Introduction
Methods
Implementation
Validation
Application
Conclusions
Timeline
References

AMSC 664 Final Presentation: Lagrangian Analysis of 2D and 3D Ocean Flows from Eulerian Velocity Data

David Russell
Second-year Ph.D. student, Applied Math and Scientific Computing

Project Advisor: Kayo Ide
Department of Atmospheric and Oceanic Science
Center for Scientific Computation and Mathematical Modeling
Earth System Science Interdisciplinary Center
Institute for Physical Science and Technology

May 11th, 2016

Project Goals

Project goals:

- Develop tools for Lagrangian (particle-based) analysis of a 2D or 3D ocean flow from Eulerian (grid-based) velocity data
- Validate on a series of simple ODE systems with well-understood dynamics
- Validate against existing tools from ROMS (Regional Ocean Modeling System)
- Test on a dataset from the Chesapeake Bay

Introduction

Dynamics background:

- Given an ODE system $\dot{\mathbf{x}} = \mathbf{u}(t, \mathbf{x})$ ($\mathbf{u} = (u, v, w)$ in $\mathbb{R}^3$) with corresponding particle trajectories $\mathbf{X}(t, \mathbf{X}_0)$
- A *hyperbolic equilibrium point* is a point $\mathbf{x}^*$ for which $\mathbf{u}(t, \mathbf{x}^*) \equiv 0$ and all eigenvalues of the Jacobian $\frac{\partial \mathbf{u}}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}^*}$ have nonzero real part
  - *Stable manifold* of $\mathbf{x}^*$: $\{\mathbf{x} \in \mathbb{R}^n \mid \lim_{t\to\infty} \mathbf{X}(t, \mathbf{x}) = \mathbf{x}^*\}$
  - *Unstable manifold* of $\mathbf{x}^*$: $\{\mathbf{x} \in \mathbb{R}^n \mid \lim_{t\to-\infty} \mathbf{X}(t, \mathbf{x}) = \mathbf{x}^*\}$
- Corresponding concept when $\mathbf{x}^*$ is moving is that of a *distinguished hyperbolic trajectory*

Introduction

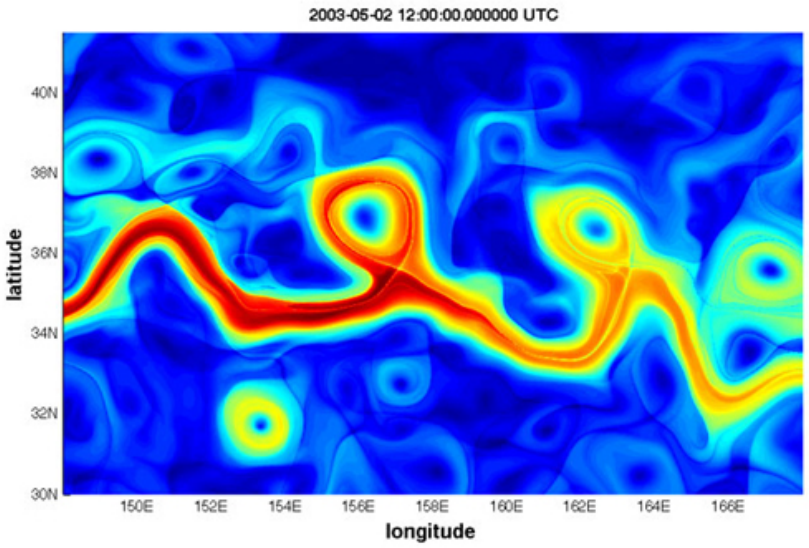
Stable and unstable manifolds:

- Act as boundaries between coherent structures in the flow (e.g. as rough barriers to pollutant transport in oceans)
- Can be found by one of two numerical methods: finite-time Lyapunov exponents (traditional) and arc lengths, also called M-function (newer)
- Each of these requires a thick soup of particle trajectories for calculation

Introduction

Example: M-function applied to dataset from the Kuroshio current, northwest Pacific Ocean [**?**]:

- Red fast, blue slow (on average)
- Thin yellow lines: stable and unstable manifolds



Methods: Overview

- Three main numerical tasks:
  - *Interpolate* velocities to off-grid particle locations
  - *Integrate* particle velocities to obtain trajectories
  - *Analyze* trajectories using numerical tools
- Implement and compare various versions of each of these

Methods: Interpolation

2D horizontal interpolation methods:

- Bilinear splines: Fit function of form

$$f(x,y) = \sum_{i,j=0}^{1} c_{ij} x^i y^j = c_{00} + c_{10}x + c_{01}y + c_{11}xy$$

  to given values of u at box corners

- Bicubic splines: Fit function of form

$$f(x,y) = \sum_{i,j=0}^{3} c_{ij} x^i y^j$$

  to given values of u, approximate values of u_x, u_y, u_{xy} at box corners (using second-order finite differences: two-sided for interior, one-sided for boundary)

Methods: Interpolation

Vertical and temporal interpolation methods:

- Linear: Fit linear function of z through two nearest vertical neighbors
- Cubic: Fit cubic polynomial in z through four nearest vertical neighbors
- Same algorithms for temporal interpolation

Intergration method:

- 4th-order Runge-Kutta

Methods: Analysis

Approaches:

- M-function (arc length over fixed time interval)
  - Forward: $M(\tau, \mathbf{X}_0) = \int_0^\tau |\mathbf{u}(t, \mathbf{X}(t, \mathbf{X}_0))| \, dt$
  - Backward: $M(\tau, \mathbf{X}_0) = \int_{-\tau}^0 |\mathbf{u}(t, \mathbf{X}(t, \mathbf{X}_0))| \, dt$
  - Both: $M(\tau, \mathbf{X}_0) = \int_{-\tau}^\tau |\mathbf{u}(t, \mathbf{X}(t, \mathbf{X}_0))| \, dt$
- Maximal finite-time Lyapunov exponent (FTLE) (maximal growth rate of distance between close trajectories):

$$\mathrm{FTLE}(\tau, \mathbf{X}_0) = \frac{1}{\tau} \ln \left(\sigma_{\max} \left(L(\tau, \mathbf{X}_0) \right) \right)$$

where $L(\tau, \mathbf{X}_0) = \frac{\partial \mathbf{X}(\tau, \mathbf{X}_0)}{\partial \mathbf{X}_0}$ is the transition matrix and $\sigma_{\max}$ is the largest singular value

Methods: Analysis

Calculating M and FTLE:

- M-function: Consider M as a state variable like u and v, integrate alongside u and v
- FTLE:
  - Track particles directly next to $\mathbf{X}_0$ up to time τ
  - Approximate $L = \frac{\partial \mathbf{X}}{\partial \mathbf{X}_0}$ using centered differences for $\Delta X_0, \Delta Y_0, \Delta Z_0$ and final locations for $\Delta X, \Delta Y, \Delta Z$
  - Calculate $\sigma_{\max}(L)$ as $\sqrt{\lambda_{\max}(L^T L)}$, where eigenvalues of $L^T L \in (\mathbb{R}^{2\times 2}$ or $\mathbb{R}^{3\times 3})$ are roots a quadratic or cubic polynomial

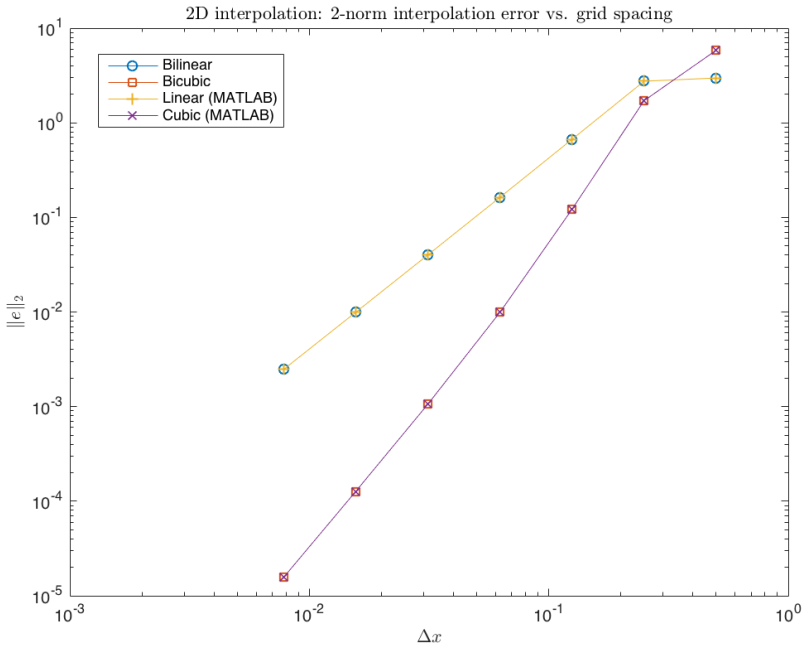
Implementation

Implementation details:

- Software: MATLAB
- Hardware:
  - Shorter runs (up to 6-hour dataset): MacBook Pro laptop, 2.6 GHz Intel Core i5, 8 GB 1600 MHz DDR3
  - Longer runs: Deepthought2
- Vectorization but no parallelization across particles
- Trajectories that leave domain set to `NaN`

Validation: 2D Interpolation

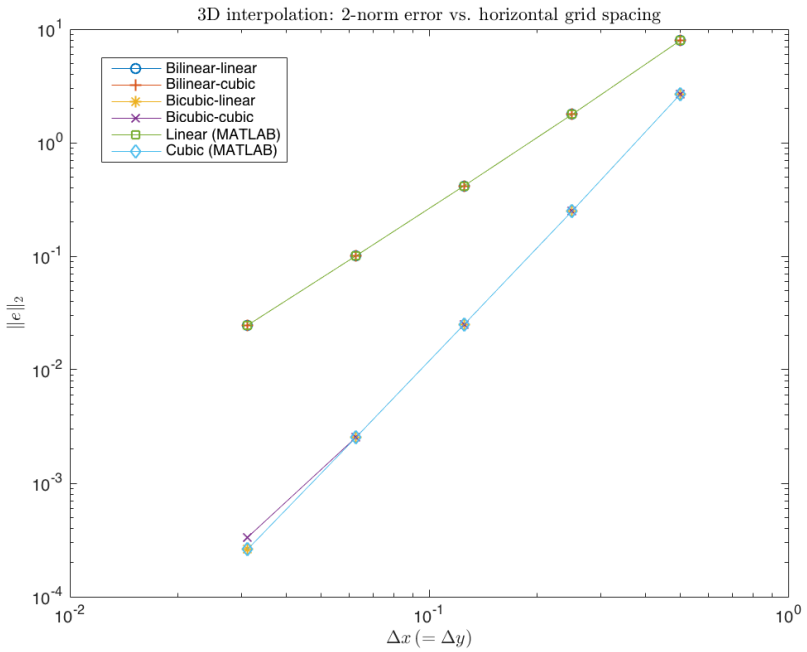
2D accuracy: $\|f_{\text{interp}} - f_{\text{semi-interp}}\|_2$ vs. Δx (interpolation mesh size) for $f(x, y) = e^x \cos(2\pi y)$ on $[-1, 1]^2$ for particle grid at 6x resolution



	Convergence rate ($\|e\|_2 \sim h^p$)
Bilinear	$p \approx 2.0064$
Bicubic	$p \approx 3.0175$

Validation: 3D Interpolation

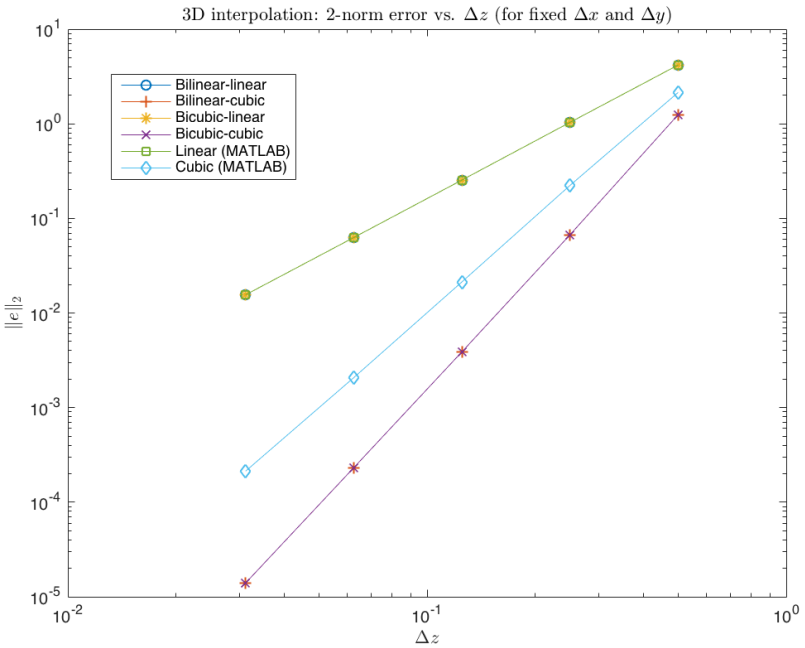
3D horizontal accuracy: $\|f_{\text{approx}} - f\|_2$ vs. $\Delta x = \Delta y$, Δz fixed, for $f(x, y, z) = e^x \cos 2\pi y \cos 2\pi z$ on $[-1, 1]^2$



Horizontal (-vertical) method	Convergence rate ($\|e\|_2 \sim h^p$)
Bilinear (-linear)	$p \approx 2.0253$
Bicubic (-linear)	$p \approx 3.2675$
Bicubic (-cubic)	$p \approx 2.9192$
MATLAB Cubic	$p \approx 3.2675$

Validation: 3D Interpolation

3D vertical accuracy: $\|f_{\text{interp}} - f_{\text{semi-interp}}\|_2$ vs. Δz, $\Delta x = \Delta y$ fixed



(Horizontal-)vertical method	Convergence rate ($\|e\|_2 \sim h^p$)
(Bilinear-)linear	$p \approx 2.0101$
(Bicubic-)cubic	$p \approx 4.0484$
MATLAB cubic	$p \approx 3.2836$

Validation: 2D Trajectory Calculation

2D test system: Undamped Duffing oscillator (1918) [**?**]:

- Simple nonlinear oscillator
- General case (forcing parameter ϵ):

$$\dot{x} = y$$
$$\dot{y} = x - x^3 + \epsilon \sin t$$

- Autonomous case ($\epsilon = 0$):
  - Hyperbolic equilibrium point at $\mathbf{x} = (0, 0)$
  - Stable and unstable manifolds both given by $y^2 = x^2 - \frac{x^4}{2}$ (figure-eight shape centered at origin)

Validation: 2D Trajectory Calculation

Autonomous Duffing oscillator:

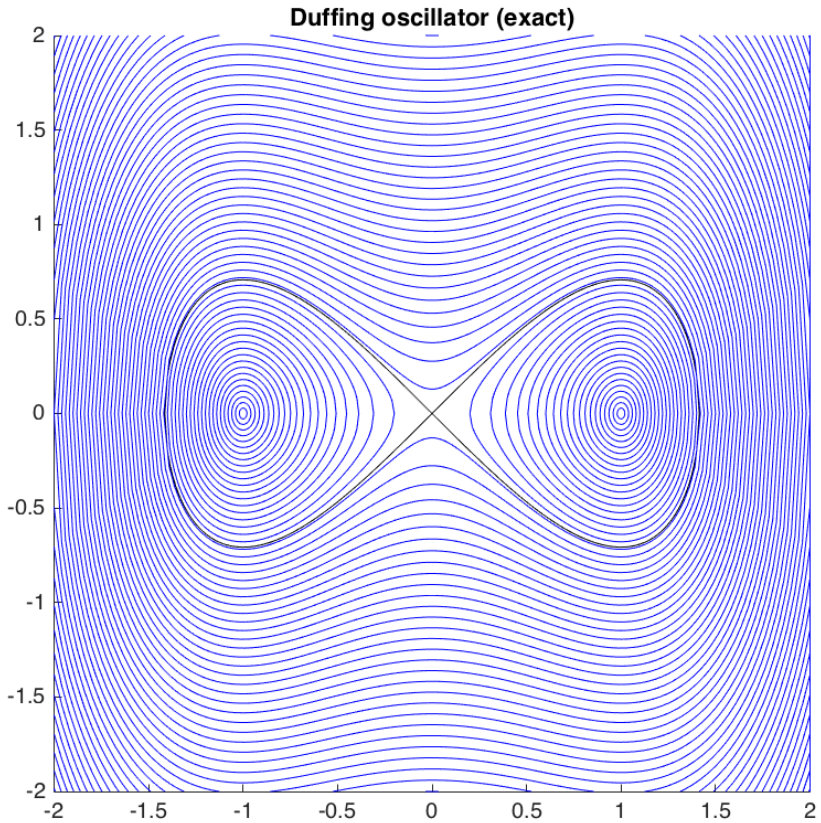
- Hamiltonian system with $H(x, y) = \frac{1}{2}y^2 - \frac{1}{2}x^2 + \frac{1}{4}x^4$ conserved along trajectories
- $x(t)$ and $y(t)$ can be computed explicitly via Jacobi elliptic functions [**?**], e.g. for $y_0 = 0$ we have

$$x(t) = x_0 \cdot \mathrm{cn}\left(t\sqrt{x_0^2 - 1}, \sqrt{\frac{x_0^2}{2(x_0^2 - 1)}}\right)$$

$$y(t) = -x_0\sqrt{x_0^2 - 1} \cdot \mathrm{sn}\left(t\sqrt{x_0^2 - 1}, \sqrt{\frac{x_0^2}{2(x_0^2 - 1)}}\right) \cdot \mathrm{dn}\left(t\sqrt{x_0^2 - 1}, \sqrt{\frac{x_0^2}{2(x_0^2 - 1)}}\right)$$

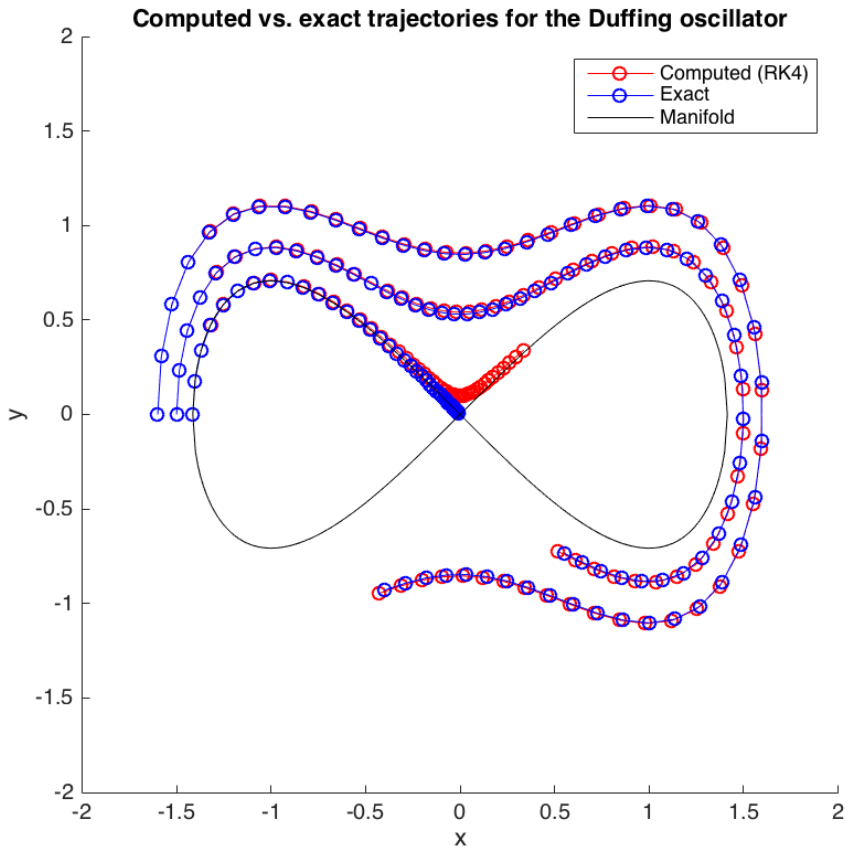
Validation: 2D Trajectory Calculation

Exact trajectories of autonomous Duffing (level sets of $H(x, y)$:)



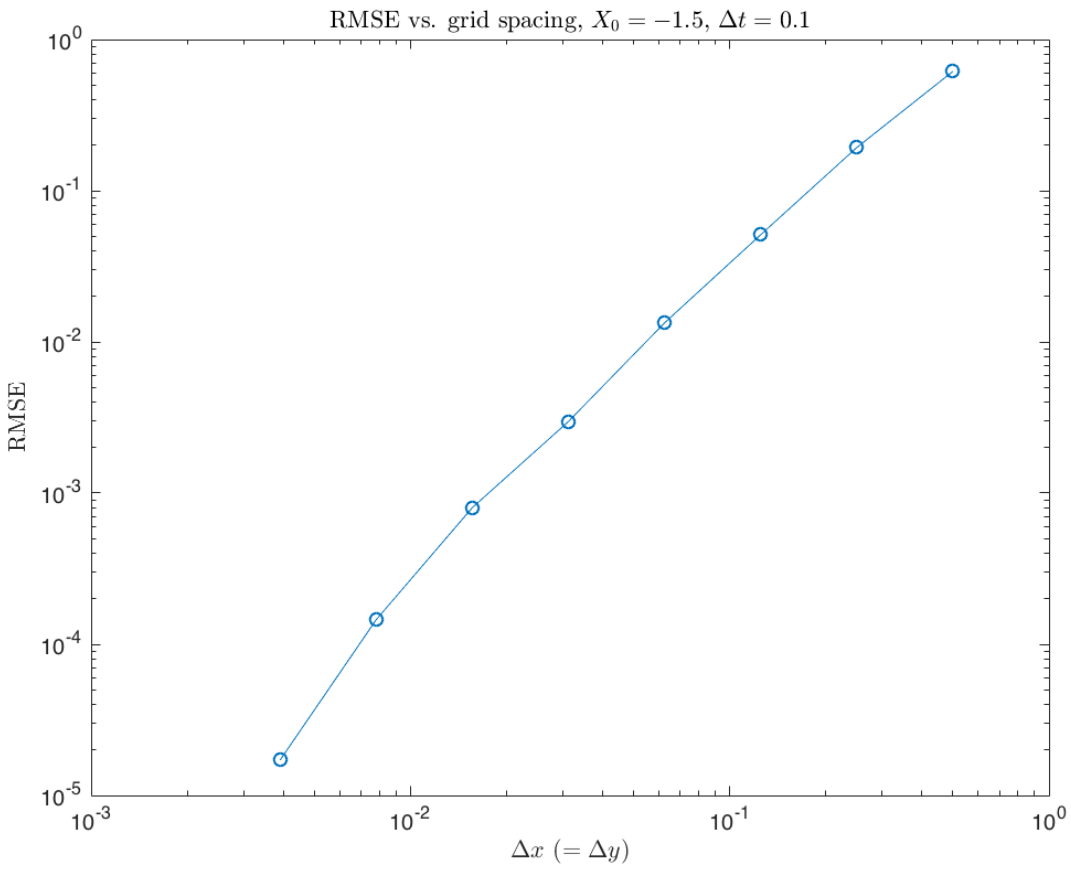
Validation: 2D Trajectory Calculation

Visual validation: A few computed vs. exact trajectories for Duffing oscillator (RK4, bilinear, $\Delta x = \Delta y = 0.1$, $\Delta t = 2^{-3}$, $t_f = 6$):



Validation: 2D Trajectory Calculation

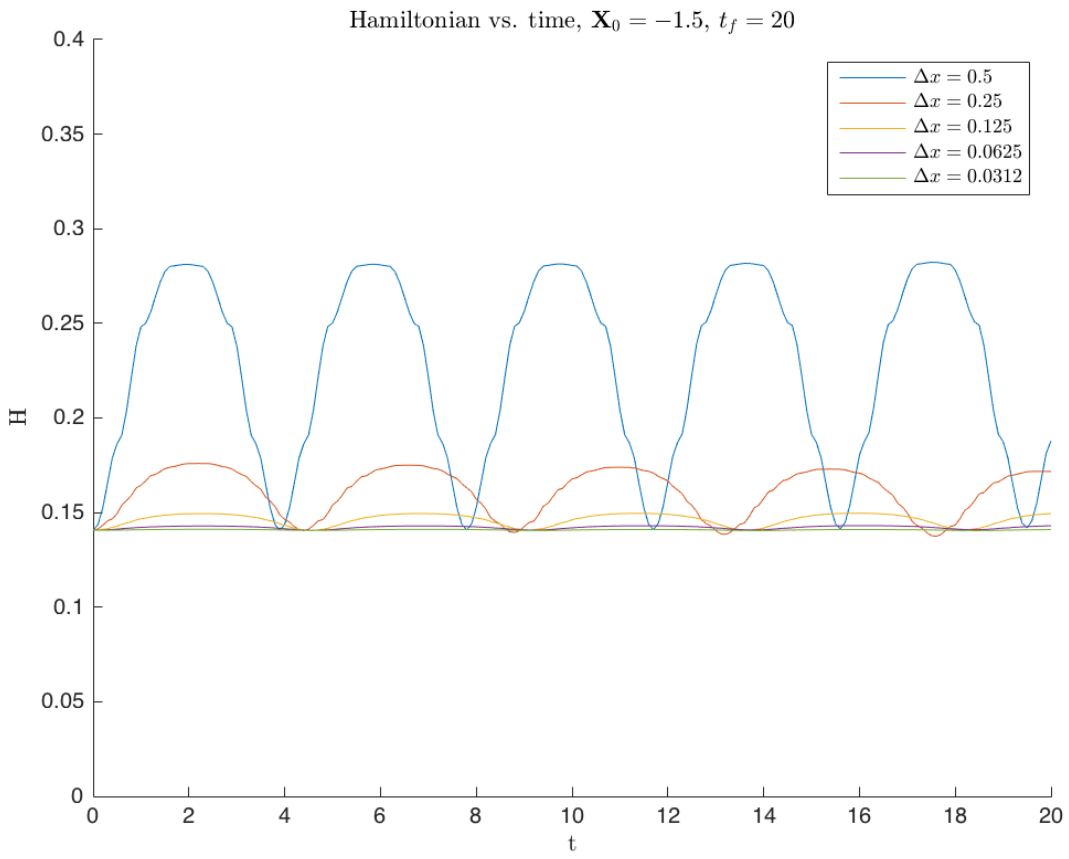
RMSE vs. interpolation Δx for individual trajectory with $\mathbf{X}_0 = (-1.5, 0)$, $\Delta t = 0.1$, $t_f = 6$:



Validation: 2D Trajectory Calculation

Hamiltonian vs. time for $\mathbf{X}_0 = (-1.5, 0)$, $t_f = 20$:

- Hamiltonian becomes constant as $\Delta x \to 0$



Visual Validation: 2D Lagrangian Tools

Autonomous Duffing oscillator: M-function, forwards and backwards, $\tau = 10$

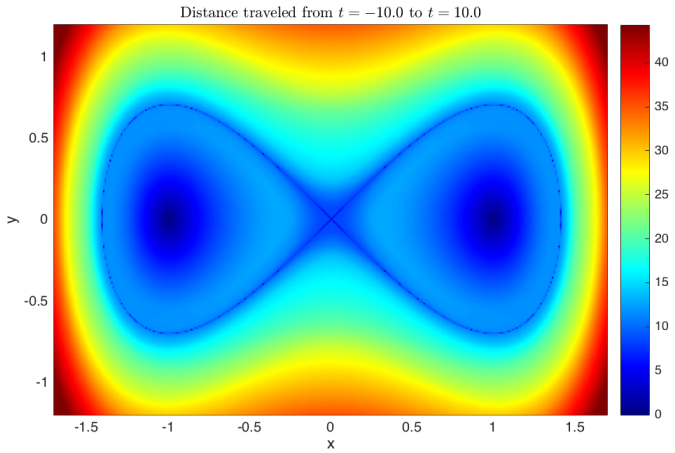


Figure: My results

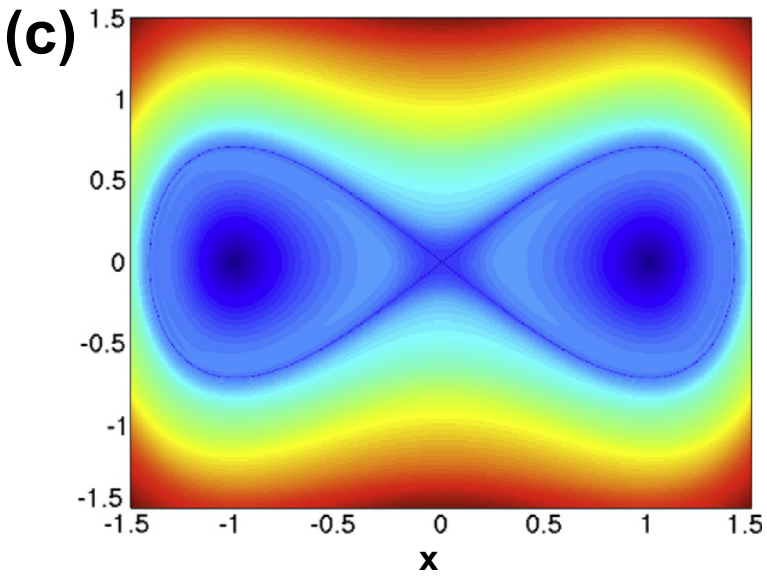


Figure: Mancho et al results [?]

Visual Validation: 2D Lagrangian Tools

Autonomous Duffing oscillator: FTLE, forwards, $\tau = 10$

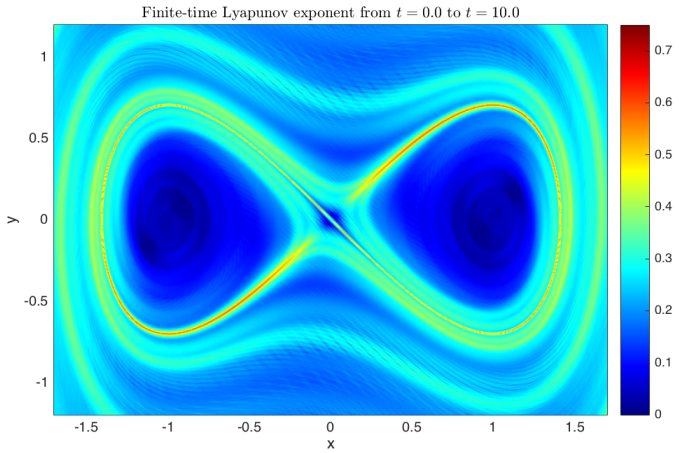


Figure: My results

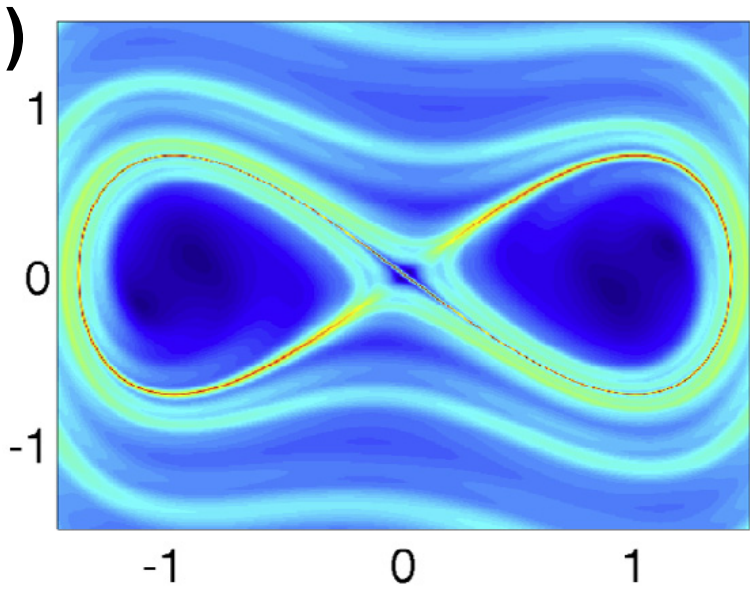


Figure: Mancho et al results [**?**]

Visual Validation: 2D + Time Lagrangian Tools

Forced Duffing oscillator ($\epsilon = 0.1$): M-function, forwards and backwards, $\tau = 10$

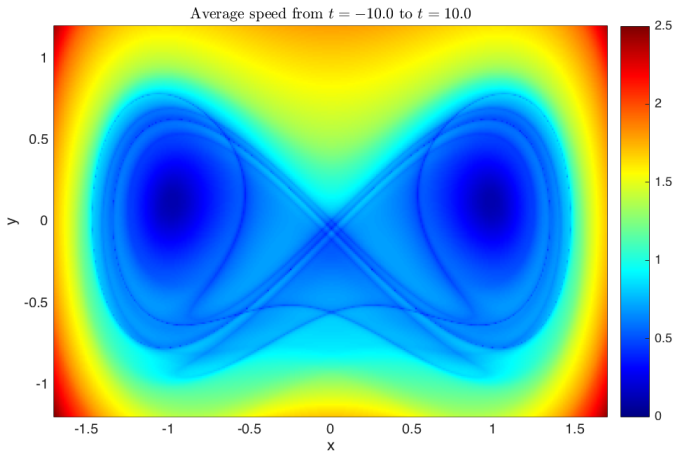


Figure: My results

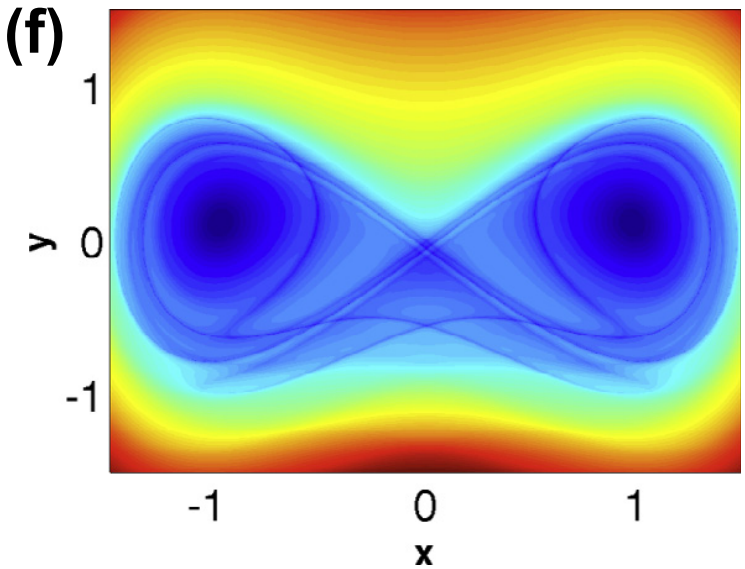


Figure: Mancho et al results [?]

Visual Validation: 2D + Time Lagrangian Tools

Forced Duffing oscillator: FTLE, forwards, $\tau = 10$

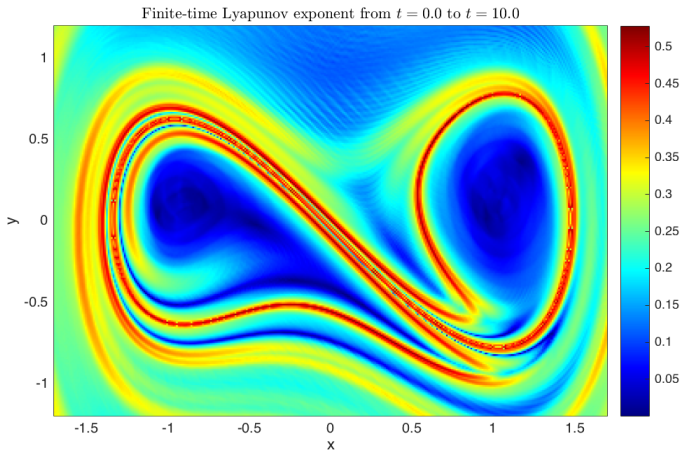


Figure: My results

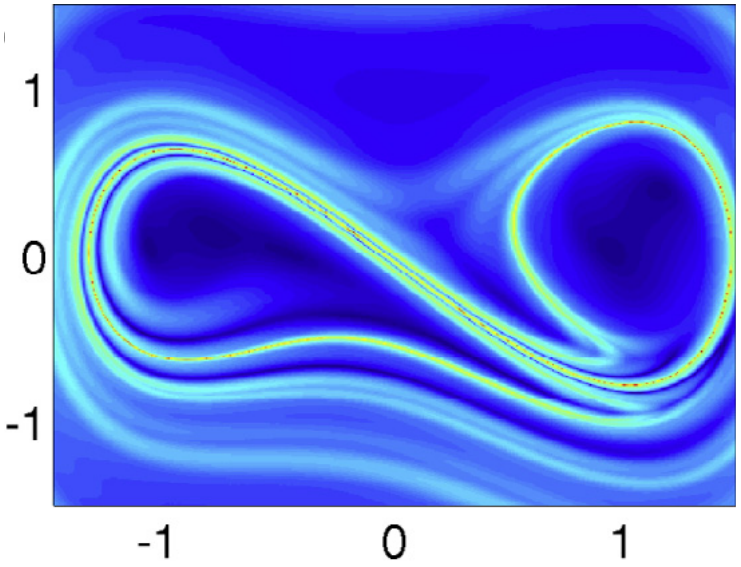


Figure: Mancho et al results [**?**]

Visual Validation: 2D + Time Lagrangian Tools

Finding manifolds: M vs. FTLE, forwards, $\tau = 10$

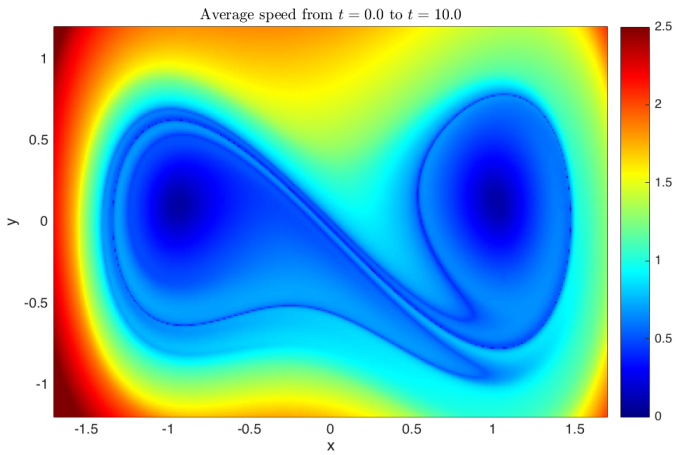


Figure: M-function

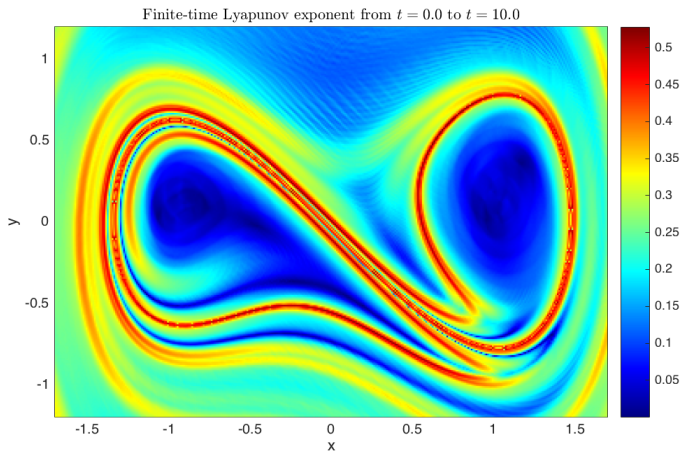


Figure: FTLE

Validation: 3D Trajectory Calculation

3D test case: Hill's spherical vortex (1894) [**?**]

- Simple model of axisymmetric flow around and inside a sphere of radius a
- Cartesian ODE system:

$$\dot{x} = -\frac{3Uxz}{2a^2}$$
$$\dot{y} = -\frac{3Uyz}{2a^2}$$
$$\dot{z} = \frac{3U(2x^2 + 2y^2 + z^2 - a^2)}{2a^2}$$

Validation: 3D Trajectory Calculation

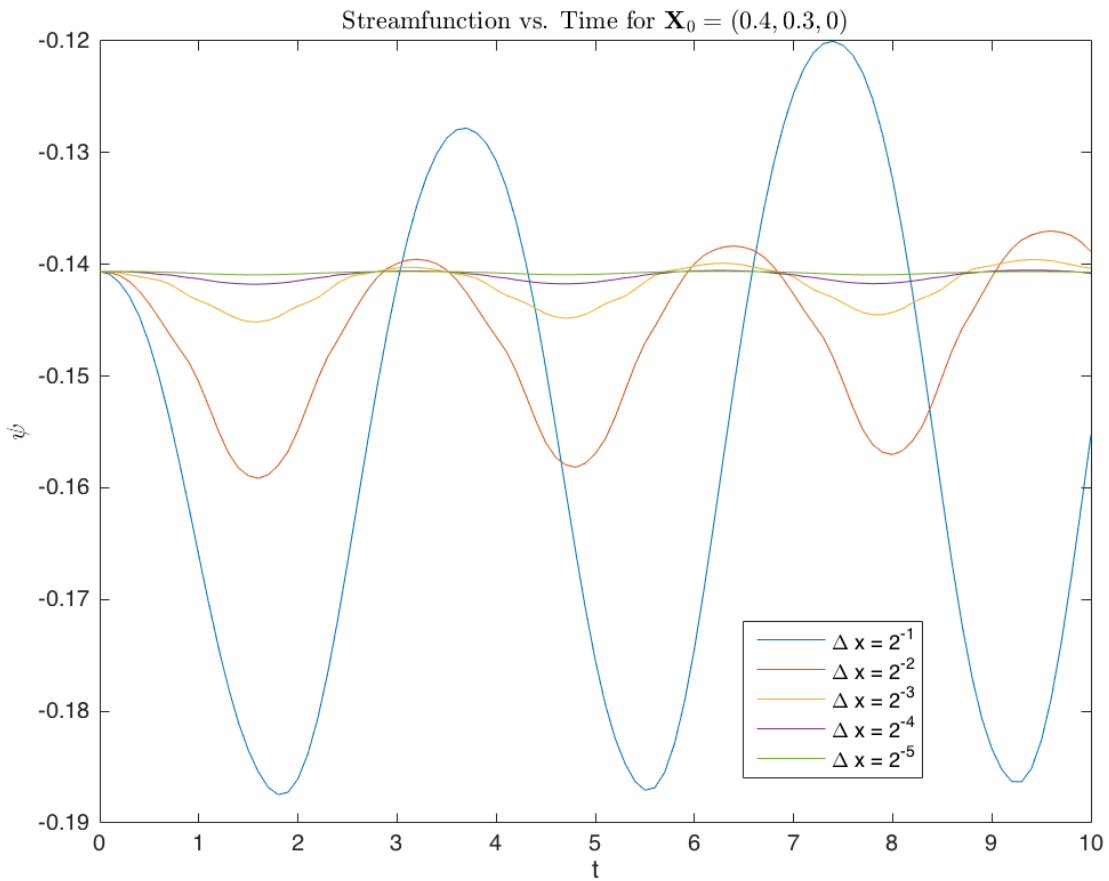
- Hyperbolic fixed points: $\mathbf{x}_1^* = (0, 0, -a)$, $\mathbf{x}_2^* = (0, 0, a)$
- Stable/unstable manifolds consist of sphere and vertical line through center of sphere
- Streamfunction (conserved along trajectories):

$$\psi(r, \theta) = -\frac{3}{4} U r^2 \left(1 - \frac{r^2}{a^2}\right) \sin^2 \theta$$

Validation: 3D Trajectory Calculation

Hill's spherical vortex: Streamfunction ψ vs. time for various interpolation Δx, $\mathbf{X}_0 = (0.4, 0.3, 0)$, $\Delta t = 0.1$, $t_f = 10$

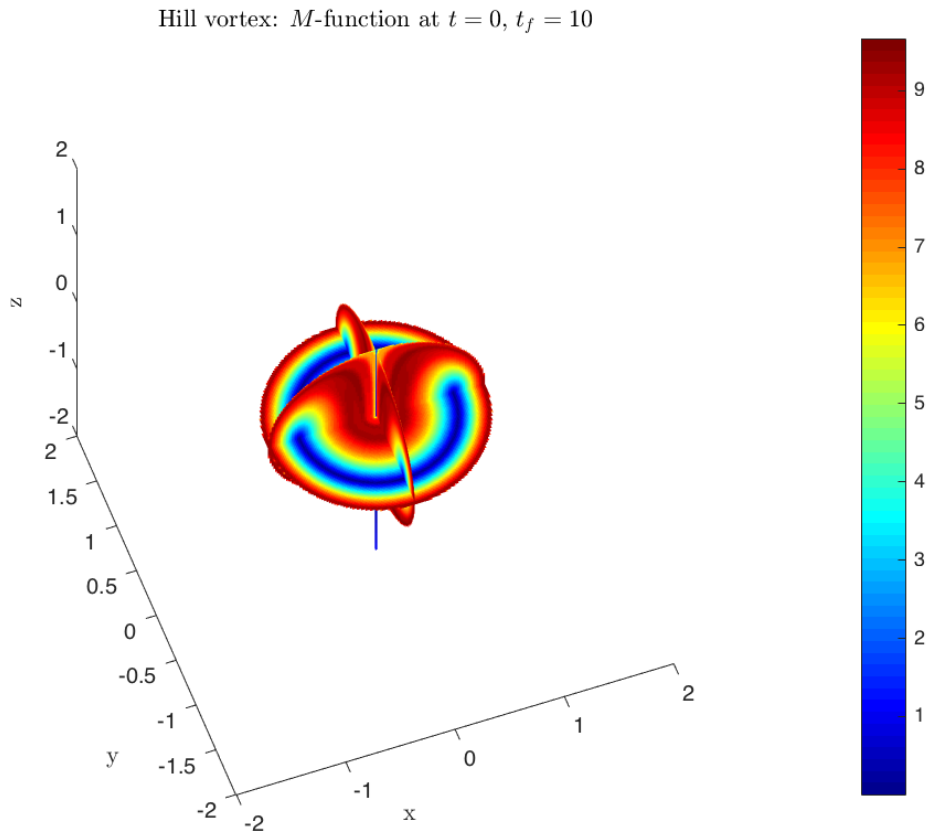
- ψ becomes more constant as $\Delta x \to 0$



Visual Validation: 3D Lagrangian Tools

Hill's vortex: M-function, $t_f = 10$:

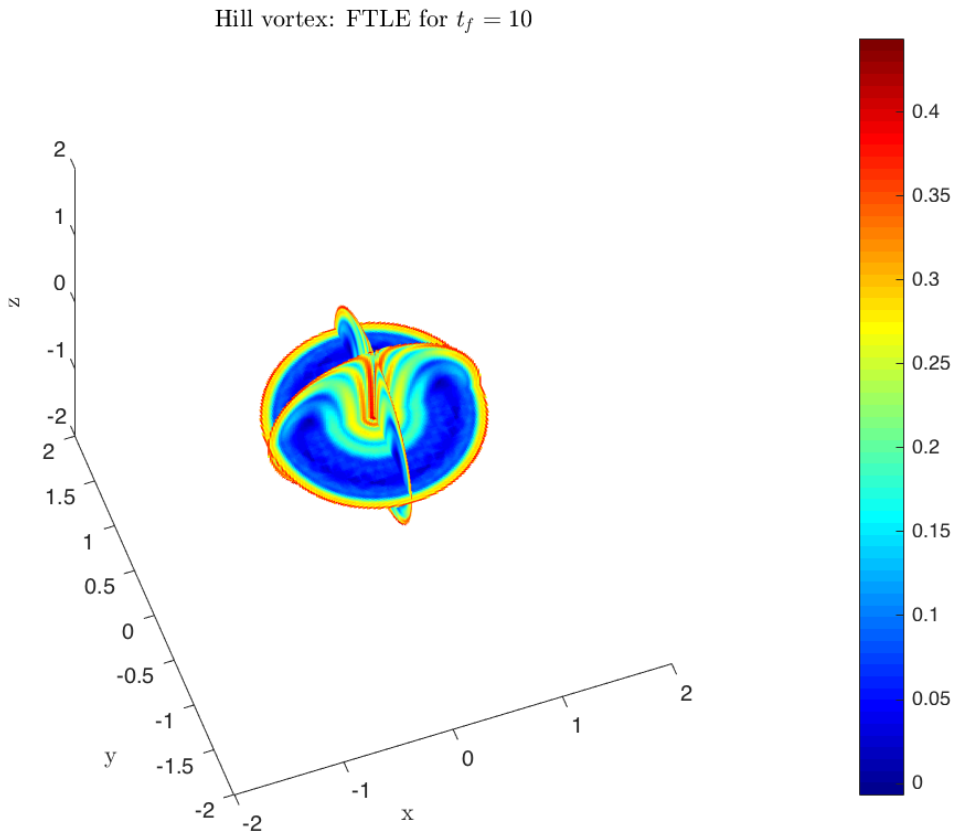
- Spherical + z-axis manifold clearly visible



Visual Validation: 3D Lagrangian Tools

Hill's vortex: FTLE, $t_f = 10$:

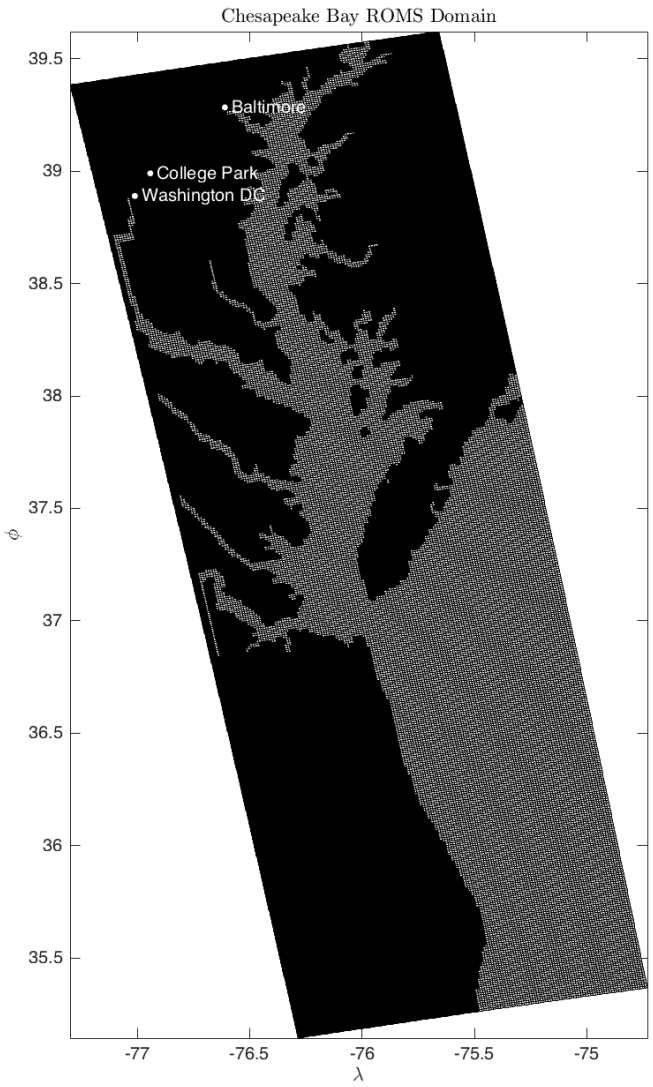
- Spherical manifold clearly visible



Application: Chesapeake Bay ROMS Dataset

Chesapeake Bay ROMS (Ches-ROMS) domain:

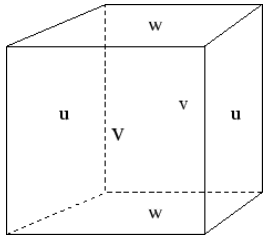
- Dimensions: $\sim$ 84 mi x 300 mi x 2.5-50 ft
- Discretized to 150 x 480 x 20 grid (terrain-following vertical coordinate)



Application: Chesapeake Bay ROMS Dataset

ChesROMS velocity data:

- Grid indexed by (ξ, η) instead of (i, j)
  - Arakawa C-grid, so (ξ_u, η_u) staggered from (ξ_v, η_v)



- Physical velocity components u_ξ, u_η, v_ξ and v_η proscribed at grid points (in m/s), also longitude (λ) and latitude (ϕ)
- Velocity given every two minutes on spatial grid

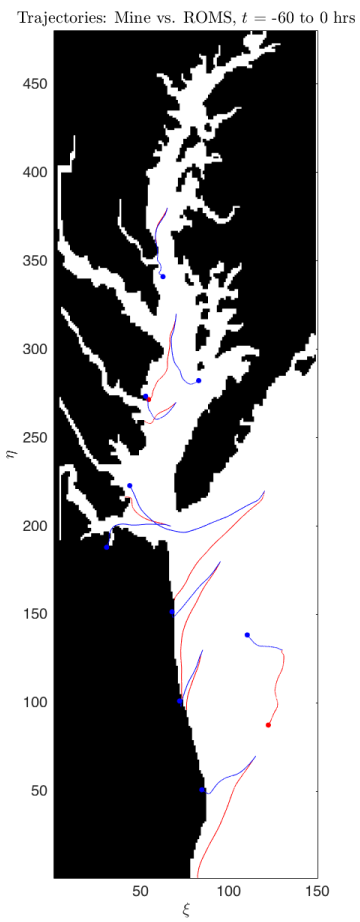
Application: Chesapeake Bay ROMS Dataset

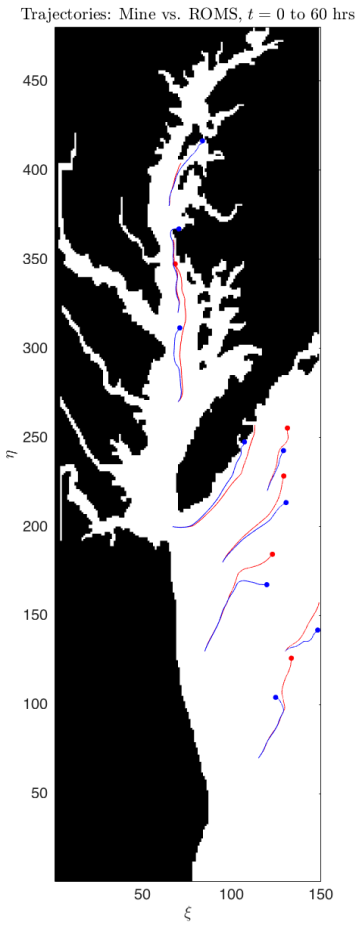
ROMS implementation details:

- Interpolation done in (ξ, η)-space (linear and bilinear only)
- Velocities normalized to index space (s^{-1}), integration done there
- M-function calculated in physical space (m)
- FTLE calculated from physical x and y displacements, estimated from λ and ϕ

Application: Chesapeake Bay ROMS Dataset

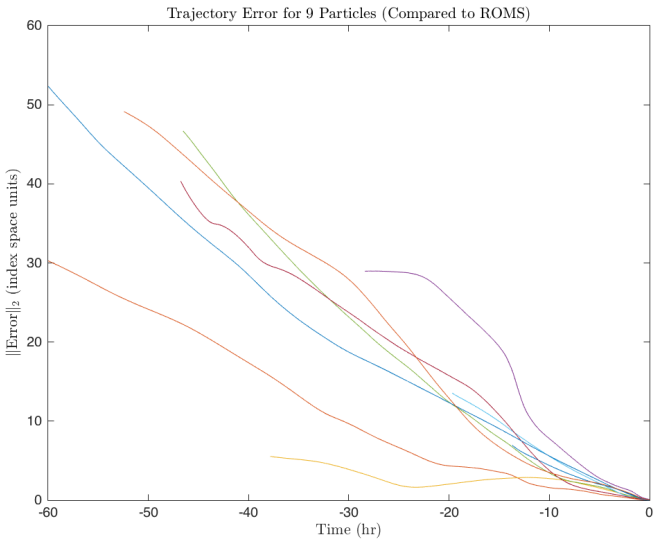
Validating Chesapeake trajectories: Mine vs. ROMS-generated, backwards (left) and forwards (right) 60 hrs, static flow field

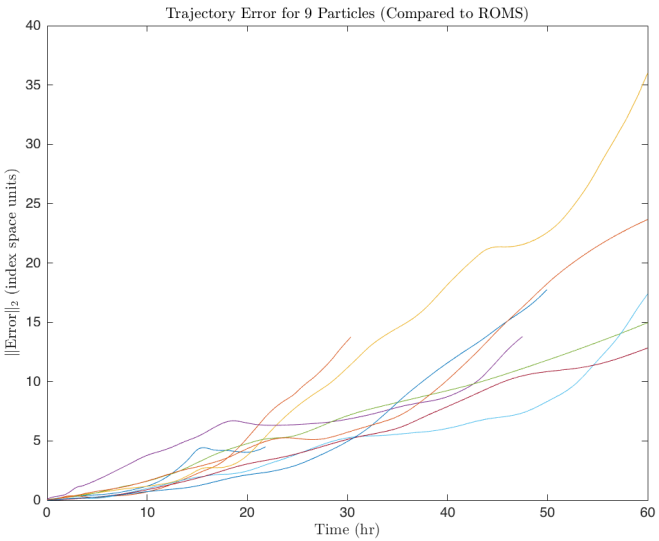




Application: Chesapeake Bay ROMS Dataset

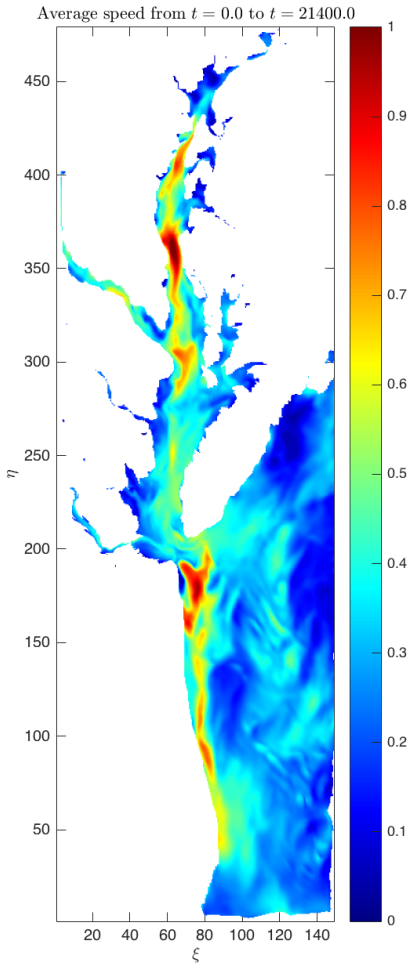
Trajectory error vs. time (2-norm in index space), backwards and forwards 60 hrs

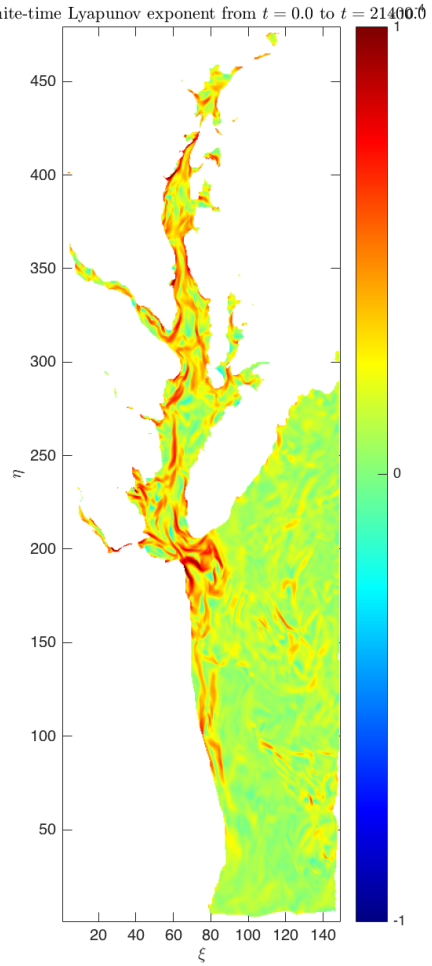




Application: Chesapeake Bay ROMS Dataset

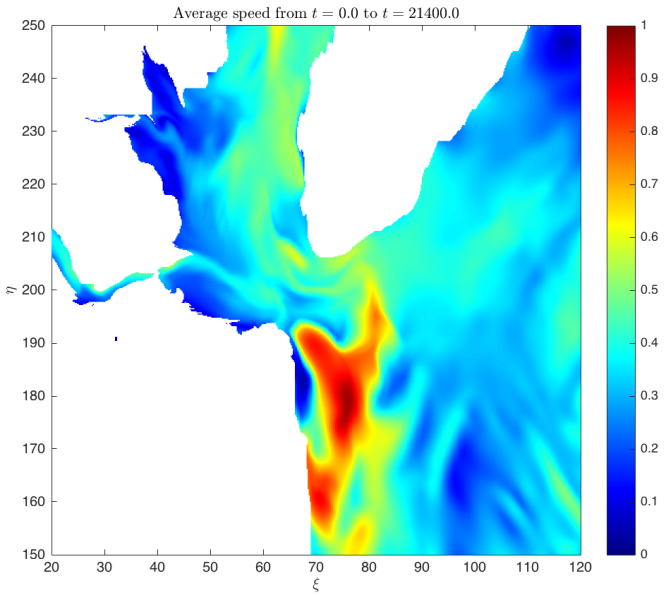
Average speed in m/s ($= \frac{M}{\tau}$) and FTLE, $\tau \approx 6$ hr, forward in time

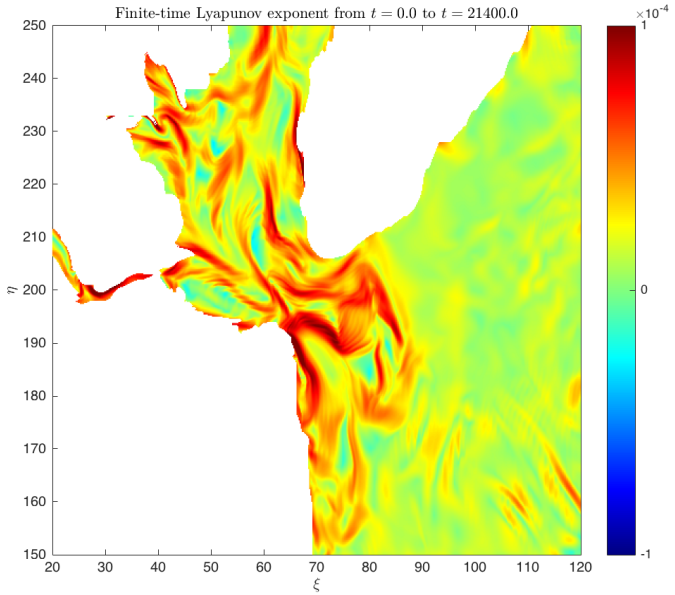




Application: Chesapeake Bay ROMS Dataset

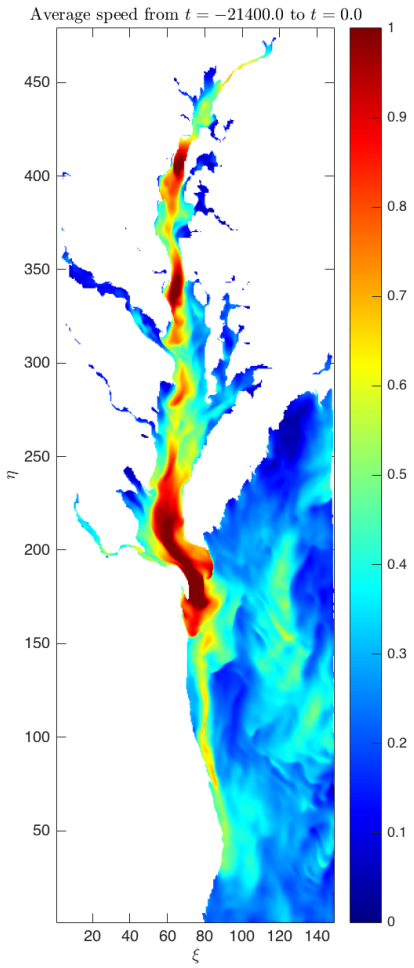
Average speed in m/s and FTLE, $\tau \approx 6$ hr, forward in time

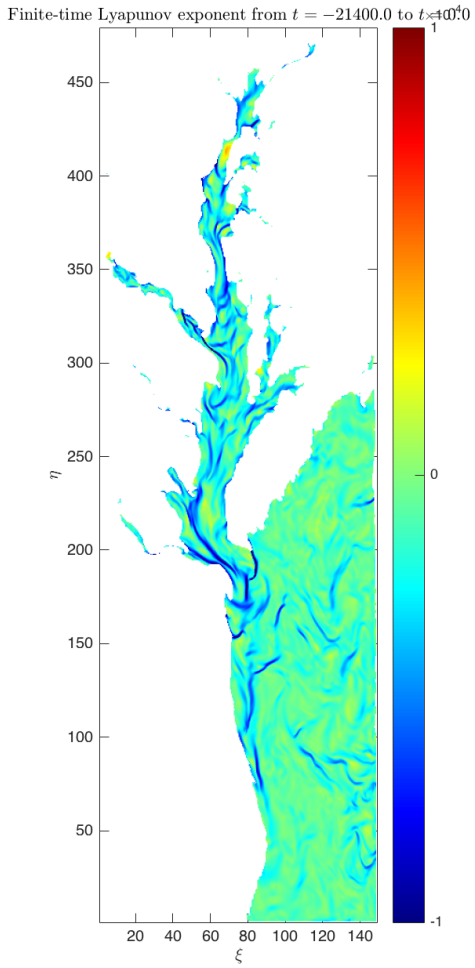




Application: Chesapeake Bay ROMS Dataset

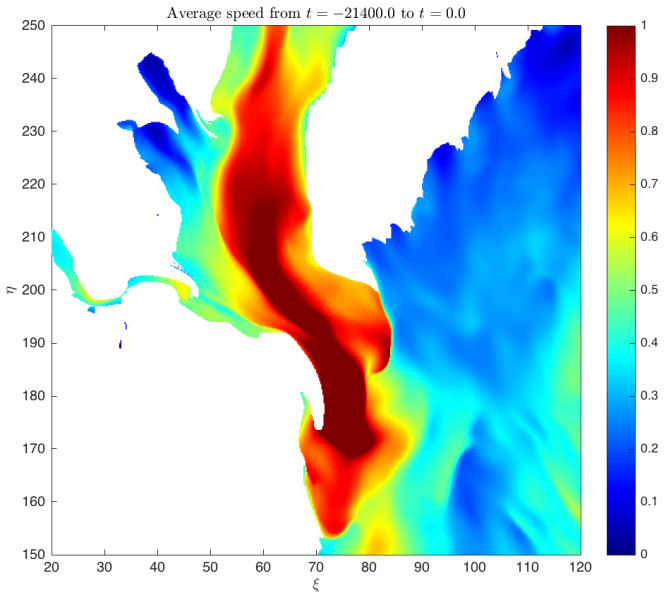
Average speed in m/s and FTLE, $\tau \approx 6$ hr, backward in time

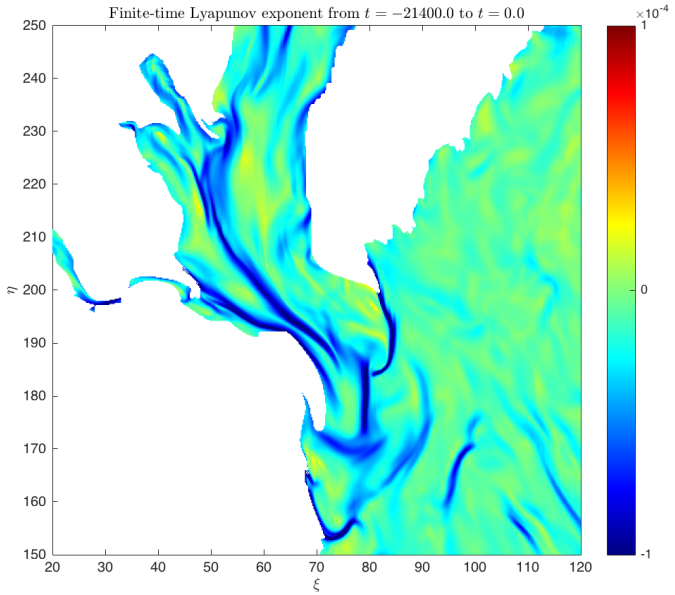




Application: Chesapeake Bay ROMS Dataset

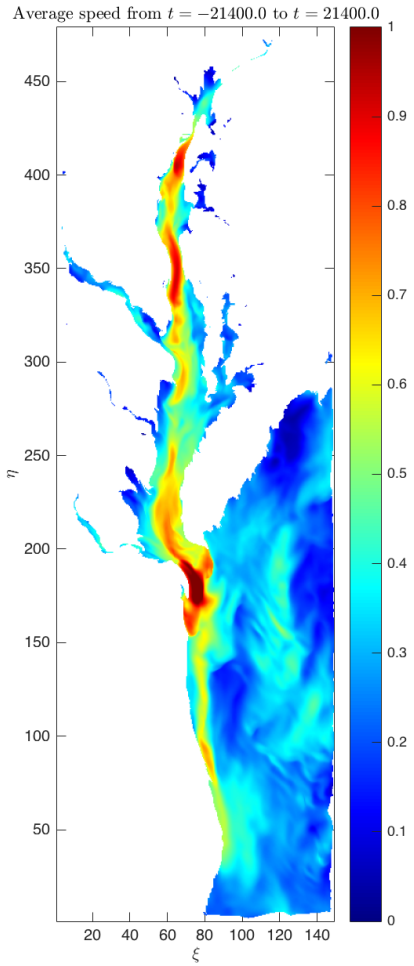
Average speed in m/s and FTLE, $\tau \approx 6$ hr, backward in time

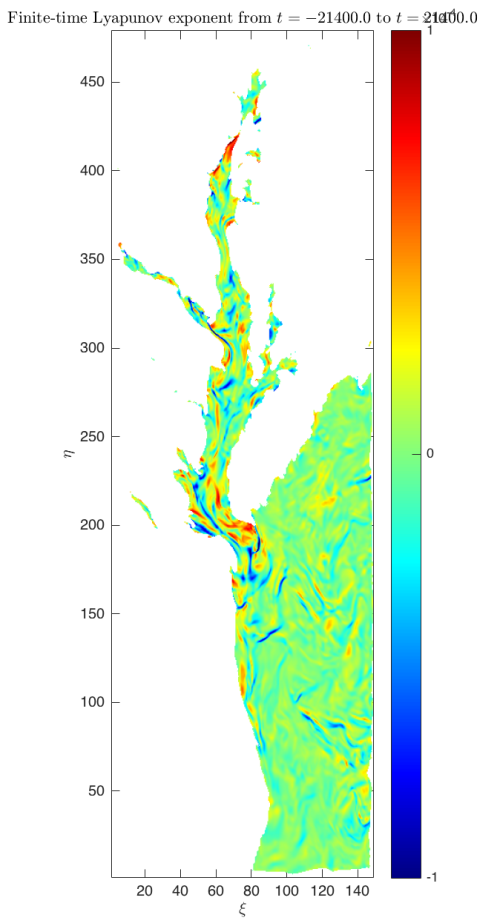




Application: Chesapeake Bay ROMS Dataset

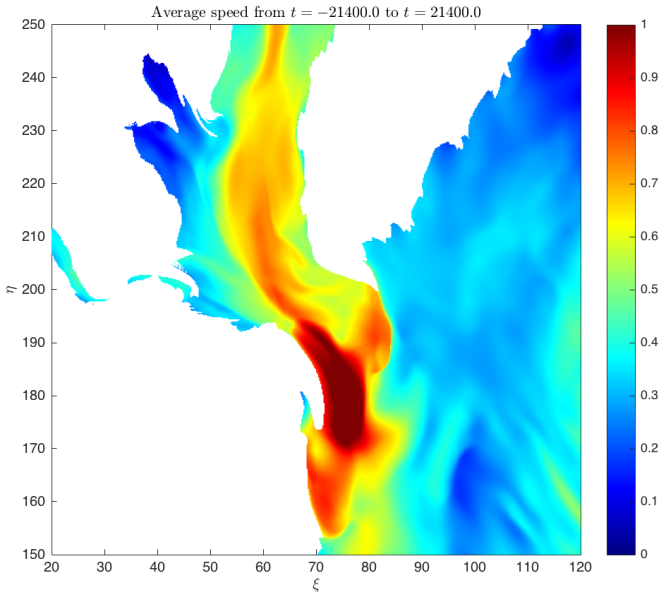
Average speed in m/s and FTLE, $\tau \approx 6$ hr, forward and backward in time

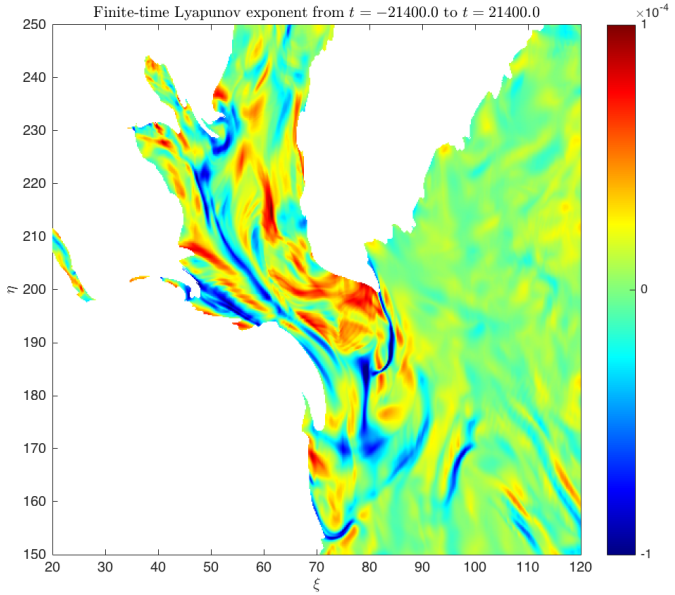




Application: Chesapeake Bay ROMS Dataset

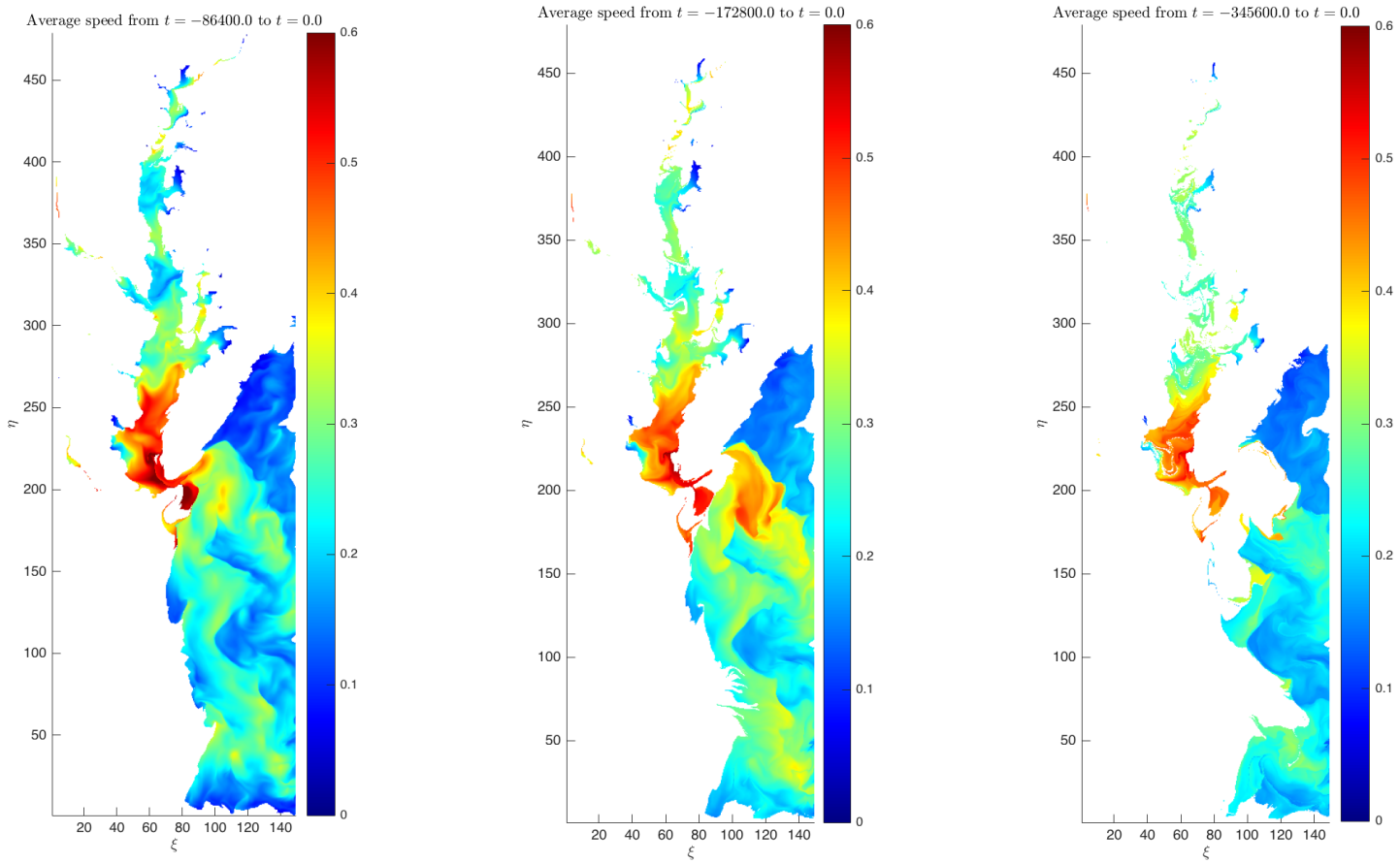
Average speed in m/s and FTLE, $\tau \approx 6$ hr, forward and backward in time





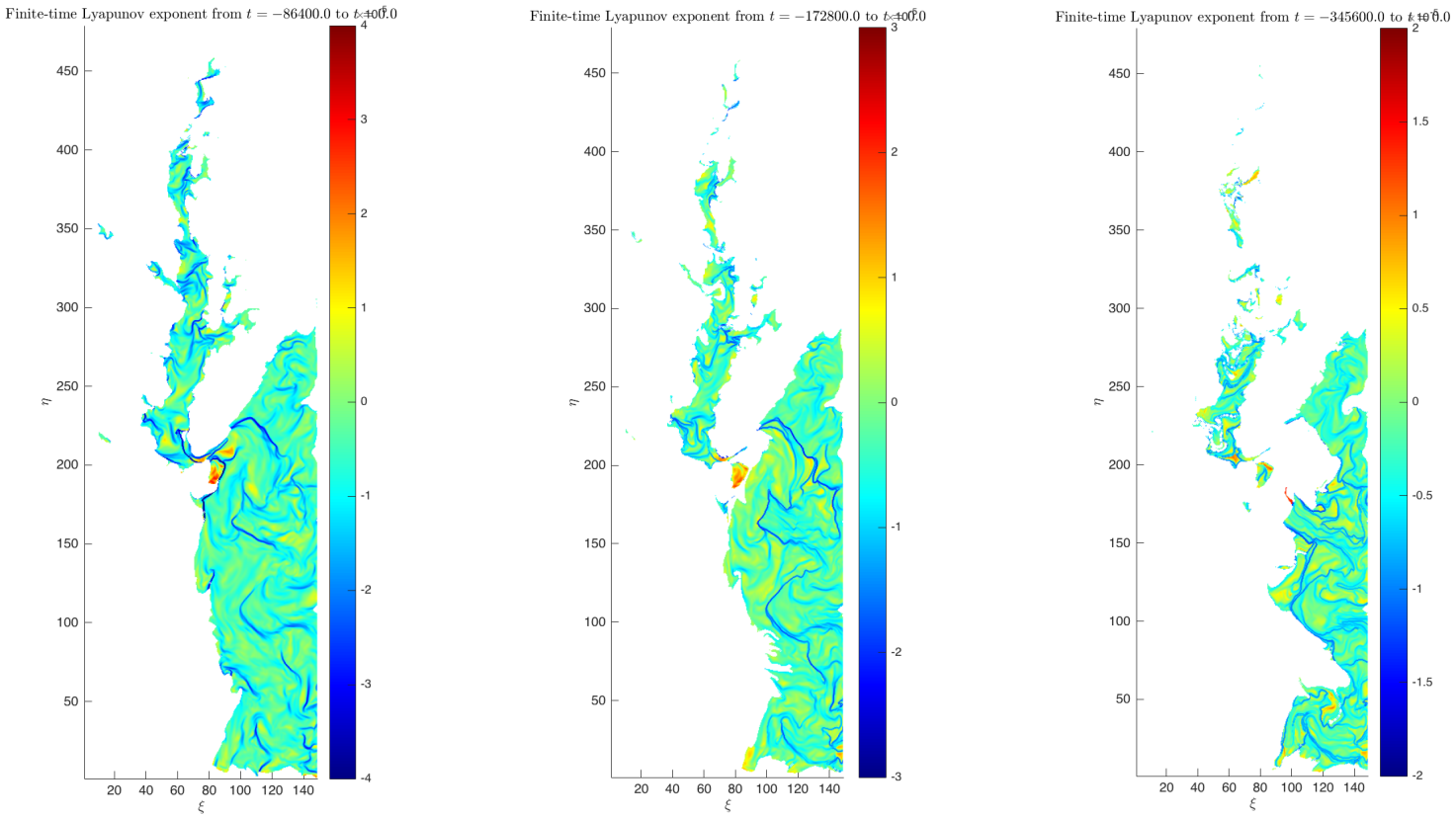
Application: Chesapeake Bay ROMS Dataset

Average speed in m/s, $\tau \approx 1$ (left), 2 (center), and 4 (right) days, backward in time



Application: Chesapeake Bay ROMS Dataset

Average speed in m/s, $\tau \approx 1$ (left), 2 (center), and 4 (right) days, backward in time (changing scale)



Conclusions

Conclusions:

- Both M-function and FTLE reveal coherent structures in an ocean flow
- Boundaries of coherent structures given by large gradients in M, ridges and troughs of FTLE
- Boundary conditions important: some version of no-slip or free-slip preferable to nothing
- Interpretation somewhat sensitive to color scale, especially for FTLE

Timeline

- October - Mid-November
  - Project proposal presentation and paper ✓
  - 2D and 3D interpolation ✓
- Mid-November - December
  - 2D trajectory implementation and validation ✓
  - M function implementation and validation ✓
  - Mid-year report and presentation ✓
- January - February
  - 3D trajectory implementation and validation (mostly)
  - FTLE implementation ✓
  - Tailor all existing code to work with ROMS data (mostly)
- March - April
  - Apply tools to Chesapeake ROMS dataset (mostly)
  - Final presentation and paper (in progress)

References

[1] Georg Duffing. *Erzwungene Schwingungen bei veränderlicher Eigenfrequenz und ihre technische Bedeutung*. Number 41-42. R, Vieweg & Sohn, 1918.

[2] Micaiah John Muller Hill. On a spherical vortex. *Proceedings of the Royal Society of London*, 55(331-335):219–224, 1894.

[3] Ana M Mancho, Stephen Wiggins, Jezabel Curbelo, and Carolina Mendoza. Lagrangian descriptors: A method for revealing phase space structures of general time dependent dynamical systems. *Communications in Nonlinear Science and Numerical Simulation*, 18(12):3530–3557, 2013.

[4] Carolina Mendoza and Ana M Mancho. Hidden geometry of ocean flows. *Physical review letters*, 105(3):038501, 2010.

[5] Alvaro H Salas. Exact solution to duffing equation and the pendulum equation. *Applied Mathematical Sciences*, 8(176):8781–8789, 2014.