Lagrangian Coherent Structures and DNS with Discontinuous Galerkin methods

Gustaaf Jacobs, San Diego State University

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

• Background and Motivation
  o Unsteady chaotic flow and higher-order DNS
  o LCS; FTLE with Finite Difference

• DG-FTLE (Finite-Time Lyapunov Exponent)
  o High-order FTLE with DG
  o Multiple FTLE fields from a single particle trace
  o Benchmark tests

• Examples
  o Rectangular cylinder
  o Airfoil

• Conclusions
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

\[ Q + F^x + G^y + H^z = \frac{1}{Re_f} (F^v + G^y + H^v) \]

  o 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...
  o Convergence/Accuracy: converge until grid independence; dispersion; numerical diffusion, geometric complexity, boundary accuracy
  o Efficiency/Feasibility;
    • Degrees of freedom scale with \( Re^3 \); relatively low Reynolds numbers must be considered
  o Numerical methods that require few number of grid points per smallest scale improve accuracy and feasibility.
## Low-Order vs. High-Order DNS

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

We can try and fix issues in low-order codes…. or …. prevent them from the start by developing high-order solvers.
Discontinuous Galerkin

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

\[ f(x_i) \approx \sum_{j=0}^{N} \hat{f}_j L_j(x_i) = \sum_{j=0}^{N} f_j \ell_j(x_i) \]

\[ f'(x_i) \approx \sum_{j=0}^{N} f_j' \ell_j(x_i) \]

- 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:

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

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

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

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

- Metric terms and derivatives are computed from the mapping

$$\nabla_x F(x) = \frac{1}{J} \sum_{i=1}^{3} \frac{\partial}{\partial \xi_i} \left[ (\mathbf{a}_j \times \mathbf{a}_k) F \right] \quad \text{where} \quad \mathbf{a}_i = \frac{\partial \mathbf{x}}{\partial \xi_i}$$
Finite-Time Lyapunov Exponent (FTLE)

- Dynamical systems of the form:

\[
\begin{align*}
\{x(t_0; t_0, x_0) &= x_0 \\
\dot{x}(t; t_0, x_0) &= v(x(t; t_0, x_0), t)
\end{align*}
\]

- Integrate particle trajectories to determine the flow map:

\[
x_0 \to \phi^t_{t_0} = x(t; t_0, x_0) = x_0 + \int_{t_0}^{t} v(x(\tau; t_0, x_0), \tau) \, d\tau
\]

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

\[
\max |\delta x| = \sqrt{\lambda_{\text{max}}(C)} |\delta x_0| \quad \text{where} \quad C = \frac{\partial \phi^t_{t_0}}{\partial x_0} \frac{\partial \phi^t_{t_0}}{\partial x_0} \quad \max |\delta x| = e^{\sigma |T|} |\delta x_0|
\]

- Maximal material stretching measured by the FTLE (\(\sigma\)):

\[
e^{\sigma |T|} = \sqrt{\lambda_{\text{max}}(C)} \quad \Rightarrow \quad \sigma = \frac{1}{|T|} \ln \sqrt{\lambda_{\text{max}}(C)} = \frac{1}{|T|} \ln \left\| \frac{\partial \phi^t_{t_0}}{\partial x_0} \right\|
\]

Lagrangian Coherent Structures
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 $\Delta t$
  - Use central FD stencil to determine Cauchy-Green strains: $\frac{\partial \xi}{\partial x}$, $\frac{\partial \xi}{\partial y}$, $\frac{\partial \eta}{\partial x}$, $\frac{\partial \eta}{\partial y}$
  - Eigenvalue of the CG tensor determines FTLE

Can we determine FTLE compatible with higher-order solvers?
- Fluid particles are initialized at the Lobatto quadrature nodes.
- Particles are integrated in time with a 3\textsuperscript{rd}-order Adams-Bashforth scheme.

\[ \Theta_N(t_0) \rightarrow \Phi_N(t) \]

The flow map is approximated by a high-order polynomial interpolant, \( \Phi_N \).

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

\[ f'(\xi_i) \approx (l_N f(\xi_i))' = \sum_{j=0}^{N} f_j l_j(\xi_i) = \sum_{j=0}^{N} D_{ij} f_j \]

- Under mapped coordinates (2D):

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

\[ [\text{Nelson and Jacobs, ASME, '13}] \]
Fluid Particle Tracking Algorithm

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

Duplicate particles are present at the subdomain 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

\[ x_0 \rightarrow \phi_{t_0}^t = x(t; t_0, x_0) = x_0 + \int_{t_0}^{t} v(x(\tau; t_0, x_0), \tau) \, d\tau \]
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: $I_{pq}^{-1}(\alpha_{lmn})\ell_j(\beta_{lmn})\ell_k(\gamma_{lmn})$
  4. Interpolate to later time ($t_2 > t_1$) or earlier time ($t_0 < t_1$) and compute FTLE.

[Nelson and Jacobs, *JCP*, '15]
• 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.
  o 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,

\[ \Psi'_p = \mathcal{T}^{-1}_{pq} \Psi_q \]

\[ p = i(N+1)^2 + j(N+1) + k \]

\[ q = l(N+1)^2 + m(N+1) + n \]

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

\[ \Delta \alpha \equiv \max |\alpha_{ij} - \alpha'_{ij}| \]

\[ \kappa(I) \equiv \frac{\lambda_{\max}(I)}{\lambda_{\min}(I)} \]
Gyre Flow

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.
Inviscid Vortex

Vortex Advection by Uniform Flow

Velocity given by:

\[
\begin{align*}
\delta u &= -(U_\infty \beta)(y - Y_c)/R \exp(-r^2/2) \\
\delta v &= (U_\infty \beta)(x - X_c)/R \exp(-r^2/2) \\
u_0 &= U_\infty + \delta u, v_0 = \delta v
\end{align*}
\]

- 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.
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 $T = 2.5$
Viscous flow over square cylinder

- $Re = 150$, based on cylinder width
- $M = 0.3$
- $6^{th}$-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.

With Interpolation

Direct Method
Multiple Forward FTLE Fields

$t = 0$

$t = 1$

$t = 2$
Interpolation Conditioning

Forward FTLE (T = 10)

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

Parameter $\Delta \alpha$
$$\Delta \alpha \equiv \max |\alpha_{ij} - \alpha'_{ij}|$$
2D Airfoil DNS

- Problem Parameters
  - $Re = 20,000$
  - $Pr = 0.72$
  - $CFL = 0.8$
  - $AOA = 4^\circ$

- 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]
Entropy: \( P = 12 \),

FTLE: \( P = 24 \),

\( T = 0.37 \)
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)

\[ u(x, y, z) = A \sin(z) + C \cos(y) \]
\[ v(x, y, z) = B \sin(x) + A \cos(z) \]
\[ w(x, y, z) = C \sin(y) + B \cos(x) \]

\[ A = \sqrt{3}, B = \sqrt{2}, C = 1 \]
3D Airfoil

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

Vorticity Magnitude

FTLE ($T = 0.15$)
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