Sequential Monte Carlo (SMC) Approaches for Ocean Biology: Our Decade of Experimentation*

Mike Dowd
Dept of Mathematics & Statistics
Dept of Oceanography
Dalhousie University, Halifax, Canada

Katja Fennel (Dal), Paul Mattern (UC SC), Jon Briggs (Auckland), Greg Britten (MIT)

DALHOUSIE UNIVERSITY
Inspiring Minds
Faculty of Science

BIRS Workshop, Nov 12-16, 2018

* Lower trophic level biology – the planktonic ecosystem.

An Acknowledgement to Weather Forecasting

Numerical Weather Prediction pioneered large-scale estimation for time dependent systems based on dynamic / numerical models.

1960s: Optimal Interpolation (Lev Gandin):
- The data assimilation cycle – sequential estimation via approximate Kalman filters

1980s: Variational Data Assimilation (Olivier Talagrand)
- online optimization using numerical weather models as an IVP, needs adjoints need for gradient

2000s: Ensemble Kalman filter (Geir Evensen)
- Modular, sample based, and incorporates dynamical model error

Clear Performance Metric: Forecast Skill

Statistical Framework for Biological Data Assimilation: State Space Model

$$\boxed{\begin{aligned} x_t &= d(x_{t-1}, \theta_1, e_t) \\ y_t &= h(x_t, \theta_2, v_t) \end{aligned}}$$

or

$$\boxed{\begin{aligned} x_t &\sim p(x_t, \theta_1 \mid x_{t-1}) \\ y_t &\sim p(y_t, \theta_2 \mid x_t) \end{aligned}}$$

← DYNAMICS

← OBSERVATIONS

$t = 1, \ldots, T$

Statistical Framework for Biological Data Assimilation: State Space Model

$$x_t = d(x_{t-1}, \theta_1, e_t)$$
$$y_t = h(x_t, \theta_2, v_t)$$

or

$$x_t \sim p(x_t, \theta_1 \mid x_{t-1})$$
$$y_t \sim p(y_t, \theta_2 \mid x_t)$$

← DYNAMICS

← OBSERVATIONS

$t = 1, \ldots, T$

Main Goals: Given observations, y_t, we estimate:

- x_t : the state (filtering for online prediction, smoothing for retrospective analysis)

- θ's: parameters (for scientific understanding)

Statistical Framework for Biological Data Assimilation: State Space Model

$$x_t = d(x_{t-1}, \theta_1, e_t)$$
$$y_t = h(x_t, \theta_2, v_t)$$

or

$$x_t \sim p(x_t, \theta_1 \mid x_{t-1})$$
$$y_t \sim p(y_t, \theta_2 \mid x_t)$$

DYNAMICS

OBSERVATIONS

$t = 1, \ldots, T$

Main Goals: Given observations, y_t, we estimate:

Often done separately

- x_t : the state (filtering for online prediction, smoothing for retrospective analysis)
- θ's: parameters (for scientific understanding)

Features of Biological Data Assimilation

Dynamics are highly informative:

- Numerical ocean models considered a good representation of reality. Encapsulate accumulated scientific knowledge

True for the physics aspects, less so for biology

Features of Biological Data Assimilation

Dynamics are highly informative:

- Numerical ocean models considered a good representation of reality. Encapsulate accumulated scientific knowledge

True for the physics aspects, less so for biology

Data Poor:

- Relies on opportunistic sampling
- A partially observable system

But this is changing .. An observing revolution is well underway

State Space Model: Dynamics

$$x_t = d(x_{t-1}, \theta, e_t)$$

or

$$x_t \sim p(x_t, \theta \mid x_{t-1})$$

DYNAMICS

[redacted]

$t = 1, \ldots, T$

State Space Model: Dynamics

A simulator is available

$$x_t = d(x_{t-1}, \theta, e_t)$$

or

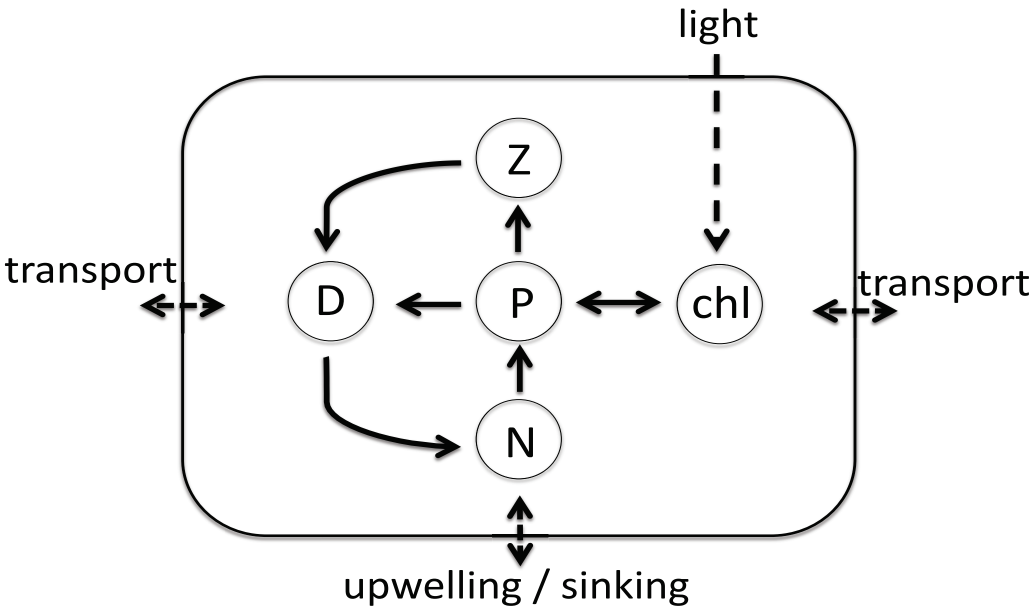
$$x_t \sim p(x_t, \theta \mid x_{t-1})$$

DYNAMICS

[redacted]

$t = 1, \ldots, T$

Basic Biology: The PZND Model Class

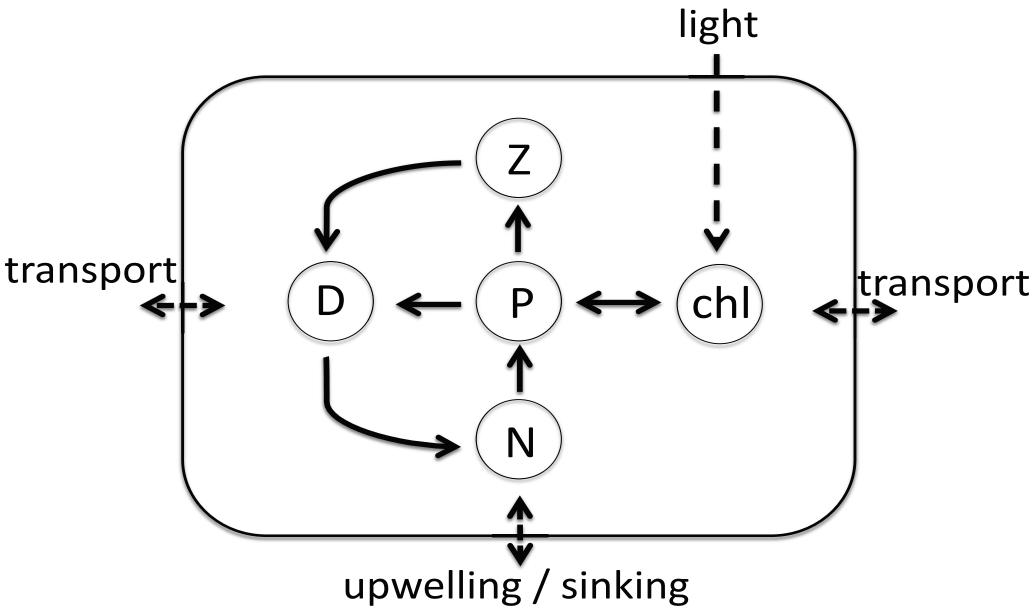


P : phytoplankton
Z: zooplankton
N: nutrients
D: detritus
chl: chlorophyll

- Systems of ODEs: (complex dynamics for biology)

$$\frac{dX}{dt} = f(X,\theta) + e(t)$$

Basic Biology: The PZND Model Class



P : phytoplankton
Z: zooplankton
N: nutrients
D: detritus
chl: chlorophyll

- Systems of ODEs: (complex dynamics for biology)

$$\frac{dX}{dt} = f(X,\theta) + e(t)$$

Role of these simple models?: Test beds for prototyping methods. But Little utility in mimicking and understanding reality (why? physical control of fluid system)

Coupling Biology to Physics

Biology:

$$\boxed{\frac{\partial X_i}{\partial t} + \vec{u} \cdot \nabla X_i - K \nabla^2 X_i = f(x_1, \ldots, x_m) + e_i}$$

Tracer Equations: Biology in a fluid flow

Coupling Biology to Physics

Biology:

$$\boxed{\frac{\partial X_i}{\partial t} + \vec{u} \cdot \nabla X_i - K \nabla^2 X_i = f(x_1, \ldots, x_m) + e_i}$$

Physics:

$$\frac{\partial \vec{u}_h}{\partial t} + \vec{u}_h \cdot \nabla \vec{u}_h + \vec{\Omega} \times \vec{u}_h = \frac{1}{\rho_0} \nabla P + K_h \nabla^2 \vec{u}_h + e_m$$

$$\nabla \cdot \vec{u}_h + \frac{\partial w}{\partial z} = 0; \quad \frac{\partial \rho}{\partial z} = -\rho g; \quad \rho = \rho(T, S, z)$$

$$\frac{\partial T}{\partial t} + \vec{u} \cdot \nabla T - K \nabla^2 T = e_T$$

$$\frac{\partial S}{\partial t} + \vec{u} \cdot \nabla S - K \nabla^2 S = e_S$$

Coupling Biology to Physics

Biology:

$$\boxed{\frac{\partial X_i}{\partial t} + \vec{u} \cdot \nabla X_i - K \nabla^2 X_i = f(x_1, \ldots, x_m) + e_i}$$

Physics:

$$\frac{\partial \vec{u}_h}{\partial t} + \vec{u}_h \cdot \nabla \vec{u}_h + \vec{\Omega} \times \vec{u}_h = \frac{1}{\rho_0} \nabla P + K_h \nabla^2 \vec{u}_h + e_m$$

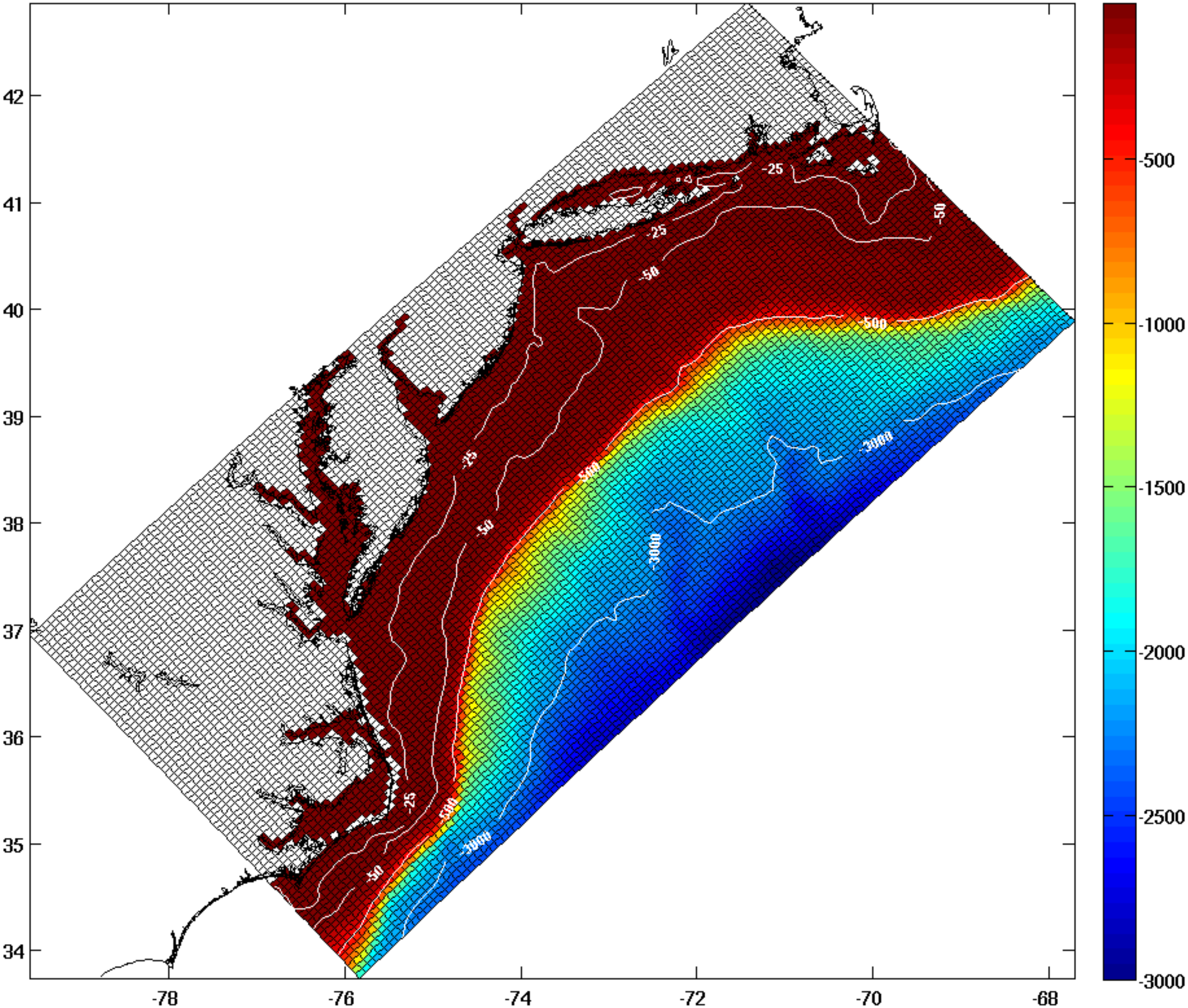
$$\nabla \cdot \vec{u}_h + \frac{\partial w}{\partial z} = 0; \quad \frac{\partial \rho}{\partial z} = -\rho g; \quad \rho = \rho(T, S, z)$$

$$\frac{\partial T}{\partial t} + \vec{u} \cdot \nabla T - K \nabla^2 T = e_T$$

$$\frac{\partial S}{\partial t} + \vec{u} \cdot \nabla S - K \nabla^2 S = e_S$$

- Numerical integration requires short time steps and fine spatial resolution (= computationally costly).
- Community ocean models available …

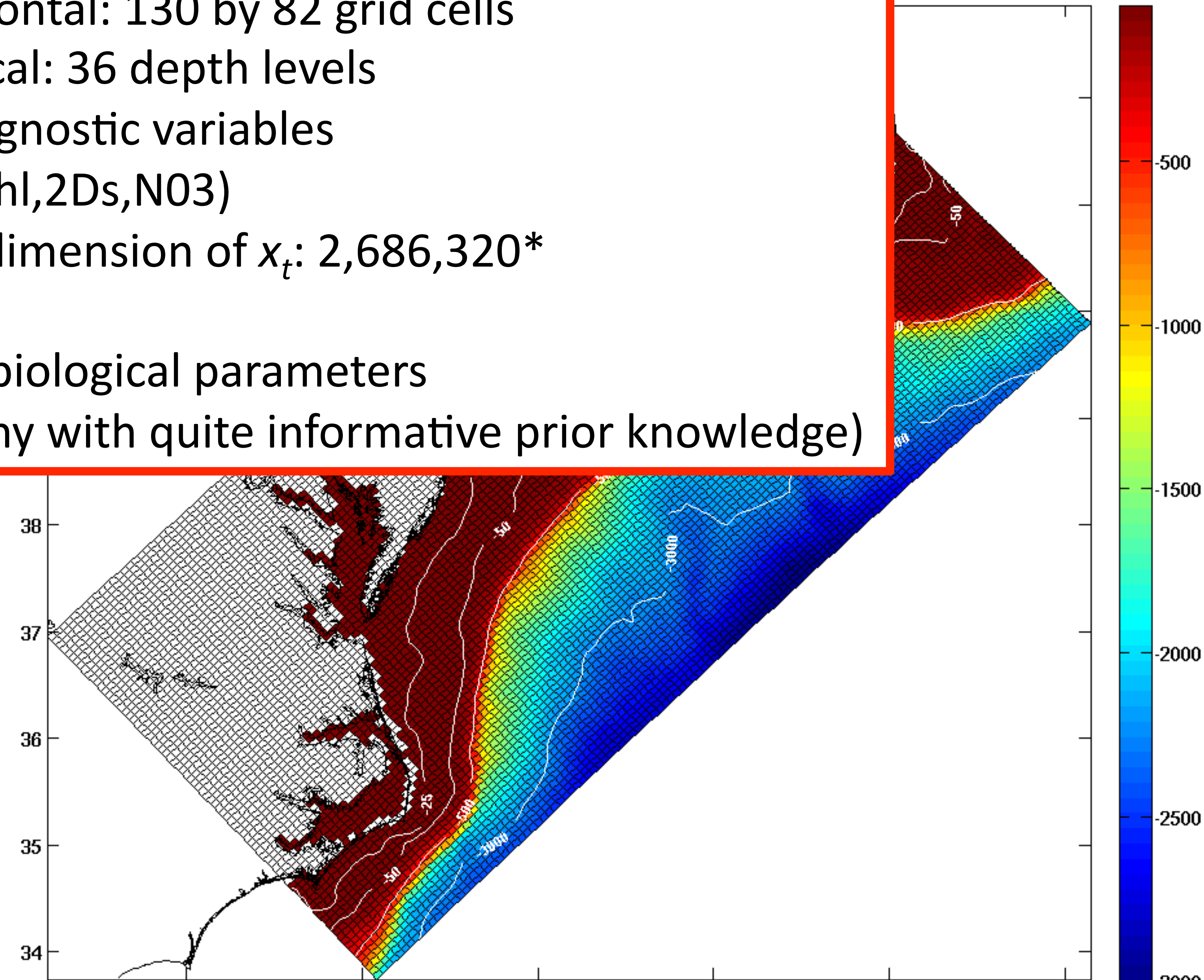
ROMS ESPRESSO bathymetry and grid



- Horizontal: 130 by 82 grid cells
- Vertical: 36 depth levels
- 7 prognostic variables (P,Z,chl,2Ds,N03)

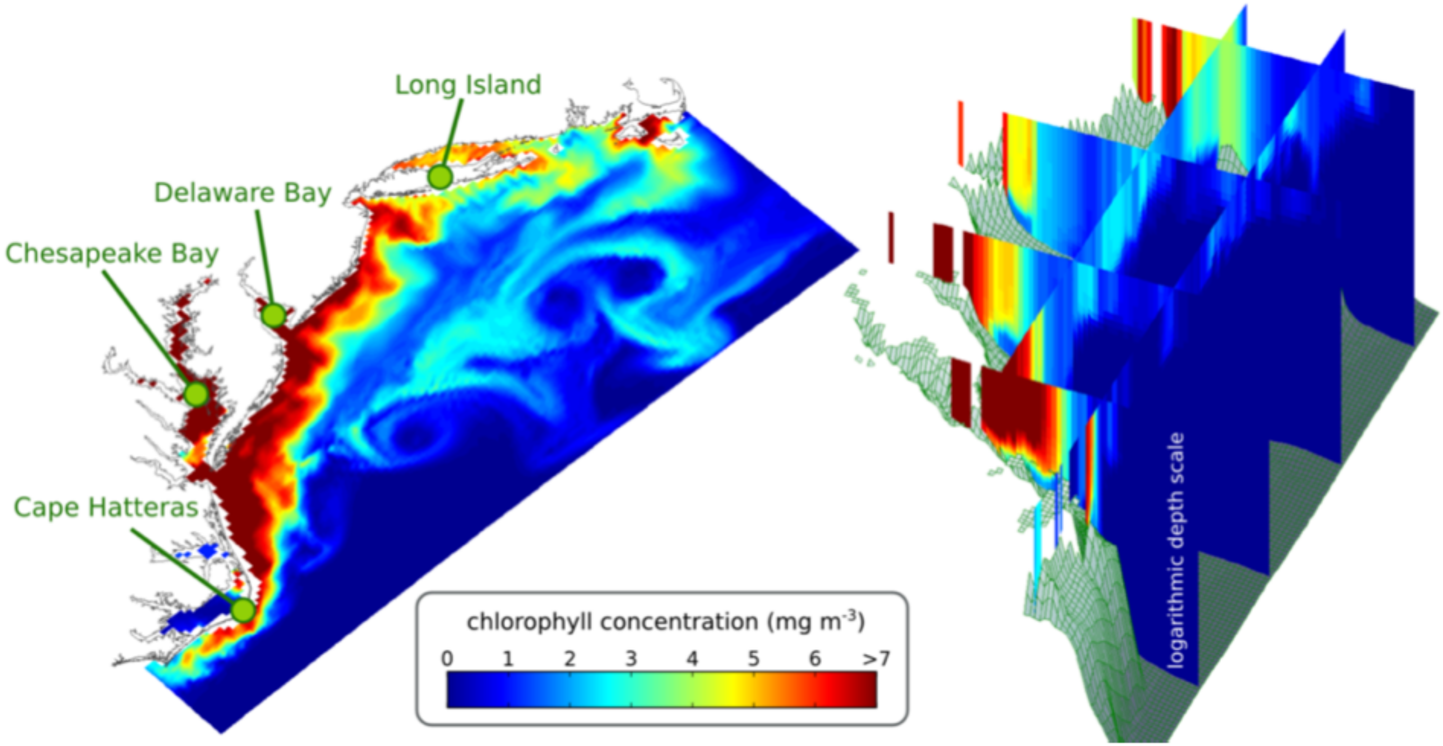
= state dimension of x_t: 2,686,320*

- ~30 biological parameters (many with quite informative prior knowledge)



*effective d.o.f. a lot smaller – spatial correlation, and variable inter-dependence

A snapshot of model predicted ocean chlorophyll



Approaches to Stochastic Simulation

… for dynamical models that are essentially formulated as deterministic ones

1. *Add Dynamical noise*:

2. *Random Initial and Boundary Conditions:*

3. *Use Multi-model Ensembles*

4. *Stochastic parameters*:

*Approaches to Stochastic Simulation**

… for dynamical models that are essentially formulated as deterministic ones

1. *Add Dynamical noise*:

2. *Random Initial and Boundary Conditions:*

3. *Use Multi-model Ensembles*

4. *Stochastic parameters*:

State Space Model: Observations

[redacted]

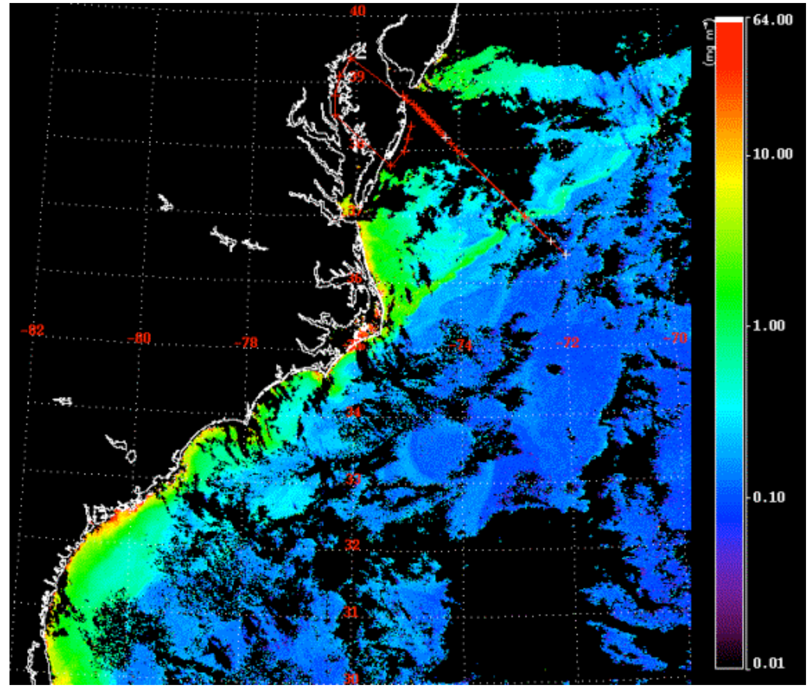
$$y_t = h(x_t, \phi, v_t)$$

$$y_t \sim p(y_t, \phi \mid x_t)$$

OBSERVATIONS

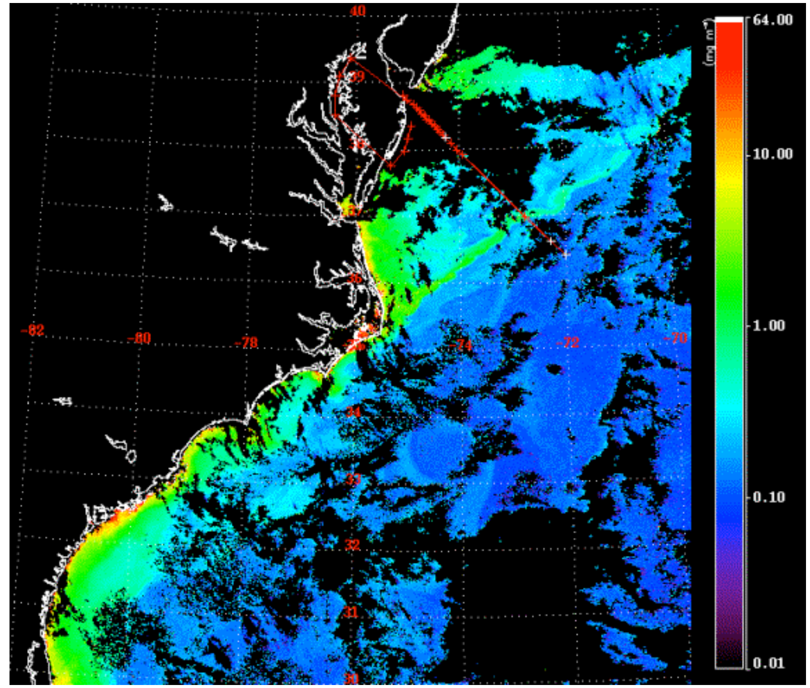
$$t = 1, \ldots, T$$

OBSERVATIONS



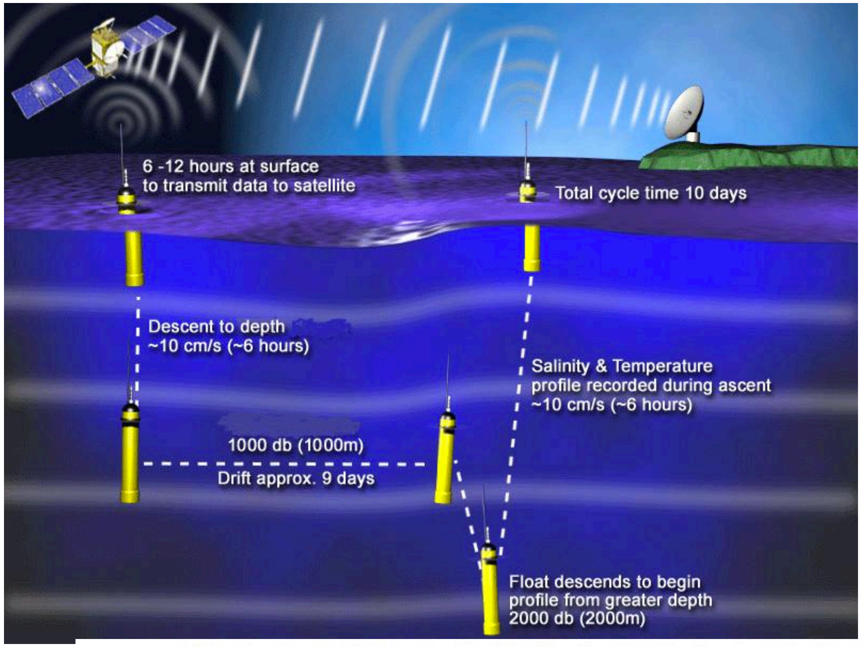
Traditional Observations: vertical profiles, time series, and ***satellite chlorphyll imagery***

OBSERVATIONS

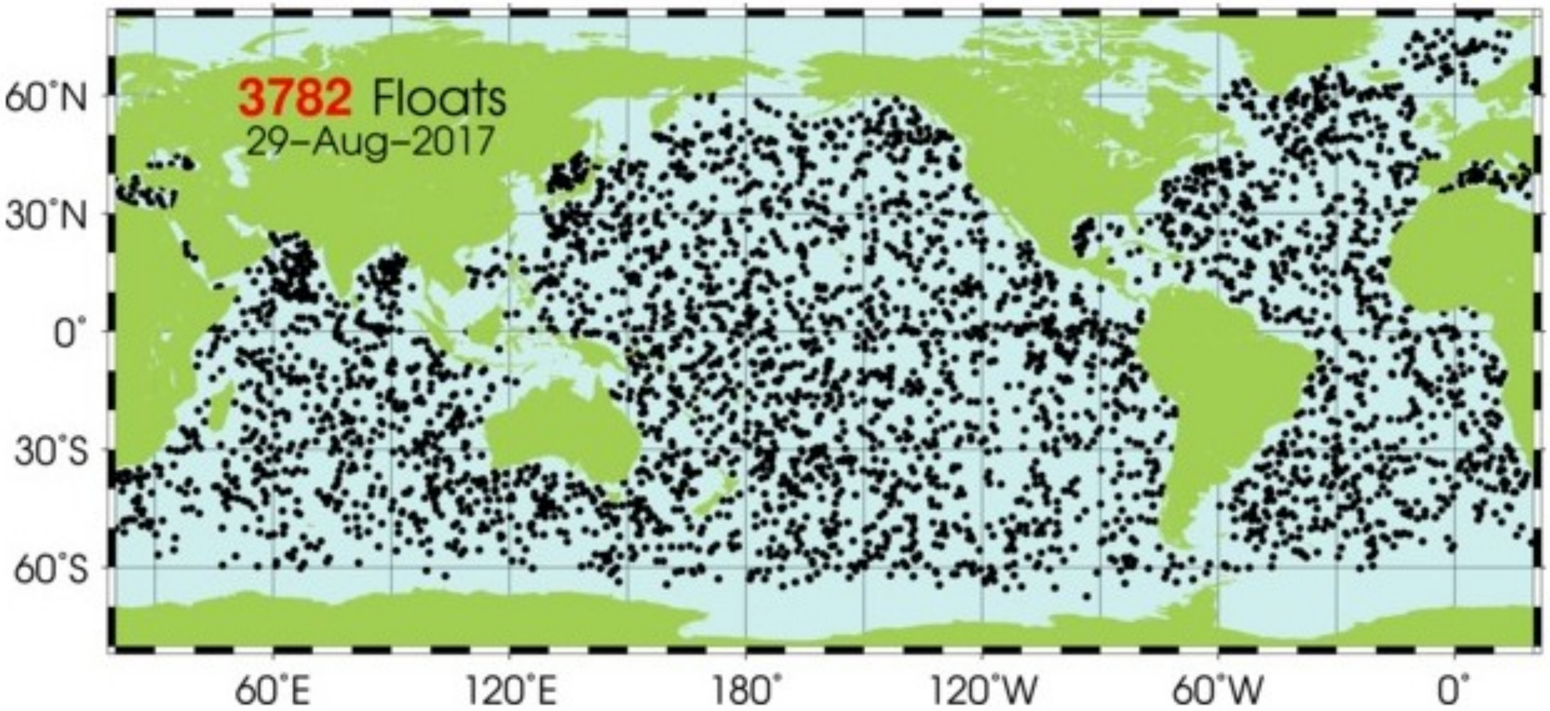


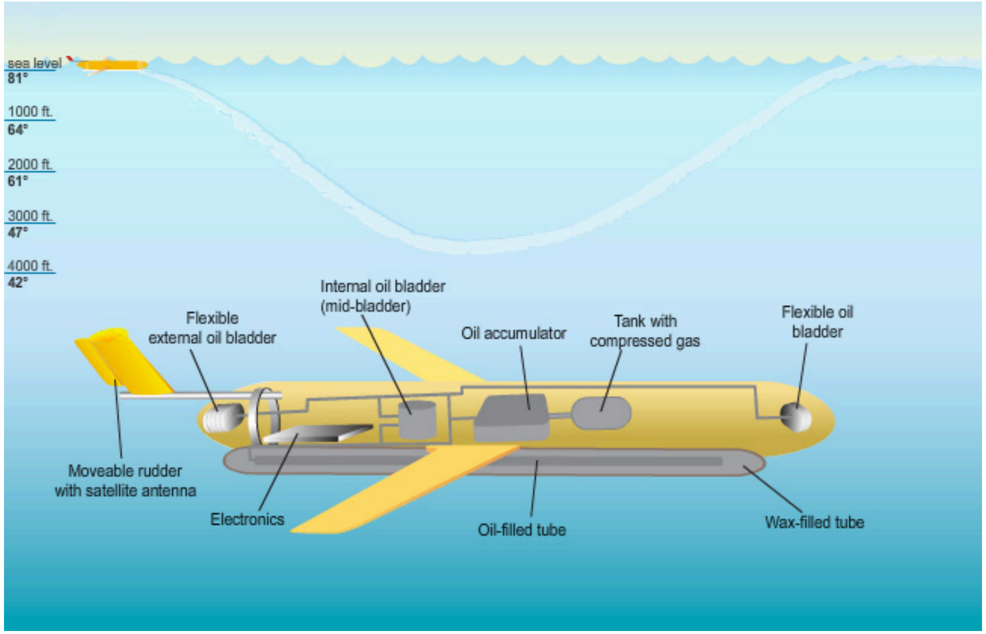
Traditional Observations: vertical profiles, time series, and ***satellite chlorphyll imagery***

New observations: complex spatio-temporal multivariable sampling via autonomous robotic sampling platforms

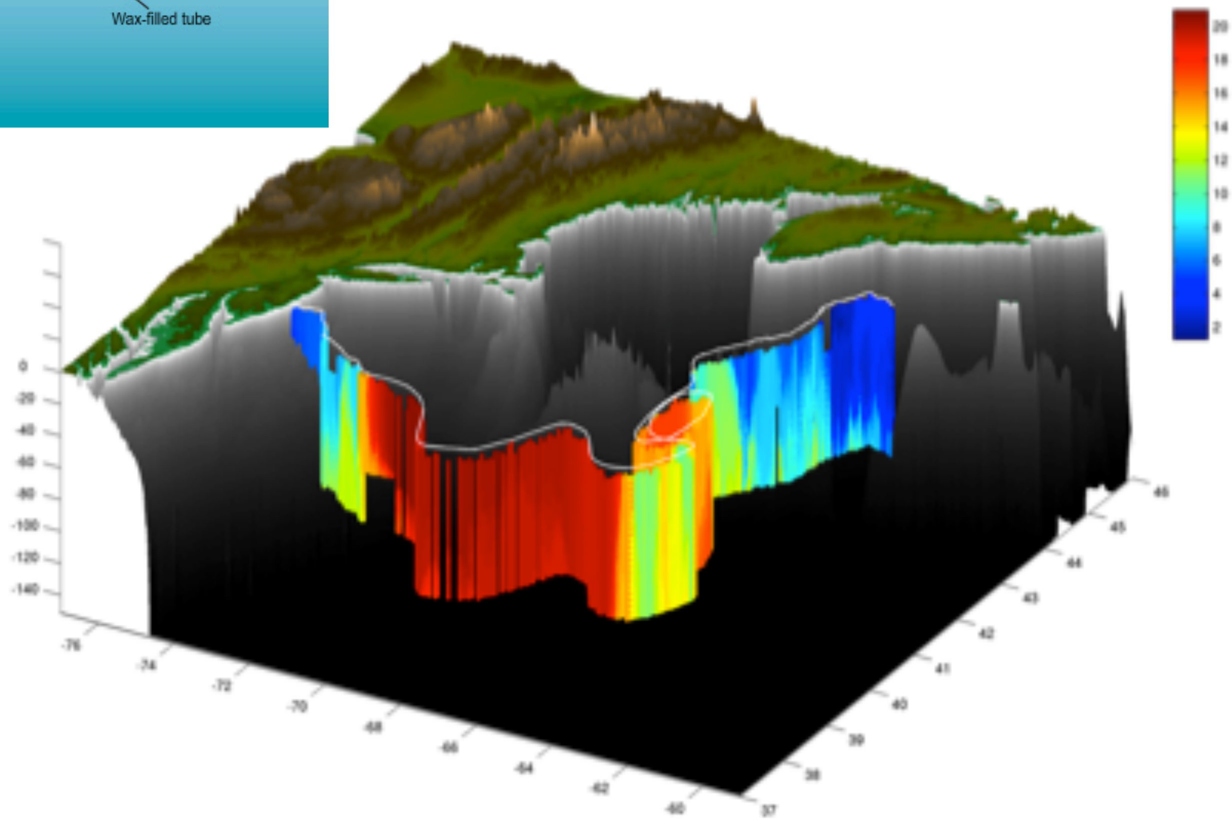

6 -12 hours at surface
to transmit data to satellite
Total cycle time 10 days
Descent to depth
~10 cm/s (~6 hours)
Salinity & Temperature
profile recorded during ascent
~10 cm/s (~6 hours)
1000 db (1000m)
Drift approx. 9 days
Float descends to begin
profile from greater depth
2000 db (2000m)

ARGO Floats


3782 Floats
29-Aug-2017
60°N
30°N
0°
30°S
60°S
60°E
120°E
180°
120°W
60°W
0°



Ocean Gliders



Keep in Mind … the General Probabilistic Solution

The Hierarchical Bayesian Model:

$$\boxed{p(x_{1:T},\theta \mid y_{1:T}) \propto p(y_{1:T} \mid x_{1:T},\theta)\cdot p(x_{1:T} \mid \theta)\cdot p(\theta)}$$

where:

$x_{1:T} = (x_1,\ldots,x_T)$ is the system state

$y_{1:T} = (y_1,\ldots,y_T)$ are the observations

θ are the dynamical model parameters

- $p(x_{1:T},\theta \mid y_{1:T})$ is our target distribution
- $p(y_{1:T} \mid x_{1:T},\theta)$ is the conditional measurement distribution
- $p(x_{1:T} \mid \theta)$ is indentified with the numerical ocean model
- $p(\theta)$ is any prior information (expert knowledge)

Rely on sampling based solutions in practice →

A General Computational Solution: Particle MCMC

for k = 1 to m

Generate candidate $\theta^* = \theta^{(k)} + \varepsilon$

Run particle filter to determine $\{x^*_{t|t}\}$ for θ^*

Evaluate likelihood $L(\theta \mid y_{1:T}) \propto \prod_{t=1}^{T} \left(\sum_{k=1}^{n} p(y_t \mid x^*_{t|t}) \right)$

Do Metropolis-Hastings accept/reject step

Compute the acceptance probability: $\alpha = \dfrac{L(\theta^* \mid y_{1:T})}{L(\theta^{k-1} \mid y_{1:T})}$

Draw $u \sim U(0,1)$

If $\min(1,\alpha) \geq u$ then $\theta^k = \theta^*$,

else $\theta^k = \theta^{(k-1)}$

end (for k)

$\longrightarrow$ yields sample drawn from target $p(x_{1:T}, \theta \mid y_{1:T})$

Computational Solution: Particle MCMC

for k = 1 to m

Generate candidate $\theta^* = \theta^{(k)} + \varepsilon$

Run particle filter to determine $\{x^*_{t|t}\}$ for θ^*

Evaluate likelihood $L(\theta \mid y_{1:T}) \propto \prod_{t=1}^{T} \left(\sum_{k=1}^{n} p(y_t \mid x^*_{t|t}) \right)$

Do Metropolis-Hastings accept/reject step

Compute the acceptance probability: $\alpha = \dfrac{L(\theta^* \mid y_{1:T})}{L(\theta^{k-1} \mid y_{1:T})}$

Draw $u \sim U(0,1)$

If $\min(1,\alpha) \geq u$ then $\theta^k = \theta^*$,

else $\theta^k = \theta^{(k-1)}$

end (for k)

Computational Solution: Particle MCMC

for k = 1 to m

Generate candidate $\theta^* = \theta^{(k)} + \varepsilon$

Run particle filter to determine $\{x^*_{t|t}\}$ for θ^*

Evaluate likelihood $L(\theta \mid y_{1:T}) \propto \prod_{t=1}^{T} \left(\sum_{k=1}^{n} p(y_t \mid x^*_{t|t}) \right)$

Do Metropolis-Hastings accept/reject step

Compute the acceptance probability: $\alpha = \dfrac{L(\theta^* \mid y_{1:T})}{L(\theta^{k-1} \mid y_{1:T})}$

Draw $u \sim U(0,1)$

If $\min(1,\alpha) \geq u$ then $\theta^k = \theta^*$,

else $\theta^k = \theta^{(k-1)}$

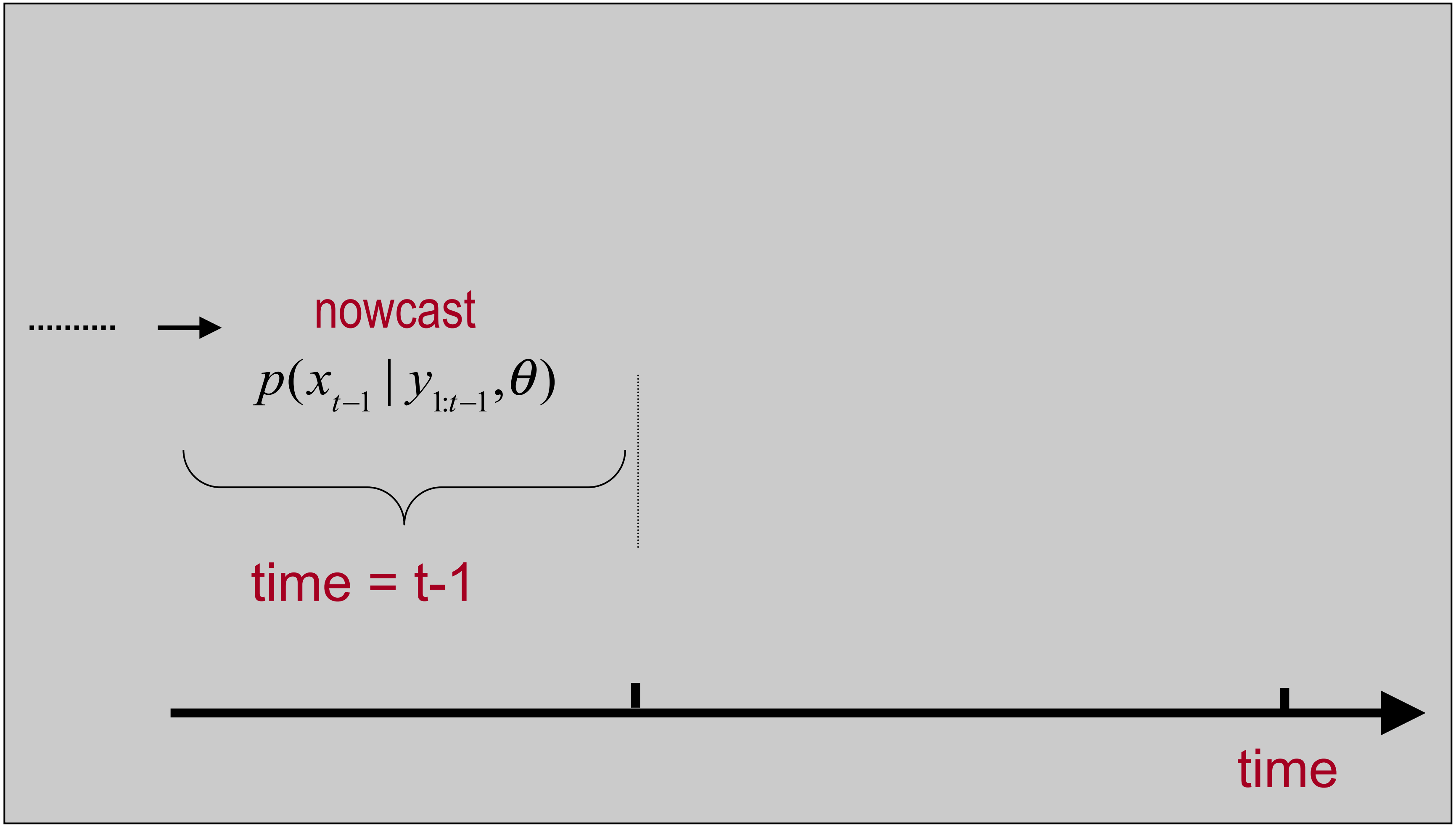
end (for k)

State estimation* via the particle filter (or SMC) is the “engine” for sample based estimation

* Focus here on estimating state, but can use to estimate parameters via sample based likelihoods or state augmentation

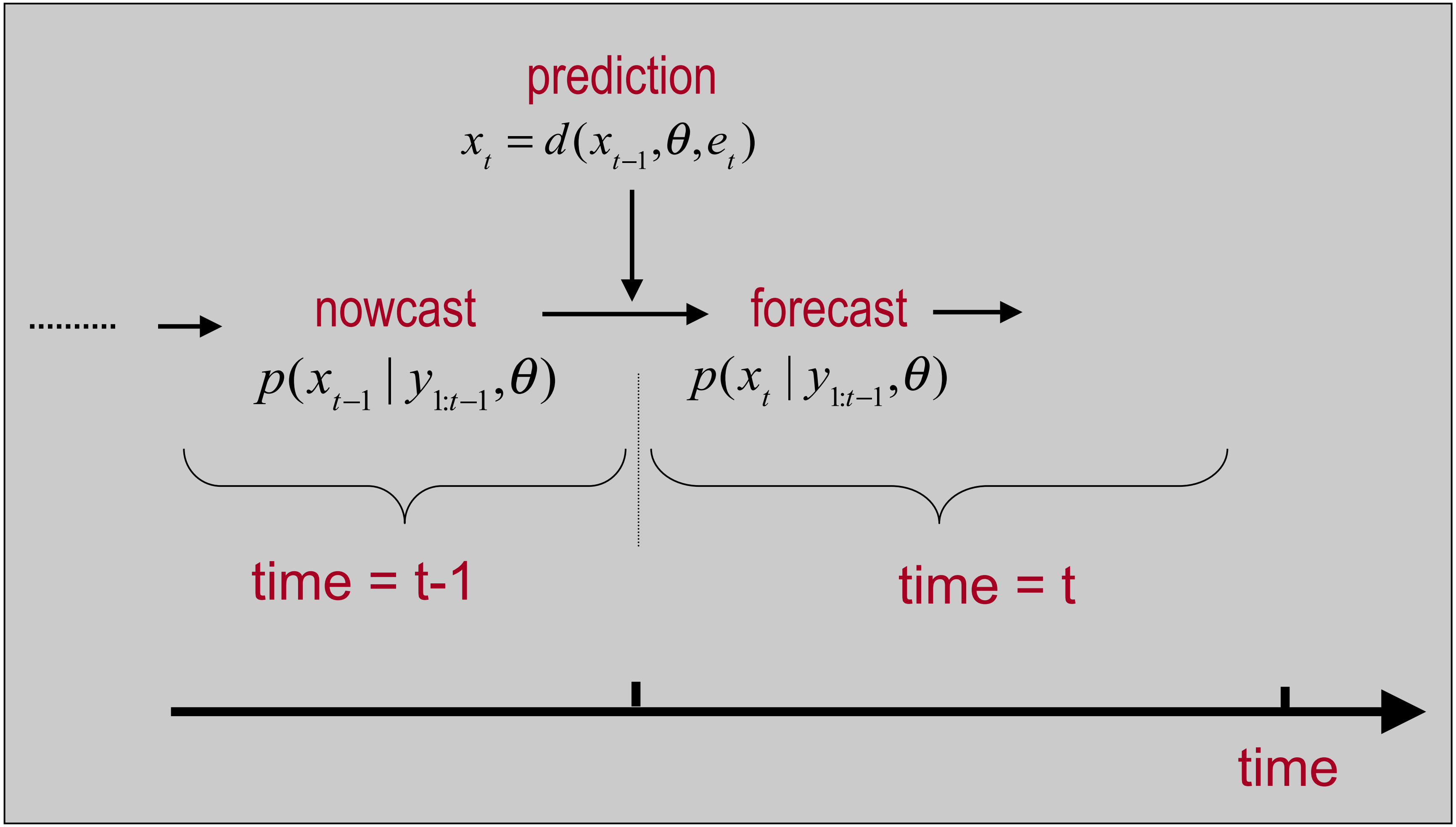
Particle Filter Schematic: Sequential State Estimation

Single stage transition of system from time *t-1* to time *t*



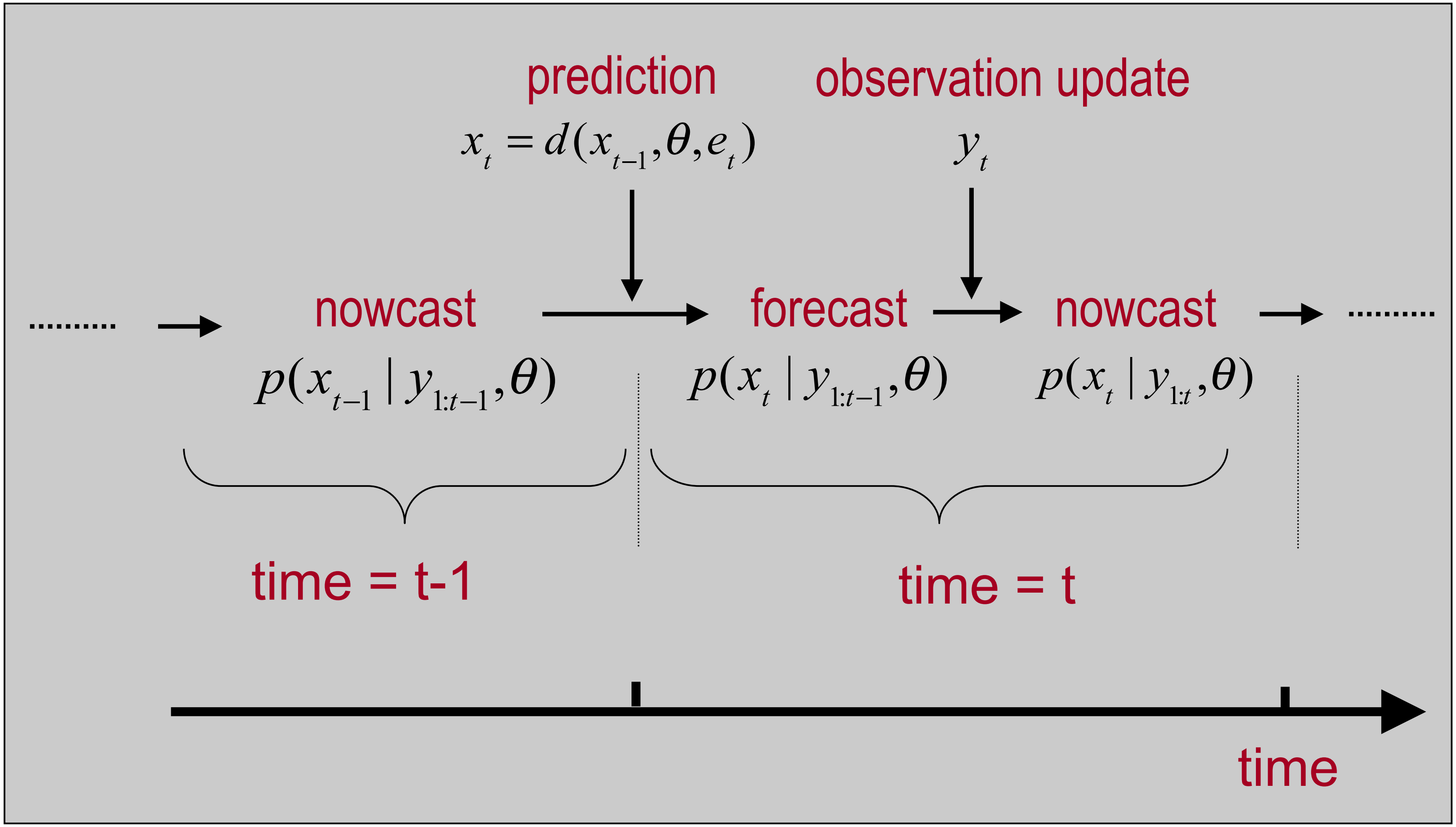
Particle Filter Schematic: Sequential State Estimation

Single stage transition of system from time *t-1* to time *t*



Particle Filter Schematic: Sequential State Estimation

Single stage transition of system from time *t-1* to time *t*



Basic Particle Filter: Sequential Importance Resampling

for t = 1 to T

(a) Prediction: generate sample $\{x_{t|t-1}^{(i)}\}$ following $p(x_t \mid y_{1:t-1}, \theta)$

$$x_{t|t-1}^{(i)} = d(x_{t-1|t-1}^{(i)}, \theta, e_t^{(i)}) \text{ for } i = 1, \ldots, n$$

(b) Observation update: Using newly available observation y_t

- $w_t^{(i)} \propto p(y_t \mid x_{t|t-1}^{(i)}, \theta)$ for $i = 1, \ldots, n$
- resample with replacement from $\{x_{t|t-1}^{(i)}\}$ using weights $w_t^{(i)}$

$\rightarrow$ yields $\{x_{t|t}^{(i)}\}$ from $p(x_t \mid y_{1:t}, \theta)$

end (for t)

Note: there are lots of other (better) particle filtering algorithms!

Bottleneck for High Dimensional Applications

NUMERICAL MODELS ARE COMPUTATIONALLY EXPENSIVE

Practical Consequence:
small ensembles must represent a *large* state space

TWO STRATEGIES:

*1. Approximate the **Observation update***

(so small ensembles work better)

*2. Approximate the **Prediction step***

(so we can generate bigger ensembles)

Approximating the Observation Update Step

(1) An Alternative Observation Update: the Ensemble Kalman Filter

IDEA: Use Kalman filter updating for observation step

$$\tilde{x}_{t|t}^{(i)} = x_{t|t-1}^{(i)} + K(y_t^{(i)} - Hx_{t|t-1}^{(i)}), \quad i = 1,\dots,n$$

$$\text{where: } y_t^{(i)} = y_t + v_t^{(i)}, \quad i = 1,\dots,n \;\; \text{and} \;\; K = PH^T(HPH^T + R)^{-1}$$

The most common approximation for inference in large-scale dynamical systems

(1) An Alternative Observation Update: the Ensemble Kalman Filter

IDEA: Use Kalman filter updating for observation step

$$\tilde{x}_{t|t}^{(i)} = x_{t|t-1}^{(i)} + K(y_t^{(i)} - Hx_{t|t-1}^{(i)}), \quad i = 1,\ldots,n$$

where: $y_t^{(i)} = y_t + v_t^{(i)}, \quad i = 1,\ldots,n$ and $K = PH^T(HPH^T + R)^{-1}$

The most common approximation for inference in large-scale dynamical systems

(1) An Alternative Observation Update: the Ensemble Kalman Filter

IDEA: Use Kalman filter updating for observation step

$$\tilde{x}_{t|t}^{(i)} = x_{t|t-1}^{(i)} + K(y_t^{(i)} - Hx_{t|t-1}^{(i)}), \quad i = 1,\ldots,n$$

where: $y_t^{(i)} = y_t + v_t^{(i)}, \quad i = 1,\ldots,n$ and $K = PH^T(HPH^T + R)^{-1}$

The most common approximation for inference in large-scale dynamical systems

- "Works" for large systems (needs localization, variance inflation).
- Easy to implement (modularity)
- "Breaks" under strong nonlinear, non-Gaussianity.
- Messes up dynamical balances

(1) An Alternative Observation Update: the Ensemble Kalman Filter

IDEA: Use Kalman filter updating for observation step

$$\tilde{x}_{t|t}^{(i)} = x_{t|t-1}^{(i)} + K(y_t^{(i)} - Hx_{t|t-1}^{(i)}), \quad i = 1,\ldots,n$$

where: $y_t^{(i)} = y_t + v_t^{(i)}, \quad i = 1,\ldots,n$ and $K = PH^T(HPH^T + R)^{-1}$

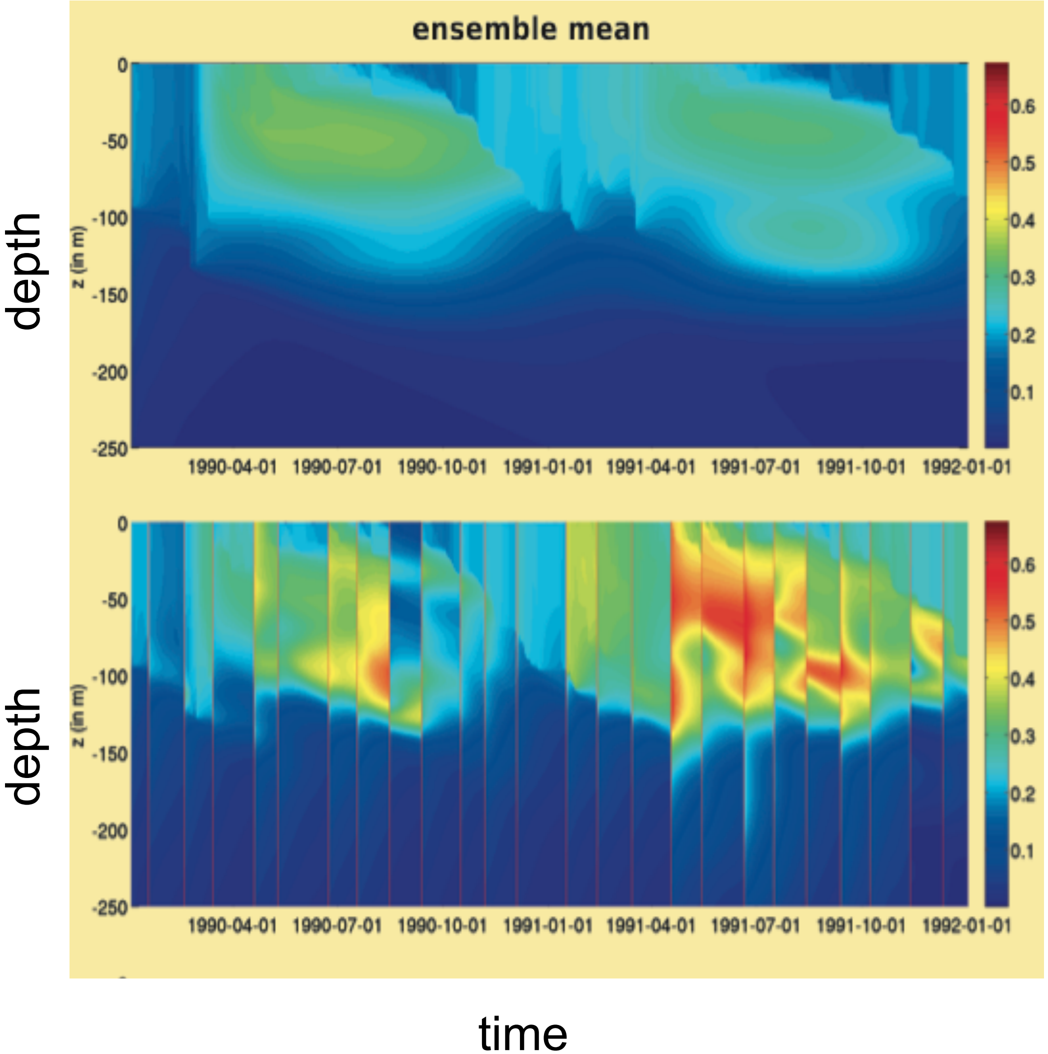
The most common approximation for inference in large-scale dynamical systems

- "Works" for large systems (needs localization, variance inflation).
- Easy to implement (modularity)
- "Breaks" under strong nonlinear, non-Gaussianity.
- Messes up dynamical balances

CAN ALSO USE AS A PROPOSAL FOR SMC ALGORITHM
(lookahead filters / conditional simulation / manifold method)

Results EnKF: Ensemble Mean

Particulate Organic Nitrogen

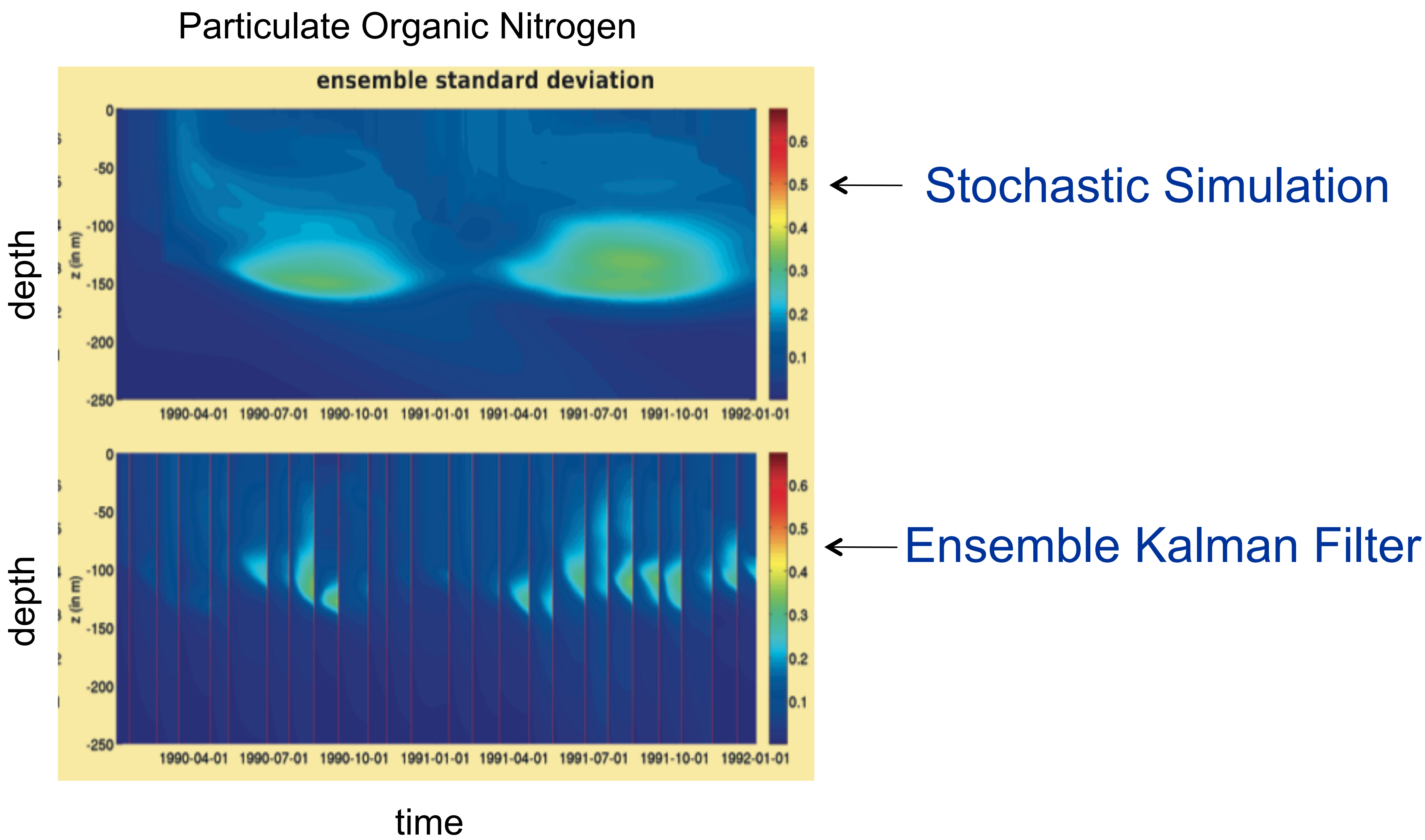


Stochastic Simulation
(no observations used)

Ensemble Kalman Filter
(MV observations assimilated using enKF)

* observed state variables: particulate organic N, dissolved inorganic N, chloropyhll, oxygen

Results EnKF: Ensemble Std Dev



* assimilated state variables: particulate organic N, dissolved inorganic N,, chloropyhll, oxygen

(2) An Alternative Observation Update: Approximate Bayesian Computation

Problems:
(1) Likelihood intractable for some data types,
(2) Difficult for proposals to populate area near posterior mode in high dimensional distribution

Approach: Replace likelihood with scalar distance metric in a particle filter. Use this to compute weights

Benefit: Allows for use of small sample sizes (no weight collapse)

(2) An Alternative Observation Update: Approximate Bayesian Computation

Problems:
(1) Likelihood intractable for some data types,
(2) Difficult for proposals to populate area near posterior mode in high dimensional distribution

Approach: Replace likelihood with scalar distance metric in a particle filter. Use this to compute weights

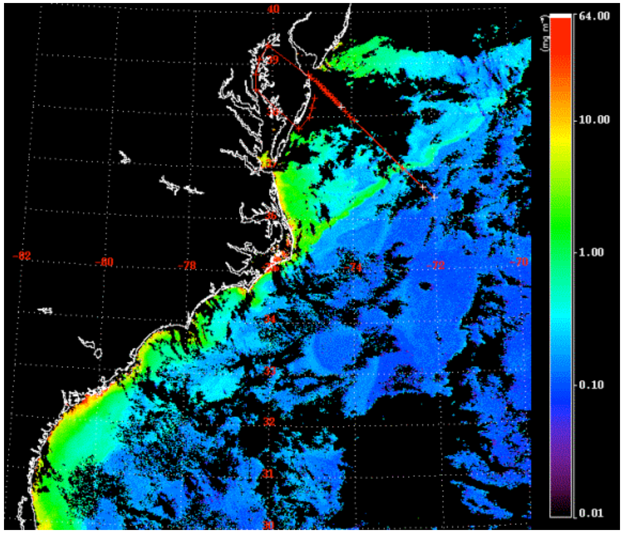
Benefit: Allows for use of small sample sizes (no weight collapse)

EXAMPLE

For image comparison, we used *Adaptive Grey Block Distance to* measure discrepancy between *model predicted spatial field* and the *observed one*.

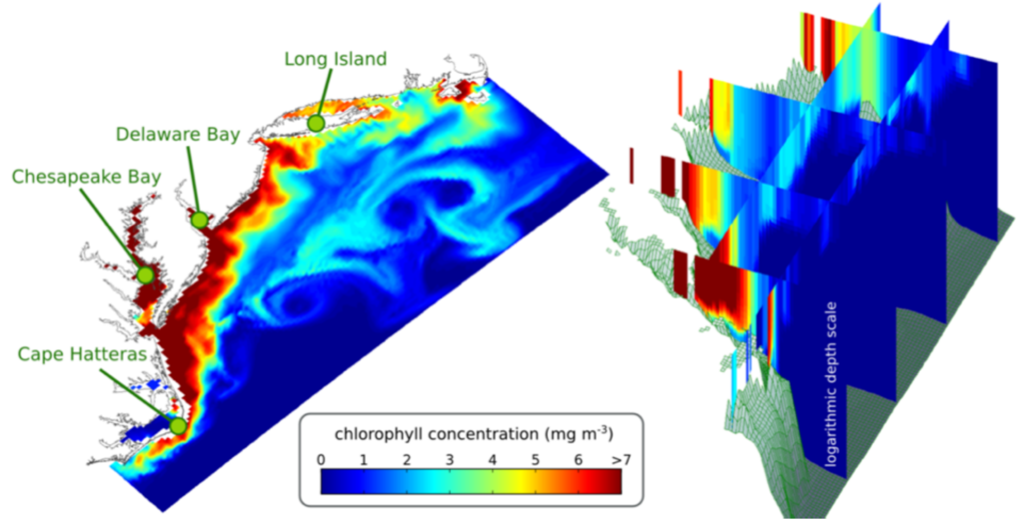
Discrepancy Metric: Adaptive Grey-Block Distance

Observations



VS

Model Prediction



Issues addressed: missing values, mis-alignment/registration errors, multiscale dependence

Application:

- State estimation using 3-D ocean model. AGBD replaced likelihood in particle filter
- Application fairly successful with small ensembles (<100). Also estimated parameters via state augmentation

Approximating the Prediction Step

(1) An Emulator for Parameter Estimation

(for Deterministic Dynamics)

GOAL: Estimate biological ocean state in the mid-Atlantic Bight using:
(1) *Data:* Satellite chlorophyll observations,
(2) *Model:* Deterministic 3-D ocean biogeochemical model

Input: two key selected 'independent' biological parameters.

Output: discrepancy metric, i.e. the AGB distance between model predicted surface field and satellite observations.

Used a Polynomical Chaos emulator →

Polynomial Chaos Emulator

$$f(x,t,\theta) = \sum_{k=0}^{k_{\max}} a_k(x,t)\,\phi_k(\theta) + \varepsilon_{trunc}(\theta)$$

where:

θ : inputs

$f(x,t,\theta)$: outputs

$a_k(x,t)$: expansion coefficents

$\phi_k(\theta)$: basis functions

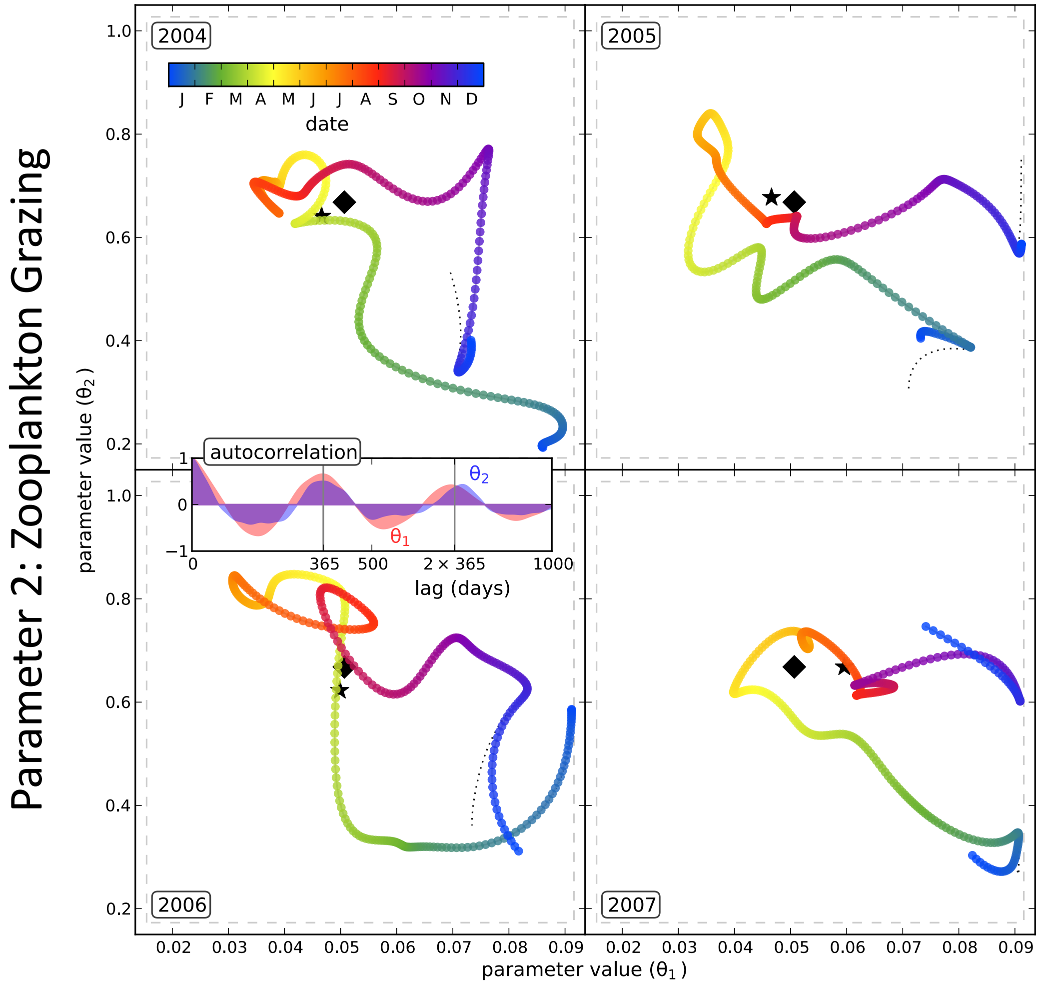
$\varepsilon_{trunc}(\theta)$: truncation error

Note:

- Assumptions about $p(\theta)$ determine which polynomial basis to use
- The polynomial basis and order determines the *n* design points.
- Mean and Variance of output are given by:

$$E\{f(x,t,\theta)\} = a_0(x,t), \quad \text{var}\{f(x,t,\theta)\} = \sum_{k=1}^{n} a_k^2(x,t)$$

Results: Seasonal co-evolution of the 2 parameters



Parameter 1: Phytoplankton Growth

(2) An Emulator for Particle Filtering

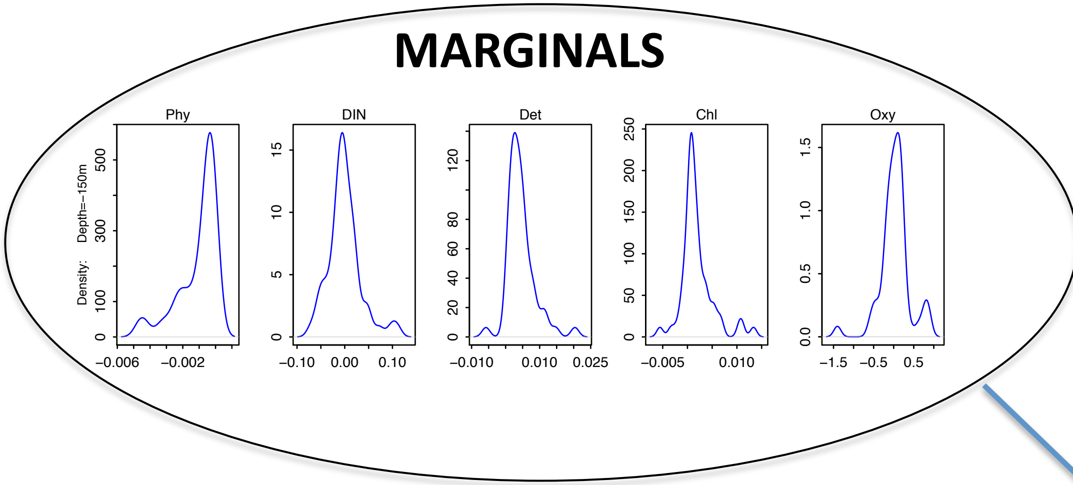
(Stochastic Dynamics)

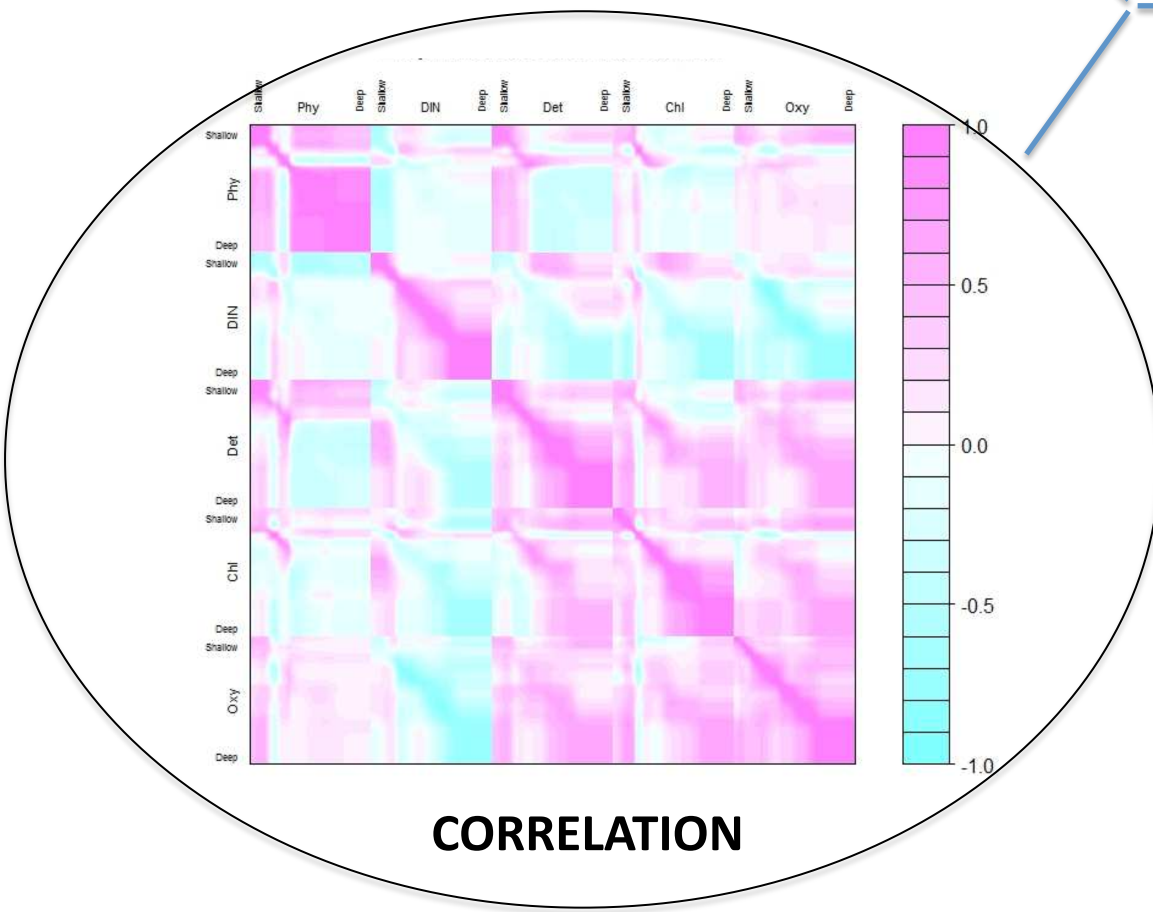
- We want: $x_t \sim p(x_t \mid x_{t-1}, \theta)$ - predictive/transition density
- We have: $x_t = d(x_{t-1}, \theta, e_t)$ - a numerical model to generate samples

Idea: create multivariate distributions for the transition density using copulas …

$$p(x_t \mid x_{t-1}) = c_K(v_1, \dots, v_K) \prod_{k=1}^{K} p(x_{t,k} \mid x_{t-1,1:K})$$

- *Used elliptical copulas (normal and t) to build the transition density*
- *Numerical simulations yield CORRELATIONS and MARGINALS used to build the desired distribution* ➔



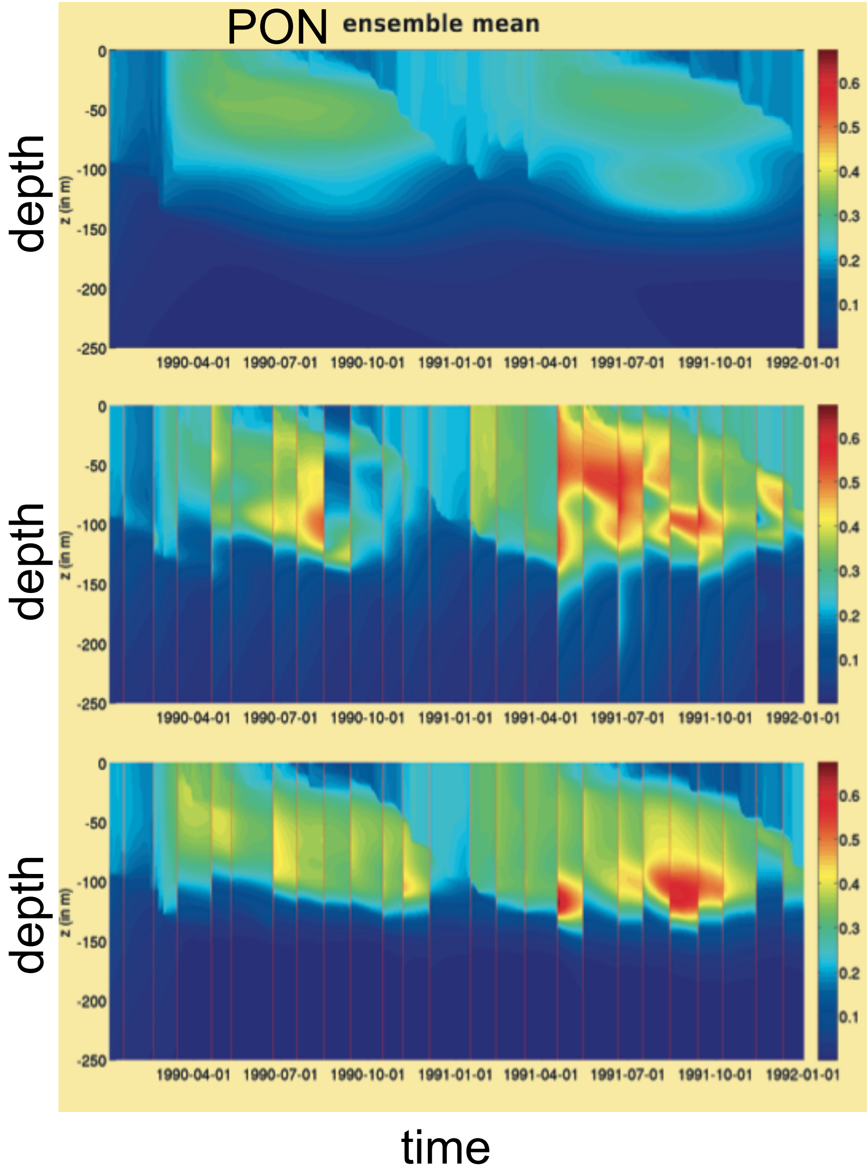


Based on 1D PZND model of BATS site

$$P(x_1,\ldots,x_n) = C(P(x_1),\ldots,P(x_n))$$

Samples from predictive density, or model errors

Results Sequential MC: Ensemble Mean

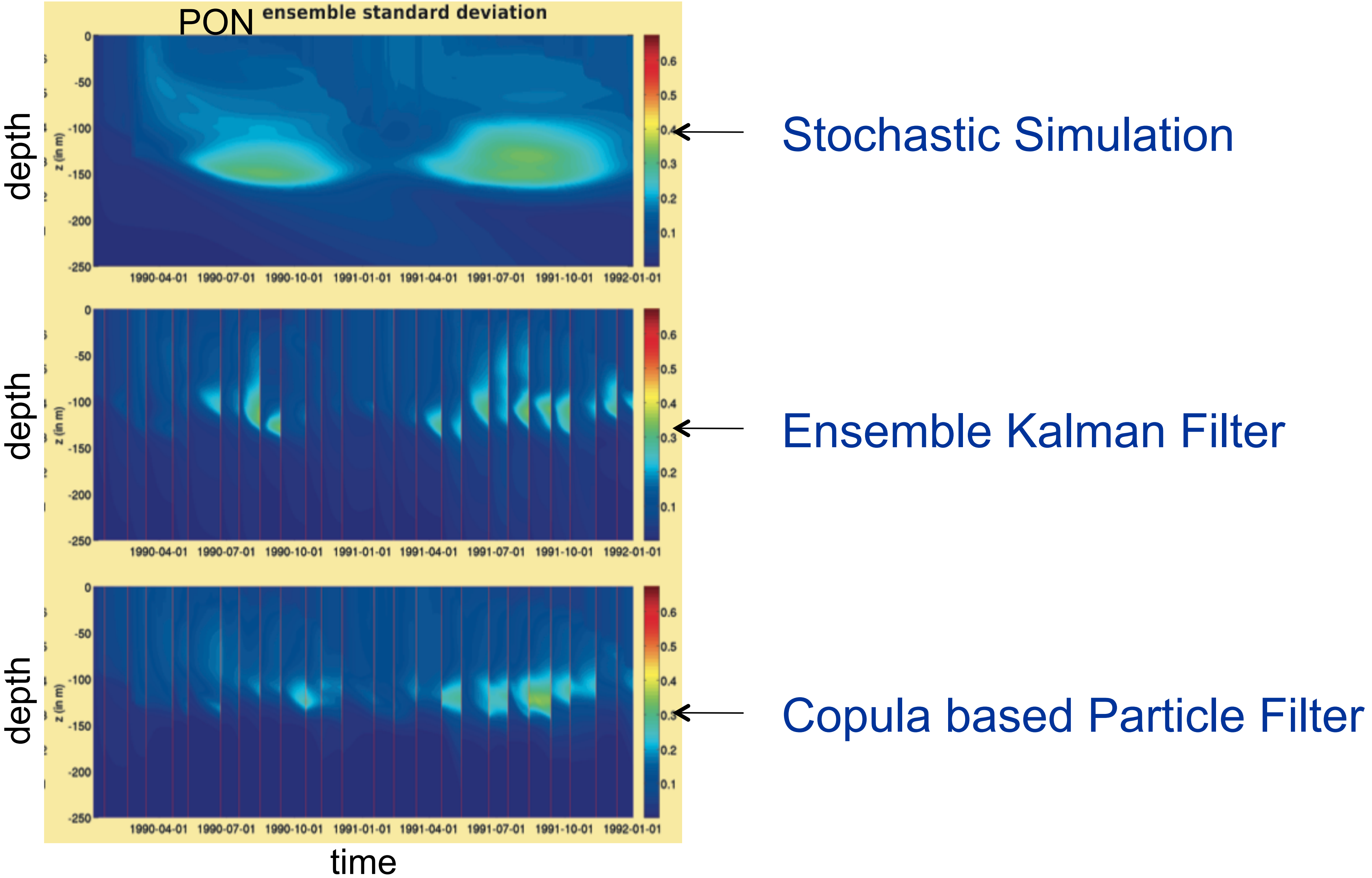


Stochastic Simulation
(no observations used)

Ensemble Kalman Filter

Copula Based Particle Filter

Results Sequential MC: Ensemble Std Dev



Concluding Remarks

- Presented an application-specific tailoring of Sequential Monte Carlo approaches for large scale dynamic systems

- Good approximations are key for workable inference in realistic (high dimension, spatio-temporal) applications for Data Assimilation.

- Ongoing Work: better stochastic simulations (conditional simulation, optimal proposals), and new filtering methods (e.g. a spatial dimension particle filter)

Questions/Comments/Concerns?