Model-reduced 4D-Var data assimilation in the phytoplankton bloom prediction

Joanna S. Pelc[1,2], Ghada El Serafy[2], Arnold W. Heemink[1]

Special thanks to Hans Los and Jan van Beek

[1]Delft University of Technology, Delft, The Netherlands
[2]Deltares, Delft, The Netherlands

JONSMOD, Delft, 10 - 12 May 2010

Outline

Introduction

Main objective

Improve the predictions of the BLOOM/GEM model

Approach

Update the uncertain parameters of the model

Introduction

Main objective

Improve the predictions of the BLOOM/GEM model

Approach

Update the uncertain parameters of the model

Which parameters are uncertain?

Introduction

What has been already done

~250 parameters → Out of approximately 250 parameters of the model, EXPERTS have selected a group of 71 the most significant

~70 parameters → Out of 71 parameters, 20 the most significant were selected by means of SENSITIVITY ANALYSIS

What is to be done

~20 parameters → Approximately 20 parameters is going to be updated to improve the model output, by means of DATA ASSIMILATION

Model: BLOOM/GEM (Generic Ecological Model)

State vector: $x_i \in \mathbb{R}^{25 \cdot 4350}$, where i stands for time
$x_i = [Alg,\ Nut\ (NH_4,\ NO_3,\ PO_4,\ Si),\ Det\ ,DetSed,\ Oxy]_i^T$

Parameter vector: $\alpha \in \mathbb{R}^{71}$
$\alpha = [ExtVL,\ NCR,\ PCR,\ SCR,\ ChlaCR,\ PPMax]^T$

Forward model
$x_i = \mathcal{M}_i(x_{i-1}, \alpha)$

$\mathcal{M}_i$ integrates the state x_{i-1} from time t_i to time t_{i-1}

Observation operator
$y_i = \mathcal{H}_i(x_i, \alpha)$

$\mathcal{H}_i$ translates model state x_i into a value in observation space

Model: BLOOM/GEM (Generic Ecological Model)

State vector: $x_i \in \mathbb{R}^{25 \cdot 4350}$, where i stands for time
$x_i = [$ **Alg**, $Nut\ (NH_4,\ NO_3,\ PO_4,\ Si),\ Det\ ,\ DetSed,\ Oxy]_i^T$

Parameter vector: $\alpha \in \mathbb{R}^{71}$
$\alpha = [ExtVL,\ NCR,\ PCR,\ SCR,$ **ChlaCR**, $PPMax]^T$

Forward model
$$x_i = \mathcal{M}_i(x_{i-1}, \alpha)$$

$\mathcal{M}_i$ integrates the state x_{i-1} from time t_i to time t_{i-1}

Observation operator
$$y_i = \mathcal{H}_i(x_i, \alpha)$$

$$\sum_{j=1}^{12} ChlaCR_j \cdot Alg_j$$

4D-Var: 4D Variational Data Assimilation

Cost function in 4D-Var

$$J(\alpha) = \sum_{i=1}^{n} \underbrace{(\mathcal{H}_i(x_i) - y_i)^T}_{Distances} R_i^{-1} \underbrace{(\mathcal{H}_i(x_i) - y_i)}_{Distances} + \underbrace{(\alpha - \alpha_b)^T B^{-1}(\alpha - \alpha_b)}_{Background \quad term}$$

minimization with constraints:
$x_i = \mathcal{M}_i(x_{i-1}, \alpha)$

B background error covariance matrix
R_i observation error covariance matrix

To minimize J over α, we need: $\nabla_\alpha J(\alpha)$

Needed to get $\nabla_\alpha J(\alpha)$

- exact derivatives of the model
- approximate the derivatives of the model with finite differences

4D-Var: 4D Variational Data Assimilation

Cost function in 4D-Var

$$J(\alpha) = \sum_{i=1}^{n} \underbrace{(\mathcal{H}_i(x_i) - y_i)^T}_{Distances} R_i^{-1} \underbrace{(\mathcal{H}_i(x_i) - y_i)}_{Distances} + \underbrace{(\alpha - \alpha_b)^T B^{-1} (\alpha - \alpha_b)}_{Background \quad term}$$

minimization with constraints:
$x_i = \mathcal{M}_i(x_{i-1}, \alpha)$

B background error covariance matrix
R_i observation error covariance matrix

To minimize J over α, we need: $\nabla_\alpha J(\alpha)$

Needed to get $\nabla_\alpha J(\alpha)$

- **Very complicated!**
- approximate the derivatives of the model with finite differences

4D-Var: 4D Variational Data Assimilation

Cost function in 4D-Var

$$J(\alpha) = \sum_{i=1}^{n} \underbrace{(\mathcal{H}_i(x_i) - y_i)^T}_{Distances} R_i^{-1} \underbrace{(\mathcal{H}_i(x_i) - y_i)}_{Distances} + \underbrace{(\alpha - \alpha_b)^T B^{-1}(\alpha - \alpha_b)}_{Background \quad term}$$

minimization with constraints:
$x_i = \mathcal{M}_i(x_{i-1}, \alpha)$

B background error covariance matrix
R_i observation error covariance matrix

To minimize J over α, we need: $\nabla_\alpha J(\alpha)$

Needed to get $\nabla_\alpha J(\alpha)$

- **Very complicated!**
- **Very time consuming!**

4D-Var: 4D Variational Data Assimilation

Cost function in 4D-Var

$$J(\alpha) = \sum_{i=1}^{n} \underbrace{(\mathcal{H}_i(x_i) - y_i)^T}_{Distances} R_i^{-1} \underbrace{(\mathcal{H}_i(x_i) - y_i)}_{Distances} + \underbrace{(\alpha - \alpha_b)^T B^{-1}(\alpha - \alpha_b)}_{Background\ \ term}$$

minimization with constraints:
$x_i = \mathcal{M}_i(x_{i-1}, \alpha)$

B background error covariance matrix
R_i observation error covariance matrix

To minimize J over α, we need: $\nabla_\alpha J(\alpha)$

Solution

POD Model-reduced 4D-Var
(Vermeulen and Heemink, MWR, 2006)

POD Model Reduced 4D-Var

Incremental cost function in 4D-Var

$$J(\delta\alpha) = \sum_{i=1}^{n} (\mathbf{H}_i(\delta x_i, \delta\alpha) + d_i)^T R_i^{-1} (\mathbf{H}_i(\delta x_i, \delta\alpha) + d_i) + \delta\alpha^T B^{-1} \delta\alpha$$

minimization with constraints

$$\delta x_i = \frac{\partial \mathcal{M}_i(x_{i-1}^b, \alpha^b)}{\partial x_{i-1}} \delta x_{i-1} + \frac{\partial \mathcal{M}_i(x_{i-1}^b, \alpha^b)}{\partial \alpha} \delta\alpha$$

POD Model Reduced 4D-Var

Incremental cost function in 4D-Var

$$J(\delta\alpha) = \sum_{i=1}^{n} (\mathbf{H}_i(\delta x_i, \delta\alpha) + d_i)^T R_i^{-1} (\mathbf{H}_i(\delta x_i, \delta\alpha) + d_i) + \delta\alpha^T B^{-1} \delta\alpha$$

minimization with constraints

$$\delta x_i = \frac{\partial \mathcal{M}_i(x_{i-1}^b, \alpha^b)}{\partial x_{i-1}} \delta x_{i-1} + \boxed{\frac{\partial \mathcal{M}_i(x_{i-1}^b, \alpha^b)}{\partial \alpha}} \delta\alpha$$

With a small number of parameters, the finite differences method is feasible

POD Model Reduced 4D-Var

Incremental cost function in 4D-Var

$$J(\delta\alpha) = \sum_{i=1}^{n} (\mathbf{H}_i(\delta x_i, \delta\alpha) + d_i)^T R_i^{-1} (\mathbf{H}_i(\delta x_i, \delta\alpha) + d_i) + \delta\alpha^T B^{-1} \delta\alpha$$

minimization with constraints

$$\delta x_i = \frac{\partial \mathcal{M}_i(x_{i-1}^b, \alpha^b)}{\partial x_{i-1}} \delta x_{i-1} + \frac{\partial \mathcal{M}_i(x_{i-1}^b, \alpha^b)}{\partial \alpha} \delta\alpha$$

If the size of the state x is huge, the finite differences would be too expensive

POD Model Reduced 4D-Var

Incremental equation

$$\delta x_i = \frac{\partial \mathcal{M}_i(x^b_{i-1}, \alpha^b)}{\partial x_{i-1}} \delta x_{i-1} + \frac{\partial \mathcal{M}_i(x^b_{i-1}, \alpha^b)}{\partial \alpha} \delta \alpha$$

Project the increments into smaller subspace

$$P^T \delta x_i = P^T \frac{\partial \mathcal{M}_i(x^b_{i-1}, \alpha^b)}{\partial x_{i-1}} \delta x_{i-1} + P^T \frac{\partial \mathcal{M}_i(x^b_{i-1}, \alpha^b)}{\partial \alpha} \delta \alpha$$

POD Model Reduced 4D-Var

Incremental equation

$$\delta x_i = \frac{\partial \mathcal{M}_i(x^b_{i-1}, \alpha^b)}{\partial x_{i-1}} \delta x_{i-1} + \frac{\partial \mathcal{M}_i(x^b_{i-1}, \alpha^b)}{\partial \alpha} \delta\alpha$$

Project the increments into smaller subspace

$$P^T \delta x_i = P^T \frac{\partial \mathcal{M}_i(x^b_{i-1}, \alpha^b)}{\partial x_{i-1}} P\, P^T \delta x_{i-1} + P^T \frac{\partial \mathcal{M}_i(x^b_{i-1}, \alpha^b)}{\partial \alpha} \delta\alpha$$

POD Model Reduced 4D-Var

Incremental equation

$$\delta x_i = \frac{\partial \mathcal{M}_i(x^b_{i-1}, \alpha^b)}{\partial x_{i-1}} \delta x_{i-1} + \frac{\partial \mathcal{M}_i(x^b_{i-1}, \alpha^b)}{\partial \alpha} \delta \alpha$$

Project the increments into smaller subspace

$$\underbrace{P^T \delta x_i}_{\delta z_i} = P^T \frac{\partial \mathcal{M}_i(x^b_{i-1}, \alpha^b)}{\partial x_{i-1}} P \underbrace{P^T \delta x_{i-1}}_{\delta z_{i-1}} + P^T \frac{\partial \mathcal{M}_i(x^b_{i-1}, \alpha^b)}{\partial \alpha} \delta \alpha$$

POD Model Reduced 4D-Var

Incremental equation

$$\delta x_i = \frac{\partial \mathcal{M}_i(x^b_{i-1}, \alpha^b)}{\partial x_{i-1}} \delta x_{i-1} + \frac{\partial \mathcal{M}_i(x^b_{i-1}, \alpha^b)}{\partial \alpha} \delta \alpha$$

Project the increments into smaller subspace

$$\underbrace{P^T \delta x_i}_{\delta z_i} = P^T \frac{\partial \mathcal{M}_i(x^b_{i-1}, \alpha^b)}{\partial x_{i-1}} P \underbrace{P^T \delta x_{i-1}}_{\delta z_{i-1}} + P^T \frac{\partial \mathcal{M}_i(x^b_{i-1}, \alpha^b)}{\partial \alpha} \delta \alpha$$

$\delta z_i = P^T \delta x_i$ increment of the state in the reduced space

POD Model Reduced 4D-Var

Incremental equation

$$\delta x_i = \frac{\partial \mathcal{M}_i(x_{i-1}^b, \alpha^b)}{\partial x_{i-1}} \delta x_{i-1} + \frac{\partial \mathcal{M}_i(x_{i-1}^b, \alpha^b)}{\partial \alpha} \delta \alpha$$

Project the increments into smaller subspace

$$\underbrace{P^T \delta x_i}_{\delta z_i} = P^T \frac{\partial \mathcal{M}_i(x_{i-1}^b, \alpha^b)}{\partial x_{i-1}} P \underbrace{P^T \delta x_{i-1}}_{\delta z_{i-1}} + P^T \frac{\partial \mathcal{M}_i(x_{i-1}^b, \alpha^b)}{\partial \alpha} \delta \alpha$$

$\delta z_i = P^T \delta x_i$ increment of the state in the reduced space

$\frac{\partial \mathcal{M}_i(x_{i-1}^b, \alpha^b)}{\partial x_{i-1}} p \simeq \frac{\mathcal{M}_i(x_{i-1}^b + \epsilon p, \alpha^b) - \mathcal{M}_i(x_{i-1}^b, \alpha^b)}{\epsilon}$ **directional derivative approximation**

How to get matrix P?

How to project into a smaller subspace, such that the most important dynamics of the system are kept?

How to get matrix P?

How to project into a smaller subspace, such that the most important dynamics of the system are kept?

STEP 1: Generate ensemble of perturbed model simulations

$$\begin{array}{lcl} \alpha^b + \Delta\alpha_1 & \rightarrow & x_1^{\Delta 1}, \; x_2^{\Delta 1}, \; \ldots \; x_n^{\Delta 1} \\ \alpha^b + \Delta\alpha_2 & \rightarrow & x_1^{\Delta 2}, \; x_2^{\Delta 2}, \; \ldots \; x_n^{\Delta 2} \end{array}$$

How to get matrix P?

How to project into a smaller subspace, such that the most important dynamics of the system are kept?

STEP 1: Generate ensemble of perturbed model simulations

$$\begin{array}{lcllll} \alpha^b + \Delta\alpha_1 & \rightarrow & x_1^{\Delta 1}, & x_2^{\Delta 1}, & \ldots & x_n^{\Delta 1} \\ \alpha^b + \Delta\alpha_2 & \rightarrow & x_1^{\Delta 2}, & x_2^{\Delta 2}, & \ldots & x_n^{\Delta 2} \end{array}$$

STEP 2: Create a covariance matrix

$$\mathbf{C_X} = \mathbf{\Delta X}\ \mathbf{\Delta X}^{\mathbf{T}}/(\mathbf{n}-\mathbf{1})$$

How to get matrix P?

How to project into a smaller subspace, such that the most important dynamics of the system are kept?

STEP 1: Generate ensemble of perturbed model simulations

$$\begin{array}{lcl} \alpha^b + \Delta\alpha_1 & \rightarrow & x_1^{\Delta 1}, \quad x_2^{\Delta 1}, \quad \ldots \quad x_n^{\Delta 1} \\ \alpha^b + \Delta\alpha_2 & \rightarrow & x_1^{\Delta 2}, \quad x_2^{\Delta 2}, \quad \ldots \quad x_n^{\Delta 2} \end{array}$$

STEP 2: Create a covariance matrix

$$\mathbf{C_X} = \mathbf{\Delta X}\ \mathbf{\Delta X}^{\mathbf{T}}/(\mathbf{n}-\mathbf{1})$$

STEP 3: Decompose $\mathbf{C_X}$ with eigenvalue decomposition

$$\mathbf{\Delta X}\ \mathbf{\Delta X}^{\mathbf{T}}/(\mathbf{n}-\mathbf{1}) = \mathbf{P}\mathbf{D}\mathbf{P}^{\mathbf{T}}$$

P - eigenvectors, **D** - diagonal matrix with eigenvalues

How to get matrix P?

Covariance matrix of ΔX:

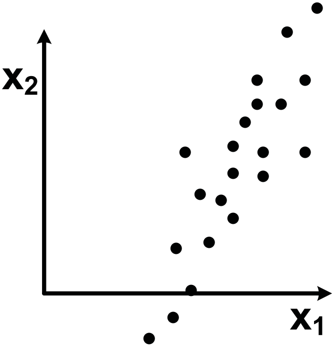
$$\Delta X\ \Delta X^T/(n-1) = PDP^T$$

$$P^T\Delta X\ (P^T\Delta X)^T/(n-1) = D$$

$$\Delta Z\ \Delta Z^T/(n-1) = D$$

Covariance matrix of ΔZ:

$$C_Z = D$$



How to get matrix P?

Covariance matrix of ΔX:

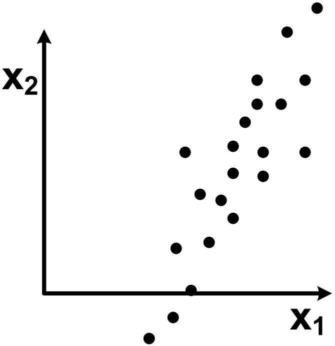
$$\Delta X \, \Delta X^T/(n-1) = PDP^T$$

$$P^T \Delta X \, (P^T \Delta X)^T/(n-1) = D$$

$$\Delta Z \, \Delta Z^T/(n-1) = D$$

Covariance matrix of ΔZ:

$$C_Z = D$$



How to get matrix P?

Covariance matrix of ΔX:

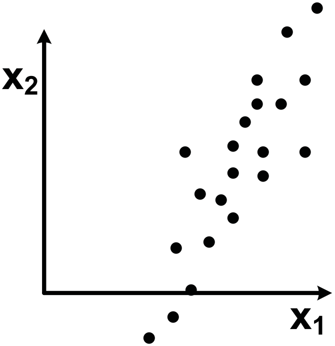
$$\Delta X \, \Delta X^T/(n-1) = PDP^T$$

$$P^T \Delta X \, (P^T \Delta X)^T/(n-1) = D$$

$$\Delta Z \, \Delta Z^T/(n-1) = D$$

Covariance matrix of ΔZ:

$$C_Z = D$$



How to get matrix P?

Covariance matrix of ΔX:

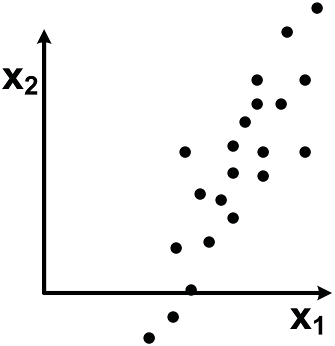
$$\Delta X\ \Delta X^T/(n-1) = PDP^T$$

$$P^T \Delta X\ (P^T \Delta X)^T/(n-1) = D$$

$$\Delta Z\ \Delta Z^T/(n-1) = D$$

Covariance matrix of ΔZ:

$$C_Z = D$$



How to get matrix P?

Covariance matrix of ΔX:

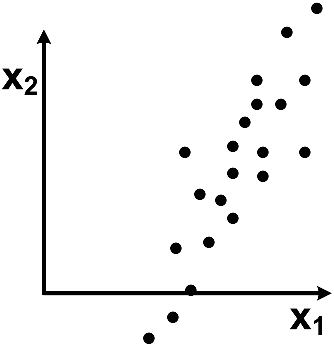
$$\Delta X \, \Delta X^T/(n-1) = PDP^T$$

$$P^T\Delta X \, (P^T\Delta X)^T/(n-1) = D$$

$$\Delta Z \, \Delta Z^T/(n-1) = D$$

Covariance matrix of ΔZ:

$$C_Z = D$$



How to get matrix P?

Covariance matrix of ΔX:

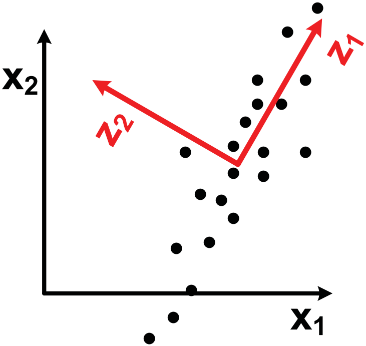
$$\Delta X \, \Delta X^T/(n-1) = PDP^T$$

$$P^T \Delta X \, (P^T \Delta X)^T/(n-1) = D$$

$$\Delta Z \, \Delta Z^T/(n-1) = D$$

Covariance matrix of ΔZ:

$$C_Z = D$$



How to get matrix P?

Covariance matrix of ΔX:

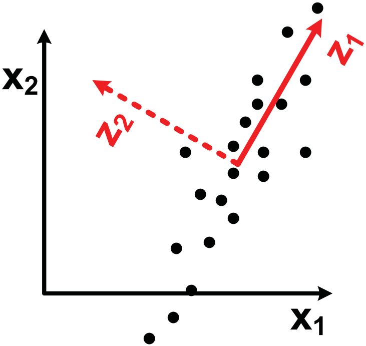
$$\Delta X\ \Delta X^T/(n-1) = PDP^T$$

$$P^T \Delta X\ (P^T \Delta X)^T/(n-1) = D$$

$$\Delta Z\ \Delta Z^T/(n-1) = D$$

Covariance matrix of ΔZ:

$$C_Z = D$$



How to get matrix P?

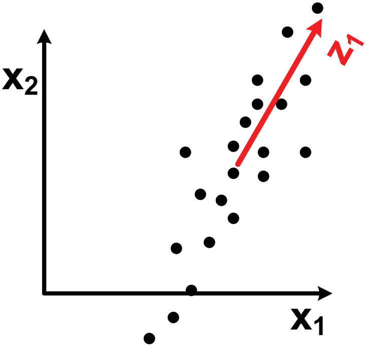
Covariance matrix of ΔX:

$$\Delta X\ \Delta X^T/(n-1) = PDP^T$$

$$P^T\Delta X\ (P^T\Delta X)^T/(n-1) = D$$

$$\Delta Z\ \Delta Z^T/(n-1) = D$$

Covariance matrix of ΔZ:
$$C_Z = D$$



How to get matrix P?

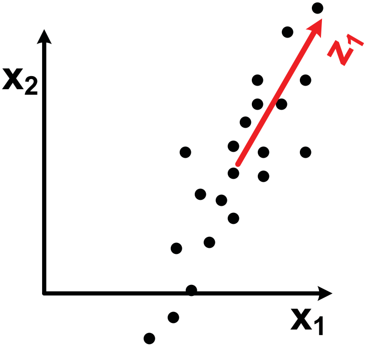
Covariance matrix of ΔX:

$$\Delta X \, \Delta X^T/(n-1) = PDP^T$$

$$P^T \Delta X \, (P^T \Delta X)^T/(n-1) = D$$

$$\Delta Z \, \Delta Z^T/(n-1) = D$$

Covariance matrix of ΔZ:
$C_Z = D$



The matrix **P**

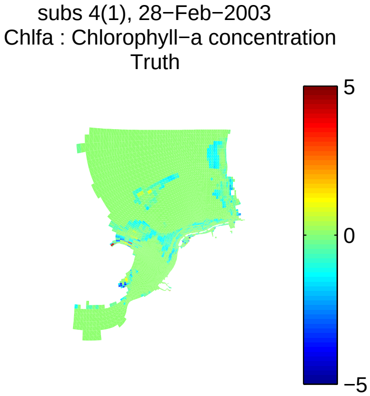
Selection of the eigenvectors corresponding to the highest eigenvalues will create the projection matrix **P**

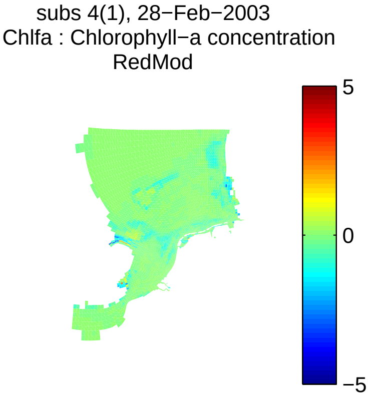
Results POD-Reduced Model

Experiment set up

parameter:	**ExtVLIM** Extinction of visible light due to inorganic matter
parameter background	0.025
parameter perturbed	0.015, 0.035
results shown for	0.03
starting time:	**26-Feb-2003**
time of simulation:	**14 days**
time step:	**1 day**
number of snapshots:	**26**
energy recovered	90 %

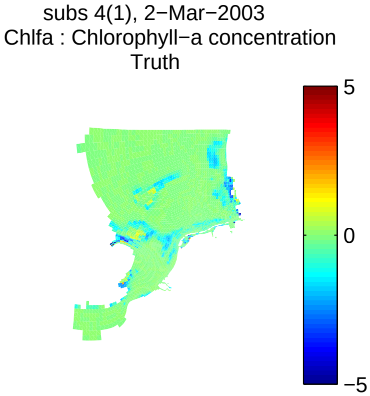
Results POD-Reduced Model

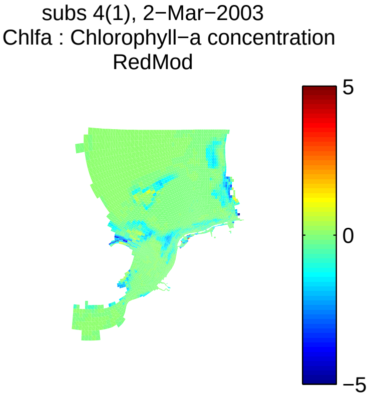




Increment of the reduced model compared with the **increment** of the true model, comparison for the Chlorophyll-a concentration

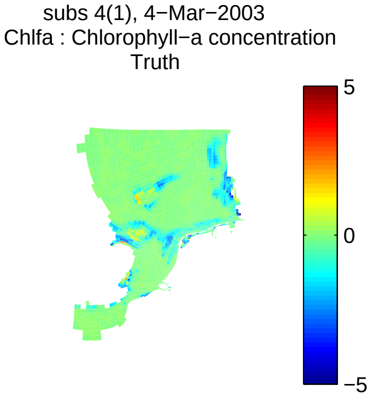
Results POD-Reduced Model

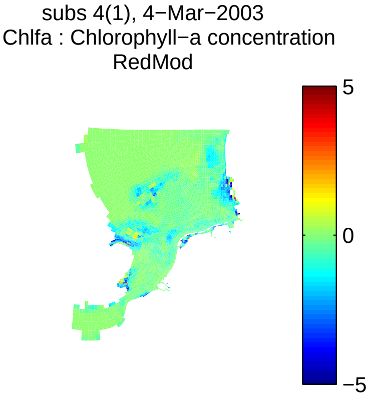




Increment of the reduced model compared with the **increment** of the true model, comparison for the Chlorophyll-a concentration

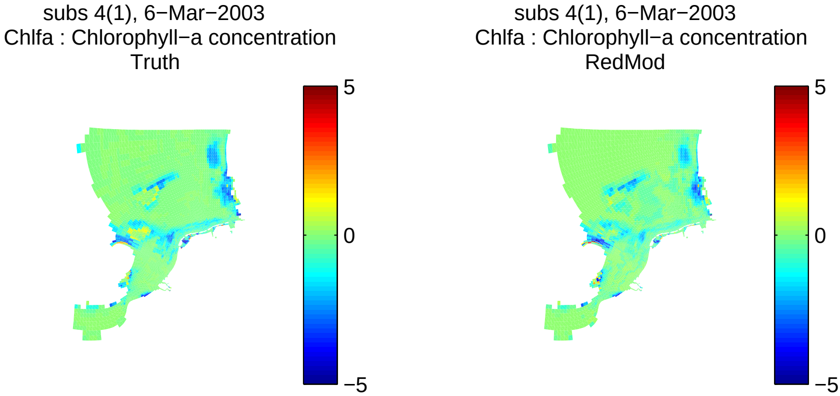
Results Reduced Model





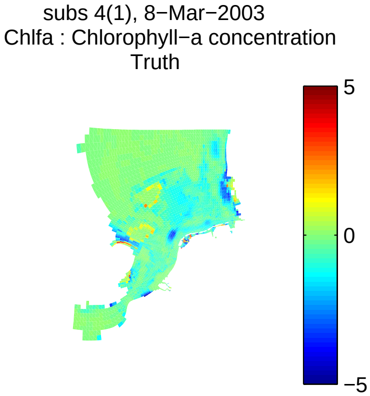
Increment of the reduced model compared with the **increment** of the true model, comparison for the Chlorophyll-a concentration

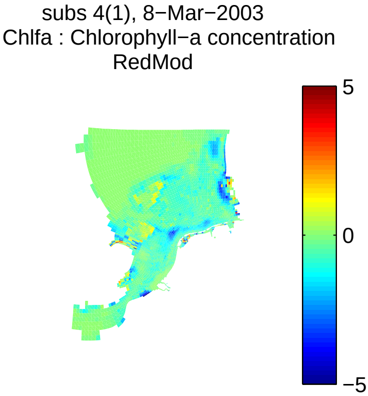
Results Reduced Model



Increment of the reduced model compared with the **increment** of the true model, comparison for the Chlorophyll-a concentration

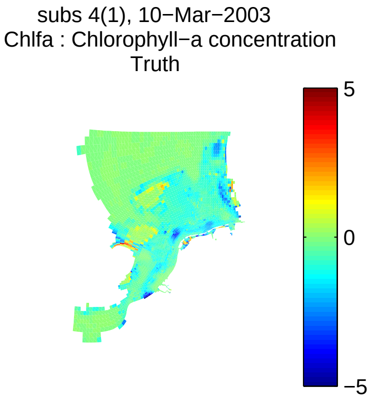
Results Reduced Model

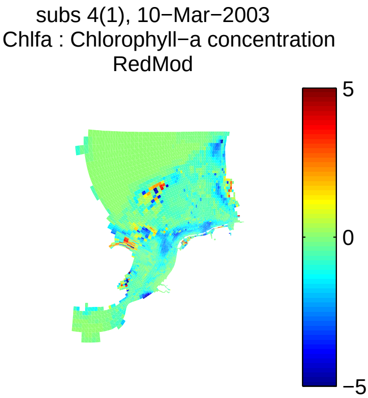




Increment of the reduced model compared with the **increment** of the true model, comparison for the Chlorophyll-a concentration

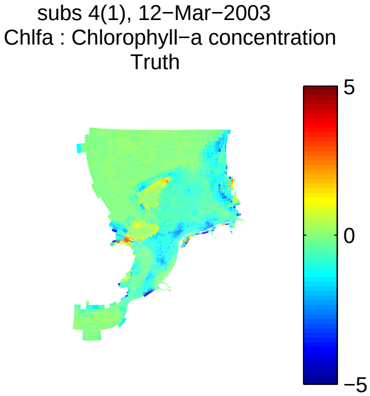
Results Reduced Model

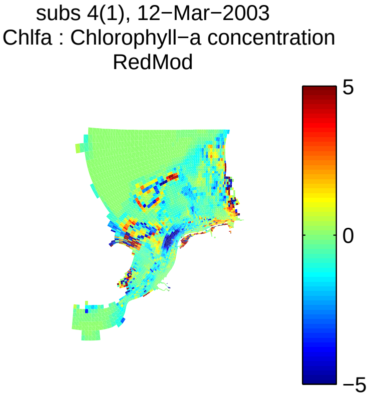




Increment of the reduced model compared with the **increment** of the true model, comparison for the Chlorophyll-a concentration

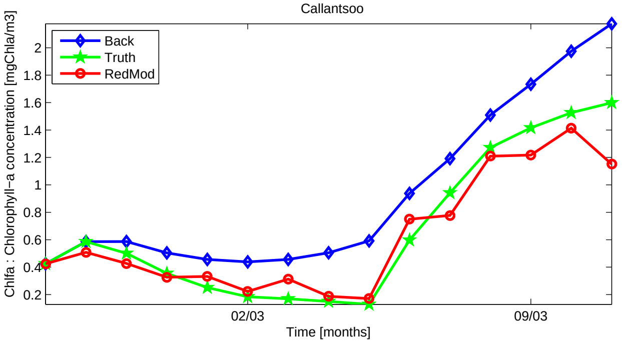
Results Reduced Model





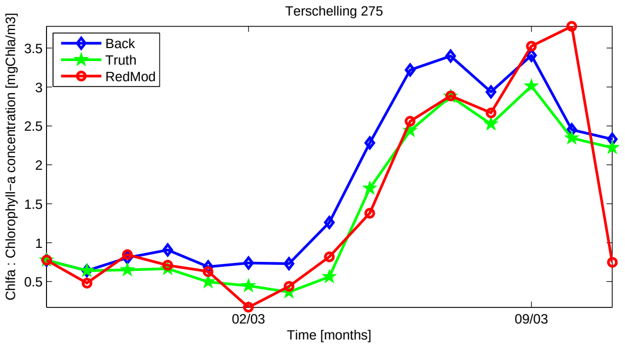
Increment of the reduced model compared with the **increment** of the true model, comparison for the Chlorophyll-a concentration

Results Reduced Model - stations



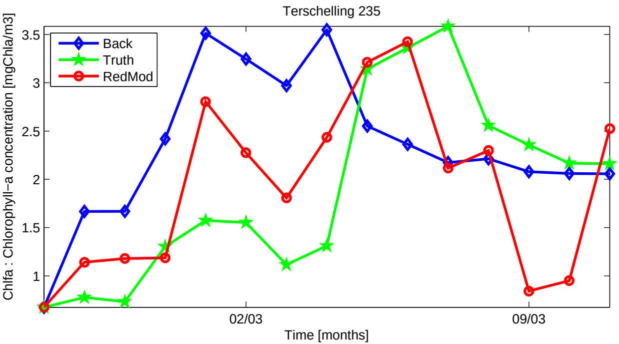
Chlorophyll-a concentration within time at station Callantsoo. Comparison between background run of the model, reduced model for a chosen parameter and true model for same parameter.

Results Reduced Model - stations



Chlorophyll-a concentration within time at station Terschelling 275km. Comparison between background run of the model, reduced model for a chosen parameter and true model for same parameter.

Results Reduced Model - stations



Chlorophyll-a concentration within time at station Terschelling 235km. Comparison between background run of the model, reduced model for a chosen parameter and true model for same parameter.

Summary

- POD-Reduced Forward model:
  - captures very well the behavior of the original model
- POD-Reduced Adjoint model approximation:
  - reasonably accurate
  - feasible in time
  - does not need exact derivatives

Summary

- POD-Reduced Forward model:
  - captures very well the behavior of the original model
- POD-Reduced Adjoint model approximation:
  - reasonably accurate
  - feasible in time
  - does not need exact derivatives

Future work

- 4D Variational Assimilation using the POD-Reduced model approximations (POD-Model Reduced 4D-Var)
  - Twin experiment
  - Real data case (satellite and in-situ measurements)

Thank you for your attention!

Questions?