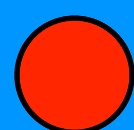
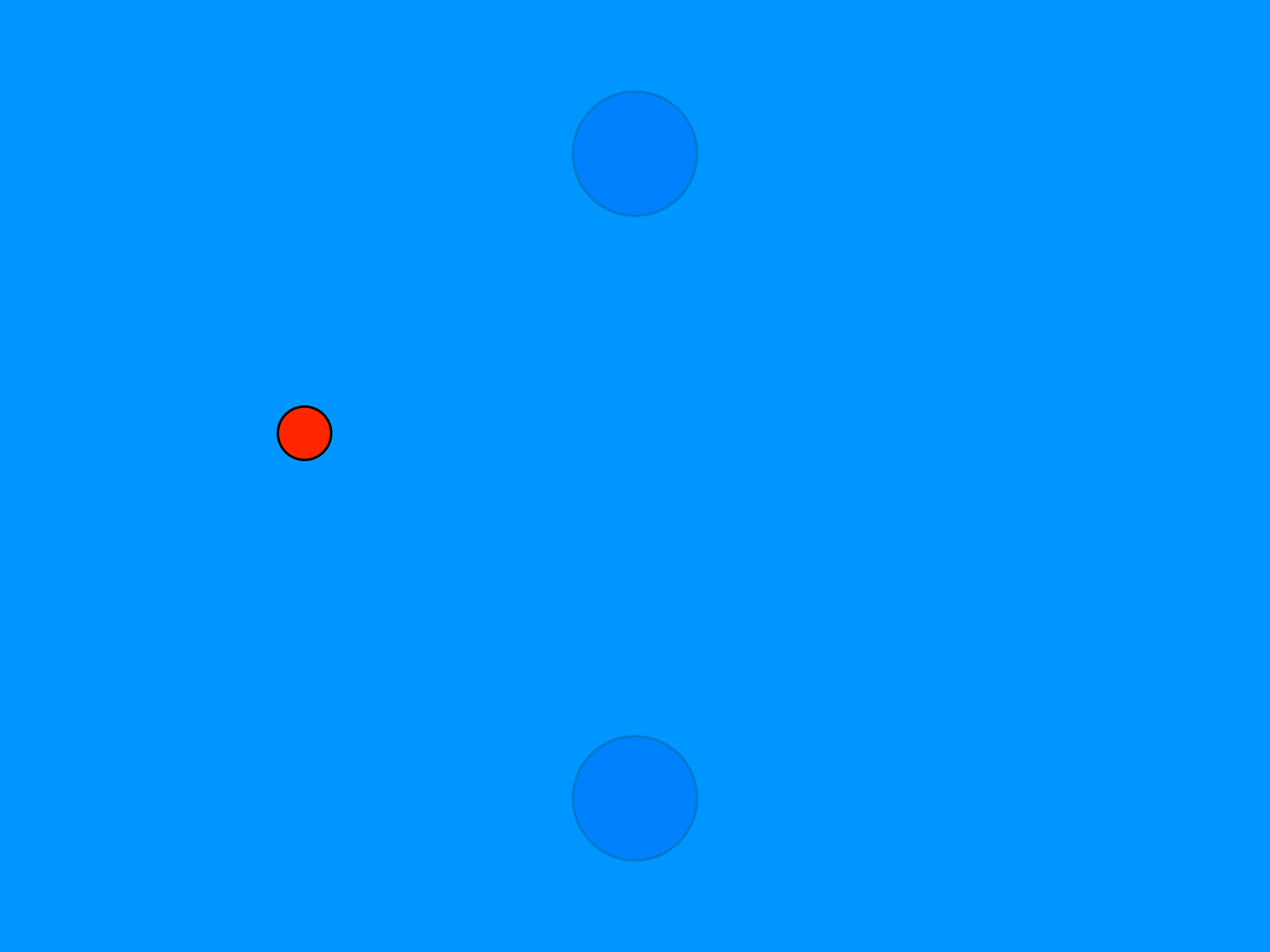
Lagrangian Data Assimilation and Manifold Detection for a Point-Vortex Model

David Darmon, AMSC

Kayo Ide, AOSC, IPST, CSCAMM, ESSIC

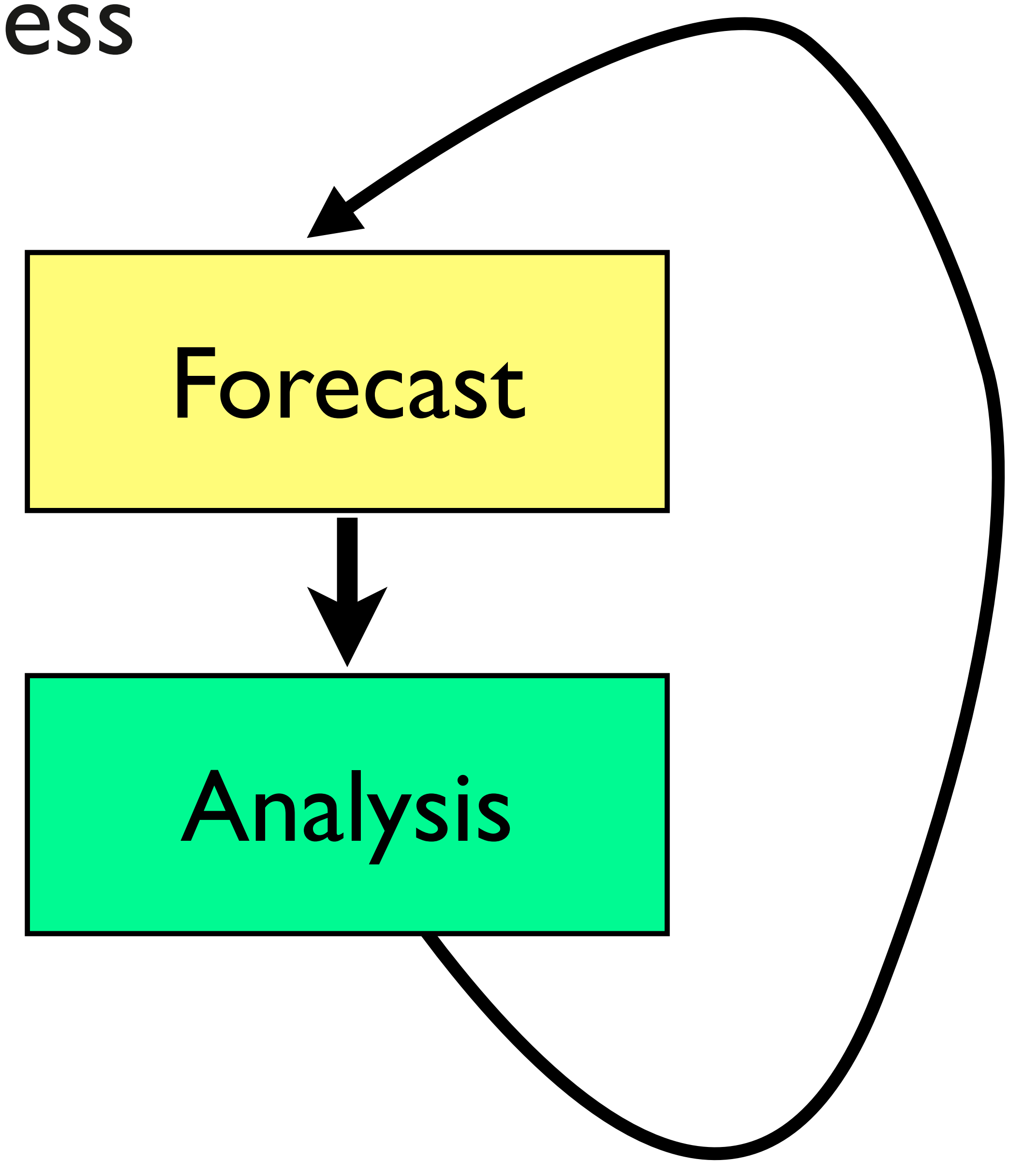




Background

Data Assimilation

Iterative process



Background

Data Assimilation

True System Dynamics

$$d\mathbf{x}^t = M(\mathbf{x}^t, t)dt + d\boldsymbol{\eta}^t$$

$\mathbf{x}^t$ – system state

$\boldsymbol{\eta}^t$ – Brownian motion

$M(\cdot, \cdot)$ – deterministic evolution operator

Background

Data Assimilation

Observations

$$\mathbf{y}_k^o = h(\mathbf{x}^t(t_k)) + \boldsymbol{\epsilon}_k^t$$

$\mathbf{y}_k^o$ – observation of system at time t_k

$\boldsymbol{\epsilon}_k^t$ – Gaussian noise

$h(\cdot)$– observation operator

Background

Data Assimilation

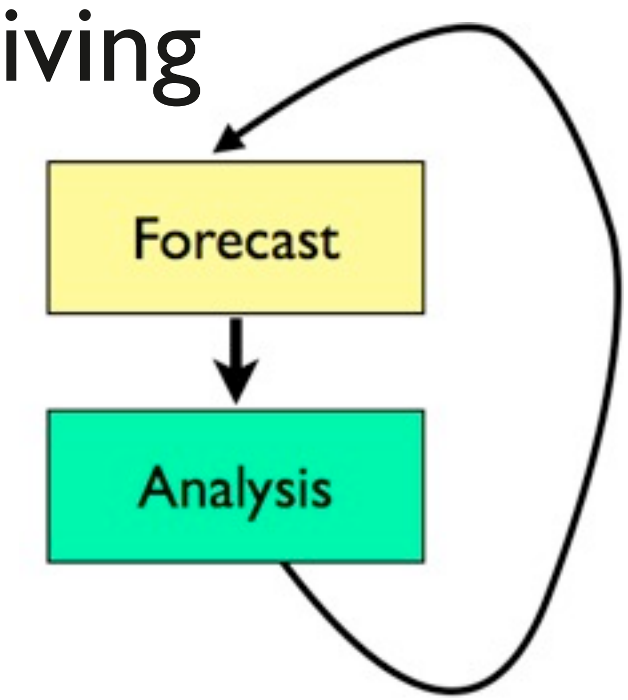
Forecast and Analysis

Evolve a forecasted state forward *somehow*

$$\frac{d}{dt}\mathbf{x}^f = f(\mathbf{x}^f, t, \boldsymbol{\eta}^f)$$

Perform analysis step upon receiving observation *somehow*

$$\mathbf{x}^a = g(\mathbf{x}^f, \mathbf{y}_k^o, \epsilon_k^t)$$

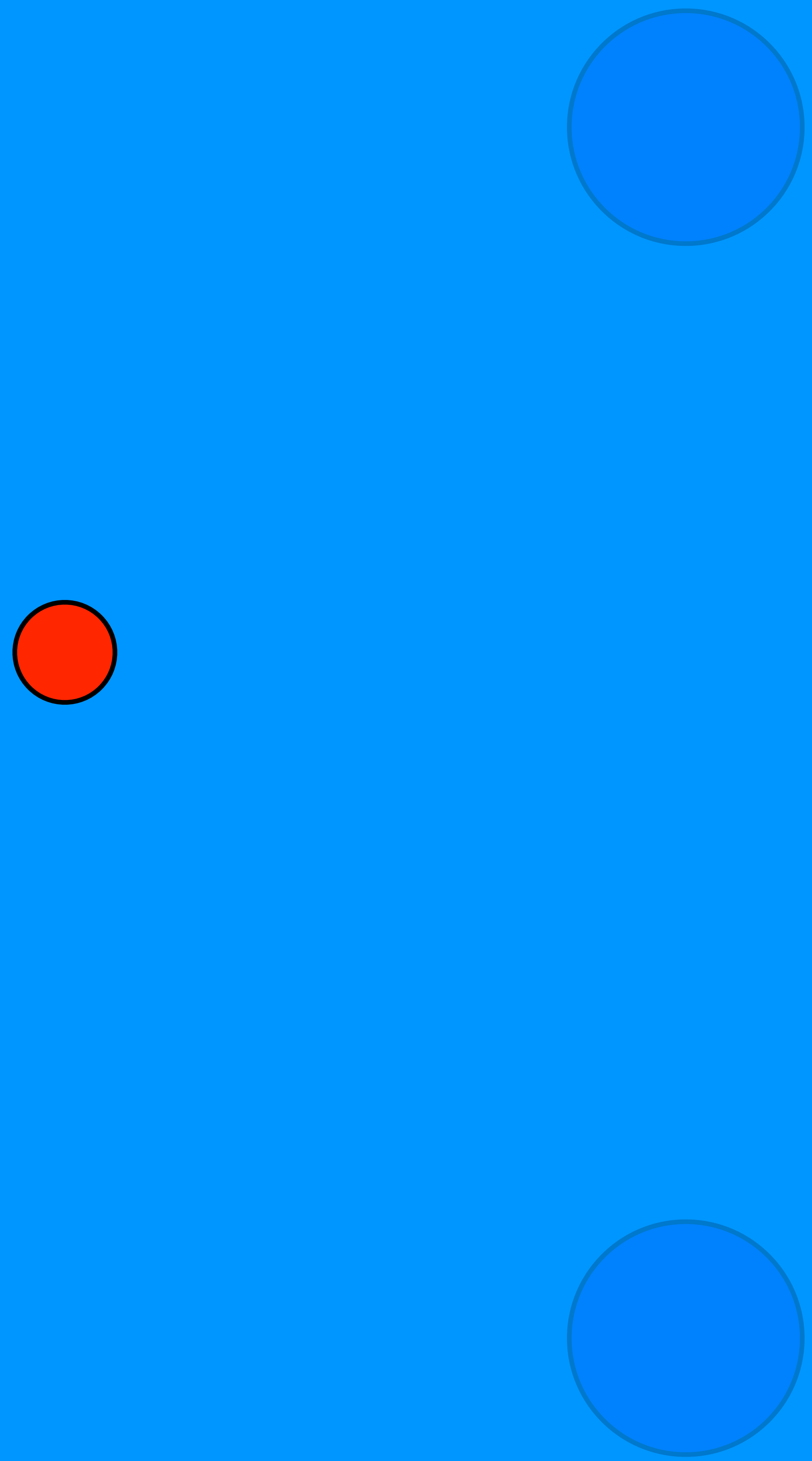


Background

Lagrangian Data Assimilation

Consider joint state of flow (*F*) and drifters (*D*)

$$\mathbf{x}^t = \begin{pmatrix} \mathbf{x}_F^t \\ \mathbf{x}_D^t \end{pmatrix}$$



Background

Model

Point-Vortex (in complex plane)

$$\frac{dz_j^*}{dt} = \sum_{j'=1, j' \neq j}^{N_v} \frac{i}{2\pi} \frac{\Gamma_{j'}}{z_j - z_{j'}}, \; j = 1, \ldots, N_v$$

$$\frac{d\xi_k^*}{dt} = \sum_{j=1}^{N_v} \frac{i}{2\pi} \frac{\Gamma_j}{\xi_k - z_j}, \qquad k = 1, \ldots, N_d$$

z_j – j^{th} vortex location

ξ_k – k^{th} drifter location

Γ_j – vorticity of j^{th} vortex

Background

Model

Demo

Background

Observing System Design

Deploying drifters is *expensive*
Obtaining measurements is *difficult*

Q: Can we design a system such that we optimally place the drifters to determine the location of the vortices?

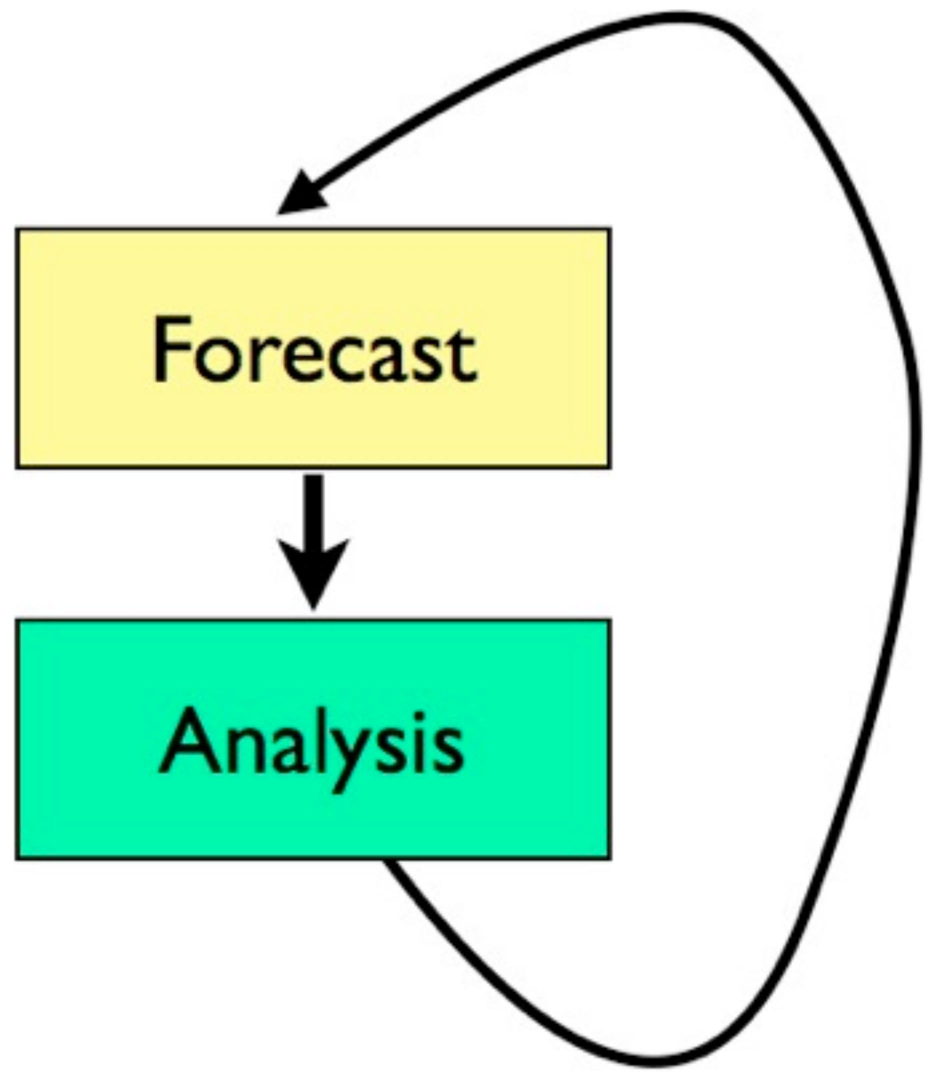
Background

Observing System Design

Deploying drifters is *expensive*
Obtaining measurements is *difficult*

Q: Can we design a system such that we optimally place the drifters to determine the location of the vortices?

Phase I



Approach, Phase I

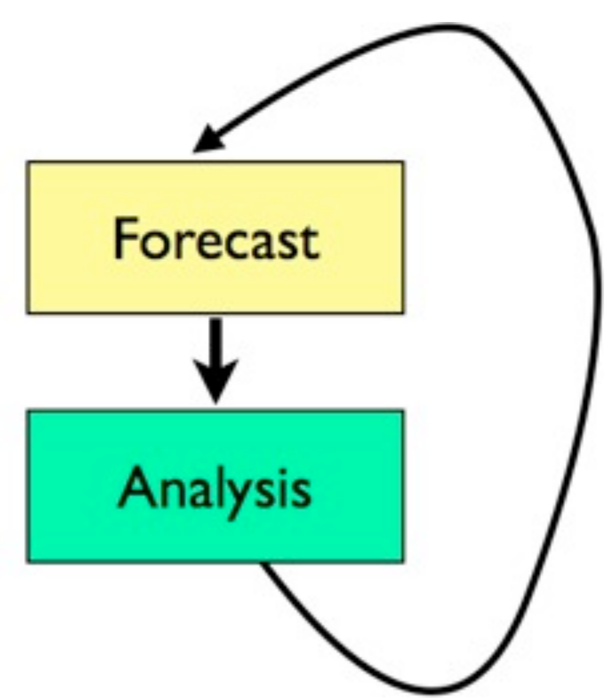
Lagrangian Data Assimilation

Fokker-Planck Equation, Forecast

$$\frac{\partial p}{\partial t} + \sum_i \frac{\partial m_i p}{\partial x_i} = \frac{1}{2} \sum_{i,j} \frac{\partial^2 p(\mathbf{Q})_{ij}}{\partial x_i \partial x_j}$$

$p \equiv p(\mathbf{x}^t, t)$ – probability density of $\mathbf{x}^t$

$\mathbf{Q}$ – covariance matrix from SDE



Approach, Phase I

Lagrangian Data Assimilation

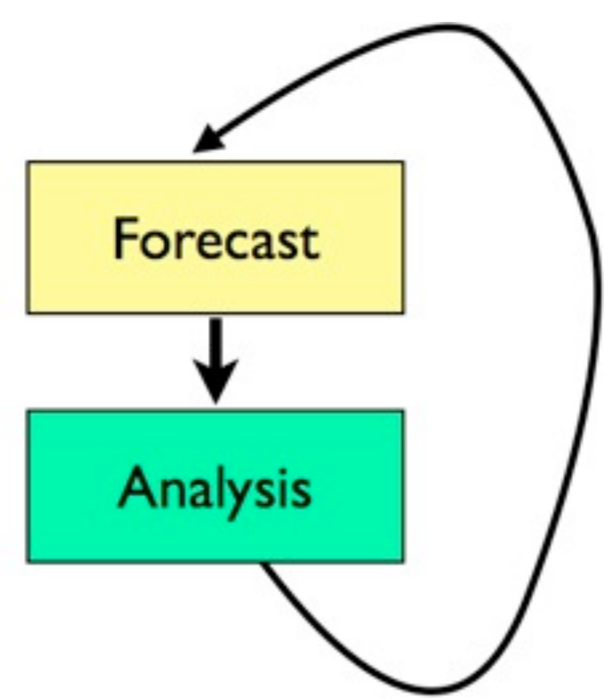
Bayes's Theorem, Analysis

$$p(\mathbf{x}^f|\mathbf{y}_k^o) = \frac{p(\mathbf{y}_k^o|\mathbf{x}^f)p(\mathbf{x}^f)}{\int p(\mathbf{y}_k^o|\mathbf{x}^f)p(\mathbf{x}^f)\,d\mathbf{x}^f}$$

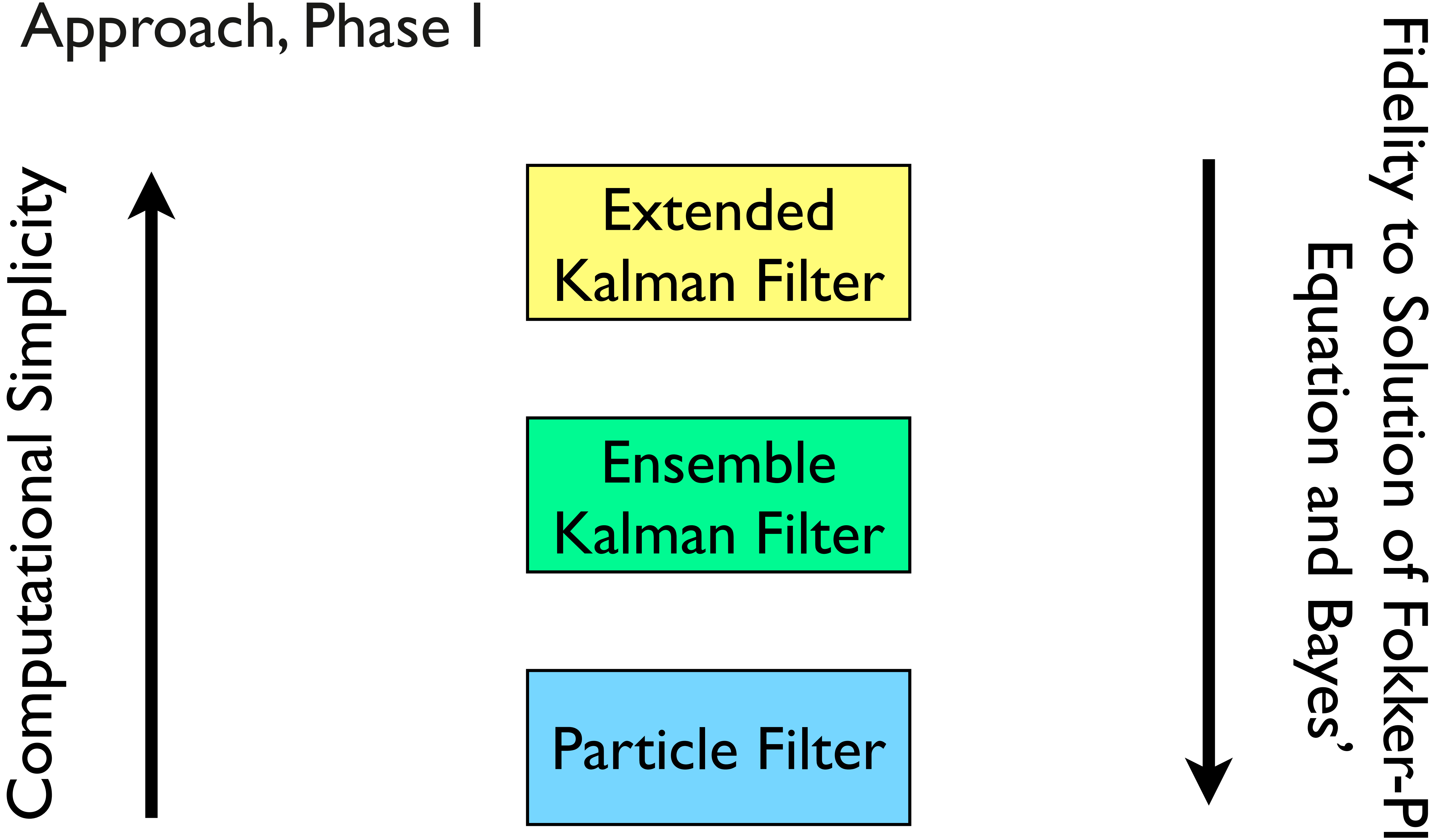
$p(\mathbf{x}^f|\mathbf{y}_k^o)$ – posterior distribution

$p(\mathbf{y}_k^o|\mathbf{x}^f)$ – likelihood

$p(\mathbf{x}^f)$ – prior distribution



Approach, Phase I



Approach, Phase I

Lagrangian Data Assimilation

Extended Kalman Filter (EKF)

Generalization of Kalman filter to nonlinear equations

Gaussian assumptions on dynamical and observational noise

Represent pdf by mean and covariance matrix

Approach, Phase I

Lagrangian Data Assimilation

Extended Kalman Filter (EKF)

$\mathbf{x}^f$ – mean state

$\mathbf{P}^f$ – covariance matrix

Approach, Phase I

Lagrangian Data Assimilation

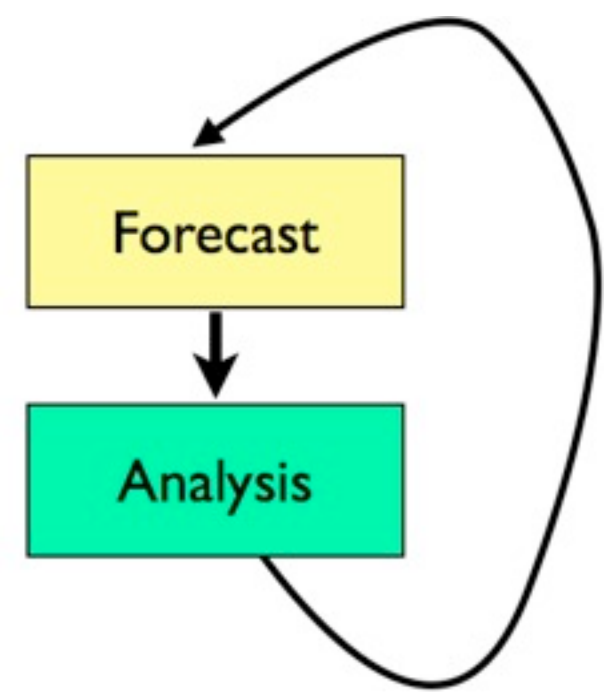
Extended Kalman Filter (EKF), Forecast

$$\frac{d}{dt}\mathbf{x}^f = M(\mathbf{x}^f, t)$$

$$\frac{d}{dt}\mathbf{P}^f = \mathbf{M}(t)\mathbf{P}^f + \mathbf{P}^f\mathbf{M}^T(t) + \mathbf{Q}$$

$\mathbf{Q}$ – covariance matrix from SDE

$\mathbf{M}(t) = J[M(\mathbf{x}, t)]\big|_{x=x^f}$ - Jacobian of M



Approach, Phase I

Lagrangian Data Assimilation

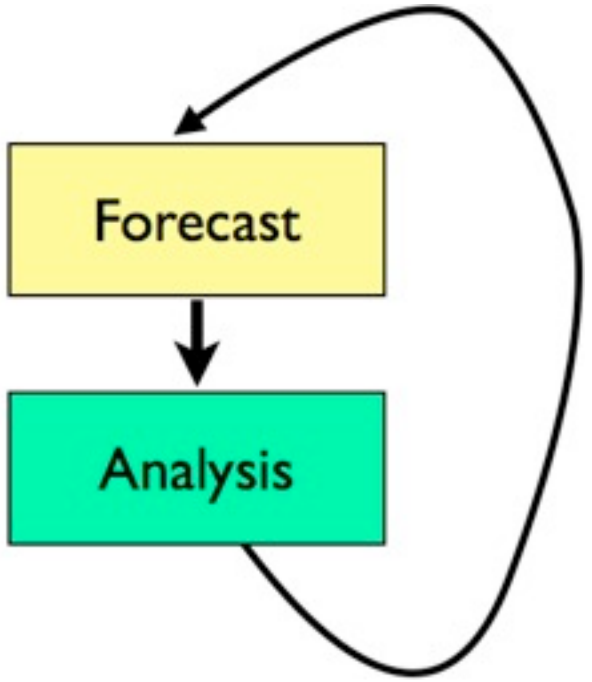
Extended Kalman Filter (EKF), Forecast

$$\frac{d}{dt}\mathbf{x}^f = M(\mathbf{x}^f, t)$$

$$\frac{d}{dt}\mathbf{P}^f = \mathbf{M}(t)\mathbf{P}^f + \mathbf{P}^f\mathbf{M}^T(t) + \mathbf{Q}$$

$\mathbf{Q}$ – covariance matrix from SDE

$\mathbf{M}(t) = J[M(\mathbf{x}, t)]\big|_{x=x^f}$ - Jacobian of M



Approach, Phase I

Lagrangian Data Assimilation

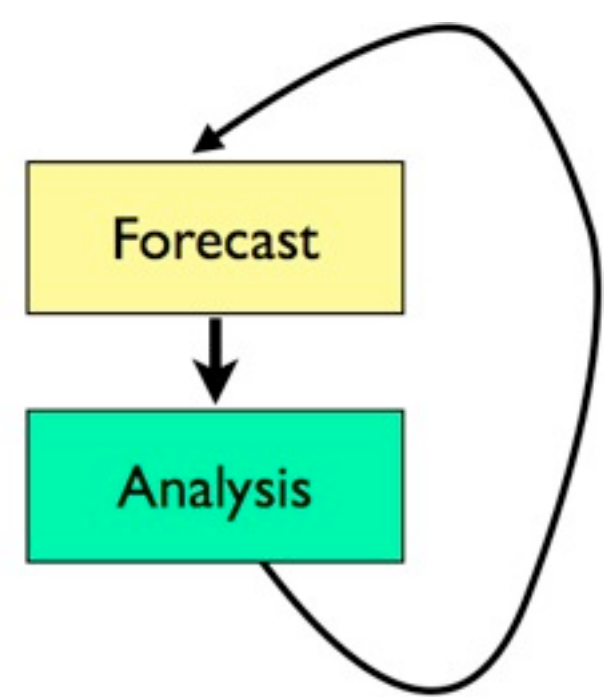
Extended Kalman Filter (EKF), Forecast

$$\frac{d}{dt}\mathbf{x}^f = M(\mathbf{x}^f, t)$$

$$\frac{d}{dt}\mathbf{P}^f = \mathbf{M}(t)\mathbf{P}^f + \mathbf{P}^f\mathbf{M}^T(t) + \mathbf{Q}$$

$\mathbf{Q}$ – covariance matrix from SDE

$\mathbf{M}(t) = J[M(\mathbf{x}, t)]\big|_{x=x^f}$ - Jacobian of M



Approach, Phase I

Lagrangian Data Assimilation

Extended Kalman Filter (EKF), Analysis

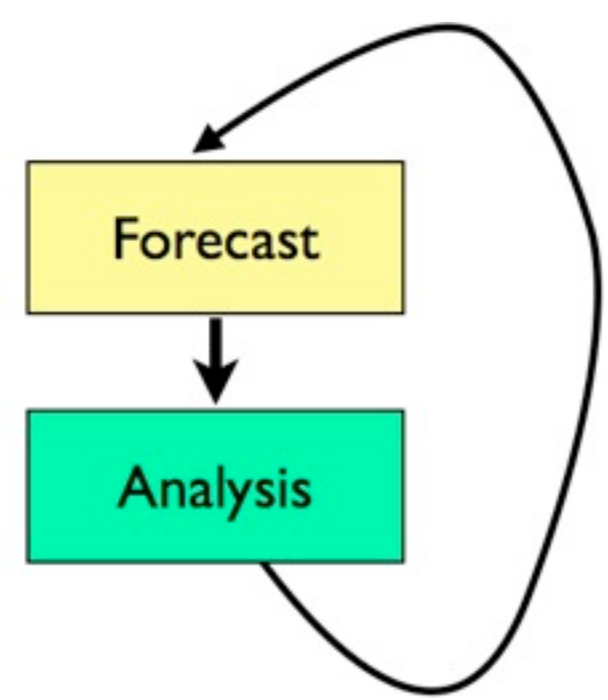
$$\mathbf{x}_k^a = \mathbf{x}^f(t_k) + \mathbf{K}_k(\mathbf{y}_k^o - h_k(\mathbf{x}^f(t_k)))$$

$$\mathbf{P}_k^a = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}^f(t_k)$$

$$\mathbf{K}_k = \mathbf{P}^f(t_k)\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}^f(t_k)\mathbf{H}_k^T + \mathbf{R}_k^o)^{-1}$$

$\mathbf{R}^o$ – covariance matrix from observation

$\mathbf{H}_k = J[h(\mathbf{x}, t_k)]\big|_{x=x^f}$ - Jacobian of h_k



Approach, Phase I

Lagrangian Data Assimilation

Extended Kalman Filter (EKF), Analysis

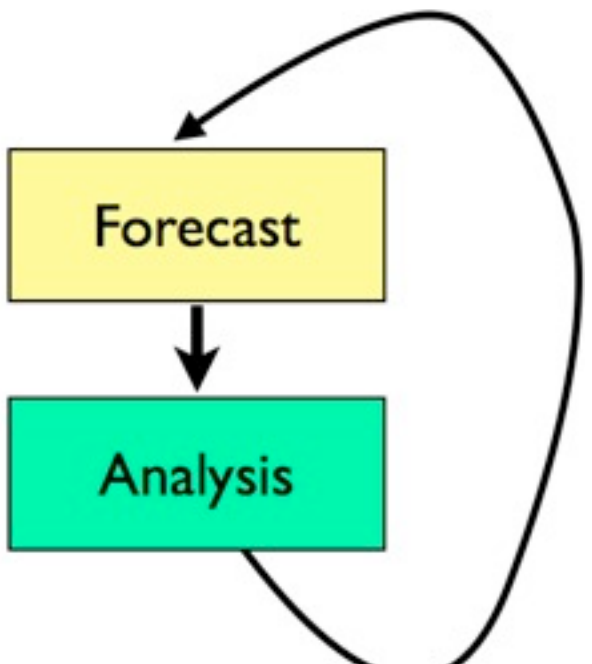
$$\mathbf{x}_k^a = \mathbf{x}^f(t_k) + \mathbf{K}_k(\mathbf{y}_k^o - h_k(\mathbf{x}^f(t_k)))$$

$$\mathbf{P}_k^a = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}^f(t_k)$$

$$\mathbf{K}_k = \mathbf{P}^f(t_k)\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}^f(t_k)\mathbf{H}_k^T + \mathbf{R}_k^o)^{-1}$$

$\mathbf{R}^o$ – covariance matrix from observation

$\mathbf{H}_k = J[h(\mathbf{x}, t_k)]\big|_{x=x^f}$ - Jacobian of h_k



Approach, Phase I

Lagrangian Data Assimilation

Extended Kalman Filter (EKF), Analysis

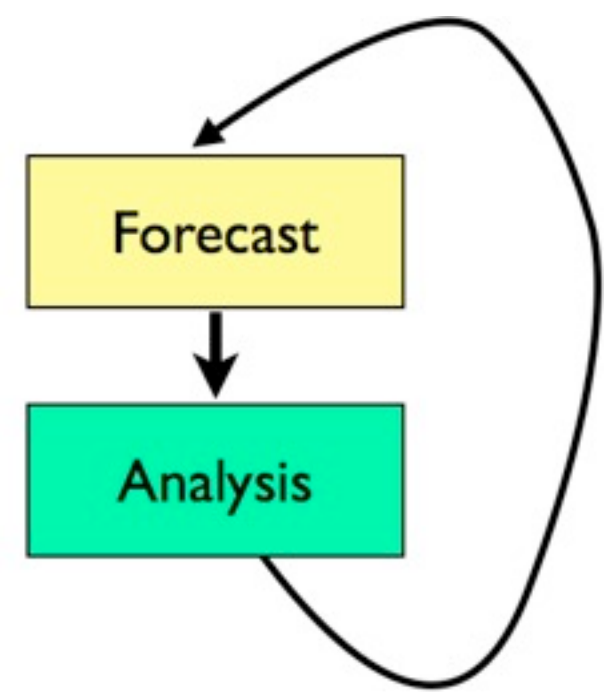
$$\mathbf{x}_k^a = \mathbf{x}^f(t_k) + \mathbf{K}_k(\mathbf{y}_k^o - h_k(\mathbf{x}^f(t_k)))$$

$$\mathbf{P}_k^a = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}^f(t_k)$$

$$\mathbf{K}_k = \mathbf{P}^f(t_k)\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}^f(t_k)\mathbf{H}_k^T + \mathbf{R}_k^o)^{-1}$$

$\mathbf{R}^o$ – covariance matrix from observation

$\mathbf{H}_k = J[h(\mathbf{x}, t_k)]\big|_{x=x^f}$ - Jacobian of h_k



Approach, Phase I

Lagrangian Data Assimilation

Extended Kalman Filter (EKF), Analysis

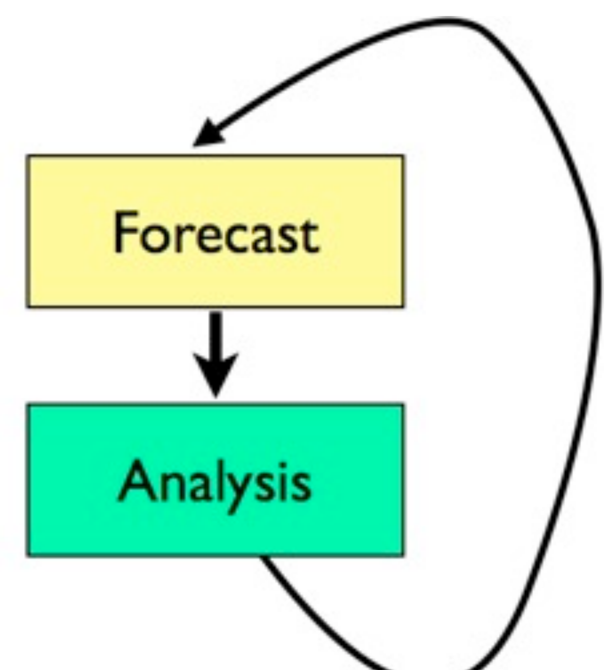
$$\mathbf{x}_k^a = \mathbf{x}^f(t_k) + \mathbf{K}_k(\mathbf{y}_k^o - h_k(\mathbf{x}^f(t_k)))$$

$$\mathbf{P}_k^a = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}^f(t_k)$$

$$\mathbf{K}_k = \mathbf{P}^f(t_k)\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}^f(t_k)\mathbf{H}_k^T + \mathbf{R}_k^o)^{-1}$$

$\mathbf{R}^o$ – covariance matrix from observation

$\mathbf{H}_k = J[h(\mathbf{x}, t_k)]\big|_{x=x^f}$ - Jacobian of h_k



Approach, Phase I

Lagrangian Data Assimilation

Extended Kalman Filter (EKF)

- Weakly assumes nonlinearity
- Assumes Gaussianity
  - **NOT** valid for nonlinear systems

Approach, Phase I

Lagrangian Data Assimilation

Ensemble Kalman Filter (EnKF)

Approximate pdf by an ensemble of particles:

$$\{\mathbf{x}_i^f\}_{i=1}^N$$

Forecast: Evolve particles forward using SDE

Analysis: Perform Kalman-like analysis

Approach, Phase I

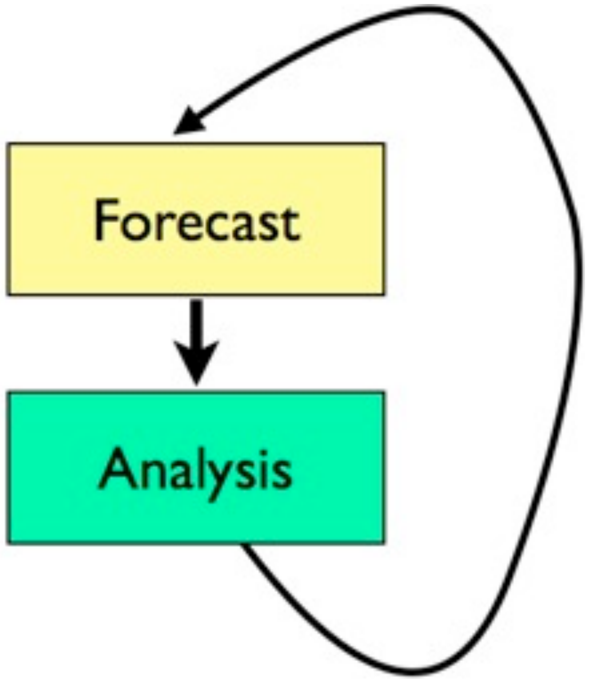
Lagrangian Data Assimilation

Ensemble Kalman Filter (EnKF)

Computing (ensemble) moments

$$\bar{\mathbf{x}}^f = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i^f .$$

$$\mathbf{P}_e^f = \frac{1}{N-1} \sum_{i=1}^{N} (\mathbf{x}_i^f - \bar{\mathbf{x}}^f)(\mathbf{x}_i^f - \bar{\mathbf{x}}^f)^T$$



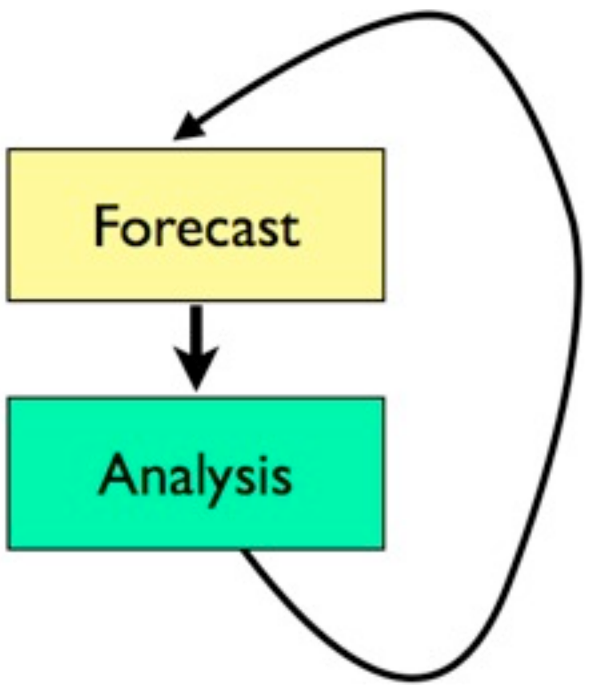
Approach, Phase I

Lagrangian Data Assimilation

Ensemble Kalman Filter (EnKF), Analysis

Stochastic: Generate ensemble of observations and perform Kalman-like analysis on combined ensemble

Deterministic: Square root filters



Approach, Phase I

Lagrangian Data Assimilation

Ensemble Kalman Filter (EnKF)

Better captures nonlinearity (in forecast)
(up to ensemble approximation)

Still assumes Gaussianity (in analysis)
NOT valid for nonlinear systems

Approach, Phase I

Lagrangian Data Assimilation

Particle Filter

Approximate pdf by an ensemble of *weighted* particles:

$$\{\mathbf{x}_i^f(t)\}_{i=1}^N \quad \{w_{i,k}\}_{i=1}^N$$

Forecast: Evolve particles forward using SDE

Analysis: Perform full Bayesian update at analysis

Approach, Phase I

Lagrangian Data Assimilation

Particle Filter, Analysis

$$\mathbf{y}_k^o = h(\mathbf{x}^t(t_k)) + \epsilon_k^t$$

$\mathbf{y}_k^o$ – observation of system at time t_k

ϵ_k^t – Gaussian noise

$h(\cdot)$– observation operator

Approach, Phase I

Lagrangian Data Assimilation

Particle Filter, Analysis

Likelihood of Observation:

$$p(\mathbf{y}_k^o|\mathbf{x}^f(t_k)) =$$

$$C_1 \exp\left(-\frac{1}{2}(\mathbf{y}_k^o - h(\mathbf{x}^f(t_k)))^T(\mathbf{R}^o)^{-1}(\mathbf{y}_k^o - h(\mathbf{x}^f(t_k)))\right)$$

Approach, Phase I

Lagrangian Data Assimilation

Particle Filter, Analysis

Likelihood of Observation:

$$p(y|x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y - h(x))^2\right)$$

Approach, Phase I

Lagrangian Data Assimilation

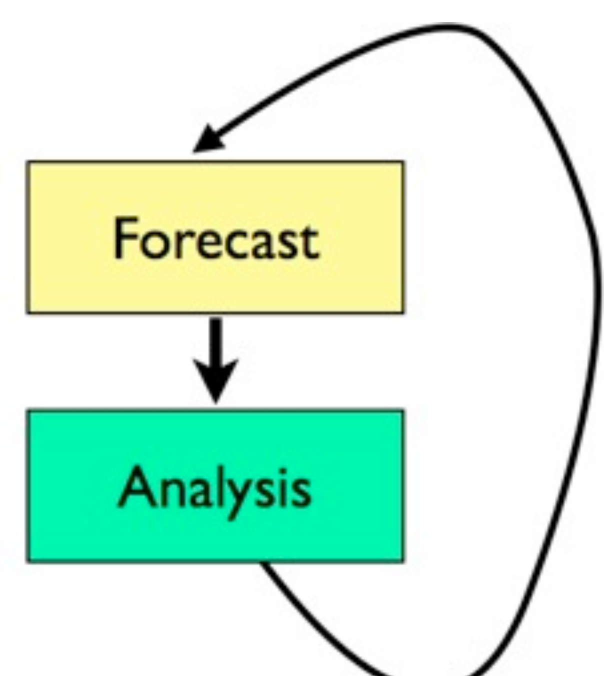
Particle Filter, Analysis

Weight of particle *i* at analysis step *k*:

$$w_{i,k}$$

Bayesian Update:

$$w_{i,k} = C_2 w_{i,k-1} p(\mathbf{y}_k^o | \mathbf{x}_i^f(t_k)))$$



Approach, Phase I

Lagrangian Data Assimilation

Particle Filter

Better captures nonlinearity (up to ensemble approximation)

Does not assume Gaussianity on posterior or prior

Approach, Phase I

Lagrangian Data Assimilation

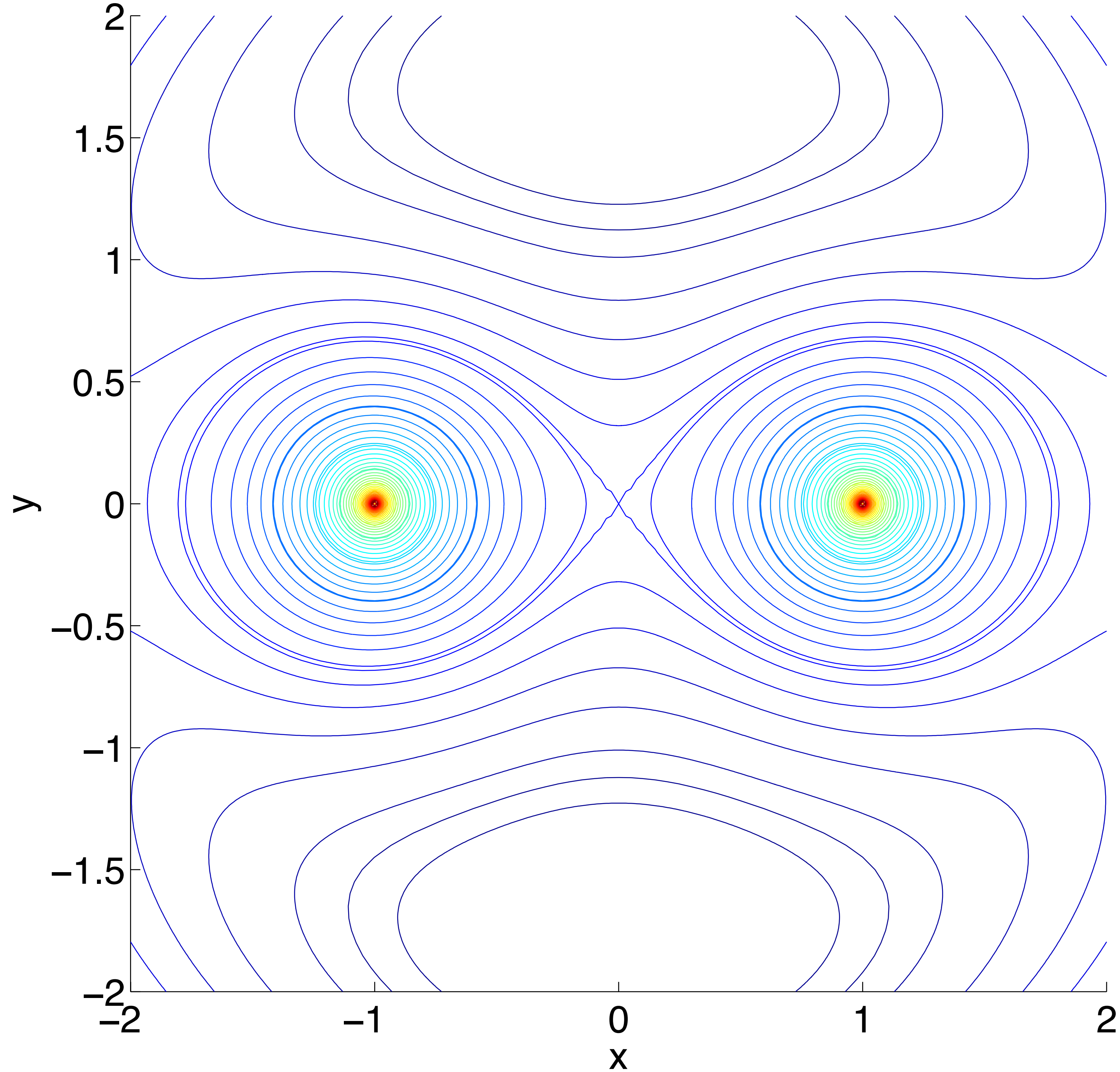
Particle Filter

BUT

Requires a large number of particles

For large, nonlinear systems, requires frequent resampling

Phase II



Approach, Phase II

Manifold Detection for Observing System Design

Stream functions

$$\psi(x, y) = C$$

Consider the trajectories in *phase space*

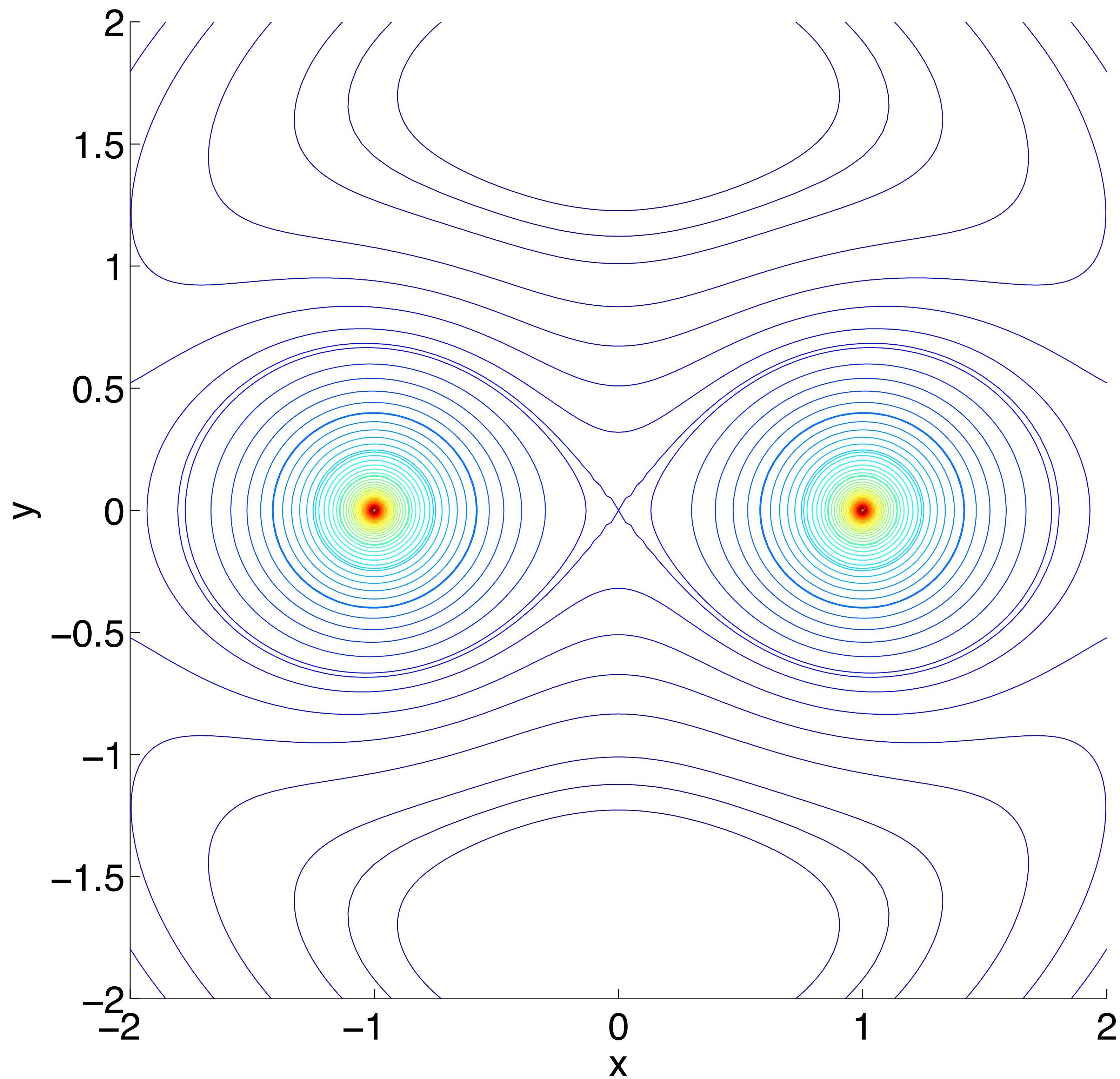
Trajectories will lie along the level curves of the stream function for steady flows

'Streamlines'

Trajectories may not cross streamlines

Uniqueness

Stream function in corotating frame



Approach, Phase II

Manifold Detection for Observing System Design

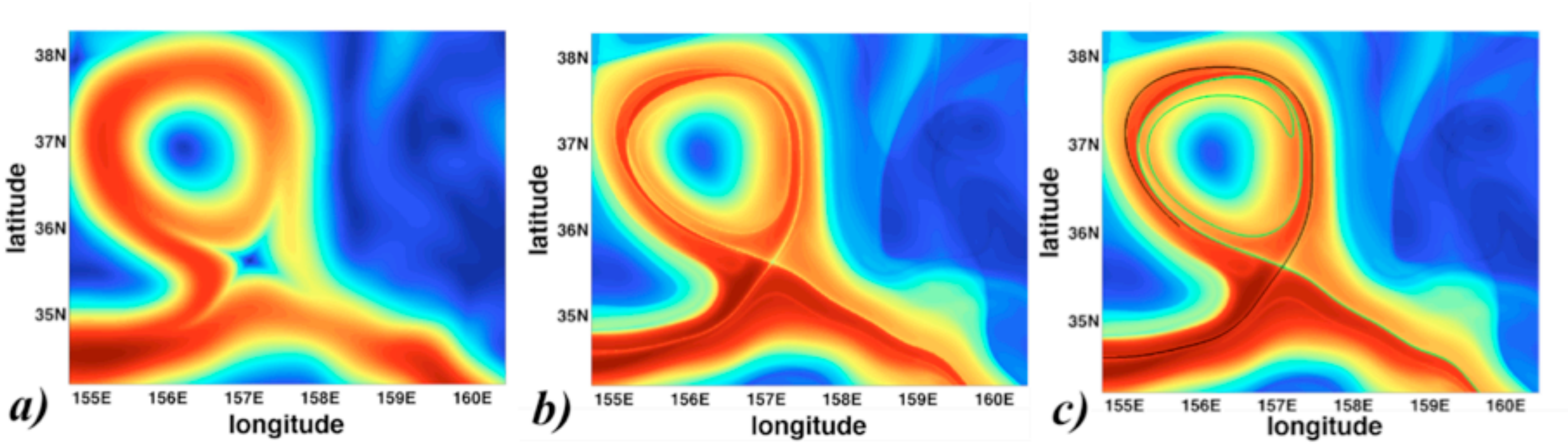
Mendoza and Mancho's Lagrangian Descriptor *M*

$$M(\mathbf{x}_D^t, t^*) = \int_{t^*-\tau}^{t^*+\tau} \left(\sum_{i=1}^{n} \left(\frac{dx_D^i(t)}{dt} \right)^2 \right)^{1/2} dt$$

Approach, Phase II

Manifold Detection for Observing System Design

Mendoza and Mancho's Lagrangian Descriptor *M*



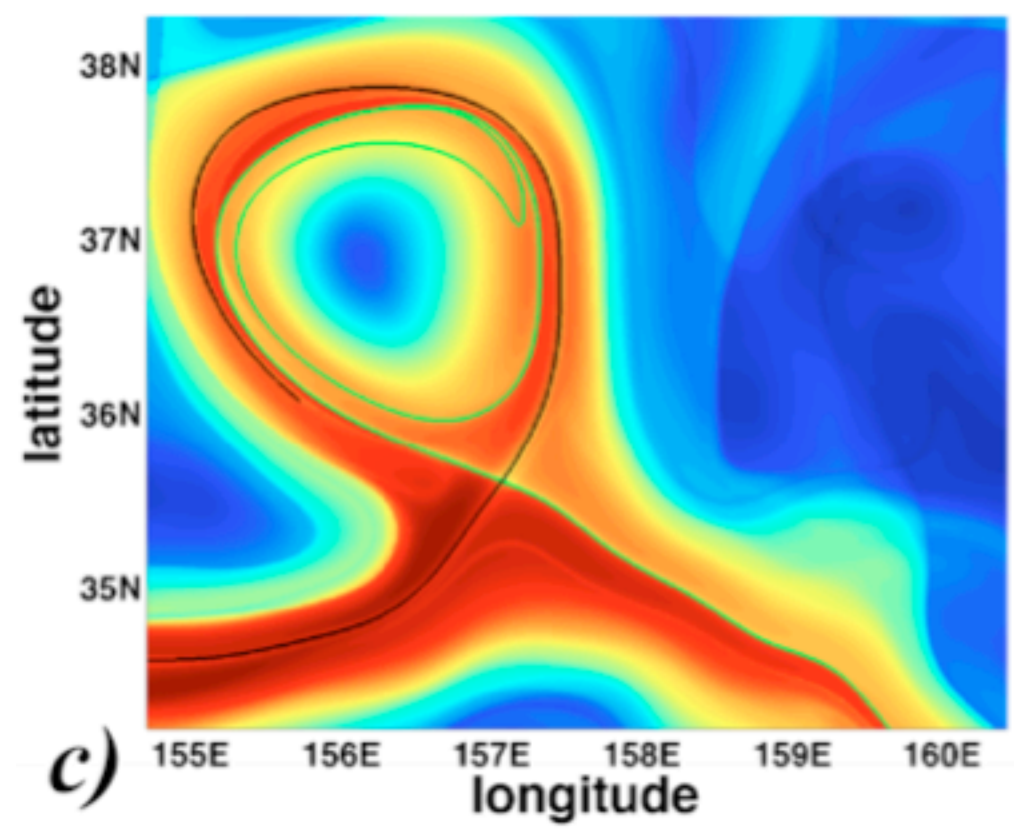
C. Mendoza and A.M. Mancho. Hidden geometry of ocean flows. *Physical review letters*, 105(3):38501, 2010.

Approach, Phase II

Manifold Detection for Observing System Design

Mendoza and Mancho's Lagrangian Descriptor *M*

Improve Phase I by using knowledge of dynamics



Implementation

Develop serial code for EKF, EnKF, and Particle Filter in MATLAB on MacBook Pro

Parallelize the EnKF and particle filters

Parallelize computation of M

Use MATLAB Parallel Computing Toolbox

Databases

Phase I

Archived numerical solutions to point-vortex model

Generate using stochastic 4th-order Runge-Kutta method

Phase II

No databases: completely model dependent

Validation, Phase I

Validate filter by varying noise

Three stages

Stage 1:

Vortices' positions known, low noise

Stage 2:

Vortices' positions unknown, low noise

Stage 3:

Vortices' positions unknown, realistic noise

Validation, Phase I

Validate filter by varying noise

Validation Metric

$$\delta^{a,f}(t) = ||\mathbf{x}^t(t) - \mathbf{x}^{a,f}(t)||_2$$

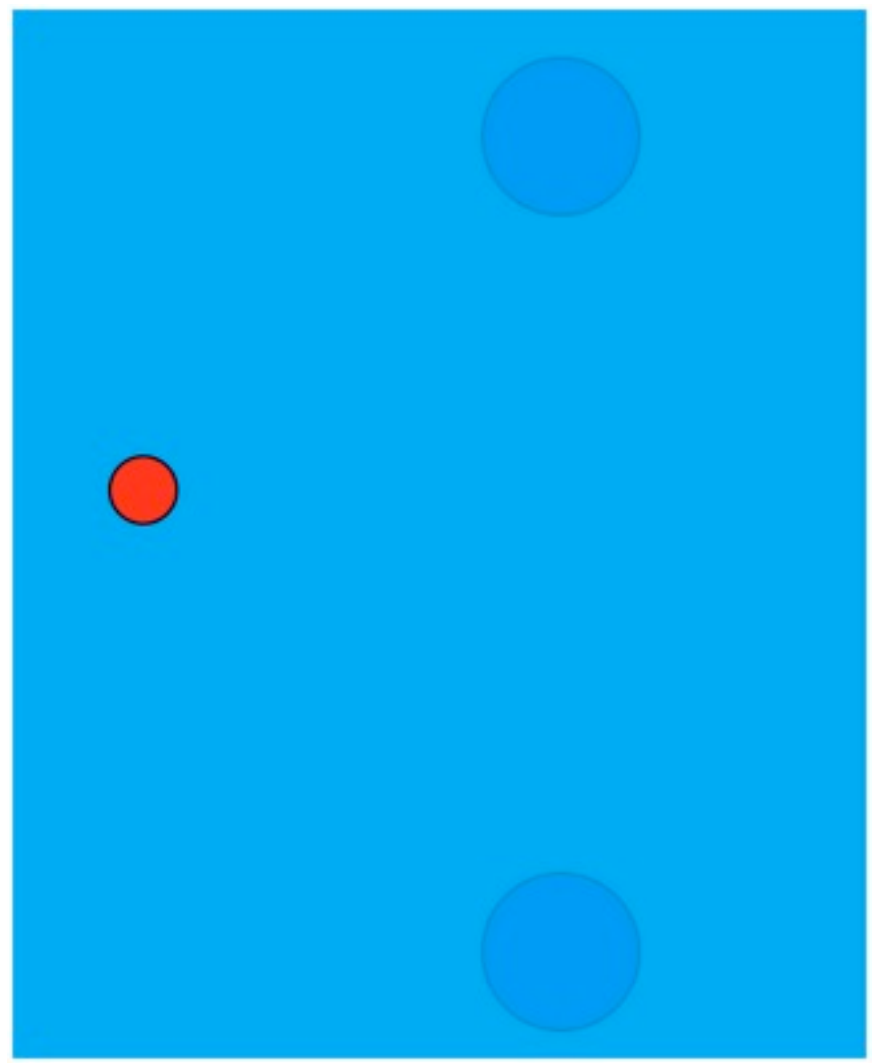
Validation, Phase I

Validate filter by comparison

Compare to published studies with

$N_v = 2$

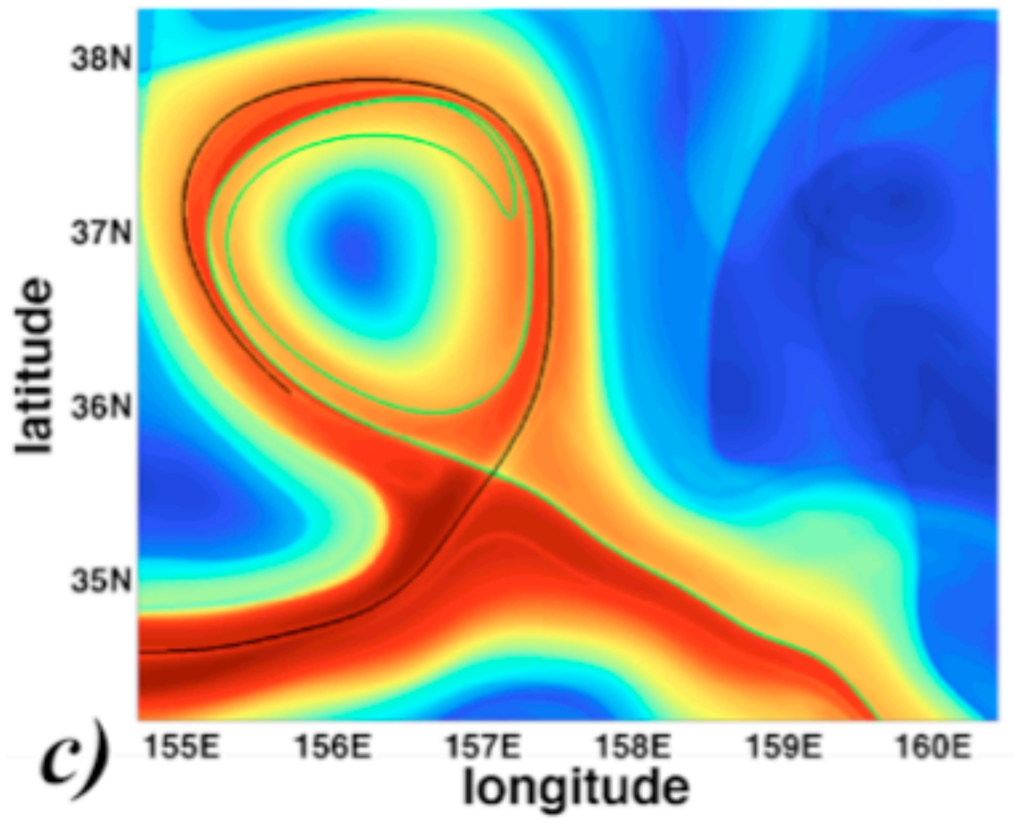
$N_d = 1$



Validation, Phase II

Validate *M* by comparison

- Compare *M* to analytically known stream function
- Compare *M* to finite-time Lyapunov exponents



Testing, Phase I

Failure statistic

Compare EKF, EnKF, and particle filter across databases using

$$\hat{f}_{\delta_{div},n}(t) =$$

fraction of times $\delta(t) > \delta_{div}$ at time t in n trials

Testing, Phase II

Failure statistic

Compare EKF, EnKF, and particle filter, including decision of initial drifter position

$$\hat{f}_{\delta_{div},n}(t, \mathbf{x}_D(0)) =$$

fraction of times $\delta(t) > \delta_{div}$ at time t in n trials

Project Schedule and Milestones

- Phase I
  - Produce database: now through mid-October
  - Develop extended Kalman Filter: now through mid-October
  - Develop ensemble Kalman Filter: mid-October through mid-November
  - Develop particle filter: mid-November through end of January
  - Validation and testing of three filters (serial): Beginning in mid-October, complete by February
  - Parallelize ensemble methods: mid-January through March
- Phase II
  - Develop serial code for manifold detection: mid-January through mid-February
  - Validate and test manifold detection: mid-February through mid-March
  - Parallelize manifold detection algorithm: mid-March through mid-April

Project Schedule and Milestones

- Phase I
  - Complete validation and testing of extended Kalman filter: beginning of November
  - Complete validation and testing of (serial) ensemble Kalman filter: beginning of December
  - Complete validation and testing of (serial) particle filter: end of January
- Phase II
  - Complete validation and testing of (serial) manifold detection: mid-March
  - Parallelize ensemble methods: beginning of April
  - Parallelize manifold detection algorithm: end of April

Deliverables

Database of sample trajectories used for validation and testing

Suite of parallelized software for performing EKF, EnKF, and particle filtering for point-vortex model

Software to compute *M* for point-vortex model

References (See Proposal for More)

A.H. Jazwinski. *Stochastic processes and filtering theory*. Dover Publications, 2007.

G. Evensen. *Data assimilation: the ensemble Kalman filter.* Springer Verlag, 2009.

A. Doucet, N. De Freitas, and N. Gordon. *Sequential Monte Carlo methods in practice.* Springer Verlag, 2001.

C. Mendoza and A.M. Mancho. Hidden geometry of ocean flows. *Physical review letters*, 105(3):38501, 2010.

Questions???