Data Assimilation training course @ CEMRACS
Introduction and variational algorithms

S. Ricci

Acknowledgement to : S. Gürol, A. Weaver, O. Thual, M. Rochoux, A. Piacentini, S. Gratton, S. Barthélémy, N. El Mocayd, E. Emili

CECI, CERFACS-CNRS Toulouse, France

22 July 2016

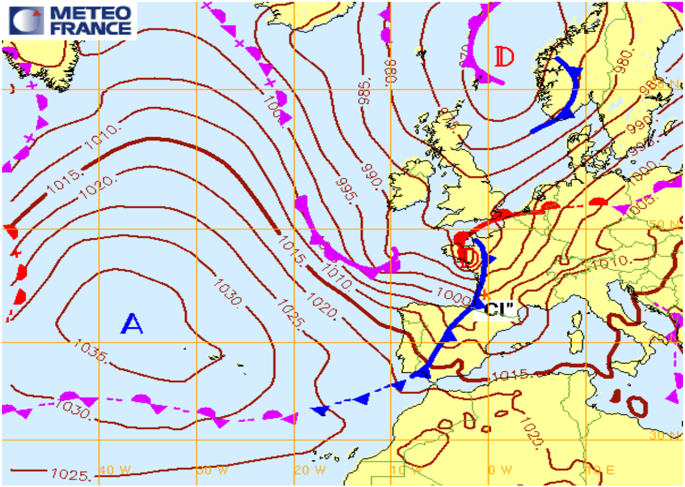
Outline

1. General concepts for Data Assimilation
2. Classical algorithms for Data Assimilation
3. Variational algorithms - Minimization algorithms
4. Error covariance matrices estimation and modeling
5. More on Data Assimilation

Introduction to Data Assimilation for NWF

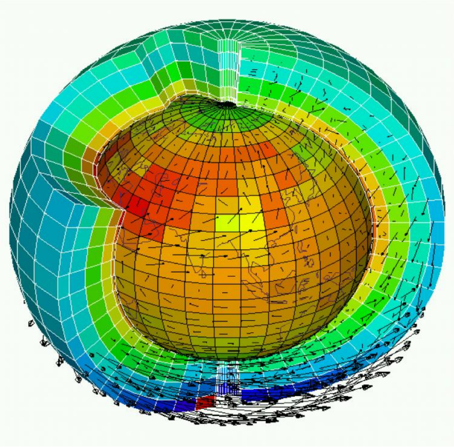
Data Assimilation is widely used for Numerical Weather Forecasting

Analyse Fronts et isobares du 08/12/2006 06hUTC (reseau: 08/12/2006 06hUTC)



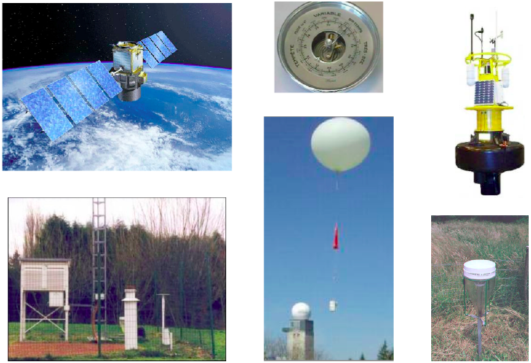
Introduction to Data Assimilation for NWF

Numerical Weather Forecasting models are constantly improving but remain imperfect.



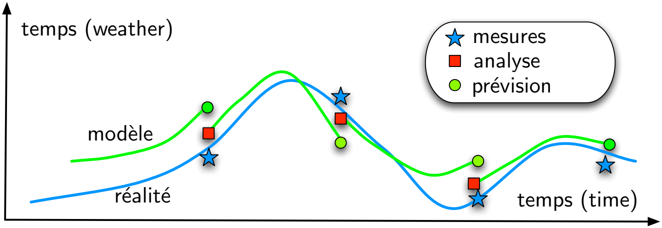
Introduction to Data Assimilation for NWF

A large amount of various types of observations are available thus not providing a complete and perfect description of the system



Data Assimilation framework (1)

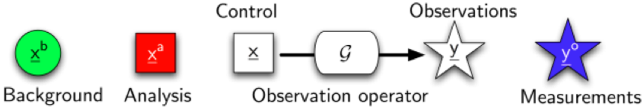
The model provides a forecast from which the observation operator extracts quantities which are compared to actual data observation. The analysis is a field that combines these observations with the model forecast. It is then used as an initial condition for the next forecast. This adjustment is repeated at fixed time intervals.



The analysis minimizes a combination of distance taking into account background, observations and respective error covariance matrices

$$J(\blacksquare) = \frac{1}{2}\| \blacksquare - \bullet \|_B^2 + \frac{1}{2}\| \bigstar - \mathcal{G}(\blacksquare)\|_R^2$$

Data Assimilation framework (2)



The analysis minimizes the cost function:

$$J(\underline{x}) = \frac{1}{2}\left(\underline{x} - \underline{x}^b\right)^T \underline{\underline{B}}^{-1}\left(\underline{x} - \underline{x}^b\right) + \frac{1}{2}\left[\underline{y}^o - \mathcal{G}(\underline{x})\right]^T \underline{\underline{R}}^{-1}\left[\underline{y}^o - \mathcal{G}(\underline{x})\right]$$

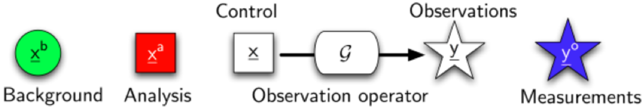
- $\underline{\underline{B}}$ is the background error covariance matrix
- $\underline{\underline{R}}$ is the observation error covariance matrix
- $\mathcal{G}$ is the observation operator

The equivalent of the observation are extracted from the vector through the observation operator $\mathcal{G}$ also denoted by $\mathcal{H}$.

The cost function associates a real number to any vector $\mathbf{x}$ of the control space, given a background state $\mathbf{x}^b$ and a vector of the observation space $\mathbf{y}$.

Control of model state

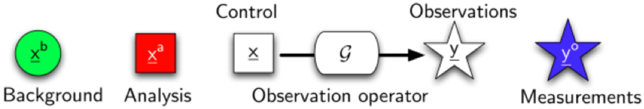
The model state is corrected at a given time, it corresponds to the initial condition for further integration



- $\underline{x}$: the initial condition of a temporal evolution model
- $\underline{x}^b$: the last forecast of the model
- $\mathcal{G}$: the link between the initial condition and the observations
- $\underline{y}$: the observations simulated by the model
- $\underline{y}^o$: the measurements performed on the real system

Control of model parameters

The model parameters are corrected and used for a new model integration for forecast



- $\underline{x}$: the model parameters to be fitted
- $\underline{x}^b$: an a priori knowledge of the parameters
- $\mathcal{G}$: the link between the model parameters and the observations
- $\underline{y}$: the observations simulated by the model
- $\underline{y}^o$: the measurements performed on the real system

Two century of data assimilation

End of the XVIIIth century:

- Planet orbit computations by Gauss
- Least square method by Legendre

Begining of the XXth century:

- Concept of maximum likelyhood by Fisher

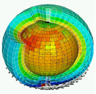
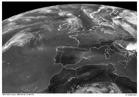
Mid of the XXth century:

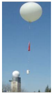
- Kalman filter for the APOLLO program
- Objective analysis of meteorological fields

End of the XXth century:

- 3D-var data assimilation method for weather forecast model
- Gain of 20% forecast quality at Météo-France going to 4D-var

Examples of application (1) - Meteorology



Control vector

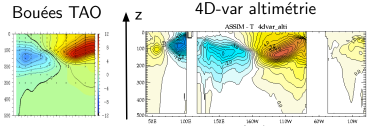
- 3D fields: temperature, pressure, humidity, winds
- Several millions of grid points

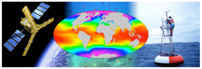
Observation vector

- Satellite data: surface temperature and winds, cloudiness, chemical concentrations...
- In-situ data: temperature, pressure, humidity, winds...
- A million of observations

Weather forecast is a significant example for data assimilation. Most of the time, the analysis is the initial condition of the meteorological model.

Examples of application (2) - Oceanography





Control vector

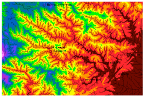
- 3D fields : temperature, salinity, currents
- 2D fields : altimetry
- One million of grid points

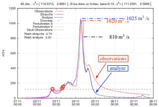
Observation vector

- Satellite data: sea surface temperature,
- In-situ data: temperature, salinity, currents
- Several thousands of observations

Operational oceanography is more recent than operational meteorology. It share common data assimilation methods.

Examples of application (3) - Hydrology





Control vector

- 1D or 2D fields : water level, velocities, ground water content, soil humidity,
- Parameters : drag coefficients, hydraulic conductivity
- A thousand of grid points

Observation vector

- Satellite data: ground water content, altimetry,
- In-situ data: river water level, piezometers, precipitations
- A hundred of observations

Hydrogology deals with much less data than meteorology or oceanography. Nevertheless, data assimilation is used to predict floods as well as scarcity of water resources.

Outline

1 General concepts for Data Assimilation

2 Classical algorithms for Data Assimilation

3 Variational algorithms - Minimization algorithms

4 Error covariance matrices estimation and modeling

5 More on Data Assimilation

A short summary

This lecture focusses on variational algorithms that imply the minimization of a cost function.
Kalman filter algorithms (Vivien Mallet's lecture) imply linear algebra computation and lead to equivalent results under similar assumptions.

- Variational data assimilation methods lead to the minimization of a cost function involving quadratic forms based on the both the background and observation covariance matrices.
- When the observation operator is linear, this formulation of the cost function leads to the Best Linear Unbiased Estimation (BLUE) method and the analyis is the sum of the background plus a gain matrice times the "innovation". This algebra can be used in the nonlinear case with an incremental approximation of the cost function.
- The 4D-Var and Kalman Filter methods deals with a time evolution model and measurement spread on a time interval.

Minimum Variance Approach

- We can find the analysis, $\mathbf{x}^a$, as a linear combination of an observation vector $\mathbf{y}$ and a priori (background) vector $\mathbf{x}^b$.

Minimum Variance Approach

- We can find the analysis, $\mathbf{x}^a$, as a linear combination of an observation vector $\mathbf{y}$ and a priori (background) vector $\mathbf{x}^b$.

Single observation: Find the analysis at observation point

An example: **Estimation of a scalar quantity at a point in space**

- Assume that we have a single observation
- Assume also that we have model forecast of the variable

Minimum Variance Approach

- We can find the analysis, $\mathbf{x}^a$, as a linear combination of an observation vector $\mathbf{y}$ and a priori (background) vector $\mathbf{x}^b$.

Single observation: Find the analysis at observation point

An example: **Estimation of a scalar quantity at a point in space**

- Assume that we have a single observation
- Assume also that we have model forecast of the variable

$$\mathbf{x}^a = \mathbf{x}^b + k(\mathbf{y} - \mathbf{x}^b)$$

Minimum Variance Approach

- We can find the analysis, $\mathbf{x}^a$, as a linear combination of an observation vector $\mathbf{y}$ and a priori (background) vector $\mathbf{x}^b$.

Single observation: Find the analysis at observation point

An example: **Estimation of a scalar quantity at a point in space**

- Assume that we have a single observation
- Assume also that we have model forecast of the variable

$$\mathbf{x}^a = \mathbf{x}^b + k(\mathbf{y} - \mathbf{x}^b)$$

- How to define k?
- Let's now consider the errors involved in this problem.

- Let's denote the truth by $\mathbf{x}^t$

$$\mathbf{x}^a = \mathbf{x}^b + k(\mathbf{y} - \mathbf{x}^b)$$

$$\mathbf{x}^a - \mathbf{x}^t = \mathbf{x}^b - \mathbf{x}^t + k(\mathbf{y} - \mathbf{x}^t - \mathbf{x}^b + \mathbf{x}^t)$$

- Let's denote the truth by $\mathbf{x}^t$

$$\mathbf{x}^a = \mathbf{x}^b + k(\mathbf{y} - \mathbf{x}^b)$$
$$\mathbf{x}^a - \mathbf{x}^t = \mathbf{x}^b - \mathbf{x}^t + k(\mathbf{y} - \mathbf{x}^t - \mathbf{x}^b + \mathbf{x}^t)$$

- Let's define the errors

$$\epsilon^a = \mathbf{x}^a - \mathbf{x}^t$$
$$\epsilon^b = \mathbf{x}^b - \mathbf{x}^t$$
$$\epsilon^y = \mathbf{y} - \mathbf{x}^t$$

- Let's denote the truth by $\mathbf{x}^t$

$$\mathbf{x}^a = \mathbf{x}^b + k(\mathbf{y} - \mathbf{x}^b)$$
$$\mathbf{x}^a - \mathbf{x}^t = \mathbf{x}^b - \mathbf{x}^t + k(\mathbf{y} - \mathbf{x}^t - \mathbf{x}^b + \mathbf{x}^t)$$

- Let's define the errors

$$\epsilon^a = \mathbf{x}^a - \mathbf{x}^t$$
$$\epsilon^b = \mathbf{x}^b - \mathbf{x}^t$$
$$\epsilon^y = \mathbf{y} - \mathbf{x}^t$$

- Then we have $\epsilon^a = \epsilon^b + k(\epsilon^y - \epsilon^b)$

- Let's denote the truth by $\mathbf{x}^t$

$$\mathbf{x}^a = \mathbf{x}^b + k(\mathbf{y} - \mathbf{x}^b)$$
$$\mathbf{x}^a - \mathbf{x}^t = \mathbf{x}^b - \mathbf{x}^t + k(\mathbf{y} - \mathbf{x}^t - \mathbf{x}^b + \mathbf{x}^t)$$

- Let's define the errors

$$\epsilon^a = \mathbf{x}^a - \mathbf{x}^t$$
$$\epsilon^b = \mathbf{x}^b - \mathbf{x}^t$$
$$\epsilon^y = \mathbf{y} - \mathbf{x}^t$$

- Then we have $\epsilon^a = \epsilon^b + k(\epsilon^y - \epsilon^b)$
- If we have many realizations of these errors, then we have the ensemble average of

$$< \epsilon^a > = < \epsilon^b > + k(< \epsilon^y > - < \epsilon^b >)$$

- Let's denote the truth by $\mathbf{x}^t$

$$\mathbf{x}^a = \mathbf{x}^b + k(\mathbf{y} - \mathbf{x}^b)$$
$$\mathbf{x}^a - \mathbf{x}^t = \mathbf{x}^b - \mathbf{x}^t + k(\mathbf{y} - \mathbf{x}^t - \mathbf{x}^b + \mathbf{x}^t)$$

- Let's define the errors

$$\epsilon^a = \mathbf{x}^a - \mathbf{x}^t$$
$$\epsilon^b = \mathbf{x}^b - \mathbf{x}^t$$
$$\epsilon^y = \mathbf{y} - \mathbf{x}^t$$

- Then we have $\epsilon^a = \epsilon^b + k(\epsilon^y - \epsilon^b)$
- If we have many realizations of these errors, then we have the ensemble average of

$$< \epsilon^a > = < \epsilon^b > + k(< \epsilon^y > - < \epsilon^b >)$$

- How to find k?

$$\boxed{< \epsilon^a > = < \epsilon^b > + k(< \epsilon^y > - < \epsilon^b >)}$$

- Let's us look at the variances of errors:

$$< (\epsilon^a)^2 > = < (\epsilon^b)^2 > + k^2 < (\epsilon^y - \epsilon^b)^2 > + 2k < \epsilon^b(\epsilon^y - \epsilon^b) >$$

$$\boxed{< \epsilon^a >=< \epsilon^b > +k(< \epsilon^y > - < \epsilon^b >)}$$

- Let's us look at the variances of errors:

$$< (\epsilon^a)^2 > =< (\epsilon^b)^2 > +k^2 < (\epsilon^y - \epsilon^b)^2 > +2k < \epsilon^b(\epsilon^y - \epsilon^b) >$$

- We want to have the analysis error variance as low as possible

$$\boxed{< \epsilon^a > = < \epsilon^b > + k(< \epsilon^y > - < \epsilon^b >)}$$

- Let's us look at the variances of errors:

$$< (\epsilon^a)^2 > = < (\epsilon^b)^2 > + k^2 < (\epsilon^y - \epsilon^b)^2 > + 2k < \epsilon^b(\epsilon^y - \epsilon^b) >$$

- We want to have the analysis error variance as low as possible
  → Minimize $< (\epsilon^a)^2 >$ with respect to k and solve for k

$$\boxed{< \epsilon^a > = < \epsilon^b > + k(< \epsilon^y > - < \epsilon^b >)}$$

- Let's us look at the variances of errors:

$$< (\epsilon^a)^2 > = < (\epsilon^b)^2 > + k^2 < (\epsilon^y - \epsilon^b)^2 > + 2k < \epsilon^b(\epsilon^y - \epsilon^b) >$$

- We want to have the analysis error variance as low as possible
  → Minimize $< (\epsilon^a)^2 >$ with respect to k and solve for k

$$2k < (\epsilon^y)^2 + (\epsilon^b)^2 > -2 < (\epsilon^b)^2 > = 0$$

where we assumed that the errors in the background and observation are uncorrelated. ($< \epsilon^b \epsilon^y > = 0$)

$$\boxed{< \epsilon^a > = < \epsilon^b > + k(< \epsilon^y > - < \epsilon^b >)}$$

- Let's us look at the variances of errors:

$$< (\epsilon^a)^2 > = < (\epsilon^b)^2 > + k^2 < (\epsilon^y - \epsilon^b)^2 > + 2k < \epsilon^b(\epsilon^y - \epsilon^b) >$$

- We want to have the analysis error variance as low as possible
  → Minimize $< (\epsilon^a)^2 >$ with respect to k and solve for k

$$2k < (\epsilon^y)^2 + (\epsilon^b)^2 > -2 < (\epsilon^b)^2 > = 0$$

where we assumed that the errors in the background and observation are uncorrelated. ($< \epsilon^b \epsilon^y > = 0$)

$$k = \frac{< (\epsilon^b)^2 >}{< (\epsilon^b)^2 > + < (\epsilon^y)^2 >} \quad \Rightarrow \quad k = \frac{(\sigma^b)^2}{(\sigma^b)^2 + (\sigma^y)^2}$$

where $(\sigma^y)^2$ is the observation error variance and $(\sigma^b)^2$ is the background error variance.

Minimum Variance Approach

$$\boxed{\mathbf{x}^a = \mathbf{x}^b + k(\mathbf{y} - \mathbf{x}^b) \quad \text{where} \quad k = \frac{(\sigma^b)^2}{(\sigma^b)^2 + (\sigma^y)^2}}$$

- We have just derived a weight k such that it produces a minimum analysis error variance.
- This weight depends on the relative accuracies of the observations and the background.

Minimum Variance Approach

$$\boxed{\mathbf{x}^a = \mathbf{x}^b + k(\mathbf{y} - \mathbf{x}^b) \quad \text{where} \quad k = \frac{(\sigma^b)^2}{(\sigma^b)^2 + (\sigma^y)^2}}$$

- We have just derived a weight k such that it produces a minimum analysis error variance.
- This weight depends on the relative accuracies of the observations and the background.
- If the observation is perfect, $(\sigma^y)^2 = 0$ and $k = 1$. Then $\mathbf{x}^a = \mathbf{y}$
- If the background is perfect, $(\sigma^b)^2 = 0$ and $k = 0$. Then $\mathbf{x}^a = \mathbf{x}^b$ (The observation is ignored)

Minimum Variance Approach

$$\boxed{\mathbf{x}^a = \mathbf{x}^b + k(\mathbf{y} - \mathbf{x}^b) \quad \text{where} \quad k = \frac{(\sigma^b)^2}{(\sigma^b)^2 + (\sigma^y)^2}}$$

- We have just derived a weight k such that it produces a minimum analysis error variance.
- This weight depends on the relative accuracies of the observations and the background.
- If the observation is perfect, $(\sigma^y)^2 = 0$ and $k = 1$. Then $\mathbf{x}^a = \mathbf{y}$
- If the background is perfect, $(\sigma^b)^2 = 0$ and $k = 0$. Then $\mathbf{x}^a = \mathbf{x}^b$ (The observation is ignored)

Two descriptions of the system are linearly combined according to their accuracies.

Minimum Variance Approach

Multiple observations: Find the analysis at different locations

- We can find the analysis, $\mathbf{x}^a$, as a linear combination of an observation vector $\mathbf{y}$ and a priori (background) vector $\mathbf{x}^b$.

  $$\mathbf{x}^a = \mathbf{x}^b + \mathbf{K}(\mathbf{y} - \mathbf{x}^b)$$

  where the weight (gain matrix) is defined as

  $$\mathbf{K} = \mathbf{B}(\mathbf{B} + \mathbf{R})^{-1}$$

  where $\mathbf{B}$ is the background error covariance matrix, $\mathbf{R}$ is the observation error covariance matrix.
- This weight is determined such that it gives a minimum variance of the estimate.

Minimum Variance Approach

An example: Find analysis at two different locations by assimilating two observations

- $\mathbf{x}^a = \begin{pmatrix} \mathbf{x}_1^a \\ \mathbf{x}_2^a \end{pmatrix}, \quad \mathbf{x}^b = \begin{pmatrix} \mathbf{x}_1^b \\ \mathbf{x}_2^b \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix}$

Minimum Variance Approach

An example: Find analysis at two different locations by assimilating two observations

- $\mathbf{x}^a = \begin{pmatrix} \mathbf{x}_1^a \\ \mathbf{x}_2^a \end{pmatrix}, \quad \mathbf{x}^b = \begin{pmatrix} \mathbf{x}_1^b \\ \mathbf{x}_2^b \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix}$
- $\mathbf{B} = \begin{pmatrix} (\sigma_1^b)^2 & \rho\sigma_1^b\sigma_2^b \\ \rho\sigma_2^b\sigma_1^b & (\sigma_2^b)^2 \end{pmatrix}$ where ρ is the correlation coefficient

Minimum Variance Approach

An example: Find analysis at two different locations by assimilating two observations

- $\mathbf{x}^a = \begin{pmatrix} \mathbf{x}_1^a \\ \mathbf{x}_2^a \end{pmatrix}, \quad \mathbf{x}^b = \begin{pmatrix} \mathbf{x}_1^b \\ \mathbf{x}_2^b \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix}$
- $\mathbf{B} = \begin{pmatrix} (\sigma_1^b)^2 & \rho\sigma_1^b\sigma_2^b \\ \rho\sigma_2^b\sigma_1^b & (\sigma_2^b)^2 \end{pmatrix}$ where ρ is the correlation coefficient
- $\mathbf{R} = \begin{pmatrix} (\sigma^y)^2 & 0 \\ 0 & (\sigma^y)^2 \end{pmatrix}$

Minimum Variance Approach

An example: Find analysis at two different locations by assimilating two observations

- $\mathbf{x}^a = \begin{pmatrix} \mathbf{x}_1^a \\ \mathbf{x}_2^a \end{pmatrix}, \quad \mathbf{x}^b = \begin{pmatrix} \mathbf{x}_1^b \\ \mathbf{x}_2^b \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix}$
- $\mathbf{B} = \begin{pmatrix} (\sigma_1^b)^2 & \rho\sigma_1^b\sigma_2^b \\ \rho\sigma_2^b\sigma_1^b & (\sigma_2^b)^2 \end{pmatrix}$ where ρ is the correlation coefficient
- $\mathbf{R} = \begin{pmatrix} (\sigma^y)^2 & 0 \\ 0 & (\sigma^y)^2 \end{pmatrix}$
- $\mathbf{K} = \mathbf{B}(\mathbf{B}+\mathbf{R})^{-1} = \frac{1}{(1+\alpha_1)(1+\alpha_2)+\rho^2} \begin{pmatrix} (1+\alpha_2)-\rho^2 & \rho\sqrt{\alpha_1}\sqrt{\alpha_2} \\ \rho\sqrt{\alpha_1}\sqrt{\alpha_2} & (1+\alpha_1)-\rho^2 \end{pmatrix}$

  with $\alpha_1 = (\sigma^y)^2/(\sigma_1^b)^2$ and $\alpha_2 = (\sigma^y)^2/(\sigma_2^b)^2$

Minimum Variance Approach

An example: Find analysis at two different locations by assimilating two observations

- If $\rho = 0 \rightarrow \mathbf{K} = \mathbf{B}(\mathbf{B}+\mathbf{R})^{-1} = \begin{pmatrix} (1+\alpha_1)^{-1} & 0 \\ 0 & (1+\alpha_2)^{-1} \end{pmatrix}$

- As in the scalar case the observation and background are combined according to their relative accuracies.

- when the correlation increases then the weight of an observation at the first(second) point on the analysis at the second(first) point increases.

Minimum Variance Approach

General formulation:

- We can find the analysis, $\mathbf{x}^a$, as a linear combination of an observation vector $\mathbf{y}$ and a priori (background) vector $\mathbf{x}^b$.

  $$\mathbf{x}^a = \mathbf{x}^b + \mathbf{K}(\mathbf{y} - \mathcal{H}(\mathbf{x}^b))$$

  where the weight (gain matrix) is defined as

  $$\mathbf{K} = \mathbf{B}\mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{B}\mathbf{H}^{\mathrm{T}} + \mathbf{R})^{-1}$$

  where $\mathbf{B}$ is the background error covariance matrix, $\mathbf{R}$ is the observation error covariance matrix and $\mathbf{H}$ is the linear model of the observation operator.

- This weight is determined such that it gives a minimum variance of the estimate.

The BLUE equation

Statistical interpolation with least square estimation or Best Linear Unbiaised Estimator (BLUE).

- Linearized observation operator : the variations of the observation operator in the vicinity of the background state are linear for any close enough to $\mathbf{x}^b$
- Non trivial errors: **B** and **R** are positive definite matrices
- Unbiased errors
- Uncorrelated errors
- Linear analysis: we look for a an analysis defined by corrections to the background which depend linearly on background observation departures.
- Optimal analysis: we look for an analysis state which is as close as possible to the true state in an r.m.s. sense (i.e. it is a minimum variance estimate)

Observation operator at a glance

The observation operator describes the equivalent of the control $\mathbf{x}$ in the observation space $\mathbf{y} = \mathcal{H}(\mathbf{x})$.

- It can be non linear, especially when controling model parameters.
- The linear model of the observation operator is a $p \times n$ matrix.
- Its formulation may requires the linear code for the physical model either formulated analytically, with an automatic auodifferentiation software or as a local approximation for instance with finite differences.

$$\underline{y} = \underline{\underline{H}}\,\underline{x} \quad \Longleftrightarrow \quad \begin{pmatrix} y_1 \\ \dots \\ y_i \\ \dots \\ y_M \end{pmatrix} = \begin{pmatrix} H_{11} & \dots & H_{1j} & \dots & H_{1N} \\ \dots & \dots & \dots & \dots & \dots \\ H_{i1} & \dots & H_{ij} & \dots & H_{iN} \\ \dots & \dots & \dots & \dots & \dots \\ H_{M1} & \dots & H_{Mj} & \dots & H_{MN} \end{pmatrix} \begin{pmatrix} x_1 \\ \dots \\ x_j \\ \dots \\ x_N \end{pmatrix}$$

Maximum a posteriori probability approach

- The analysis $\mathbf{x}^a$ can be alternatively defined as the maximum of the *a posteriori* p.d.f of the state given the observations and the background:

$$\mathbf{x}^a = \arg\max_{\mathbf{x}} p(\mathbf{x}|\mathbf{y} \text{ and } \mathbf{x}^b)$$

- Using Bayesian approach we have:

$$p(\mathbf{x}|\mathbf{y} \text{ and } \mathbf{x}^b) = \frac{p(\mathbf{y} \text{ and } \mathbf{x}^b|\mathbf{x})p(\mathbf{x})}{p(\mathbf{y} \text{ and } \mathbf{x}^b)}$$

- Using the fact that $p(\mathbf{y} \text{ and } \mathbf{x}^b)$ is not a function of $\mathbf{x}$ and under the assumptions that observation and background errors are uncorrelated we have

$$p(\mathbf{x}|\mathbf{y} \text{ and } \mathbf{x}^b) \propto p(\mathbf{y}|\mathbf{x})p(\mathbf{x}^b|\mathbf{x})$$

Maximum a posteriori probability approach

- Let us define a cost function

$$\begin{aligned}\mathcal{J}(\mathbf{x}) &= -log(p(\mathbf{y}|\mathbf{x})p(\mathbf{x}^b|\mathbf{x})) + \text{const}\\ &= -log(p(\mathbf{y}|\mathbf{x})) - log(p(\mathbf{x}^b|\mathbf{x})) + \text{const}\end{aligned}$$

- The analysis $\mathbf{x}^a$ can then be calculated also by solving a minimization problem:

$$\mathbf{x}^a = \arg\min_{\mathbf{x}} \mathcal{J}(\mathbf{x})$$

Maximum a posteriori probability approach (Special Case)

- Gaussian assumption on the errors leads to

$$\begin{aligned} p(\mathbf{x}^b|\mathbf{x}) &= (2\pi)^{-n/2}|\mathbf{B}|^{-1/2}\exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{x}^b)^T\mathbf{B}^{-1}(\mathbf{x}-\mathbf{x}^b)\right) \\ p(\mathbf{y}|\mathbf{x}) &= (2\pi)^{-m/2}|\mathbf{R}|^{-1/2}\exp\left(-\frac{1}{2}(\mathbf{y}-\mathcal{H}(\mathbf{x}))^T\mathbf{R}^{-1}(\mathbf{y}-\mathcal{H}(\mathbf{x}))\right) \end{aligned}$$

from which the cost function can be written as

$$\mathcal{J}(\mathbf{x}) = \frac{1}{2}(\mathbf{x}^b-\mathbf{x})^{\mathrm{T}}\mathbf{B}^{-1}(\mathbf{x}^b-\mathbf{x}) + \frac{1}{2}(\mathbf{y}-\mathcal{H}(\mathbf{x}))^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}-\mathcal{H}(\mathbf{x}))$$

Maximum a posteriori probability approach (Special Case)

- Gaussian assumption on the errors leads to

$$
\begin{aligned}
p(\mathbf{x}^b|\mathbf{x}) &= (2\pi)^{-n/2}|\mathbf{B}|^{-1/2}\exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{x}^b)^T\mathbf{B}^{-1}(\mathbf{x}-\mathbf{x}^b)\right)\\
p(\mathbf{y}|\mathbf{x}) &= (2\pi)^{-m/2}|\mathbf{R}|^{-1/2}\exp\left(-\frac{1}{2}(\mathbf{y}-\mathcal{H}(\mathbf{x}))^T\mathbf{R}^{-1}(\mathbf{y}-\mathcal{H}(\mathbf{x}))\right)
\end{aligned}
$$

from which the cost function can be written as

$$\mathcal{J}(\mathbf{x}) = \frac{1}{2}(\mathbf{x}^b-\mathbf{x})^{\mathrm{T}}\mathbf{B}^{-1}(\mathbf{x}^b-\mathbf{x}) + \frac{1}{2}(\mathbf{y}-\mathcal{H}(\mathbf{x}))^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}-\mathcal{H}(\mathbf{x}))$$

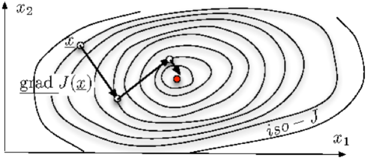
This is the cost function of 3D-Variational Approach!

Minimizing the cost function

If $\mathcal{H}$ is linear, the optimal $\mathbf{x}$ is such that

$$\nabla \mathcal{J}(\mathbf{x}) = 0 \quad \Longleftrightarrow \quad \mathbf{x} = \mathbf{x}^b + \mathbf{B}\mathbf{H}^{\mathrm{T}} \left(\mathbf{H}\mathbf{B}\mathbf{H}^{\mathrm{T}} + \mathbf{R}\right)^{-1} \left(\mathbf{y} - \mathcal{H}(\mathbf{x}^b)\right)$$

$J(\underline{x})$	grad $J(\underline{x})$
$\underline{u}^T \underline{x}$	$\underline{u}$
$\underline{u}^T \underline{\underline{B}}\, \underline{x}$	$\underline{\underline{B}}^T \underline{u}$
$\underline{x}^T \underline{\underline{B}}\, \underline{x}$	$2\, \underline{\underline{B}}^T \underline{x}$
$\underline{x}^T \underline{\underline{H}}^T \underline{\underline{R}}\, \underline{\underline{H}}\, \underline{x}$	$2\, \underline{\underline{H}}\, \underline{\underline{R}}^T \underline{\underline{H}}^T \underline{x}$



This is the same solution at the minimum variance !

Limits of matrix formulation

- Formulating the background and observation error covariances matrices is not straightforward
- When the dimension of the problem is large, either the control vector or the observation vectors are large (or both)
- When the dimension of the problem is large, these matrix can not easily be stored and used for algebra
- The inversion of **B** and **R** is a key challenge whether it is formulated as a function or as an operator.

Limits of matrix formulation

- Formulating the background and observation error covariances matrices is not straightforward
- When the dimension of the problem is large, either the control vector or the observation vectors are large (or both)
- When the dimension of the problem is large, these matrix can not easily be stored and used for algebra
- The inversion of **B** and **R** is a key challenge whether it is formulated as a function or as an operator.

The Matrix formulation is well adapted to small dimension problems!

Limits of matrix formulation

- Formulating the background and observation error covariances matrices is not straightforward
- When the dimension of the problem is large, either the control vector or the observation vectors are large (or both)
- When the dimension of the problem is large, these matrix can not easily be stored and used for algebra
- The inversion of **B** and **R** is a key challenge whether it is formulated as a function or as an operator.

The Matrix formulation is well adapted to small dimension problems!

Defining what a small/large problem is vague: it depends on computational resources and on the definition of the control vector!

Outline

1. General concepts for Data Assimilation
2. Classical algorithms for Data Assimilation
3. Variational algorithms - Minimization algorithms
4. Error covariance matrices estimation and modeling
5. More on Data Assimilation

Variational approach: formulation of the cost function

3D-Var: (Observations are distributed in space but at a single point time)

$$\min_{\mathbf{x}} \in \mathbb{R}^n = \frac{1}{2}\|\mathbf{x}-\mathbf{x}^b\|^2_{\mathbf{B}^{-1}} + \frac{1}{2}\|\mathcal{H}_j(\mathbf{x})-\mathbf{y}_j\|^2_{\mathbf{R}_j^{-1}}$$

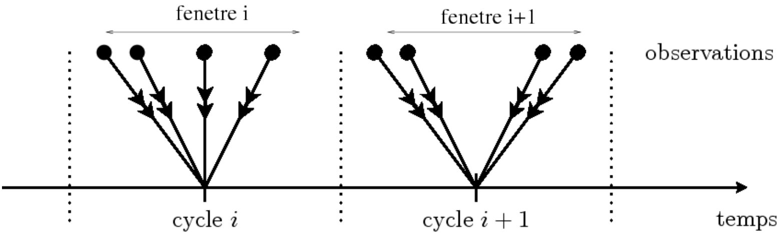
4D-Var: Extension to 3D-Var (Observations are distributed in time)

$$\min_{\mathbf{x}} \in \mathbb{R}^n = \frac{1}{2}\|\mathbf{x}-\mathbf{x}^b\|^2_{\mathbf{B}^{-1}} + \frac{1}{2}\sum_{j=0}^{N}\|\mathcal{H}_j(\mathcal{M}_j(\mathbf{x}))-\mathbf{y}_j\|^2_{\mathbf{R}_j^{-1}}$$

- For large size problem, the minimization is achieved iteratively.

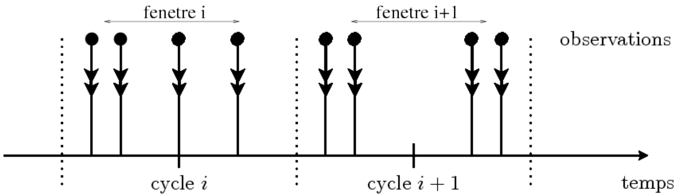
Cycling of the variational analysis (3D-Var case)

- The analysis is achieved at a given time t.
- The observations are gathered at a single time step for each cycle covering several observation times.



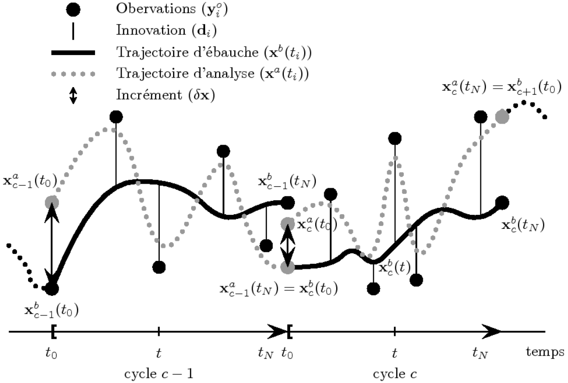
From 3D-Var to 4D-Var : temporal dimension

- The observations are used at their correct time, over an assimilation window.



From 3D-Var to 4D-Var : temporal dimension

- **x** is the optimal initial condition for each cyle, such that the analysed trajectory is in better agreement with the obs. than the background trajectory.


● Obervations ($\mathbf{y}_i^o$)
| Innovation ($\mathbf{d}_i$)
Trajectoire d'ébauche ($\mathbf{x}^b(t_i)$)
Trajectoire d'analyse ($\mathbf{x}^a(t_i)$)
↕ Incrément ($\delta\mathbf{x}$)
$\mathbf{x}_c^a(t_N) = \mathbf{x}_{c+1}^b(t_0)$
$\mathbf{x}_{c-1}^a(t_0)$
$\mathbf{x}_{c-1}^b(t_N)$
$\mathbf{x}_c^a(t_0)$
$\mathbf{x}_c^b(t_N)$
$\mathbf{x}_c^b(t)$
$\mathbf{x}_{c-1}^a(t_N) = \mathbf{x}_c^b(t_0)$
$\mathbf{x}_{c-1}^b(t_0)$
t_0
t
t_N t_0
t
t_N
temps
cycle $c-1$
cycle c

Weak constraint 4D-Var

- So far we assume that the model is perfect. Weak constraint 4D-Var relaxes this assumption.

Weak-constraint 4D-Var:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \frac{1}{2}\|x_0-x_b\|^2_{\mathbf{B}^{-1}}+\frac{1}{2}\sum_{j=0}^{N}\left\|\mathcal{H}_j\left(\mathcal{M}_j\left(x_j\right)\right)-y_j\right\|^2_{\mathbf{R}_j^{-1}}+\frac{1}{2}\sum_{j=1}^{N}\|\underbrace{x_j-\mathcal{M}_j(x_{j-1})}_{q_j}-\bar{q}\|^2_{\mathbf{Q}_j^{-1}}$$

- $\mathbf{x}=\begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_N \end{pmatrix} \in \mathbb{R}^n$ is the control variable (with $x_j = x(t_j)$)
- $\mathbf{Q}_j$ is the model error covariance matrix
- q_j accounts for the model error

Strong constraint 4D-Var

Solve a large-scale non-linear weighted least-squares problem:

$$\arg\min_{\mathbf{x}\in\mathbb{R}^n} = \frac{1}{2}\|\mathbf{x}-\mathbf{x}^b\|^2_{\mathbf{B}^{-1}} + \frac{1}{2}\sum_{j=0}^{N}\left\|\mathcal{H}_j\left(\mathcal{M}_j(\mathbf{x})\right)-\mathbf{y}_j\right\|^2_{\mathbf{R}_j^{-1}}$$

where

- $\mathbf{x} \equiv \mathbf{x}(t_0)$ is the control variable (e.g. the model initial conditions)
- $\mathcal{M}_j$ are model operators: $\mathbf{x}(t_j) = \mathcal{M}_j(\mathbf{x}(t_0))$
- $\mathcal{H}_j$ are observation operators: $\mathbf{y}_j \approx \mathcal{H}_j(\mathbf{x}(t_j))$
- the obervations $\mathbf{y}_j$ and the background $\mathbf{x}^b$ are noisy
- $\mathbf{B}$ and $\mathbf{R}_j$ are error covariance matrices

Strong constraint 4D-Var: Linearized sub-problem

Solve a large-scale non-linear weighted least-squares problem:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \; = \quad \frac{1}{2}\|\mathbf{x}-\mathbf{x}^b\|^2_{\mathbf{B}^{-1}} + \frac{1}{2}\sum_{j=0}^{N}\|\mathcal{H}_j(\mathcal{M}_j(\mathbf{x}))-\mathbf{y}_j\|^2_{\mathbf{R}_j^{-1}}$$

Typically solved using a truncated Gauss-Newton algorithm
(known as incremental 4D-Var in the DA community).

$\longrightarrow$ linearize

$$\begin{aligned}\mathcal{H}_j\left(\mathcal{M}_j(\mathbf{x}^{(k)}+\delta\mathbf{x}^{(k)})\right) &\approx \mathcal{G}_j(\mathbf{x}^{(k)}+\delta\mathbf{x}^{(k)}) \\ &\approx \mathcal{G}_j\left(\mathbf{x}^{(k)}\right)+\mathbf{G}_j\delta\mathbf{x}^{(k)}\end{aligned}$$

$\longrightarrow$ solve the linearized subproblem at iteration k

$$\min_{\delta x^{(k)}} \in \mathbb{R}^n \;\; \frac{1}{2}\|\mathbf{x}^{(k)}+\delta\mathbf{x}^{(k)}-\mathbf{x}^b\|^2_{\mathbf{B}^{-1}} + \frac{1}{2}\sum_{j=0}^{N}\left\|\mathcal{G}_j\left(\mathbf{x}^{(k)}\right)+\mathbf{G}_j\delta\mathbf{x}^{(k)}-\mathbf{y}_j\right\|^2_{\mathbf{R}_j^{-1}}$$

Variational Approach: Linearized sub-problem

$\longrightarrow$ solve the linearized subproblem at iteration k

$$\min_{\delta \mathbf{x}^{(k)} \in \mathbb{R}^n} \quad \frac{1}{2} \|\delta \mathbf{x}^{(k)} - (\mathbf{x}^b - \mathbf{x}^{(k)})\|^2_{\mathbf{B}^{-1}} + \frac{1}{2} \sum_{j=0}^{N} \left\| \mathbf{G}_j \delta \mathbf{x}^{(k)} + \mathcal{G}_j(\mathbf{x}^{(k)}) - \mathbf{y}_j \right\|^2_{\mathbf{R}_j^{-1}}$$

Let's define: $\mathbf{d}_j^{(k)} = \mathbf{y}_j - \mathcal{G}_j(\mathbf{x}^{(k)})$

$$\mathbf{d}^{(k)} = \begin{pmatrix} \mathbf{d}_0^{(k)} \\ \vdots \\ \mathbf{d}_j^{(k)} \\ \vdots \\ \mathbf{d}_N^{(k)} \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} \mathbf{G}_0 \\ \vdots \\ \mathbf{G}_1 \\ \vdots \\ \mathbf{G}_N \end{pmatrix}, \quad \mathbf{R}^{-1} = \begin{pmatrix} \mathbf{R}_0^{-1} & & & & \\ & \ddots & & & \\ & & \mathbf{R}_j^{-1} & & \\ & & & \ddots & \\ & & & & \mathbf{R}_N^{-1} \end{pmatrix}$$

Variational Approach: Linearized sub-problem

Iterative resolution of a linearized sub-problem:

1. $k = 0$
2. Choose an initial vector $\mathbf{x}^{(0)}$, for instance, $\mathbf{x}^{(0)} = \mathbf{x}^b$
3. Solve the linearized subproblem

$$\min_{\delta\mathbf{x}^{(k)} \in \mathbf{R}^n} \quad \frac{1}{2}\|\delta\mathbf{x}^{(k)} - (\mathbf{x}^b - \mathbf{x}^{(k)})\|^2_{\mathbf{B}^{-1}} + \frac{1}{2}\left\|\mathbf{G}^{(k)}\delta\mathbf{x}^{(k)} - \mathbf{d}^{(k)}\right\|^2_{\mathbf{R}^{-1}}$$

Sequence of quadratic minimization problems

4. update $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \delta\mathbf{x}^{(k)}$
5. $k = k + 1$ and goto step 3 till convergence

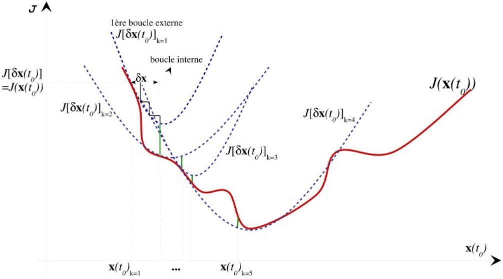
Incremental 4D-Var

→ Nonlinear problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) = \frac{1}{2}||\mathbf{x}-\mathbf{x}^b||^2_{\mathbf{B}^{-1}} + \frac{1}{2}\sum_{j=0}^{N}||\mathcal{H}_j(\mathcal{M}_j(\mathbf{x}))-\mathbf{y}_j||^2_{\mathbf{R}_j^{-1}}$$

→ Quadratic problem

$$\min_{\delta\mathbf{x}^{(k)}\in\mathbb{R}^n} J(\delta\mathbf{x}) = \frac{1}{2}||\delta\mathbf{x}^{(k)} - (\mathbf{x}^b - \mathbf{x}^{(k)})||^2_{\mathbf{B}^{-1}} + \frac{1}{2}\left\|\mathbf{G}\delta\mathbf{x}^{(k)} - \mathbf{d}^{(k)}\right\|^2_{\mathbf{R}^{-1}}$$



Implementation for Incremental 4D-Var resolution

Solve a large-scale non-linear weighted least-squares problem:

$$\min_{\mathbf{x}\in\mathbb{R}^n} = \frac{1}{2}\|\mathbf{x}-\mathbf{x}^b\|^2_{\mathbf{B}^{-1}} + \frac{1}{2}\sum_{j=0}^{N}\left\|\mathcal{H}_j(\mathcal{M}_j(\mathbf{x})) - \mathbf{y}_j\right\|^2_{\mathbf{R}_j^{-1}}$$

Solution algorithm: Incremental 4D-VAR

1. $k = 0$
2. Choose an initial vector $\mathbf{x}^{(0)}$
3. Solve the linearized subproblem

$$\underbrace{(\mathbf{B}^{-1} + \mathbf{G}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{G})}_{\mathbf{A}}\,\delta x = \underbrace{\mathbf{B}^{-1}(\mathbf{x}^b - \mathbf{x}) + \mathbf{G}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{d}}_{\mathbf{b}}$$

4. update $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \delta x^{(k)}$
5. $k = k + 1$ and goto step 3 till convergence

Evaluation of the right hand side b

$$\nabla \mathcal{J}(\mathbf{x}_0) = \underbrace{\mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}_b)}_{\nabla \mathcal{J}_b} + \underbrace{\sum_{i=0}^{N} \mathbf{M}_{i,0}^T \mathbf{H}_i^T \mathbf{R}^{-1}(\mathcal{H}(\mathcal{M}_{i,0}(\mathbf{x}_0)) - \mathbf{y}_i)}_{\nabla \mathcal{J}_o}$$

$$\begin{aligned} \nabla \mathcal{J}_o &= -\{\mathbf{H}_0^T \mathbf{R}^{-1} \mathbf{d}_0 + \mathbf{M}_0^T \mathbf{H}_1^T \mathbf{R}^{-1} \mathbf{d}_1 + ... + \mathbf{M}_0^T ... \mathbf{M}_{Nt-1}^T \mathbf{H}_{Nt}^T \mathbf{R}^{-1} \mathbf{d}_{Nt}\} \\ &= -\{\mathbf{H}_0^T d_0 + \mathbf{M}_0^T [\mathbf{H}_1^T d_1 + \mathbf{M}_1^T [\mathbf{H}_2^T d_2 + ... + \mathbf{M}_{Nt-1}^T \mathbf{H}_{Nt}^T d_{Nt}]]...]\} \end{aligned}$$

Algorithm:Compute $\nabla \mathcal{J}(\mathbf{x}_0)$

1. Forward integration of the dynamical model from $\mathbf{x}_0$
2. Backward integration of the adjoint model
    - $\lambda_N = 0$ *(Initialize adjoint variable)*
    - $\lambda_{k-1} = \mathbf{M}_{k-1}^T (\mathbf{H}_k^T d_k + \lambda_k)$
    - $\lambda_0 = \mathbf{M}_0^T (\mathbf{H}_1^T d_1 + \lambda_1)$
3. $\nabla \mathcal{J}_o = -\lambda_0 + \mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}^b)$

$\longrightarrow$ This algorithm requires only 1 integration of the adjoint model for each k

Computational cost for 4D-Var (Incremental)

→ Computation of the cost function at each iteration of the minimization:

- Integration of the non linear model from $\mathbf{x}(t_0)$ to $\mathbf{x}(t_N)$ in order to express each $\mathbf{x}(t_i)$
- Compute and store the difference between the model state and the observation : $\mathbf{y}_i^o - H_i(\mathbf{x}_i)$, cumulate in J^o

→ Computation of the gradient of the cost function:

- Integrate the adjoint model $\mathbf{M}_{t_i,t_{i-1}}^T$ (at least 4 times more expensive than the direct model)
- The last value of the adjoint variable $\mathbf{x}^*(t_0)$ is the gradient of J^o

Get a new optimal from the minimizer
do loop.... until minimizer has converged

Assumption: the tangent linear and the adjoint codes are available.

Incremental formulation 3D-FGAT

Strong assumption on the tangent linear model $\mathbf{M}_{0,i} = I$

→ The increment $\delta\mathbf{x}$ is not propagated by the T.L physics.
→ It is constant over the assimilation cycle and can be applied at any time over cycle i.

- The 3D-FGAT cost function and its gradient read:

$$J\left(\delta\mathbf{x}\right) = \frac{1}{2}\delta\mathbf{x}^T\mathbf{B}^{-1}\delta\mathbf{x} + \frac{1}{2}\sum_{i=0}^{N}\left(\mathbf{d}_i + \mathbf{H}_i\delta\mathbf{x}\right)^T\mathbf{R}_i^{-1}\left(\mathbf{d}_i + \mathbf{H}_i\delta\mathbf{x}\right)$$

- The generalized obs. op. sums up to a selection, interpolation H, potentially linear.

$$\nabla J\left(\delta\mathbf{x}\right) = \mathbf{B}^{-1}\delta\mathbf{x} + \sum_{i=0}^{N}\mathbf{H}_i^T\mathbf{R}_i^{-1}\left(\mathbf{d}_i + \mathbf{H}_i\delta\mathbf{x}\right)$$

- The comp. cost for the estimation of J and ∇J is significantly reduced.

Pros and cons of the incremental formulation

The 4D-Inc algorithm implies

- a single integration of the non linear model to compute the innovation
- as many integrations of the T.L and adjoint as needed for the minimization

The 3D-FGAT algorithm implies

- no need for T.L and adjoint codes
- reduction of the computational cost for the optimization
- approximation of the linear physics around the background

Solution with primal approach

From optimality conditions ($\nabla J(\delta\mathbf{x}) = 0$), at Gauss-Newton iteration k we have

$$\underbrace{\left(\mathbf{B}^{-1} + \mathbf{G}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{G}\right)}_{\mathbf{A}} \delta\mathbf{x} = \underbrace{\mathbf{B}^{-1}(\mathbf{x}^{b} - \mathbf{x}) + \mathbf{G}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{d}}_{\mathbf{b}}$$

where **A** is a large, symmetric and positive definite matrix.(The superscript k has been dropped.)

Algorithmic considerations:

- very large problem size: $m \sim 10^7$ observations, $n \sim 10^8$ unknowns

Solution with primal approach

From optimality conditions ($\nabla J(\delta \mathbf{x}) = 0$), at Gauss-Newton iteration k we have

$$\underbrace{(\mathbf{B}^{-1} + \mathbf{G}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{G})}_{\mathbf{A}}\,\delta\mathbf{x} = \underbrace{\mathbf{B}^{-1}(\mathbf{x}^{b} - \mathbf{x}) + \mathbf{G}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{d}}_{\mathbf{b}}$$

where **A** is a large, symmetric and positive definite matrix.(The superscript k has been dropped.)

Algorithmic considerations:

- very large problem size: $m \sim 10^7$ observations, $n \sim 10^8$ unknowns
- very few iterations performed

  → use iterative methods for spd systems (Krylov subspace methods)?

  → numerical efficiency ?

  → reduce the computational cost and memory (dual approach, preconditioning?)

- noisy data: need reliable error estimates

Conjugate Gradient method

- CG method generates the sequences $\delta\mathbf{x}_k, k = 1, ..., l$

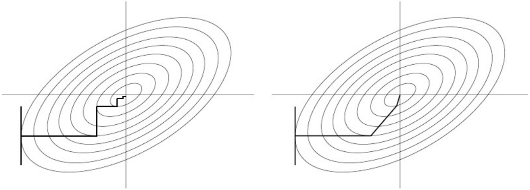
$$\delta\mathbf{x}_{k+1} = \delta\mathbf{x}_k + \alpha_k \mathbf{p}_k$$

where $\mathbf{p}_k$ is the search direction and conjugate w.r.t $\mathbf{A}$, i.e.

$$\mathbf{p}_i^{\mathrm{T}} \mathbf{A} \mathbf{p}_k = 0, \text{ for all } i \neq k$$

and α_k is the step length minimizing the cost function along $\mathbf{x}_k + \alpha \mathbf{p}_k$,

$$\alpha_k = \frac{\mathbf{r}_k^{\mathrm{T}} \mathbf{p}_k}{\mathbf{p}_k^{\mathrm{T}} \mathbf{A} \mathbf{p}_k}$$



Steepest descent vs. conjugate gradient.

Alternative Methods: Dual Approach

- Exact solution writes

$$\mathbf{x}^b - \mathbf{x} + \underbrace{\left(\mathbf{B}^{-1} + \mathbf{G}^T\mathbf{R}^{-1}\mathbf{G}\right)^{-1}\mathbf{G}^T\mathbf{R}^{-1}\left(\mathbf{d} - \mathbf{G}(\mathbf{x}^b - \mathbf{x})\right)}_{\text{requires solving a linear system } \textbf{iteratively} \text{ in } \mathbb{R}^n}$$

- Alternatively, the exact solution can be rewritten from duality theory or using Sherman-Morrison-Woodbury formula

$$\mathbf{x}^b - \mathbf{x} + \mathbf{B}\mathbf{G}^T \quad \underbrace{(\mathbf{G}\mathbf{B}\mathbf{G}^T + \mathbf{R})^{-1}(\mathbf{d} - \mathbf{G}(\mathbf{x}^b - \mathbf{x}))}_{\text{requires solving a linear system } \textbf{iteratively} \text{ in } \mathbb{R}^m}$$

Alternative Methods: Dual Approach

- Exact solution writes

$$\mathbf{x}^b - \mathbf{x} + \underbrace{\left(\mathbf{B}^{-1} + \mathbf{G}^T\mathbf{R}^{-1}\mathbf{G}\right)^{-1}\mathbf{G}^T\mathbf{R}^{-1}\left(\mathbf{d} - \mathbf{G}(\mathbf{x}^b - \mathbf{x})\right)}_{\text{requires solving a linear system } \textbf{iteratively} \text{ in } \mathbb{R}^n}$$

- Alternatively, the exact solution can be rewritten from duality theory or using Sherman-Morrison-Woodbury formula

$$\mathbf{x}^b - \mathbf{x} + \mathbf{B}\mathbf{G}^T \underbrace{(\mathbf{G}\mathbf{B}\mathbf{G}^T + \mathbf{R})^{-1}(\mathbf{d} - \mathbf{G}(\mathbf{x}^b - \mathbf{x}))}_{\text{requires solving a linear system } \textbf{iteratively} \text{ in } \mathbb{R}^m}$$

- If $m << n$, then performing the minimization in $\mathbb{R}^m$ can reduce memory and computational cost.
- This situation is typical in ocean data assimilation systems where $n \approx 10^7$ and $m \approx 10^5$

Outline

1. General concepts for Data Assimilation
2. Classical algorithms for Data Assimilation
3. Variational algorithms - Minimization algorithms
4. Error covariance matrices estimation and modeling
5. More on Data Assimilation

Role of **B**

- Recall the solution:

$$\delta\mathbf{x}^{\mathrm{a}} = \mathbf{B}\,\mathbf{G}^{\mathrm{T}}\left(\mathbf{G}\,\mathbf{B}\,\mathbf{G}^{\mathrm{T}} + \mathbf{R}\right)^{-1}\mathbf{d}$$

- To help understand the role of **B**, we can write

$$\delta\mathbf{x}^{\mathrm{a}} = \mathbf{B}\,\alpha = \mathbf{B}\mathbf{G}^{\mathrm{T}}\beta$$

where

$$\beta = \left(\mathbf{G}\,\mathbf{B}\,\mathbf{G}^{\mathrm{T}} + \mathbf{R}\right)^{-1}\mathbf{d}$$

is a P-dimensional vector, and

$$\alpha = \mathbf{G}^{\mathrm{T}}\beta$$

is a N-dimensional vector.

Role of **B**

- Writing the solution in explicit matrix notation:

$$\delta\mathbf{x}^{\mathrm{a}} = \mathbf{B}\,\boldsymbol{\alpha} = \begin{pmatrix} \mathbf{b}_1 & \dots & \mathbf{b}_i & \dots & \mathbf{b}_N \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_i \\ \vdots \\ \alpha_N \end{pmatrix} = \sum_{i=1}^{N} \alpha_i \mathbf{b}_i$$

  where $\mathbf{b}_i$ are the columns of **B**.

- We can also write it as

$$\delta\mathbf{x}^{\mathrm{a}} = \mathbf{B}\mathbf{G}^{\mathrm{T}}\boldsymbol{\beta} = \begin{pmatrix} \mathbf{r}_1 & \dots & \mathbf{r}_j & \dots & \mathbf{r}_P \end{pmatrix} \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_j \\ \vdots \\ \beta_P \end{pmatrix} = \sum_{j=1}^{P} \beta_j \mathbf{r}_j$$

  where $\mathbf{r}_j$ are the columns of $\mathbf{B}\mathbf{G}^{\mathrm{T}}$.

Role of $\mathbf{B}$

- So the solution is in the column space of $\mathbf{B} \approx E[\tilde{\epsilon}^{\mathrm{b}}(\tilde{\epsilon}^{\mathrm{b}})^{\mathrm{T}}]$:

$$\delta \mathbf{x}^{\mathrm{a}} = \sum_{i=1}^{N} \alpha_i \mathbf{b}_i$$

- The column vector $\mathbf{b}_i$ is the covariance of the background error at model grid-point i with the background error at all other grid points.
- The solution is also in the column space of $\mathbf{B}\mathbf{G}^{\mathrm{T}} \approx E[\tilde{\epsilon}^{\mathrm{b}}(\mathbf{G}\tilde{\epsilon}^{\mathrm{b}})^{\mathrm{T}}]$:

$$\delta \mathbf{x}^{\mathrm{a}} = \sum_{j=1}^{P} \beta_j \mathbf{r}_j$$

- The column vector $\mathbf{r}_j$ is the covariance of the background error of the observed quantity j with the background error at all model grid-points.
- The $\mathbf{r}_j$ vectors are often called the representers. Note the role of the adjoint operator $\mathbf{G}^{\mathrm{T}}$ in determining the solution.

Example: 3D-Var with a single observation

- Consider a single innovation ($\mathbf{d} = d$) of a model variable located exactly at a model grid point (the ith element of the state vector $\mathbf{x}$):

$$\mathbf{G} = \mathbf{H} = (\cdots\ 0 \underbrace{\ 1\ }_{i\text{th element}} 0\ \cdots) = \mathbf{e}_i^{\mathrm{T}}$$

$$\mathbf{R} = \sigma_{\mathrm{o}}^2 \qquad \text{(scalar variance)}$$

$$\mathbf{G}\,\mathbf{B}\,\mathbf{G}^{\mathrm{T}} \;=\; \mathbf{e}_i^{\mathrm{T}}\mathbf{B}\,\mathbf{e}_i \;=\; \{\sigma_{\mathrm{b}}^2\}_i \qquad \text{(scalar variance)}$$

$$\mathbf{r}_1 \;=\; \mathbf{B}\mathbf{G}^{\mathrm{T}} \;=\; \mathbf{B}\,\mathbf{e}_i \;=\; \mathbf{b}_i \qquad \text{(one representer)}$$

$$\delta\mathbf{x}^{\mathrm{a}} = \mathbf{B}\mathbf{G}^{\mathrm{T}}\left(\mathbf{G}\,\mathbf{B}\,\mathbf{G}^{\mathrm{T}} + \mathbf{R}\right)^{-1}\mathbf{d} = \left(\frac{d}{\{\sigma_{\mathrm{b}}^2\}_i \,+\, \sigma_{\mathrm{o}}^2}\right)\mathbf{b}_i$$

Example: 3D-Var with a single observation

- The solution at the observation point is

$$\delta y^{\mathrm{a}} = \mathbf{G}\,\delta\mathbf{x}^{\mathrm{a}} = \left(\frac{d}{\{\sigma_{\mathrm{b}}^2\}_i + \sigma_{\mathrm{o}}^2}\right)\mathbf{e}_i^{\mathrm{T}}\mathbf{b}_i = \underbrace{\left(\frac{\{\sigma_{\mathrm{b}}^2\}_i}{\{\sigma_{\mathrm{b}}^2\}_i + \sigma_{\mathrm{o}}^2}\right)}_{\gamma} d$$

- $|\delta y^{\mathrm{a}}| < |d| = |y^{\mathrm{o}} - y^{\mathrm{b}}|$ since $0 < \gamma < 1$ which implies that the analysis lies between the background and observation (as expected):

$$\begin{aligned}
|y^{\mathrm{a}} - y^{\mathrm{b}}| &= |\delta y^{\mathrm{a}}| \\
&< |d| \\
|y^{\mathrm{a}} - y^{\mathrm{o}}| &= |y^{\mathrm{b}} + \delta y^{\mathrm{a}} - y^{\mathrm{o}}| \\
&= |\delta y^{\mathrm{a}} - d| \\
&= |\gamma d - d| \\
&= (1-\gamma)|d| \\
&< |d|
\end{aligned}$$

Example: 4D-Var with a single observation

- Consider a single innovation as in the 3D-Var example but located at the **end** of a time interval $t_0 \leq t_i \leq t_M$.
- The linearized model is now used to transport the increment so

$$\mathbf{G} \; = \; \mathbf{e}_i^{\mathrm{T}} \, \mathbf{M}(t_0, t_M)$$

$$\mathbf{G}\,\mathbf{B}\,\mathbf{G}^{\mathrm{T}} \; = \; \mathbf{e}_i^{\mathrm{T}} \, \underbrace{\mathbf{M}(t_0, t_M)\,\mathbf{B}\,\mathbf{M}(t_0, t_M)^{\mathrm{T}}}_{\mathbf{P}^{\mathrm{b}}(t_M)} \, \mathbf{e}_i \; = \; \{\sigma_{\mathrm{b}}^2(t_M)\}_i$$

- $\mathbf{P}^{\mathrm{b}}(t_M)$ describes forecast error covariance propagation from t_0 to t_M in a perfect model (cf. Extended Kalman Filter).

$$\begin{aligned}
\mathbf{r}_1(t_0) \; &= \; \mathbf{B}\,\mathbf{M}(t_0, t_M)^{\mathrm{T}}\mathbf{e}_i && \text{(representer at } t_0\text{)} \\
\mathbf{r}_1(t_M) \; &= \; \mathbf{P}^{\mathrm{b}}(t_M)\,\mathbf{e}_i \; = \; \mathbf{p}_i^{\mathrm{b}}(t_M) && \text{(representer at } t_M\text{)} \\
\delta\mathbf{x}^{\mathrm{a}}(t_0) \; &= \; \left(\frac{d}{\{\sigma_{\mathrm{b}}^2(t_M)\}_i \; + \; \sigma_{\mathrm{o}}^2} \right) \mathbf{r}_1(t_0) \\
\delta\mathbf{x}^{\mathrm{a}}(t_M) \; &= \; \left(\frac{d}{\{\sigma_{\mathrm{b}}^2(t_M)\}_i \; + \; \sigma_{\mathrm{o}}^2} \right) \mathbf{r}_1(t_M)
\end{aligned}$$

Example: 4D-Var with a single observation

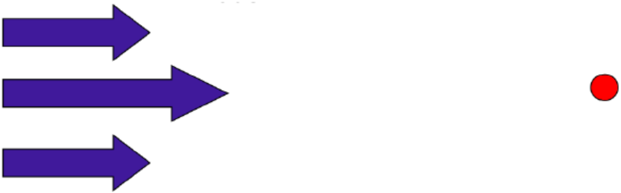
- The solution at the observation point is

$$\begin{aligned}\delta y^{\mathrm{a}} &= \left(\frac{d}{\{\sigma_{\mathrm{b}}^2(t_M)\}_i \; + \; \sigma_{\mathrm{o}}^2} \right) \mathbf{e}_i^{\mathrm{T}} \mathbf{p}_i^{\mathrm{b}}(t_M) \\ &= \left(\frac{\{\sigma_{\mathrm{b}}^2(t_M)\}_i}{\{\sigma_{\mathrm{b}}^2(t_M)\}_i \; + \; \sigma_{\mathrm{o}}^2} \right) d\end{aligned}$$

- The 3D-Var solution earlier is just a special case with $\mathbf{M}(t_0, t_M) = \mathbf{I}$.

Schematic illustration of $\mathbf{GBG}^T in4D-Var$

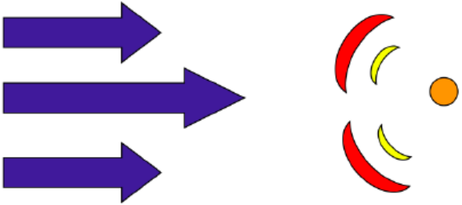
$$\mathbf{H} = \mathbf{e}_i = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \leftarrow i\text{th element}$$



Zonal shear flow

Schematic illustration of $\mathbf{GBG}^T in4D - Var$

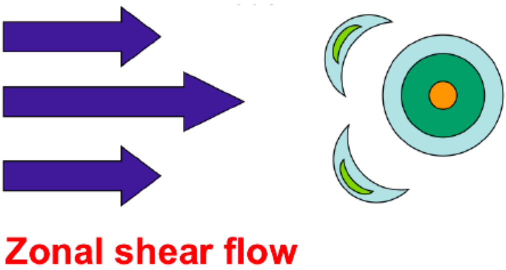
$$\widehat{\mathbf{r}}_1(t_0) = \mathbf{M}(t_0, t_M)^{\mathrm{T}}\,\mathbf{e}_i$$



Zonal shear flow

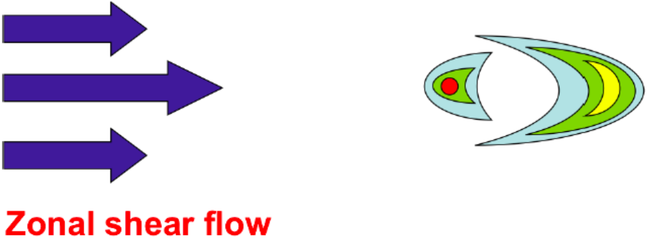
Schematic illustration of $\mathbf{GBG}^T in4D - Var$

$$\mathbf{r}_1(t_0) = \mathbf{B}\,\mathbf{M}(t_0, t_M)^{\mathrm{T}}\,\mathbf{e}_i$$



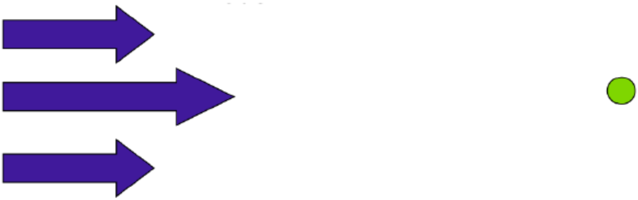
Schematic illustration of $\mathbf{GBG}^T in4D - Var$

$$\mathbf{r}_1(t_M) = \mathbf{M}(t_0, t_M)\,\mathbf{B}\,\mathbf{M}(t_0, t_M)^{\mathrm{T}}\,\mathbf{e}_i = \mathbf{p}_i^{\mathrm{b}}(t_M)$$



Schematic illustration of $\mathbf{GBG}^T in4D-Var$

$$\left\{\sigma_{\mathrm{b}}^2(t_M)\right\}_i = \mathbf{e}_i^{\mathrm{T}}\,\mathbf{M}(t_0,t_M)\,\mathbf{B}\,\mathbf{M}(t_0,t_M)^{\mathrm{T}}\,\mathbf{e}_i$$



Zonal shear flow

Specifying **B** in practice

- Fundamental role of **B** in determining the solution of the data assimilation problem.
- Recall the expression for the analysis increment

$$\delta\mathbf{x}^{\mathrm{a}} \;=\; \mathbf{B}\,\mathbf{G}^{\mathrm{T}}\left(\mathbf{G}\,\mathbf{B}\,\mathbf{G}^{\mathrm{T}} + \mathbf{R}\right)^{-1}\mathbf{d}$$

  - $\mathbf{d} = \mathbf{y}^{\mathrm{o}} - G(\mathbf{x}^{\mathrm{b}})$ is the P-dimensional innovation vector.
  - $\mathbf{y}^{\mathrm{o}}$ is the P-dimensional observation vector.
  - $\mathbf{x}^{\mathrm{b}}$ is the N-dimensional background state.
  - $G(\,\cdot\,)$ is the (generalized) observation operator.
  - $\mathbf{G}$ is the (generalized) observation operator linearized about $\mathbf{x}^{\mathrm{b}}$.
  - $\mathbf{R}$ is the specified observation error covariance matrix.
- The analysis increment can be expressed as a linear combination of the column vectors $\mathbf{b}_i$ of **B**:

$$\delta\mathbf{x}^{\mathrm{a}} \;=\; \sum_{i=1}^{N} \alpha_i\,\mathbf{b}_i$$

Specifying **B** in practice

- **B** is a symmetric $N \times N$ matrix, so the number of independent elements that need to be specified is:

$$\sum_{i=1}^{N} i = \frac{1}{2}(N^2 + N)$$

- For large problems, of the order of $N \sim 10^6$ or greater, we need to define a matrix with more than 10^{12} elements! It is simply not possible to store and manipulate a matrix this big.
- Even for medium size problems ($N \sim 10^3$), full matrix representations of **B** are hardly practical.
- Even if we could store **B** as a matrix, we usually don't have enough reliable information to specify all elements correctly. This is especially true in ocean and atmospheric data assimilation.

Covariance estimation using innovations

- The innovation vector contains valuable information about observation and background errors:

$$\begin{aligned}
\mathbf{d}^{\mathrm{b}} &= \mathbf{y}^{\mathrm{o}} - G(\mathbf{x}^{\mathrm{b}}) \\
&= \mathbf{y}^{\mathrm{o}} - \mathbf{y}^{\mathrm{t}} + \mathbf{y}^{\mathrm{t}} - G(\mathbf{x}^{\mathrm{t}}) + G(\mathbf{x}^{\mathrm{t}}) - G(\mathbf{x}^{\mathrm{b}}) \\
&= \boldsymbol{\epsilon}^{\mathrm{m}} + \boldsymbol{\epsilon}^{\mathrm{r}} + G(\mathbf{x}^{\mathrm{t}}) - G(\mathbf{x}^{\mathrm{t}} + \boldsymbol{\epsilon}^{\mathrm{b}}) \\
&\approx \boldsymbol{\epsilon}^{\mathrm{o}} + G(\mathbf{x}^{\mathrm{t}}) - G(\mathbf{x}^{\mathrm{t}}) - \mathbf{G}\,\boldsymbol{\epsilon}^{\mathrm{b}} \\
&= \boldsymbol{\epsilon}^{\mathrm{o}} - \mathbf{G}\,\boldsymbol{\epsilon}^{\mathrm{b}}
\end{aligned}$$

- The innovation covariance matrix is then

$$\begin{aligned}
E[\widetilde{\mathbf{d}}^{\mathrm{b}}(\widetilde{\mathbf{d}}^{\mathrm{b}})^{\mathrm{T}}] &\approx E[(\widetilde{\boldsymbol{\epsilon}}^{\mathrm{o}} - \mathbf{G}\,\widetilde{\boldsymbol{\epsilon}}^{\mathrm{b}})(\widetilde{\boldsymbol{\epsilon}}^{\mathrm{o}} - \mathbf{G}\,\widetilde{\boldsymbol{\epsilon}}^{\mathrm{b}})^{\mathrm{T}}] \\
&= E[\widetilde{\boldsymbol{\epsilon}}^{\mathrm{o}}(\widetilde{\boldsymbol{\epsilon}}^{\mathrm{o}})^{\mathrm{T}}] + \mathbf{G}\,E[\widetilde{\boldsymbol{\epsilon}}^{\mathrm{b}}(\widetilde{\boldsymbol{\epsilon}}^{\mathrm{b}})^{\mathrm{T}}]\,\mathbf{G}^{\mathrm{T}} \\
&= \mathbf{R}^{\mathrm{t}} + \mathbf{G}\,\mathbf{B}^{\mathrm{t}}\,\mathbf{G}^{\mathrm{T}},
\end{aligned}$$

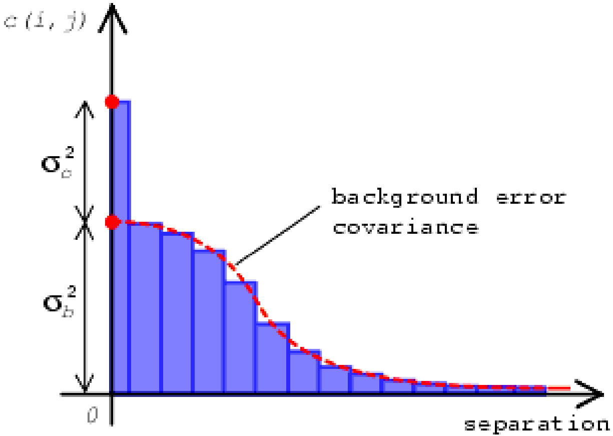
assuming $\boldsymbol{\epsilon}^{\mathrm{o}}$ and $\boldsymbol{\epsilon}^{\mathrm{b}}$ are uncorrelated: $E[\widetilde{\boldsymbol{\epsilon}}^{\mathrm{o}}(\widetilde{\boldsymbol{\epsilon}}^{\mathrm{b}})^{\mathrm{T}}] = \mathbf{0}$.

Covariance estimation using innovations

- In practice, we invoke the ergodicity assumption and replace $E[\,\cdot\,]$ by a sample average $\overline{(\,\cdot\,)} = \frac{1}{N_\mathrm{s}} \sum_1^{N_\mathrm{s}} (\,\cdot\,)$ over an appropriately chosen space and/or time domain Ω.
- The fundamental problem is how to deduce information about $\mathbf{R}^\mathrm{t}$ and $\mathbf{B}^\mathrm{t}$ given information about their sum.
- The trick is to select observations with approximately uncorrelated errors and then attribute any correlated signal in the innovation covariance to $\mathbf{B}$:

$$\begin{aligned} \mathrm{var}(d_i)_\Omega &= \mathrm{var}(\epsilon_i^\mathrm{o})_\Omega + \mathrm{var}(\epsilon_i^\mathrm{b})_\Omega \\ \mathrm{cov}(d_i, d_j)_\Omega &= \mathrm{cov}(\epsilon_i^\mathrm{b}, \epsilon_j^\mathrm{b})_\Omega, \quad i \neq j \end{aligned}$$

Covariance estimation using innovations



From Bouttier and Courtier (1999)

Covariance estimation using innovations

- This is arguably the purest method for estimating error covariances ☺, **but** . . .
- Requires a dense observation network.
  - If not available then considerable spatial and temporal averaging is required.
- Requires observations with uncorrelated errors.
  - Even though we often specify $\mathbf{R}$ to be diagonal, this assumption is known to be suspect for important data sets (e.g., satellite data).
  - Densely distributed observations will likely contain a substantial amount of correlated representativeness error.
- Requires observations of background variables.
  - Generates estimates of $\mathbf{G}\,\mathbf{B}\,\mathbf{G}^{\mathrm{T}}$ (i.e., in observation space) but what we want is an estimate of $\mathbf{B}$ (i.e., in model control space).
- The last three points are serious limitations in practice. ☹
- This method is most useful for checking covariances generated by other methods ☺

Covariance estimation using Desroziers diagnostics

- The analysis increment is

$$\delta \mathbf{x}^{\mathrm{a}} \;=\; \mathbf{B}\,\mathbf{G}^{\mathrm{T}}\left(\mathbf{G}\,\mathbf{B}\,\mathbf{G}^{\mathrm{T}} + \mathbf{R}\right)^{-1}\mathbf{d}^{\mathrm{b}}$$

- The covariance of $\mathbf{G}\,\delta\mathbf{x}^{\mathrm{a}} = \mathbf{d}^{\mathrm{b}} - \mathbf{d}^{\mathrm{a}}$ and $\mathbf{d}^{\mathrm{b}}$ is then

$$\begin{aligned}
E[(\widetilde{\mathbf{d}}^{\mathrm{b}} - \widetilde{\mathbf{d}}^{\mathrm{a}})(\widetilde{\mathbf{d}}^{\mathrm{b}})^{\mathrm{T}}] &= \mathbf{G}\,E[\,\delta\widetilde{\mathbf{x}}^{\,\mathrm{a}}(\widetilde{\mathbf{d}}^{\mathrm{b}})^{\mathrm{T}}] \\
&= \mathbf{G}\,\mathbf{B}\,\mathbf{G}^{\mathrm{T}}\left(\mathbf{G}\,\mathbf{B}\,\mathbf{G}^{\mathrm{T}} + \mathbf{R}\right)^{-1} E[\widetilde{\mathbf{d}}^{\mathrm{b}}(\widetilde{\mathbf{d}}^{\mathrm{b}})^{\mathrm{T}}] \\
&= \mathbf{G}\,\mathbf{B}\,\mathbf{G}^{\mathrm{T}}\left(\mathbf{G}\,\mathbf{B}\,\mathbf{G}^{\mathrm{T}} + \mathbf{R}\right)^{-1}\left(\mathbf{R}^{\mathrm{t}} + \mathbf{G}\,\mathbf{B}^{\mathrm{t}}\,\mathbf{G}^{\mathrm{T}}\right)
\end{aligned}$$

- Assuming that **B** and **R** are good approximations to $\mathbf{B}^{\mathrm{t}}$ and $\mathbf{R}^{\mathrm{t}}$ then

$$\begin{aligned}
E[(\widetilde{\mathbf{d}}^{\mathrm{b}} - \widetilde{\mathbf{d}^{\mathrm{a}}})(\widetilde{\mathbf{d}}^{\mathrm{b}})^{\mathrm{T}}] &\approx \mathbf{G}\,\mathbf{B}^{\mathrm{t}}\mathbf{G}^{\mathrm{T}} \\
E[\,\widetilde{\mathbf{d}}^{\mathrm{a}}(\widetilde{\mathbf{d}}^{\mathrm{b}})^{\mathrm{T}}] &\approx E[\,\widetilde{\mathbf{d}}^{\mathrm{b}}(\widetilde{\mathbf{d}}^{\mathrm{b}})^{\mathrm{T}}] - \mathbf{G}\mathbf{B}^{\mathrm{t}}\mathbf{G}^{\mathrm{T}} \\
&= \mathbf{R}^{\mathrm{t}} + \mathbf{G}\mathbf{B}^{\mathrm{t}}\mathbf{G}^{\mathrm{T}} - \mathbf{G}\mathbf{B}^{\mathrm{t}}\mathbf{G}^{\mathrm{T}} \\
&= \mathbf{R}^{\mathrm{t}}
\end{aligned}$$

Covariance estimation using Desroziers diagnostics

$$
\begin{aligned}
E[(\widetilde{\mathbf{d}}^{\mathrm{b}} - \widetilde{\mathbf{d}}^{\mathrm{a}})\,(\widetilde{\mathbf{d}}^{\mathrm{b}})^{\mathrm{T}}] \;&\approx\; \mathbf{G}\,\mathbf{B}^{\mathrm{t}}\mathbf{G}^{\mathrm{T}} \\
E[(\widetilde{\mathbf{d}}^{\mathrm{a}})\,(\widetilde{\mathbf{d}}^{\mathrm{b}})^{\mathrm{T}}] \;&\approx\; \mathbf{R}^{\mathrm{t}}
\end{aligned}
$$

- The covariance matrices on the left hand side can be estimated from sample averages of standard output from data assimilation. ☺
- They can be compared with the specified covariance matrices $\mathbf{G}\,\mathbf{B}\,\mathbf{G}^{\mathrm{T}}$ and $\mathbf{R}$ to assess their consistency. ☺
- Unlike the innovation covariance matrix, they provide separate estimates of $\mathbf{B}^{\mathrm{t}}$ (in observation space) and $\mathbf{R}^{\mathrm{t}}$, and thus can also be used to recalibrate the specified covariance matrices ☺, **but** . . .
- The relationship between the left and right hand sides is somewhat incestuous since the derivation is based on the assumption that $\mathbf{B}$ and $\mathbf{R}$ are correct in the first place! ☹

Covariance estimation using the Kalman filter

- Propagate the background-error covariance matrix as in the (Extended) Kalman filter:

$$\mathbf{P}^{\mathrm{f}}(t_i) \;=\; \mathbf{M}(t_i, t_{i-1})\,\mathbf{P}^{\mathrm{a}}(t_i)\,\mathbf{M}(t_i, t_{i-1})^{\mathrm{T}} \;+\; \mathbf{Q}(t_i) \tag{1}$$

  and then set $\mathbf{B} = \mathbf{P}^{\mathrm{f}}(t_i)$ at the assimilation time.

- In 4D-Var, the covariances are propagated *implicitly* via (1) ☺, **but** only *within* the assimilation window on a given assimilation cycle. ☹
- In strong-constraint 4D-Var, $\mathbf{Q}(t_i) = 0$, whereas in weak-constraint 4D-Var, $\mathbf{Q}(t_i) \neq 0$.
- This procedure will naturally produce flow-dependent and dynamically balanced background error covariances ☺, **but** . . .
- Equation (1) is only approximate for nonlinear problems. ☹
- Direct use of equation (1) is only practical for very small problems ☹, and $\mathbf{P}^{\mathrm{a}}$ and $\mathbf{Q}$ are difficult to specify. ☹
- For large problems, reduced-rank methods can be used ☺, **but** are of questionable relevance when there are no dominant error modes. ☹

Covariance estimation using model-generated errors

- Define a model proxy for background error.
  1. Generate differences between pairs of model forecasts valid at the same time (called the NMC method in meteorology).
  2. Generate an ensemble of model forecast perturbations by perturbing input parameters to the model and data assimilation system.
- The advantage of these methods is that they provide statistical information of **all** model variables at **all** model grid points (even in data-sparse regions). ☺
- Method 1 is straightforward to implement ☺ **but** has a dubious link to actual background error. ☹
  - It produces artificially small variances in data-sparse regions.
- Method 2 produces the correct cycling mechanism of errors in the data assimilation/forecast system ☺ **but** is expensive and requires good prior knowledge of the error statistics of the input parameters to produce sensible perturbations (these can be difficult to obtain). ☹
  - Perturbed input fields include the observations, model parameters,

Covariance estimation using ensemble perturbations

- An estimate of the background error covariance matrix from a sample of $k = 1, \ldots, N_\mathrm{e}$ perturbed background states, $\mathbf{x}_k^\mathrm{b} = \mathbf{x}_0^\mathrm{b} + \boldsymbol{\epsilon}_k^\mathrm{b}$, is

$$\mathbf{B} \;=\; \frac{1}{N_\mathrm{e}-1}\sum_{k=1}^{N_\mathrm{e}} \left(\boldsymbol{\epsilon}_k^{\mathrm{b}\prime}\right)\left(\boldsymbol{\epsilon}_k^{\mathrm{b}\prime}\right)^\mathrm{T}$$

where

$$\boldsymbol{\epsilon}_k^{\mathrm{b}\prime} \;=\; \boldsymbol{\epsilon}_k^\mathrm{b} - \frac{1}{N_\mathrm{e}}\sum_{k=1}^{N_\mathrm{e}} \boldsymbol{\epsilon}_k^\mathrm{b} \;=\; \left(\left(\epsilon_k^{\mathrm{b}\prime}\right)_1, \ldots, \left(\epsilon_k^{\mathrm{b}\prime}\right)_N\right)^\mathrm{T}$$

- In compact notation

$$\mathbf{B} \;=\; \mathbf{X}'\,\mathbf{X}'^\mathrm{T}$$

where

$$\mathbf{X}' = \frac{1}{\sqrt{N_\mathrm{e}-1}}\left(\boldsymbol{\epsilon}_1^{\mathrm{b}\prime}, \ldots, \boldsymbol{\epsilon}_{N_\mathrm{e}}^{\mathrm{b}\prime}\right)$$

Covariance estimation using ensemble perturbations

$$\mathbf{B} \;=\; \mathbf{X}'\,\mathbf{X}'^{\mathrm{T}}$$

$$\mathbf{X}' = \frac{1}{\sqrt{N_{\mathrm{e}}-1}}\left(\boldsymbol{\epsilon}_1^{\mathrm{b}\,\prime}, \ldots, \boldsymbol{\epsilon}_{N_{\mathrm{e}}}^{\mathrm{b}\,\prime}\right)$$

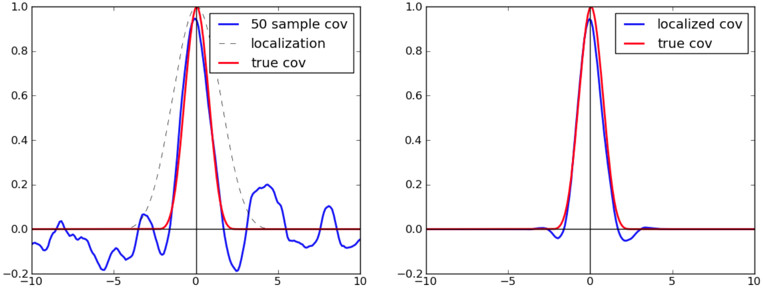
- If $\boldsymbol{\epsilon}_k^{\mathrm{b}}$ are random samples from the true background error pdf then **B** will converge to $\mathbf{B}^{\mathrm{t}}$ for large N_{e} ☺, **but** . . .
- For large problems, $N_{\mathrm{e}} \ll N$ so sampling error will be large. ☹
- In practice, $\boldsymbol{\epsilon}_k^{\mathrm{b}}$ will not be a perfect sample from the true pdf. ☹
- Covariance localization, filtering and inflation are needed to rectify these problems in practical implementations of the Ensemble Kalman filter (EnKF) or Ensemble Variational assimilation (EnVar).

Covariance localization

- Removal of spurious covariances at large separation distances can be done using a Schur product with a space-limited correlation function:

$$\mathbf{B} \;=\; \mathbf{X}'\mathbf{X}'^{\mathrm{T}} \circ \mathbf{C}_{\mathrm{loc}}$$

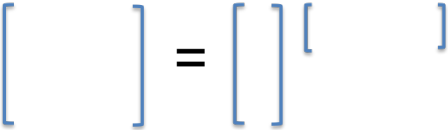
- $\mathbf{A} = \mathbf{B} \circ \mathbf{C}$ implies element-by-element multiplication: $A_{ij} = B_{ij}C_{ij}$



"*covariance localization...is the secret ingredient that makes it all possible*" Whitaker (2011), ECMWF Seminar on DA for atmosphere and ocean.

Modeling $\mathbf{B}$ in practice

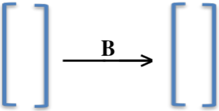
- With reduced-rank methods, $\mathbf{B}$ is formulated as $\mathbf{B} = \mathbf{X}\mathbf{X}^{\mathrm{T}}$ where $\mathbf{X}$ is an $N \times L$ matrix with $L \ll N$.



- The resulting $\mathbf{B}$ matrix has rank at most equal to L.
- The L columns of $\mathbf{X}$ can be EOFs, singular vectors, ensemble perturbations, or some other basis functions.
- Reduced-rank methods are commonly used with Kalman filters since the KF equations require direct matrix-matrix multiplications.

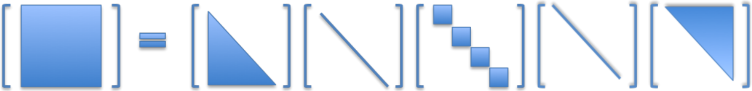
Specifying **B** in practice

- An alternative approach is to specify **B** as a covariance operator: $\mathbf{B}\,\boldsymbol{\alpha} \rightarrow \delta\mathbf{x}$.



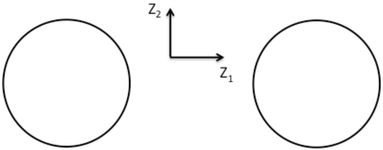
- Multiplication of the different components of the **B** matrix produces a full-rank matrix in model state space:

$$\mathbf{B} = \widehat{\mathbf{K}}\,\widehat{\boldsymbol{\Sigma}}\,\widehat{\mathbf{C}}\,\widehat{\boldsymbol{\Sigma}}\,\widehat{\mathbf{K}}^{\mathrm{T}}$$

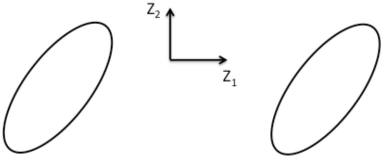


Illustrations in $\mathbb{R}^2$

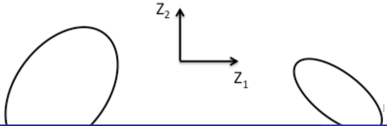
1. Homogeneous and isotropic



2. Homogeneous and **anisotropic**



3. Inhomogeneous and **anisotropic**



Outline

1. General concepts for Data Assimilation
2. Classical algorithms for Data Assimilation
3. Variational algorithms - Minimization algorithms
4. Error covariance matrices estimation and modeling
5. More on Data Assimilation

References on variational Data Assimilation

- R. Daley, Atmospheric data analysis, Cambridge University Press, 1991.
- Advanced Data Assimilation for Geosciences, Lecture Notes of the Les Houches School of Physics: Special Issue, June 2012, edited by Eric Blayo, Marc Bocquet, Emmanuel Cosme, and Leticia F. Cugliandolo, 2015
- Albert Tarantola, Inverse Problem Theory and Methods for Model Parameter Estimation, SIAM, 2005
- ECMWF Technical Memoranda
- Bouttier, F. and P. Courtier, 1999: Data assimilation concepts and methods, ECMWF Training Course Notes. Available on line.
- Kalnay, E., 2003: Atmospheric Modeling, Data Assimilation and Predictability, Cambridge Uni. Press.
- Gaspari, G. and S. E. Cohn, 1999: Construction of correlation functions in two and three dimensions. Q. J. R. Meteorol. Soc., 125, 723757.

Data Assimilation Training courses

- Data Assimilation Training course @CERFACS (Toulouse June 2017 - Contact Selime Gurol gurol@cerfacs.fr + Anthony Weaver weaver@cerfacs.fr - http://cerfacs.fr/en/event/?category=training)
- Data Assimilation Training course @Grenoble (Dec 2016- Jan.2017 - Contact Emmanuel Cosme emmanuel.cosme@ujf-grenoble.fr)
- Data Assimilation Training Course @ECMWF (March 2017 - http://www.ecmwf.int/en/learning/training)
- Data Assimilation @LesHouches (2017 ?)
- Data Assimilation and Uncertainty Quantification @CERFACS (May 2017 - Contact Sophie Ricci ricci@cerfacs.fr + Mlanie Rochoux rochoux@cerfacs.fr - http://cerfacs.fr/en/event/?category=training)