# POD Model Reduction of Large Scale Geophysical Models


Ionel M. Navon

School of Computational Science
Florida State University
Tallahassee, Florida

# Thanks to

- Prof. D. N. Daescu
  Dept. of Mathematics and Statistics,
  Portland State University, USA
- Prof. J. Zhu and Dr. Yanhua Cao
  Institute of Atmospheric Physics
  Chinese Academy of Science, Beijing, China
  and Prof. Z. D. Luo et al.
  Department of Mathematics
  Beijing Jiaotong University,Beijing, China
- Prof Chris. C. Pain , Dr. Fangxin Fang et al.
  Applied Modelling and Computation Group
  Imperial College, London, UK

# Thanks to

- Prof. D. N. Daescu
  Dept. of Mathematics and Statistics,
  Portland State University, USA
- Prof. J. Zhu and Dr. Yanhua Cao
  Institute of Atmospheric Physics
  Chinese Academy of Science, Beijing, China
  and Prof. Z. D. Luo et al.
  Department of Mathematics
  Beijing Jiaotong University,Beijing, China
- Prof Chris. C. Pain , Dr. Fangxin Fang et al.
  Applied Modelling and Computation Group
  Imperial College, London, UK

# Thanks to

- Prof. D. N. Daescu
  Dept. of Mathematics and Statistics,
  Portland State University, USA
- Prof. J. Zhu and Dr. Yanhua Cao
  Institute of Atmospheric Physics
  Chinese Academy of Science, Beijing, China
  and Prof. Z. D. Luo et al.
  Department of Mathematics
  Beijing Jiaotong University,Beijing, China
- Prof Chris. C. Pain , Dr. Fangxin Fang et al.
  Applied Modelling and Computation Group
  Imperial College, London, UK

# Outline

Dimension reduction means representing a vector in high dimensional space $x \in \mathbb{R}^n$ with a corresponding vector in a much lower dimensional space $\tilde{x} \in \mathbb{R}^m$.

- Consider the state equations

$$\mathbb{S} : \dot{x}(t) = f(x(t), u(t)), \; y(t) = h(x(t), u(t))$$

  - $x(\cdot) \in \mathbb{R}^n$ : state vector
  - $y(\cdot) \in \mathbb{R}^p$ : observation vector
  - $u(\cdot) \in \mathbb{R}^m$ : input vector

$$n \gg m, p$$

- Find $\tilde{\mathbb{S}} := (\tilde{f}, \tilde{h})$ with $\tilde{x}(t) \in \mathbb{R}^k$, $k \ll n$ assuring
  - Preservation of stability
  - Computational stability and efficient
  - Approximation error small-global error bound

Dimension reduction means representing a vector in high dimensional space $x \in \mathbb{R}^n$ with a corresponding vector in a much lower dimensional space $\tilde{x} \in \mathbb{R}^m$.

- Consider the state equations

$$\mathbb{S} : \dot{x}(t) = f(x(t), u(t)), \; y(t) = h(x(t), u(t))$$

  - $x(\cdot) \in \mathbb{R}^n$ : state vector
  - $y(\cdot) \in \mathbb{R}^p$ : observation vector
  - $u(\cdot) \in \mathbb{R}^m$ : input vector

$$n \gg m, p$$

- Find $\tilde{\mathbb{S}} := (\tilde{f}, \tilde{h})$ with $\tilde{x}(t) \in \mathbb{R}^k$, $k \ll n$ assuring
  - Preservation of stability
  - Computational stability and efficient
  - Approximation error small-global error bound

- POD is related to the principal component analysis, Karhunen-Loève expansion in the stochastic process theory, and principal of empirical orthogonal eigenfunctions.
- POD is the most used technique for the reduced-order modeling of nonlinear PDEs.
- POD proceeds by retaining the characteristics of the data set that contribute most to its variance.

## POD Generalities:

- POD is related to the principal component analysis, Karhunen-Loève expansion in the stochastic process theory, and principal of empirical orthogonal eigenfunctions.
- POD is the most used technique for the reduced-order modeling of nonlinear PDEs.
- POD proceeds by retaining the characteristics of the data set that contribute most to its variance.

# POD Generalities:

- POD is related to the principal component analysis, Karhunen-Loève expansion in the stochastic process theory, and principal of empirical orthogonal eigenfunctions.
- POD is the most used technique for the reduced-order modeling of nonlinear PDEs.
- POD proceeds by retaining the characteristics of the data set that contribute most to its variance.

## POD Principles:

- Find a projection $\mathbb{P}_r : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ of fixed rank $r$ that minimizes the error

$$\int_0^T \|x(t) - \mathbb{P}_r x(t)\|^2 dt, \; x(t) \in \mathbb{R}^n, \text{ with } 0 \leq t \leq T$$

- Introduce the symmetric, positive-semi-definite $n \times n$ matrix

$$\mathcal{C} = \int_0^T x(t) \, [x(t)]^T \, dt$$

- Solve the eigenvalue problem

$$\mathcal{C} \, \phi_k = \lambda_k \phi_k, \;\; k = 1, \cdots, n$$

with

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n, \text{ and } \int_0^T \phi_i \phi_j \, dt = \delta_{i,j}$$

- The optimal projection is $\mathbb{P}_r = \sum_{k=1}^{r} \phi_k \, [\phi_k]^T$.

## POD Principles:

- Find a projection $\mathbb{P}_r : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ of fixed rank $r$ that minimizes the error

$$\int_0^T \|x(t) - \mathbb{P}_r x(t)\|^2 dt, \; x(t) \in \mathbb{R}^n, \text{ with } 0 \le t \le T$$

- Introduce the symmetric, positive-semi-definite $n \times n$ matrix

$$\mathcal{C} = \int_0^T x(t) \, [x(t)]^T \, dt$$

- Solve the eigenvalue problem

$$\mathcal{C} \, \phi_k = \lambda_k \phi_k, \quad k = 1, \cdots, n$$

with

$$\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_n, \text{ and } \int_0^T \phi_i \phi_j \, dt = \delta_{i,j}$$

- The optimal projection is $\mathbb{P}_r = \sum_{k=1}^{r} \phi_k \, [\phi_k]^T$.

## POD Principles:

- Find a projection $\mathbb{P}_r : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ of fixed rank $r$ that minimizes the error

$$\int_0^T \|x(t) - \mathbb{P}_r x(t)\|^2 dt,\ x(t) \in \mathbb{R}^n,\ \text{with } 0 \le t \le T$$

- Introduce the symmetric, positive-semi-definite $n \times n$ matrix

$$\mathcal{C} = \int_0^T x(t)\,[x(t)]^T\,dt$$

- Solve the eigenvalue problem

$$\mathcal{C}\,\phi_k = \lambda_k \phi_k,\ \ k = 1, \cdots, n$$

with

$$\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_n,\ \text{and } \int_0^T \phi_i \phi_j\,dt = \delta_{i,j}$$

- The optimal projection is $\mathbb{P}_r = \sum_{k=1}^{r} \phi_k\,[\phi_k]^T$.

## POD Principles:

- Find a projection $\mathbb{P}_r : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ of fixed rank $r$ that minimizes the error

$$\int_0^T \|x(t) - \mathbb{P}_r x(t)\|^2 dt,\ x(t) \in \mathbb{R}^n,\ \text{with } 0 \leq t \leq T$$

- Introduce the symmetric, positive-semi-definite $n \times n$ matrix

$$\mathcal{C} = \int_0^T x(t)\,[x(t)]^T\,dt$$

- Solve the eigenvalue problem

$$\mathcal{C}\,\phi_k = \lambda_k \phi_k,\ \ k = 1, \cdots, n$$

with

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n,\ \text{and}\ \int_0^T \phi_i \phi_j\,dt = \delta_{i,j}$$

- The optimal projection is $\mathbb{P}_r = \sum_{k=1}^{r} \phi_k\,[\phi_k]^T$.

## POD Steps:

- Replace the set of data $x(t)$ by the snapshots $x(t_j)$ at discrete times $t_1, t_2, \cdots, t_m$.
- Transform the $n \times n$ eigenvalue problem into the $m \times m$ eigenvalue problem with $\mathcal{C}$ replaced by

$$\tilde{\mathcal{C}} = \sum_{j=1}^{m} x(t_j)\,[x(t_j)]^T\,\omega_i,\ \omega_i \text{ quadrature weights}$$

- Define the $n \times m$ matrix $\mathcal{X} = [x(t_1)\,x(t_2), \ldots, x(t_m)]$ and the $m \times m$ matrix $\mathcal{W} = diag\{\omega(t_1), \omega(t_2), \ldots, \omega(t_m)\}$
- The reduced eigenvalue problem becomes

$$\mathcal{X}^T \mathcal{W} \mathcal{X} u_k = \lambda_k u_k,\ \ u_k \in \mathbb{R}^m$$

- Replace the set of data $x(t)$ by the snapshots $x(t_j)$ at discrete times $t_1, t_2, \cdots, t_m$.
- Transform the $n \times n$ eigenvalue problem into the $m \times m$ eigenvalue problem with $\mathcal{C}$ replaced by

$$\tilde{\mathcal{C}} = \sum_{j=1}^{m} x(t_j)\,[x(t_j)]^T\,\omega_i, \;\; \omega_i \text{ quadrature weights}$$

- Define the $n \times m$ matrix $\mathcal{X} = [x(t_1)\; x(t_2), \ldots, x(t_m)]$ and the $m \times m$ matrix $\mathcal{W} = diag\{\omega(t_1), \omega(t_2), \ldots, \omega(t_m)\}$
- The reduced eigenvalue problem becomes

$$\mathcal{X}^T \mathcal{W} \mathcal{X} u_k = \lambda_k u_k, \;\; u_k \in \mathbb{R}^m$$

- Replace the set of data $x(t)$ by the snapshots $x(t_j)$ at discrete times $t_1, t_2, \cdots, t_m$.
- Transform the $n \times n$ eigenvalue problem into the $m \times m$ eigenvalue problem with $\mathcal{C}$ replaced by

$$\tilde{\mathcal{C}} = \sum_{j=1}^{m} x(t_j)\,[x(t_j)]^T\,\omega_i,\ \ \omega_i \text{ quadrature weights}$$

- Define the $n \times m$ matrix $\mathcal{X} = [x(t_1)\ x(t_2), \ldots, x(t_m)]$ and the $m \times m$ matrix $\mathcal{W} = diag\{\omega(t_1), \omega(t_2), \ldots, \omega(t_m)\}$
- The reduced eigenvalue problem becomes

$$\mathcal{X}^T \mathcal{W} \mathcal{X} u_k = \lambda_k u_k,\ \ u_k \in \mathbb{R}^m$$

# POD Steps:

- Replace the set of data $x(t)$ by the snapshots $x(t_j)$ at discrete times $t_1, t_2, \cdots, t_m$.
- Transform the $n \times n$ eigenvalue problem into the $m \times m$ eigenvalue problem with $\mathcal{C}$ replaced by

$$\tilde{\mathcal{C}} = \sum_{j=1}^{m} x(t_j)\,[x(t_j)]^T\,\omega_i,\ \omega_i \text{ quadrature weights}$$

- Define the $n \times m$ matrix $\mathcal{X} = [x(t_1)\ x(t_2), \ldots, x(t_m)]$ and the $m \times m$ matrix $\mathcal{W} = diag\{\omega(t_1), \omega(t_2), \ldots, \omega(t_m)\}$
- The reduced eigenvalue problem becomes

$$\mathcal{X}^T \mathcal{W} \mathcal{X} u_k = \lambda_k u_k, \quad u_k \in \mathbb{R}^m$$

- POD is sensitive to details of snapshots used
- POD is sensitive to the choice of inner products
- POD depends on how well the data ensemble captures the relevant system behavior
- POD sometimes yields unstable models despite the original system being stable
- POD does not take account of system outputs when performing the reduction, and hence the reduced-order models produced may be inefficient

# POD Shortcomings:

- POD is sensitive to details of snapshots used
- POD is sensitive to the choice of inner products
- POD depends on how well the data ensemble captures the relevant system behavior
- POD sometimes yields unstable models despite the original system being stable
- POD does not take account of system outputs when performing the reduction, and hence the reduced-order models produced may be inefficient

# POD Shortcomings:

- POD is sensitive to details of snapshots used
- POD is sensitive to the choice of inner products
- POD depends on how well the data ensemble captures the relevant system behavior
- POD sometimes yields unstable models despite the original system being stable
- POD does not take account of system outputs when performing the reduction, and hence the reduced-order models produced may be inefficient

# POD Shortcomings:

- POD is sensitive to details of snapshots used
- POD is sensitive to the choice of inner products
- POD depends on how well the data ensemble captures the relevant system behavior
- POD sometimes yields unstable models despite the original system being stable
- POD does not take account of system outputs when performing the reduction, and hence the reduced-order models produced may be inefficient

# POD Shortcomings:

- POD is sensitive to details of snapshots used
- POD is sensitive to the choice of inner products
- POD depends on how well the data ensemble captures the relevant system behavior
- POD sometimes yields unstable models despite the original system being stable
- POD does not take account of system outputs when performing the reduction, and hence the reduced-order models produced may be inefficient

- Four dimensional variational data assimilation is in principle a least-squares fit in 4 dimensions between the predicted state of the atmosphere and the observations.
  The adjustment to the predicted state is made at the initial time $t_0$, which ensures that the analysis state (4-dimensional) is a model trajectory .
- 4D-Var is a method of estimating a set of parameters by optimizing the fit between the solution of the model and a set of observations which the model is meant to predict. In this context, the procedure of adjusting the parameters until the model 'best predicts' the observables, is known as optimization.

# 4D-Var Principles:

- Four dimensional variational data assimilation is in principle a least-squares fit in 4 dimensions between the predicted state of the atmosphere and the observations.
  The adjustment to the predicted state is made at the initial time $t_0$, which ensures that the analysis state (4-dimensional) is a model trajectory .
- 4D-Var is a method of estimating a set of parameters by optimizing the fit between the solution of the model and a set of observations which the model is meant to predict. In this context, the procedure of adjusting the parameters until the model 'best predicts' the observables, is known as optimization.

# 4D-Var data assimilation Results:

- Given
  - $x, y, x_b$ state, observation and background vectors,
  - $\mathcal{M}, \mathcal{H}$ model and observation operators,
  - $\mathfrak{B}, \mathfrak{R}$ background and observational error covariance matrices,
- Find an optimal estimate (analysis) state vector $x^a$ solution of

$$\min_{x \in \mathbb{R}^n} \mathscr{J}(x); \quad x^a = \arg\min \mathscr{J}$$

where the cost function $\mathscr{J}$ is

$$\mathscr{J} = \frac{1}{2}(x - x_b)^T \mathfrak{B}(x - x_b) + \frac{1}{2}\left[y - \mathcal{H}(x)\right]^T \mathfrak{R}^{-1}\left[y - \mathcal{H}(x)\right]$$

or the discrete form

$$\mathscr{J} = \frac{1}{2}(x - x_b)^T \mathfrak{B}(x - x_b) + \frac{1}{2}\sum_{k=1}^{p}\left[y_k - \mathcal{H}_k(\mathcal{M}_k(x))\right]^T \mathfrak{R}_k^{-1}\left[y_k - \mathcal{H}_k(\mathcal{M}_k(x))\right]$$

- Given
  - $x, y, x_b$ state, observation and background vectors,
  - $\mathcal{M}, \mathcal{H}$ model and observation operators,
  - $\mathfrak{B}, \mathfrak{R}$ background and observational error covariance matrices,
- Find an optimal estimate (analysis) state vector $x^a$ solution of

$$\min_{x \in \mathbb{R}^n} \mathscr{J}(x); \quad x^a = \arg\min \mathscr{J}$$

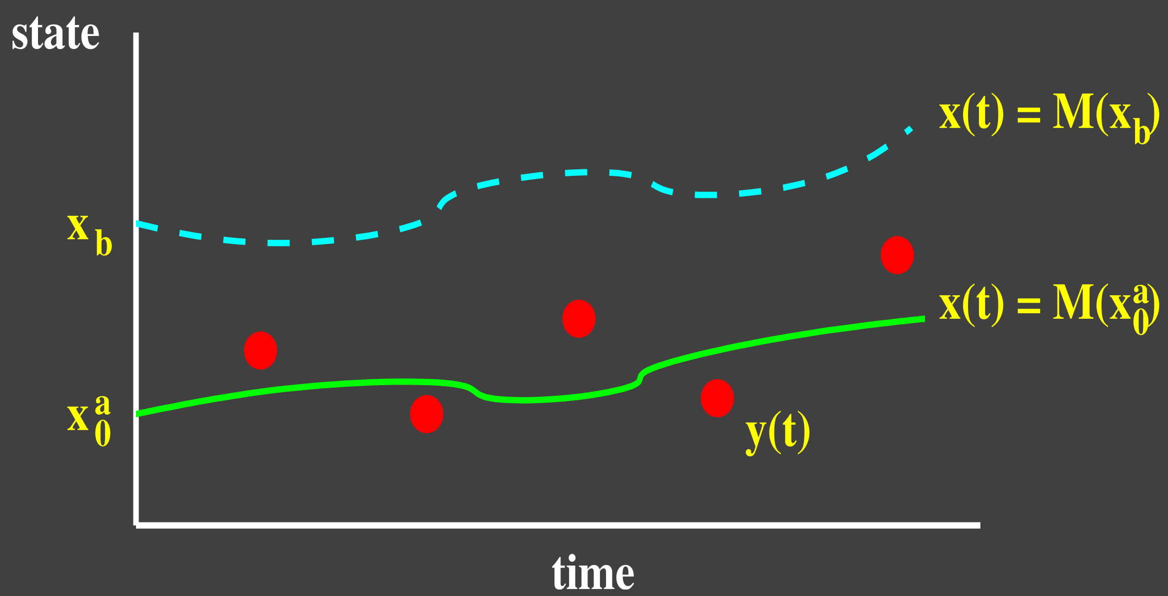
where the cost function $\mathscr{J}$ is

$$\mathscr{J} = \frac{1}{2}(x - x_b)^T \mathfrak{B}(x - x_b) + \frac{1}{2}\left[y - \mathcal{H}(x)\right]^T \mathfrak{R}^{-1}\left[y - \mathcal{H}(x)\right]$$

or the discrete form

$$\mathscr{J} = \frac{1}{2}(x - x_b)^T \mathfrak{B}(x - x_b) + \frac{1}{2}\sum_{k=1}^{p}\left[y_k - \mathcal{H}_k(\mathcal{M}_k(x))\right]^T \mathfrak{R}_k^{-1}\left[y_k - \mathcal{H}_k(\mathcal{M}_k(x))\right]$$

# Four-dimensional variational data assimilation

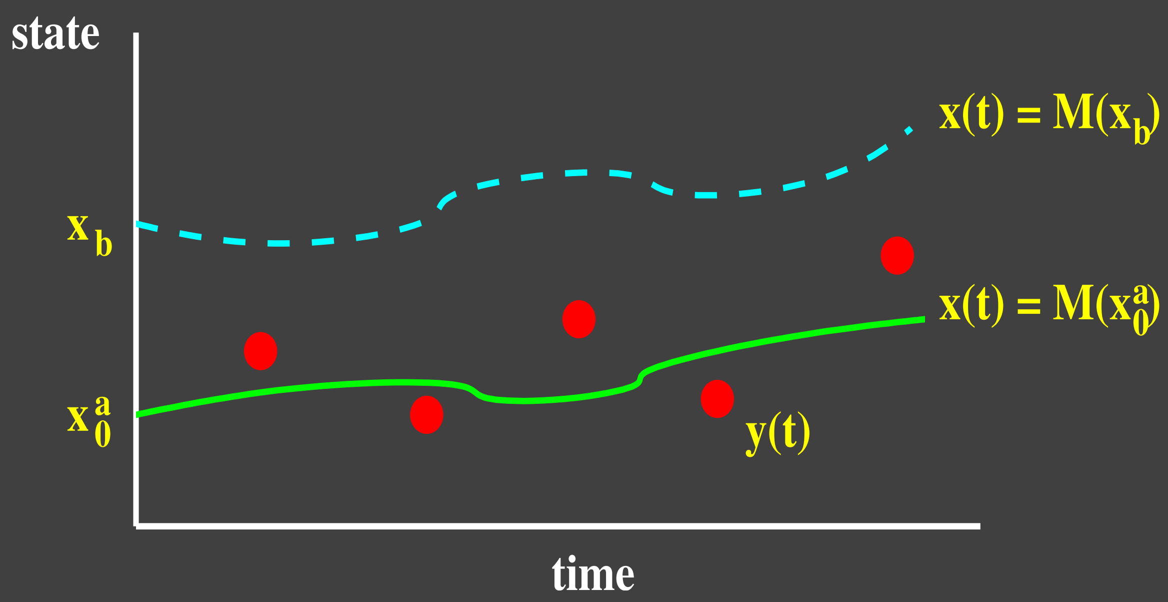



$$\min_{\mathbf{x}_0 \in R^m} \mathcal{J}(\mathbf{x}_0)$$

$$\mathbf{x}_0^a = \text{Arg} \min \mathcal{J}$$

$$\mathcal{J}(\mathbf{x}_0) = \frac{1}{2}\|\mathbf{x}_0 - \mathbf{x}_b\|^2_{\mathbf{B}^{-1}} + \frac{1}{2}\sum_{k=1}^{N}\|\mathbf{H}_k\mathbf{x}_k - \mathbf{y}_k\|^2_{\mathbf{R}_k^{-1}}$$

# Four-dimensional variational data assimilation




$$\min_{\mathbf{x}_0 \in R^m} \mathcal{J}(\mathbf{x}_0)$$

$$\mathbf{x}_0^a = \text{Arg } \min \mathcal{J}$$

$$\mathcal{J}(\mathbf{x}_0) = \frac{1}{2}\|\mathbf{x}_0 - \mathbf{x}_b\|^2_{\mathbf{B}^{-1}} + \frac{1}{2}\sum_{k=1}^{N} \|\mathbf{H}_k \mathbf{x}_k - \mathbf{y}_k\|^2_{\mathbf{R}_k^{-1}}$$

**Typical dimension is of order** $m \sim 10^6 - 10^7$

## Framework :

- Given an ensemble data set collected from observational data at various instants in time $t_1, t_2, \ldots, t_p$ snapshots

$$\left[x^{(1)}, x^{(2)}, \ldots, x^{(p)}\right] \quad x^{(k)} \in \mathbb{R}^n$$

- Define
  - $*$ the weighted ensemble average of the data and the perturbation $\mathcal{X}$

$$\bar{x} = \sum_{k=1}^{p} \omega_k x^{(k)}, \text{ with } 0 \leq \omega_k \leq 1, \text{ and } \sum_{k=1}^{p} \omega_k = 1$$

$$\mathcal{X} = \left[x^{(1)} - \bar{x}, x^{(2)} - \bar{x}, \ldots, x^{(p)} - \bar{x}\right]$$

  - $*$ the weighted covariance matrix

$$\mathcal{C} = \mathcal{X}^T \mathcal{W} \mathcal{X} \text{ where } \mathcal{W} = diag\{\omega_1, \omega_2, \ldots, \omega_p\}$$

  - $*$ the norm $||x||^2_{\mathcal{A}} = < x, x >_{\mathcal{A}} = x^T \mathcal{A} x$, $\mathcal{A} \in \mathbb{R}^{n \times n}$ is an SPD matrix,

$$\mathcal{A} = \begin{cases} Id & \text{for the Euclidean norm} \\ \Lambda & \text{a diagonal matrix for the total energy metric.} \end{cases}$$

- Given an ensemble data set collected from observational data at various instants in time $t_1, t_2, \ldots, t_p$ snapshots

$$\left[x^{(1)}, x^{(2)}, \ldots, x^{(p)}\right] \quad x^{(k)} \in \mathbb{R}^n$$

- Define
  - the weighted ensemble average of the data and the perturbation $\mathcal{X}$

$$\bar{x} = \sum_{k=1}^{p} \omega_k x^{(k)}, \text{ with } 0 \leq \omega_k \leq 1, \text{ and} \sum_{k=1}^{p} \omega_k = 1$$

$$\mathcal{X} = \left[x^{(1)} - \bar{x}, x^{(2)} - \bar{x}, \ldots, x^{(p)} - \bar{x}\right]$$

  - the weighted covariance matrix

$$\mathcal{C} = \mathcal{X}^T \mathcal{W} \mathcal{X} \text{ where } \mathcal{W} = diag\{\omega_1, \omega_2, \ldots, \omega_p\}$$

  - the norm $\|x\|_{\mathcal{A}}^2 = < x, x >_{\mathcal{A}} = x^T \mathcal{A} x$, $\mathcal{A} \in \mathbb{R}^{n \times n}$ is an SPD matrix,

$$\mathcal{A} = \begin{cases} Id & \text{for the Euclidean norm} \\ \Lambda & \text{a diagonal matrix for the total energy metric.} \end{cases}$$

# POD basis:

- Find a projection operator $\mathbb{P}_r : \mathbb{R}^n \longrightarrow \mathbb{R}^p$ of fixed rank $r$ that minimizes the error

$$\sum_{k=1}^{p} \omega_k \|(x^{(k)} - \bar{x}) - \mathbb{P}_r(x^{(k)} - \bar{x})\|_{\mathcal{A}}^2$$

- Solve the eigenvalue problem

$$\mathcal{C}\,\mathcal{A}\phi_k = \sigma_k^2 \phi_k,\ \ k = 1, \cdots, p \ \text{ with } \ <\phi_i, \phi_j>_{\mathcal{A}} = \delta_{i,j}, 1 \le i,j \le p$$

- The optimal $r$-dimensional subspace is $\{\phi_1, \phi_2, \cdots, \phi_r\}$, and the optimal projection is

$$\mathbb{P}_r = \sum_{k=1}^{r} [\phi_k]^T \mathcal{A}\phi_k$$

# POD basis:

- Find a projection operator $\mathbb{P}_r : \mathbb{R}^n \longrightarrow \mathbb{R}^p$ of fixed rank $r$ that minimizes the error

$$\sum_{k=1}^{p} \omega_k \|(x^{(k)} - \bar{x}) - \mathbb{P}_r(x^{(k)} - \bar{x})\|_{\mathcal{A}}^2$$

- Solve the eigenvalue problem

$$\mathcal{C}\,\mathcal{A}\phi_k = \sigma_k^2 \phi_k, \;\; k = 1, \cdots, p \;\text{ with }\; < \phi_i, \phi_j >_{\mathcal{A}} = \delta_{i,j}, 1 \le i, j \le p$$

- The optimal $r$-dimensional subspace is $\{\phi_1, \phi_2, \cdots, \phi_r\}$, and the optimal projection is

$$\mathbb{P}_r = \sum_{k=1}^{r} [\phi_k]^T \mathcal{A}\phi_k$$

# POD basis:

- Find a projection operator $\mathbb{P}_r : \mathbb{R}^n \longrightarrow \mathbb{R}^p$ of fixed rank $r$ that minimizes the error

$$\sum_{k=1}^{p} \omega_k \|(x^{(k)} - \bar{x}) - \mathbb{P}_r(x^{(k)} - \bar{x})\|_{\mathcal{A}}^2$$

- Solve the eigenvalue problem

$$\mathcal{C}\,\mathcal{A}\phi_k = \sigma_k^2 \phi_k, \;\; k = 1, \cdots, p \;\text{ with }\; < \phi_i, \phi_j >_{\mathcal{A}} = \delta_{i,j}, 1 \leq i, j \leq p$$

- The optimal $r$-dimensional subspace is $\{\phi_1, \phi_2, \cdots, \phi_r\}$, and the optimal projection is

$$\mathbb{P}_r = \sum_{k=1}^{r} [\phi_k]^T \mathcal{A}\phi_k$$

- Solve the eigenvalue problem

$$\mathcal{W}^{1/2}\mathcal{X}^T\mathcal{A}\mathcal{X}\mathcal{W}^{1/2}\psi_k = \sigma_k^2$$

- Use the singular value decomposition (SVD)

$$\mathcal{A}^{1/2}\mathcal{X}\mathcal{W}^{1/2} = \mathcal{U}\Sigma\mathcal{V}^T$$

- Compute the POD modes

$$\phi_k = \frac{1}{\sigma_k}\mathcal{X}\mathcal{W}^{1/2}\psi_k$$

- Test of the fraction of total information captured
  - ∗ for $0 < \gamma \leq 1$
  - ∗ select $l$ such that $\{\sum_{k=1}^{l}\sigma_k^2\}/\{\sum_{k=1}^{r}\sigma_k^2\} \geq \gamma$

# Computational aspects:

- Solve the eigenvalue problem

$$\mathcal{W}^{1/2}\mathcal{X}^T\mathcal{A}\mathcal{X}\mathcal{W}^{1/2}\psi_k = \sigma_k^2$$

- Use the singular value decomposition (SVD)

$$\mathcal{A}^{1/2}\mathcal{X}\mathcal{W}^{1/2} = \mathcal{U}\Sigma\mathcal{V}^T$$

- Compute the POD modes

$$\phi_k = \frac{1}{\sigma_k}\mathcal{X}\mathcal{W}^{1/2}\psi_k$$

- Test of the fraction of total information captured
  - * for $0 < \gamma \leq 1$
  - * select $l$ such that $\{\sum_{k=1}^{l}\sigma_k^2\}/\{\sum_{k=1}^{r}\sigma_k^2\} \geq \gamma$

## Computational aspects:

- Solve the eigenvalue problem

$$\mathcal{W}^{1/2}\mathcal{X}^T\mathcal{A}\mathcal{X}\mathcal{W}^{1/2}\psi_k = \sigma_k^2$$

- Use the singular value decomposition (SVD)

$$\mathcal{A}^{1/2}\mathcal{X}\mathcal{W}^{1/2} = \mathcal{U}\Sigma\mathcal{V}^T$$

- Compute the POD modes

$$\phi_k = \frac{1}{\sigma_k}\mathcal{X}\mathcal{W}^{1/2}\psi_k$$

- Test of the fraction of total information captured
  - for $0 < \gamma \leq 1$
  - select $l$ such that $\{\sum_{k=1}^{l}\sigma_k^2\}/\{\sum_{k=1}^{r}\sigma_k^2\} \geq \gamma$

- Solve the eigenvalue problem

$$\mathcal{W}^{1/2}\mathcal{X}^T\mathcal{A}\mathcal{X}\mathcal{W}^{1/2}\psi_k = \sigma_k^2$$

- Use the singular value decomposition (SVD)

$$\mathcal{A}^{1/2}\mathcal{X}\mathcal{W}^{1/2} = \mathcal{U}\Sigma\mathcal{V}^T$$

- Compute the POD modes

$$\phi_k = \frac{1}{\sigma_k}\mathcal{X}\mathcal{W}^{1/2}\psi_k$$

- Test of the fraction of total information captured
  - $*$ for $0 < \gamma \leq 1$
  - $*$ select $l$ such that $\{\sum_{k=1}^{l}\sigma_k^2\}/\{\sum_{k=1}^{r}\sigma_k^2\} \geq \gamma$

# Reduced order control Results:

- Set the following
  * $\Phi = [\phi_1, \phi_2, \cdots, \phi_r]$ the $\mathbb{R}^{n\times r}$ matrix of POD basis, and
  $\boldsymbol{\eta} = (\eta_1, \eta_2, \ldots, \eta_r)$ the coordinates vector in $\mathbb{R}^r$
  * Project $x - \bar{x}$ onto the $r$-dimensional subspace $\{\phi_1, \phi_2, \cdots, \phi_r\}$

$$\mathbb{P}_r(x-\bar{x}) = \sum_{k=1}^{r} \eta_k(t)\phi_k, \text{ where } \eta_k = \phi_k^T \mathcal{A}\mathbb{P}_r(x-\bar{x}) \text{ or } \boldsymbol{\eta} = \Phi^T \mathcal{A}\mathbb{P}_r(x-\bar{x})$$

- Find an optimal estimate (analysis) state vector $\boldsymbol{\eta}^a \in \mathbb{R}^r$ solution of

$$\min_{\boldsymbol{\eta}\in\mathbb{R}^r} \hat{\mathscr{J}}(\boldsymbol{\eta}); \quad \boldsymbol{\eta}^a = \arg\min \hat{\mathscr{J}}$$

where the reduced cost function $\hat{\mathscr{J}}$ is given by

$$\hat{\mathscr{J}}(x) = \frac{1}{2}\left[\mathbb{P}_r(x-x_b)^T\right]\mathbb{P}_r^T\mathfrak{B}\mathbb{P}_r\left[\mathbb{P}_r(x-x_b)\right]$$

$$+\frac{1}{2}\sum_{k=1}^{r}\left[\mathbb{P}_r\{y_k - \mathcal{H}_k(\mathcal{M}_k(x))\}^T\right]\mathbb{P}_r^T\mathfrak{R}_k^{-1}\mathbb{P}_r\left[\mathbb{P}_r\{y_k - \mathcal{H}_k(\mathcal{M}_k(x))\}\right]$$

# Reduced order control Results:

- Set the following
  $*$ $\Phi = [\phi_1, \phi_2, \cdots, \phi_r]$ the $\mathbb{R}^{n\times r}$ matrix of POD basis, and
  $\boldsymbol{\eta} = (\eta_1, \eta_2, \ldots, \eta_r)$ the coordinates vector in $\mathbb{R}^r$
  $*$ Project $x - \bar{x}$ onto the $r$-dimensional subspace $\{\phi_1, \phi_2, \cdots, \phi_r\}$

$$\mathbb{P}_r(x-\bar{x}) = \sum_{k=1}^{r} \eta_k(t)\phi_k, \text{ where } \eta_k = \phi_k^T \mathcal{A}\mathbb{P}_r(x-\bar{x}) \text{ or } \boldsymbol{\eta} = \Phi^T \mathcal{A}\mathbb{P}_r(x-\bar{x})$$

- Find an optimal estimate (analysis) state vector $\boldsymbol{\eta}^a \in \mathbb{R}^r$ solution of

$$\min_{\boldsymbol{\eta}\in\mathbb{R}^r} \hat{\mathscr{J}}(\boldsymbol{\eta}); \quad \boldsymbol{\eta}^a = \arg\min \hat{\mathscr{J}}$$

where the reduced cost function $\hat{\mathscr{J}}$ is given by

$$\hat{\mathscr{J}}(x) = \tfrac{1}{2}\left[\mathbb{P}_r(x - x_b)^T\right] \mathbb{P}_r^T \mathfrak{B} \mathbb{P}_r \left[\mathbb{P}_r(x - x_b)\right]$$
$$+\tfrac{1}{2}\sum_{k=1}^{r} \left[\mathbb{P}_r\{y_k - \mathcal{H}_k(\mathcal{M}_k(x))\}^T\right] \mathbb{P}_r^T \mathfrak{R}_k^{-1} \mathbb{P}_r \left[\mathbb{P}_r\{y_k - \mathcal{H}_k(\mathcal{M}_k(x))\}\right]$$

## Evaluation of the weights $\omega_i$ using the 4D-Var cost function :

- From the tangent linear model $\mathcal{M}(t_i, t)$ and $\mathcal{M}^T(t, t_i)$

$$\delta \mathscr{J} \approx \langle \nabla_{x(t)} \mathscr{J}(\mathbf{x}(t)), \delta\mathbf{x}(t) \rangle = \langle \nabla_{x(t)} \mathscr{J}(\mathbf{x}(t)), \mathcal{M}(t_i, t)\delta\mathbf{x}(t_i) \rangle =$$

$$\langle \mathcal{M}^T(t, t_i)\nabla_{x(t)} \mathscr{J}(\mathbf{x}(t)), \delta\mathbf{x}(t_i) \rangle = \langle \mathcal{A}^{-1}\mathcal{M}^T(t, t_i)\nabla_{x(t)} \mathscr{J}(\mathbf{x}(t)), \delta\mathbf{x}(t_i) \rangle_{\mathcal{A}}$$

- The dual-weights $\omega_i$ to the snapshots are the normalized values

$$\alpha_i = \|\mathcal{A}^{-1}\mathcal{M}^T(t, t_i)\nabla_{x(t)} \mathscr{J}(\mathbf{x}(t))\|_{\mathcal{A}}, \quad \omega_k = \frac{\alpha_k}{\sum_{j=1}^{r} \alpha_j}, \quad k = 1, 2, \ldots r$$

-

- From the tangent linear model $\mathcal{M}(t_i, t)$ and $\mathcal{M}^T(t, t_i)$

$$\delta \mathscr{J} \approx \langle \nabla_{x(t)} \mathscr{J}(\mathbf{x}(t)), \delta \mathbf{x}(t) \rangle = \langle \nabla_{x(t)} \mathscr{J}(\mathbf{x}(t)), \mathcal{M}(t_i, t) \delta \mathbf{x}(t_i) \rangle =$$

$$\langle \mathcal{M}^T(t, t_i) \nabla_{x(t)} \mathscr{J}(\mathbf{x}(t)), \delta \mathbf{x}(t_i) \rangle = \langle \mathcal{A}^{-1} \mathcal{M}^T(t, t_i) \nabla_{x(t)} \mathscr{J}(\mathbf{x}(t)), \delta \mathbf{x}(t_i) \rangle_{\mathcal{A}}$$

- The dual-weights $\omega_i$ to the snapshots are the normalized values

$$\alpha_i = \| \mathcal{A}^{-1} \mathcal{M}^T(t, t_i) \nabla_{x(t)} \mathscr{J}(\mathbf{x}(t)) \|_{\mathcal{A}}, \quad \omega_k = \frac{\alpha_k}{\sum_{j=1}^r \alpha_j}, \quad k = 1, 2, \ldots r$$

## Evaluation of the weights $\omega_i$ using the 4D-Var cost function :

- From the tangent linear model $\mathcal{M}(t_i, t)$ and $\mathcal{M}^T(t, t_i)$

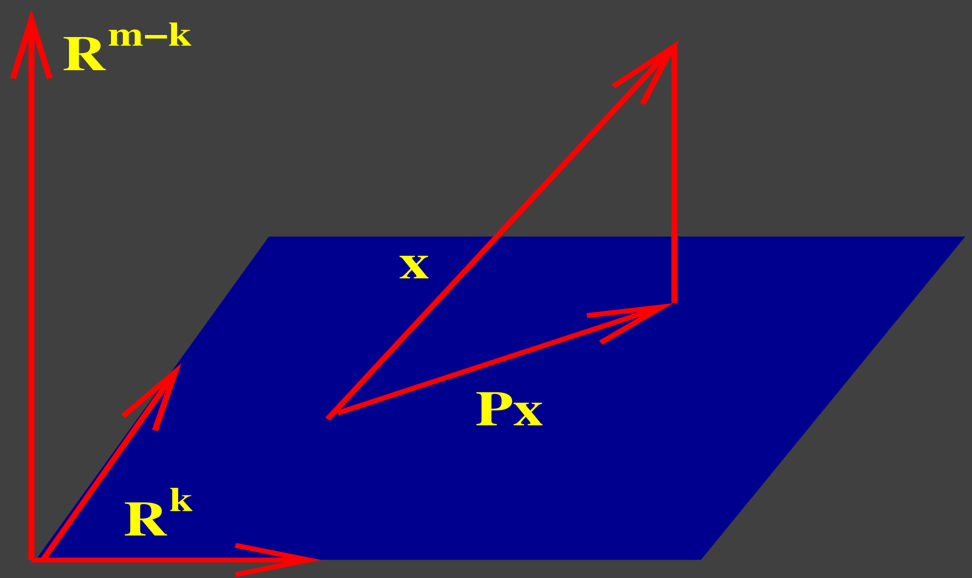
$$\delta \mathscr{J} \approx \langle \nabla_{x(t)} \mathscr{J}(\mathbf{x}(t)), \delta \mathbf{x}(t) \rangle = \langle \nabla_{x(t)} \mathscr{J}(\mathbf{x}(t)), \mathcal{M}(t_i, t) \delta \mathbf{x}(t_i) \rangle =$$

$$\langle \mathcal{M}^T(t, t_i) \nabla_{x(t)} \mathscr{J}(\mathbf{x}(t)), \delta \mathbf{x}(t_i) \rangle = \langle \mathcal{A}^{-1} \mathcal{M}^T(t, t_i) \nabla_{x(t)} \mathscr{J}(\mathbf{x}(t)), \delta \mathbf{x}(t_i) \rangle_{\mathcal{A}}$$

- The dual-weights $\omega_i$ to the snapshots are the normalized values

$$\alpha_i = \| \mathcal{A}^{-1} \mathcal{M}^T(t, t_i) \nabla_{x(t)} \mathscr{J}(\mathbf{x}(t)) \|_{\mathcal{A}}, \quad \omega_k = \frac{\alpha_k}{\sum_{j=1}^{r} \alpha_j}, \quad k = 1, 2, \ldots r$$

# Reduced order 4D-Var - general framework




$$\hat{\mathcal{J}}(\boldsymbol{\eta}) := \mathcal{J}(\overline{\mathbf{x}} + \boldsymbol{\Psi}\boldsymbol{\eta})$$

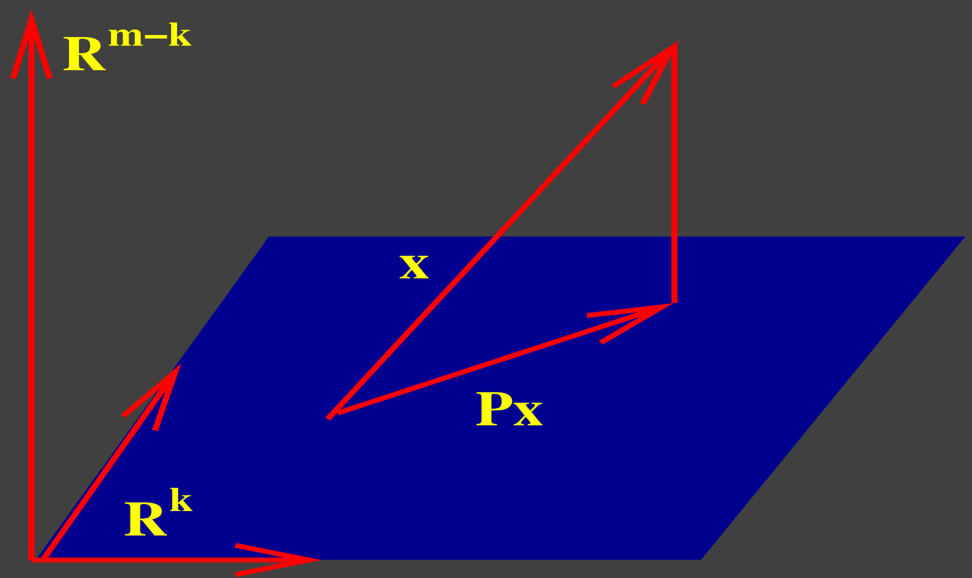
$$\min_{\boldsymbol{\eta} \in R^k} \hat{\mathcal{J}}(\boldsymbol{\eta})$$

$$\mathbf{x}_0^a \approx \overline{\mathbf{x}} + \boldsymbol{\Psi}\boldsymbol{\eta}^a$$

Projection on $Span\{\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \ldots, \boldsymbol{\psi}_k\}$: $\mathcal{P}_{\boldsymbol{\psi},k} = \boldsymbol{\Psi}\boldsymbol{\Psi}^T\mathbf{A}$, $k \ll m$

$$\mathcal{P}_{\boldsymbol{\psi},k}(\mathbf{x}_0 - \overline{\mathbf{x}}) = \boldsymbol{\Psi}\boldsymbol{\Psi}^T\mathbf{A}(\mathbf{x}_0 - \overline{\mathbf{x}}) = \boldsymbol{\Psi}\boldsymbol{\eta}, \qquad \boldsymbol{\eta} \in R^k$$

# Reduced order 4D-Var - general framework




$$\hat{\mathcal{J}}(\boldsymbol{\eta}) := \mathcal{J}(\overline{\mathbf{x}} + \boldsymbol{\Psi}\boldsymbol{\eta})$$

$$\min_{\boldsymbol{\eta} \in R^k} \hat{\mathcal{J}}(\boldsymbol{\eta})$$

$$\mathbf{x}_0^a \approx \overline{\mathbf{x}} + \boldsymbol{\Psi}\boldsymbol{\eta}^a$$

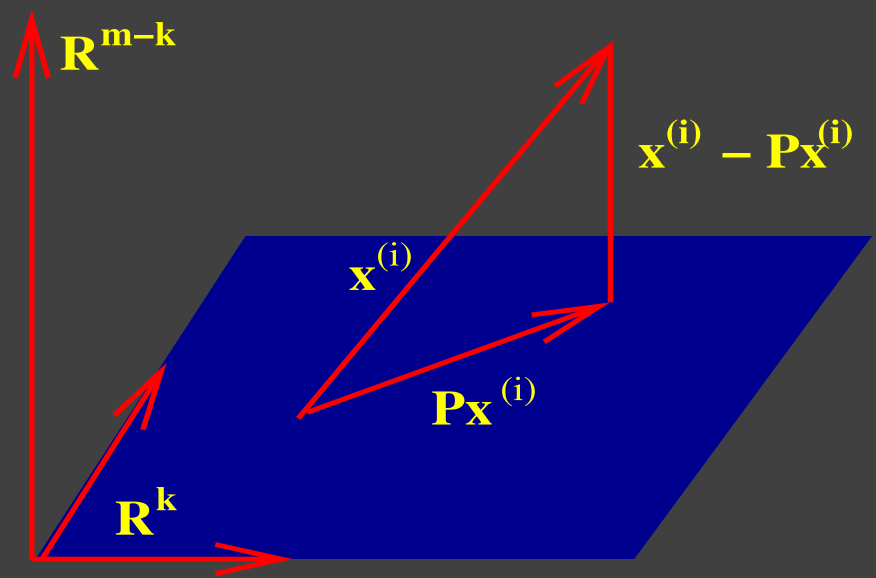
Projection on $Span\{\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \ldots, \boldsymbol{\psi}_k\}$: $\mathcal{P}_{\boldsymbol{\psi},k} = \boldsymbol{\Psi}\boldsymbol{\Psi}^T\mathbf{A}$, $k \ll m$

$$\mathcal{P}_{\boldsymbol{\psi},k}(\mathbf{x}_0 - \overline{\mathbf{x}}) = \boldsymbol{\Psi}\boldsymbol{\Psi}^T\mathbf{A}(\mathbf{x}_0 - \overline{\mathbf{x}}) = \boldsymbol{\Psi}\boldsymbol{\eta}, \qquad \boldsymbol{\eta} \in R^k$$

**Low-order optimization problem $k \sim 10 - 10^2$**

# The Proper Orthogonal Decomposition Method

*Empirical Orthogonal Functions, Karhunen-Loève decomposition*




Ensemble data $\{\mathbf{x}^{(i)}\}, i = \overline{1, n}$

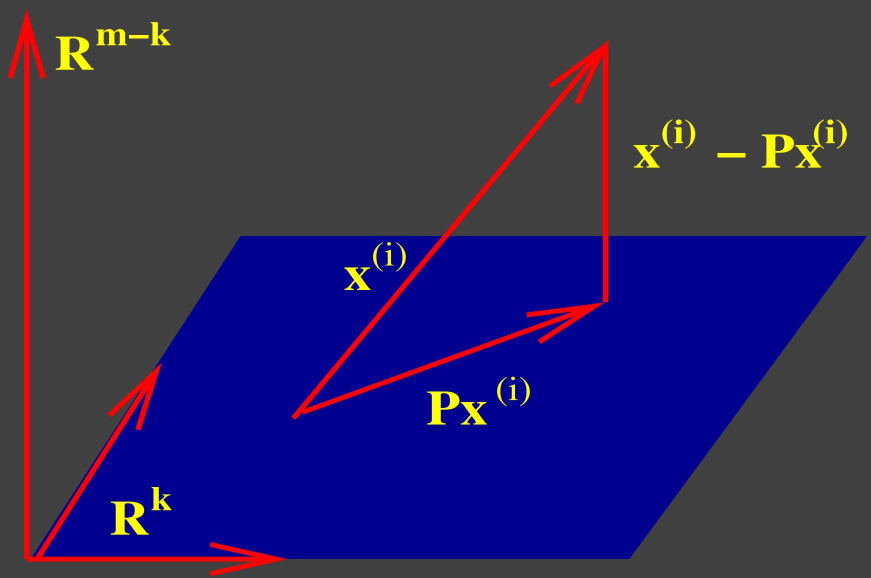
Optimal order k representation

$$\min_{\{\boldsymbol{\psi}\}} \sum_{i=1}^{n} \omega_j \left\| \mathbf{x}^{(i)} - \mathcal{P}_{\boldsymbol{\psi},k} \mathbf{x}^{(i)} \right\|_{\mathbf{A}}^2$$

$$\langle \boldsymbol{\psi}_i, \boldsymbol{\psi}_j \rangle_{\mathbf{A}} = \delta_{ij}, 1 \leq i \leq j \leq k$$

# The Proper Orthogonal Decomposition Method

*Empirical Orthogonal Functions, Karhunen-Loève decomposition*




Ensemble data $\{\mathbf{x}^{(i)}\}, i = \overline{1, n}$

Optimal order k representation

$$\min_{\{\boldsymbol{\psi}\}} \sum_{i=1}^{n} \omega_j \left\| \mathbf{x}^{(i)} - \mathcal{P}_{\boldsymbol{\psi},k} \mathbf{x}^{(i)} \right\|_{\mathbf{A}}^2$$

$$\langle \boldsymbol{\psi}_i, \boldsymbol{\psi}_j \rangle_{\mathbf{A}} = \delta_{ij}, 1 \le i \le j \le k$$

**Dependence on the metric A and the weights $\omega$**

## Problem Description:

- Model: A two-dimensional global shallow-water (SW) model
- Two data assimilation experiments are set up:
  ∗ DAS-I, is a model inversion problem where data is provided for all discrete state components and no background term is included in the cost functional
  ∗DAS-II, the background term is included in the cost and data is provided at every 4th grid point on the longitudinal and latitudinal directions (i.e. only 6% of the state is observed every six hours).
- Algorithms & schemes
  ∗ the explicit flux-form semi-Lagrangian (FFSL) scheme of Lin and Rood (1997)
  ∗ The adjoint model to the SW-FFSL scheme of Akella and Navon (2006) and TAMC software (Giering and Kaminski 1998).

# Problem Description:

- Model: A two-dimensional global shallow-water (SW) model
- Two data assimilation experiments are set up:
  * DAS-I, is a model inversion problem where data is provided for all discrete state components and no background term is included in the cost functional
  *DAS-II, the background term is included in the cost and data is provided at every 4th grid point on the longitudinal and latitudinal directions (i.e. only 6% of the state is observed every six hours).
- Algorithms & schemes
  * the explicit flux-form semi-Lagrangian (FFSL) scheme of Lin and Rood (1997)
  * The adjoint model to the SW-FFSL scheme of Akella and Navon (2006) and TAMC software (Giering and Kaminski 1998).

# Problem Description:

- Model: A two-dimensional global shallow-water (SW) model
- Two data assimilation experiments are set up:
  $*$ DAS-I, is a model inversion problem where data is provided for all discrete state components and no background term is included in the cost functional
  $*$DAS-II, the background term is included in the cost and data is provided at every 4th grid point on the longitudinal and latitudinal directions (i.e. only  6% of the state is observed every six hours).
- Algorithms & schemes
  $*$ the explicit flux-form semi-Lagrangian (FFSL) scheme of Lin and Rood (1997)
  $*$ The adjoint model to the SW-FFSL scheme of Akella and Navon (2006) and TAMC software (Giering and Kaminski 1998).

# Problem Description: continued

- Input: the ECMWF ERA-40 atmospheric data to specify the SW model state variables at the initial time
- Resolution & time step : ($144 \times 72$ grid cells) such that the dimension of the discrete state vector $\mathbf{x} = (h, u, v)$ is $\sim 3 \times 10^4$. The time step $\Delta t = 450\,\mathrm{s}$
- Reference initial state $\mathbf{x}_0^{ref}$ : the 500mb ECMWF ERA-40 data valid for 06h UTC 15 March 2002.
- Snapshots: from small random perturbations $\delta\mathbf{x}_0$ in the reference initial conditions and a full model integration starting with $\mathbf{x}_0^{ref} + \delta\mathbf{x}_0$.

- Input: the ECMWF ERA-40 atmospheric data to specify the SW model state variables at the initial time
- Resolution & time step : ($144 \times 72$ grid cells) such that the dimension of the discrete state vector $\mathbf{x} = (h, u, v)$ is $\sim 3 \times 10^4$. The time step $\Delta t = 450\,\mathrm{s}$
- Reference initial state $\mathbf{x}_0^{ref}$ : the 500mb ECMWF ERA-40 data valid for 06h UTC 15 March 2002.
- Snapshots: from small random perturbations $\delta\mathbf{x}_0$ in the reference initial conditions and a full model integration starting with $\mathbf{x}_0^{ref} + \delta\mathbf{x}_0$.

- Input: the ECMWF ERA-40 atmospheric data to specify the SW model state variables at the initial time
- Resolution & time step : ($144 \times 72$ grid cells) such that the dimension of the discrete state vector $\mathbf{x} = (h, u, v)$ is $\sim 3 \times 10^4$. The time step $\Delta t = 450\,\mathrm{s}$
- Reference initial state $\mathbf{x}_0^{ref}$: the 500mb ECMWF ERA-40 data valid for 06h UTC 15 March 2002.
- Snapshots: from small random perturbations $\delta\mathbf{x}_0$ in the reference initial conditions and a full model integration starting with $\mathbf{x}_0^{ref} + \delta\mathbf{x}_0$.

- Input: the ECMWF ERA-40 atmospheric data to specify the SW model state variables at the initial time
- Resolution & time step : ($144 \times 72$ grid cells) such that the dimension of the discrete state vector $\mathbf{x} = (h, u, v)$ is $\sim 3 \times 10^4$. The time step $\Delta t = 450\,\mathrm{s}$
- Reference initial state $\mathbf{x}_0^{ref}$: the 500mb ECMWF ERA-40 data valid for 06h UTC 15 March 2002.
- Snapshots: from small random perturbations $\delta\mathbf{x}_0$ in the reference initial conditions and a full model integration starting with $\mathbf{x}_0^{ref} + \delta\mathbf{x}_0$.

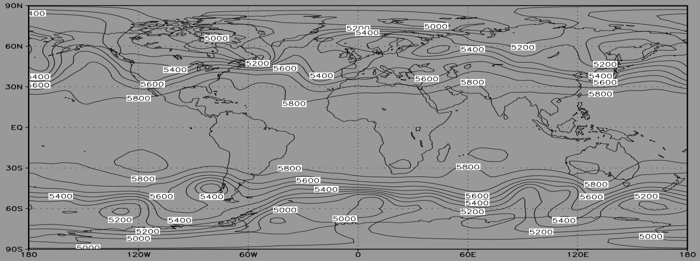



Figure: Isopleths of the geopotential height (m) for the reference run configuration at the initial time specified from ECMWF ERA-40 data sets Bottom:the 24h forecast of the shallow water model

# Results:

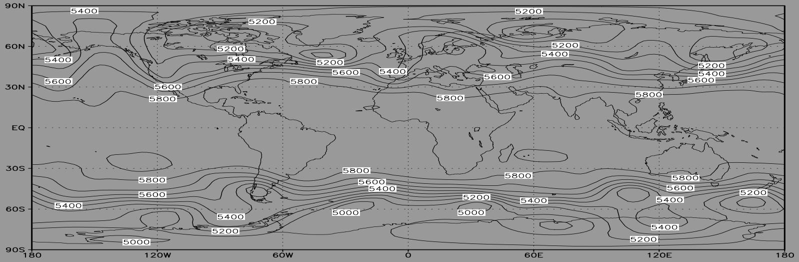



Figure: Isopleths of the geopotential height (m) for the reference run the 24h forecast of the shallow water model

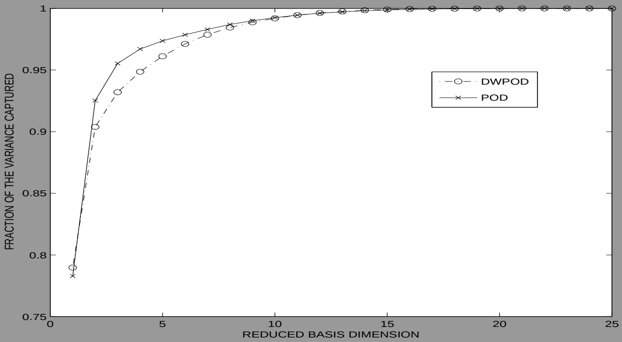



Figure: The fraction of the variance captured by the POD and DWPOD modes from the snapshot data as a function of the dimension of the reduced space.

# Results:

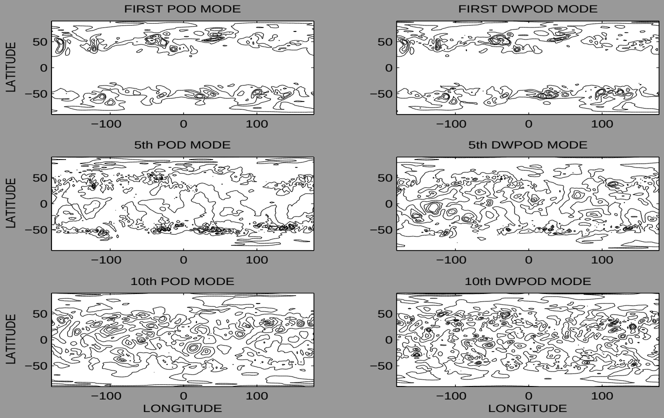



Figure: Isopleths of the POD and DWPOD modes of rank 1, 5, and 10. A total energy norm is used to provide point values.

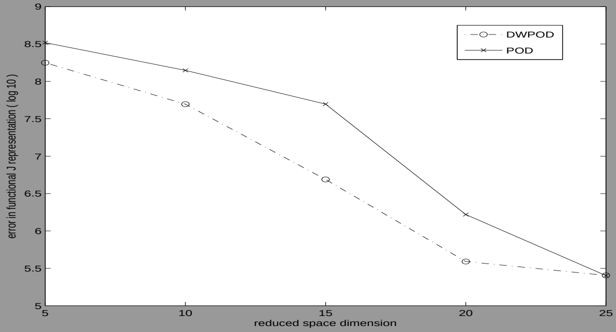



Figure: Comparative results for the reduced-order POD and DWPOD forecasts for $k = 5, 10, 15, 20, 25$. Top figure: error (log 10) in the reduced-order representation of the time integrated total energy of the system

# Results:

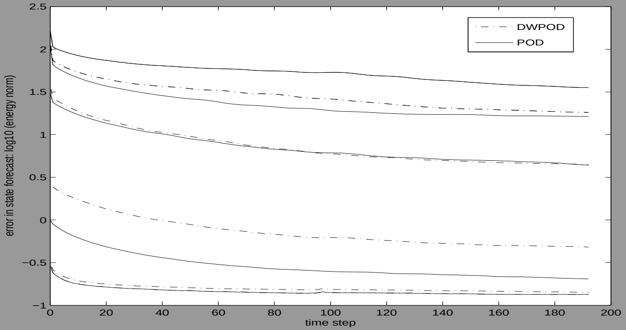



Figure: Comparative results for the reduced-order POD and DWPOD forecasts for $k = 5, 10, 15, 20, 25$. Total energy error (log 10) $\|\mathbf{x}_i^{ref} - \hat{\mathbf{x}}_i\|_{\mathbf{A}}$ of the reduced-order representation of the forecast at each time step in the interval 0-24h.

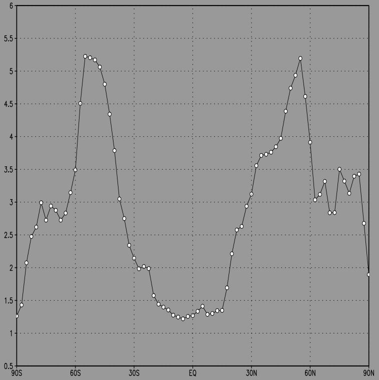



Figure: Zonal averaged errors in the background estimate to the initial conditions.

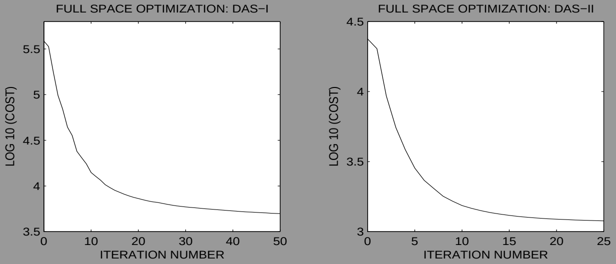



Figure: The iterative minimization process in the full state space for DAS-I (left) and DAS-II (right).

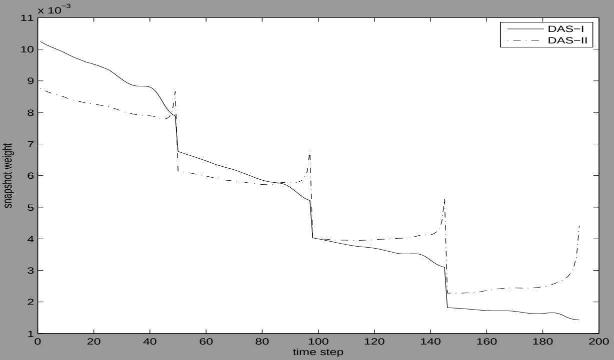



Figure: The dual weights to the snapshot data determined by the adjoint model in DAS-I and in DAS-II

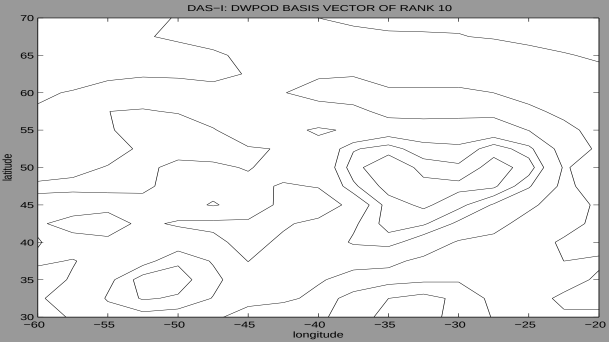



Figure: Isopleths of the $10^{th}$ mode in the DWPOD basis for DAS-I

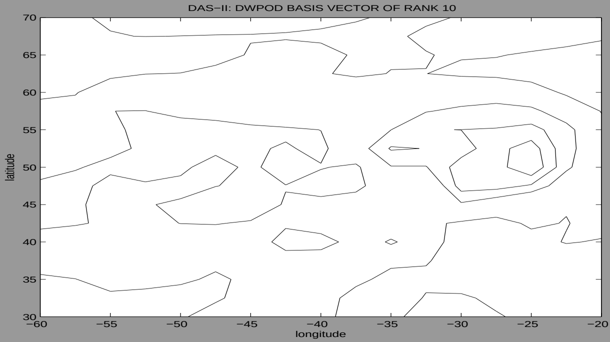



Figure: Isopleths of the $10^{th}$ mode in the DWPOD basis for DAS-II. A distinct configuration it is noticed since the DWPOD basis is adjusted to the optimization problem at hand.

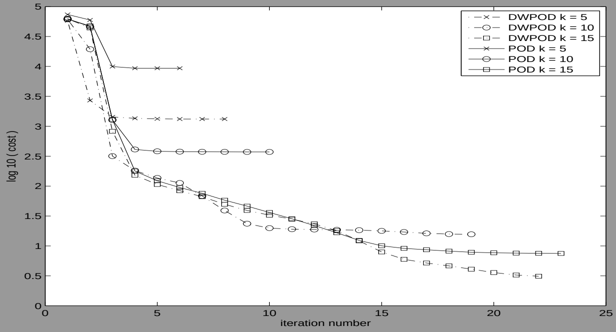



Figure: The iterative minimization process in the reduced space for the POD and DWPOD spaces of dimension 5, 10, and 15. Optimization without background term and dense observations, corresponding to DAS-I

## Results:

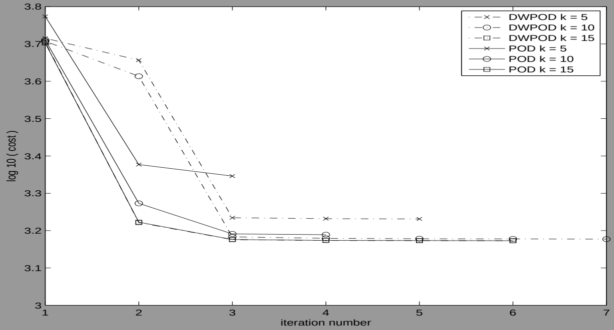



Figure: The iterative minimization process in the reduced space for the POD and DWPOD spaces of dimension 5, 10, and 15.Optimization with background term and sparse observations, corresponding to DAS-II

# Results:

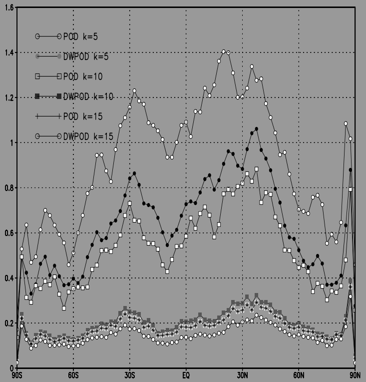



Figure: Zonal averaged errors in the analysis provided by the reduced order 4D-Var data assimilation Results for the DAS-I experiments with POD and DWPOD spaces of dimension 5, 10, and 15.

# Results:

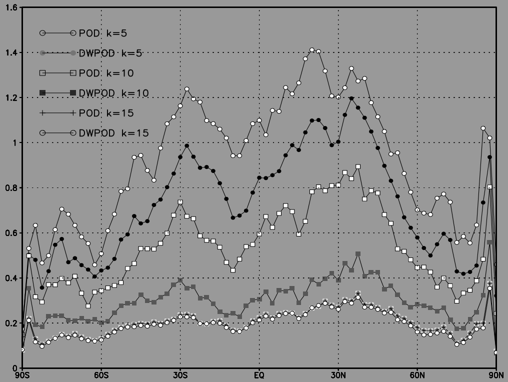



Figure: Zonal averaged errors in the analysis provided by the reduced order 4D-Var data assimilation. Results for the DAS-II experiments with POD and DWPOD spaces of dimension 5, 10, and 15.

# Conclusions:

- An adjoint-model approach is proposed to directly incorporate information from the DAS into the optimality criteria that defines the reduced space basis.
- The dual weighted POD method is novel in reduced order 4D-Var data assimilation and relies on a weighted ensemble data mean and weighted snapshots with weights determined by the adjoint DAS ( Data Assimilation System).
- The DWPOD space was found to increase the accuracy in the representation of a forecast aspect by as much as an order of magnitude versus the POD space representation.

# Conclusions:

- An adjoint-model approach is proposed to directly incorporate information from the DAS into the optimality criteria that defines the reduced space basis.
- The dual weighted POD method is novel in reduced order 4D-Var data assimilation and relies on a weighted ensemble data mean and weighted snapshots with weights determined by the adjoint DAS ( Data Assimilation System).
- The DWPOD space was found to increase the accuracy in the representation of a forecast aspect by as much as an order of magnitude versus the POD space representation.

# Conclusions:

- An adjoint-model approach is proposed to directly incorporate information from the DAS into the optimality criteria that defines the reduced space basis.
- The dual weighted POD method is novel in reduced order 4D-Var data assimilation and relies on a weighted ensemble data mean and weighted snapshots with weights determined by the adjoint DAS ( Data Assimilation System).
- The DWPOD space was found to increase the accuracy in the representation of a forecast aspect by as much as an order of magnitude versus the POD space representation.

# Conclusions: continued

- The benefit gained from the dual-weighted procedure diminishes as the dimension of the reduced space increases from 10 to 15, indicating that most of the information provided by the snapshot data is captured by the reduced basis.
- In 4D-Var data assimilation twin experiments, optimization in the DWPOD space provided a reduction in the analysis errors by as much as a factor of three when compared to the POD-based optimization.
- Use of dual weighted goal oriented criteria may also serve also as goal-oriented a posteriori error estimate to drive grid adaptivity.

# Conclusions: continued

- The benefit gained from the dual-weighted procedure diminishes as the dimension of the reduced space increases from 10 to 15, indicating that most of the information provided by the snapshot data is captured by the reduced basis.
- In 4D-Var data assimilation twin experiments, optimization in the DWPOD space provided a reduction in the analysis errors by as much as a factor of three when compared to the POD-based optimization.
- Use of dual weighted goal oriented criteria may also serve also as goal-oriented a posteriori error estimate to drive grid adaptivity.

# Conclusions: continued

- The benefit gained from the dual-weighted procedure diminishes as the dimension of the reduced space increases from 10 to 15, indicating that most of the information provided by the snapshot data is captured by the reduced basis.
- In 4D-Var data assimilation twin experiments, optimization in the DWPOD space provided a reduction in the analysis errors by as much as a factor of three when compared to the POD-based optimization.
- Use of dual weighted goal oriented criteria may also serve also as goal-oriented a posteriori error estimate to drive grid adaptivity.

A reduced order approach to four-dimensional variational data assimilation using proper orthogonal decomposition. Yanhua Cao, Jiang Zhu, I.M. Navon and Zhendong Luo International Journal for Numerical Methods in Fluids , Volume 53, Issue 10 , 1571-1583 (2007)

## The reduced-gravity model:

$$\begin{cases} \dfrac{\partial u}{\partial t} - fv = -g'\dfrac{\partial h}{\partial x} + \dfrac{\partial \tau^x}{\partial \rho_0 H} + A\nabla^2 u - \alpha u \\ \dfrac{\partial v}{\partial t} - fu = -g'\dfrac{\partial h}{\partial y} + \dfrac{\partial \tau^y}{\partial \rho_0 H} + A\nabla^2 v - \alpha v \\ \dfrac{\partial h}{\partial t} + H(\dfrac{\partial u}{\partial x} + \dfrac{\partial v}{\partial y}) = 0 \end{cases}$$

- $(u, v)$ : the horizontal velocity components of the depth-averaged currents
- $h$ : the total layer thickness
- $f$ : the Coriolis force
- $H$ : the mean depth of the layer
- $\rho_0$ : the density of water
- $A$ : the horizontal eddy coefficient
- $\alpha$ : the friction coefficient
- $(\tau^x, \tau^y)$ : the wind stress

- The dynamic model governing the ocean flow $U(t, x)$

$$\begin{cases} \dfrac{dU}{dt} = \mathcal{F}(t, U) \\ U(0, x) = U_0(x) \end{cases}$$

- the reduced dynamic model: the forward model

$$\begin{cases} \dfrac{dc_k}{dt} = \langle \mathcal{F}(t, \bar{U} + \sum_{i=1}^{p} c_i \phi_i), \phi_k \rangle \\ c_k(t = 0) = c_k(0) \end{cases}$$

- The dynamic model governing the ocean flow $U(t, x)$

$$\begin{cases} \dfrac{dU}{dt} = \mathcal{F}(t, U) \\ \\ U(0, x) = U_0(x) \end{cases}$$

- the reduced dynamic model: the forward model

$$\begin{cases} \dfrac{dc_k}{dt} = \langle \mathcal{F}(t, \bar{U} + \sum_{i=1}^{p} c_i \phi_i), \phi_k \rangle \\ \\ \qquad c_k(t=0) = c_k(0) \end{cases}$$

* $U_0^{POD}, V, U_b, \bar{U}$ state, observation, background and mean vectors,
* $\mathcal{M}, \mathcal{H}$ model and observation operators,
* $\mathfrak{B}, \mathfrak{R}$ background and observational error covariance matrices

- Find an optimal estimate (analysis) state vector $\{U_0^{POD}\}^a$ solution of

$$\mathscr{J}(U_0^{POD}) = (U_0^{POD} - U_b)^T \mathfrak{B}(U_0^{POD} - U_b) \ +$$

$$+ \ \left[V - \mathcal{H}(U^{POD})\right]^T \mathfrak{R}^{-1} \left[V - \mathcal{H}(U^{POD})\right]$$

with

$$(U_0^{POD})(x) = (U_0^{POD})(0,x) = \bar{U}(x) + \sum_{k=1}^{p} c_k(0)\phi_k(x)$$

$$(U^{POD})(x) = (U_0^{POD})(t,x) = \bar{U}(x) + \sum_{k=1}^{p} c_k(t)\phi_k(x)$$

* $U_0^{POD}, V, U_b, \bar{U}$ state, observation, background and mean vectors,
* $\mathcal{M}, \mathcal{H}$ model and observation operators,
* $\mathfrak{B}, \mathfrak{R}$ background and observational error covariance matrices

- Find an optimal estimate (analysis) state vector $\{U_0^{POD}\}^a$ solution of

$$\mathscr{J}(U_0^{POD}) = (U_0^{POD} - U_b)^T \mathfrak{B}(U_0^{POD} - U_b) \ +$$

$$+ \ \left[V - \mathcal{H}(U^{POD})\right]^T \mathfrak{R}^{-1} \left[V - \mathcal{H}(U^{POD})\right]$$

with

$$(U_0^{POD})(x) = (U_0^{POD})(0,x) = \bar{U}(x) + \sum_{k=1}^{p} c_k(0)\phi_k(x)$$

$$(U^{POD})(x) = (U_0^{POD})(t,x) = \bar{U}(x) + \sum_{k=1}^{p} c_k(t)\phi_k(x)$$

# Computational aspects :

- The adjoint model of the reduced forward model is used to calculate the gradient of the cost function $\mathscr{J}(U_0^{POD})$
- The initial value of the cost function in the full model space is different from the initial value of the cost function in the POD space.
  - $*$ $U_0 = U_b$ for the full model
  - $*$ $\beta_0 = \mathbb{P}_r^T(U_b - \bar{U})$ for the reduced model
- Use of an adaptive POD 4D-Var

# Computational aspects :

- The adjoint model of the reduced forward model is used to calculate the gradient of the cost function $\mathscr{J}(U_0^{POD})$
- The initial value of the cost function in the full model space is different from the initial value of the cost function in the POD space.
  - $U_0 = U_b$ for the full model
  - $\beta_0 = \mathbb{P}_r^T(U_b - \bar{U})$ for the reduced model
- Use of an adaptive POD 4D-Var

# Computational aspects :

- The adjoint model of the reduced forward model is used to calculate the gradient of the cost function $\mathscr{J}(U_0^{POD})$
- The initial value of the cost function in the full model space is different from the initial value of the cost function in the POD space.
  $*$ $U_0 = U_b$ for the full model
  $*$ $\eta_0 = \mathbb{P}_r^T(U_b - \bar{U})$ for the reduced model
- Use of an adaptive POD 4D-Var

1. Establish POD model using background initial conditions
2. Perform optimization iterations to obtain the optimal solution
3. Generate a new set of snapshots if after a preset number of iterations, the cost function cannot be reduced.
4. Establish a new POD model using the new snapshots and continue the optimization process
5. Return back to 2 if the optimality conditions are not reached

1. Establish POD model using background initial conditions
2. Perform optimization iterations to obtain the optimal solution
3. Generate a new set of snapshots if after a preset number of iterations, the cost function cannot be reduced.
4. Establish a new POD model using the new snapshots and continue the optimization process
5. Return back to 2 if the optimality conditions are not reached

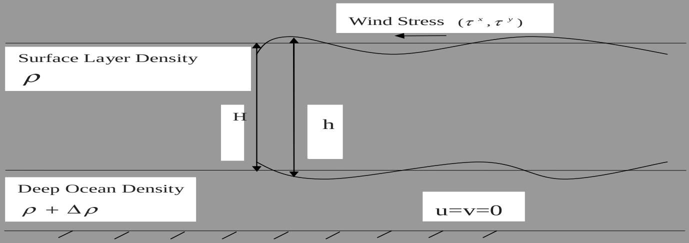
# Adaptive POD 4D-Var :

1. Establish POD model using background initial conditions
2. Perform optimization iterations to obtain the optimal solution
3. Generate a new set of snapshots if after a preset number of iterations, the cost function cannot be reduced.
4. Establish a new POD model using the new snapshots and continue the optimization process
5. Return back to 2 if the optimality conditions are not reached

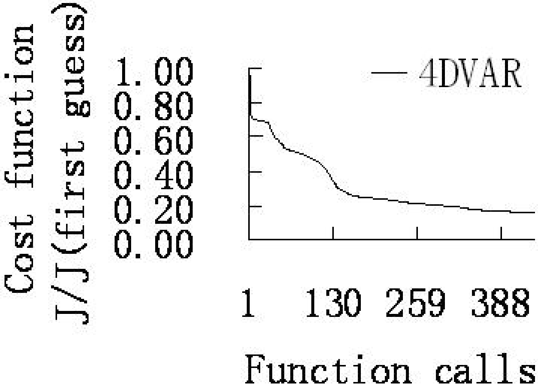
1. Establish POD model using background initial conditions
2. Perform optimization iterations to obtain the optimal solution
3. Generate a new set of snapshots if after a preset number of iterations, the cost function cannot be reduced.
4. Establish a new POD model using the new snapshots and continue the optimization process
5. Return back to 2 if the optimality conditions are not reached

1. Establish POD model using background initial conditions
2. Perform optimization iterations to obtain the optimal solution
3. Generate a new set of snapshots if after a preset number of iterations, the cost function cannot be reduced.
4. Establish a new POD model using the new snapshots and continue the optimization process
5. Return back to 2 if the optimality conditions are not reached

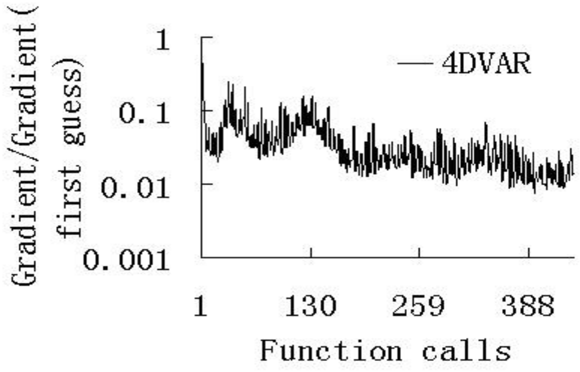
# Results:


Wind Stress $(\tau^x, \tau^y)$
Surface Layer Density
$\rho$
H
h
Deep Ocean Density
$\rho + \Delta\rho$
u=v=0

## Results:

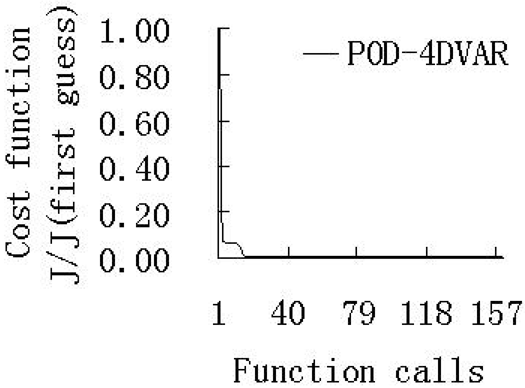


Cost function
J/J(first guess)
1.00
0.80
0.60
0.40
0.20
0.00
— 4DVAR
1
130
259
388
Function calls

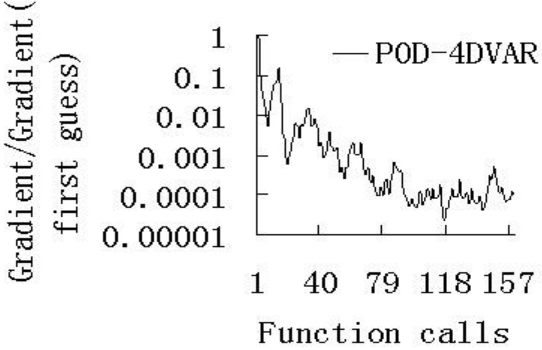



Evolution of the cost function and gradient in 4DVAR experiment. (a) cost function; (b) gradient as a function of the number of minimization iterations.

## Results:

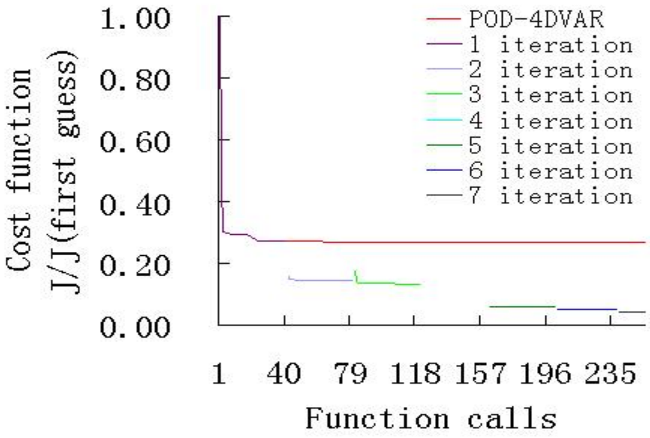


Cost function
J/J(first guess)
1.00
0.80
0.60
0.40
0.20
0.00
— POD-4DVAR
1
40
79
118
157
Function calls



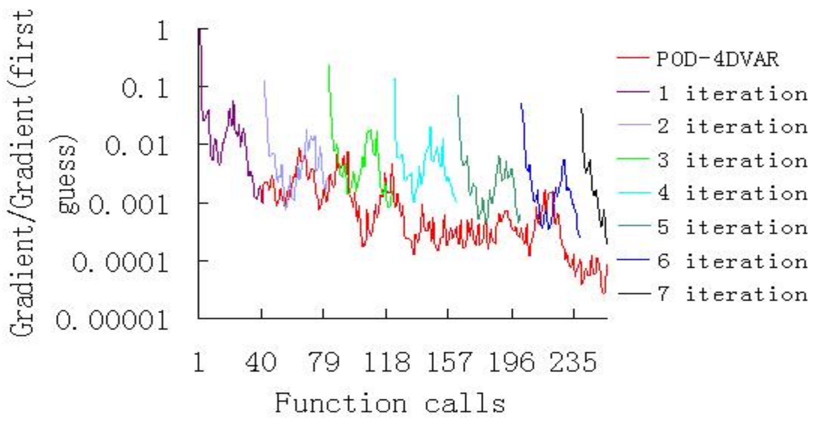

Evolution of the cost function and gradient in the POD 4D-Var

## Results:



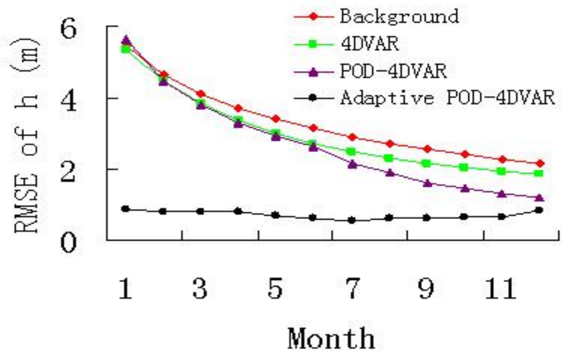
1.00
0.80
0.60
0.40
0.20
0.00
Cost function
J/J(first guess)
POD-4DVAR
1 iteration
2 iteration
3 iteration
4 iteration
5 iteration
6 iteration
7 iteration
1
40
79
118
157
196
235
Function calls

# Results:



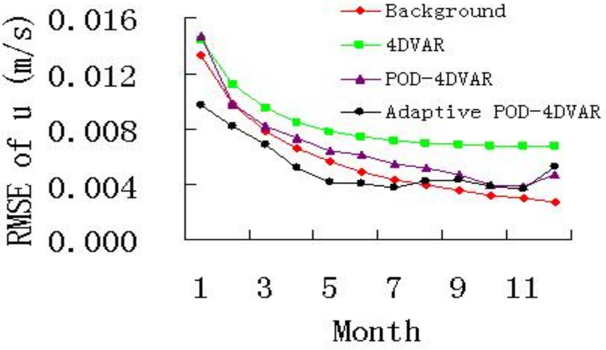

Evolution of the cost function and gradient in adaptive POD 4D-Var

# Results:



Background
4DVAR
POD-4DVAR
Adaptive POD-4DVAR
RMSE of h (m)
6
4
2
0
1
3
5
7
9
11
Month

# Results:




RMSE of the results compared to the true state for upper layer thickness

# Results:

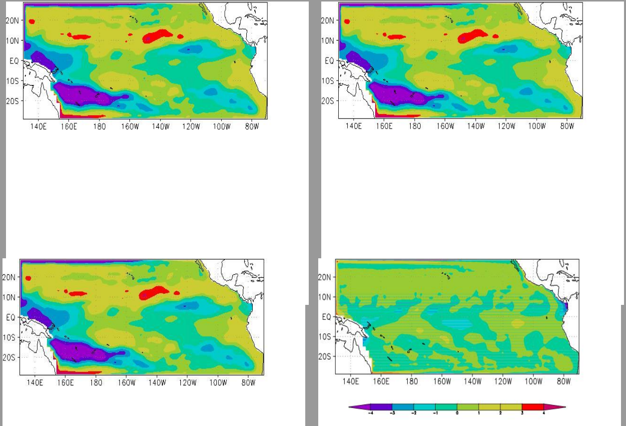



Figure: Errors between the true state and the numerical approximations for upper layer thickness at the initial time

# Results:

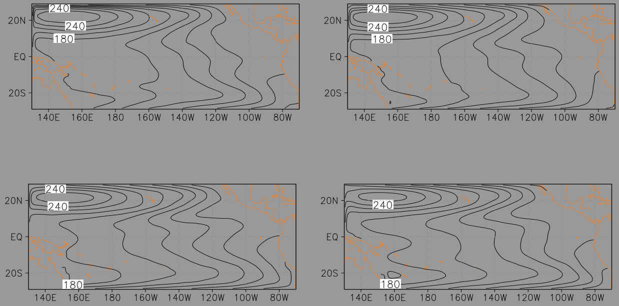



Figure: Upper layer thickness Feb.,May,Aug.and Nov. in case of 5,20,30 snapshots

# Results:

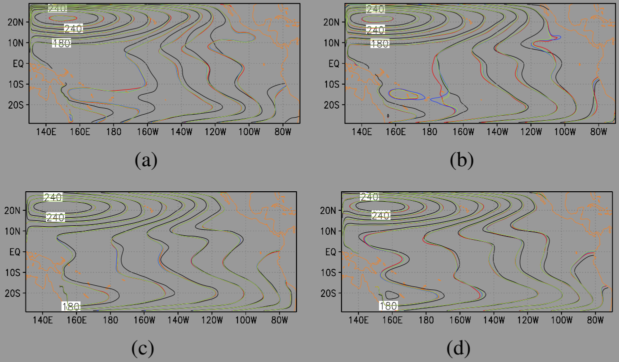



Figure: Figure 4 Upper layer thickness in February, May, August and November in case of 5 snapshots, 20 snapshots, 30 snapshots, energy capture 95%, the full model approximation and the reduced order approximation. Black isoline: full order approximation, red isoline: 5 snapshots, green isoline: 20 snapshots, blue isoline: 30 snapshots.

- Developing a new (open source, community) modeling framework using a range of powerful and novel numerical methods
- Wide range of applicability from process/laboratory scales to regional/global
- Build upon a fluids code used for atmosphere/ocean modeling, coastal engineering, impact cratering, engineering fluids, multiphase flows, computational biology etc.

- Developing a new (open source, community) modeling framework using a range of powerful and novel numerical methods
- Wide range of applicability from process/laboratory scales to regional/global
- Build upon a fluids code used for atmosphere/ocean modeling, coastal engineering, impact cratering, engineering fluids, multiphase flows, computational biology etc.

- Developing a new (open source, community) modeling framework using a range of powerful and novel numerical methods
- Wide range of applicability from process/laboratory scales to regional/global
- Build upon a fluids code used for atmosphere/ocean modeling, coastal engineering, impact cratering, engineering fluids, multiphase flows, computational biology etc.

# Summary of ICOM-continued

- Take advantage of cross-fertilization of computational techniques between disciplines
- The target problems have in common complex geometries and typically multi-scale and anisotropic solution structures
- which makes unstructured meshes and dynamic mesh adaptivity a natural choice with which to tackle these problems

# Summary of ICOM-continued

- Take advantage of cross-fertilization of computational techniques between disciplines
- The target problems have in common complex geometries and typically multi-scale and anisotropic solution structures
- which makes unstructured meshes and dynamic mesh adaptivity a natural choice with which to tackle these problems

# Summary of ICOM-continued

- Take advantage of cross-fertilization of computational techniques between disciplines
- The target problems have in common complex geometries and typically multi-scale and anisotropic solution structures
- which makes unstructured meshes and dynamic mesh adaptivity a natural choice with which to tackle these problems

# Motivation for a new ocean model utilizing unstructured and adaptive mesh methods

- Need to resolve a wide range of spatial and temporal scales
- Model internal waves, boundary currents, eddies, overflows, convection events etc., accurately and efficiently within a global and coupled context
- Need for accurate and efficient representation of highly complex domains

# Motivation for a new ocean model utilizing unstructured and adaptive mesh methods

- Need to resolve a wide range of spatial and temporal scales
- Model internal waves, boundary currents, eddies, overflows, convection events etc., accurately and efficiently within a global and coupled context
- Need for accurate and efficient representation of highly complex domains

## Motivation for a new ocean model utilizing unstructured and adaptive mesh methods

- Need to resolve a wide range of spatial and temporal scales

- Model internal waves, boundary currents, eddies, overflows, convection events etc., accurately and efficiently within a global and coupled context

- Need for accurate and efficient representation of highly complex domains

# Motivation for a new ocean model utilizing unstructured and adaptive mesh methods-continued:

- Ability to model interaction of flow with small scale topography, shelf seas, coastal regions, islands, estuaries, harbors,etc.
- Exploit anisotropy prevalent in this application area and attempt to take advantage of recent developments in CFD, numerical analysis, etc.
- Health in the genetic diversity of models and modeling approaches used

## Motivation for a new ocean model utilizing unstructured and adaptive mesh methods-continued:

- Ability to model interaction of flow with small scale topography, shelf seas, coastal regions, islands, estuaries, harbors,etc.
- Exploit anisotropy prevalent in this application area and attempt to take advantage of recent developments in CFD, numerical analysis, etc.
- Health in the genetic diversity of models and modeling approaches used

## Motivation for a new ocean model utilizing unstructured and adaptive mesh methods-continued:

- Ability to model interaction of flow with small scale topography, shelf seas, coastal regions, islands, estuaries, harbors,etc.
- Exploit anisotropy prevalent in this application area and attempt to take advantage of recent developments in CFD, numerical analysis, etc.
- Health in the genetic diversity of models and modeling approaches used

- Start with Fluidity:an open source control volume finite element solver for 3D compressible multi-phase fluids. Has been developed by AMCG for more than a decade and is the basis for a range of multi-physics multi-scale applications
- Add an adaptivity library which performs topological operations on the mesh, and mesh movement, to optimize the size and shape of elements in response to error measures
- Working robustly and efficiently in parallel

- Start with Fluidity:an open source control volume finite element solver for 3D compressible multi-phase fluids. Has been developed by AMCG for more than a decade and is the basis for a range of multi-physics multi-scale applications

- Add an adaptivity library which performs topological operations on the mesh, and mesh movement, to optimize the size and shape of elements in response to error measures

- Working robustly and efficiently in parallel

# A quick overview of the numerical technology:

- Start with Fluidity:an open source control volume finite element solver for 3D compressible multi-phase fluids. Has been developed by AMCG for more than a decade and is the basis for a range of multi-physics multi-scale applications

- Add an adaptivity library which performs topological operations on the mesh, and mesh movement, to optimize the size and shape of elements in response to error measures

- Working robustly and efficiently in parallel

- Refer to the models oceanographic form as ICOM: note it is obviously non-hydrostatic
- Also building an adjoint to the model for data assimilation, sensitivity studies and goal-based error estimation
- Make use of open source solutions for solvers, preconditioners, I/O, visualization, etc

## A quick overview of the numerical technology-continued

- Refer to the models oceanographic form as ICOM: note it is obviously non-hydrostatic
- Also building an adjoint to the model for data assimilation, sensitivity studies and goal-based error estimation
- Make use of open source solutions for solvers, preconditioners, I/O, visualization, etc

- Refer to the models oceanographic form as ICOM: note it is obviously non-hydrostatic
- Also building an adjoint to the model for data assimilation, sensitivity studies and goal-based error estimation
- Make use of open source solutions for solvers, preconditioners, I/O, visualization, etc

- A finite element discretization of the Boussinesq equations in 3-D
- Use of a non-hydrostatic solver to model dense water formation and flow over steep topography;
- Accurate and robust representation of hydrostatic and geostrophic balance;
- Inclusion of barotropic and baroclinic modes;
- Anisotropic unstructured meshes in the vertical as well as the horizontal to capture the details of local flow in all three directions.

- A finite element discretization of the Boussinesq equations in 3-D
- Use of a non-hydrostatic solver to model dense water formation and flow over steep topography;
- Accurate and robust representation of hydrostatic and geostrophic balance;
- Inclusion of barotropic and baroclinic modes;
- Anisotropic unstructured meshes in the vertical as well as the horizontal to capture the details of local flow in all three directions.

- A finite element discretization of the Boussinesq equations in 3-D
- Use of a non-hydrostatic solver to model dense water formation and flow over steep topography;
- Accurate and robust representation of hydrostatic and geostrophic balance;
- Inclusion of barotropic and baroclinic modes;
- Anisotropic unstructured meshes in the vertical as well as the horizontal to capture the details of local flow in all three directions.

# A quick overview of the numerical technology-continued

- A finite element discretization of the Boussinesq equations in 3-D
- Use of a non-hydrostatic solver to model dense water formation and flow over steep topography;
- Accurate and robust representation of hydrostatic and geostrophic balance;
- Inclusion of barotropic and baroclinic modes;
- Anisotropic unstructured meshes in the vertical as well as the horizontal to capture the details of local flow in all three directions.

- A finite element discretization of the Boussinesq equations in 3-D
- Use of a non-hydrostatic solver to model dense water formation and flow over steep topography;
- Accurate and robust representation of hydrostatic and geostrophic balance;
- Inclusion of barotropic and baroclinic modes;
- Anisotropic unstructured meshes in the vertical as well as the horizontal to capture the details of local flow in all three directions.

- The second-order Crank-Nicolson scheme
- A semi-implicit projection method for pressure, ensuring the flow remains divergence-free while decoupling the computations for the momentum and continuity equations
- A fourth-order pressure filter is employed to aid stability
- Standard Petrov-Galerkin weightings are also used to improve numerical stability in the presence of advection dominated flows.

- The second-order Crank-Nicolson scheme
- A semi-implicit projection method for pressure, ensuring the flow remains divergence-free while decoupling the computations for the momentum and continuity equations
- A fourth-order pressure filter is employed to aid stability
- Standard Petrov-Galerkin weightings are also used to improve numerical stability in the presence of advection dominated flows.

# A quick overview of the numerical technology-continued

- The second-order Crank-Nicolson scheme
- A semi-implicit projection method for pressure, ensuring the flow remains divergence-free while decoupling the computations for the momentum and continuity equations
- A fourth-order pressure filter is employed to aid stability
- Standard Petrov-Galerkin weightings are also used to improve numerical stability in the presence of advection dominated flows.

- The second-order Crank-Nicolson scheme
- A semi-implicit projection method for pressure, ensuring the flow remains divergence-free while decoupling the computations for the momentum and continuity equations
- A fourth-order pressure filter is employed to aid stability
- Standard Petrov-Galerkin weightings are also used to improve numerical stability in the presence of advection dominated flows.

- The 3-D non-hydrostatic Boussinesq equations:

$$\nabla \cdot \mathbf{u} = 0,$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + f\mathbf{k} \times \mathbf{u} = -\nabla p - \rho g \mathbf{k} + \nabla \cdot \tau$$

- The variables can be expressed as an expansion of the POD basis functions for $u, v, w, p$

$$u(t,x,y,z) = \bar{u} + \sum_{m=1}^{M_u} \alpha_{m,u}(t)\Phi_{m,u}(x,y,z),$$

$$v(t,x,y,z) = \bar{v} + \sum_{m=1}^{M_v} \alpha_{m,v}(t)\Phi_{m,v}(x,y,z),$$

$$w(t,x,y,z) = \bar{w} + \sum_{m=1}^{M_w} \alpha_{m,w}(t)\Phi_{m,w}(x,y,z),$$

$$p(t,x,y,z) = \bar{p} + \sum_{m=1}^{M_p} \alpha_{m,p}(t)\Phi_{m,p}(x,y,z)$$

# Geostrophic pressure:

- The pressure is divided into two parts: $p = p_{ng} + p_g$. The geostrophic pressure has to satisfy the geostrophic balance:

$$-\nabla p_g = f\mathbf{k}\nabla\mathbf{u}$$

- Taking the divergence of the above equation, an elliptic equation for geostrophic pressure is obtained

$$-\nabla^2 p_g = \frac{\partial(-fv)}{\partial x} + \frac{\partial(fu)}{\partial y}$$

# Geostrophic pressure:

- The pressure is divided into two parts: $p = p_{ng} + p_g$. The geostrophic pressure has to satisfy the geostrophic balance:

$$-\nabla p_g = f\mathbf{k}\nabla\mathbf{u}$$

- Taking the divergence of the above equation, an elliptic equation for geostrophic pressure is obtained

$$-\nabla^2 p_g = \frac{\partial(-fv)}{\partial x} + \frac{\partial(fu)}{\partial y}$$

# Geostrophic pressure-continued:

- To accurately represent geostrophic pressure its basis functions are split into two sets: $\Phi_{pgu}$ and $\Phi_{pgv}$ which are associated with the $u$- and $v$-velocity components. The geostrophic pressure can be obtained from a quadratic finite element representation while linear finite element representations are used for the velocity components

- Furthermore the geostrophic pressure can be represented by a summation of the two sets of geostrophic basis functions, which are calculated by solving the following elliptic equations using a conjugate gradient iterative method:

$$-\nabla^2 \Phi_{pgu,m} = \frac{\partial(f\Phi_{m,u})}{\partial y}$$

$$-\nabla^2 \Phi_{pgv,m} = \frac{\partial(-f\Phi_{m,v})}{\partial x}$$

- The geostrophic pressure can therefore be expressed as:

$$p_g = \bar{p}_g + \sum_{m=1}^{M} \alpha_{m,u} \Phi_{m,u} + \sum_{m=1}^{M} \alpha_{m,v} \Phi_{m,v}$$

- In addition the average geostrophic pressure is calculated from:

$$-\nabla^2 \bar{p}_g = \frac{\partial(-f\bar{v})}{\partial x} + \frac{\partial(f\bar{u})}{\partial y}$$

- The geostrophic pressure can therefore be expressed as:

$$p_g = \bar{p}_g + \sum_{m=1}^{M} \alpha_{m,u} \Phi_{m,u} + \sum_{m=1}^{M} \alpha_{m,v} \Phi_{m,v}$$

- In addition the average geostrophic pressure is calculated from:

$$-\nabla^2 \bar{p}_g = \frac{\partial(-f\bar{v})}{\partial x} + \frac{\partial(f\bar{u})}{\partial y}$$

Challenges:

- The snapshots can be different length at different time levels
- The POD mesh of the forward model can diff from the POD mesh of the adjoint model

Solution:

- To overcome these difficulties, a standard reference fixed mesh is adopted for the reduced model. The solutions from the original full model are interpolated from their own mesh onto the same reference fixed mesh at each time level, and then stored in the snapshots
- A goal-based error measurement approach is employed to find an optimal mesh for both reduced forward and adjoint models

Challenges:

- The snapshots can be different length at different time levels
- The POD mesh of the forward model can diff from the POD mesh of the adjoint model

Solution:

- To overcome these difficulties, a standard reference fixed mesh is adopted for the reduced model. The solutions from the original full model are interpolated from their own mesh onto the same reference fixed mesh at each time level, and then stored in the snapshots
- A goal-based error measurement approach is employed to find an optimal mesh for both reduced forward and adjoint models

Challenges:

- The snapshots can be different length at different time levels
- The POD mesh of the forward model can diff from the POD mesh of the adjoint model

Solution:

- To overcome these difficulties, a standard reference fixed mesh is adopted for the reduced model. The solutions from the original full model are interpolated from their own mesh onto the same reference fixed mesh at each time level, and then stored in the snapshots
- A goal-based error measurement approach is employed to find an optimal mesh for both reduced forward and adjoint models

Challenges:

- The snapshots can be different length at different time levels
- The POD mesh of the forward model can diff from the POD mesh of the adjoint model

Solution:

- To overcome these difficulties, a standard reference fixed mesh is adopted for the reduced model. The solutions from the original full model are interpolated from their own mesh onto the same reference fixed mesh at each time level, and then stored in the snapshots
- A goal-based error measurement approach is employed to find an optimal mesh for both reduced forward and adjoint models

## Goal-based approach to choose an optimal mesh for both POD reduced order forward and adjoint models

- A function (defined below as the model reduction errors or the solution $\mathbf{u}$ which is of interest, say, in the 'target' regions) is used to (1) determine an optimal for both reduced forward and adjoint models. Suppose that the functional whose accuracy is to be optimized is represented as $\Im \equiv \Im(\psi)$, and

$$\Im(\psi) = \int_{\Omega} f(\psi)\, dV$$

- In addition, the above functional can be also used to optimize uncertainties (inversion problem ) in models; optimize the POD bases and thus improve the accuracy of reduced models.

# Goal-based approach to choose an optimal mesh for both POD reduced order forward and adjoint models

- A function (defined below as the model reduction errors or the solution **u** which is of interest, say, in the 'target' regions) is used to (1) determine an optimal for both reduced forward and adjoint models. Suppose that the functional whose accuracy is to be optimized is represented as $\Im \equiv \Im\left(\psi\right)$, and

$$\Im\left(\psi\right) = \int_{\Omega} f\left(\psi\right)\, dV$$

- In addition, the above functional can be also used to optimize uncertainties (inversion problem ) in models; optimize the POD bases and thus improve the accuracy of reduced models.

# Goal-based approach to choose an optimal mesh for both POD reduced order forward and adjoint models-continued

- The nodal Metric Tensor $\bar{M}_i$ is obtained from the reduced forward model

  $$\bar{M}_i = \frac{\gamma}{|\bar{\epsilon}_i|} |\bar{H}_i|$$

  where $\bar{H}_i$ is the forward Hessian matrix at node $i$, $\bar{\epsilon}_i$ is the forward interpolation error at node $i$

- The nodal Metric Tensor $\bar{M}_i$ is obtained from the reduced forward model

  $$\bar{M}_i^* = \frac{\gamma}{|\bar{\epsilon}_i^*|} |\bar{H}_i^*|$$

  where $\bar{H}_i^*$ is the adjoint Hessian matrix at node $i$, $\bar{\epsilon}_i^*$ is the adjoint interpolation error at node $i$

# Goal-based approach to choose an optimal mesh for both POD reduced order forward and adjoint models-continued

- The nodal Metric Tensor $\bar{M}_i$ is obtained from the reduced forward model

$$\bar{M}_i = \frac{\gamma}{|\bar{\epsilon}_i|} \left| \bar{H}_i \right|$$

where $\bar{H}_i$ is the forward Hessian matrix at node $i$, $\bar{\epsilon}_i$ is the forward interpolation error at node $i$

- The nodal Metric Tensor $\bar{M}_i$ is obtained from the reduced forward model

$$\bar{M}_i^* = \frac{\gamma}{|\bar{\epsilon}_i^*|} \left| \bar{H}_i^* \right|$$

where $\bar{H}_i^*$ is the adjoint Hessian matrix at node $i$, $\bar{\epsilon}_i^*$ is the adjoint interpolation error at node $i$

# Goal-based approach to choose an optimal mesh for both POD reduced order forward and adjoint models-continued:

- The interpolation error is related to the error contribution of the functional
- To satisfy the goal, the minimal ellipsoid is obtained by superscribing both ellipses and used to determine an optimal mesh

$$\bar{\bar{M}}_i^{\mathcal{G}_s} = \mathcal{G}_s\left(\bar{M}_i, \bar{M}_i^*\right)$$

# Goal-based approach to choose an optimal mesh for both POD reduced order forward and adjoint models-continued:

- The interpolation error is related to the error contribution of the functional
- To satisfy the goal, the minimal ellipsoid is obtained by superscribing both ellipses and used to determine an optimal mesh

$$\bar{\bar{M}}_i^{\mathcal{G}_s} = \mathcal{G}_s\left(\bar{M}_i, \bar{M}_i^*\right)$$

# Goal-based approach to choose an optimal mesh for both POD reduced order forward and adjoint models-continued

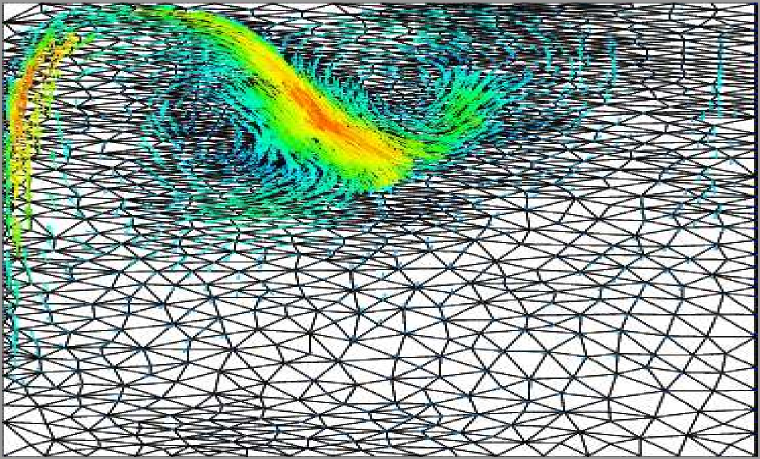


Figure: An optimal mesh obtained by the goal-based error measure approach

- The aim of 4D-Var is to determine optimal control variables (e.g., initial conditions). Optimal solution is obtained by minimizing the functional $\Im(U^0)$:

$$\Im(U^0) = \frac{1}{2}(U^0 - U_b)^T \mathbf{B}^{-1}(U^0 - U_b) + \frac{1}{2}\sum_{n=1}^{N_t}(\mathbf{H}U^n - y_o^n)^T W_o(\mathbf{H}U^n - y_o^n)$$

- The functional in reduced space:

$$\Im(\alpha(0)) = \frac{1}{2}\left(\left(\bar{U} + \sum_{m=1}^{M}\alpha_m(0)\Phi_m(\mathbf{x})\right) - U_b\right)^T \mathbf{B}^{-1}\left(\left(\bar{U} + \sum_{m=1}^{M}\alpha_m(0)\Phi_m(\mathbf{x})\right) - U_b\right)$$

$$+\frac{1}{2}\sum_{n=1}^{N_t}\left(\mathbf{H}\left(\bar{U} + \sum_{m=1}^{M}\alpha_m(t^n)\Phi_m(\mathbf{x}),\right) - y_o^n\right)^T W^o\left(\mathbf{H}\left(\bar{U} + \sum_{m=1}^{M}\alpha_m(t^n)\Phi_m(\mathbf{x}),\right) - y_o^n\right)$$

- The aim of 4D-Var is to determine optimal control variables (e.g., initial conditions). Optimal solution is obtained by minimizing the functional $\Im(U^0)$:

$$\Im(U^0) = \frac{1}{2}(U^0 - U_b)^T \mathbf{B}^{-1}(U^0 - U_b) + \frac{1}{2}\sum_{n=1}^{N_t}(\mathbf{H}U^n - y_o^n)^T W_o(\mathbf{H}U^n - y_o^n)$$

- The functional in reduced space:

$$\Im(\alpha(0)) = \frac{1}{2}\left(\left(\bar{U} + \sum_{m=1}^{M}\alpha_m(0)\Phi_m(\mathbf{x})\right) - U_b\right)^T \mathbf{B}^{-1}\left(\left(\bar{U} + \sum_{m=1}^{M}\alpha_m(0)\Phi_m(\mathbf{x})\right) - U_b\right)$$

$$+\frac{1}{2}\sum_{n=1}^{N_t}\left(\mathbf{H}\left(\bar{U} + \sum_{m=1}^{M}\alpha_m(t^n)\Phi_m(\mathbf{x}),\right) - y_o^n\right)^T W^o\left(\mathbf{H}\left(\bar{U} + \sum_{m=1}^{M}\alpha_m(t^n)\Phi_m(\mathbf{x}),\right) - y_o^n\right)$$

- The POD model is based on the solution of the original model for specified control variables (e.g., initial and boundary conditions, etc). It is therefore necessary to reconstruct the POD model when the resulting control variables from the latest optimization iteration are significantly different from the ones upon which the POD model is based.

- The reduced basis is recalculated using a refreshed set of snapshots based on the latest results obtained from the full forward model using a restart criterion of the adaptive POD procedure based on convergence of the minimization process. One can also consider the Trust Region Method for restart criterion.

## Adaptive POD in 4-D Var-continued:

- The POD model is based on the solution of the original model for specified control variables (e.g., initial and boundary conditions, etc). It is therefore necessary to reconstruct the POD model when the resulting control variables from the latest optimization iteration are significantly different from the ones upon which the POD model is based.

- The reduced basis is recalculated using a refreshed set of snapshots based on the latest results obtained from the full forward model using a restart criterion of the adaptive POD procedure based on convergence of the minimization process. One can also consider the Trust Region Method for restart criterion.

# Adaptive POD procedure:

1. Set the POD iteration level $it = 1$ and the initial guess controls $c_{it}$;
2. Set up the snapshots $U_{it}$ from the solution of the full forward model with the controls $c_{it}$;
3. Calculate the POD bases (the number of POD bases is chosen to capture a prescribed energy level);
4. Project the controls $c_{it}$ on the reduced space $\alpha_{it,jt}$ ($jt = 1$);
5. Optimize the initial controls $\alpha_{it,jt}$ (note: the optimization procedure is carried out completely on the reduced space. The Polak-Ribiere nonlinear conjugate gradient (CG) technique is employed here and $jt$ is the Nonlinear CG iteration level );

# Adaptive POD procedure:

1. Set the POD iteration level $it = 1$ and the initial guess controls $c_{it}$;
2. Set up the snapshots $U_{it}$ from the solution of the full forward model with the controls $c_{it}$;
3. Calculate the POD bases (the number of POD bases is chosen to capture a prescribed energy level);
4. Project the controls $c_{it}$ on the reduced space $\alpha_{it,jt}$ ($jt = 1$);
5. Optimize the initial controls $\alpha_{it,jt}$ (note: the optimization procedure is carried out completely on the reduced space. The Polak-Ribiere nonlinear conjugate gradient (CG) technique is employed here and $jt$ is the Nonlinear CG iteration level );

# Adaptive POD procedure:

1. Set the POD iteration level $it = 1$ and the initial guess controls $c_{it}$;
2. Set up the snapshots $U_{it}$ from the solution of the full forward model with the controls $c_{it}$;
3. Calculate the POD bases (the number of POD bases is chosen to capture a prescribed energy level);
4. Project the controls $c_{it}$ on the reduced space $\alpha_{it,jt}$ ($jt = 1$);
5. Optimize the initial controls $\alpha_{it,jt}$ (note: the optimization procedure is carried out completely on the reduced space. The Polak-Ribiere nonlinear conjugate gradient (CG) technique is employed here and $jt$ is the Nonlinear CG iteration level );

# Adaptive POD procedure:

1. Set the POD iteration level $it = 1$ and the initial guess controls $c_{it}$;
2. Set up the snapshots $U_{it}$ from the solution of the full forward model with the controls $c_{it}$;
3. Calculate the POD bases (the number of POD bases is chosen to capture a prescribed energy level);
4. Project the controls $c_{it}$ on the reduced space $\alpha_{it,jt}$ ($jt = 1$);
5. Optimize the initial controls $\alpha_{it,jt}$ (note: the optimization procedure is carried out completely on the reduced space. The Polak-Ribiere nonlinear conjugate gradient (CG) technique is employed here and $jt$ is the Nonlinear CG iteration level );

# Adaptive POD procedure:

1. Set the POD iteration level $it = 1$ and the initial guess controls $c_{it}$;

2. Set up the snapshots $U_{it}$ from the solution of the full forward model with the controls $c_{it}$;

3. Calculate the POD bases (the number of POD bases is chosen to capture a prescribed energy level);

4. Project the controls $c_{it}$ on the reduced space $\alpha_{it,jt}$ $(jt = 1)$;

5. Optimize the initial controls $\alpha_{it,jt}$ (note: the optimization procedure is carried out completely on the reduced space. The Polak-Ribiere nonlinear conjugate gradient (CG) technique is employed here and *jt* is the Nonlinear CG iteration level );

# Adaptive POD procedure-continued

6.
   - check the value of the functional. If $|\Im_{jt}| < \epsilon$ (where, $\epsilon$ is the tolerance for the optimization), then go to step 7;
   - if $|\Im_{jt}| > \epsilon$ and $|\Im_{jt} - \Im_{jt-1}| > 10^{-3}$ (where, $jt-1$ and $jt$ are the consecutive optimization iteration levels), then set $jt = jt + 1$ and go back step 5;
   - if $|\Im_{jt}| > \epsilon$ and $|\Im_{jt} - \Im_{jt-1}| < 10^{-3}$, then update the POD bases:
     - i. find the new controls $c_{it+1}$ by projecting the optimization controls $\alpha_{it,j}$ onto the original flow domain, and
     - ii. set $it = it + 1$ and go back step 2;

7. The adaptive POD optimization procedure is completed.

# Adaptive POD procedure-continued

6.

- check the value of the functional. If $|\Im_{jt}| < \epsilon$ (where, $\epsilon$ is the tolerance for the optimization), then go to step 7;

- if $|\Im_{jt}| > \epsilon$ and $|\Im_{jt} - \Im_{jt-1}| > 10^{-3}$ (where, $jt - 1$ and $jt$ are the consecutive optimization iteration levels), then set $jt = jt + 1$ and go back step 5;

- if $|\Im_{jt}| > \epsilon$ and $|\Im_{jt} - \Im_{jt-1}| < 10^{-3}$, then update the POD bases:
  i. find the new controls $c_{it+1}$ by projecting the optimization controls $\alpha_{it,j}$ onto the original flow domain, and
  ii. set $it = it + 1$ and go back step 2;

7. The adaptive POD optimization procedure is completed.

6.

- check the value of the functional. If $|\Im_{jt}| < \epsilon$ (where, $\epsilon$ is the tolerance for the optimization), then go to step 7;
- if $|\Im_{jt}| > \epsilon$ and $|\Im_{jt} - \Im_{jt-1}| > 10^{-3}$ (where, $jt-1$ and $jt$ are the consecutive optimization iteration levels), then set $jt = jt + 1$ and go back step 5;
- if $|\Im_{jt}| > \epsilon$ and $|\Im_{jt} - \Im_{jt-1}| < 10^{-3}$, then update the POD bases:
  - i. find the new controls $c_{it+1}$ by projecting the optimization controls $\alpha_{it,j}$ onto the original flow domain, and
  - ii. set $it = it + 1$ and go back step 2;

7. The adaptive POD optimization procedure is completed.

6.

- check the value of the functional. If $|\Im_{jt}| < \epsilon$ (where, $\epsilon$ is the tolerance for the optimization), then go to step 7;
- if $|\Im_{jt}| > \epsilon$ and $|\Im_{jt} - \Im_{jt-1}| > 10^{-3}$ (where, $jt-1$ and $jt$ are the consecutive optimization iteration levels), then set $jt = jt + 1$ and go back step 5;
- if $|\Im_{jt}| > \epsilon$ and $|\Im_{jt} - \Im_{jt-1}| < 10^{-3}$, then update the POD bases:
  i. find the new controls $c_{it+1}$ by projecting the optimization controls $\alpha_{it,j}$ onto the original flow domain, and
  ii. set $it = it + 1$ and go back step 2;

7. The adaptive POD optimization procedure is completed.

# Adaptive POD procedure-continued

6.

- check the value of the functional. If $|\Im_{jt}| < \epsilon$ (where, $\epsilon$ is the tolerance for the optimization), then go to step 7;
- if $|\Im_{jt}| > \epsilon$ and $|\Im_{jt} - \Im_{jt-1}| > 10^{-3}$ (where, $jt-1$ and $jt$ are the consecutive optimization iteration levels), then set $jt = jt + 1$ and go back step 5;
- if $|\Im_{jt}| > \epsilon$ and $|\Im_{jt} - \Im_{jt-1}| < 10^{-3}$, then update the POD bases:
    i. find the new controls $c_{it+1}$ by projecting the optimization controls $\alpha_{it,j}$ onto the original flow domain, and
    ii. set $it = it + 1$ and go back step 2;

7. The adaptive POD optimization procedure is completed.

6.
- check the value of the functional. If $|\Im_{jt}| < \epsilon$ (where, $\epsilon$ is the tolerance for the optimization), then go to step 7;
- if $|\Im_{jt}| > \epsilon$ and $|\Im_{jt} - \Im_{jt-1}| > 10^{-3}$ (where, $jt-1$ and $jt$ are the consecutive optimization iteration levels), then set $jt = jt+1$ and go back step 5;
- if $|\Im_{jt}| > \epsilon$ and $|\Im_{jt} - \Im_{jt-1}| < 10^{-3}$, then update the POD bases:
  i. find the new controls $c_{it+1}$ by projecting the optimization controls $\alpha_{it,j}$ onto the original flow domain, and
  ii. set $it = it+1$ and go back step 2;

7. The adaptive POD optimization procedure is completed.

6.
- check the value of the functional. If $|\Im_{jt}| < \epsilon$ (where, $\epsilon$ is the tolerance for the optimization), then go to step 7;
- if $|\Im_{jt}| > \epsilon$ and $|\Im_{jt} - \Im_{jt-1}| > 10^{-3}$ (where, $jt - 1$ and $jt$ are the consecutive optimization iteration levels), then set $jt = jt + 1$ and go back step 5;
- if $|\Im_{jt}| > \epsilon$ and $|\Im_{jt} - \Im_{jt-1}| < 10^{-3}$, then update the POD bases:
  i. find the new controls $c_{it+1}$ by projecting the optimization controls $\alpha_{it,j}$ onto the original flow domain, and
  ii. set $it = it + 1$ and go back step 2;

7. The adaptive POD optimization procedure is completed.

# Case 1: flow past a cylinder (Re = 100)

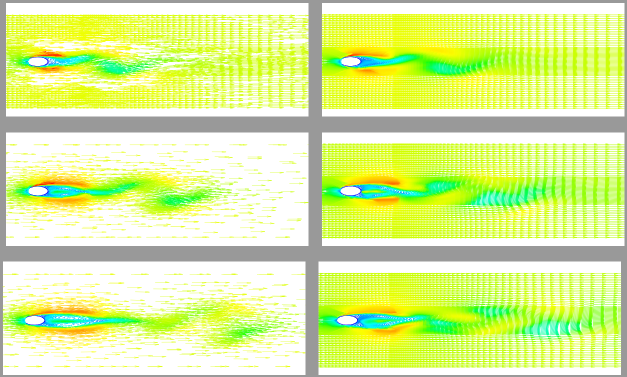


Figure: Case 1: comparison of velocity field between the full and reduced models ($Re = 100$) (left panel: the full model; right panel: the reduced model; top panel: at the initial time level $t = 8$; middle panel: at the time level $t = 10$; bottom panel: at the time level $t = 12$). 20 snapshots and 10 basis functions are chosen for $u$, $v$, $w$ and $p$, for which 95 percent of energy is captured.

## Case 2: Gyre (Re = 250)

| Energy (%) captured (41 snapshots) | Energy (%) captured (81 snapshots) |
|---|---|
| 77.373 (for u, 10 bases)<br>76.003 (for v, 10 bases)<br>81.103 (for p, 10 bases) | 88.614 (for u, 20 bases)<br>89.723 (for v, 20 bases)<br>92.880 (for p, 20 bases) |
| 91.448 (for u, 20 bases)<br>91.693 (for v, 20 bases)<br>94.343 (for p, 20 bases) | 97.025 (for u, 40 bases)<br>97.738 (for v, 40 bases)<br>98.614 (for p, 40 bases) |
| 97.386 (for u, 30 bases)<br>97.624 (for v, 30 bases)<br>98.584 (for p, 30 bases) | 99.458 (for u, 60 bases)<br>99.600 (for v, 60 bases)<br>99.766 (for p, 60 bases) |

Table 1: Energy percentage captured by the POD bases for velocity components, $u$, $v$ and pressure $p$.

# Case 2: Gyre (Re = 250)-computer efficiency

| number of POD bases | CPU time (*hrs*) (41 snapshots) | CPU (*hrs*) (81 snapshots) |
|---|---|---|
| 10 bases for 41 snapshots<br>20 bases for 81 snapshots | 0.77 (reduced by 97% of CPU time compared to the full model) | 1.4 (reduced by 95% of CPU time for the full model) |
| 20 bases for 41 snapshots<br>40 bases for 81 snapshots | 1.30 (reduced by 95% of CPU time compared to the full model) | 2.47 (reduced by 92% of CPU time for the full model) |
| 30 bases for 41 snapshots<br>60 bases for 81 snapshots | 2.00 (reduced by 93% of CPU time compared to the full model) | 11.0 (reduced by 63% of CPU time for the full model) |

Table 2: a list of CPU times required for running the reduced model and the reduced percent of CPU compared with that (30 *hrs*) required for running the full model.Note the actual CPU time required to running the reduced model during the simulation period is less than 1 minute after the POD bases and the time-independent sub-matrices (section 4.3) are calculated.

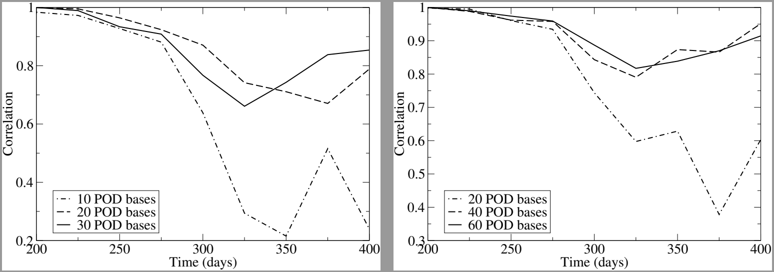



Figure: Case 2: Correlation at time levels(left panel: 41 snapshots; right panel: 81 snapshots)

# Case 2: Gyre (Re = 250)-error analysis

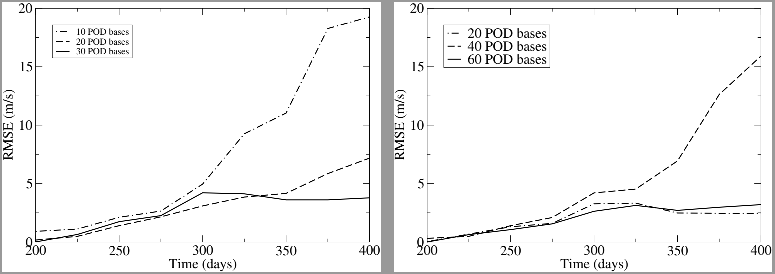



Figure: Case 2: RMS at time levels(left panel: 41 snapshots; right panel: 81 snapshots)

- The POD reduced adjoint model is tested in a computational domain, 1000 *km* by 1000 *km* with a depth of $H = 500\ m$

- The wind forcing on the free surface is given

$$\tau_y = \tau_0 cos(\pi y/L), \quad \tau_x = 0.0$$

where $\tau_x$ and $\tau_y$ are the wind stresses on the free surface along the $x$ and $y$ directions respectively, and $L = 1000\ km$. A maximum zonal wind stress of $\tau_0 = 0.1\ Nm^{-1}$ is applied in the latitude ($y$) direction

- The POD reduced adjoint model is tested in a computational domain, 1000 *km* by 1000 *km* with a depth of $H = 500\ m$

- The wind forcing on the free surface is given

$$\tau_y = \tau_0 cos(\pi y/L), \quad \tau_x = 0.0$$

  where $\tau_x$ and $\tau_y$ are the wind stresses on the free surface along the $x$ and $y$ directions respectively, and $L = 1000\ km$. A maximum zonal wind stress of $\tau_0 = 0.1\ Nm^{-1}$ is applied in the latitude ($y$) direction

- The Coriolis terms are taken into account with the beta-plane approximation ($f = \beta y$) where $\beta = 1.8 \times 10^{-11}$ and the reference density $\rho_0 = 1000\ kgm^{-1}$
- The pseudo-observational data is taken on days 125, 150 and 175 over the computational domain

- The Coriolis terms are taken into account with the beta-plane approximation ($f = \beta y$) where $\beta = 1.8 \times 10^{-11}$ and the reference density $\rho_0 = 1000\ kgm^{-1}$

- The pseudo-observational data is taken on days 125, 150 and 175 over the computational domain

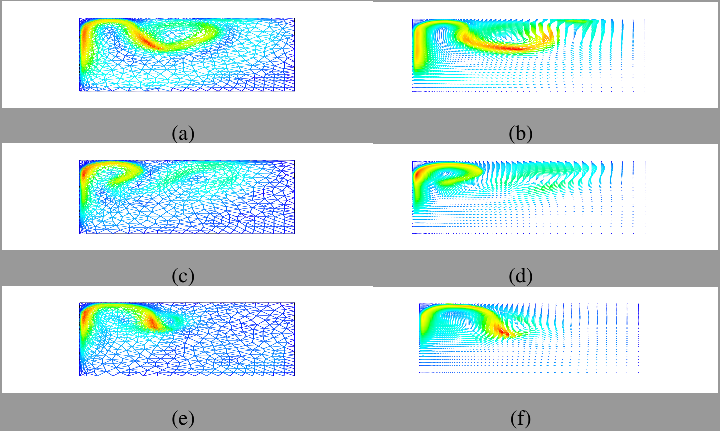



Figure: Comparison between the true velocity field and that from the POD reduced model (driven by the optimised initial conditions) at the time levels: (a) (b) $t = 125$ *days*; (c) (d) $t = 150$ *days*; (e) (f) $t = 175$ *days*. Left panel: the true velocity field; right panel: the optimised velocity field)

## Conclusions :

- Model reduction is proving to be an essential component for real time 4D-Var data assimilation
- While POD is both useful and popular reduction technique for large scale geophysical models its lack of rigorous guarantees requires further research.
- Goal-oriented or dual weighting formulation in which reduced model is chosen to optimally represent a particular output functional (DAS) is found to improve selection of appropriate set of POD snapshots.

# Conclusions :

- Model reduction is proving to be an essential component for real time 4D-Var data assimilation
- While POD is both useful and popular reduction technique for large scale geophysical models its lack of rigorous guarantees requires further research.
- Goal-oriented or dual weighting formulation in which reduced model is chosen to optimally represent a particular output functional (DAS) is found to improve selection of appropriate set of POD snapshots.

- Model reduction is proving to be an essential component for real time 4D-Var data assimilation
- While POD is both useful and popular reduction technique for large scale geophysical models its lack of rigorous guarantees requires further research.
- Goal-oriented or dual weighting formulation in which reduced model is chosen to optimally represent a particular output functional (DAS) is found to improve selection of appropriate set of POD snapshots.

- Full model reduction with adaptive 4D-Var applied to simple ocean model verified well against full model 4D-Var
- Application of POD to global 3D adaptive global Imperial College Ocean Model finite element model (based on goal-oriented mesh adaptivity) yielded encouraging results.

# Conclusions: continued

- Full model reduction with adaptive 4D-Var applied to simple ocean model verified well against full model 4D-Var
- Application of POD to global 3D adaptive global Imperial College Ocean Model finite element model (based on goal-oriented mesh adaptivity) yielded encouraging results.

# Future Efforts:

- Future efforts directed towards full model reduction of 4D-Var in operational atmospheric science models (spectral or finite volume discretization) to be benchmarked against incremental 4D-Var.
- Idealized 4D-Var observation sensitivity experiments indicate that the reduced-order approach is able to properly identify data sets and observation locations of largest forecast sensitivity while providing significant computational savings

# Future Efforts:

- Future efforts directed towards full model reduction of 4D-Var in operational atmospheric science models (spectral or finite volume discretization) to be benchmarked against incremental 4D-Var.
- Idealized 4D-Var observation sensitivity experiments indicate that the reduced-order approach is able to properly identify data sets and observation locations of largest forecast sensitivity while providing significant computational savings