Parameter Estimation in a Large Scale Dutch Continental Shelf Model

M. Umer Altaf [1], **Julius Sumihar** [2]
Martin Verlaan [1,2], **Arnold Heemink** [1]
[1] Delft University of Technology, Netherlands
[2] Deltares, Netherlands

Presented at: JONSMOD meeting, Deltares, May 10, 2010

Outline

- The Dutch Continental Shelf Model DCSM
- Variational Data Assimilation
- POD Reduced Order Modeling
- Ensemble Approach
- Calibration Experiment and Results
- Conclusions

The DCSM

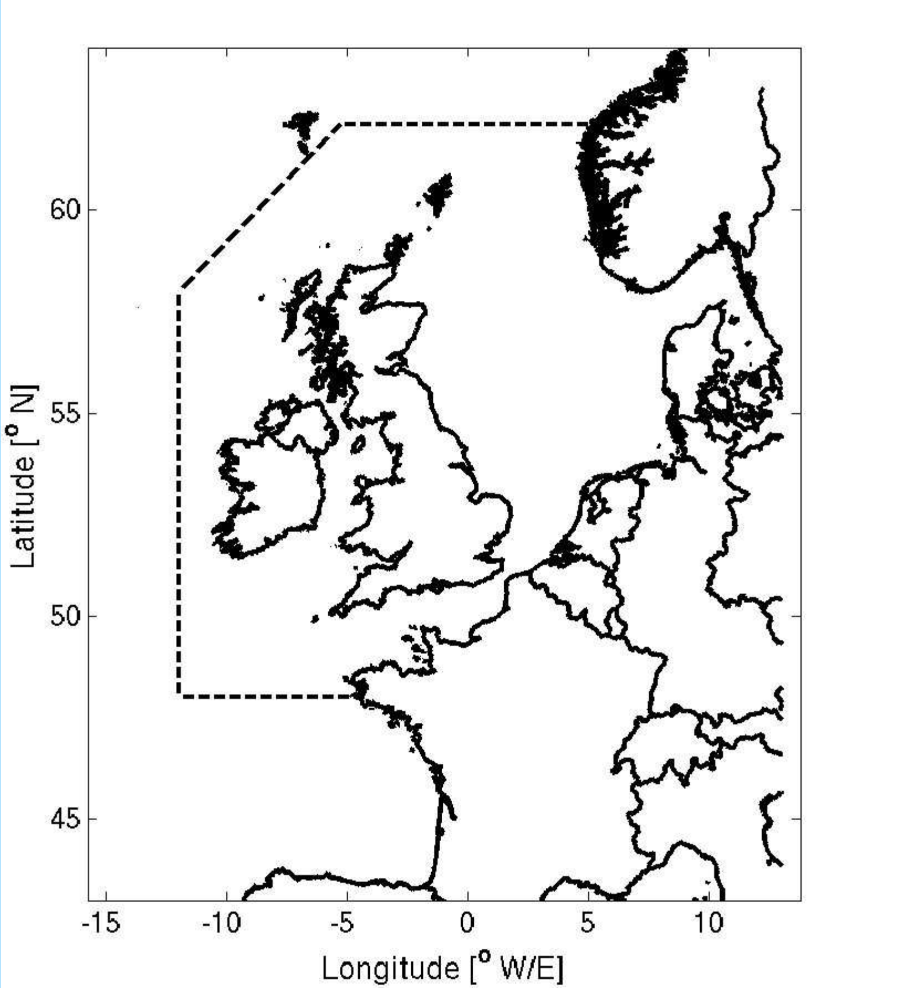
- Large part of the area lies below mean sea water level
- 1 Feb 1953: severe storm surge, casualities in southwestern part
- Delta project: dikes, moveable surge barriers at the entrance of Harbor
- DCSM is used in the Netherlands for Storm surge warning service

DCSM (v6)

- DCSM version 6 is the recently designed water level model
- Covers a much larger deep water area than the operational DCSM
- Spatial res. that is a factor 5 finer in both lat. and long. directions
- Objectives:

  Extend the time horizon of the water level forecasts
  Forecasts for a dense distribution of locations along the Dutch coast

DCSM (v6)



Grid size: 1/40° × 1/60° (~2 km)

Grids: 1121 x 1261

Oper. grid points: 869544

Time step: 2 minutes

Variational Data Assimilation

- Consider a nonlinear discrete model for the state vector $x \in \Re^n$ from time t_i to time t_{i+1} is given by;

$$\mathbf{x}(t_{i+1}) = M_i[\mathbf{x}(t_i), \gamma] \tag{1}$$

$$\mathbf{y}(t_i) = H[\mathbf{x}(t_i)] \tag{2}$$

$H : R^n \rightarrow R^q$ is an operator that maps the model fields on observation space with $\mathbf{y} \in R^q$

Variational Data Assimilation

- Control(cost) function is introduced

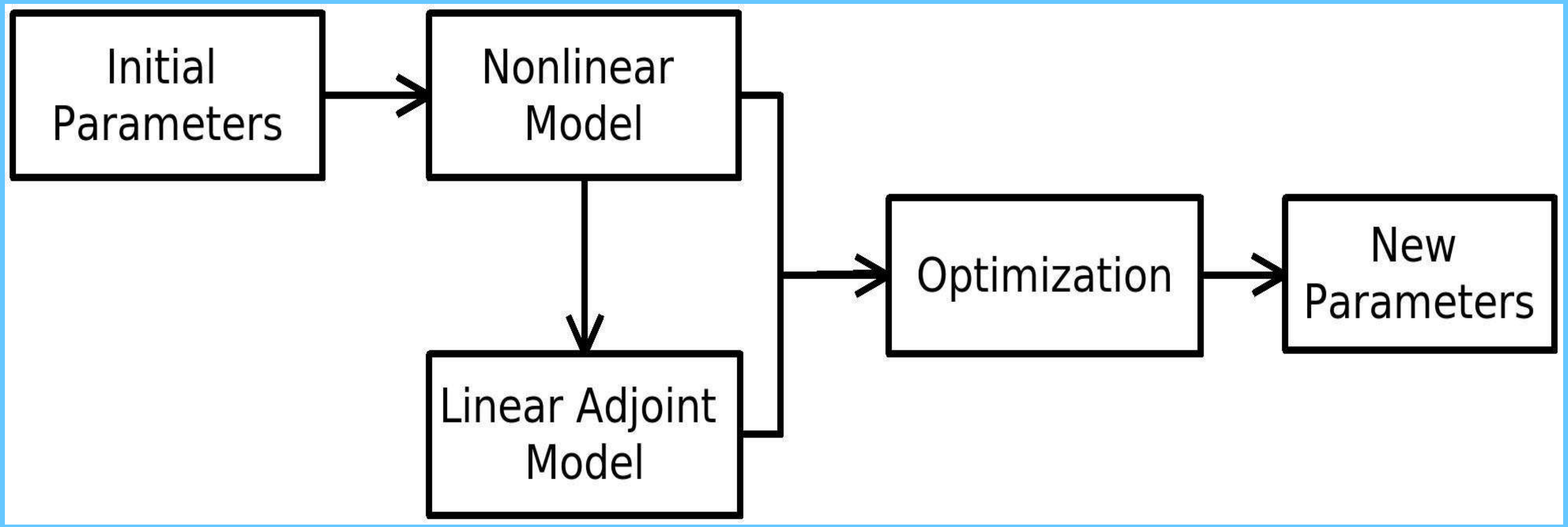
$$J(\gamma) = \sum_{i=1} [\mathbf{y}(t_i) - H(\mathbf{x}(t_i))]^T R^{-1} [\mathbf{y}(t_i) - H(\mathbf{x}(t_i))] \qquad (3)$$

The difference between data and simulation results is only due to measurement errors and incorrectly prescribed model parameters.

- Cost function is usually minimized using a gradient based algorithm which determines the gradient.

- Gradient is usually obtained by solving the adjoint problem.

Adjoint (References)



- Meteorology Courtier and Talagrand (1990)
- Oceonography Tziperman et.al (1992)
- Ground water Carrera and Neuman (1986)
- Shallow water Heemink et.al (2002), Lardner (1993)

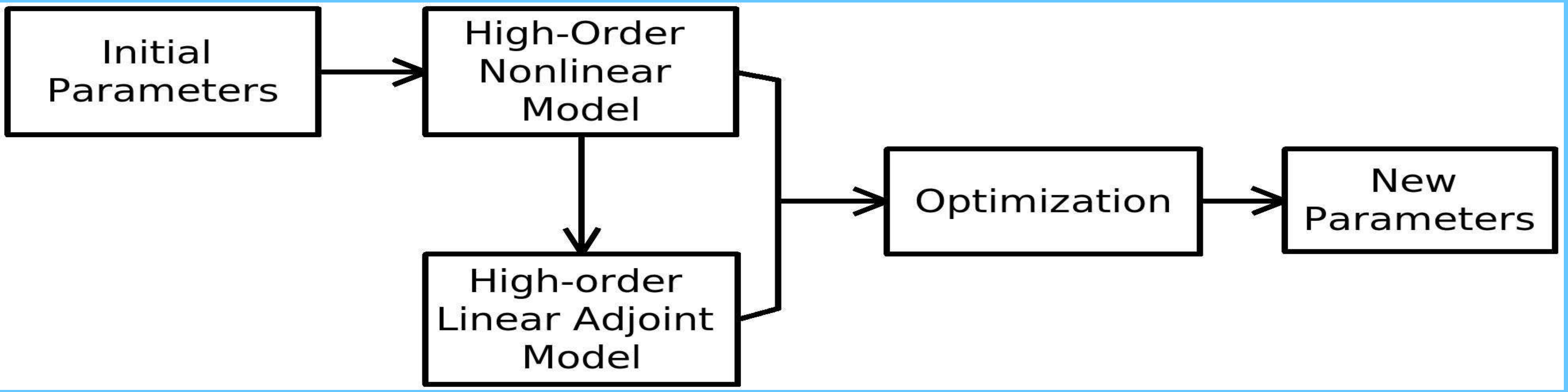
Adjoint Method

- Advantages
  - Adjoint method (ADJ) efficiently computes the gradient.
  - It is independent of the number of variables to be estimated.
  - Exact gradient
- Hurdles
  - Implementation
  - Memory

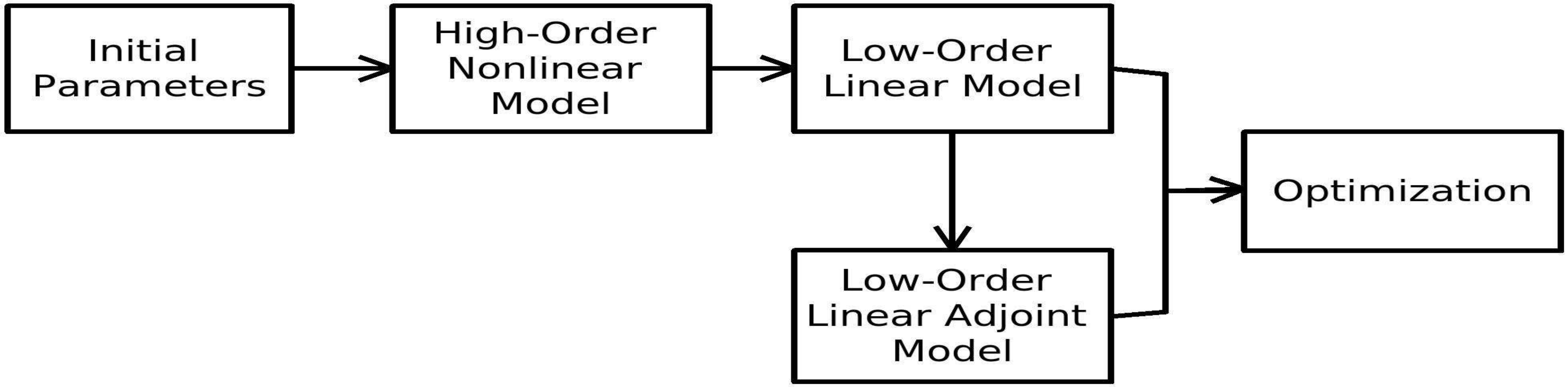
    The adjoint equation need to be integrated backward in time. The original problem must be stored for all time steps. The memory access will therefore be very huge for large scale problems

Comparison and Motivation

- Adjoint Method:

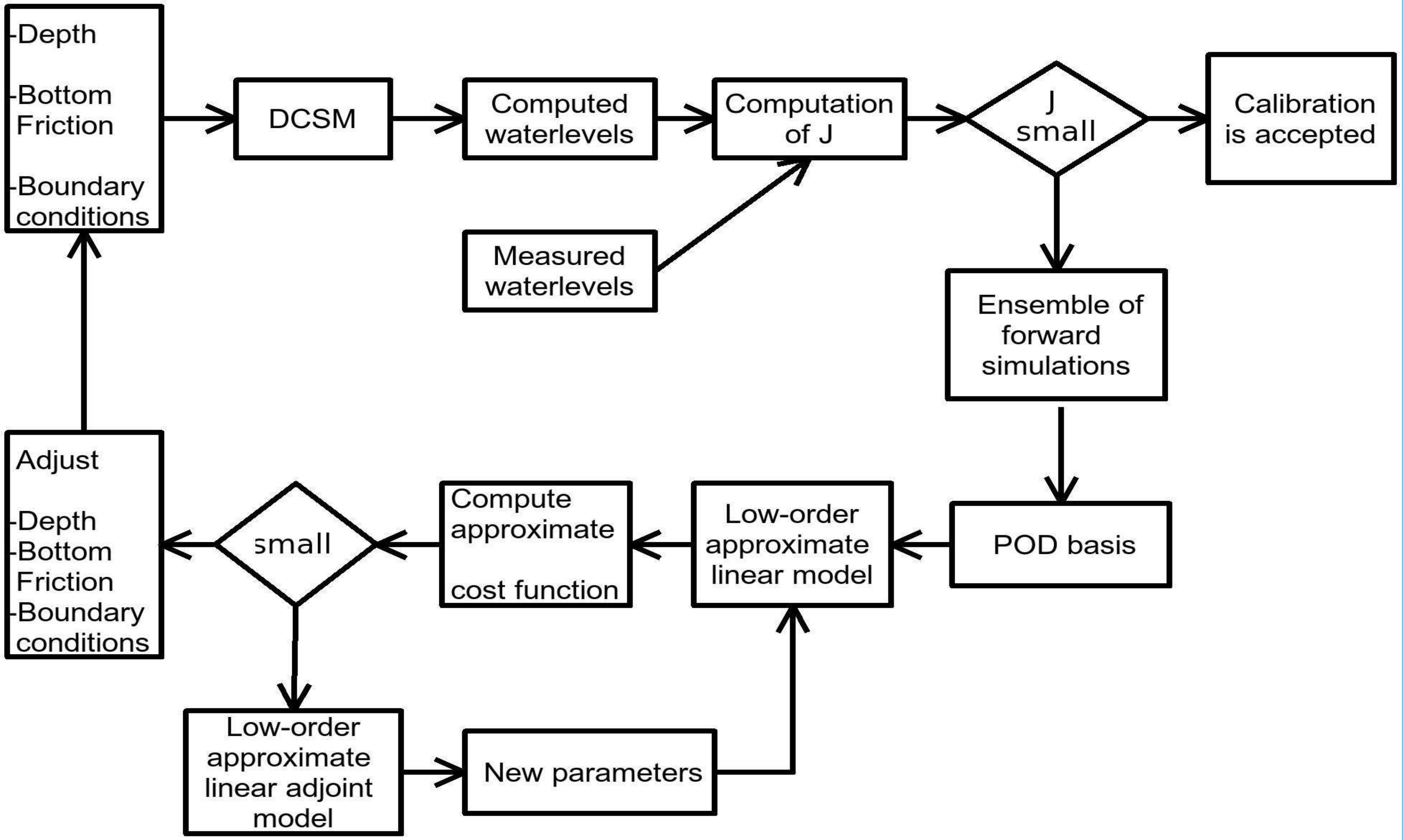


- POD Method:



DCSM

- One outer iteration with POD method (Altaf et al. 2009 (IJMSCE)):



Proper Orthogonal Decomposition (POD)

- Statistical tool to analyze experimental data:

  The POD is used to analyze the set of realizations with a view to extracting dominant features and trends (coherent structures called patterns in space)

- Reduced Order Modeling (ROM):

  The POD is used to provide a relevant set of basis functions with which we can identify a low-dimensional subspace on which to construct a model by projection of the governing equations

POD

- A set of s snapshots $E = \{e_1, e_2, \cdots, e_s\} \epsilon \Re^n$ are collected for some physical process taken at position e.

- Construct the covariance matrix Q $\epsilon \Re^{n \times n}$

$$Q = EE^T \tag{4}$$

- $P = \{p_1, p_2, p_3, \cdots\}$ are eigenvectors of a $n \times n$ eigenvalue problem with eigenvalues $\lambda_1 \gg \lambda_2 \gg \lambda_3 \cdots$

- Select the most dominant eigenmodes (patterns) based on the dominant eigenvalues λ_i

Ensemble Approach

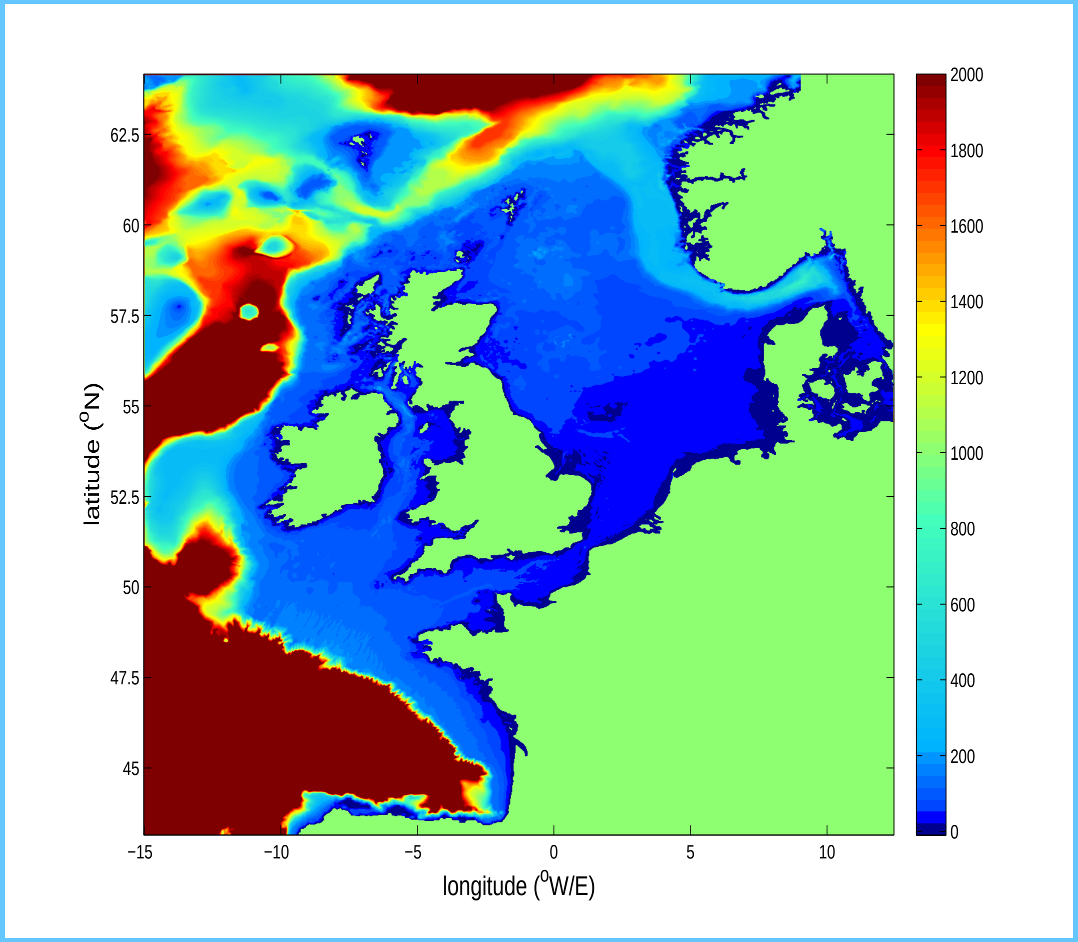
- An ensemble of snapshot vectors of the forward model simulations is collected.
- The snapshots are perturbations with repect to estimated parameters γ_k;

$$e_k(t_i) = \frac{\partial M_i[\mathbf{x}^b(t_{i-1}), \gamma_k]}{\partial \gamma_k} = \frac{M_i[\mathbf{x}^b(t_{i-1}), \gamma_k^b + \Delta\gamma_k] - M_i[\mathbf{x}^b(t_{i-1}), \gamma_k^b]}{\Delta\gamma_k} \quad (5)$$

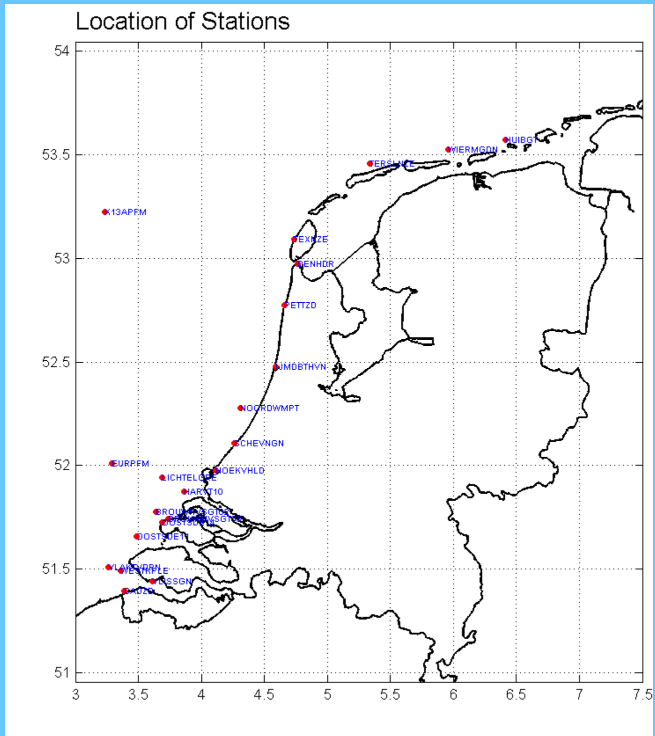
- A reduced POD basis is obtained on the basis of this ensemble.
- The dimension of reduce model is small than that of original model.
- Reduced model has linear characteristics. So it is easy to build a adjoint model for the computation of gradient.

DCSM

- The North sea is much shallower, with maximum depth around 200m
- English channel the depth are mostly less than 50m

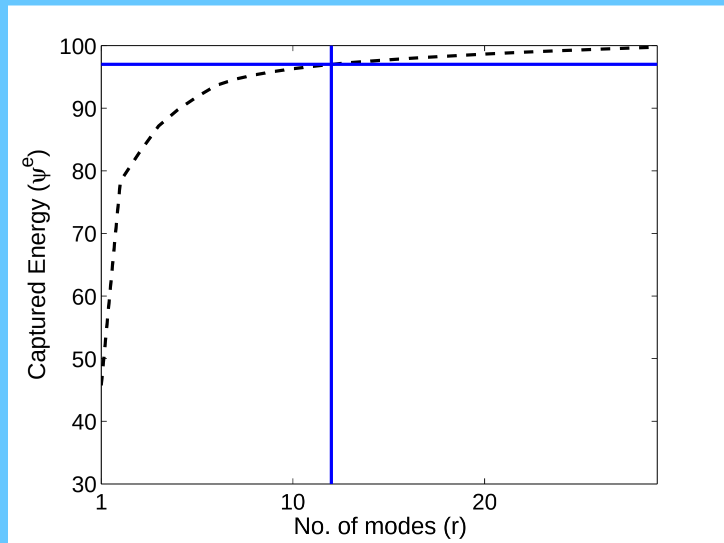


Experiment



Ensemble: 01 Jan 2007 to 04 Jan 2007 (33 snapshot vectors?)

13 POD modes are required to capture 97 % energy



Period: January 2007

stations: 22

additive corrections

results

- comparison of POD results with DUD method
- results are approximately equal
- Initial RMSE: 24.87

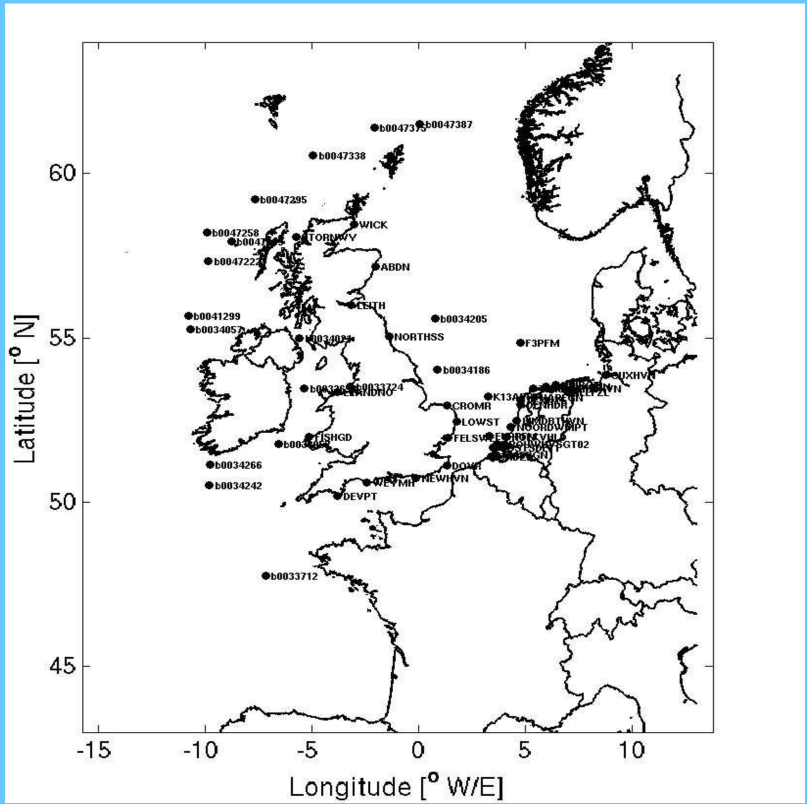
	POD	*DUD*
Adjustment(m)	2.30	2.56
RMSE(cm)	9.34	9.0
Simulations	3.5	5.0

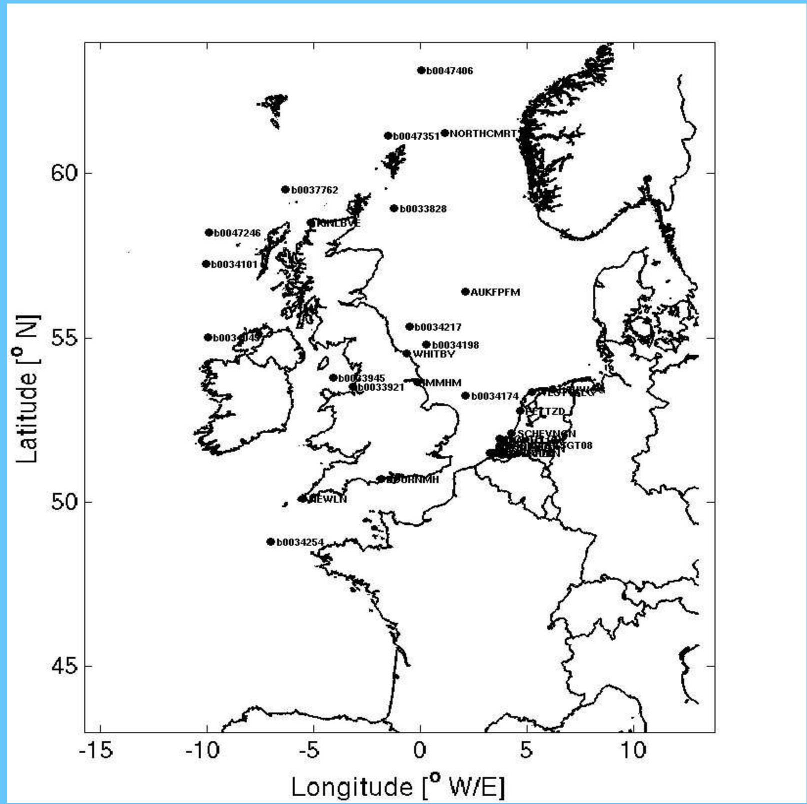
- encouragement: To investigate more parameters

results

StationName	*InitialRMSE*	*RMSE*
*Brouwhvsgt*02	0.256	0.10
*Brouwhvsgt*08	0.238	0.101
Cadzd	0.366	0.111
Denhdr	0.169	0.082
Eurpfm	0.200	0.096
*Harvt*10	0.248	0.105
Hoekvhld	0.210	0.120
Huibgt	0.244	0.076
Ijmd	0.206	0.085
*K*13	0.112	0.044
Lichtelgre	0.248	0.109
Noordwmpt	0.220	0.104
*Oostede*11	0.289	0.098
Pettzd	0.205	0.071
Schevngn	0.213	0.099

Calibration Experiment

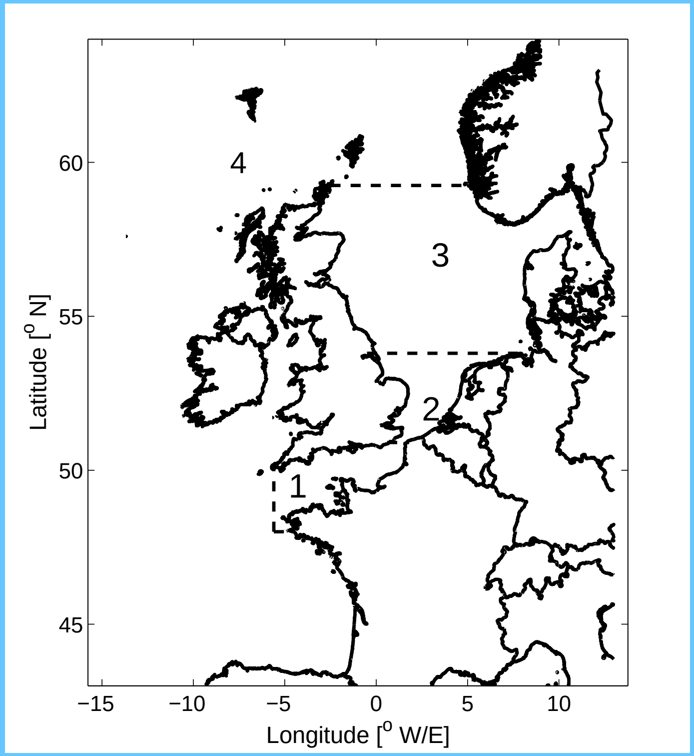




Assimilation st. 50

Validation st. 32

Experiment

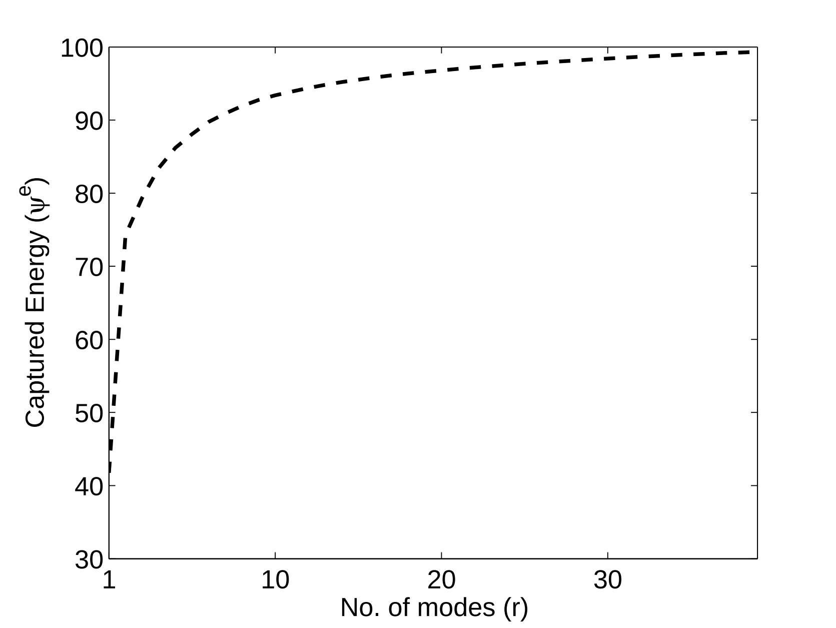


Divide model area in 4 sub-domains

relative adjustments

Ensemble size: 132 snapshot vectors?

24 POD modes are required to capture 97 % energy



- Same POD modes are used in first and 2nd outer iteration (β)

$Outeriteration(\beta)$	$Calibration$	$Validation$	
$Initial$	21.75	19.94	
I^{st}	14.74	13.22	
2^{nd}	12.98	11.72	

Table 1: Shows the results for the minimization with 97% energy.

$OuterIterations(\beta)$	$Calibration$	$Validation$	
$Initial$	21.75	19.94	
I^{st}	15.44	13.85	
2^{nd}	13.80	12.42	

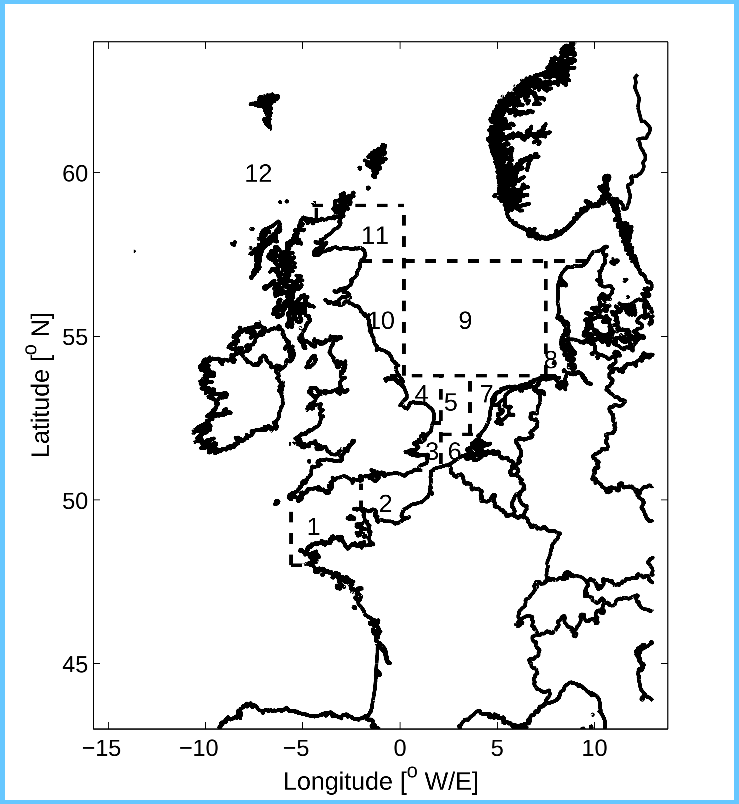
Table 2: Shows the results for the minimization with 90% energy.

results

- Comparison of results if POD modes are obtained with new ensemble in 2nd outer iteration
- results are slightly better but the cost of generating this new ensemble is huge, specially when the number of parameters are more.

	$Calibration$	$Validation$
$NewEnsemble$	12.53	11.41
$SameEnsemble$	12.98	11.72

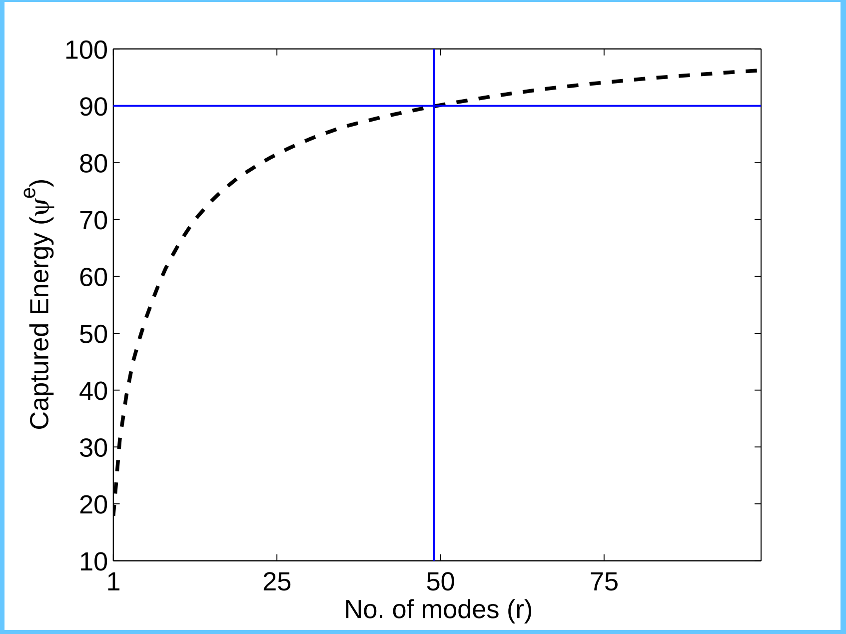
Experiment



Divide model area in 12 sub-domains

Ensemble size: 396 snapshot vectors?

49 POD modes are required to capture 90 % energy



results

- Similar results are obtained as compared with DUD method

- more efficient than DUD in this case

	Calibration	*Validation*	*Simulations*
POD	10.55	10.54	6
DUD	10.47	11.02	24

Table 1: Shows the comparison of results for the minimization.

Conclusions

- Classical method, adjoint of tangent linear model
- POD based method gives adjoint of linear reduce forward model
- Selection of boxes is very impotant for realistic results.
- The POD method is dependent on the number of parameters. If the number of parameters are too large, the size of ensemble is too big and it is difficult to find a good approximate model.
- The cost of ensemble in each outer iteration can be reduced by using the same ensemble.
- Next step is to implement POD method in OpenDA

THANK YOU

Ensemble Approach

- The reduced basis P is used to obtain approximate objective function:

$$J(\Delta\gamma) = \sum_{i=1}[\{\mathbf{y}(t_i) - H(\mathbf{x}^b(t_i))\} - \bar{H}\xi(t_i, \Delta\gamma)]^T$$

$$R^{-1}[\{\mathbf{y}(t_i) - H(\mathbf{x}^b(t_i))\} - \bar{H}\xi(t_i, \Delta\gamma)] \quad (6)$$

ξ is a reduce time-varing state vector;

$$\begin{pmatrix} \xi(t_i) \\ \Delta\gamma \end{pmatrix} = \begin{pmatrix} \widetilde{M}_i & \widetilde{M}_\gamma \\ 0 & I \end{pmatrix} \begin{pmatrix} \xi(t_{i-1}) \\ \Delta\gamma \end{pmatrix} \quad (7)$$

$\widetilde{M}_i$ and $\widetilde{M}_\gamma$ are reduced dynamics operators which are computed as:

$$\widetilde{M}_i = P^T \frac{\partial M_i}{\partial x^b(t_{i-1})} P \quad (8)$$

$$\widetilde{M}_\gamma = P^T (\frac{\partial M_i}{\partial \gamma_1}, \cdots, \frac{\partial M_i}{\partial \gamma_u}) \quad (9)$$