An Eulerian Approach for Computing the Finite Time Lyapunov Exponent (FTLE)

Shingyu Leung

Department of Mathematics,
Hong Kong University of Science and Technology

masyleung@ust.hk

May 22, 2011

Outline

Introduction to FTLE

Eulerian Coherent Structure (ECS)

To segment the domain into different regions with similar behavior according to an Eulerian quantity such as the strain, kinetic energy, or the vorticity.

Lagrangian Coherent Structure (LCS)

To partition the space-time domain into different regions according to a Lagrangian quantity advected along with passive tracers.

One such Lagrangian quantity is the FTLE,

- rate of separation between adjacent particles over a finite time interval with an infinitesimal perturbation in the initial location.
- $$\mathbf{x}'(t;\mathbf{x}_0,t_0)=\mathbf{u}(\mathbf{x}(t;\mathbf{x}_0,t_0),t)$$
  with the initial condition $\mathbf{x}(t_0)=\mathbf{x}_0$ and $\mathbf{u}:\mathbb{R}^d\times\mathbb{R}\to\mathbb{R}^d$.
- Flow map $\Phi:\Omega\times\mathbb{R}\times\mathbb{R}\to\mathbb{R}^d$,
  $$\Phi(\mathbf{x};t_0,T)=\mathbf{x}(T;\mathbf{x}_0,t_0)$$
- The leading order in the perturbation
  $$\|\delta\mathbf{x}(T)\|=\sqrt{\langle\delta\mathbf{x}(0),[\mathcal{D}\Phi(\mathbf{x};t_0,T)]^*\mathcal{D}\Phi(\mathbf{x};t_0,T)\delta\mathbf{x}(0)\rangle}$$

- Denoting $\Delta(\mathbf{x}; t_0, T)$ the Cauchy-Green deformation tensor

$$\Delta(\mathbf{x}; t_0, T) = [\mathcal{D}\Phi(\mathbf{x}; t_0, T)]^* \mathcal{D}\Phi(\mathbf{x}; t_0, T)\,,$$

  the largest strength deformation

$$\max_{\delta\mathbf{x}(0)} \|\delta\mathbf{x}(T)\| = \sqrt{\lambda_{\max}[\Delta(\mathbf{x}; t_0, T)]}\|\mathbf{e}(0)\| = \exp[\sigma^T(\mathbf{x}, t_0)|T|]\|\mathbf{e}(0)\|$$

- FTLE

$$\boxed{\sigma^T(\mathbf{x}, t_0) = \frac{1}{|T|} \ln \sqrt{\lambda_{\max}[\Delta(\mathbf{x}; t_0, T)]}}$$

- *forward* FTLE if $T > 0$ and the *backward* FTLE if $T < 0$.
- LCS: (roughly speaking) Ridge of the FTLE.

Lagrangian formulation to FTLE

Lagrangian approach to FTLE

1. Compute the flow map $\Phi(\mathbf{x}; t_0, T)$ by solving the ODE system
$$\mathbf{x}'(t; \mathbf{x}_0, t_0) = \mathbf{u}(\mathbf{x}(t; \mathbf{x}_0, t_0), t)$$
on an initial Cartesian mesh at time $t = t_0$.
2. Construct the deformation tensor $\Delta(\mathbf{x}; t_0, T)$ by finite differencing the flow map $\Phi(\mathbf{x}_i; t_0, T)$ and then find it's largest eigenvalue.

Remarks

1. The velocity field might be defined discretely on a mesh.
2. The numerical accuracy when using a high order adaptive time integrator such as ode45 in MATLAB.
3. For computing the FTLE at the next time step $t = t_1 = t_0 + \Delta t$, all rays obtained on the previous time step $t = t_0$ are all discarded.

Part 1: Eulerian formulation to FTLE for one time level

Goal:

Lagrangian computations $\Rightarrow$ Eulerian formulation

Define $\Psi = (\Psi^1, \Psi^2, \cdots, \Psi^d) : \Omega \times \mathbb{R} \to \mathbb{R}^d$ with the initial condition

$$\Psi(\mathbf{x}, t_0) = \mathbf{x} = (x^1, x^2, \cdots, x^d)\,.$$

Following the particle trajectory with $\mathbf{x} = \mathbf{x}_0$, any particle identity should be preserved in the Lagrangian framework, i.e.

$$\boxed{\frac{D\Psi(\mathbf{x}, t)}{Dt} = \frac{\partial \Psi(\mathbf{x}, t)}{\partial t} + (\mathbf{u} \cdot \nabla)\Psi(\mathbf{x}, t) = \mathbf{0}}$$

These level set functions defined on a uniform Cartesian mesh give the *backward* flow map from $t = t_0 + T$ to $t = t_0$, i.e.

$$\boxed{\Phi(\mathbf{y}; t_0 + T, -T) = \Psi(\mathbf{y}, t_0 + T)}$$

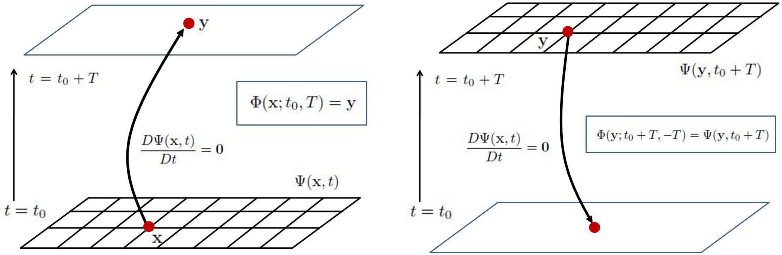


Figure: Lagrangian and Eulerian interpretations of the function Ψ.

Computing the *backward* FTLE $\sigma^{-T}(\mathbf{x}, t_0 + T)$

1. Discretize the computational domain
2. Initialize $\Psi^1(x_i, y_j, t_0) = x_i$ and $\Psi^2(x_i, y_j, t_0) = y_j$
3. Solve the Liouville equations
$$\frac{\partial \Psi^m}{\partial t} + (\mathbf{u} \cdot \nabla)\Psi^m = 0\,,\ m = 1, 2$$
for up to $t = t_K = t_0 + T$.
4. Compute the Cauchy-Green deformation tensor with $\Psi = (\Psi^1, \Psi^2)^T$,
$$\Delta(x_i, y_j; t_K, -T) = [\mathcal{D}\Psi(x_i, y_j, t_K)]^* \mathcal{D}\Psi(x_i, y_j, t_K)$$
5. Determine the *backward* FTLE at $(x_i, y_j, t_0 + T)$
$$\sigma^{-T}(x_i, y_j, t_K) = \frac{1}{T} \ln \sqrt{\lambda_{\max}[\Delta(x_i, y_j; t_K, -T)]}$$

Computing the *forward* FTLE $\sigma^T(\mathbf{x}, t_0)$

1. Initialize the level set functions at $t = t_0 + T$ by $\Psi(\mathbf{x}, t_0 + T) = \mathbf{x}$.
2. Solve the corresponding level set equations *backward* in time.
3. Determining the Jacobian of the resulting flow map and then computing the largest eigenvalue of the deformation tensor $\Delta(\mathbf{x}; t_0, T)$.
4. The *forward* FTLE is formed by

$$\sigma^T(\mathbf{x}, t_0) = \frac{1}{T} \ln \sqrt{\lambda_{\max}[\Delta(\mathbf{x}; t_0, T)]} \, .$$

Part 2: Propagating the FTLE for many time-levels

Goal:

Propagate the *backward* FTLE $\sigma^{-T}(\mathbf{x}, t_0 + T)$ *forward* in time from $t_0 + T$ to T_f in order to approximate $\sigma^{-T}(\mathbf{x}, t)$ or $\sigma^{-(t-t_0)}(\mathbf{x}, t)$ for $t > t_0 + T$.

Theorem (Theorem 3.1 in *Shadden, Lekien and Marsden 2005*)

The traditional (forward) Lyapunov exponent is constant along trajectories. Also, the (forward) finite-time Lyapunov exponent becomes constant along trajectories for large integration times T.

The difference between $\sigma^T(\mathbf{x}, t_0)$ and $\sigma^T(\Phi^s(\mathbf{x}; t_0), t_0 + s)$ for some arbitrary but fixed s,

$$|\sigma^T(\mathbf{x}, t_0) - \sigma^T(\Phi^s(\mathbf{x}; t_0), t_0 + s)| \leq \frac{2|s|}{|T|} \max_{t^*} \sigma^{t^* - t_0}(\mathbf{x}, t_0) = O(|T|^{-1})$$

The material derivative of the (*forward*) FTLE is bounded by

$$\left| \frac{D\sigma^T(\mathbf{x}, t)}{Dt} \right| = \lim_{s \to 0} \frac{|\sigma^T(\Phi^s(\mathbf{x}; t), t + s) - \sigma^T(\mathbf{x}, t)|}{|s|} = O(|T|^{-1})$$

If T is large enough, we obtain the approximation

$$\boxed{\frac{D\sigma^T(\mathbf{x}, t)}{Dt} = 0}$$

If T is large enough, we can approximate the solution $\sigma^{-T}(\mathbf{x}, t)$ for $t > t_0 + T$ by simply solving the following Liouville equation

$$\boxed{\frac{\partial \sigma^{-T}(\mathbf{x}, t)}{\partial t} + \mathbf{u} \cdot \nabla \sigma^{-T}(\mathbf{x}, t) = 0}$$

with the initial condition at $t = t_0 + T$.
Let S be the time difference from $t = t_0 + T$. Since

$$\begin{aligned} & \left| \sigma^{-T}(\Phi^S(\mathbf{x}; t_0 + T), t_0 + T + S) - \sigma^{-T}(\mathbf{x}, t_0 + T) \right| \\ \leq \quad & \frac{2S}{T} \max_{t^*} \sigma^{t^*}(\mathbf{x}, T) = O\left(\frac{S}{T}\right), \end{aligned}$$

the error in the approximation is linear in S.

Advantages and limitations of the Eulerian formulation

- Timestep constraint
  1. Lagrangian: Δt, stiffness of the velocity field
  2. Eulerian: $\Delta t = O(\Delta x)$, CFL condition
- Computational Complexity
  - d: the dimension of the domain
  - $N = O(\Delta x^{-1})$: number of mesh points in each spacial dimension
  - $M = T/\Delta t = O(\Delta t^{-1})$: number of time steps for $t = T$
  - $K = (T_f - t_0)/\Delta\tilde{t} = O(\Delta\tilde{t}^{-1})$: number of time steps until $t = T_f$ (number of time levels of FTLE to be computed)

  1. Lagrangian: $O(N^d MK) = O(\Delta x^{-d}\Delta\tilde{t}^{-1}\Delta t^{-1})$
  2. Eulerian: $O(N^d[(d-1)M + K + 1]) = O(\Delta x^{-d}\Delta\tilde{t}^{-1})$

- Velocity Field
  1. CFD solver
  2. Real life measurements
- Storage for the velocity field
  1. Lagrangian approach: $O(dN^{d+1})$, N^{d+1} mesh points in the space-time domain and a d-vector per node.
  2. Eulerian approach: $O(dN^{d})$, N^{d} mesh points in the space domain and a d-vector per node.
- Storage for the flow map and the FTLE
  1. Lagrangian approach: $O(d)$
  2. Eulerian approach: $O(dN^{d})$
- Numerical dissipation in the Eulerian formulation
- An order of $O(S/T)$ error is introduced in the approximation

Part 3: FTLE on a Codimension One Manifold

Goal:

A simple approach to compute the FTLE on a codimension one manifold.

- A codimension one manifold $\mathcal{M}(t)$ in $\mathbf{R}^d$ evolving in time under the velocity field $\tilde{\mathbf{u}} : \mathcal{M}(t) \times t \to \mathbb{R}^d$.
- A possible approach: triangulate the surface, evolve all vertices according to the flow, compute the Jacobian matrix based on a local coordinate system on the surface,...
- Our proposed approach: the evolving manifold represented implicitly using a level set function.

Computing the *backward* FTLE $\sigma^{-T}(\mathbf{x}, t_0+T)$ for $\mathbf{x} \in \mathcal{M}(t_0+T)$

1. With $\phi(x_i, y_j, t_0)$ at $t = t_0$ so that $\mathcal{M} = \{\phi = 0\}$, solve

$$\boxed{\frac{\partial \phi}{\partial t} + (\mathbf{u} \cdot \nabla)\phi = 0} \text{ for up to } t = t_0 + T$$

2. With $\Psi^1(x_i, y_j, t_0) = x_i$, $\Psi^2(x_i, y_j, t_0) = y_j$, solve the Liouville equations

$$\boxed{\frac{\partial \Psi^m}{\partial t} + (\mathbf{u} \cdot \nabla)\Psi^m = 0\,,\ m = 1, 2} \text{ from } t = t_0 \text{ to } t = t_0 + T$$

3. Extend the map off the manifold by

$$\boxed{\frac{\partial \Psi^m}{\partial \tau} + \operatorname{sgn}(\phi)(\mathbf{n} \cdot \nabla)\Psi^m = 0\,,\ m = 1, 2}$$

4. Compute the deformation tensor $\boxed{\Delta = (\mathcal{D}\Psi)^* \mathcal{D}\Psi}$

5. Determine the *backward* FTLE $\tilde{\sigma}^T = \ln \lambda_{\max}(\Delta)$

Example: Double-Gyre flow

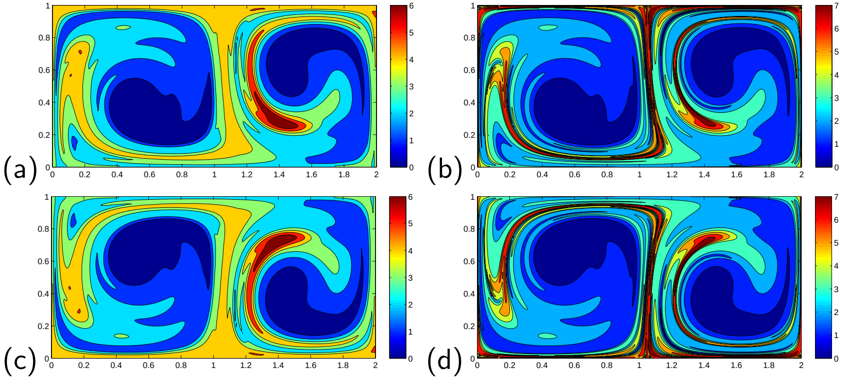
The flow is modeled by the following stream-function

$$\psi(x, y, t) = A \sin[\pi f(x, t)] \sin(\pi y),$$

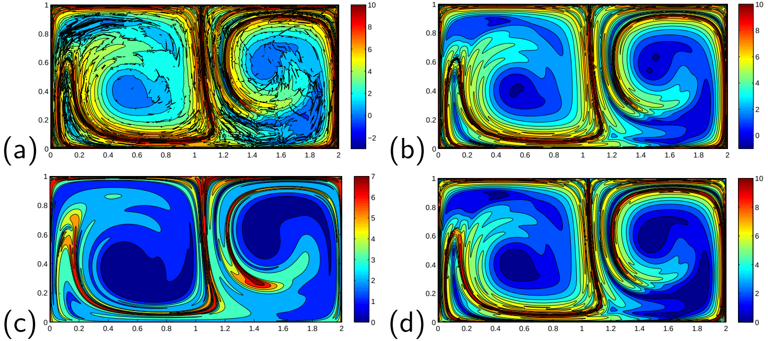
where

$$\begin{aligned} f(x,t) &= a(t)x^2 + b(t)x, \\ a(t) &= \epsilon \sin(\omega t), \\ b(t) &= 1 - 2\epsilon \sin(\omega t) \end{aligned}$$

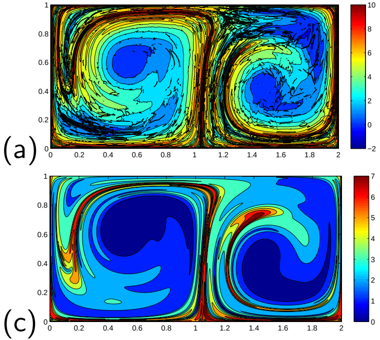
with $A = 0.1$, $\omega = 2\pi/10$ and $\epsilon = 0.1$



(a) The *backward* FTLE using $\Delta x = \Delta y = 1/128$ at $t = 15$, $T = 5$ and $S = 10$.
(b) The *backward* FTLE using $\Delta x = \Delta y = 1/512$ at $t = 15$, $T = 5$ and $S = 10$.
(c) The *forward* FTLE using $\Delta x = \Delta y = 1/128$ at $t = 0$, $T = 5$ and $S = 10$.
(d) The *forward* FTLE using $\Delta x = \Delta y = 1/512$ at $t = 0$, $T = 5$ and $S = 10$.



(a) The *backward* FTLE with $\Delta x = \Delta y = 1/512$ at $t = 15.0$ using the Lagrangian approach (MATLAB function ode45) with $T = 15$.
(b) The *backward* FTLE with $\Delta x = \Delta y = 1/512$ at $t = 15.0$ using the Lagrangian approach (RK4 with a fixed timestep) with $T = 15$.
(c) The *backward* FTLE at $t = 15$ using the proposed Eulerian approach with $\Delta x = \Delta y = 1/512$ and $T = 5$ and $S = 10$.
(d) The *backward* FTLE at $t = 15$ using the proposed Eulerian approach with $\Delta x = \Delta y = 1/512$ and $T = 15$.



(a) the *forward* FTLE with $\Delta x = \Delta y = 1/512$ at $t = 0.0$ using the Lagrangian approach (MATLAB function ode45) with $T = 15$.
(b) The *forward* FTLE with $\Delta x = \Delta y = 1/512$ at $t = 0.0$ using the Lagrangian approach (RK4 with a fixed timestep) with $T = 15$.
(d) The *forward* FTLE at $t = 0$ using the proposed Eulerian approach with $\Delta x = \Delta y = 1/512$, $T = 5$ and $S = 10$.
(c) The *forward* FTLE at $t = 0$ using the proposed Eulerian approach with $\Delta x = \Delta y = 1/512$ and $T = 15$.

Δx	Eul.	Rate	Lag. ($\Delta t = 1$)	Rate	Lag. ($\Delta t = \Delta x$)	Rate
$1/64$	22.87	-	1085	-	4591	-
$1/128$	110.9	2.28	8424	2.95	71440	3.96
$1/256$	1021	3.20	66030	2.97	1123000	3.97
$1/512$	8859	3.11	525700	2.99	-	-

CPU times (in seconds) for the Eulerian approach with $T = 5$ and $S = 10$, and the Lagrangian approach with $\Delta t = 1$ and $\Delta t = \Delta x$.

Eulerian approach $\simeq O(\Delta x^{-3})$

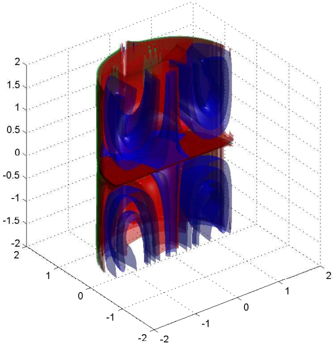
Lagrangian approach $\simeq O(\Delta x^{-3}\Delta t^{-1})$

Example: Rayleigh-Bénard convection

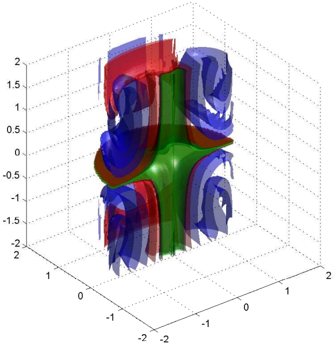
The velocity field is given by

$$\begin{aligned} u &= \frac{A}{k}\xi \sin kr \cos z\,, \\ v &= \frac{A}{k}y \sin kr \cos z\,, \\ w &= -A \sin z \left(r \cos kr + \frac{2}{k} \sin kr \right)\,, \end{aligned}$$

where $A = 0.24$, $k = 0.2$, $r^2 = \xi^2 + y^2$, $\xi = x - g(t)$, and $g(t) = 0.1 \cos(2\pi t)$.



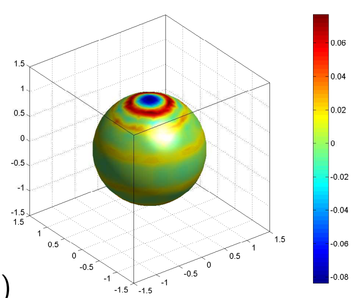
(a)



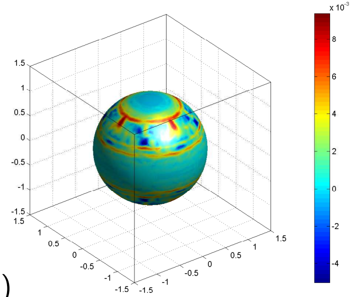
(b)

(a) The *backward* FTLE using $\Delta x = \Delta y = 1/32$ at $t = 50$, $T = 12$ and $S = 38$.
(b) The *forward* FTLE using $\Delta x = \Delta y = 1/32$ at $t = 0$, $T = 12$ and $S = 38$.

Example: FTLE on a Sphere

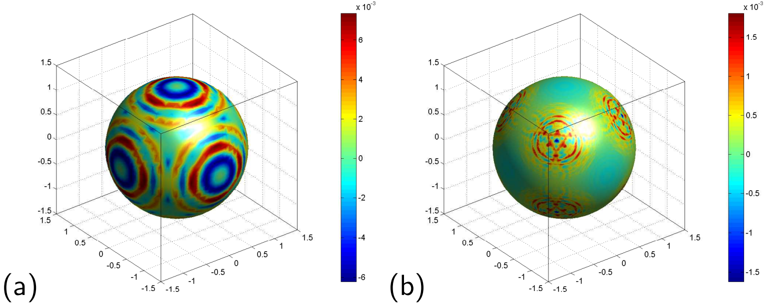


(a)

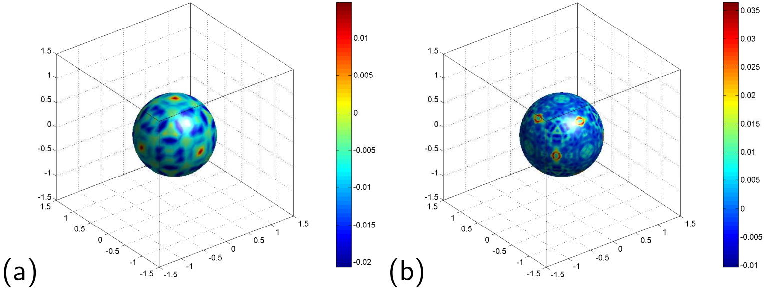


(b)

The *forward* FTLE on a sphere under the rigid body rotation. The exact solution is zero everywhere on the sphere. The underlying mesh size (a) $\Delta x = 3/32$ and (b) $\Delta x = 3/128$.



The error in the *forward* FTLE on a sphere under the motion in the outward normal direction. The underlying mesh size (a) $\Delta x = 3/32$ and (b) $\Delta x = 3/128$.



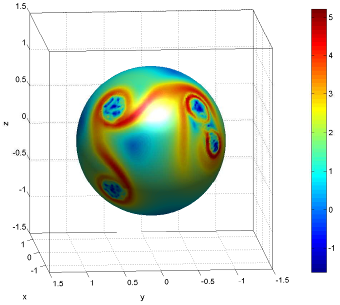
The error in the *forward* FTLE on a sphere under the motion in the inward normal direction. The underlying mesh size (a) $\Delta x = 3/32$ and (b) $\Delta x = 3/128$.

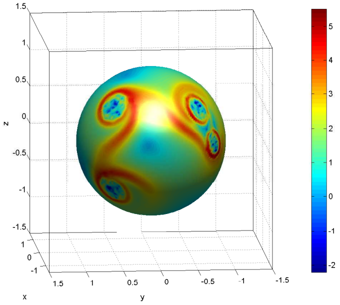
Example: Four point vortices on a sphere

The velocity of any marker particle satisfies the motion

$$\mathbf{x}' = \frac{1}{2\pi} \sum_{i=1}^{4} \frac{\mathbf{x}_i \times \mathbf{x}}{2(1 - \mathbf{x} \cdot \mathbf{x}_i)} ,$$

with the point vortices centered at $(1/\sqrt{3}, -1/\sqrt{3}, 1/\sqrt{3})$, $(1/\sqrt{3}, -1/\sqrt{3}, -1/\sqrt{3})$, $(-\sqrt{2/3}, 1/\sqrt{3}, 0)$ and $(-1/\sqrt{3}, -1/\sqrt{3}, 1/\sqrt{3})$.

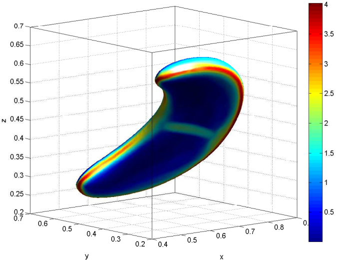
(a) 

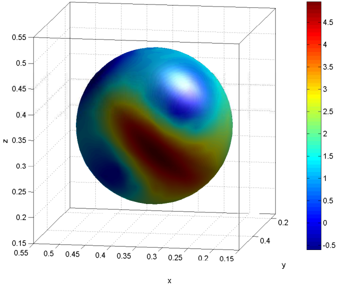
(b) 

(a) The *backward* FTLE at $t = 2$ with $T = 2$. (b) The *forward* FTLE at $t = 0.0$ with $T = 2$.

The sphere evolves under the vortex motion governed by

$$\begin{aligned} u &= 2\sin^2(\pi x)\sin(2\pi y)\sin(2\pi z) \\ v &= -\sin^2(\pi y)\sin(2\pi x)\sin(2\pi z) \\ w &= -\sin^2(\pi z)\sin(2\pi y)\sin(2\pi x)\,. \end{aligned}$$

(a)


(b)


(a) The *backward* FTLE at $t = 0.5$ with $T = 0.5$. (b) The *forward* FTLE at $t = 0.0$ with $T = 0.5$.

Conclusion

1. Eulerian framework for approximating the FTLE based on the level set formulation
   - Compute the related flow map by solving Liouville equations
   - Determine the FTLE on an evolving manifold implicitly represented by a level set function
2. Future work
   - Extract the Lagrangian coherent structure from the FTLE
   - Combine the current approaches with a CFD solver
   - Incorporate our algorithms with flow fields from real life measurements

References

Shingyu Leung (2011)
An Eulerian Approach for Computing the Finite Time Lyapunov Exponent.
Journal of Computational Physics, Volume 230, Issue 29, May 1 2011, Pages 3500-3524.

Thank you