High Dimensional Applications of State Space Models (a/k/a Data Assimilation)

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

BIRS Workshop: Inferential Challenges for Large Spatio-Temporal Data Structures. Dec 3-8, 2017

Approximation
Approximation
Approximation

Outline

1. *Problem Statement / Overview:* Review of sampling based approaches for SSMs including computational algorithms.

2. *Approximating the Observation Update:* ensemble Kalman filter, and approximate Bayesian computation

3. *Approximating Dynamical Model Prediction:* Model Surrogates and Emulators for SSMs / Data Assimilation

…. illustrated with applications in ocean data assimilation

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, approximate Kalman filter updating step

1980s: Variational Data Assimilation (Olivier Talagrand)
- Time dependent optimization, adjoints need for gradient, initial

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

Performance Metric for Data Assimilation: Forecast Skill

General Problem Statement

DATA ASSIMILATION = (SIMPLIFIED) SSMS

1. *Dynamical Models:* Ocean, Atmosphere, Earth, Space; Physics, Chemistry, Biology

2. *Observations:* many and varied, temporal and/or spatial

3. *Prior Knowledge*: accumulated scientific knowledge

General Problem Statement

DATA ASSIMILATION = (SIMPLIFIED) SSMS

1.Dynamical Models: Ocean, Atmosphere, Earth, Space;
Physics, Chemistry, Biology

2. *Observations:* many and varied, temporal and/or spatial

3. Prior Knowledge: accumulated scientific knowledge

General Problem Statement

DATA ASSIMILATION = (SIMPLIFIED) SSMS

1. *Dynamical Models:* Ocean, Atmosphere, Earth, Space; Physics, Chemistry, Biology

2. *Observations:* many and varied, temporal and/or spatial

3. *Prior Knowledge*: accumulated scientific knowledge

GENERAL GOAL: to improve scientific understanding

- Estimate the system state and its parameters
- Model selection / system identification
- Sampling and observing array design

Features

- **Dynamics centric**: Numerical models considered a good representation of reality.

- **Data Paucity** (relative to scales of variation), partially observable system

Science driven by assessing data/model discrepancy and using to identify knowledge gaps

Engineering approach to methodology : Do what "works" ….

A Useful Statistical Framework: State Space Model

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

or

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

← DYNAMICS

← OBSERVATIONS

$t = 1, \ldots, T$

State Space Model: Dynamics

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

or

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

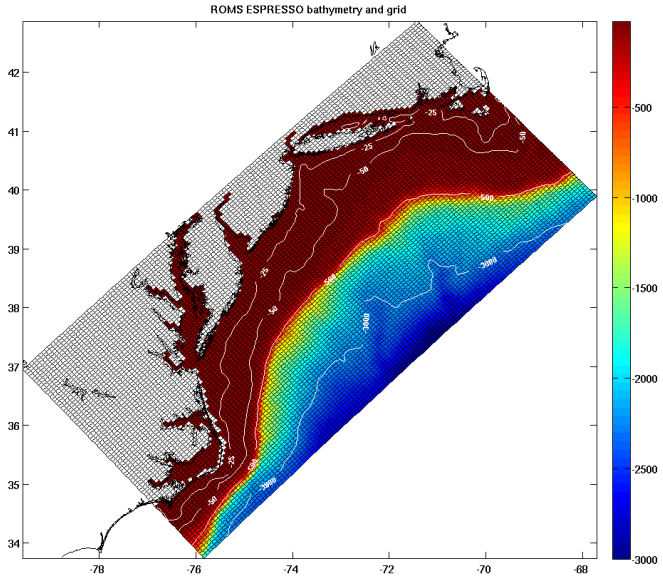
← DYNAMICS

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

Dynamical Models →

DYNAMICAL MODELS

(numerical models/ complex computer code)



Incompressible Navier–Stokes equations (*convective form*)

$$\left(\frac{\partial}{\partial t} + u_j \frac{\partial}{\partial x_j} - \nu \frac{\partial^2}{\partial x_j \partial x_j}\right) u_i = -\frac{\partial w}{\partial x_i} + g_i$$

Tracer equations: biogeochemistry

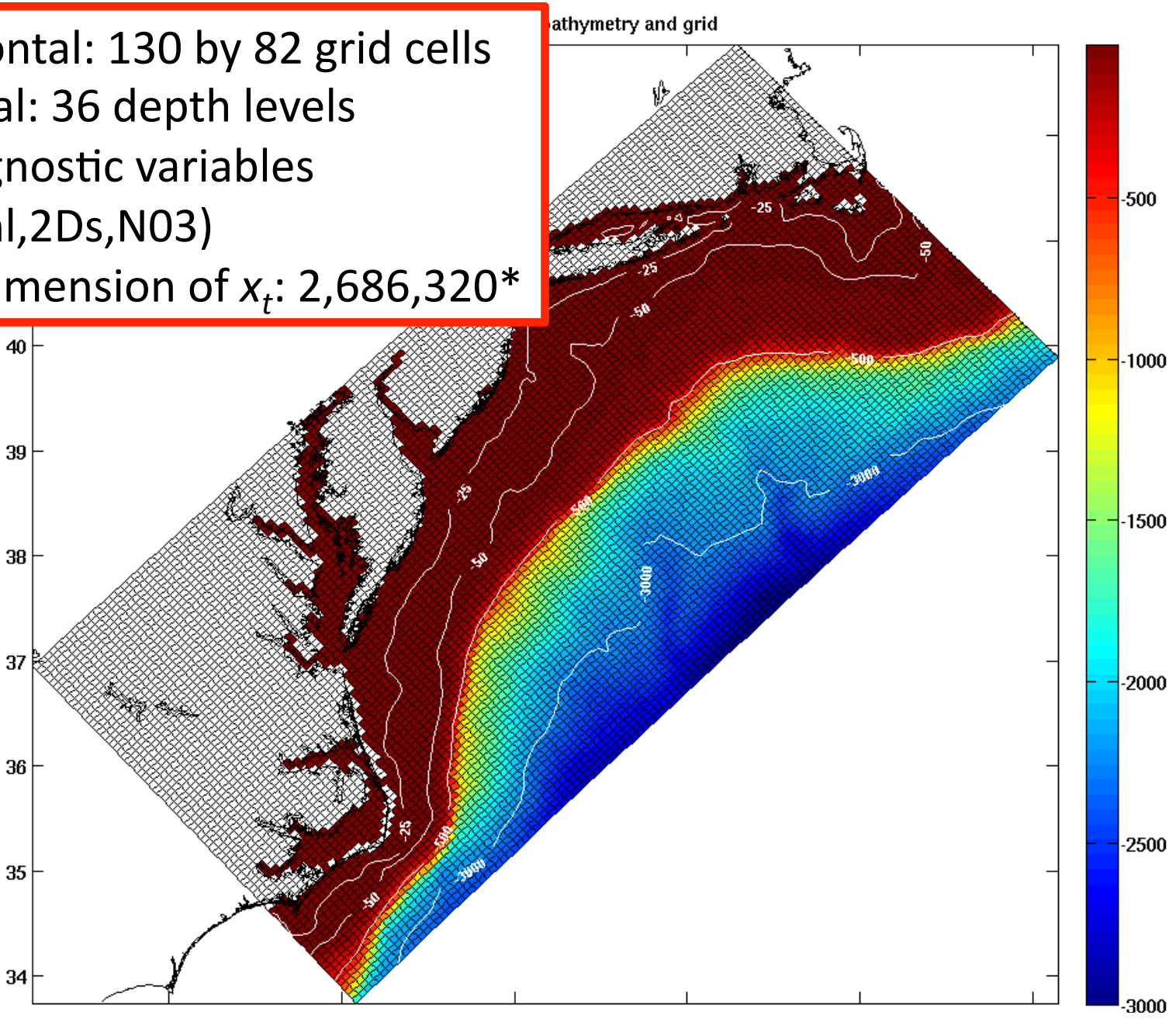
$$\frac{\partial X_i}{\partial t} + \vec{u} \cdot \nabla X_i - \nabla \cdot (K \nabla X_i) = f_i(X_1, \ldots, X_m, \theta) + e(\vec{x}, t)$$

- Integer time index in SSMs = time between observations
- Numerical integration in dynamical models requires short time steps and fine spatial resolution (so computationally costly)

Dimension of x_t is large ➔

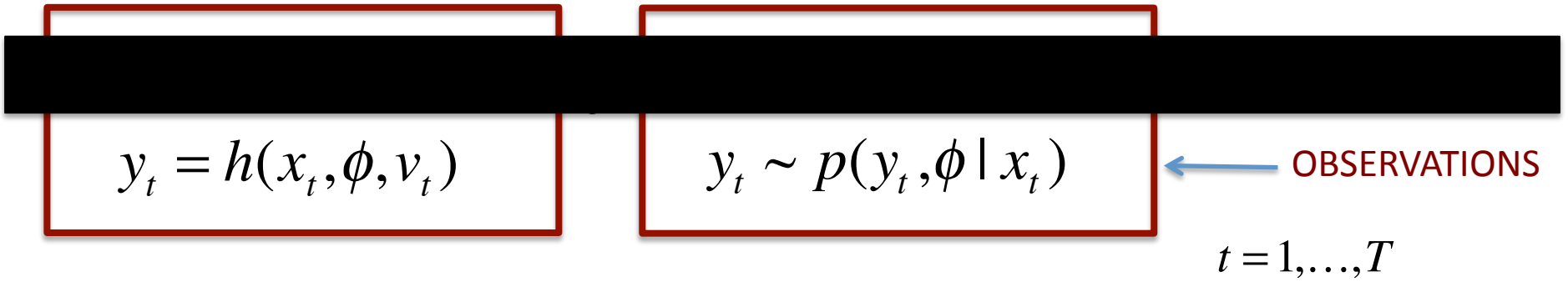
- Horizontal: 130 by 82 grid cells
- Vertical: 36 depth levels
- 7 prognostic variables (P,Z,chl,2Ds,N03)

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



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

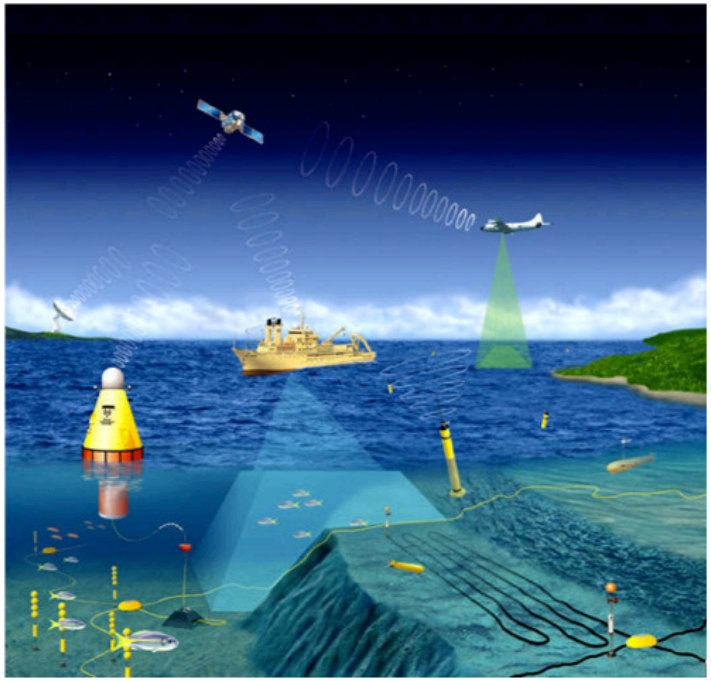
State Space Model: Observations



Observations ➔

OBSERVATIONS

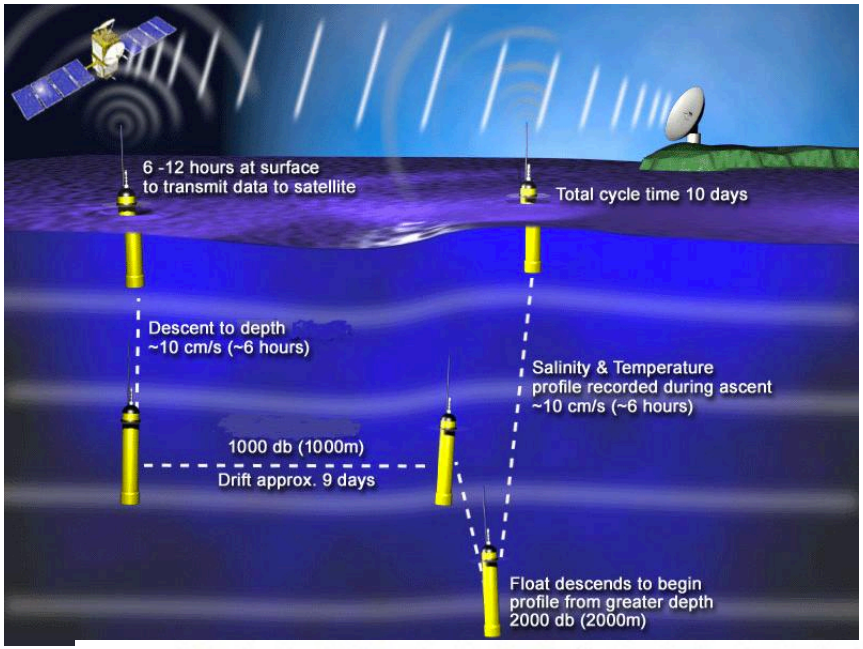
(a true technological revolution …)

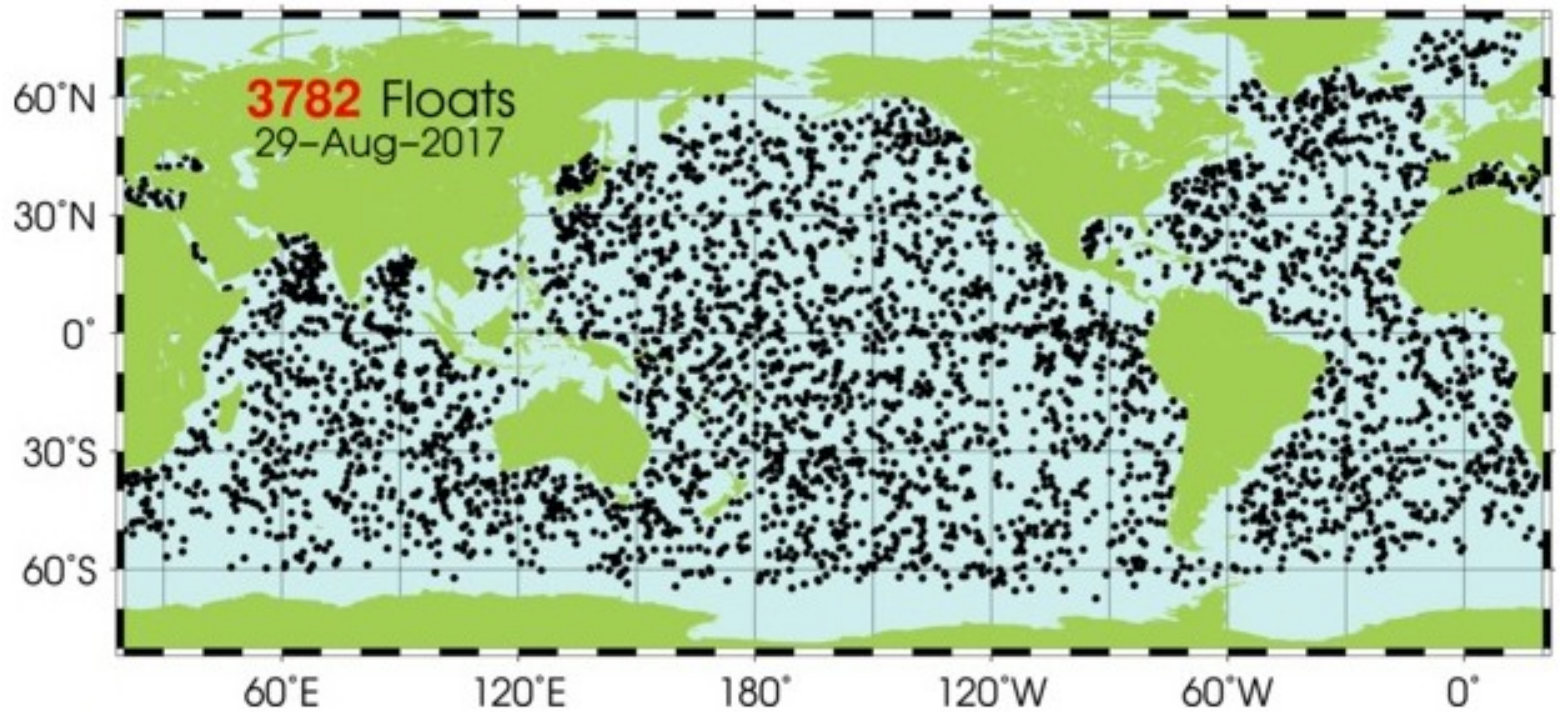


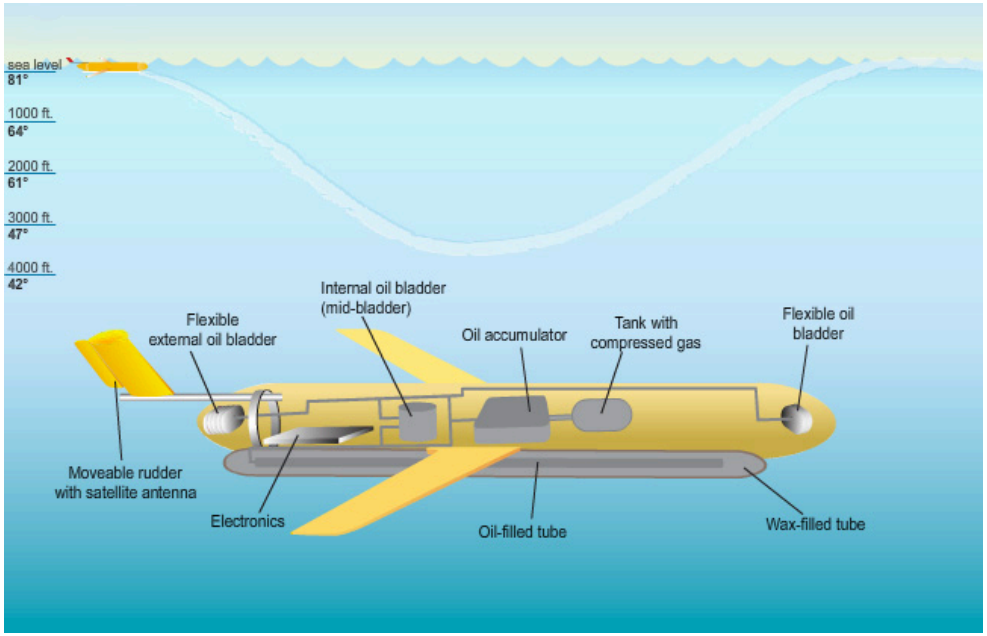
***Traditional Observations*:** point observations, time series, or spatial imagery

***New observations*:** complex spatio-temporal multivariable sampling via autonomous robotic sampling platforms (high information content but hard to visualize/interpret/analyse)

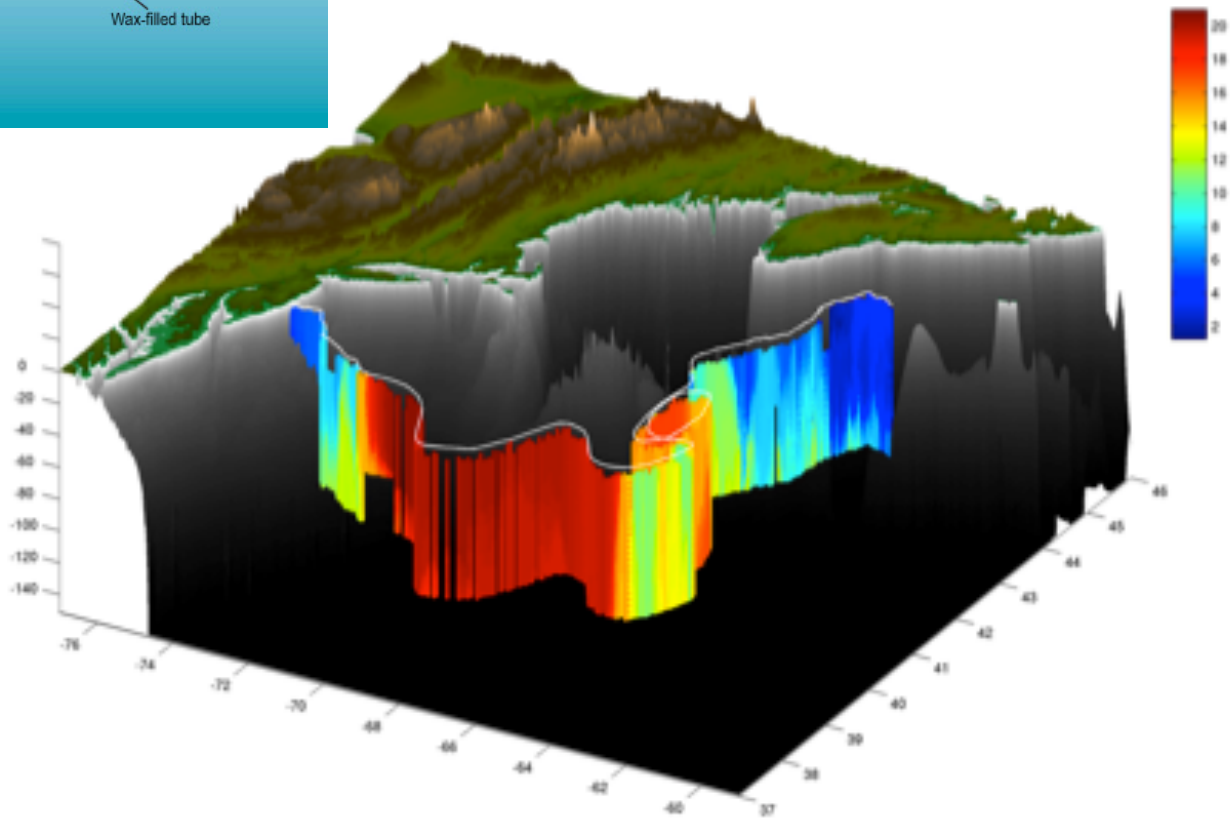
ARGO Floats


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)


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:

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

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 (from literature)

Rely on sampling based solutions in practice ➔

*Aside: Common Simplifications**

***(i) Deterministic Numerical Model*:**

- System state is a deterministic function of the parameters.
- Yields optimization problem (wrt likelihood or cost function)
- Most common large-scale DA approach: variational DA
- Parameters are often include initial (or boundary conditions)

***(ii) Parameters are Fixed and Known*:**

- State estimation via filtering (and smoothing)
- Sample based solutions for nonlinear and non-Gaussian problems rely on sequential Monte Carlo methods (e.g. particle filter)

*You'll see these later

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)

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 $\theta^k = \theta^*$,

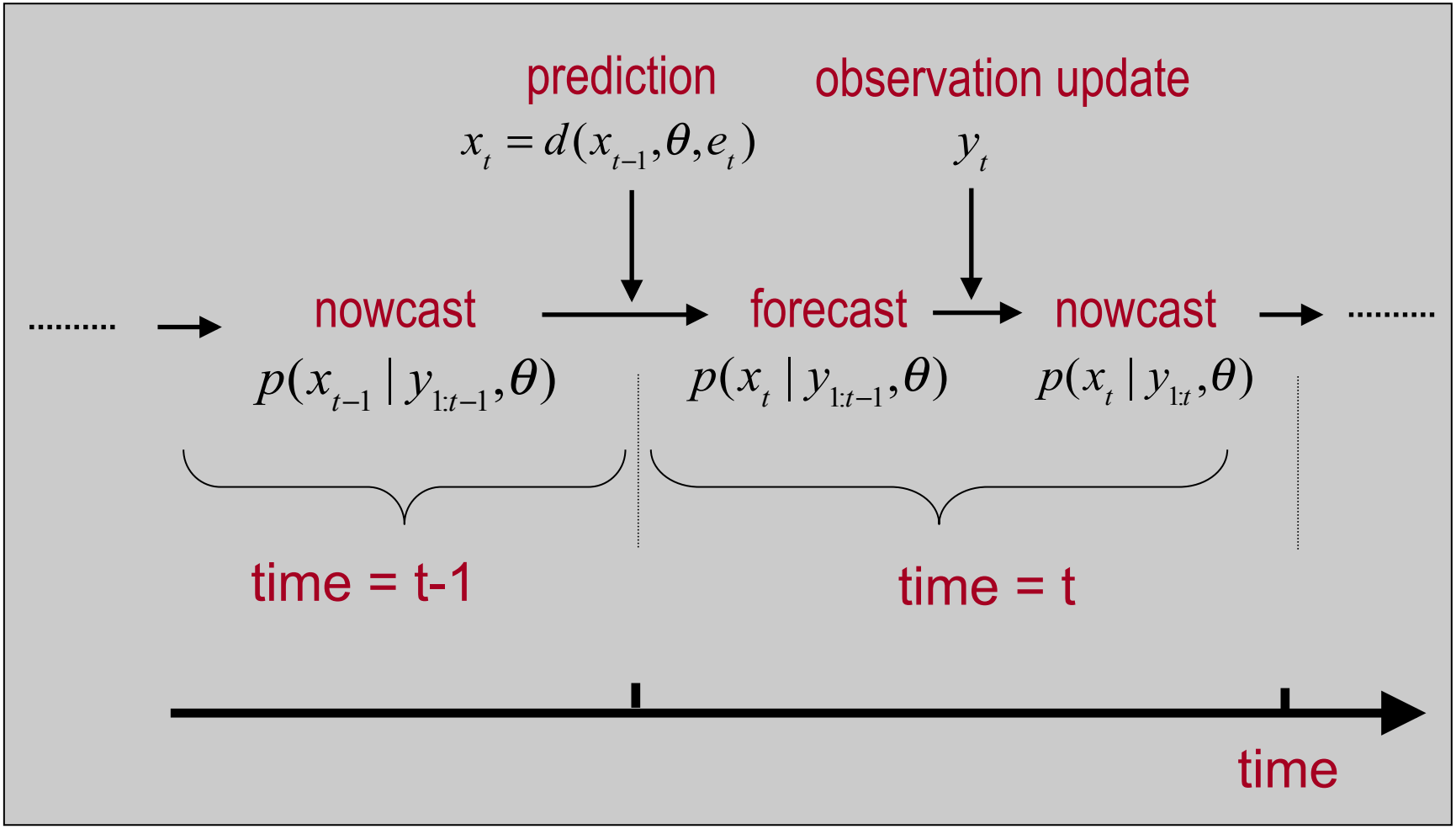
else $\theta^k = \theta^{(k-1)}$

end (for k)

The particle filter is the “engine” for sample based estimation in time dependent systems

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 $p(x_t \mid y_{1:t-1}, \theta)$

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

(b) Observation update: Using newly available observation y_t

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

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

$\rightarrow$ yields $\{x_{t|t}^{(i)}\}$

end (for t)

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

Bottleneck for High Dimensional Applications

NUMERICAL MODELS ARE COMPUTATIONALLY EXPENSIVE

Sample size required for particle filter to work is exponential in *effective* dimension of problem; this which is set by dimension of state and the observations (Bickel et al. 2008)

Practical Issue:
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: Instead of doing weighted resampling for observation update (like SIR based particle filter), instead use Kalman filter updating:

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

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

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

IDEA: Instead of doing weighted resampling for observation update (like particle filter), instead approximate it with Kalman filter updating:

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

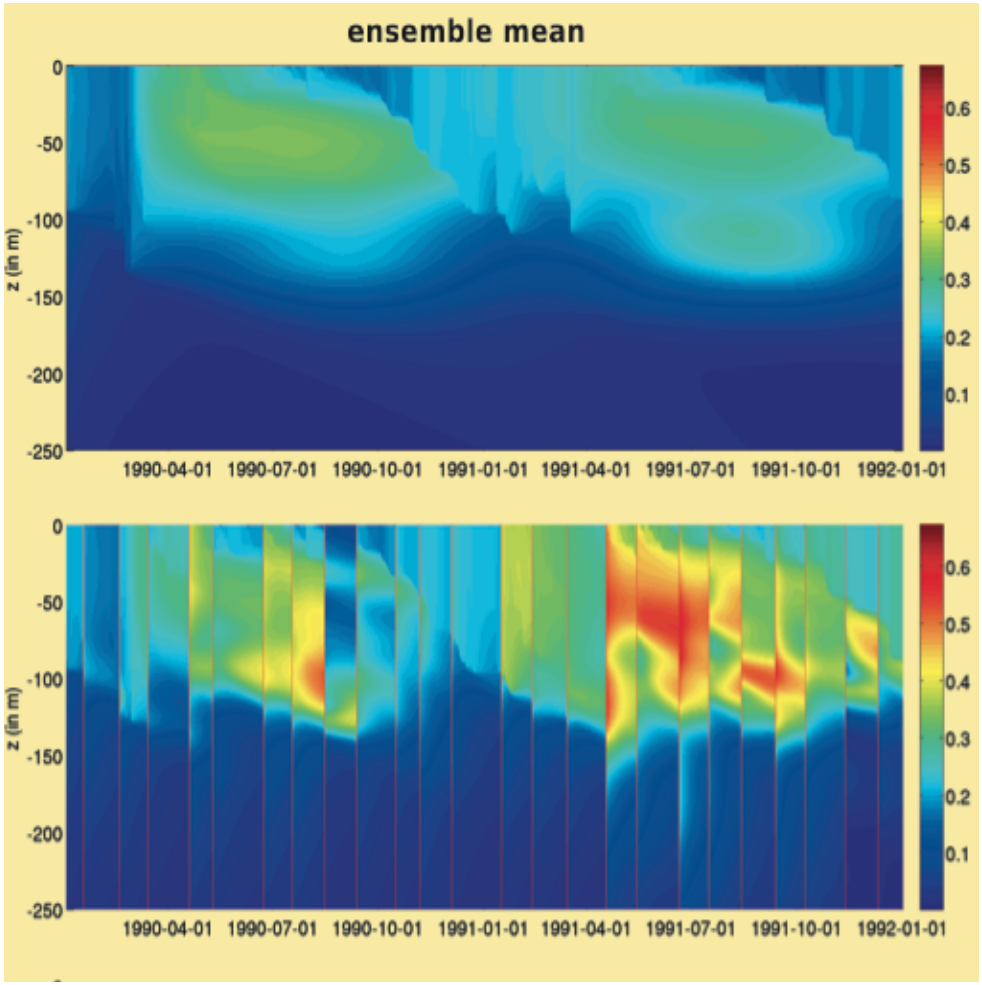
where: $y_t^{(i)} = y_t + v_t^{(i)}, \quad i = 1,\ldots,n$ and $K = PH^T(HPH^T + R)^{-1}$

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

- "Works" for large systems (with a couple of fixes: localization, variance inflation).
- Easy to implement.
- "Breaks" under strong nonlinear, non-Gaussianity.

Results EnKF: Ensemble Mean

Particulate Organic Nitrogen



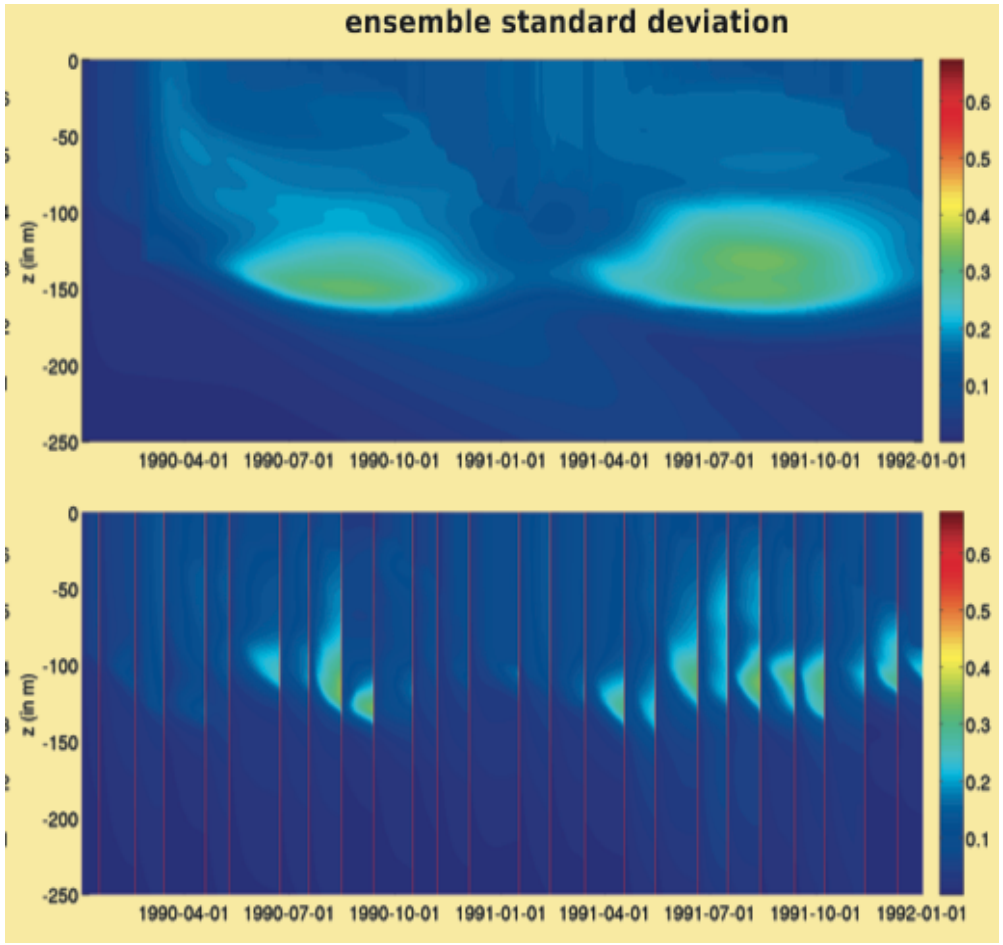
← Stochastic Simulation
(no observations used)

← Ensemble Kalman Filter
(MV observations assimilated using enKF)

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

Results EnKF: Ensemble Std Dev

Particulate Organic Nitrogen



← Stochastic Simulation

← Ensemble Kalman Filter

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

(2) An Alternative Observation Update: Approximate Bayesian Computation:

Problem: Likelihood 'hard to formulate'. Measurement distribution includes: instrument error, environmental variation, errors of representativeness, etc

Approach: Replace likelihood with scalar distance metric.

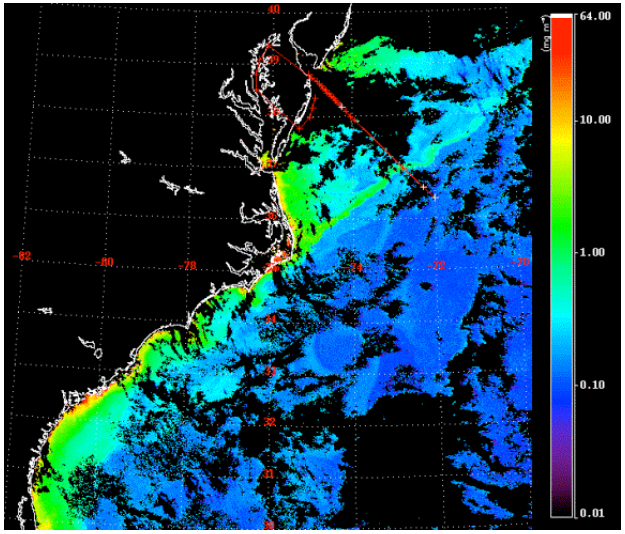
Benefit: Eliminates sample impoverishment in particle filter. Allows for use of small sample sizes.

EXAMPLE

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

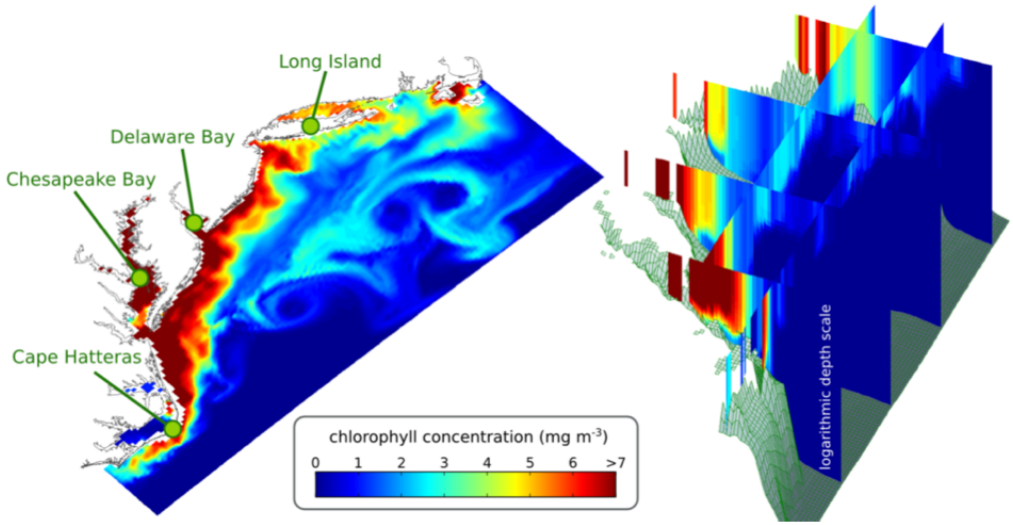
Image Comparison: Adaptive Grey-Block Distance

Observations



VS

Model Prediction



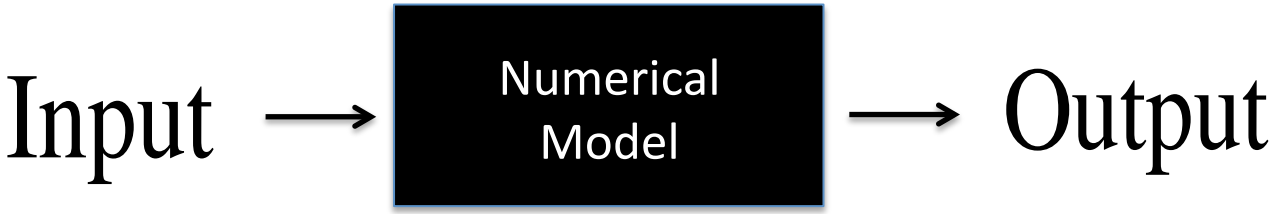
Issues addressed: missing values, mis-alignment/registration errors, scale dependence

***Application*:**

- State estimation using 3-D ocean model. AGBD replaced likelihood in particle filter (used for resampling weights)
- Application proved successful (not shown) and with small ensembles (<100)

Approximating the Prediction Step

Approximating the Dynamical Model : Emulators



Idea: Approximate *targeted aspects* of a computationally costly numerical model (a **simulator**) with an efficient 'statistical' model (an **emulator***)

Approach:

(1) Identify inputs and outputs of interest
(2) Run selected input/output simulations with simulator (experimental design aspect)
(3) Build an emulator from input/output data
(4) Apply it to your inference problem!

* simplest emulator is coarse-resolution numerical model with simplified dynamics

(1) An Emulator for Parameter Estimation

(for Deterministic Dynamics)

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

***Input*:** two selected 'independent' biological parameters.

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

Application:

- build a statistical emulator using specified input/output simulations
- estimate seasonal evolution of the two parameters by minimizing the discrepancy metric.

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

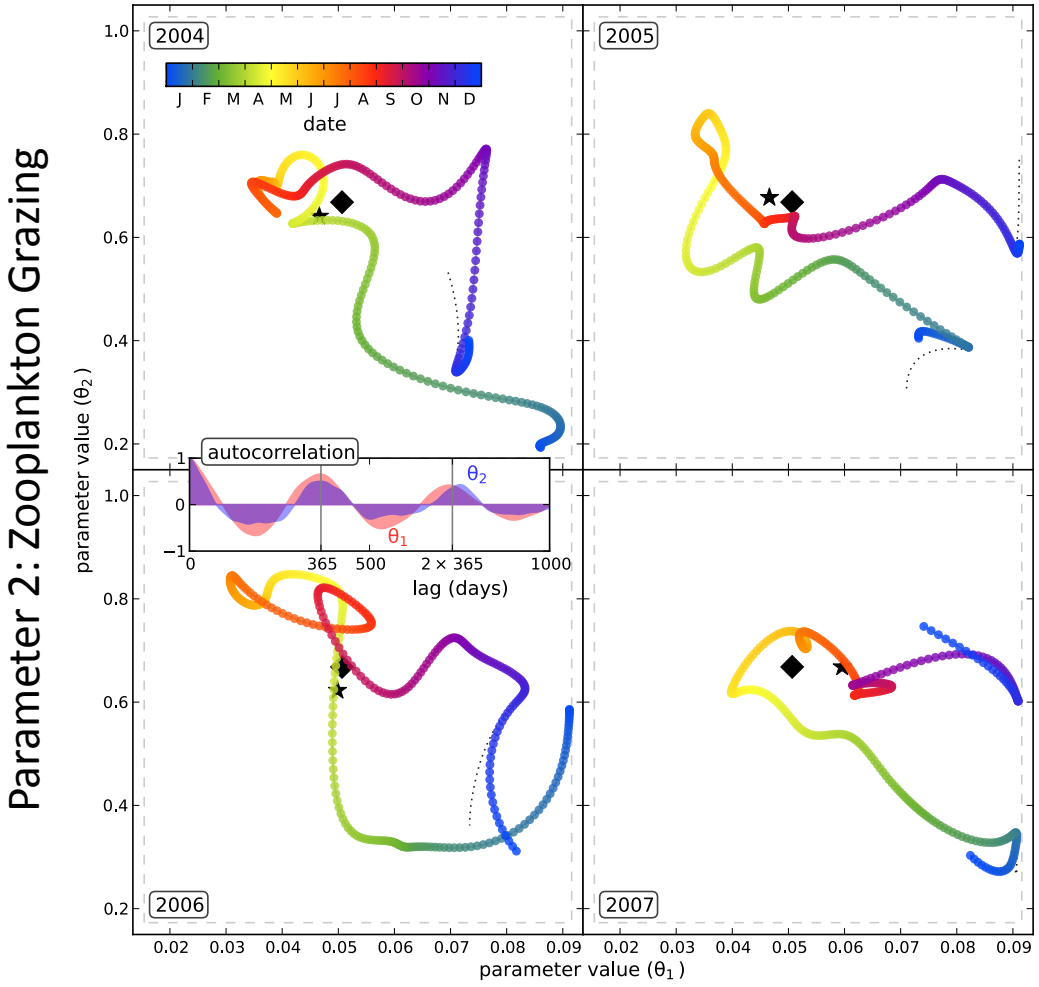
$\varepsilon_{trunc}(\theta)$: truncation error

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



(2) An Emulator for Particle Filtering

(Stochastic Dynamics)

Rationale: the one-step-ahead state prediction (x_{t-1} to x_t) is a key quantity for SSMs

Idea: Replace numerical model prediction with an emulator

➔ *allow for computationally efficient sample generation*

Input: system state at time t-1: x_{t-1}

Output: system state at time t: x_t

Approach:

(1) Emulate the state transition with copula-based MV distribution

(2) Use these approximate dynamics in particle filter/smoother

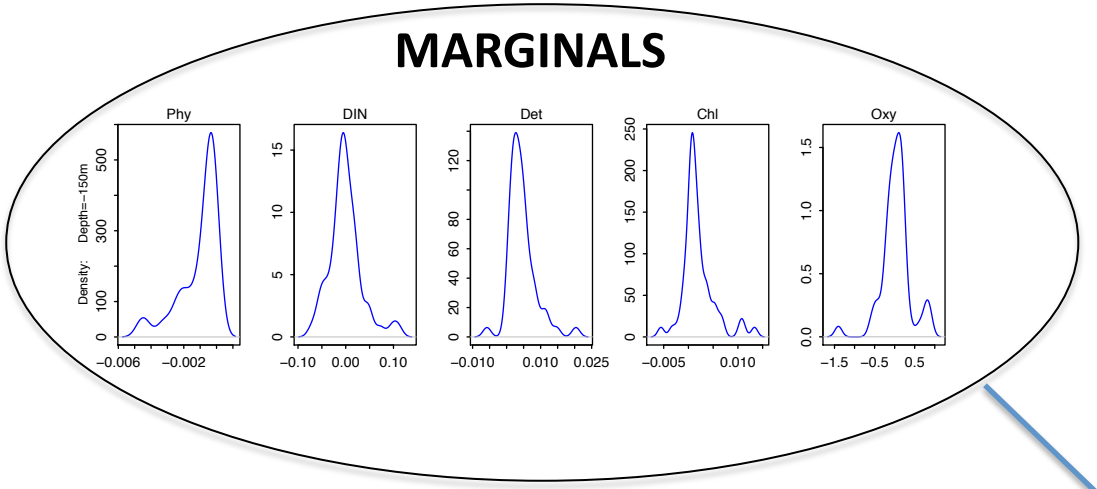
Building a Transition Density with Copulas

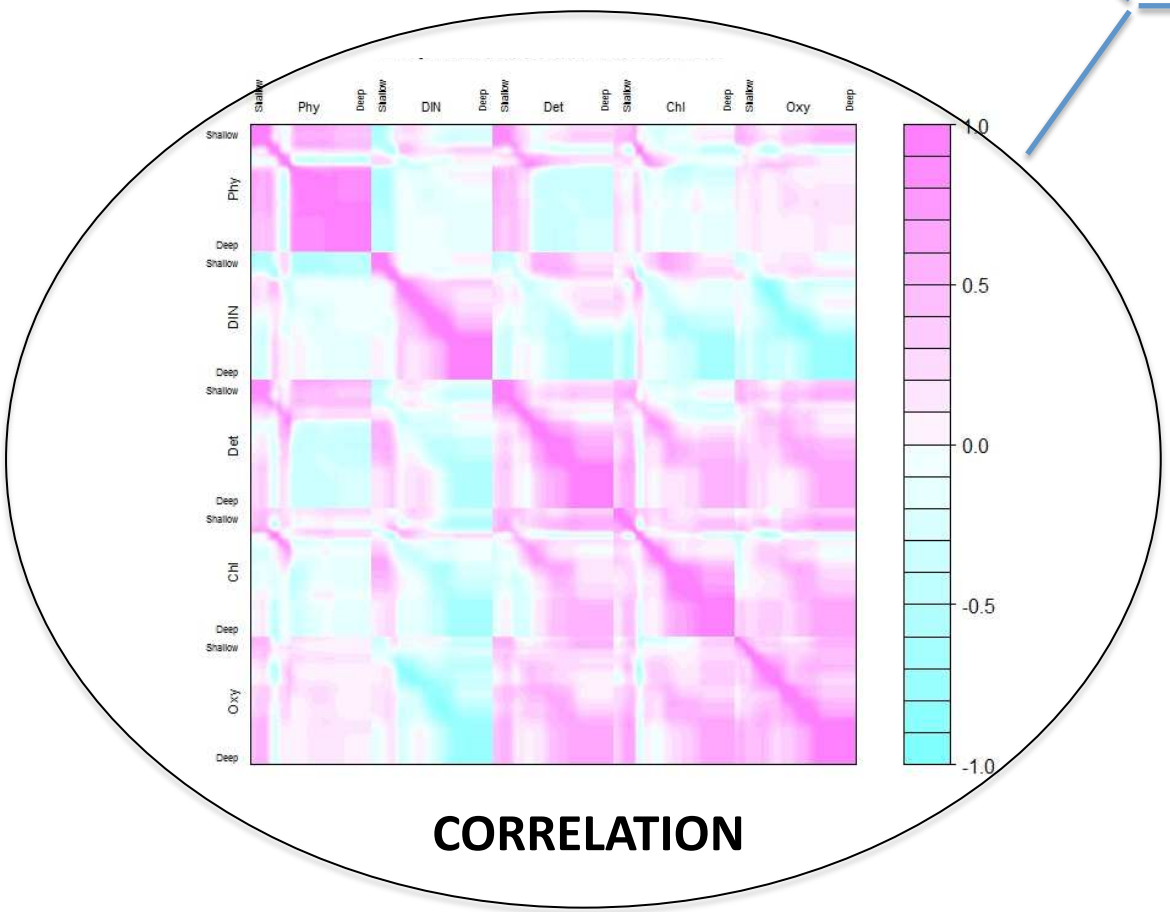
- We want: $x_t \sim p(x_t \mid x_{t-1}, \theta)$ - predictive/transition density
- We have: $x_t = d(x_{t-1}, \theta, e_t)$ - a numerical model to generate samples

Idea: create multivariate distributions using copulas …

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

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



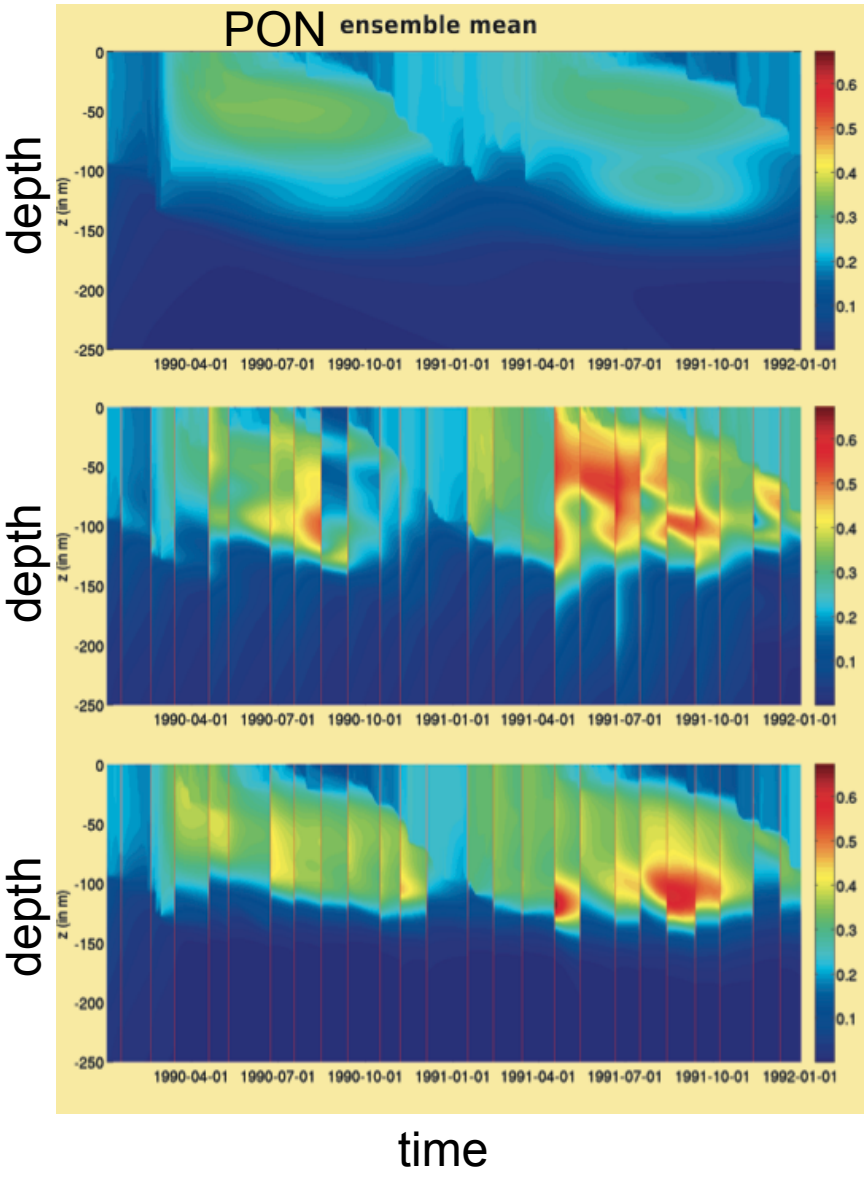


Based on 1D PZND model of BATS site

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

Samples from predictive density, or model errors

Results Sequential MC: Ensemble Mean

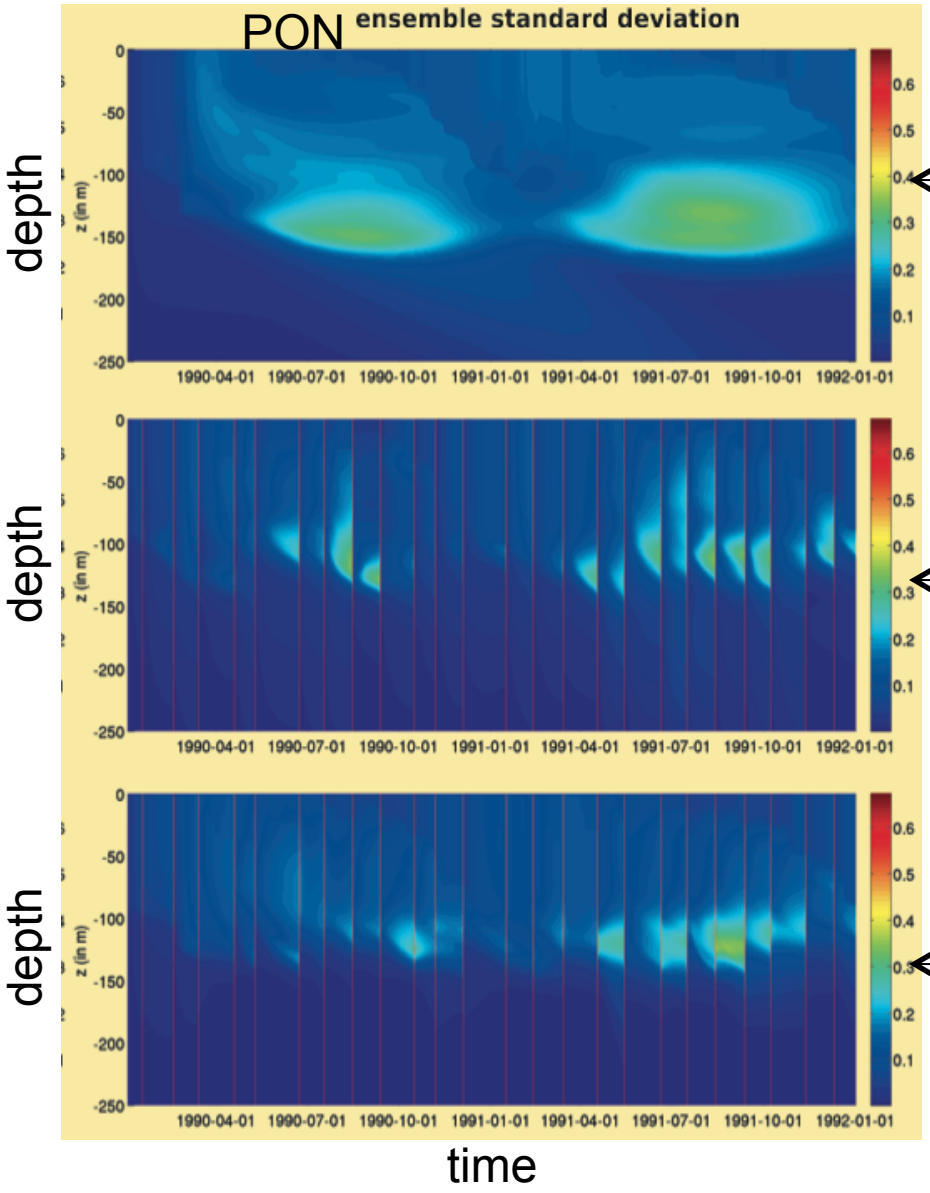


Stochastic Simulation
(no observations used)

Ensemble Kalman Filter

Copula Based Particle Filter

Results Sequential MC: Ensemble Std Dev



Stochastic Simulation

Ensemble Kalman Filter

Copula based Particle Filter

Concluding Remark

Good approximations needed for estimation for realistic (high dimension, spatio-temporal) applications of State Space Models for Data Assimilation

General guidelines, but no easy ("one size fits all") answer.

Questions/Comments/Concerns?

Challenges/Ideas

Small: <10
Moderate: 10-100
High: > 100

Stochastic dynamics

Interpreting complex spatio-temporal observations – really hard with DA

Use subject matter specific numerical models. Otherwise no one will care. If so, big impact

Towards full Bayesian problems

Separate state and parameter estimation? Model calbration vs online prediction?

Characterizing model errors (ensemble simulations). Characterizing approximation errors

Characterizing measurement distributions: instrument error, errors of represenativess/change of support (point observation vs grid cell average)

Alternatives to sample based estimation? Functional, variational weak-constraint

Move beyond state and parameter estimation. Mainly in online prediction, some reconstruction. Want model selection, etc.

How to make most effective use of small samples?

Incorporating Emulator in Hierarchy

The Hierarchical Bayesian Model with an emulator 'level':

$$p(x_{1:T}, \tilde{x}_{1:T}, \theta \mid y_{1:T})$$

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

Would alter particle-MCMC algorithm as follows:

- particle filter now uses emulator approximation as proposal
  ➔ alter weight calculation
- M-H acceptance probability now uses of emulator error, rather than just likelihood ratio, in its calculation

Computationally more efficient, but would lose dynamical balances of basic (SIR) particle filter.

Adaptations of PF for Ocean DA for 3-D BGC

Alter the Likelihood function: change its functional form, or inflate or alter the measurement error.

Error Subspace: confine stochasticity to parameters only. Dimension reduction.

Use Fixed lag smoother, Batch processing incorporate observations from multiple times into observation update. Robustness.

Clever Proposal Distributions and look-ahead filters: move beyond using “prior” (predictive density) as proposal, e.g. Use EnKF

Goal: Incorporating Emulator in Hierarchy

The Hierarchical Bayesian Model with an emulator 'level':

$$p(x_{1:T}, \tilde{x}_{1:T}, \theta \mid y_{1:T})$$

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

Would alter particle-MCMC algorithm as follows:

- particle filter now uses emulator approximation as proposal
  ➔ alter weight calculation
- M-H acceptance probability now uses of emulator error, rather than just likelihood ratio, in its calculation

Computationally more efficient, BUT do lose dynamical balances between prognostic variables inherent in basic (SIR) particle filter.

The Filtering Problem: State Estimation

A single stage transition of the system for time *t-1* to *t* involves:

Dynamic Model Prediction:

$$p(x_t \mid y_{1:t-1}, \theta) = \int p(x_t \mid x_{t-1}, \theta) \cdot p(x_{t-1} \mid y_{1:t-1}, \theta)\, dx_{t-1}$$

Observation Update:

$$p(x_t \mid y_{1:t}, \theta) = \frac{p(y_t \mid x_t, \theta) \cdot p(x_t \mid y_{1:t-1}, \theta)}{p(y_{1:t})}$$

do for $t=1, \ldots, T$, given $p(x_0)$ and $y_{1:T} = (y_1, y_2, .., y_T)$