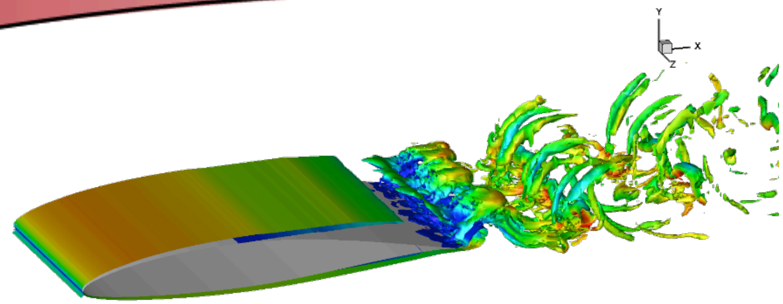


Lagrangian Coherent Structures and DNS with Discontinuous Galerkin methods

Gustaaf Jacobs, San Diego State University

Acknowledgment: Daniel Nelson (PhD Fall 2015), Bjoern Klose (PhD)

CPL

Solar Turbines
A Caterpillar Company

U.S. AIR FORCE

SAN DIEGO STATE UNIVERSITY

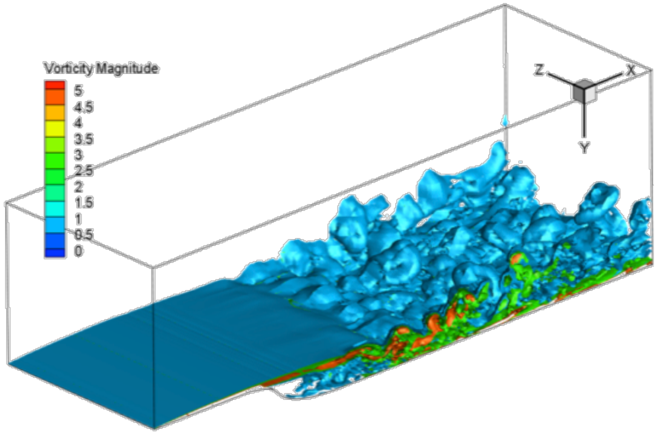
2017 SIAM Conference on Dynamical Systems , May 25, 2017

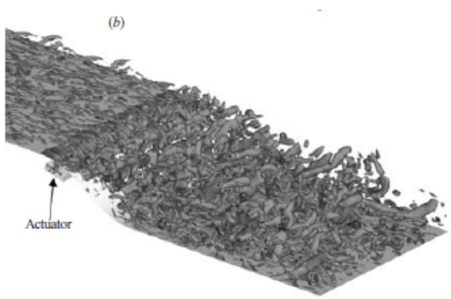
Outline

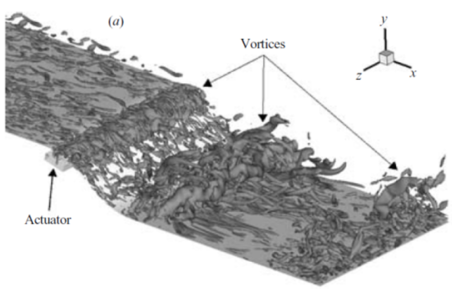
- Background and Motivation
  - Unsteady chaotic flow and higher-order DNS
  - LCS; FTLE with Finite Difference
- DG-FTLE (Finite-Time Lyapunov Exponent)
  - High-order FTLE with DG
  - Multiple FTLE fields from a single particle trace
  - Benchmark tests
- Examples
  - Rectangular cylinder
  - Airfoil
- Conclusions

Vortex Dominated Flows

- Small scale perturbations upstream and downstream of the separation point have a big impact on the global flow features
  - Directly related to effective flow control methods
- High-fidelity numerical methods are required that combine the following characteristics:
  - Accurately captures small scale features and unstable modes
  - Long time accuracy to trace vortex structures
  - High-fidelity boundary representation
- High-fidelity (quantitative) analysis of the flow topology is also required







Application of synthetic jet to separated flow.

[Dandois *et al.*, JFM,’07]

Direct Numerical Simulation

- Navier-Stokes Model

$$\mathbf{Q} + \mathbf{F}_x^a + \mathbf{G}_y^a + \mathbf{H}_z^a = \frac{1}{\mathrm{Re}_f}(\mathbf{F}_x^v + \mathbf{G}_y^v + \mathbf{H}_z^v)$$

  - First principle model with potential assumptions of constant density and temperature independent viscosity for low Mach number
- Requirement: Resolve the smallest scales
    - Turbulence up to the Kolmogorov scales
    - General unsteady flow: not perse known a priori
- Numerical Methods: FD, FV, FEM, SEM, etc…
  - Convergence/Accuracy: converge until grid independence; dispersion; numerical diffusion, geometric complexity, boundary accuracy
  - Efficiency/Feasiblity;
    - Degrees of freedom scale with Re^3 ; relatively low Reynolds numbers must be considered
  - Numerical methods that require few number of grid points per smallest scale improve accuracy and feasibility.

CPL

Low-Order vs. High-Order DNS

San Diego State University

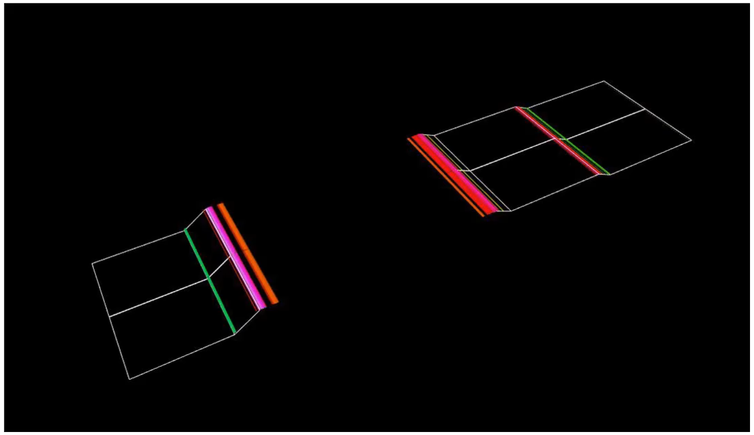
	Low Order	Higher Order
Polynomial order	p<=2	p>3
Implementation	Easy	Doable
Resolution per wave number	20-30 points	3-7 points
Smooth turbulence	Dissipation	No or Low Dissipation
Wave Propagation	Dispersion	No or Low Dispersion
Shocks/Discontinuity	Upwind stable, but dissipative	Gibb's phenomena
Fidelity	Limited or excessive resolution	Very good
Robustness	Typically very stable	Robust if done the right way
Flexiblitiy	Any complexity, overlap at boundary reduces accuracy	Any complexity with curved boundary elements

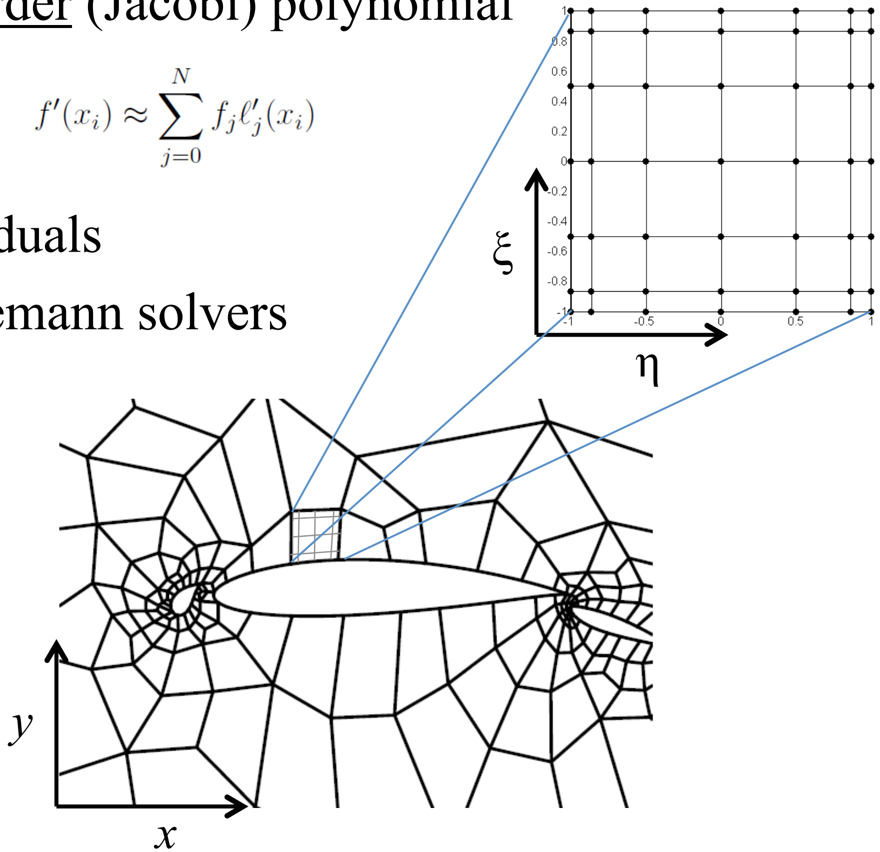
Discontinuous Galerkin

- Divide computational domain into elements
- Map each physical element onto a master element
- Approximate solution with higher-order (Jacobi) polynomial

$$f(x_i) \approx \sum_{j=0}^{N} \hat{f}_j L_j(x_i) = \sum_{j=0}^{N} f_j \ell_j(x_i) \qquad f'(x_i) \approx \sum_{j=0}^{N} f_j \ell'_j(x_i)$$

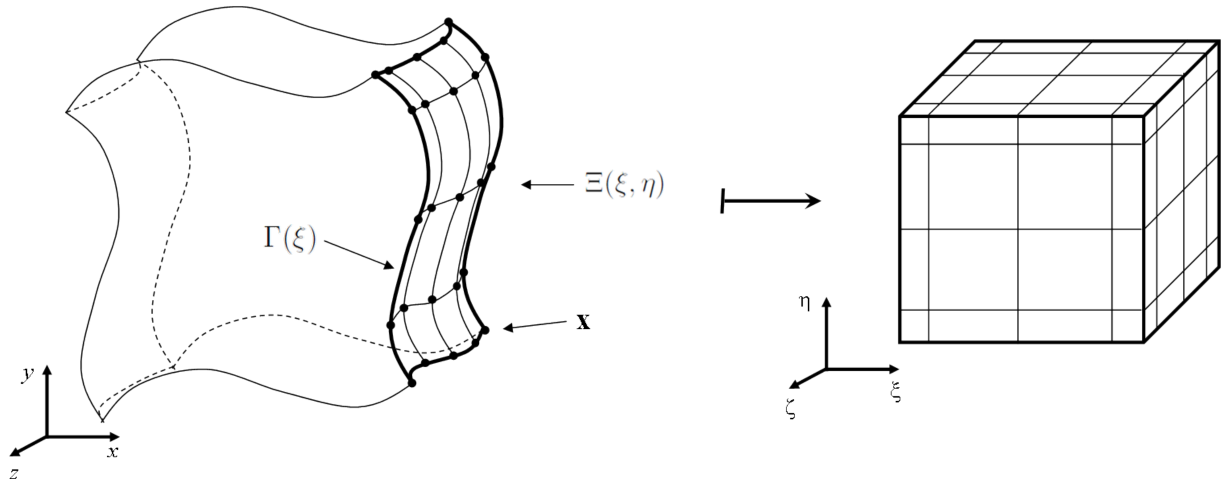
- Based on Method of Weighted Residuals
- Elements are connected through Riemann solvers





Deformed Elements

- The solution is mapped from physical space to the reference element:



- Mapping incorporates contributions from the faces, edges and corners:

$$\mathbf{x}(\xi,\eta,\zeta) = \sum_{i=1}^{6} p_i \Xi_i + \sum_{i=1}^{12} q_i \Gamma_i + \sum_{i=1}^{8} r_i \mathbf{x}_i$$

Faces: $$\Xi(\xi,\eta) = \sum_{i=0}^{N}\sum_{j=0}^{N} \mathbf{x}_{ij}\ell_i(\xi)\ell_j(\eta)$$

Edges: $$\Gamma(\xi) = \sum_{i=0}^{N} \mathbf{x}_i \ell_i(\xi)$$

p_i, q_i and r_i are shape functions:
e.g. $r_1 = (1-\xi)(1-\eta)(1-\zeta)$

- Metric terms and derivatives are computed from the mapping

$$\nabla_{\mathbf{x}} F(\mathbf{x}) = \frac{1}{J}\sum_{i=1}^{3} \frac{\partial}{\partial \xi^i}[(\mathbf{a}_j \times \mathbf{a}_k)F] \quad \text{where} \quad \mathbf{a}_i = \partial \mathbf{x}/\partial \xi_i$$

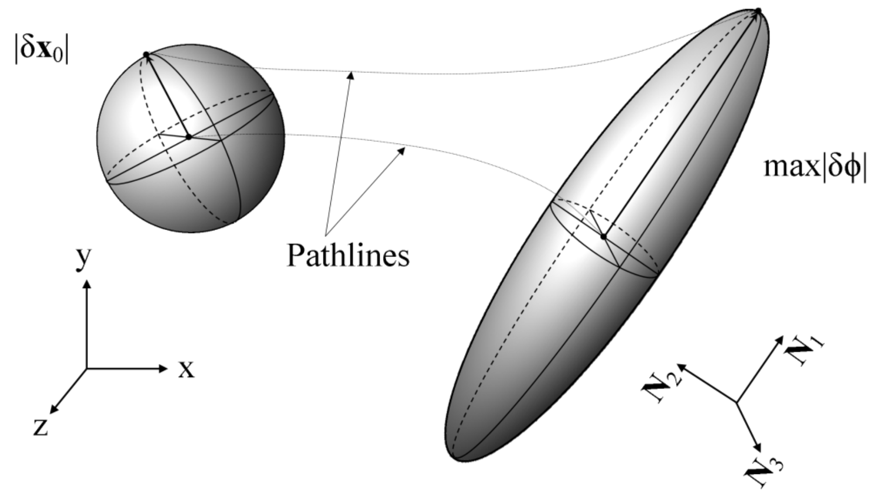
Finite-Time Lyapunov Exponent (FTLE)

- Dynamical systems of the form:

$$\begin{cases} \boldsymbol{x}(t_0; t_0, \boldsymbol{x}_0) = \boldsymbol{x}_0 \\ \dot{\boldsymbol{x}}(t; t_0, \boldsymbol{x}_0) = \boldsymbol{v}(\boldsymbol{x}(t; t_0, \boldsymbol{x}_0), t) \end{cases}$$

- Integrate particle trajectories to determine the flow map:

$$\boldsymbol{x}_0 \rightarrow \phi_{t_0}^{t} = \boldsymbol{x}(t; t_0, \boldsymbol{x}_0)$$
$$= \boldsymbol{x}_0 + \int_{t_0}^{t} \boldsymbol{v}(\boldsymbol{x}(\tau; t_0, \boldsymbol{x}_0), \tau)\, d\tau$$



- Exponentially growing perturbations in the flow map quantify a stretching rate:

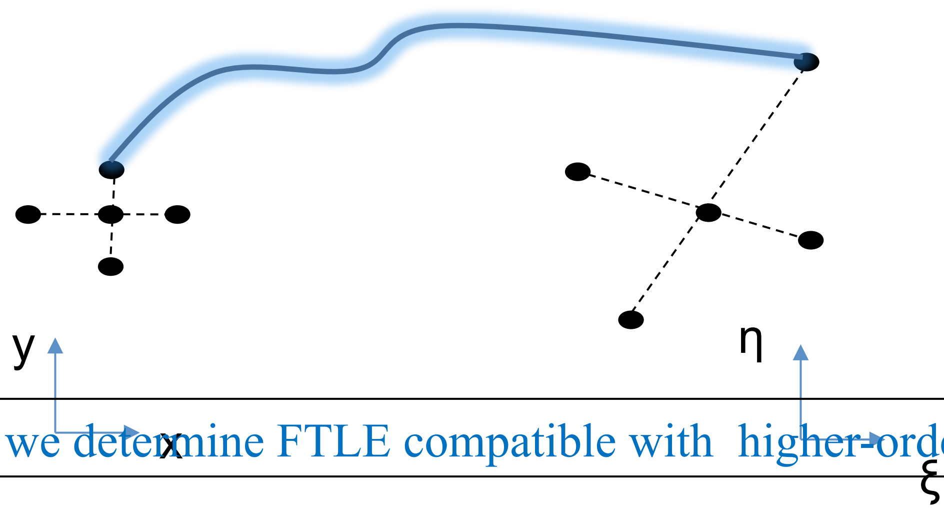
$$\max|\delta \boldsymbol{x}| = \sqrt{\lambda_{\max}(\boldsymbol{C})}\left|\overline{\delta \boldsymbol{x}_0}\right| \quad \text{where} \quad \boldsymbol{C} = \frac{\partial \phi_{t_0}^{t}}{\partial \boldsymbol{x}_0}^{*} \frac{\partial \phi_{t_0}^{t}}{\partial \boldsymbol{x}_0} \qquad \max|\delta \boldsymbol{x}| = e^{\sigma|T|}\left|\overline{\delta \boldsymbol{x}_0}\right|$$

- Maximal material stretching measured by the FTLE (σ):

$$e^{\sigma|T|} = \sqrt{\lambda_{\max}(\boldsymbol{C})} \quad \Rightarrow \quad \sigma = \frac{1}{|T|}\ln\sqrt{\lambda_{\max}(\boldsymbol{C})} = \frac{1}{|T|}\ln\left\|\frac{\partial \phi_{t_0}^{t}}{\partial \boldsymbol{x}_0}\right\|$$

FD and FTLE

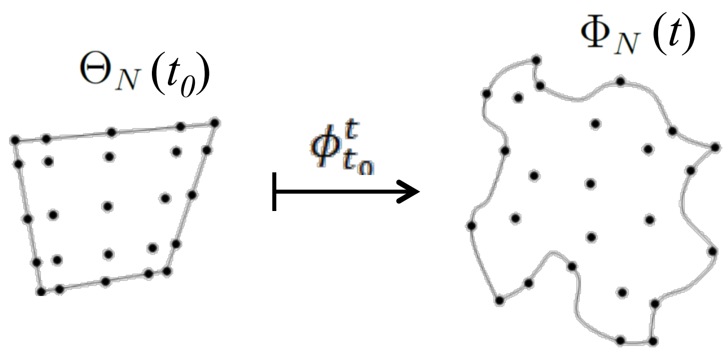
- Use Finite Difference to determine Cauchy-Green strain tensor
  - Seed five particles on an orthogonal grid
  - Trace fluid particles in velocity field, which is usually stored in separate files and post-processed
    - requires lots of memory/storage
    - large Δt
  - Use central FD stencil to determine Cauchy-Green strains: $\partial\xi/\partial x$, $\partial\xi/\partial y$, $\partial\eta/\partial x$, $\partial\eta/\partial y$
  - Eigenvalue of the CG tensor determines FTLE



Can we determine FTLE compatible with higher-order solvers?

DG-FTLE

- Fluid particles are initialized at the Lobatto quadrature nodes.
- Particles are integrated in time with a 3rd-order Adams-Bashforth scheme.



The flow map is approximated by a high-order polynomial interpolant, Φ_N.

- After the time interval, *T,* DG operators are used to determine the deformation gradient:

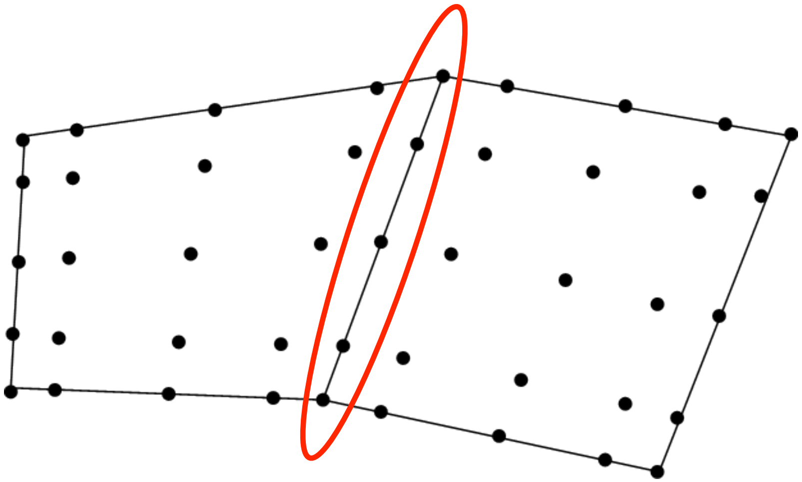
$$f'(\xi_i) \approx \left(I_N f(\xi_i)\right)' = \sum_{j=0}^{N} f_j \ell_j'(\xi_i) = \sum_{j=0}^{N} D_{ij} f_j$$

- Under mapped coordinates (2D):

$$\begin{pmatrix} \frac{\partial \Phi}{\partial x_0} \\ \frac{\partial \Phi}{\partial y_0} \end{pmatrix} = \begin{pmatrix} \frac{\partial \xi}{\partial x_0} & \frac{\partial \eta}{\partial x_0} \\ \frac{\partial \xi}{\partial y_0} & \frac{\partial \eta}{\partial y_0} \end{pmatrix} \begin{pmatrix} \frac{\partial \Phi}{\partial \xi} \\ \frac{\partial \Phi}{\partial \eta} \end{pmatrix} \quad \Rightarrow \quad \begin{aligned} \frac{\partial \Phi}{\partial x_0} &= \frac{1}{J}\left[\left(\sum_{k=0}^{N} D_{ik}^{(\xi)} \Phi_{kj}\right)\frac{\partial y_0}{\partial \eta} - \left(\sum_{k=0}^{N} D_{jk}^{(\eta)} \Phi_{ik}\right)\frac{\partial y_0}{\partial \xi}\right] \\ \frac{\partial \Phi}{\partial y_0} &= \frac{1}{J}\left[\left(\sum_{k=0}^{N} D_{jk}^{(\eta)} \Phi_{ik}\right)\frac{\partial x_0}{\partial \xi} - \left(\sum_{k=0}^{N} D_{ik}^{(\xi)} \Phi_{kj}\right)\frac{\partial x_0}{\partial \eta}\right] \end{aligned}$$

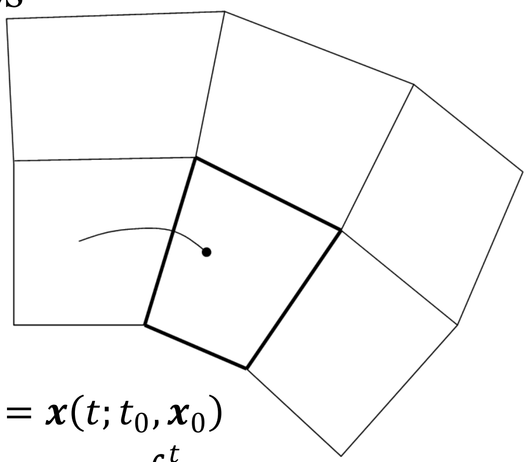
[Nelson and Jacobs, *ASME, '13*]

Fluid Particle Tracking Algorithm



Fluid particles are initialized at the Lobatto quadrature nodes=>
no connectivity issues

Duplicate particles are present at the subdomian boundaries, are removed to trace fewer particles



Fluid tracers are integrated in a 3-step algorithm:

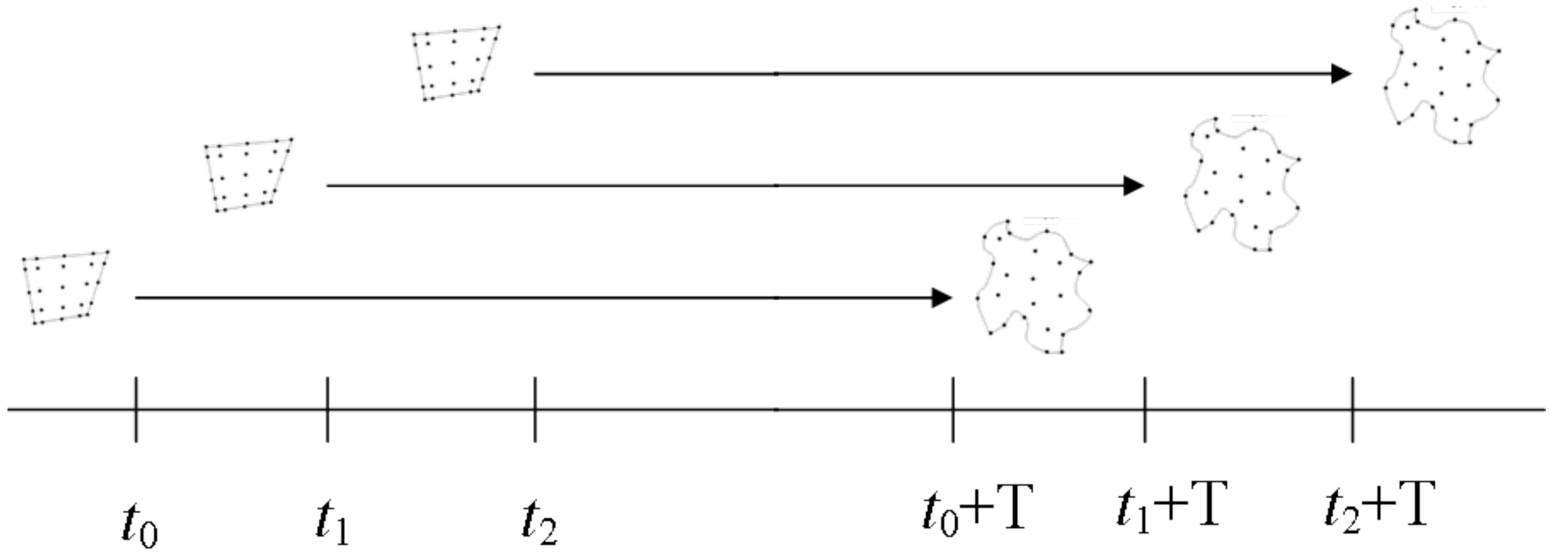
1. The host cell of the particle is located
2. The fluid velocity is interpolated from the DG grid to the particle's location: expensive!
3. The particle velocity is integrated in time with a 3rd-order Adams-Bashforth scheme

$$\boldsymbol{x}_0 \rightarrow \phi_{t_0}^{t} = \boldsymbol{x}(t; t_0, \boldsymbol{x}_0)$$

$$= \boldsymbol{x}_0 + \int_{t_0}^{t} \boldsymbol{v}(\boldsymbol{x}(\tau; t_0, \boldsymbol{x}_0), \tau)\, d\tau$$

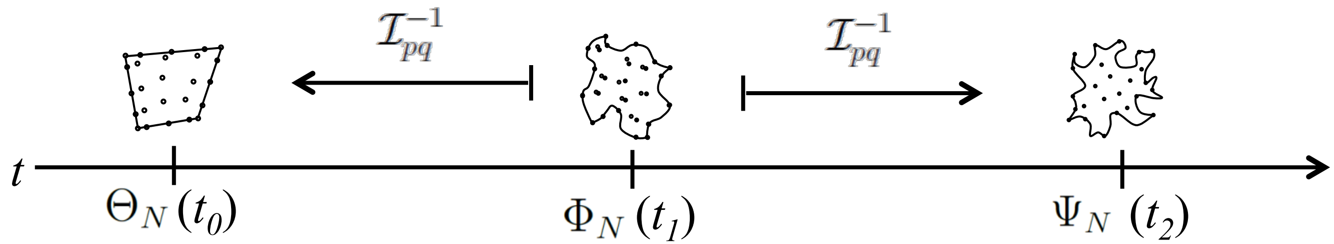
Sequence of FTLEs

- To analyze the temporal evolution of a flow with the FTLE field, multiple FTLE fields must be computed, normally requiring redundant particle integrations.



Multiple Flow Maps

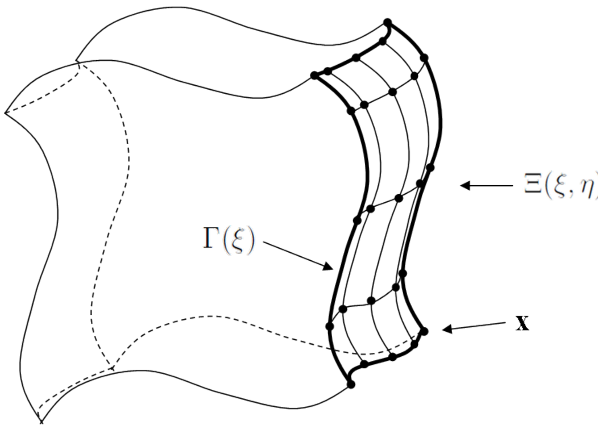
- Multiple FTLEs can be computed from a single particle trace with interpolation:
  1. Orthogonal polynomial basis is constructed at time t_1.
  2. The particles at t_1 are mapped to the unit square.
  3. Construct the interpolation operator: $\mathcal{I}_{pq} = \ell_i(\alpha_{lmn})\ell_j(\beta_{lmn})\ell_k(\gamma_{lmn})$
  4. Interpolate to later time ($t_2 > t_1$) or earlier time ($t_0 < t_1$) and compute FTLE.



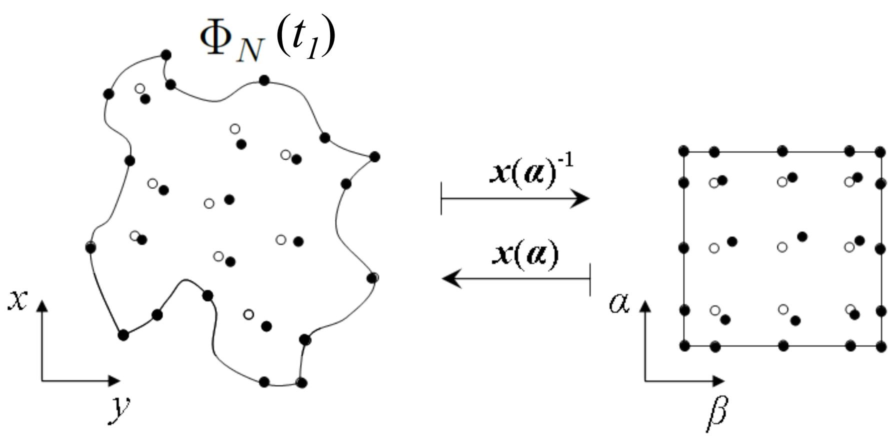
[Nelson and Jacobs, *JCP, '15*]

Inverse Mapping

- Deformed subdomains are constructed from the particle locations at a given time.
- The isoparametric mapping is built from the particles initialized in the original subdomain.
    - The faces are parametrized with particles initialized at the edges of the original subdomain.



- Once the faces are constructed, the interior particle locations are mapped to the reference element through the inverse of the isoparametric map.



Interpolation

- Given the locations of the particles in the reference element, the flow map is interpolated from the quadrature points as follow

$$\Psi'_{ijk} = \sum_{l=0}^{N}\sum_{m=0}^{N}\sum_{n=0}^{N} \Psi_{lmn}\ell_l(\alpha'_{ijk})\ell_m(\beta'_{ijk})\ell_n(\gamma'_{ijk})$$

- Hence,

$$\bar{\Psi}'_p = \mathcal{I}^{-1}_{pq}\bar{\Psi}_q$$

$$p = i(N+1)^2 + j(N+1) + k$$

$$q = l(N+1)^2 + m(N+1) + n$$

- The conditioning of the I operator is related to the deformation of the flow map

$$\Delta\alpha \equiv \max|\boldsymbol{\alpha}_{ij} - \boldsymbol{\alpha}'_{ij}|$$

$$\kappa(I) \equiv \frac{\lambda_{max}(I)}{\lambda_{min}(I)}$$

Gyre Flow

CPL

San Diego State University

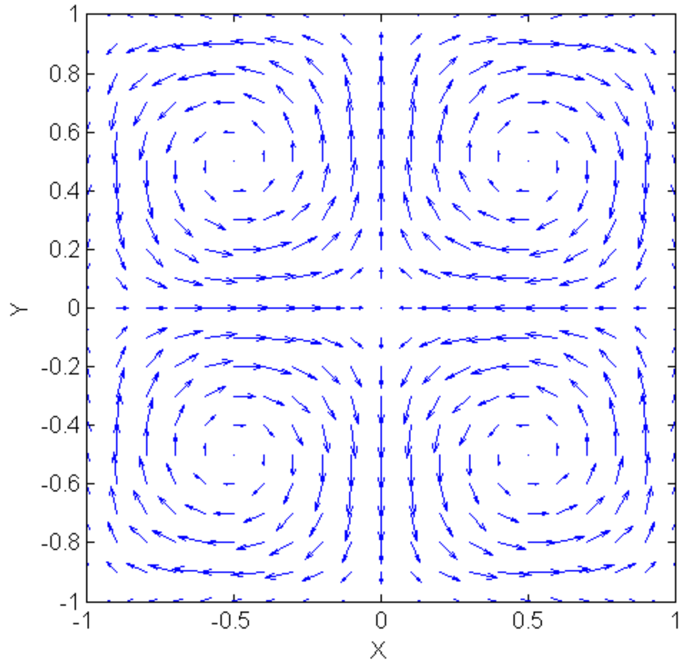
2D Gyre Flow

Velocity given by:

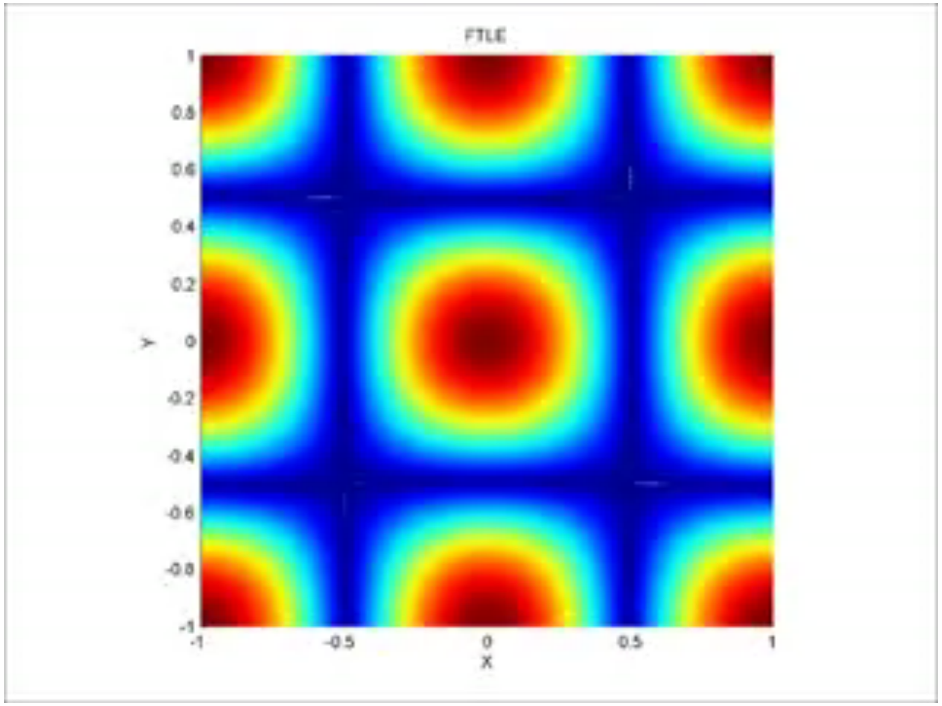
$$u = -\pi A \sin(\pi x)\cos(\pi y)$$
$$v = \pi A \sin(\pi y)\cos(\pi x)$$

- Gyre is a spatially periodic flow consisting of recirculating cells.
- Note FTLE ridges forming around the edges of the vortices.

Velocity Field

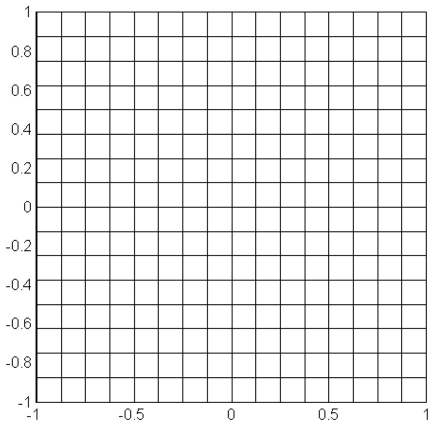


FTLE Field

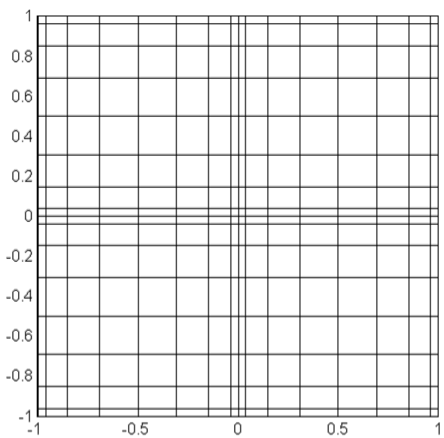


Spectral Convergence

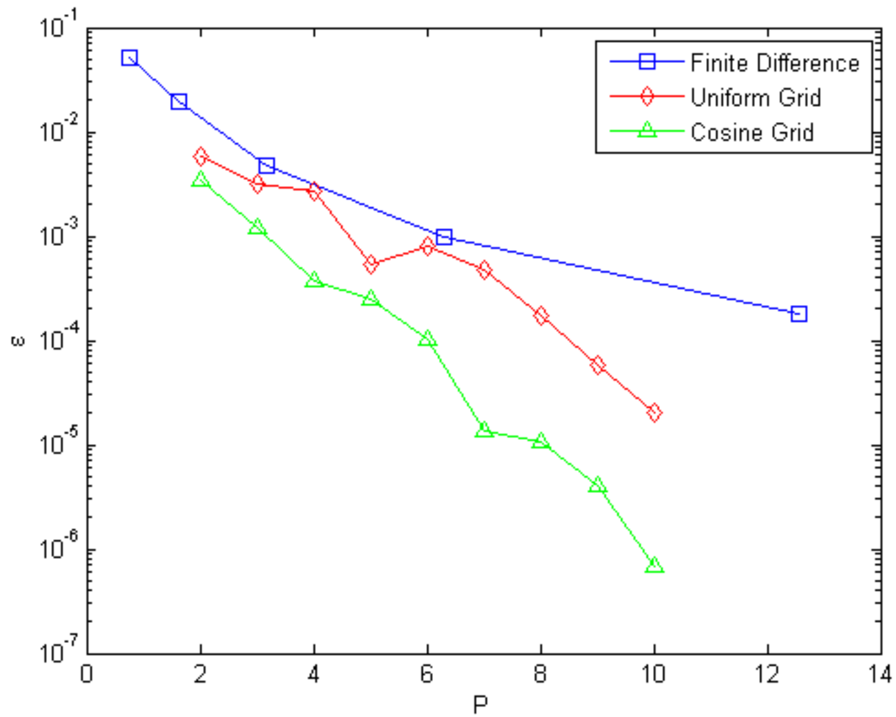
- Accuracy and convergence rate increases with grid refinement.



Regular Grid



Cosine Grid



Inviscid Vortex

Vortex Advected by Uniform Flow

Velocity given by:

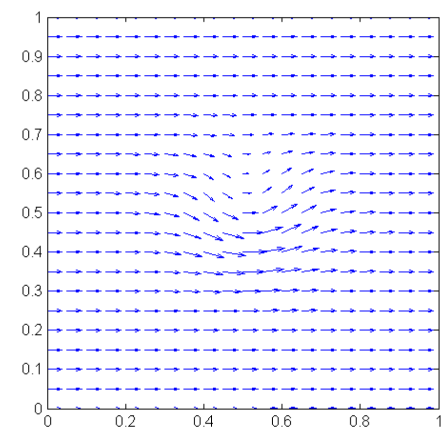
$$\delta u = -(U_\infty * \beta) * (y - Y_c)/R * \exp(-r^2/2)$$

$$\delta v = (U_\infty * \beta) * (x - X_c)/R * \exp(-r^2/2)$$

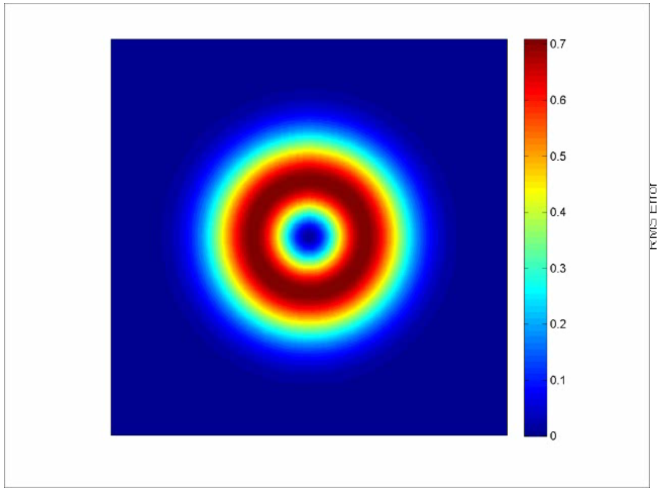
$$u_0 = U_\infty + \delta u, v_0 = \delta v$$

- The vortex flow is computed using the DG Euler solver.
- The spectral FTLE algorithm is implemented within the code and computed on-the-fly.

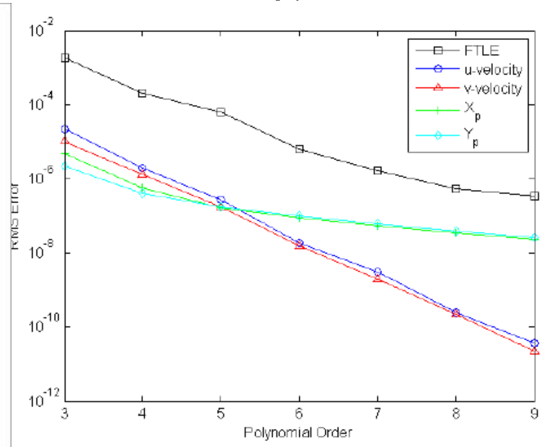
Velocity Field



FTLE Field



Convergence



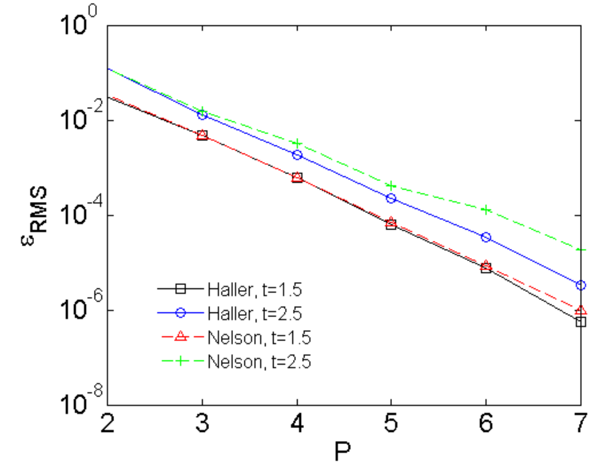
- Errors include numerical errors in particle tracking, computation of the deformation gradient, and numerical errors in DG.

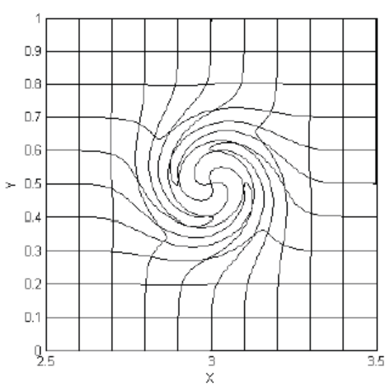
Conditioning

SAN DIEGO STATE UNIVERSITY

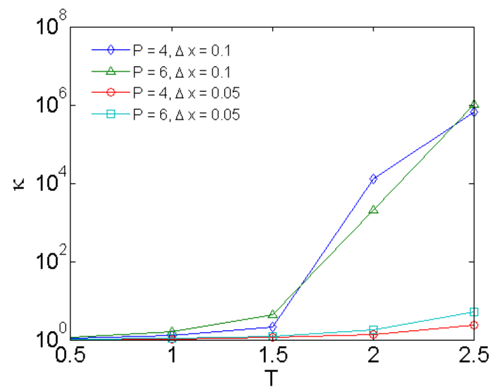
Error Analysis

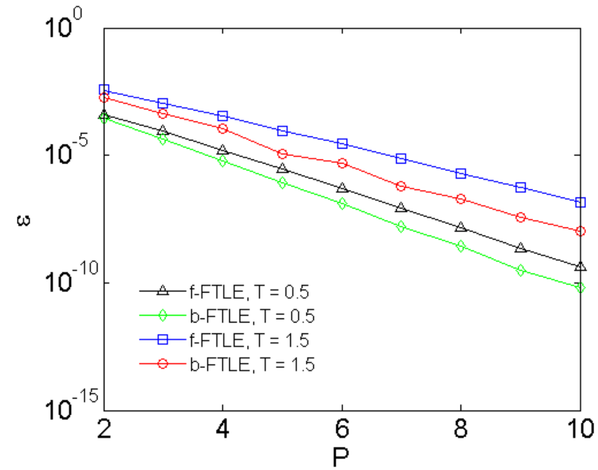
- Spectral convergence
- High deformation leads to high condition number
- Condition number increases with larger subdomains and higher-order polynomial
- Condition number decreases with grid refinement





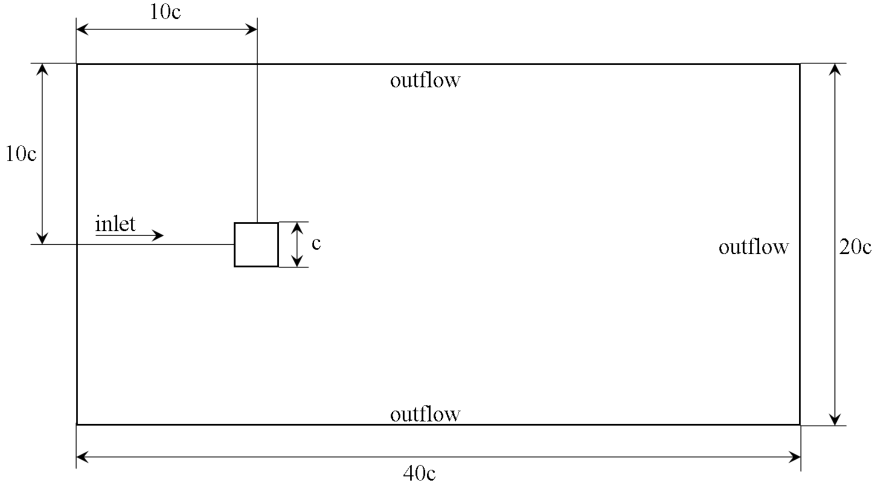
Subdomain deformation at T = 2.5



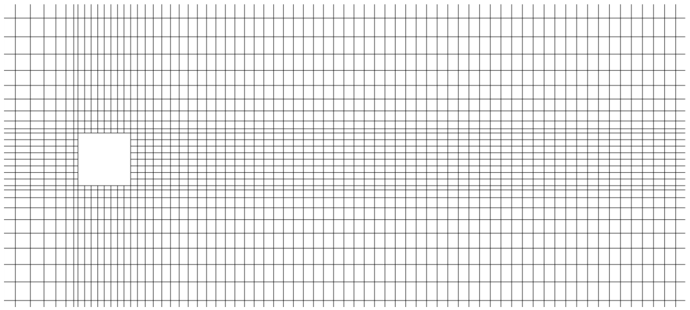


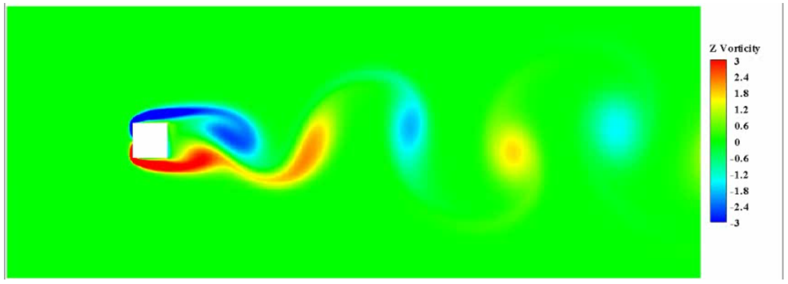
Square Cylinder

Viscous flow over square cylinder



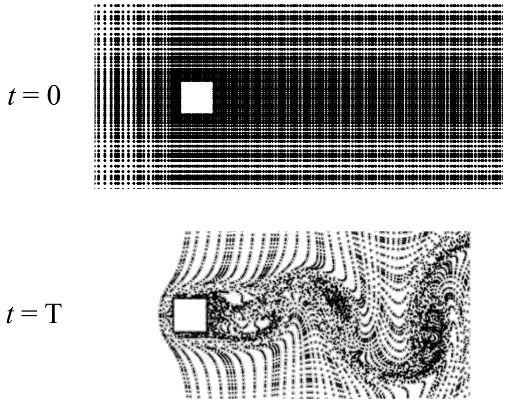
- Re = 150, based on cylinder width
- M = 0.3
- 6th-order

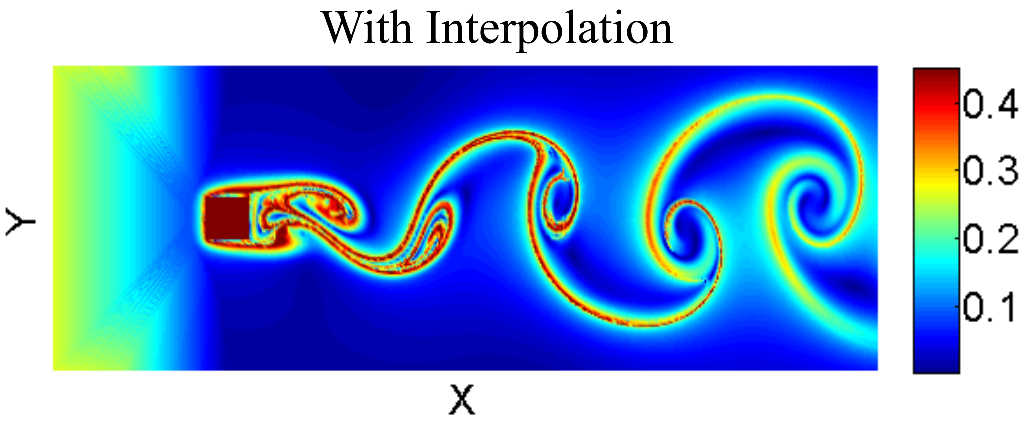


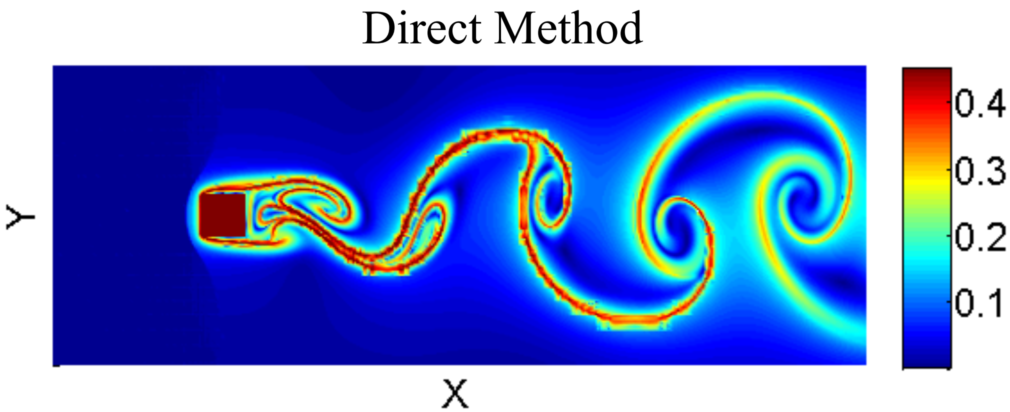


Backward FTLE

- High-order method from forward-time flow map.

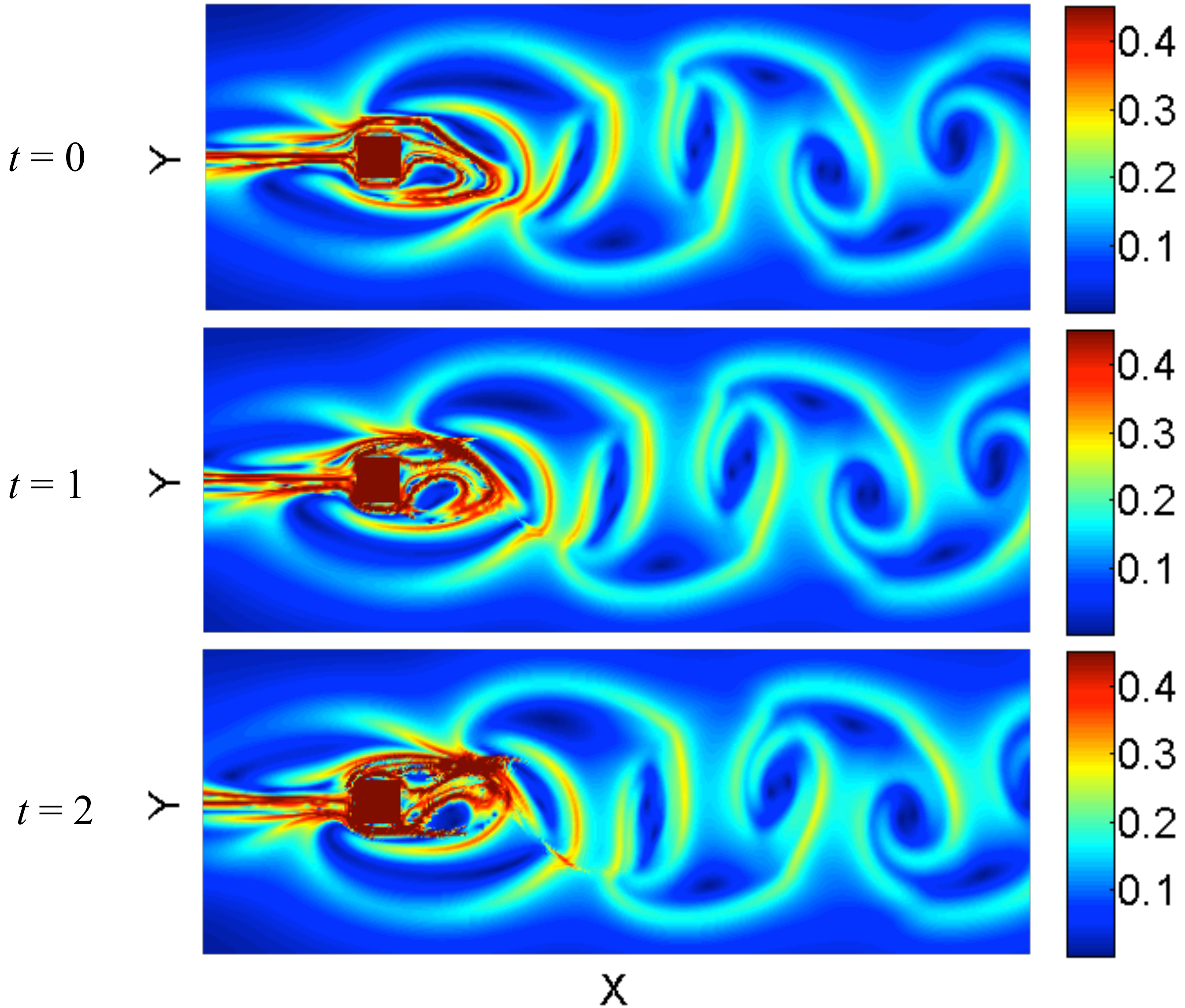






- Agrees well with standard method.
- Some difference in near wake due to poor conditioning.

Multiple Forward FTLE Fields



Interpolation Conditioning

San Diego State University

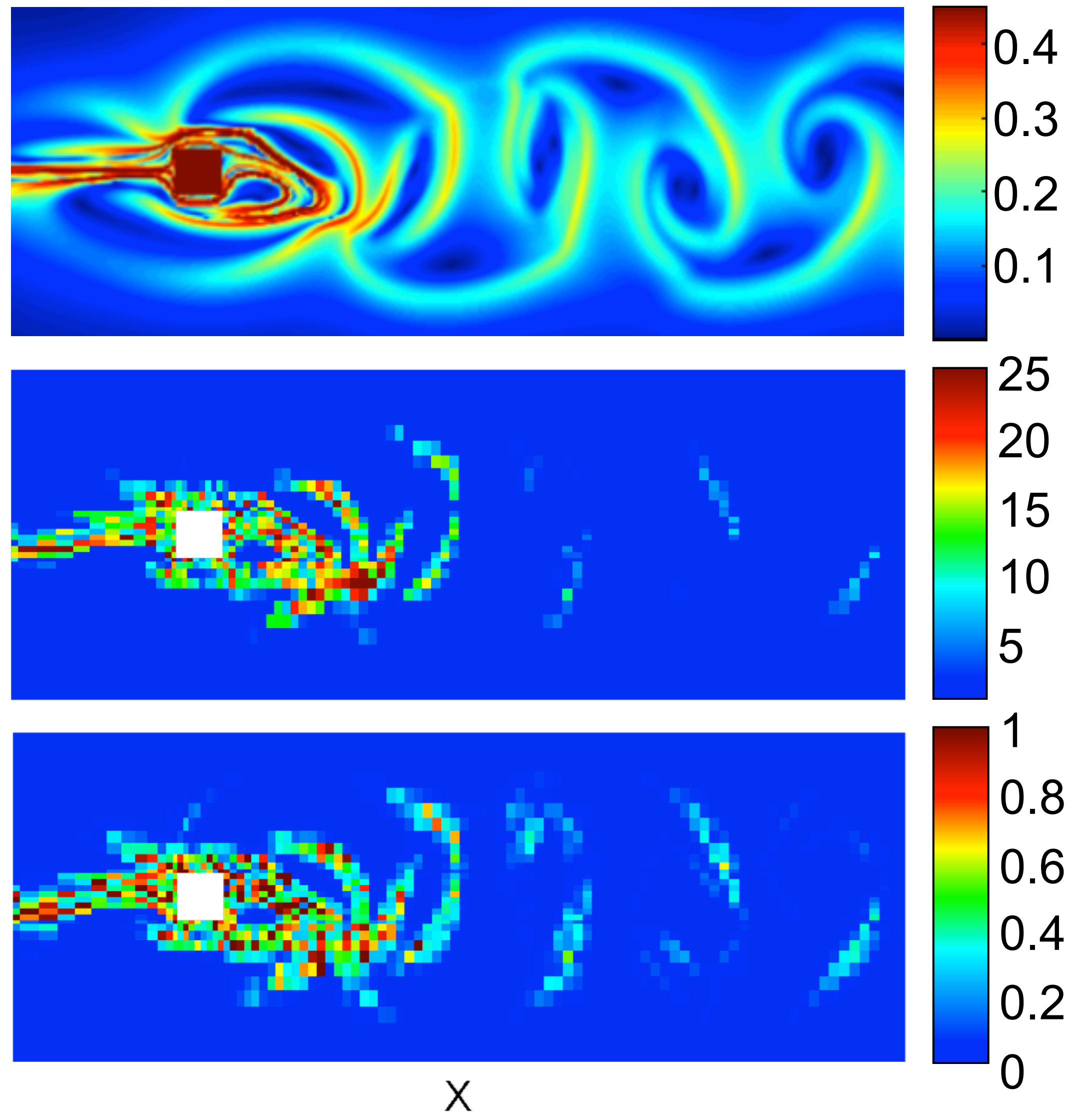
Forward FTLE (T = 10)

Condition Number

$$\kappa(I) \equiv \frac{\lambda_{max}(I)}{\lambda_{min}(I)}$$

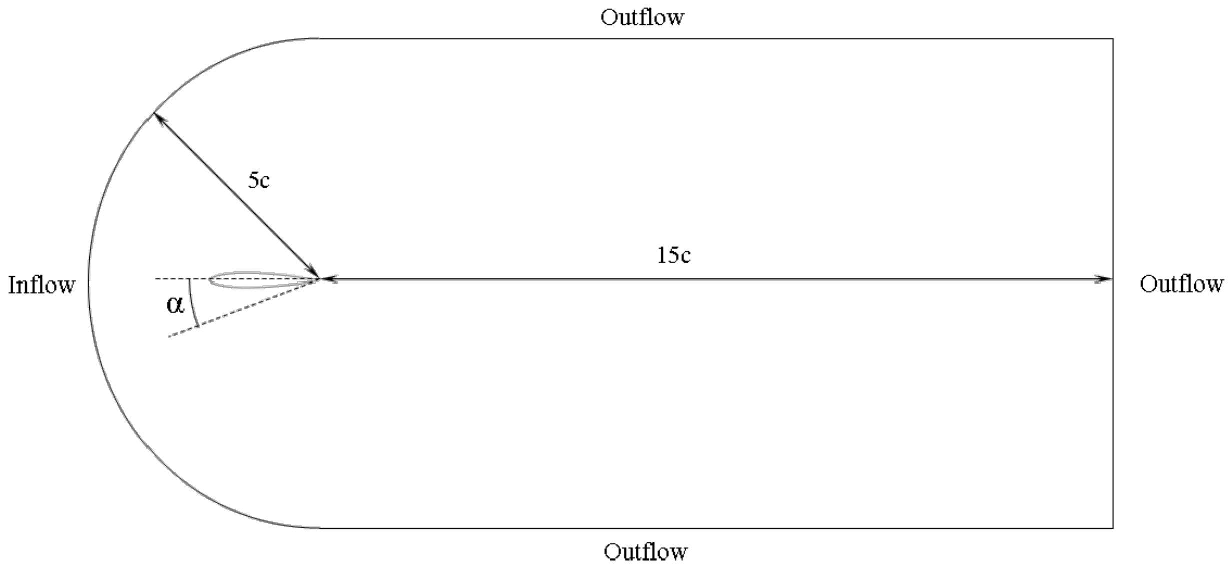
Parameter $\Delta\alpha$

$$\Delta\alpha \equiv \max|\boldsymbol{\alpha}_{ij} - \boldsymbol{\alpha}'_{ij}|$$



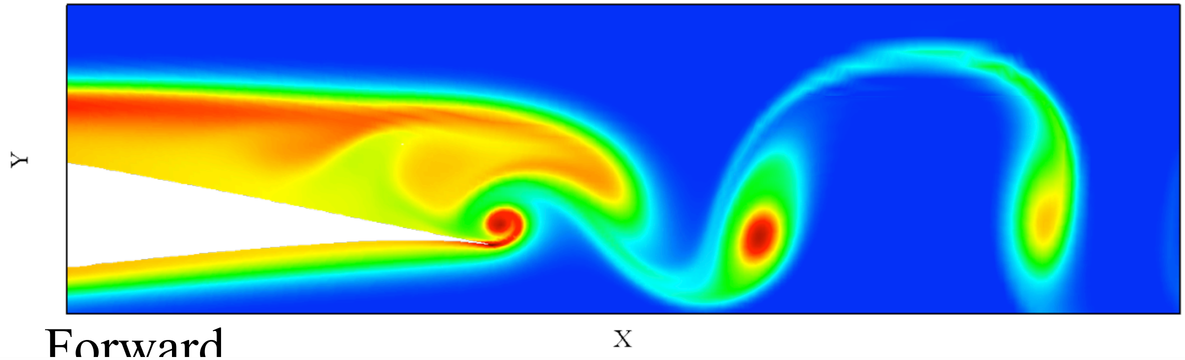
2D Airfoil DNS

- Problem Parameters
  - Re = 20,000
  - Pr = 0.72
  - CFL = 0.8
  - AOA = 4°
- NACA 65(1)-412 Airfoil
- Polynomial orders
  - Curved-sided mesh, P = 4, 6, 8, 10, 12
  - Straight-sided mesh, P = 4, 6, 8, 10, 12



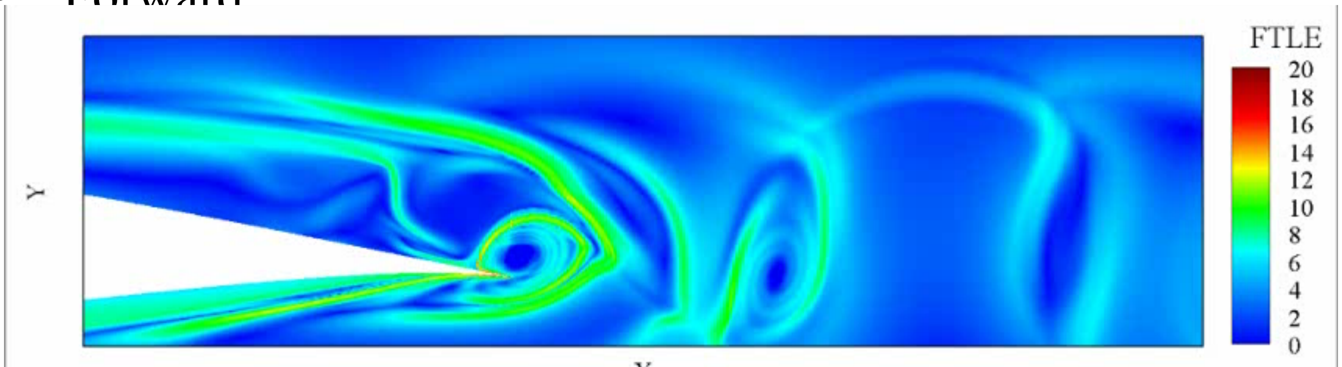
[Nelson, Jacobs & Kopriva, *TCFD, '15*]

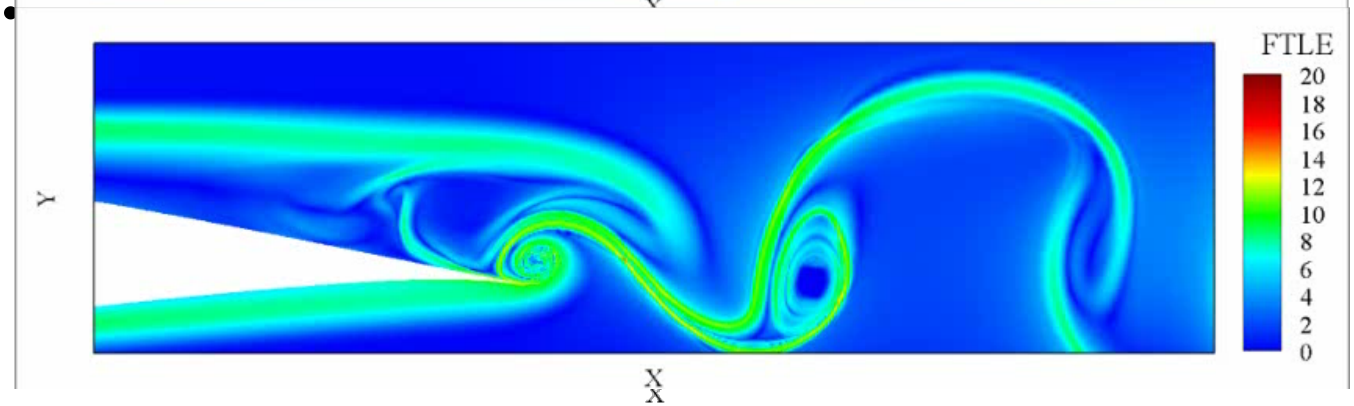
FTLE Field



Entropy: P = 12,
FTLE: P = 24,
T = 0.37

- Forward

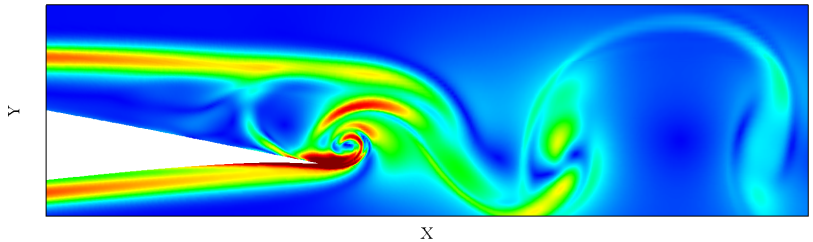


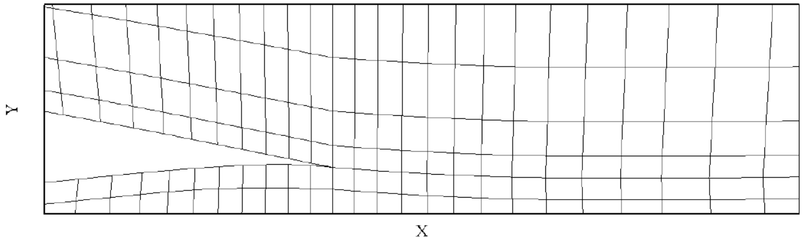


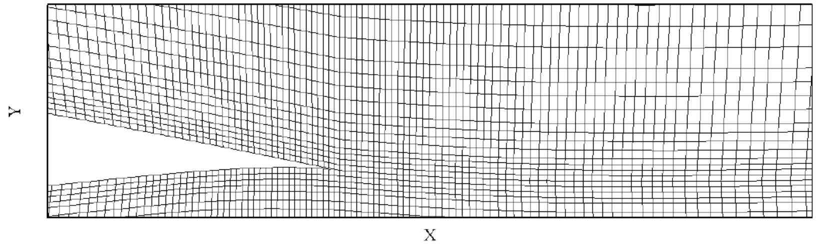
Backward FTLE

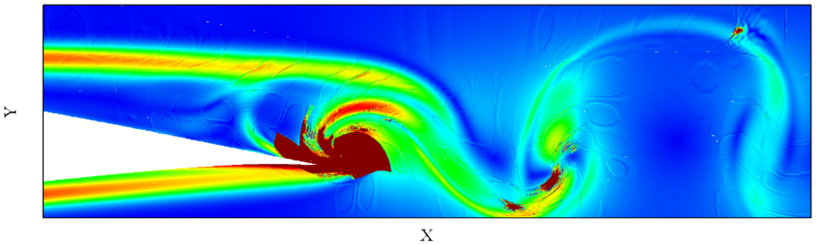
Mesh Refinement

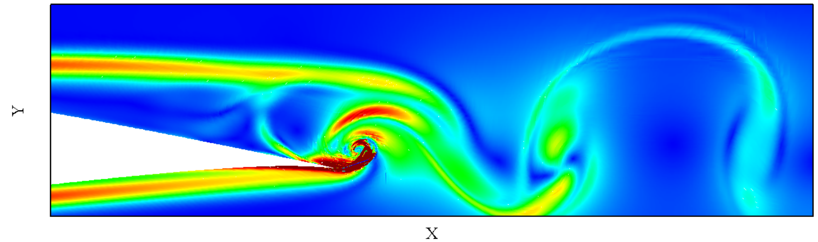
- Conditioning can be improved by refining the mesh.
- Coarse: N = 24, Fine: N = 6





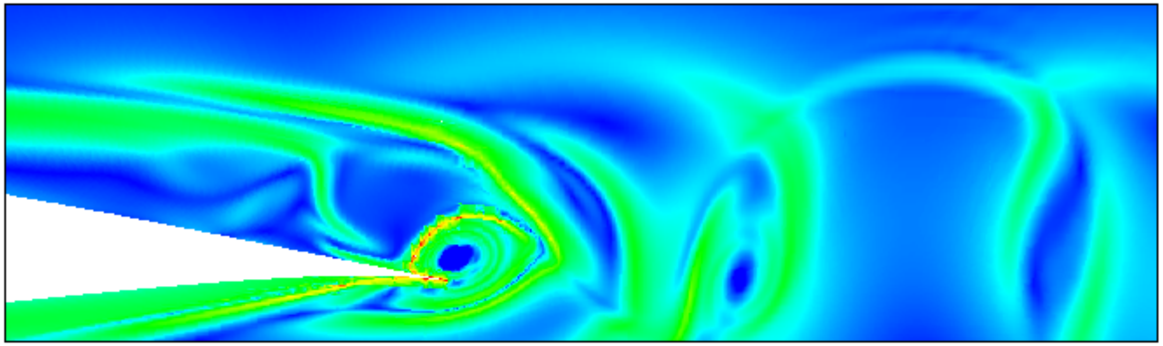




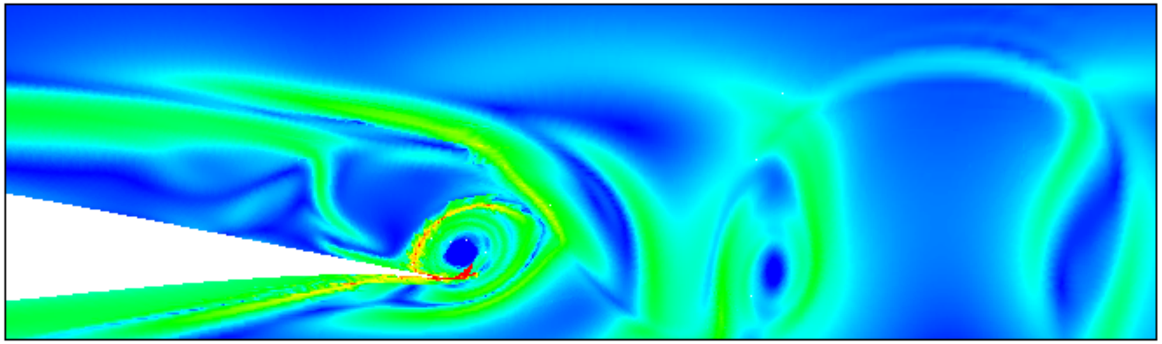


Multiple Forward FTLE Fields

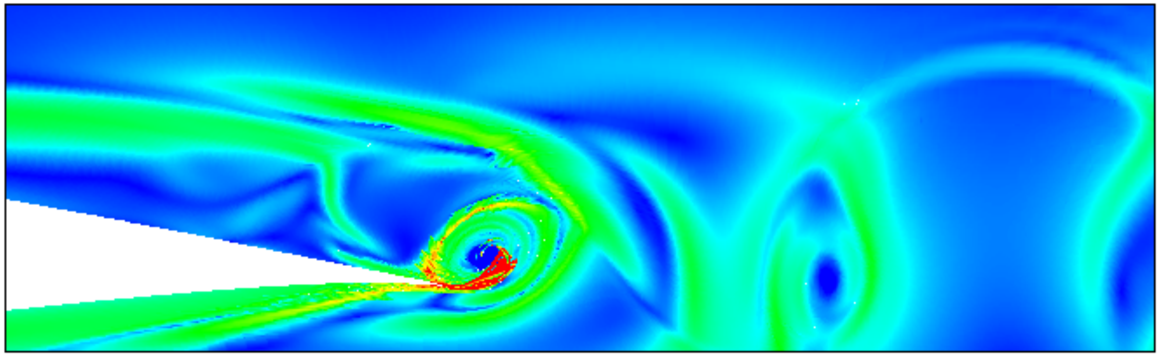
t = 0



t = 0.02



t = 0.04

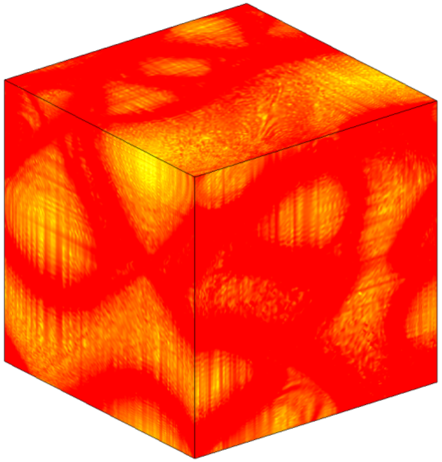


ABC Flow

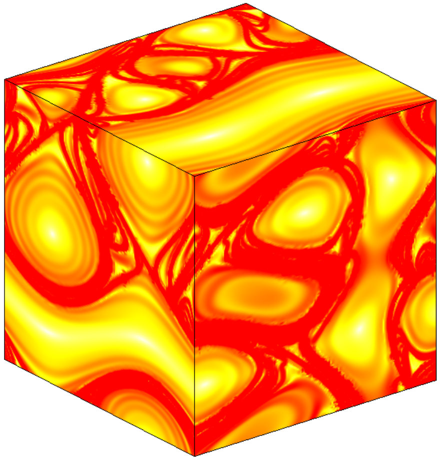
- 1,000 elements
- P = 24 (13,997,521 particles)
- T = 2
- Exact velocity (no interpolation)

$$u(x,y,z) = A\sin(z) + C\cos(y)$$
$$v(x,y,z) = B\sin(x) + A\cos(z)$$
$$w(x,y,z) = C\sin(y) + B\cos(x)$$
$$A = \sqrt{3}, B = \sqrt{2}, C = 1$$

Unfiltered



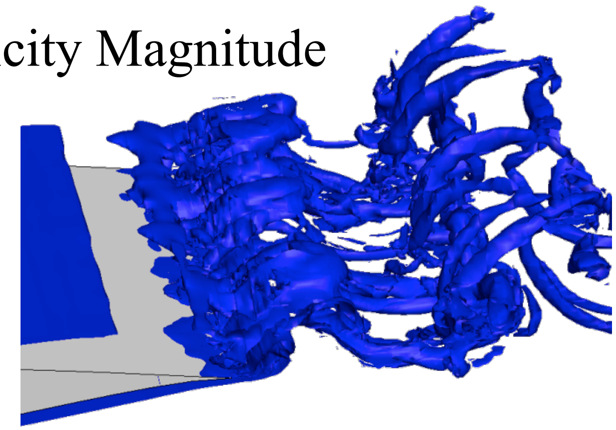
Filtered

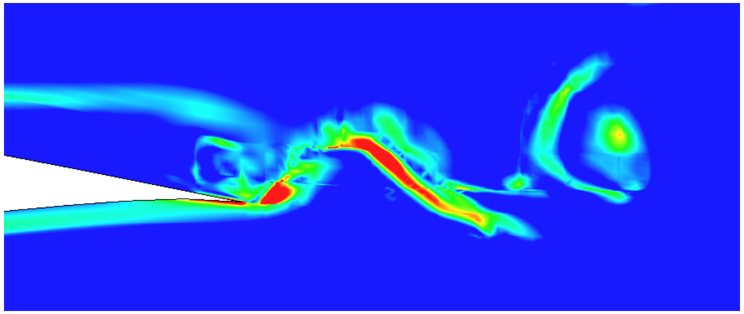


3D Airfoil

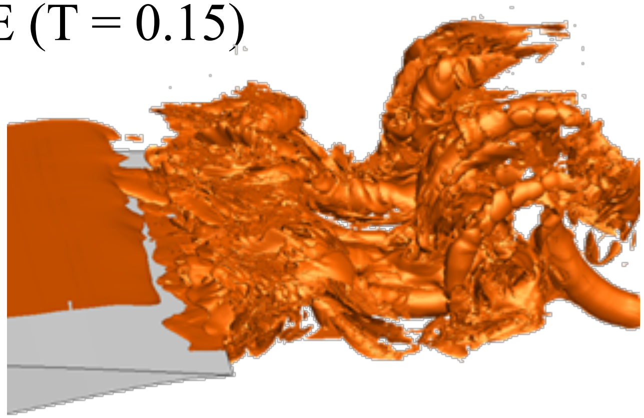
- Fluid Solution: P = 8
- FTLE: P = 36

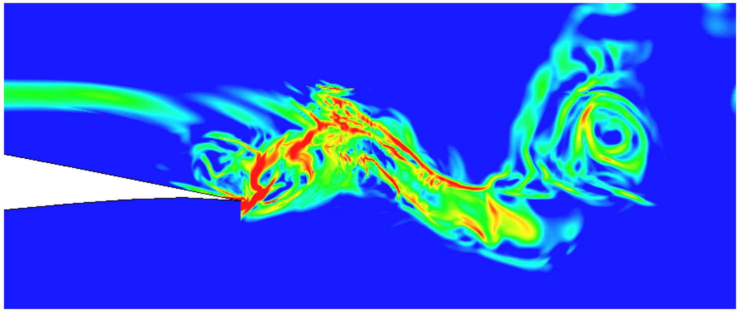
Vorticity Magnitude





FTLE (T = 0.15)





Conclusions

A FTLE algorithm is developed that commutes with a higher-order DG-based DNS solver

- Exponentially convergent
- Uses same grid as fluid solver
  - Geometric complexity
  - Prevents expensive interpolation to determine flow map
- Multiple FTLEs can determined in parallel with DNS preventing expensive post-processing
- Overhead is 10-50% depending on polynomial order