Introduction to Data Assimilation

Maëlle Nodet

Université de Grenoble, INRIA, LJK

Journée “Inversion et assimilation d’images”
16 juin 2011
What is data assimilation?

Combine at best different sources of information to estimate the state of a system:

- model equations
- observations, data
- background, a priori information
- statistics
What is data assimilation?

You use a kind of data assimilation scheme if you sneeze whilst driving along the motorway.

As your eyes close involuntary, you retain in your mind a picture of the road ahead and of the traffic nearby [background], as well as a mental ‘model’ of how the car will behave in the short time [model equations] before you re-open your eyes and make a course correction [adjustment to observations].

What is data assimilation for?

Historically: meteorology. Later, oceanography.

Today, many other fields

- glaciology,
- seismology,
- nuclear fusion,
- medicine,
- agronomy,
- etc.
What is data assimilation for?

Historically: initial state estimation, for weather forecasting.

Today, many other applications:

- initial conditions for predictions,
- calibration and validation,
- observing system design, monitoring and assessment,
- reanalysis,
- better understanding (model errors, data errors, physical process interactions, parameters, etc),
- etc.
Subjective analysis (19th century)

(From Daley, 1991, after Loomis, 1885)
Richardson’s numerical weather prediction (1922)

(From Daley, 1991, after Richardson, 1922)
Cressman’s objective analysis (1950’s)

Correction at the grid point $j$ with an observation at $i$:

$$x_j^a = x_j^b + \sum_{i=1}^{s} \frac{w(i,j)(y_i - x_i^b)}{\sum_{i=1}^{s} w(i,j)}$$

with

$$w(i,j) = \begin{cases} R^2 - r(i,j)^2 & \text{if } r(i,j) \leq R \\ R^2 + r(i,j)^2 & \text{if } r(i,j) > R \end{cases}$$

(From Bouttier and Courtier, 2002)
Nudging (1970’s)

If the model writes:

\[
\frac{dx}{dt} = M(x)
\]

then the nudging equation is:

\[
\frac{dx}{dt} = M(x) + \alpha(y - x)
\]

where \( y \) is a direct observation of \( x \).
After the 1970’s

Toy least squares problem

Two different available measurements for a same quantity. Which estimation of the true value?

Example: 2 obs $y_1 = 1$ and $y_2 = 2$ of some unknown quantity $x$
Toy least squares problem

Two different available measurements for a same quantity. Which estimation of the true value?

**Example** : 2 obs $y_1 = 1$ and $y_2 = 2$ of some unknown quantity $x$

$$\text{Min} \ (x - 1)^2 + (x - 2)^2 \quad \longrightarrow \hat{x} = \frac{3}{2}$$
Toy least squares problem

Two different available measurements for a same quantity. Which estimation of the true value?

Example: 2 obs $y_1 = 1$ and $y_2 = 2$ of some unknown quantity $x$

$$\text{Min} \ (x - 1)^2 + (x - 2)^2 \quad \rightarrow \hat{x} = \frac{3}{2}$$

Problems:

- Sensitivity to any change of unit:
  1 obs $y_1 = 1$ of $x$, and 1 obs $y_2 = 4$ of $2x$

$$\text{Min} \ (x - 1)^2 + (2x - 4)^2 \quad \rightarrow \hat{x} = \frac{9}{5}$$
Toy least squares problem

Two different available measurements for a same quantity. Which estimation of the true value ?

**Example** : 2 obs $y_1 = 1$ and $y_2 = 2$ of some unknown quantity $x$

$$\text{Min } (x - 1)^2 + (x - 2)^2 \quad \rightarrow \hat{x} = \frac{3}{2}$$

**Problems** :

- Sensitivity to any change of unit:
  1 obs $y_1 = 1$ of $x$, and 1 obs $y_2 = 4$ of $2x$

  $$\text{Min } (x - 1)^2 + (2x - 4)^2 \quad \rightarrow \hat{x} = \frac{9}{5}$$

- No sensitivity to the accuracy of the measurement:
  same estimate even if $y_1$ is more accurate than $y_2$
Reformulation in a statistical framework

\( y_i \) is a realization of some random variable \( Y_i \).

We define: \( Y_i = x + e_i \) with

**Hypotheses:**

- \( E(e_i) = 0 \) \((i = 1, 2)\) unbiased measurements
- \( \text{Var}(e_i) = \sigma_i^2 \) \((i = 1, 2)\) the accuracy is known
- \( \text{Cov}(e_1, e_2) = 0 \) i.e. \( E(e_1 e_2) = 0 \) errors are independent

We seek out an estimator (i.e. a random variable) \( \hat{X} \) which is

- **linear**: \( \hat{X} = \alpha_1 Y_1 + \alpha_2 Y_2 \) (to be simple)
- **unbiased**: \( E(\hat{X}) = x \) (natural)
- **of minimal variance**: \( \text{Var}(\hat{X}) \) minimal (optimal accuracy)

\( \rightarrow \quad \text{BLUE}: \) Best Linear Unbiased Estimator
Best linear unbiased estimator

Best Linear Unbiased Estimator is:

\[ \hat{X} = \frac{1}{\sigma_1^2} y_1 + \frac{1}{\sigma_2^2} y_2 \]

\[ \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} \]
Best linear unbiased estimator

Best Linear Unbiased Estimator is:

\[ \hat{X} = \frac{1}{\sigma_1^2} y_1 + \frac{1}{\sigma_2^2} y_2 \]

Equivalent variational formulation

\[ \hat{X} \text{ is the minimizer of } J(x) = \frac{1}{2} \left[ \frac{(x - y_1)^2}{\sigma_1^2} + \frac{(x - y_2)^2}{\sigma_2^2} \right] \]
Best linear unbiased estimator

Best Linear Unbiased Estimator is:

\[
\hat{X} = \frac{1}{\sigma_1^2} y_1 + \frac{1}{\sigma_2^2} y_2
\]

Equivalent variational formulation

\[
\hat{X} \text{ is the minimizer of } J(x) = \frac{1}{2} \left[ \frac{(x - y_1)^2}{\sigma_1^2} + \frac{(x - y_2)^2}{\sigma_2^2} \right]
\]

Remarks:

- This gives a rationale for the choice of the norm in \( J \).
Best linear unbiased estimator

Best Linear Unbiased Estimator is:

\[ \hat{X} = \frac{1}{\sigma_1^2} y_1 + \frac{1}{\sigma_2^2} y_2 \]

Equivalent variational formulation

\[ \hat{X} \] is the minimizer of

\[ \mathcal{J}(x) = \frac{1}{2} \left[ \frac{(x - y_1)^2}{\sigma_1^2} + \frac{(x - y_2)^2}{\sigma_2^2} \right] \]

Remarks:

- This gives a rationale for the choice of the norm in \( \mathcal{J} \).
- This solves the problem of sensitivity to the units and non-sensitivity to the accuracies.
**Best linear unbiased estimator**

Best Linear Unbiased Estimator is:

$$\hat{X} = \frac{1}{\sigma_1^2} y_1 + \frac{1}{\sigma_2^2} y_2$$

**Equivalent variational formulation**

$$\hat{X}$$ is the minimizer of

$$J(x) = \frac{1}{2} \left[ \frac{(x - y_1)^2}{\sigma_1^2} + \frac{(x - y_2)^2}{\sigma_2^2} \right]$$

**Remarks:**

- This gives a rationale for the choice of the norm in $J$.
- This solves the problem of sensitivity to the units and non-sensitivity to the accuracies.

- **Accuracy:**

$$Var(\hat{X}) = \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2} \Rightarrow \frac{1}{Var(\hat{X})} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} = J''(x)$$
If one considers that $Y_1$ is a prior (or background) estimate $X_b$ of $x$, and $Y_2 = Y$ is an independent observation, then:

$$J(x) = \frac{1}{2} \left[ \frac{(x - x_b)^2}{\sigma_b^2} + \frac{(x - y)^2}{\sigma_o^2} \right]$$

and

$$\hat{X} = X_b + \frac{\sigma_b^2}{\sigma_b^2 + \sigma_o^2} \underbrace{(Y - X_b)}_{\text{gain}} \underbrace{(Y - X_b)}_{\text{innovation}}$$
Data assimilation methods

Two types of methods:

1. Direct computation of the BLUE.
   Main algorithm: Kalman filter
   → stochastic data assimilation, section 1.

   Main algorithm: 4D-Var
   → variational data assimilation, section 2.
Outline

1. Stochastic data assimilation
   - Notations and vocabulary
   - Best linear unbiased estimator
   - Kalman filter algorithm

2. Variational Data Assimilation
   - Principle of variational methods
   - Gradient-based optimization
   - Variational algorithms

3. Implementation issues
   - Non linearities
   - High dimensional problems
   - Practical adjoint coding
Where am I?

1. Stochastic data assimilation
   - Notations and vocabulary
   - Best linear unbiased estimator
   - Kalman filter algorithm

2. Variational Data Assimilation
   - Principle of variational methods
   - Gradient-based optimization
   - Variational algorithms

3. Implementation issues
   - Non linearities
   - High dimensional problems
   - Practical adjoint coding
Where am I?

1. Stochastic data assimilation
   - Notations and vocabulary
   - Best linear unbiased estimator
   - Kalman filter algorithm

2. Variational Data Assimilation
   - Principle of variational methods
   - Gradient-based optimization
   - Variational algorithms

3. Implementation issues
   - Non linearities
   - High dimensional problems
   - Practical adjoint coding
Errors statistics

Mean:

\[ E(x) = \langle x \rangle \text{ scalar}, \quad E(x) = (E(x_1), E(x_2), \ldots, E(x_n)) \text{ vector-valued} \]

Variance, covariance (\( x, y \) scalar):

\[ \text{Var}(x) = E((x - E(x))^2), \quad \text{Cov}(x, y) = E((x - E(x))(y - E(y))) \]

We say that errors are:

- unbiased if \( E(\epsilon) = 0 \);
- uncorrelated if \( E(\epsilon_1 \epsilon_2^T) = 0 \);
- white in time if \( E(\epsilon_t \epsilon_t^T) = 0 \).
Covariance matrix (\( \mathbf{x} \) vector-valued):

\[
\text{Cov}(\mathbf{x}) = \mathbb{E}((\mathbf{x} - \mathbb{E}(\mathbf{x}))(\mathbf{x} - \mathbb{E}(\mathbf{x}))^T)
\]

\[
(C\text{ov}(\mathbf{x}))_{i,j} = \text{Cov}(x_i, x_j) = \mathbb{E}((x_i - \mathbb{E}(x_i))(x_j - \mathbb{E}(x_j)))
\]

E.g. for \( \mathbf{x} = (x_1, x_2, x_3) \):

\[
\text{Cov}(\mathbf{x}) = \begin{pmatrix}
\text{Var}(x_1) & \text{Cov}(x_1, x_2) & \text{Cov}(x_1, x_3) \\
\text{Cov}(x_1, x_2) & \text{Var}(x_2) & \text{Cov}(x_2, x_3) \\
\text{Cov}(x_1, x_3) & \text{Cov}(x_2, x_3) & \text{Var}(x_3)
\end{pmatrix}
\]
Notations

State

- \( \mathbf{x} \) state vector
- \( \mathbf{x}^t \) true state (unknown)
- \( \mathbf{x}^b \) background state (a priori information), background error \( \epsilon^b = \mathbf{x}^b - \mathbf{x}^t \), covariance matrix \( \mathbf{B} \)
- \( \mathbf{x}^a \) analyzed state (result of the assimilation process), analysis error \( \epsilon^a = \mathbf{x}^a - \mathbf{x}^t \), covariance matrix \( \mathbf{A} \)

Observations

- observation vector \( \mathbf{y}^o \)
- observation operator \( \mathbf{H} \), mapping state space to observation space: \( \mathbf{y}^o = \mathbf{H}(\mathbf{x}^t) + \epsilon^o \)
- observation error \( \epsilon^o \), covariance matrix \( \mathbf{R} \)
Where am I?

1 Stochastic data assimilation
   - Notations and vocabulary
   - Best linear unbiased estimator
   - Kalman filter algorithm

2 Variational Data Assimilation
   - Principle of variational methods
   - Gradient-based optimization
   - Variational algorithms

3 Implementation issues
   - Non linearities
   - High dimensional problems
   - Practical adjoint coding
Problem position: what we have

We are given:
- a background estimate $x^b$, whose error $\epsilon^b$ are assumed unbiased and non trivial, with covariance matrix $B$ given,
- partial observations $y^o = H(x^t) + \epsilon^o$, where $\epsilon^o$ are unbiased and non trivial, with covariance matrix $R$ given.

We also assume that:
- for any $x$ close enough to $x^b$, $H(x) - H(x^b) = H(x - x^b)$ where $H$ is a linear operator,
- $\epsilon^o$ and $\epsilon^b$ are not correlated.
Problem position: what we look for

We aim at producing an estimate $\mathbf{x}^a$ of the true state $\mathbf{x}^t$ of the system.

The best estimate is searched for as a linear combination of the background estimate and the observation:

$$\mathbf{x}^a = K_1 \mathbf{x}^b + K_2 \mathbf{y}^o$$

Optimality criterium

We look for an unbiased estimate $\mathbf{x}^a$, with minimal variance $\text{tr}(A)$. 
Best linear unbiased estimator, or least squares analysis

1. BLUE analysis:

\[
\begin{align*}
{x^a} &= {x^b} + K{(y - H({x^b}))} \\
K &= BH^T(HBH^T + R)^{-1}
\end{align*}
\]

\(K\): gain, or weight matrix, \(y - H({x^b})\) innovation.
Best linear unbiased estimator, or least squares analysis

1. **BLUE analysis:**

\[
\begin{cases}
  x^a = x^b + K(y - \mathcal{H}(x^b)) \\
  K = BH^T(\mathbf{H}B^2 + R)^{-1}
\end{cases}
\]

- **K**: gain, or weight matrix, \(y - \mathcal{H}(x^b)\) innovation.

2. **Analysis covariance matrix:**

\[A = (I - KH)B\]
Best linear unbiased estimator, or least squares analysis

1. **BLUE analysis:**

\[
\begin{align*}
    x^a &= x^b + K(y - \mathcal{H}(x^b)) \\
    K &= BH^T( HBH^T + R )^{-1}
\end{align*}
\]

- **K:** gain, or weight matrix, \( y - \mathcal{H}(x^b) \) innovation.

2. **Analysis covariance matrix:** 

\[ A = (I-KH)B \]

3. **Equivalent variational optimization problem:** (optimal least squares)

\[
\begin{align*}
    x^a &= \text{arg min } \mathcal{J} \\
    \mathcal{J}(x) &= (x - x^b)^T B^{-1} (x - x^b) + (y - \mathcal{H}(x))^T R^{-1} (y - \mathcal{H}(x))
\end{align*}
\]

- **\( \mathcal{J} \):** cost function.
Data assimilation methods

Two types of methods:

1. Direct computation of the BLUE, and the gain matrix $K$. Main algorithm: Kalman filter
   $\rightarrow$ stochastic data assimilation, this section.

   $\rightarrow$ variational data assimilation, section 2.
Where am I?

1. Stochastic data assimilation
   - Notations and vocabulary
   - Best linear unbiased estimator
   - Kalman filter algorithm

2. Variational Data Assimilation
   - Principle of variational methods
   - Gradient-based optimization
   - Variational algorithms

3. Implementation issues
   - Non linearities
   - High dimensional problems
   - Practical adjoint coding
The Kalman filter sequence

The figure illustrates the Kalman filter sequence in time. The state is shown in three steps:

1. At time $k-2$, the predicted state is $x^f$.
2. At time $k-1$, the predicted state is updated to $y^o$ using the measurement $y^o$.
3. At time $k$, the updated state is $y^o$.

The diagram shows the progression of the state and measurements over time.
The Kalman filter sequence
The Kalman filter sequence

\[ x^o \]

\[ y^o \]

\[ x^a \]

\[ y^o \]

state

\[ x^f \]

\[ k - 2 \]

\[ k - 1 \]

\[ k \]
The Kalman filter sequence

\[ \begin{align*}
\text{state} & \\
\text{Observations} & = y^o \\
\text{Forecast} & = x^f \\
\text{Analysis} & = x^a
\end{align*} \]
The Kalman filter sequence

The Kalman filter algorithm

Stochastic data assimilation
The Kalman filter sequence
Back to notations

Vectors

- $k$: time index
- $x^f_k$: forecast state (background), forecast error covariance matrix $P^f_k$
- $x^a_k$: analyzed state (result of the assimilation process), analysis error covariance matrix $P^a_k$

Operators

- model operator $x^t_{k+1} = M_{k,k+1}(x^t_k) + \eta_{k,k+1}$, model error $\eta_{k,k+1}$, covariance matrix $Q_k$
- observation operator $y^o_k = H_k(x^t) + \epsilon^o_k$, observation error $\epsilon^o_k$, covariance matrix $R_k$
Kalman’s hypotheses

Schematically:

- Model and observations operators are linear, denoted $M_{k,k+1}$ and $H_k$;
- Errors are unbiased, gaussian, independant and white in time.
Kalman’s hypotheses

Uncensored version:

- Initial state is gaussian: $x_0^t \sim \mathcal{N}(x_0^b, P_0^b)$;
Kalman’s hypotheses

Uncensored version:

- Initial state is gaussian: \( x_0^t \sim \mathcal{N}(x_0^b, P_0^b) \);
- The dynamical model \( M_k \) is linear and denoted \( M_{k,k+1} \);
- The model errors are unbiased and gaussian: \( \eta_k \sim \mathcal{N}(0, Q_k) \);
- The model errors are white in time: \( \langle \eta_k \eta_j^T \rangle = 0 \) if \( k \neq j \);
Kalman’s hypotheses

Uncensored version:

- Initial state is gaussian: \( x_t^0 \sim \mathcal{N}(x_0^b, P_0^b) \);
- The dynamical model \( \mathcal{M}_k \) is linear and denoted \( \mathcal{M}_{k,k+1} \);
- The model errors are unbiased and gaussian: \( \eta_k \sim \mathcal{N}(0, Q_k) \);
- The model errors are white in time: \( \langle \eta_k \eta_j^T \rangle = 0 \) if \( k \neq j \);
- The observation operators \( \mathcal{H}_k \) are linear and denoted \( \mathcal{H}_k \);
- The observation errors are unbiased and gaussian: \( \epsilon_k^o \sim \mathcal{N}(0, R_k) \);
- The observation errors are white in time: \( \langle \epsilon_k^o \epsilon_j^o T \rangle = 0 \) if \( k \neq j \);
Kalman’s hypotheses

Uncensored version:

- Initial state is gaussian: \( \mathbf{x}_0^t \sim \mathcal{N}(\mathbf{x}_0^b, \mathbf{P}_0^b) \);
- The dynamical model \( \mathcal{M}_k \) is linear and denoted \( \mathbf{M}_{k,k+1} \);
- The model errors are unbiased and gaussian: \( \eta_k \sim \mathcal{N}(0, \mathbf{Q}_k) \);
- The model errors are white in time: \( \langle \eta_k \eta_j^T \rangle = 0 \) if \( k \neq j \);
- The observation operators \( \mathcal{H}_k \) are linear and denoted \( \mathbf{H}_k \);
- The observation errors are unbiased and gaussian: \( \epsilon_k^o \sim \mathcal{N}(0, \mathbf{R}_k) \);
- The observation errors are white in time: \( \langle \epsilon_k^o \epsilon_j^o^T \rangle = 0 \) if \( k \neq j \);
- Errors of different types are independent: \( \langle \eta_k \epsilon_j^o^T \rangle = 0, \langle \eta_k \epsilon_0^b^T \rangle = 0, \langle \epsilon_k^o \epsilon_0^b^T \rangle = 0 \).
The Kalman filter equations

1 **Initialization:**
   \( x^f_0 \) and \( P^f_0 \) are given, e.g. equal to \( x^b \) and \( B \)

2 **Analysis step:**
   
   \[
   \begin{align*}
   K_k &= (H_k P^f_k)^T [H_k (H_k P^f_k)^T + R_k]^{-1}, \\
   x^a_k &= x^f_k + K_k (y^o_k - H_k x^f_k), \\
   P^a_k &= (I - K_k H_k) P^f_k.
   \end{align*}
   \]

3 **Forecast step:**
   
   \[
   \begin{align*}
   x^f_{k+1} &= M_{k,k+1} x^a_k, \\
   P^f_{k+1} &= M_{k,k+1} P^a_k M_{k,k+1}^T + Q_k.
   \end{align*}
   \]
Where am I?

1. Stochastic data assimilation
   - Notations and vocabulary
   - Best linear unbiased estimator
   - Kalman filter algorithm

2. Variational Data Assimilation
   - Principle of variational methods
   - Gradient-based optimization
   - Variational algorithms

3. Implementation issues
   - Non linearities
   - High dimensional problems
   - Practical adjoint coding
Where am I?

1. Stochastic data assimilation
   - Notations and vocabulary
   - Best linear unbiased estimator
   - Kalman filter algorithm

2. Variational Data Assimilation
   - Principle of variational methods
   - Gradient-based optimization
   - Variational algorithms

3. Implementation issues
   - Non linearities
   - High dimensional problems
   - Practical adjoint coding
Back to notations

Vectors

- $x$ state vector
- $x^b$ background state (a priori information), background error $\epsilon^b = x^b - x^t$, covariance matrix $B$
- $x^a$ analyzed state (result of the assimilation process)
- $y$ observation vector

Operators

- model operator $x^t_{k+1} = M_{k,k+1}(x^t_k) + \eta_{k,k+1}$
- observation operator $y^o = H(x^t) + \epsilon^o$, $y^o_k = H_k(x^t) + \epsilon^o_k$, observation error $\epsilon^o_k$, covariance matrix $R_k$
Back to the BLUE

Variational approach of BLUE consists in finding \( x^a = \arg \min J \):

\[
J(x) = (x-x^b)^T B^{-1} (x-x^b) + (y-H(x))^T R^{-1} (y-H(x))
\]

\[
= \frac{1}{2} \| x - x^b \|_B^2 + \frac{1}{2} \| H(x) - y \|_R^2
\]

\[
J^b + J^o
\]
Back to the BLUE

Variational approach of BLUE consists in finding $x^a = \arg \min J$:

$$J(x) = (x-x^b)^T B^{-1}(x-x^b) + (y-H(x))^T R^{-1}(y-H(x))$$

$$= \frac{1}{2} \|x - x^b\|_B^2 + \frac{1}{2} \|H(x) - y\|_R^2$$

$$J^b \quad J^o$$

If the problem is time-dependent:

$$J(x) = \frac{1}{2} \|x - x_b\|_B^2 + \frac{1}{2} \sum_k \|H_k(x_k) - y_k\|_{R_k}^2$$

$$= \frac{1}{2} \|x - x_b\|_B^2 + \frac{1}{2} \sum_k \|H_k(M_{0\rightarrow k}(x)) - y_k\|_{R_k}^2$$

$$J^b \quad J^o$$
Variational methods principle
Fundamental remark

Once $J$ is defined (i.e. once all the ingredients are chosen: control variables, error statistics, norms, observations...), the problem is entirely defined. Hence its solution.

$\Rightarrow$ The “physical part” of data assimilation lies in the definition of $J$.

The rest of the job, i.e. minimizing $J$, is “technical” work.
Where am I?

1. Stochastic data assimilation
   - Notations and vocabulary
   - Best linear unbiased estimator
   - Kalman filter algorithm

2. Variational Data Assimilation
   - Principle of variational methods
   - Gradient-based optimization
   - Variational algorithms

3. Implementation issues
   - Non linearities
   - High dimensional problems
   - Practical adjoint coding
Descent methods to minimize a function require knowledge of (an estimate of) its gradient.

\[ x_{k+1} = x_k + \alpha_k d_k \]

(k iteration number)

with \( d_k = \begin{cases} -\nabla J(x_k) & \text{gradient method} \\ -[\text{Hess}(J)(x_k)]^{-1} \nabla J(x_k) & \text{(quasi-)Newton method} \\ -\nabla J(x_k) + \frac{\|\nabla J(x_k)\|^2}{\|\nabla J(x_{k-1})\|^2} d_{k-1} & \text{conjugate gradient} \\ \ldots \end{cases} \)
Variational Data Assimilation
Gradient-based optimization

Gradient computation

The computation of \( \nabla J(x_k) \) may be difficult if the dependency of \( J \) with regard to the control variable \( x \) is not direct.

Example:

- \( u(x) \) solution of a differential equation
- \( K \) a coefficient of this equation
- \( u^{obs}(x) \) an observation of \( u(x) \)
- \( J(K) = \frac{1}{2} \| u(x) - u^{obs}(x) \|^2 \)
Gradient computation

The computation of $\nabla J(x_k)$ may be difficult if the dependency of $J$ with regard to the control variable $x$ is not direct.

Example:

- $u(x)$ solution of a differential equation
- $K$ a coefficient of this equation
- $u^{\text{obs}}(x)$ an observation of $u(x)$
- $J(K) = \frac{1}{2} \|u(x) - u^{\text{obs}}(x)\|^2$

$$\nabla J[K](k) = \langle \hat{u}, u - u^{\text{obs}} \rangle \quad \text{with} \quad \hat{u} = \lim_{\alpha \to 0} \frac{u_{K+\alpha k} - u_K}{\alpha}$$
It is often difficult (or even impossible) to obtain the gradient through the computation of growth rates.

**Example:**

\[
\begin{align*}
\frac{dx(t)}{dt} &= M(x(t)) \quad t \in [0, T] \\
x(t = 0) &= u
\end{align*}
\]

with \( u = \begin{pmatrix} u_1 \\ \vdots \\ u_N \end{pmatrix} \)

\[
J(u) = \frac{1}{2} \int_0^T \| x(t) - x^{obs}(t) \|^2 dt
\]

\[
\nabla J(u) = \begin{pmatrix}
\frac{\partial J}{\partial u_1}(u) \\
\vdots \\
\frac{\partial J}{\partial u_N}(u)
\end{pmatrix} \approx \begin{pmatrix}
[ J(u + \alpha e_1) - J(u) ] / \alpha \\
\vdots \\
[ J(u + \alpha e_N) - J(u) ] / \alpha
\end{pmatrix}
\]

\[\rightarrow N + 1\] model runs
Gradient computation

In actual applications like meteorology / oceanography,
\[ N = \mathbf{u} = \mathcal{O}(10^6 - 10^8) \rightarrow \text{this method cannot be used.} \]

**Adjoint method**

The adjoint method provides a very efficient way to compute \( \nabla \mathcal{J} \), with only one run of the **adjoint model** (computationally 4-10 times the cost of the direct model).
Gradient computation

In actual applications like meteorology / oceanography, $N = [u] = \mathcal{O}(10^6 - 10^8)$ → this method cannot be used.

**Adjoint method**

The adjoint method provides a very efficient way to compute $\nabla J$, with only one run of the **adjoint model** (computationally 4-10 times the cost of the direct model).

**Attention!**

On the contrary, do not forget that, if the size of the control variable is very small ($< 10$), $\nabla J$ can be easily estimated by the computation of growth rates.
A reminder on adjoints

**Adjoint operator**

Let $\mathcal{X}$ and $\mathcal{Y}$ two prehilbertian spaces (i.e. vector spaces with scalar products). Let $A : \mathcal{X} \rightarrow \mathcal{Y}$ an operator. The adjoint operator, $A^* : \mathcal{Y} \rightarrow \mathcal{X}$, is defined by:

$$\forall x \in \mathcal{X}, \forall y \in \mathcal{Y}, \quad < Ax, y >_{\mathcal{Y}} = < x, A^* y >_{\mathcal{X}}$$

In the case where $\mathcal{X}$ and $\mathcal{Y}$ are Hilbert spaces and $A$ is linear, then $A^*$ always exists (and is unique).
A reminder on adjoints

Adjoint operator

Let $\mathcal{X}$ and $\mathcal{Y}$ two prehilbertian spaces (i.e. vector spaces with scalar products). Let $A : \mathcal{X} \rightarrow \mathcal{Y}$ an operator. The adjoint operator, $A^* : \mathcal{Y} \rightarrow \mathcal{X}$, is defined by:

$$\forall x \in \mathcal{X}, \forall y \in \mathcal{Y}, \quad <Ax, y>_{\mathcal{Y}} = <x, A^*y>_{\mathcal{X}}$$

In the case where $\mathcal{X}$ and $\mathcal{Y}$ are Hilbert spaces and $A$ is linear, then $A^*$ always exists (and is unique).

Adjoint operator in finite dimension

$A : \mathbb{R}^n \rightarrow \mathbb{R}^m$ a linear operator (i.e. a matrix). Then its adjoint operator $A^*$ (w.r. to Euclidian norms) is $A^T$. 

Where am I?

1. Stochastic data assimilation
   - Notations and vocabulary
   - Best linear unbiased estimator
   - Kalman filter algorithm

2. Variational Data Assimilation
   - Principle of variational methods
   - Gradient-based optimization
   - Variational algorithms

3. Implementation issues
   - Non linearities
   - High dimensional problems
   - Practical adjoint coding
Time-independant problems: 3D-Var

Cost function:

\[ J(x_0) = (x_0 - x^b)^T B^{-1}(x_0 - x^b) + (y - H(x_0))^T R^{-1}(y - H(x_0)) \]

Gradient:

\[ \nabla J(x) = 2B^{-1}(x - x^b) - 2H^T R^{-1}(y - H(x)) \]

Iterative minimization algorithm

- Initialisation: \( x_0 = x^b, \ n = 0 \)
- While \( \| \nabla J \| > \varepsilon \) or \( n \leq n_{\text{max}} \), do:

1. Compute \( J \)
2. Compute \( \nabla J \)
3. Descent and update of \( x_0 \)
4. \( n = n + 1 \)
Time-dependant problems: 4D-Var

Cost function:

\[ J(x_0) = \|x_0 - x_0^b\|_B^2 + \sum_{i=0}^{n} \|y_i^o - H_i(M_i(M_{i-1}(\ldots M_1(x_0))))\|_R^2 \]

Gradient more complicated, involves the adjoint model:

\[ -\frac{1}{2} \nabla J^o(x) = \sum_{i=0}^{n} M_1^T \ldots M_{i-1}^T M_i^T H_i^T R_i^{-1} d_i \]

Innovation vector \( d_i \):

\[ d_i = y_i^o - H_i(M_i(M_{i-1}(\ldots M_1(x_0)))) \]
4D-Var algorithm

- Initialization : $x = x^0, \ n = 0$
- While $\|\nabla J\| > \varepsilon$ or $n \leq n_{\text{max}}$, do :

  1. Compute $J$ thanks to the direct model $M$ and the observation operator $H$
  2. Store innovation vectors $d_i$
  3. Compute $\nabla J$ thanks to the backward integration of the adjoint model $M^T$ and the adjoint of the observation operator $H^T$
  4. Descent and update of $x$
  5. $n = n + 1$
Equivalence 4D-Var – Kalman filter

Under the Kalman Filter hypotheses, 4D-Var and Kalman filter algorithms are equivalent.

More precisely: starting with the same background, with equal covariance matrices at the beginning of the time-window, and assimilating the same observations, the same result is reached at the end of the time-window.

These algorithms are both optimal in a least squares and minimal variance point of view.
Where am I?

1. **Stochastic data assimilation**
   - Notations and vocabulary
   - Best linear unbiased estimator
   - Kalman filter algorithm

2. **Variational Data Assimilation**
   - Principle of variational methods
   - Gradient-based optimization
   - Variational algorithms

3. **Implementation issues**
   - Non linearities
   - High dimensional problems
   - Practical adjoint coding
Where am I?

1. Stochastic data assimilation
   - Notations and vocabulary
   - Best linear unbiased estimator
   - Kalman filter algorithm

2. Variational Data Assimilation
   - Principle of variational methods
   - Gradient-based optimization
   - Variational algorithms

3. Implementation issues
   - Non linearities
   - High dimensional problems
   - Practical adjoint coding
Kalman Filter and 4D-Var are valid (and produce equivalent results) if model operator $\mathbf{M}$ and observation operator $\mathbf{H}$ are linear.
Validity of BLUE analysis

Kalman Filter and 4D-Var are valid (and produce equivalent results) if model operator $M$ and observation operator $H$ are linear.

In presence of nonlinearities:

1. KF and 4DV not equivalent anymore,
2. optimality of analysis is lost.
Tangent linear hypothesis

In case of weak non-linearities: we can hope the linear analysis still gives some information...
Tangent linear hypothesis

In case of weak non linearities: we can hope the linear analysis still gives some information...

\[
\begin{align*}
\mathcal{M}_{0\rightarrow i}(x_0) - \mathcal{M}_{0\rightarrow i}(x^b_0) & \simeq M_{0\rightarrow i}(x_0 - x^b) \\
\mathcal{H}_i(x_i) - \mathcal{H}_i(x^b_i) & \simeq H_i(x_i - x^b_i)
\end{align*}
\]
In case of weak non linearities: we can hope the linear analysis still gives some information...

\[ M_{0\rightarrow i}(x_0) - M_{0\rightarrow i}(x_0^b) \approx M_{0\rightarrow i}(x_0 - x^b) \]
\[ H_i(x_i) - H_i(x_i^b) \approx H_i(x_i - x_i^b) \]

- Stochastic $\rightarrow$ extended Kalman Filter
- Variational $\rightarrow$ incremental 3D- and 4D-Var
Extended Kalman Filter

Non linear operators $\mathcal{M}$ and $\mathcal{H}$ are replaced with their tangent operators $\mathcal{M}$ and $\mathcal{H}$ at some points in the algo (innovation and state forecast):

1. **Initialization:**
   $x_0^f$ and $P_0^f$ are given, e.g. equal to $x^b$ and $B$

2. **Analysis step:**
   \[
   K_k = (H_k P_k^f)^T [H_k (H_k P_k^f)^T + R_k]^{-1},
   \]
   \[
   x_k^a = x_k^f + K_k (y_o^k - H_k x_k^f),
   \]
   \[
   P_k^a = (I - K_k H_k) P_k^f.
   \]

3. **Forecast step:**
   \[
   x_{k+1}^f = M_{k,k+1} x_k^a,
   \]
   \[
   P_{k+1}^f = M_{k,k+1} P_k^a M_{k,k+1}^T + Q_k.
   \]
Incremental 4D-Var

Increment: \( \delta x_0 = x_0 - x_0^b \)

Innovation vector \( d_i \):

\[
d_i = y_i^o - H_i(M_i(M_{i-1}(\ldots M_1(x_0^b))))
\]

Incremental cost function (quadratic):

\[
J(\delta x_0) = \|\delta x_0\|_B^2 + \sum_{i=0}^{n} \|d_i - H_i(M_i(M_{i-1}(\ldots M_1(\delta x_0))))\|_R^2
\]
Incremental 4D-Var

- Initialization: \( x_0^r = x_0^b \)

**START OUTER LOOP**

- Non linear model integration: \( x_i^r = M_{0,i}[x_i^r] \)
- Innovation vector computation: \( d_i = y_i^o - H_i(x_i^r) \)

**START INNER LOOP**

- Computation of \( \mathbf{J} \), using \( \mathbf{M} \) and \( \mathbf{H} \) linearized operators around \( x^r \)
- Computation of \( \nabla \mathbf{J} \), using adjoint operators \( \mathbf{M}^T \) and \( \mathbf{H}^T \)
- Minimization via a descent method

**END OF INNER LOOP**

- Analysis increment update \( \delta x_0^a = \delta x_0 \)
- Reference state update \( x_0^r = x_0^r + \delta x_0^a \)

**END OF OUTER LOOP**

- Compute final analysis: \( x_0^a = x_0^r, x_i^a = M_{0,i}[x_0^a] \).
Incremental 4D-Var

\[ J \left( x(t_0) \right) \]

\[ J \left[ \delta x(t_0) \right] \]

\[ = J(x(t_0)) \]

\[ J[\delta x(t_0)]_{k=1} \]

1ère boucle externe

boucle interne

\[ J[\delta x(t_0)]_{k=2} \]

\[ J[\delta x(t_0)]_{k=3} \]

\[ J[\delta x(t_0)]_{k=4} \]

(from YAO user's guide)
Where am I?

1. Stochastic data assimilation
   - Notations and vocabulary
   - Best linear unbiased estimator
   - Kalman filter algorithm

2. Variational Data Assimilation
   - Principle of variational methods
   - Gradient-based optimization
   - Variational algorithms

3. Implementation issues
   - Non linearities
   - **High dimensional problems**
   - Practical adjoint coding
Matrix size problems

Vector and matrix sizes

- size of $\mathbf{x}$: $n$
- size of $\mathbf{B}$: $n \times n$
- size of $\mathbf{y}$: $m$
- size of $\mathbf{H}$: $m \times n$

For some applications, $n$ and $m$ are large ($10^6$ to $10^8$) ⇒ impossible to store/compute/multiply/inverse data assimilation matrices!
Matrix size problems

Vector and matrix sizes

- size of \( \mathbf{x} \): \( n \)
- size of \( \mathbf{y} \): \( m \) \implies\ size of \( \mathbf{B} \): \( n \times n \)
- size of \( \mathbf{H} \): \( m \times n \)

For some applications, \( n \) and \( m \) are large (\( 10^6 \) to \( 10^8 \)) \implies\ impossible to store/compute/multiply/inverse data assimilation matrices!

Possible solutions to model covariance matrices:

- Rank reduction methods: e.g. replace \( \mathbf{B} \) by \( \mathbf{S} \mathbf{S}^T \) where \( \mathbf{S} \) is smaller \( (n \times r \), with \( r \ll n \) \);
- Ensemble modelling, using Monte-Carlo method.
Rank reduction

Square root decomposition

A symmetric positive definite matrix $B$ can be decomposed into $SS^T$, where $S$ is an $n \times n$ matrix.

As before, if $n$ is large, $S$ cannot be computed/stored.
Rank reduction

Square root decomposition

A symmetric positive definite matrix $B$ can be decomposed into $SS^T$, where $S$ is a $n \times n$ matrix.

As before, if $n$ is large, $S$ cannot be computed/stored.

**Rank reduction** consists in

1. choosing only a small number $r$ of *significative* columns, to get matrix $S_r$ with size $n \times r$,

2. setting $B_r = S_rS_r^T$ and hope for $B_r \simeq B$. 
Rank reduction

Square root decomposition

A symmetric positive definite matrix $B$ can be decomposed into $SS^T$, where $S$ is a $n \times n$ matrix

As before, if $n$ is large, $S$ cannot be computed/stored.

Rank reduction consists in

1. choosing only a small number $r$ of significative columns, to get matrix $S_r$ with size $n \times r$,
2. setting $B_r = S_r S_r^T$ and hope for $B_r \simeq B$.

Methods to compute columns of $S_r$: empirical orthogonal functions, principal component analysis, proper orthogonal decomposition, singular value decomposition, etc.
Ensemble modelling

Monte-Carlo estimation

If $x_1, x_2, ..., x_N$ are $N$ realisations of $x$, then an estimator of $E(x)$, based on the law of large numbers, is given by

\[ \hat{x} = \frac{1}{N} \sum_{i=1}^{N} x_i \]
Monte-Carlo estimation

If \( x_1, x_2, \ldots, x_N \) are \( N \) realisations of \( x \), then an estimator of \( \mathbb{E}(x) \), based on the law of large numbers, is given by

\[
\hat{x} = \frac{1}{N} \sum_{i=1}^{N} x_i
\]

Similarly, if \( x_1, x_2, \ldots, x_N \) are \( N \) well-chosen states of a physical system, the background error covariance matrix \( B \) can be estimated by

\[
B \approx \hat{B} = \frac{1}{N} \sum_{i=1}^{N} (x_i - \hat{x})(x_i - \hat{x})^T
\]

\( \longrightarrow \) ensemble Kalman filter, ensemble modelling of covariances.
Where am I?

1. **Stochastic data assimilation**
   - Notations and vocabulary
   - Best linear unbiased estimator
   - Kalman filter algorithm

2. **Variational Data Assimilation**
   - Principle of variational methods
   - Gradient-based optimization
   - Variational algorithms

3. **Implementation issues**
   - Non-linearities
   - High dimensional problems
   - Practical adjoint coding
Gradient of $J^o$ factorization:

$$-\frac{1}{2} \nabla J^o(x) = \sum_{i=0}^{n} M_1^T \ldots M_{i-1}^T M_i^T H_i^T R_i^{-1} d_i$$

$$= H_0^T R_0^{-1} d_0 + M_1^T H_1^T R_1^{-1} d_1 + M_1^T M_2^T H_2^T R_2^{-1} d_2 + \ldots +$$

$$M_1^T \ldots M_{n-1}^T M_n^T H_n^T R_n^{-1} d_n$$

$$= H_0^T R_0^{-1} d_0 + M_1^T \left[ H_1^T R_1^{-1} d_1 + M_2^T \left[ H_2^T R_2^{-1} d_2 + \ldots +

M_n^T H_n^T R_n^{-1} d_n \right] \right]$$

\[\begin{align*}
\begin{cases}
x^*_k = M_{k+1}^T x^*_{k+1} + H_k^T R_k^{-1} d_k, & k = n, 0 \\
x^*_n = H_n^T R_n^{-1} d_n
\end{cases}
\Rightarrow \nabla J^o = -2x^*_0
\]
Automatic differentiation

Adjoint code construction can be very technical work:

- time-dependance and non-linearities
- non-differentiabilities, thresholds
- iterative solvers

Community portal for automatic differentiation: http://www.autodiff.org

Our favorite, with advanced features and tailored for all kind of applications, even large-scale time dependant:
Tapenade  http://www-sop.inria.fr/tropics
Merci de votre attention !