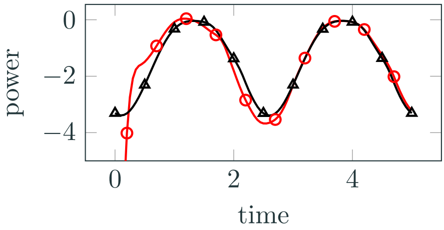
Time-periodic solutions desired when optimizing cyclic motion

- To properly optimize a cyclic, or periodic problem, need to simulate a **representative** period
- Necessary to avoid transients that will impact quantity of interest and may cause simulation to crash
- **Task**: Find initial condition, $\boldsymbol{u}_0$, such that flow is periodic, i.e. $\boldsymbol{u}_{N_t} = \boldsymbol{u}_0$

Time-periodic solutions desired when optimizing cyclic motion

Vorticity around airfoil with flow initialized from steady-state (left) and time-periodic flow (right)





Time history of power on airfoil of flow initialized from steady-state (—○—) and from a time-periodic solution (—△—)

Time-periodicity constraints in PDE-constrained optimization

Slight modification leads to fully discrete periodic PDE-constrained optimization

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

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

$$\boldsymbol{\lambda}_{N_t} = \boldsymbol{\lambda}_0 + \frac{\partial F}{\partial \boldsymbol{u}_{N_t}}^T$$

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

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

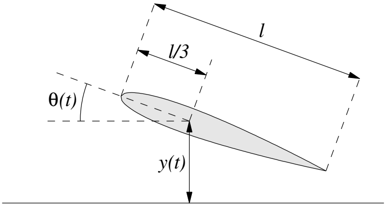
- Dual problem is also periodic
- Solve *linear, periodic* problem using Krylov shooting method

[Zahr et al., 2016]

Energetically optimal flapping: x-impulse, time-periodic

$$
\begin{aligned}
\underset{\boldsymbol{\mu}}{\text{minimize}} \quad & -\int_0^T \int_{\boldsymbol{\Gamma}} \boldsymbol{f} \cdot \boldsymbol{v} \, dS \, dt \\
\text{subject to} \quad & \int_0^T \int_{\boldsymbol{\Gamma}} \boldsymbol{f} \cdot \boldsymbol{e}_1 \, dS \, dt = q \\
& \boldsymbol{U}(\boldsymbol{x}, 0) = \boldsymbol{U}(\boldsymbol{x}, T) \\
& \frac{\partial \boldsymbol{U}}{\partial t} + \nabla \cdot \boldsymbol{F}(\boldsymbol{U}, \nabla \boldsymbol{U}) = 0
\end{aligned}
$$

- Isentropic, compressible, Navier-Stokes
- Re = 1000, M = 0.2
- $y(t)$, $\theta(t)$, $c(t)$ parametrized via periodic cubic splines
- Black-box optimizer: SNOPT



Airfoil schematic, kinematic description

Solution of time-periodic, energetically optimal flapping

$$\text{Energy} = 9.4096$$
$$x\text{-impulse} = \text{-}0.1766$$

$$\text{Energy} = 0.45695$$
$$x\text{-impulse} = 0.000$$

Structure: semi-discretization, first-order form

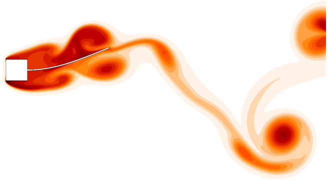
$$\boxed{\boldsymbol{M}^s \frac{\partial \boldsymbol{u}^s}{\partial t} = \boldsymbol{r}^s(\boldsymbol{u}^s; \boldsymbol{t}) = \boldsymbol{r}^{ss}(\boldsymbol{u}^s) + \boldsymbol{r}^{sf} \cdot \boldsymbol{t}}$$

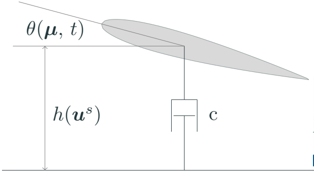
- Semidiscretization (CG-FEM) of **continuum** (hyperelasticity)

$$\begin{aligned} \frac{\partial \boldsymbol{p}}{\partial t} - \nabla \cdot \boldsymbol{P}(\boldsymbol{G}) = \boldsymbol{b} \qquad & \text{in } \Omega_0 \\ \boldsymbol{P}(\boldsymbol{G}) \cdot \boldsymbol{N} = \boldsymbol{t} \qquad & \text{on } \Gamma_N \\ \boldsymbol{x} = \boldsymbol{x}_D \quad & \text{on } \Gamma_D \end{aligned}$$

- Force balance on **rigid body**

$$\boldsymbol{M}\frac{\partial^2 \boldsymbol{q}}{\partial t^2} + \boldsymbol{C}\frac{\partial \boldsymbol{q}}{\partial t} + \boldsymbol{K}\boldsymbol{q} = \boldsymbol{t}$$





Coupled fluid-structure formulation

- Write discretized fluid and structure equations as ODEs

$$\begin{aligned} \boldsymbol{M}^f \dot{\boldsymbol{u}}^f &= \boldsymbol{r}^f(\boldsymbol{u}^f; \boldsymbol{x}) \\ \boldsymbol{M}^s \dot{\boldsymbol{u}}^s &= \boldsymbol{r}^s(\boldsymbol{u}^s; \boldsymbol{t}) \\ &= \boldsymbol{r}^{ss}(\boldsymbol{u}^s) + \boldsymbol{r}^{sf} \cdot \boldsymbol{t} \end{aligned}$$

  in the fluid $\boldsymbol{u}^f$ and structure $\boldsymbol{u}^s$ variables
- Apply couplings
  - Structure-to-fluid: deform fluid domain $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{u}^s)$
  - Fluid-to-structure: apply boundary traction $\boldsymbol{t} = \boldsymbol{t}(\boldsymbol{u}^f)$
- Write coupled system as $\boldsymbol{M}\dot{\boldsymbol{u}} = \boldsymbol{r}(\boldsymbol{u})$

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}^f \\ \boldsymbol{u}^s \end{bmatrix} \qquad \boldsymbol{r}(\boldsymbol{u}) = \begin{bmatrix} \boldsymbol{r}^f(\boldsymbol{u}^f; \boldsymbol{x}(\boldsymbol{u}^s)) \\ \boldsymbol{r}^s(\boldsymbol{u}^s; \boldsymbol{t}(\boldsymbol{u}^f)) \end{bmatrix} \qquad \boldsymbol{M} = \begin{bmatrix} \boldsymbol{M}^f & \\ & \boldsymbol{M}^s \end{bmatrix}$$

- Structure of linearized residual

$$\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}) = \begin{bmatrix} \dfrac{\partial \boldsymbol{r}^f}{\partial \boldsymbol{u}^f} & \dfrac{\partial \boldsymbol{r}^f}{\partial \boldsymbol{x}} \dfrac{\partial \boldsymbol{x}}{\partial \boldsymbol{u}^s} \\ \dfrac{\partial \boldsymbol{r}^{sf}}{\partial \boldsymbol{t}} \dfrac{\partial \boldsymbol{t}}{\partial \boldsymbol{u}^f} & \dfrac{\partial \boldsymbol{r}^s}{\partial \boldsymbol{u}^s} \end{bmatrix}$$

High-order partitioned FSI solver: IMEX Runge-Kutta[1]

- Exploit **linear dependence** of structure residual ($\boldsymbol{r}^s$) on traction ($\boldsymbol{t}$)

$$\boldsymbol{r}(\boldsymbol{u}) = \begin{bmatrix} \boldsymbol{r}^f(\boldsymbol{u}^f;\, \boldsymbol{x}(\boldsymbol{u}^s)) \\ \boldsymbol{r}^s(\boldsymbol{u}^s;\, \boldsymbol{t}(\boldsymbol{u}^f)) \end{bmatrix} = \underbrace{\begin{bmatrix} \\ \boldsymbol{r}^{sf} \cdot (\boldsymbol{t}(\boldsymbol{u}^f) - \tilde{\boldsymbol{t}}) \end{bmatrix}}_{\boldsymbol{f}(\boldsymbol{u})} + \underbrace{\begin{bmatrix} \boldsymbol{r}^f(\boldsymbol{u}^f;\, \boldsymbol{x}(\boldsymbol{u}^s)) \\ \boldsymbol{r}^s(\boldsymbol{u}^s;\, \tilde{\boldsymbol{t}}) \end{bmatrix}}_{\boldsymbol{g}(\boldsymbol{u})}$$

- Apply **high-order** implicit-explicit Runge-Kutta scheme to discretize

$$\boldsymbol{M}\frac{\partial \boldsymbol{u}}{\partial t} = \boldsymbol{r}(\boldsymbol{u}) = \underbrace{\boldsymbol{f}(\boldsymbol{u})}_{\text{explicit}} + \underbrace{\boldsymbol{g}(\boldsymbol{u})}_{\text{implicit}}$$

  - Explicit Runge-Kutta scheme $\hat{c}$, $\hat{A}$, $\hat{b}$ for $\boldsymbol{f}(\boldsymbol{u})$
  - Diagonally implicit scheme c, A, b for $\boldsymbol{g}(\boldsymbol{u})$

$$\boldsymbol{u}_n = \boldsymbol{u}_{n-1} + \sum_{i=1}^{s} \hat{b}_i \hat{\boldsymbol{k}}_{n,i} + \sum_{i=1}^{s} b_i \boldsymbol{k}_{n,i}$$

$$\boldsymbol{M}\boldsymbol{k}_{n,i} = \Delta t_n \boldsymbol{g}\left(\boldsymbol{u}_{n-1} + \sum_{j=1}^{i-1} \hat{a}_{ij} \hat{\boldsymbol{k}}_{n,j} + \sum_{j=1}^{i} a_{ij} \boldsymbol{k}_{n,j}\right)$$

$$\boldsymbol{M}\hat{\boldsymbol{k}}_{n,i} = \Delta t_n \boldsymbol{f}\left(\boldsymbol{u}_{n-1} \sum_{j=1}^{i-1} \hat{a}_{ij} \hat{\boldsymbol{k}}_{n,j} + \sum_{j=1}^{i} a_{ij} \boldsymbol{k}_{n,j}\right)$$

[1] [van Zuijlen and Bijl, 2005, Froehle and Persson, 2014]

High-order partitioned FSI solver: IMEX Runge-Kutta[1]

- Exploit **linear dependence** of structure residual ($\boldsymbol{r}^s$) on traction ($\boldsymbol{t}$)

$$\boldsymbol{r}(\boldsymbol{u}) = \begin{bmatrix} \boldsymbol{r}^f(\boldsymbol{u}^f;\, \boldsymbol{x}(\boldsymbol{u}^s)) \\ \boldsymbol{r}^s(\boldsymbol{u}^s;\, \boldsymbol{t}(\boldsymbol{u}^f)) \end{bmatrix} = \underbrace{\begin{bmatrix} \\ \boldsymbol{r}^{sf} \cdot (\boldsymbol{t}(\boldsymbol{u}^f) - \tilde{\boldsymbol{t}}) \end{bmatrix}}_{\boldsymbol{f}(\boldsymbol{u})} + \underbrace{\begin{bmatrix} \boldsymbol{r}^f(\boldsymbol{u}^f;\, \boldsymbol{x}(\boldsymbol{u}^s)) \\ \boldsymbol{r}^s(\boldsymbol{u}^s;\, \tilde{\boldsymbol{t}}) \end{bmatrix}}_{\boldsymbol{g}(\boldsymbol{u})}$$

- Structure of linearized implicit residual

$$\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{u}}(\boldsymbol{u}) = \begin{bmatrix} \dfrac{\partial \boldsymbol{r}^f}{\partial \boldsymbol{u}^f}(\boldsymbol{u}^f;\, \boldsymbol{x}(\boldsymbol{u}^s)) & \dfrac{\partial \boldsymbol{r}^f}{\partial \boldsymbol{x}}(\boldsymbol{u}^f;\, \boldsymbol{x}(\boldsymbol{u}^s))\dfrac{\partial \boldsymbol{x}}{\partial \boldsymbol{u}^s}(\boldsymbol{u}^s) \\ & \dfrac{\partial \boldsymbol{r}^s}{\partial \boldsymbol{u}^s}(\boldsymbol{u}^s;\, \tilde{\boldsymbol{t}}) \end{bmatrix}$$

- Solve: (1) implicit **structure**, (2) implicit **fluid**, (3) explicit **structure**
  - Sequence of solves can be performed at **linearized** or **nonlinear** level
- Due to choice of IMEX partition: **no explicit fluid stages**

[1] [van Zuijlen and Bijl, 2005, Froehle and Persson, 2014]

High-order partitioned FSI solver: IMEX Runge-Kutta

- Define **stage solutions**

$$\boldsymbol{u}^s_{n,i} = \boldsymbol{u}^s_{n-1} + \sum_{j=1}^{i} a_{ij}\boldsymbol{k}^s_{n,j} + \sum_{j=1}^{i-1} \hat{a}_{ij}\hat{\boldsymbol{k}}^s_{n,j}$$

$$\boldsymbol{u}^f_{n,i} = \boldsymbol{u}^f_{n-1} + \sum_{j=1}^{i} a_{ij}\boldsymbol{k}^f_{n,j}$$

- Define **traction predictor** as true traction at previous stage

$$\tilde{\boldsymbol{t}}_{n,i} = \boldsymbol{t}(\boldsymbol{u}_{n,i-1})$$

- Solve for **stage velocities** $(i = 1, \ldots, s)$

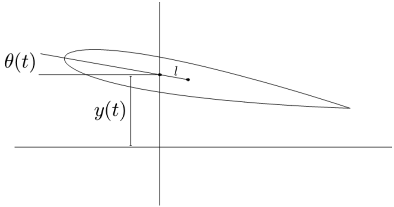
$$\begin{aligned} \boldsymbol{M}^s\boldsymbol{k}^s_{n,i} &= \Delta t_n \boldsymbol{r}^s(\boldsymbol{u}^s_{n,i};\, \tilde{\boldsymbol{t}}_{n,i}) \\ \boldsymbol{M}^f\boldsymbol{k}^f_{n,i} &= \Delta t_n \boldsymbol{r}^f(\boldsymbol{u}^f_{n,i};\, \boldsymbol{x}(\boldsymbol{u}^s_{n,i})) \\ \boldsymbol{M}^s\hat{\boldsymbol{k}}^s_{n,i} &= \Delta t_n \boldsymbol{r}^{sf}(\boldsymbol{t}(\boldsymbol{u}^f_{n,i}) - \tilde{\boldsymbol{t}}_{n,i}) \end{aligned}$$

- Update state solution at new time

$$\boldsymbol{u}^f_n = \boldsymbol{u}^f_{n-1} + \sum_{j=1}^{s} b_j\boldsymbol{k}^f_{n,j}, \qquad \boldsymbol{u}^s_n = \boldsymbol{u}^s_{n-1} + \sum_{j=1}^{s} b_j\boldsymbol{k}^s_{n,j} + \sum_{j=1}^{s} \hat{b}_j\hat{\boldsymbol{k}}^s_{n,j}$$

Validation: benchmark pitching airfoil system

- Simple FSI benchmark problem for studying the high-order accuracy of the IMEX scheme
- Rigid pitching/heaving NACA 0012 airfoil, torsional spring
- Smooth heaving step $y(t)$ prescribed, angle $\theta(t)$ measured

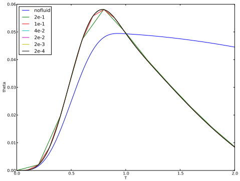


Setup

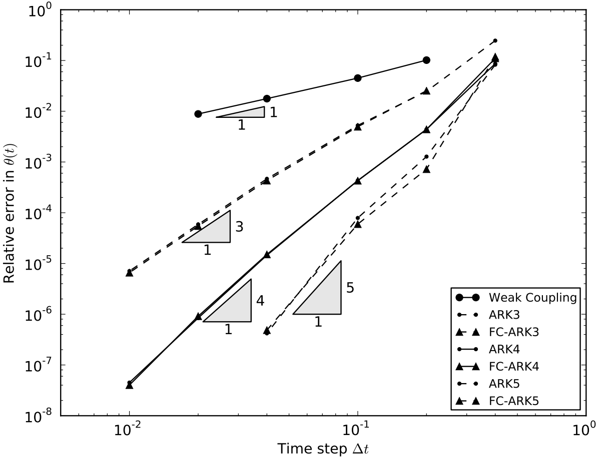
Mach number

Validation: benchmark pitching airfoil system

- Up to 5th order of convergence in time.
- Similar accuracy as solving fully coupled system



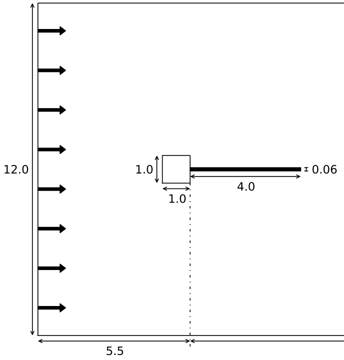
Angle $\theta(t)$ vs time t



Entropy

Validation: cantilever system

- Standard FSI benchmark problem.
- Elastic cantilever behind a square bluff body in incompressible flow.

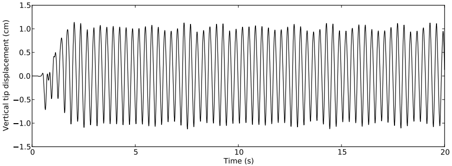


- Cantilever:
  $\rho_s = 100\,\mathrm{kg/m^3}$, $\nu_s = 0.35$,
  $E = 2.5 \times 10^5\,\mathrm{Pa}$.
- Fluid & Flow:
  $\rho_f = 1.18\,\mathrm{kg/m^3}$,
  $\nu_f = 1.54 \times 10^{-5}\,\mathrm{m^2/s}$,
  $v_f = 0.513\,\mathrm{m/s}$, $\mathrm{Re} = 333$,
  $\mathrm{Ma} = 0.2$.

- Vortex shedding frequency: $\sim 6.3\,\mathrm{Hz}$
  Cantilever first mode: $3.03\,\mathrm{Hz}$

Validation: cantilever system

Entropy



Tip displacement

- Tip frequency: $f = 3.14\,\mathrm{Hz}$ (Literature: $2.98 - 3.25\,\mathrm{Hz}$)
- Tip displacement: $d_{max} = 1.09\,\mathrm{cm}$ (Literature: $0.95 - 1.25\,\mathrm{cm}$)

Flow around Membrane, 3-D

- Angle of attack 22.6°, Reynolds number 2000.
- Flexible structure prevents leading edge separation.

Adjoint equations for high-order partitioned IMEX FSI solver

- **Define**

$$\boldsymbol{r}^f_{n,i} = \boldsymbol{r}^f(\boldsymbol{u}^f_{n,i};\, \boldsymbol{x}(\boldsymbol{u}^s_{n,i})) \qquad \boldsymbol{r}^s_{n,i} = \boldsymbol{r}^s(\boldsymbol{u}^s_{n,i};\, \tilde{\boldsymbol{t}}_{n,i})$$

- **Final condition** for state Lagrange multipliers (F is quantity of interest)

$$\boldsymbol{\lambda}^f_{N_t} = \frac{\partial F}{\partial \boldsymbol{u}^f_{N_t}}^T, \quad \boldsymbol{\lambda}^s_{N_t} = \frac{\partial F}{\partial \boldsymbol{u}^s_{N_t}}^T$$

- Solve for **stage** Lagrange multipliers ($j = s, \ldots, 1$)
  - **Explicit structure** stage

$$\boldsymbol{M}^{s\,T}\hat{\boldsymbol{\kappa}}^s_{n,j} = \frac{\partial F}{\partial \hat{\boldsymbol{k}}^s_{n,j}}^T + \hat{b}_j \boldsymbol{\lambda}^s_n + \Delta t_n \sum_{i=j+1}^{s} \hat{a}_{ij} \frac{\partial \boldsymbol{r}^f_{n,i}}{\partial \boldsymbol{u}^s}^T \boldsymbol{\kappa}^f_{n,i} + \Delta t_n \sum_{i=j+1}^{s} \hat{a}_{ij} \frac{\partial \boldsymbol{r}^s_{n,i}}{\partial \boldsymbol{u}^s}^T \boldsymbol{\kappa}^s_{n,i}$$

  - **Implicit fluid** stage

$$\boldsymbol{M}^{f\,T}\boldsymbol{\kappa}^f_{n,j} = \frac{\partial F}{\partial \boldsymbol{k}^f_{n,j}}^T + b_j \boldsymbol{\lambda}^f_n + \Delta t_n \sum_{i=j}^{s} a_{ij} \frac{\partial \boldsymbol{r}^f_{n,i}}{\partial \boldsymbol{u}^f}^T \boldsymbol{\kappa}^f_{n,i} + \Delta t_n \sum_{i=j+1}^{s} a_{ij} \frac{\partial \tilde{\boldsymbol{t}}_{n,i}}{\partial \boldsymbol{u}^f}^T \boldsymbol{r}^{sf\,T} \boldsymbol{\kappa}^s_{n,i}$$
$$- \Delta t_n \sum_{i=j}^{s} a_{ij} \frac{\partial \boldsymbol{t}_{n,i}}{\partial \boldsymbol{u}^f}^T \boldsymbol{r}^{sf\,T} \hat{\boldsymbol{\kappa}}^s_{n,i} + \Delta t_n \sum_{i=j+1}^{s} a_{ij} \frac{\partial \tilde{\boldsymbol{t}}_{n,i}}{\partial \boldsymbol{u}^f}^T \boldsymbol{r}^{sf\,T} \hat{\boldsymbol{\kappa}}^s_{n,i}$$

  - **Implicit structure** stage

$$\boldsymbol{M}^{s\,T}\boldsymbol{\kappa}^s_{n,j} = \frac{\partial F}{\partial \boldsymbol{k}^s_{n,j}}^T + b_j \boldsymbol{\lambda}^s_n + \Delta t_n \sum_{i=j}^{s} a_{ij} \frac{\partial \boldsymbol{r}^f_{n,i}}{\partial \boldsymbol{u}^s}^T \boldsymbol{\kappa}^f_{n,i} + \Delta t_n \sum_{i=j}^{s} a_{ij} \frac{\partial \boldsymbol{r}^s_{n,i}}{\partial \boldsymbol{u}^s}^T \boldsymbol{\kappa}^s_{n,i}$$

Adjoint equations for high-order partitioned IMEX FSI solver

- Update state Lagrange multipliers at new time

$$\boldsymbol{\lambda}_{n-1}^{f} = \boldsymbol{\lambda}_{n}^{f} + \frac{\partial F}{\partial \boldsymbol{u}_{n-1}^{f}}^{T} + \Delta t_n \sum_{i=1}^{s} \frac{\partial \boldsymbol{r}_{n,i}^{f}}{\partial \boldsymbol{u}^{f}}^{T} \boldsymbol{\kappa}_{n,i}^{f} + \Delta t_n \sum_{i=1}^{s} \frac{\partial \tilde{\boldsymbol{t}}_{n,i}}{\partial \boldsymbol{u}^{f}}^{T} \boldsymbol{r}_{n,i}^{sf\,T} \boldsymbol{\kappa}_{n,i}^{s} + \Delta t_n \sum_{i=1}^{s} \left[\frac{\partial \tilde{\boldsymbol{t}}_{n,i}}{\partial \boldsymbol{u}^{f}} - \frac{\partial \boldsymbol{t}_{n,i}}{\partial \boldsymbol{u}^{f}} \right]^{T} \boldsymbol{r}_{n,i}^{sf\,T} \hat{\boldsymbol{\kappa}}_{n,i}^{s}$$

$$\boldsymbol{\lambda}_{n-1}^{s} = \boldsymbol{\lambda}_{n}^{s} + \frac{\partial F}{\partial \boldsymbol{u}_{n-1}^{s}}^{T} + \Delta t_n \sum_{i=1}^{s} \frac{\partial \boldsymbol{r}_{n,i}^{f}}{\partial \boldsymbol{u}^{s}}^{T} \boldsymbol{\kappa}_{n,i}^{f} + \Delta t_n \sum_{i=1}^{s} \frac{\partial \boldsymbol{r}_{n,i}^{s}}{\partial \boldsymbol{u}^{s}}^{T} \boldsymbol{\kappa}_{n,i}^{s}$$

- Reconstruct **total derivative** of quantity of interest F as

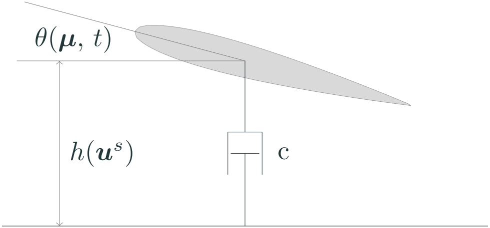
$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{\mu}} = \frac{\partial F}{\partial \boldsymbol{\mu}} + \boldsymbol{\lambda}_0^{f\,T} \frac{\partial \bar{\boldsymbol{u}}^{f}}{\partial \boldsymbol{\mu}} + \boldsymbol{\lambda}_0^{s\,T} \frac{\partial \bar{\boldsymbol{u}}^{s}}{\partial \boldsymbol{\mu}} - \sum_{n=0}^{N_t} \Delta t_n \sum_{i=1}^{s} \boldsymbol{\kappa}_{n,i}^{f}{}^{T} \frac{\partial \boldsymbol{r}_{n,i}^{f}}{\partial \boldsymbol{\mu}} - \sum_{n=0}^{N_t} \Delta t_n \sum_{i=1}^{s} \boldsymbol{\kappa}_{n,i}^{s}{}^{T} \frac{\partial \boldsymbol{r}_{n,i}^{s}}{\partial \boldsymbol{\mu}} - \sum_{n=0}^{N_t} \Delta t_n \sum_{i=1}^{s} \hat{\boldsymbol{\kappa}}_{n,i}^{s,T} \frac{\partial \boldsymbol{r}_{n,i}^{sf}}{\partial \boldsymbol{\mu}}$$

Optimal energy harvesting from foil-damper system

Goal: Maximize energy harvested from foil-damper system

$$\underset{\boldsymbol{\mu}}{\text{maximize}} \quad \frac{1}{T}\int_0^T (c\dot{h}^2(\boldsymbol{u}^s) - M_z(\boldsymbol{u}^f)\dot{\theta}(\boldsymbol{\mu}, t))\, dt$$

- Fluid: Isentropic Navier-Stokes on deforming domain (ALE)
- Structure: Force balance in y-direction between foil and damper
- Motion driven by *imposed* $\theta(\boldsymbol{\mu}, t) = \mu_1 \cos(2\pi f t)$; $\mu_1 \in (-45°, 45°)$



$\mu_1^* = 45°$

References I

Alexander, R. (1977).
Diagonally implicit Runge-Kutta methods for stiff o.d.e.'s.
SIAM J. Numer. Anal., 14(6):1006–1021.

Froehle, B. and Persson, P.-O. (2014).
A high-order discontinuous galerkin method for fluid–structure interaction with efficient implicit–explicit time stepping.
Journal of Computational Physics, 272:455–470.

Peraire, J. and Persson, P.-O. (2008).
The Compact Discontinuous Galerkin (CDG) method for elliptic problems.
SIAM Journal on Scientific Computing, 30(4):1806–1824.

Persson, P.-O., Bonet, J., and Peraire, J. (2009).
Discontinuous galerkin solution of the navier–stokes equations on deformable domains.
Computer Methods in Applied Mechanics and Engineering, 198(17):1585–1595.

van Zuijlen, A. H. and Bijl, H. (2005).
Implicit and explicit higher order time integration schemes for structural dynamics and fluid-structure interaction computations.
Computers & structures, 83(2):93–105.

Zahr, M. J. and Persson, P.-O. (2016).
An adjoint method for a high-order discretization of deforming domain conservation laws for optimization of flow problems.
Journal of Computational Physics.

Zahr, M. J., Persson, P.-O., and Wilkening, J. (2016).
A fully discrete adjoint method for optimization of flow problems on deforming domains with time-periodicity constraints.
Computers & Fluids.