A High-Order Implicit-Explicit Discontinuous Galerkin Scheme for Fluid-Structure Interaction

Per-Olof Persson and Bradley Froehle

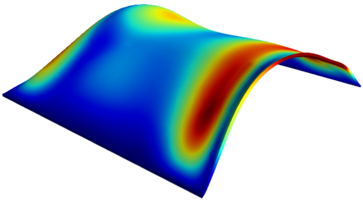
Department of Mathematics, University of California, Berkeley
Mathematics Department, Lawrence Berkeley National Laboratory

SIAM Conference on Computational Science and Engineering
Boston, Massachusetts

February 25, 2013

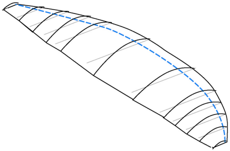
Motivation

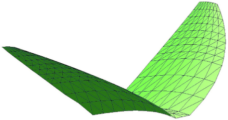
- Many important problems require predictions of fluid-structure interaction (FSI):
  - Oscillatory interactions in engineering systems (e.g. aircraft, turbines, and bridges) can lead to failure
  - The blood flow in arteries and artificial heart valves is highly dependent on structural interactions
- Requirements on numerical solvers:
  - High-order accuracy, to capture non-linear interactions and multiscale phenomena
  - Unstructured meshes, for complex geometries and adaptivity

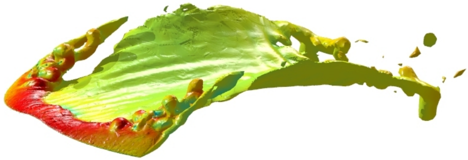


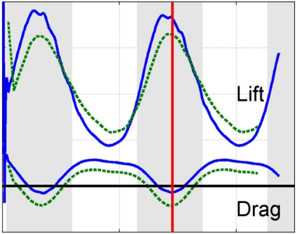
Application: Optimal Design of Flapping Wings

- Automatic generation of optimized flapping wing kinematics [Persson/Willis ’11]
- Camber crucial to avoid excessive flow separation – can be imposed using compliant wings and fluid-structure interaction



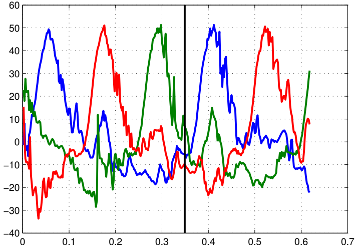


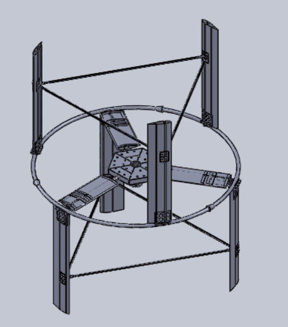


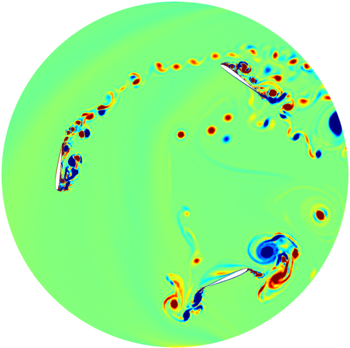


Application: Vertical Axis Wind Turbines

- Recent interest in vertical axis wind turbines (VAWT) due to several attractive properties
- Modeling of structural interactions important for study of sensitivities to design conditions and fatigue

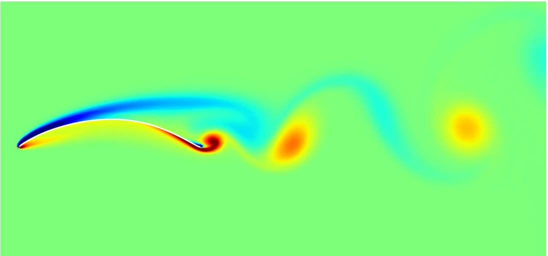






Full vs. weak coupling

- Two main numerical approaches for the coupling:
  - Fully coupled (monolithic): Solve the fluid/structure equations simultaneously. Accurate, but requires specialized codes and solvers are often slow.
  - Weakly coupled (partitioned): Use standard solvers for fluid/structure and apply a separate coupling scheme, often together with subiterations. Efficient and simple, but issues with accuracy and stability.

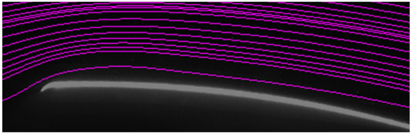


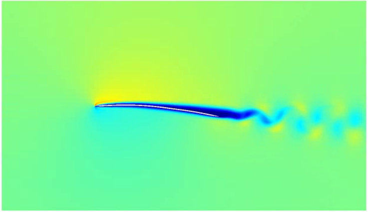
Explicit time integration

- In [Persson, Peraire, Bonet, 2007], we developed a fully coupled FSI solver using DG, membrane models, and explicit time stepping
- Implicit solvers required for more challenging problems in 3-D

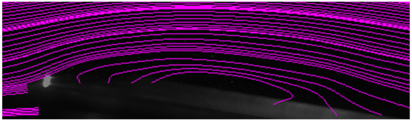
Experiment (A. Song, Brown U)

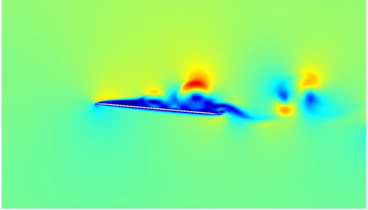
Fluid/membrane simulation





Compliant membrane

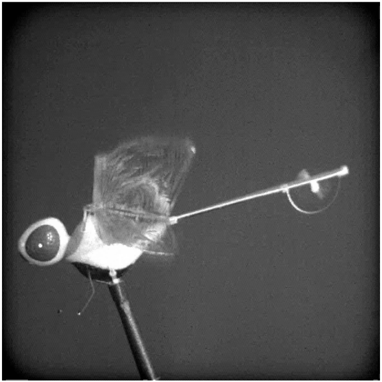




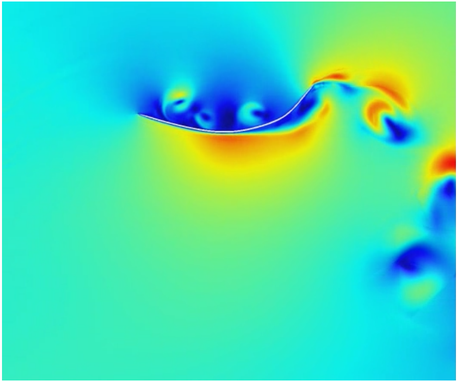
Rigid flat plate

Example: Dragonfly, Compliant Wings

Experiment (A. Song, Brown U)



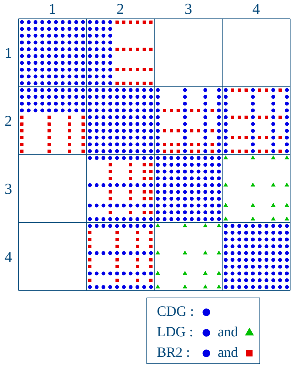
Fluid/membrane simulation



Discontinuous Galerkin Discretization for Fluids

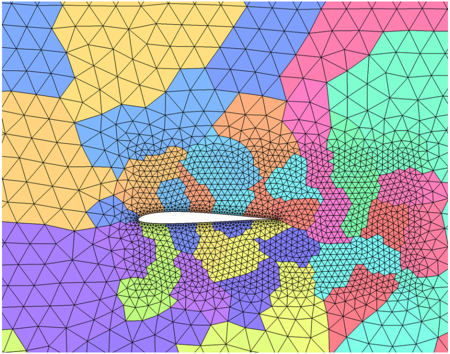
- High-order nodal-DG method for unstructured simplex meshes
- Compressible Navier-Stokes equations, Roe's numerical fluxes
- CDG fluxes for second-order terms [Peraire/Persson 2008],

  $\Longrightarrow$ High level of sparsity in Jacobian matrices
- Implicit time integration by matrix-based Newton-Krylov solvers
  - L-stable Diagonally Implicit Runge-Kutta (DIRK) methods
  - Block-ILU(0) preconditioners and automatic element ordering [Persson/Peraire '08]
  - Implicit-Explicit Runge-Kutta schemes for LES-type problems [Persson '11]



Parallel Solvers

- Implicit solvers typically required because of CFL restrictions from viscous effects, low Mach numbers, and adaptive/anisotropic grids
- Jacobian matrices are large even at $p = 2$ or $p = 3$, however:
  - They are required for non-trivial preconditioners
  - They are very expensive to recompute
- Distributed parallel solvers developed in [Persson '09]



- Parallelization to 1000's of processes by domain decomposition
- Close to perfect speedup for time accurate simulations

Lagrangian FEM Discretization of Structures

- Map from reference domain V to physical domain $v(t)$

$$\boldsymbol{F} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}}\,, \qquad J = \det \boldsymbol{F}\,, \qquad \boldsymbol{v}(\boldsymbol{X},t) = \frac{\partial \boldsymbol{x}}{\partial t}\,, \qquad \boldsymbol{p} = \rho_0 \boldsymbol{v}$$

- Conservation of linear momentum:

$$\frac{\partial \boldsymbol{p}}{\partial t} = \boldsymbol{\nabla} \cdot \boldsymbol{P} + \rho_0 \boldsymbol{b}$$

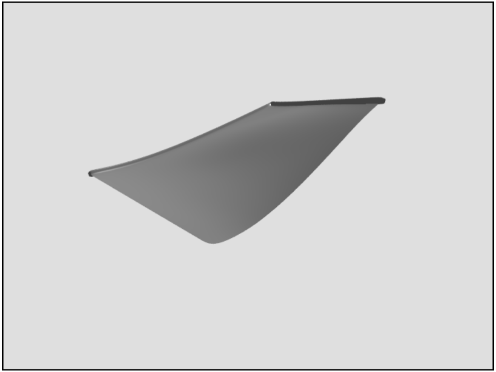
  with *first Piola-Kirchhoff stress tensor* $\boldsymbol{P}(\boldsymbol{F})$
- Hyperelastic Neo-Hookean Constitutive Model
- Straight-forward second-order formulation in terms of material points $\boldsymbol{x}$ and momentum $\boldsymbol{p}$:

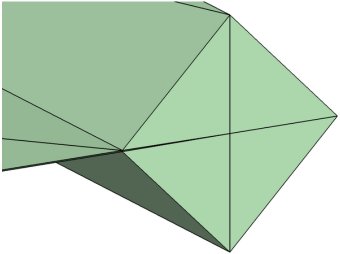
$$\frac{\partial \boldsymbol{x}}{\partial t} = \boldsymbol{p}/\rho_0, \quad \frac{\partial \boldsymbol{p}}{\partial t} - \nabla \cdot \boldsymbol{P}(\boldsymbol{F}) = \rho_0 \boldsymbol{b}$$

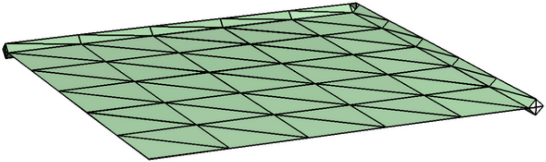
- Discretize by standard high-order continuous Galerkin FEM method, temporal integration by high-order DIRK schemes

Nonlinear elasticity solvers for thin structures

- Volumetric modeling of thin structures $\Longrightarrow$ stiff nonlinear systems
- However, direct solvers scale well due to 2-D nature of the mesh
- Parallel MPI solvers using the MUMPS package







Coupled Fluid-Structure Formulation

- Lagrangian CG-FEM formulation for the solid dynamics

$$\frac{\partial \boldsymbol{u}^s}{\partial t} + \nabla \cdot \boldsymbol{F}^s(\boldsymbol{u}^s; \boldsymbol{\ell}^{fs}) = 0$$

  written as a system of first-order ODEs

- Structure motion and an (algebraic) mesh deformation scheme induce a deformation of the fluid domain, $\boldsymbol{x}^f = \boldsymbol{x}^f(\boldsymbol{u}^s)$
- Fluid flow governed by the compressible Navier-Stokes equations:

$$\frac{\partial \boldsymbol{u}^f}{\partial t} + \nabla \cdot \boldsymbol{F}^f(\boldsymbol{u}^f; \boldsymbol{x}^f) = 0$$

  with mapping-based ALE formulation for the deforming domain

- Fluid induces forces on the structure, $\boldsymbol{\ell}^{fs} = \boldsymbol{\ell}^{fs}(\boldsymbol{u}^f, \boldsymbol{x}^f)$

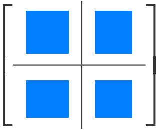
Coupled system structure

- Eliminate the mesh deformation $\boldsymbol{x}^f$ and include interface forces explicitly in the structure residual, to obtain a system of ODEs $M\dot{\boldsymbol{u}} = \boldsymbol{r}(\boldsymbol{u})$ where

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}^f \\ \boldsymbol{u}^s \end{bmatrix}, \quad \boldsymbol{r} = \begin{bmatrix} \boldsymbol{r}^f(\boldsymbol{u}^f, \boldsymbol{u}^s) \\ \boldsymbol{r}^s(\boldsymbol{u}^s) + \boldsymbol{r}^{fs}(\boldsymbol{\ell}^{fs}(\boldsymbol{u}^f, \boldsymbol{u}^s)) \end{bmatrix}, \quad M = \begin{bmatrix} M^f & \\ & M^s \end{bmatrix}$$

- A fully coupled implicit solver requires solution of systems of the form $(M - \alpha \Delta t K)\boldsymbol{u} = \boldsymbol{f}$, with Jacobian matrix structure

$$K = \frac{d\boldsymbol{r}}{d\boldsymbol{u}} =$$



- Using IMEX schemes, we will treat the terms involving $\boldsymbol{\ell}^{fs}(\boldsymbol{u}^f, \boldsymbol{u}^s)$ explicitly, which makes the Jacobian matrix block upper-triangular

Implicit-Explicit Runge-Kutta Methods

- Based on a splitting $\frac{du}{dt} = f(u) + g(u)$ where $f(u)$ is considered nonstiff terms and $g(u)$ stiff terms
- Two Runge-Kutta schemes
  1. Diagonally Implicit Runge-Kutta (DIRK) scheme c, A, b for $g(u)$
  2. Explicit Runge-Kutta (ERK) scheme $\hat{c}, \hat{A}, \hat{b}$ for $f(u)$

$\hat{k}_1 = f(u_n)$

for $i = 1$ **to** s

  Solve for k_i in $k_i = g(u_{n,i})$, where $u_{n,i} = u_n + \Delta t \sum_{j=1}^{i} a_{i,j} k_j + \Delta t \sum_{j=1}^{i} \hat{a}_{i+1,j} \hat{k}_j$

  Evaluate $\hat{k}_{i+1} = f(u_{n,i})$

end for

$$u_{n+1} = u_n + \Delta t \sum_{i=1}^{s} b_j k_j + \Delta t \sum_{i=1}^{s+1} \hat{b}_j \hat{k}_j$$

IMEX Schemes

IMEX1: 2-stage, 2nd order DIRK + 3-stage, 2nd order ERK

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array} = \begin{array}{c|cc} \alpha & \alpha & 0 \\ 1 & 1-\alpha & \alpha \\ \hline & 1-\alpha & \alpha \end{array} \qquad \begin{array}{c|c} \hat{c} & \hat{A} \\ \hline & \hat{b}^T \end{array} = \begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \alpha & \alpha & 0 & 0 \\ 1 & \delta & 1-\delta & 0 \\ \hline & 0 & 1-\alpha & \alpha \end{array}$$

where $\alpha = 1 - \frac{\sqrt{2}}{2}$, $\delta = -2\sqrt{2}/3$. 2nd order, L-stable.

IMEX2: 2-stage, 3rd order DIRK + 3-stage, 3rd order ERK

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array} = \begin{array}{c|cc} \alpha & \alpha & 0 \\ 1-\alpha & 1-2\alpha & \alpha \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array} \qquad \begin{array}{c|c} \hat{c} & \hat{A} \\ \hline & \hat{b}^T \end{array} = \begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \alpha & \alpha & 0 & 0 \\ 1-\alpha & \alpha-1 & 2(1-\alpha) & 0 \\ \hline & 0 & \frac{1}{2} & \frac{1}{2} \end{array}$$

where $\alpha = (3 + \sqrt{3})/6$. 3rd order accurate, no L-stability.

IMEX Schemes

IMEX3: 3-stage, 3rd order DIRK + 4-stage, 3rd order ERK

$$
\begin{array}{c|c} c & A \\ \hline & b^T \end{array} = \begin{array}{c|ccc}
0.43586652 & 0.43586652 & 0 & 0 \\
0.71793326 & 0.28206673 & 0.43586652 & 0 \\
1 & 1.2084966 & -0.64436317 & 0.43586652 \\
\hline
 & 1.2084966 & -0.64436317 & 0.43586652
\end{array}
$$

$$
\begin{array}{c|c} \hat{c} & \hat{A} \\ \hline & \hat{b}^T \end{array} = \begin{array}{c|cccc}
0 & 0 & 0 & 0 & 0 \\
0.43586652 & 0.43586652 & 0 & 0 & 0 \\
0.71793326 & 0.32127888 & 0.39665437 & 0 & 0 \\
1 & -0.10585829 & 0.55292914 & 0.55292914 & 0 \\
\hline
 & 0 & 1.20849664 & -0.64436317 & 0.43586652
\end{array}
$$

3rd order accurate, L-stable.

Partitioned FSI using IMEX schemes

- The IMEX schemes can be used to derive accurate partitioning methods for fully coupled FSI problems [van Zuijlen, 2006]
- For our FSI system, we treat the interface forces $\boldsymbol{\ell}^{fs}(\boldsymbol{u}^f, \boldsymbol{u}^s)$ explicitly and everything else implicitly:

$$\boldsymbol{r} = \begin{bmatrix} \boldsymbol{r}^f(\boldsymbol{u}^f, \boldsymbol{u}^s) \\ \boldsymbol{r}^s(\boldsymbol{u}^s, \boldsymbol{\ell}^{fs}) \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{r}^{fs}(\boldsymbol{\ell}^{fs}(\boldsymbol{u}^f, \boldsymbol{u}^s)) \end{bmatrix} + \begin{bmatrix} \boldsymbol{r}^f(\boldsymbol{u}^f, \boldsymbol{u}^s) \\ \boldsymbol{r}^s(\boldsymbol{u}^s) \end{bmatrix} = \boldsymbol{f}(\boldsymbol{u}) + \boldsymbol{g}(\boldsymbol{u})$$

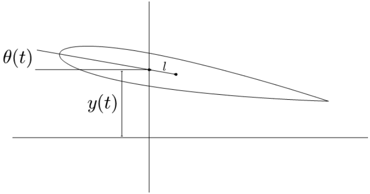
- The interface forces can then be solved for algebraically:

$$\hat{\boldsymbol{\ell}}_{n,i} = \sum_{j=1}^{i-1} \frac{\hat{a}_{ij} - a_{ij}}{a_{ii}} \boldsymbol{\ell}_{n,j}$$

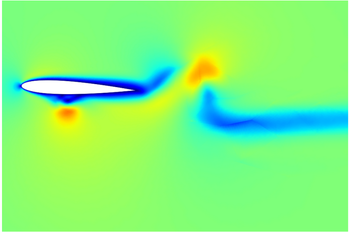
- The remaining structure and fluid components can be solved by back-solution of the block upper-triangular system
- Use new fluid/structure stage solutions $\boldsymbol{u}^f_{n,i}$, $\boldsymbol{u}^s_{n,i}$ to update the interface forces $\hat{\boldsymbol{\ell}}_{n,i} \to \boldsymbol{\ell}_{n,i}$
- Consistent forces, no subiterations required

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



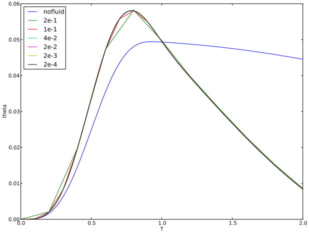
Setup



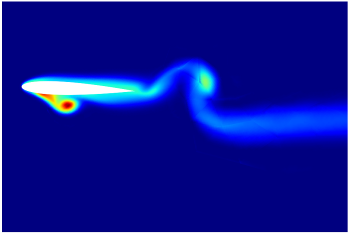
Mach number

Validation, Benchmark Pitching Airfoil System

- High-order DG for Navier-Stokes, ALE for moving domain
- Study convergence of $\theta(t)$ as $\Delta t \to 0$



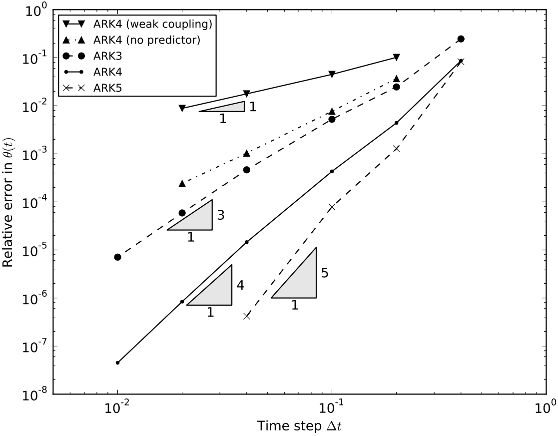
Angle $\theta(t)$ vs time t



Entropy

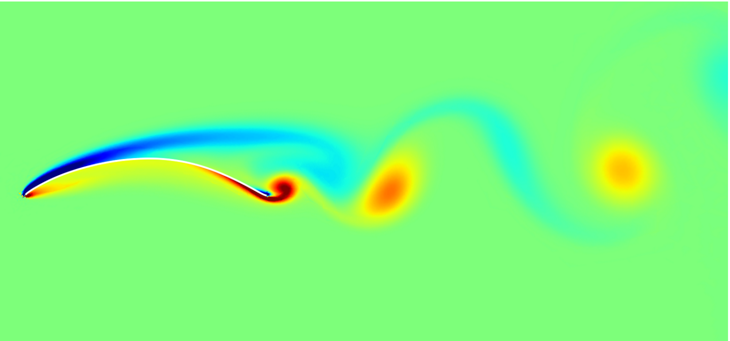
Validation, Benchmark Pitching Airfoil System

- Up to 5th order of convergence in time
- Without the predictor, at most 2nd order convergence



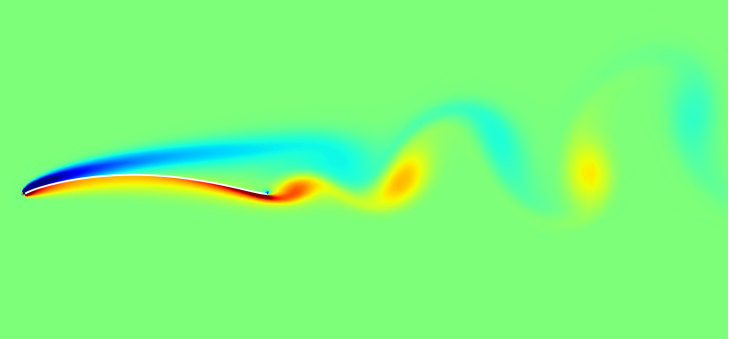
Flow around membrane, 2-D

- Volumetric modeling of Lagrangian Neo-Hookean membrane
- Membrane ends are held fixed but allowed to rotate
- Angle of attack 20°, Reynolds number $2,000$
- Implicit schemes handle complex behavior with large time-steps
- Low membrane stiffness



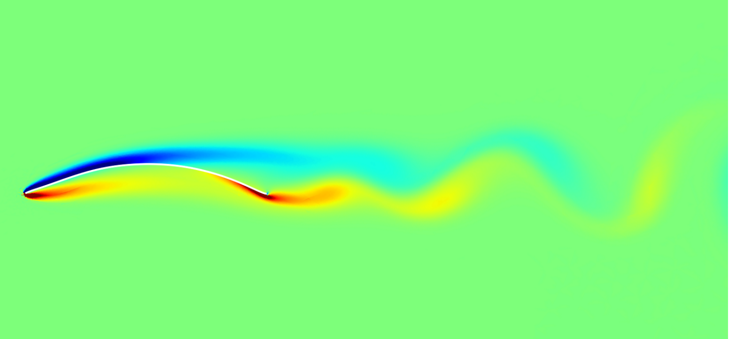
Flow around membrane, 2-D

- Higher membrane stiffness



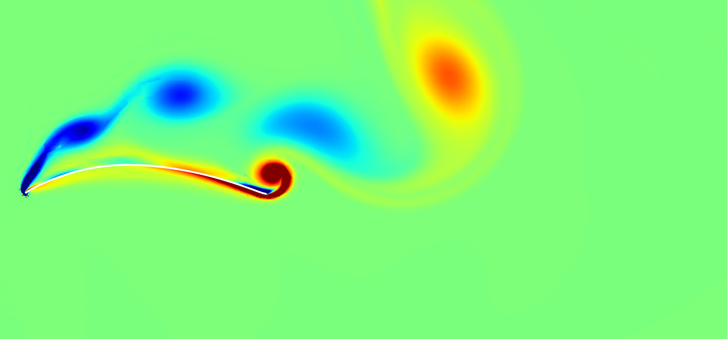
Flow around membrane, 2-D

- Lower angle of attack



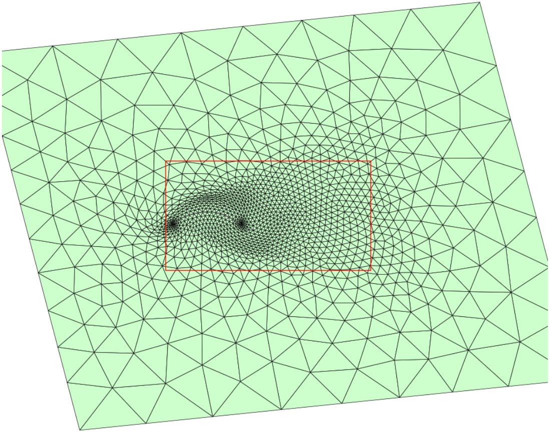
Flow around membrane, 2-D

- Higher angle of attack



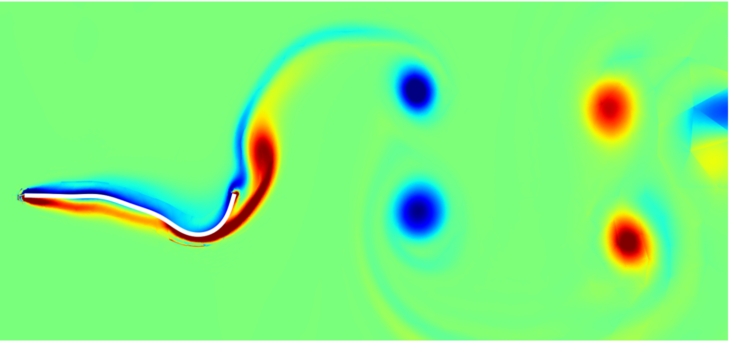
Flow around membrane, 2-D

- Mesh motion



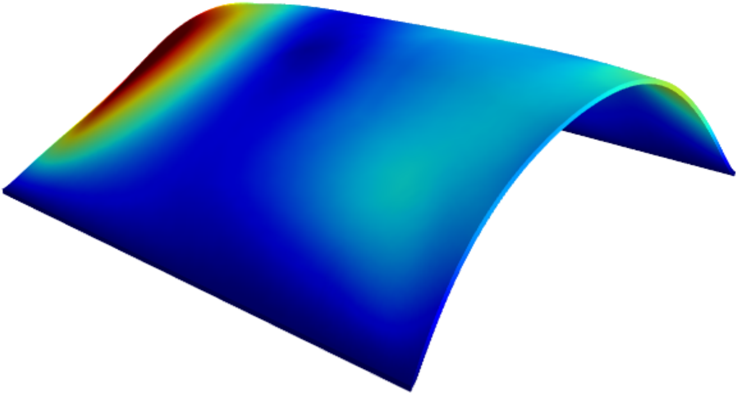
Flow around flag, 2-D

- Model “flag” by hinging left edge only



Membrane only, 3-D

- Preliminary results for single membrane simulation



Summary

- High-order accurate time integration of fully coupled FSI problems
- Partitioned Runge-Kutta methods derived from IMEX schemes
- Volumetric modeling of thin membrane structures
- Current work includes 3D simulations, more sophisticated mesh deformation, and real-world applications