The Inversion Problem: solving parameters inversion and assimilation problems

UE Numerical Methods Workshop

Romain Brossier
romain.brossier@univ-grenoble-alpes.fr

ISTerre, Univ. Grenoble Alpes

Master 08/09/2016

Outline

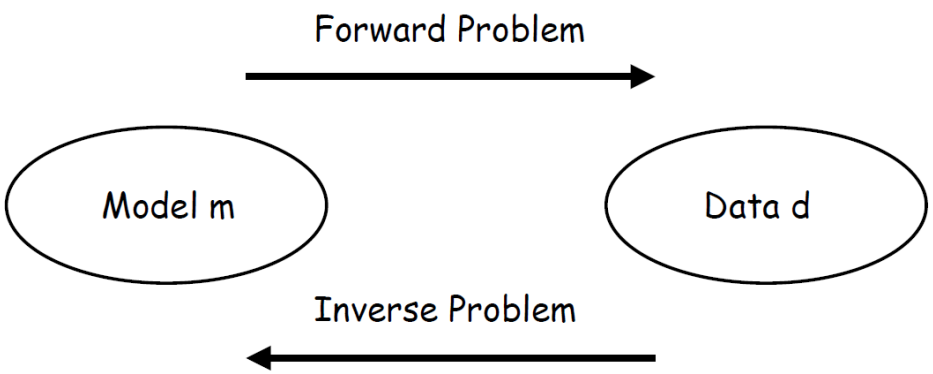
1. Introduction
2. Presentation of the inversion problem
3. Recalls on Linear Algebra
4. Inversion
5. Link with Data Assimilation
6. Project suggestion

Outline

1. Introduction
2. Presentation of the inversion problem
3. Recalls on Linear Algebra
4. Inversion
5. Link with Data Assimilation
6. Project suggestion

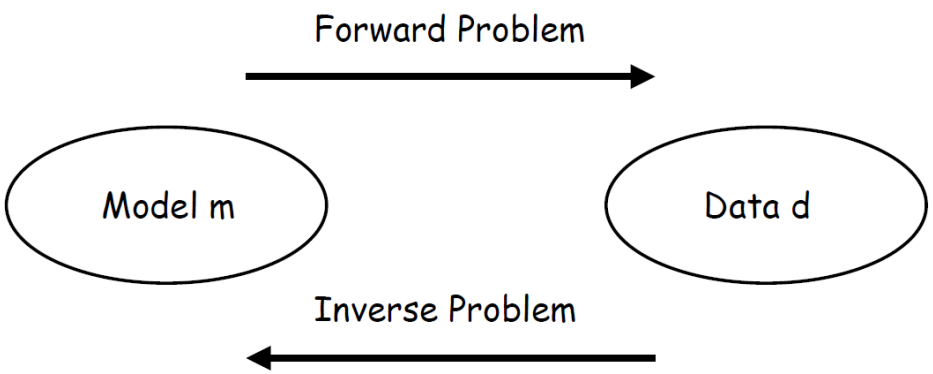
What is an inverse problem ?

- Determine and characterize the causes of a (physical) phenomenon from the (observed) effects
- *Inversion* and *inverse problem* terms are often used for the particular case of *parameters inversion* problem.
  $\rightarrow$ In the following, *inversion* = *parameters inversion*...
- If a model of a physical phenomenon is known
  - going from the model to observations is the **forward problem**
  - going back from the observations to the parameters of the model is the **inverse problem**



What is an inverse problem ?

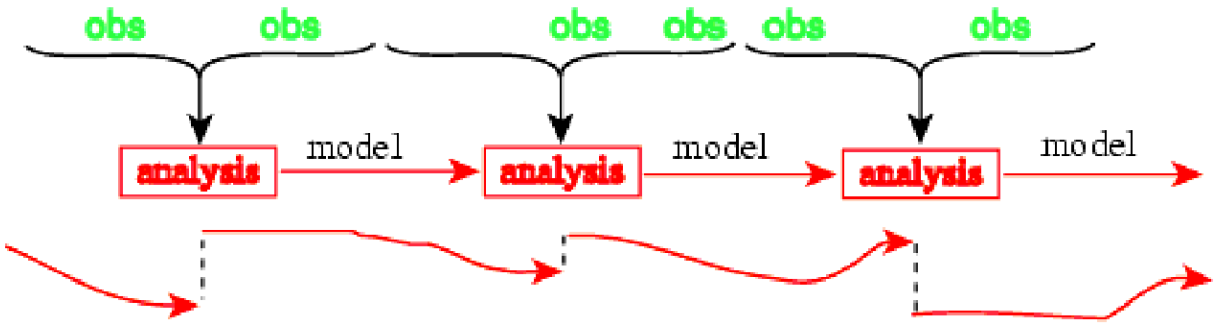
- Determine and characterize the causes of a (physical) phenomenon from the (observed) effects
- *Inversion* and *inverse problem* terms are often used for the particular case of *parameters inversion* problem.
  $\rightarrow$ In the following, *inversion* = *parameters inversion*...
- If a model of a physical phenomenon is known
  - going from the model to observations is the **forward problem**
  - going back from the observations to the parameters of the model is the **inverse problem**



- Examples of inversion : imaging (geophysics, medical sciences, astrophysics, ocean sciences...), rock physics, chemistry, signal processing, structural mechanics, finance...

What is data assimilation ?

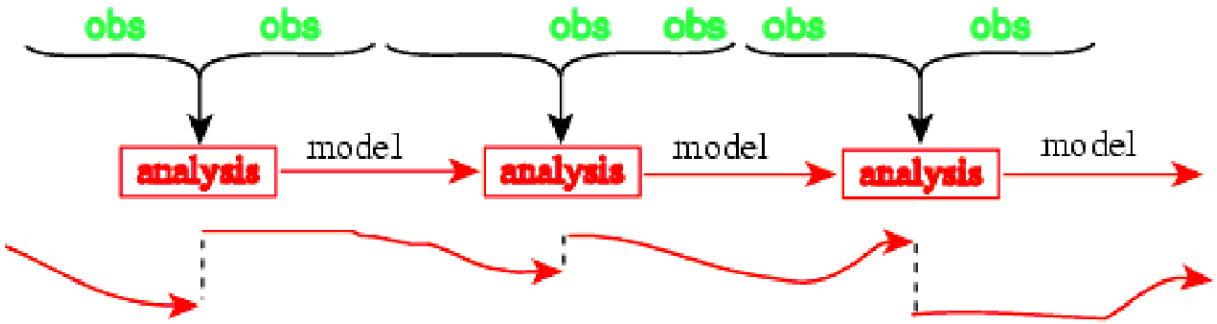
- Determine and predict the state of a dynamic system from local measurements of the state
- Assuming a model of the system's dynamic and some observations
  - determine the full state of the system : **analysis**
  - predict the future state of the system : **forecast**



- Examples of assimilation : meteorological forecast, hydrology, oceanography, MHD

What is data assimilation ?

- Determine and predict the state of a dynamic system from local measurements of the state
- Assuming a model of the system's dynamic and some observations
  - determine the full state of the system : **analysis**
  - predict the future state of the system : **forecast**



- Examples of assimilation : meteorological forecast, hydrology, oceanography, MHD
- Nothing else than an other particular inverse problem...

Inversion *vs* Assimilation

- Common features
  - Need **observations** (and associated errors/uncertainties)
  - Need a **model** (and associated errors/uncertainties)
  - Relies on an **optimization** procedure

Inversion *vs* Assimilation

- Common features
  - Need **observations** (and associated errors/uncertainties)
  - Need a **model** (and associated errors/uncertainties)
  - Relies on an **optimization** procedure
- Main differences
  - Inversion
    - ⋆ the initial state is assumed to be (well) known
    - ⋆ the observation and the inverse problem solution (the parameters) are independent from the time
  - Assimilation
    - ⋆ the initial state is a part of the optimization's solution
    - ⋆ the dynamic system and the solution are dependent from the time

Outline

1. Introduction
2. Presentation of the inversion problem
3. Recalls on Linear Algebra
4. Inversion
5. Link with Data Assimilation
6. Project suggestion

General comments

"Determine and characterize the causes of a (physical) phenomenon from the (observed) effects and consequences"

- this sentence implies
  - forward problem : "natural" and "easy"
    "the same cause(s) gives the same consequence(s)"
  - the inverse problem : "not natural" and "complex"
    "a same fact can have different distinct origins"

General comments

"Determine and characterize the causes of a (physical) phenomenon from the (observed) effects and consequences"

- this sentence implies
  - forward problem : "natural" and "easy"
    "the same cause(s) gives the same consequence(s)"
  - the inverse problem : "not natural" and "complex"
    "a same fact can have different distinct origins"
- in term of mathematics
  - the **forward problem** is in general **well-posed**
  - the **inverse problem** is in general less well-posed or **ill-posed**

A simple example : the gravimetry problem

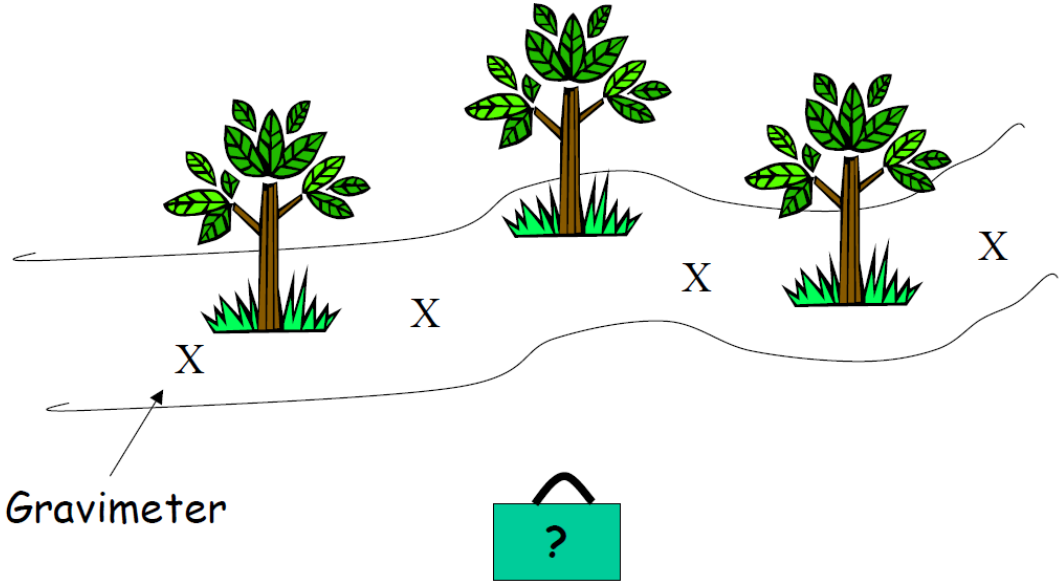


figure from H. Igel

- We measure the gravity field value at each X position
  $\rightarrow$ How can we relate the measure to the subsurface properties ?
- We know how to relate the subsurface properties to the gravitational potential
  $\rightarrow$ the **forward** problem equation

$$\Phi(r) = \int \frac{G\rho(r')}{|r - r'|} dV \tag{1}$$

A simple example : the gravimetry problem

going to a discrete world

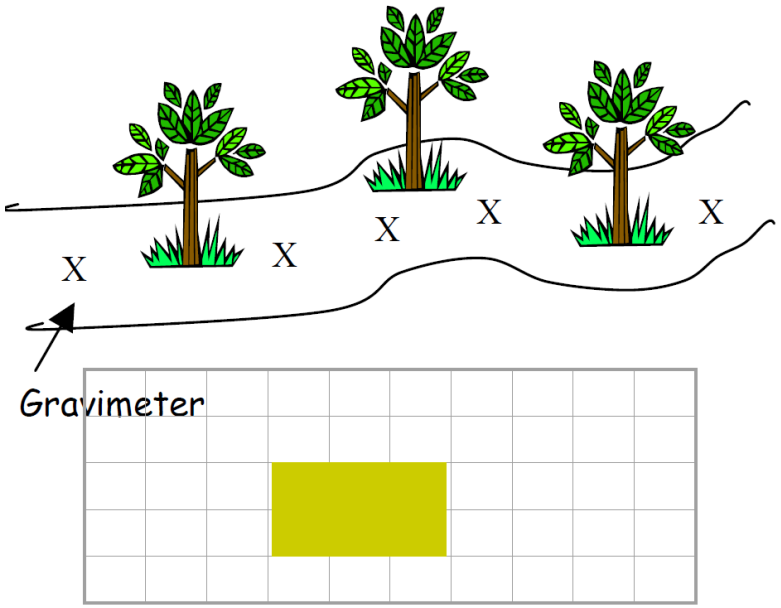


figure from H. Igel

- In order to solve the inversion problem, we go to a discrete world. But first questions :
  1. how to discretize ? which base function type : pixel ? polynomial ? wavelet ?...
  2. for box discretization (pixel type base function), which size ?

  $\rightarrow$ the physics have to be used to constrain the discretization : first **prior information**

A simple example : the gravimetry problem

Naive trial and error approach

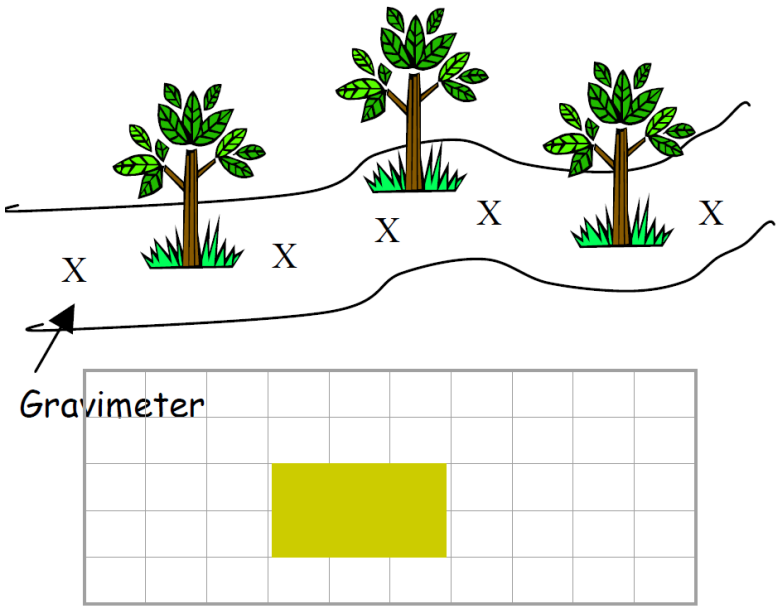


figure from H. Igel

- Assume only two density types (second prior...) : sand (2.2 kg/m^3) and gold (19.3 kg/m^3)
- Can we try all the possible models ?
  - on a 10 x 5 grid with 2 possible values per point : 2^{50} models $\approx 10^{15}$
  - assume a cost of 1 μs per forward modeling
  - cost of $\approx 10^9$ s

A simple example : the gravimetry problem

Naive trial and error approach

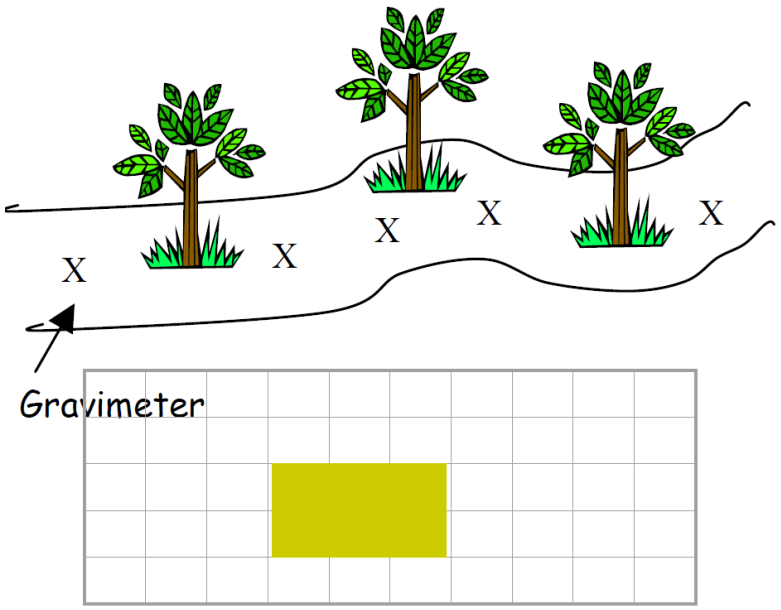


figure from H. Igel

- Assume only two density types (second prior...) : sand (2.2 kg/m^3) and gold (19.3 kg/m^3)
- Can we try all the possible models ?
  - on a 10 x 5 grid with 2 possible values per point : 2^{50} models $\approx 10^{15}$
  - assume a cost of 1 μs per forward modeling
  - cost of $\approx 10^9$ s $\approx$ 31 years... $\rightarrow$ does not seem a good approach...

A simple example : the gravimetry problem

Non-uniqueness problem

- and even if we have a BIG computer...

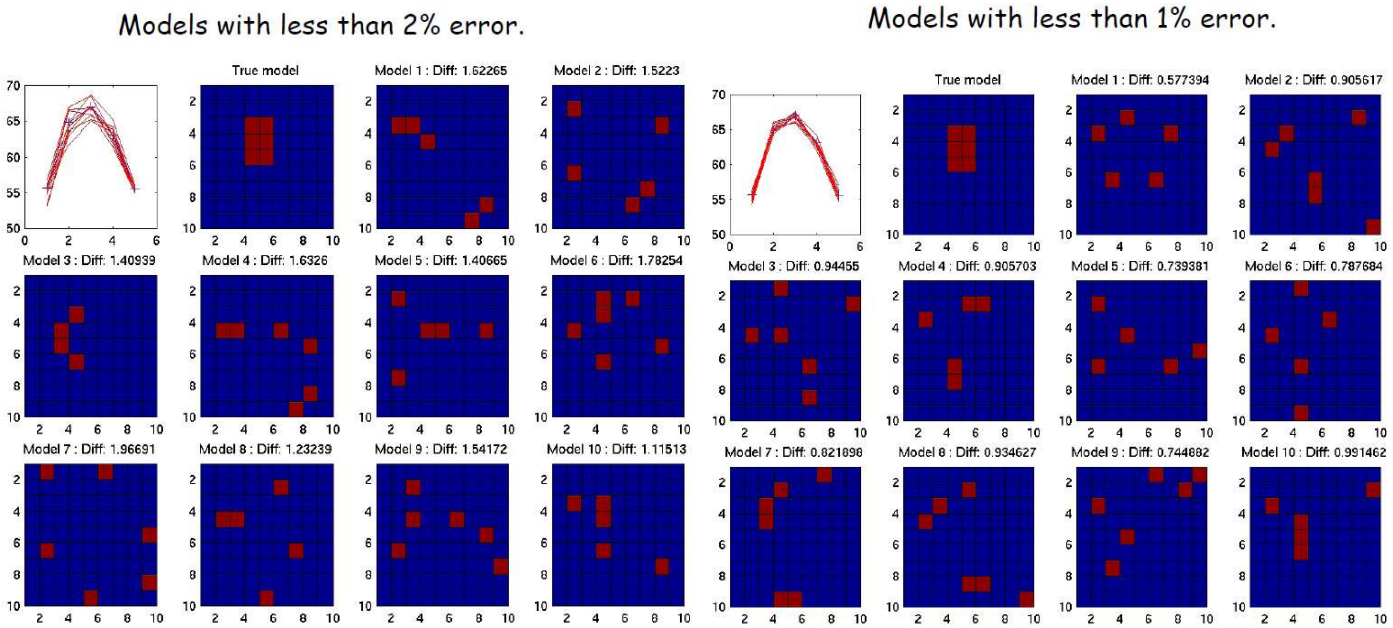


figure from H. Igel

A simple example : the gravimetry problem

Non-uniqueness problem

- and even if we have a BIG computer...

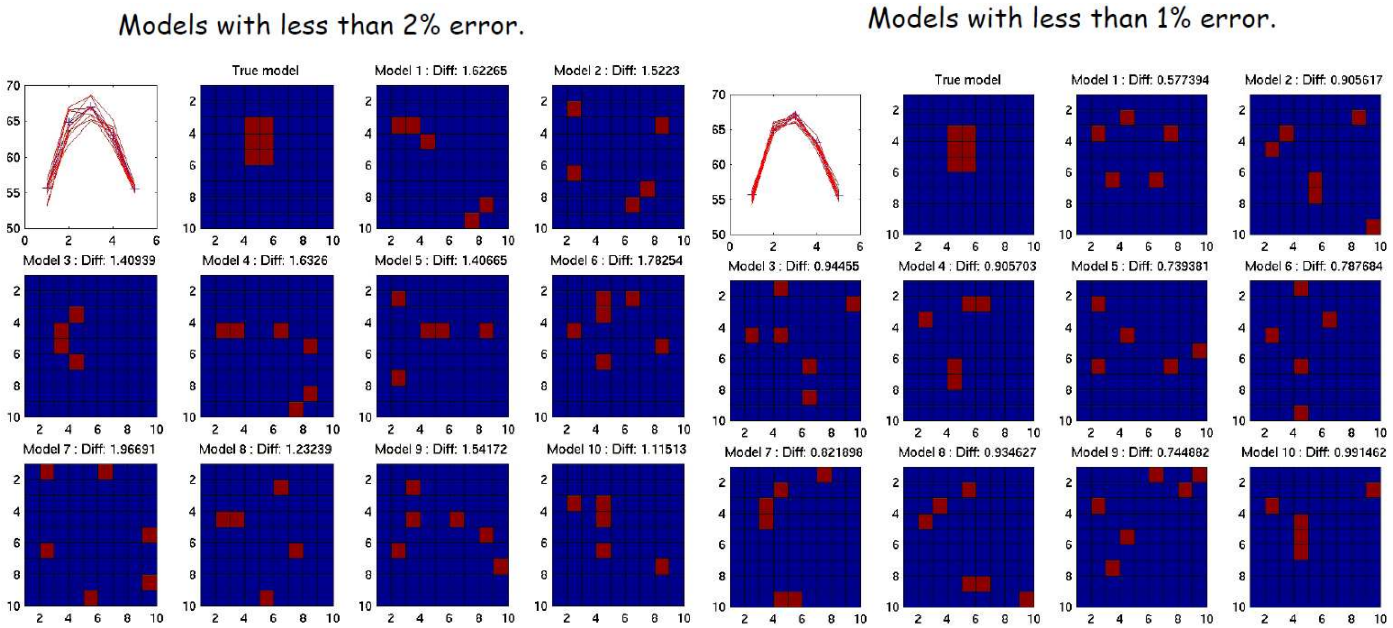


figure from H. Igel

$\rightarrow$ with only 5 observations (even perfect), inversion is strongly non unique..

A simple example : the gravimetry problem

Prior information

- we need to use other information as much as possible : **prior**
  - from the physics of gravimetry
  - from the geology
  - from other physics ? other measurements ?

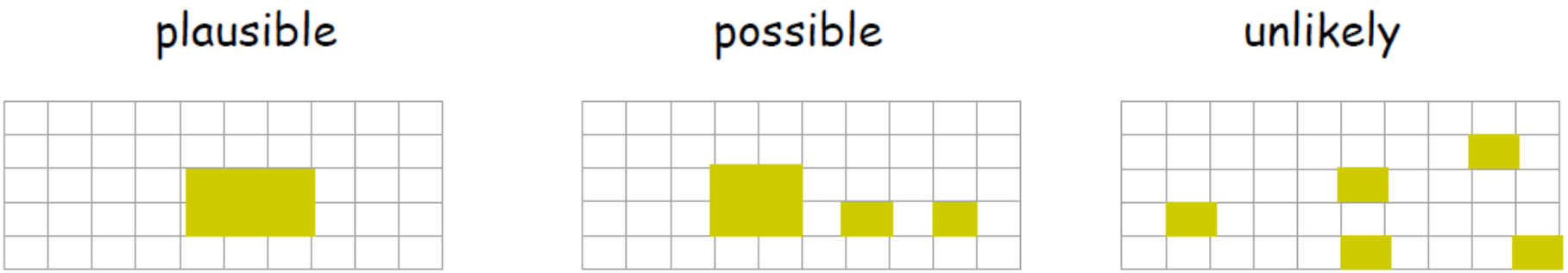


figure from H. Igel

Outline

1. Introduction
2. Presentation of the inversion problem
3. Recalls on Linear Algebra
4. Inversion
5. Link with Data Assimilation
6. Project suggestion

Recalls on Linear Algebra

- Considering a matrix A, and a vector b,

$$Ax = b \tag{2}$$

  is a linear system. x is the unknown vector
- If we have a system of two equations as

$$\begin{aligned} ax_1 + bx_2 &= l \\ cx_1 + dx_2 &= m \end{aligned} \tag{3}$$

  the matrix and vectors are filled as

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \tag{4}$$

$$b = \begin{bmatrix} l \\ m \end{bmatrix} \tag{5}$$

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{6}$$

Recalls on Linear Algebra

- giving

$$\begin{aligned} Ax &= b \\ \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} &= \begin{bmatrix} l \\ m \end{bmatrix} \end{aligned} \tag{7}$$

- If A is full ranked, the system of equations can be solved with various techniques
  - Iterative methods : Jacobi, Gauss-Siedel, Conjugate-Gradient, GMRES...
  - Direct methods : Gauss-elimination, LU decomposition...

Recalls on Linear Algebra

- giving

$$\begin{aligned} Ax &= b \\ \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} &= \begin{bmatrix} l \\ m \end{bmatrix} \end{aligned} \tag{7}$$

- If A is full ranked, the system of equations can be solved with various techniques
  - Iterative methods : Jacobi, Gauss-Siedel, Conjugate-Gradient, GMRES...
  - Direct methods : Gauss-elimination, LU decomposition...
- Attention : solving $Ax = b$ does not require to have A^{-1}, we just want x ! !

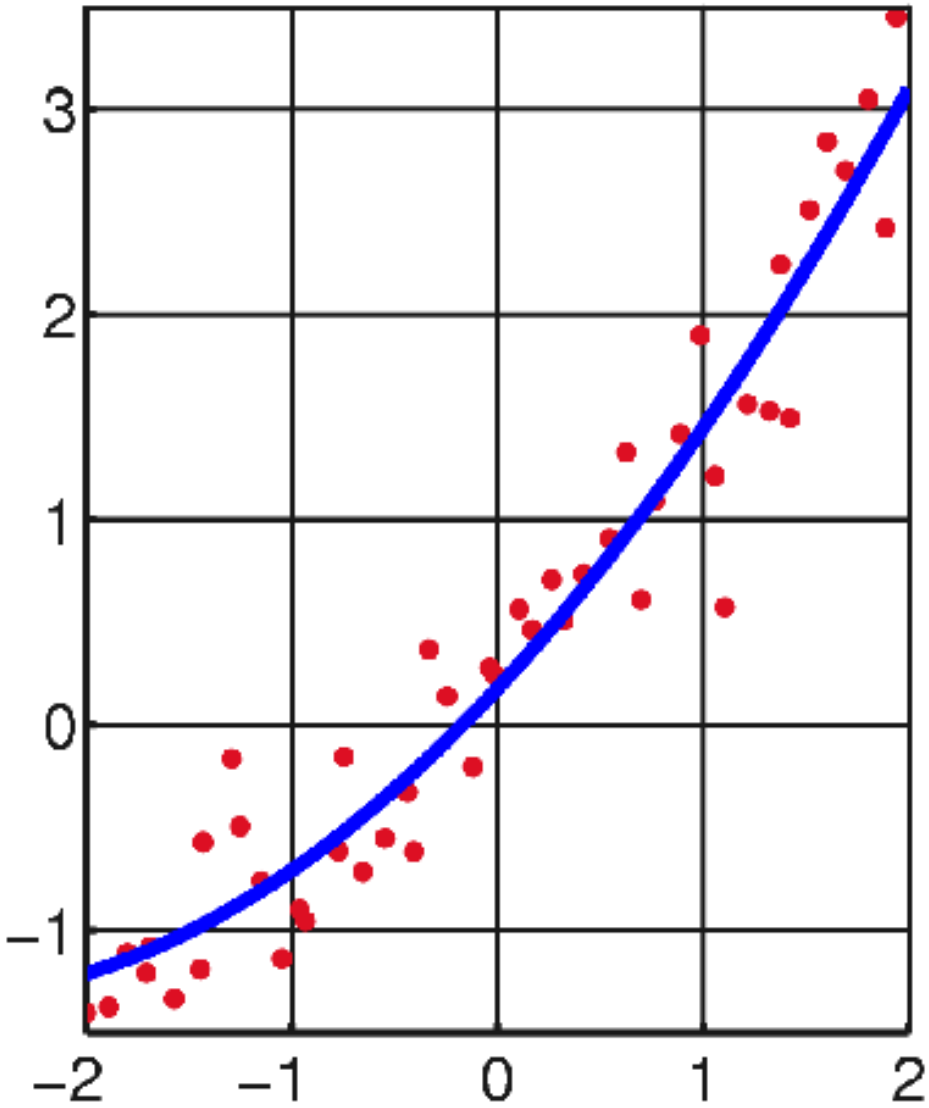
Outline

1. Introduction
2. Presentation of the inversion problem
3. Recalls on Linear Algebra
4. Inversion
   - Linear case
   - Non-linear case
   - Sum-up
5. Link with Data Assimilation
6. Project suggestion

Exemple

One inversion that you well know

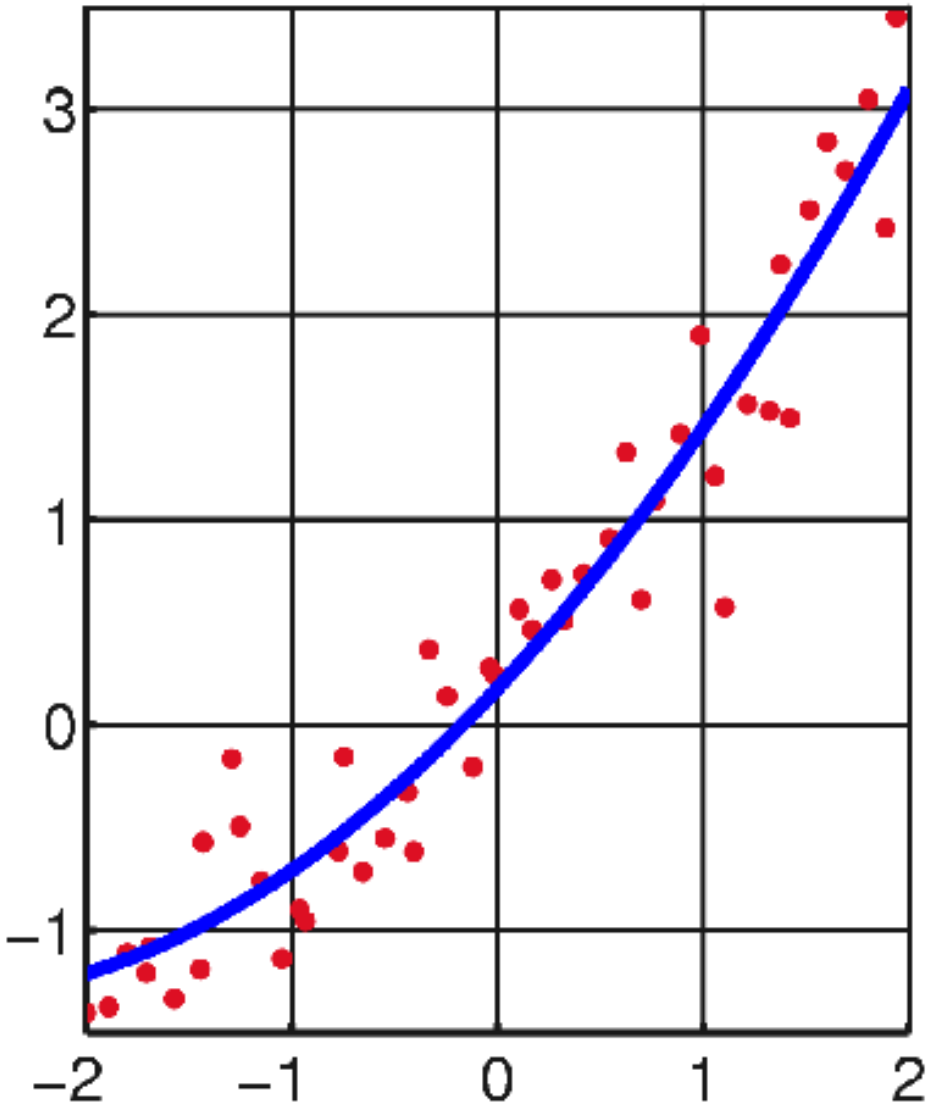
- Fitting a cloud of points under Excel
- We assume a model : for example a second order polynom $y = ax^2 + bx + c$
- We have three parameters to adjust : a, b and c



Exemple

One inversion that you well know

- Fitting a cloud of points under Excel
- We assume a model : for example a second order polynom $y = ax^2 + bx + c$
- We have three parameters to adjust : a, b and c
- Hypothesis : the model is adapted (prior information)

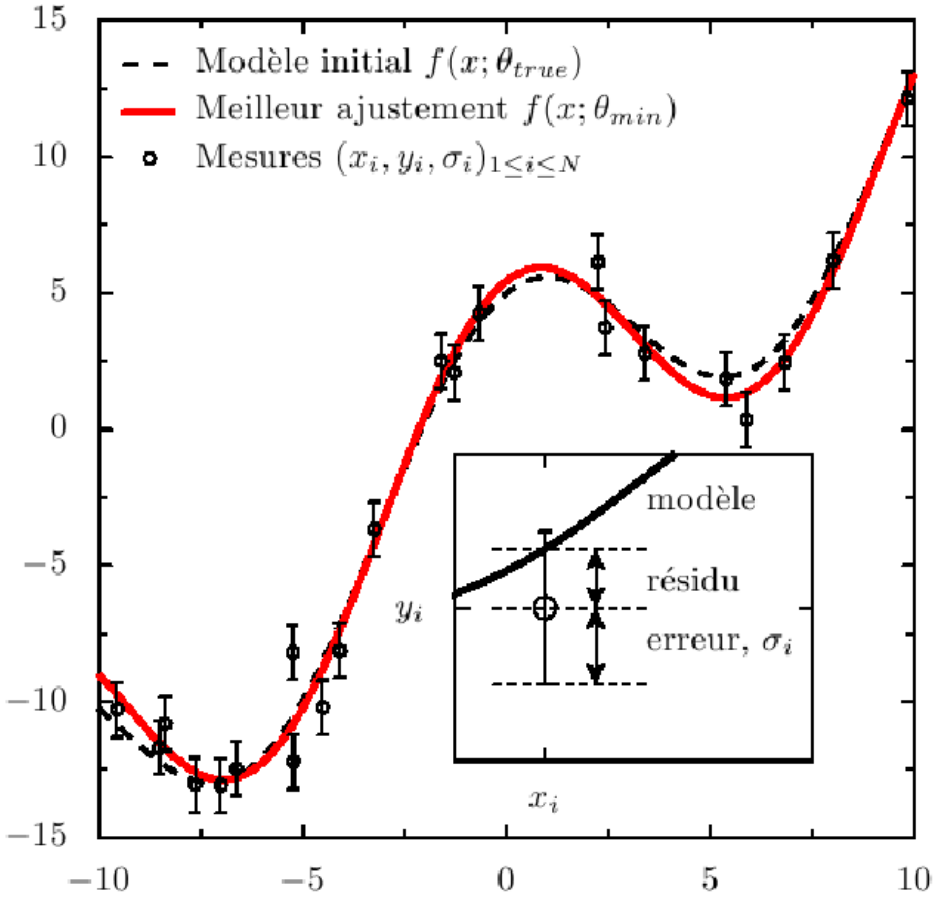


Example

To find the best set of parameter a, b and c, we define the problem

$$C = \frac{1}{2} \sum_{n_{data}} (d_{obs} - d_{cal})^2 \qquad (8)$$

- d_{obs} : observation
- d_{cal} : prediction by the model
- objective : find a and b that minimize C

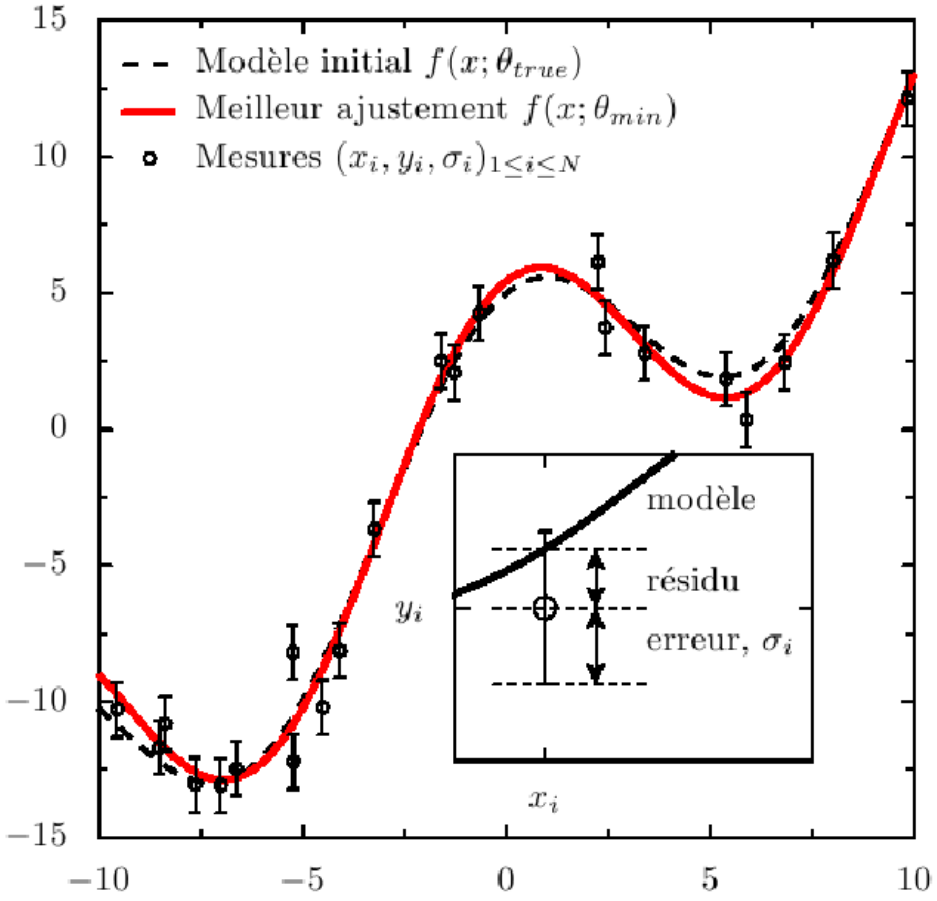


Example

To find the best set of parameter a, b and c, we define the problem

$$C = \frac{1}{2} \sum_{n_{data}} (d_{obs} - d_{cal})^2 \qquad (8)$$

- d_{obs} : observation
- d_{cal} : prediction by the model
- objective : find a and b that minimize C
- how to do in practice (what is inside Excel ?) ?



Outline

1. Introduction
2. Presentation of the inversion problem
3. Recalls on Linear Algebra
4. Inversion
   - Linear case
   - Non-linear case
   - Sum-up
5. Link with Data Assimilation
6. Project suggestion

Linear Case : presentation

- If the relation between the model parameters and the state is **linear**, we have

$$d_i = \sum_j A_{ij} m_j \tag{9}$$

that can be written in the matrix form

$$\mathbf{d} = \mathbf{A}\mathbf{m} \tag{10}$$

(forward problem equation)

Linear Case : presentation

- If the relation between the model parameters and the state is **linear**, we have

$$d_i = \sum_j A_{ij} m_j \tag{9}$$

that can be written in the matrix form

$$\mathbf{d} = \mathbf{A}\mathbf{m} \tag{10}$$

(forward problem equation)

- if $\mathbf{A}^{-1}$ can be computed, the inversion should be

$$\mathbf{m} = \mathbf{A}^{-1}\mathbf{d}_{\mathbf{obs}} \tag{11}$$

but often

  - $\mathbf{A}$ is a **rectangular matrix** of size $n_{data} \times n_{param}$
  - even if $n_{data} = n_{param}$, $\mathbf{A}$ can be **rank deficient**

Linear Case : presentation

- If the relation between the model parameters and the state is **linear**, we have

$$d_i = \sum_j A_{ij} m_j \tag{9}$$

that can be written in the matrix form

$$\mathbf{d} = \mathbf{A}\mathbf{m} \tag{10}$$

(forward problem equation)

- if $\mathbf{A}^{-1}$ can be computed, the inversion should be

$$\mathbf{m} = \mathbf{A}^{-1}\mathbf{d}_{\mathbf{obs}} \tag{11}$$

but often
  - $\mathbf{A}$ is a **rectangular matrix** of size $n_{data} \times n_{param}$
  - even if $n_{data} = n_{param}$, $\mathbf{A}$ can be **rank deficient**
- We replace the inverse $\mathbf{A}^{-1}$ by the pseudo-inverse $\mathbf{A}^{\dagger}$

$$\mathbf{m} = \mathbf{A}^{\dagger}\mathbf{d}_{\mathbf{obs}} \tag{12}$$

but how to compute $\mathbf{A}^{\dagger}$?

Linear Case : minimization

We write the problem as a minimization problem for C :

$$C(m) = ||\mathbf{d_{cal}} - \mathbf{d_{obs}}||_X = ||\mathbf{Am} - \mathbf{d_{obs}}||_X \tag{13}$$

- C is the **misfit function**, also called **cost function**
- X is the norm. In general, the l^2 norm, also called **Euclidean norm** is used (see later why)

Linear Case : minimization

- In case of Euclidean norm we minimize :

$$C = \frac{1}{2}||\mathbf{Am} - \mathbf{d_{obs}}||_2^2 \tag{14}$$

- finding the minimum of (14) is equivalent to find a zero of the misfit function gradient

$$\frac{\partial C}{\partial \mathbf{m}} = 0 \tag{15}$$

Linear Case : minimization

- In case of Euclidean norm we minimize :

$$C = \frac{1}{2}||\mathbf{Am} - \mathbf{d_{obs}}||_2^2 \tag{14}$$

- finding the minimum of (14) is equivalent to find a zero of the misfit function gradient

$$\frac{\partial C}{\partial \mathbf{m}} = 0 \tag{15}$$

- the definition of the Euclidean norm is :

$$C = \frac{1}{2}(\mathbf{Am} - \mathbf{d_{obs}})^T(\mathbf{Am} - \mathbf{d_{obs}}) \tag{16}$$

where ^{T} is transpose

Linear Case : minimization

- In case of Euclidean norm we minimize :

$$C = \frac{1}{2}||\mathbf{Am} - \mathbf{d_{obs}}||_2^2 \tag{14}$$

- finding the minimum of (14) is equivalent to find a zero of the misfit function gradient

$$\frac{\partial C}{\partial \mathbf{m}} = 0 \tag{15}$$

- the definition of the Euclidean norm is :

$$C = \frac{1}{2}(\mathbf{Am} - \mathbf{d_{obs}})^T(\mathbf{Am} - \mathbf{d_{obs}}) \tag{16}$$

  where ^{T} is transpose
- the gradient is computed as :

$$\begin{aligned} \mathcal{G} = \frac{\partial C}{\partial \mathbf{m}} &= \frac{1}{2}\left(\mathbf{A}^T(\mathbf{Am} - \mathbf{d_{obs}}) + (\mathbf{Am} - \mathbf{d_{obs}})^T\mathbf{A}\right) \\ &= \mathbf{A}^T\mathbf{Am} - \mathbf{A}^T\mathbf{d_{obs}} \end{aligned} \tag{17}$$

Linear Case : minimization

- and we want to zero the gradient :

$$\begin{aligned} \mathcal{G} = \frac{\partial C}{\partial \mathbf{m}} &= 0 \\ \mathbf{A}^T\mathbf{A}\mathbf{m} - \mathbf{A}^T\mathbf{d}_{\mathbf{obs}} &= 0 \\ \mathbf{A}^T\mathbf{A}\mathbf{m} &= \mathbf{A}^T\mathbf{d}_{\mathbf{obs}} \end{aligned} \tag{18}$$

which is a linear system resolution

Linear Case : minimization

- and we want to zero the gradient :

$$\begin{aligned} \mathcal{G} = \frac{\partial C}{\partial \mathbf{m}} &= 0 \\ \mathbf{A}^T\mathbf{A}\mathbf{m} - \mathbf{A}^T\mathbf{d_{obs}} &= 0 \\ \mathbf{A}^T\mathbf{A}\mathbf{m} &= \mathbf{A}^T\mathbf{d_{obs}} \end{aligned} \tag{18}$$

  which is a linear system resolution

- computing $\mathcal{G} = 0$ is equivalent to solve the Newton system (root-finding) :

$$\begin{aligned} \mathcal{H}\Delta\mathbf{m} &= -\mathcal{G} \\ \mathbf{A}^T\mathbf{A}\Delta\mathbf{m} &= -\mathbf{A}^T\mathbf{A}\mathbf{m_0} + \mathbf{A}^T\mathbf{d_{obs}} \end{aligned} \tag{19}$$

Linear Case : minimization

- and we want to zero the gradient :

$$\begin{aligned} \mathcal{G} = \frac{\partial C}{\partial \mathbf{m}} &= 0 \\ \mathbf{A}^T\mathbf{A}\mathbf{m} - \mathbf{A}^T\mathbf{d}_{\mathbf{obs}} &= 0 \\ \mathbf{A}^T\mathbf{A}\mathbf{m} &= \mathbf{A}^T\mathbf{d}_{\mathbf{obs}} \end{aligned} \tag{18}$$

  which is a linear system resolution

- computing $\mathcal{G} = 0$ is equivalent to solve the Newton system (root-finding) :

$$\begin{aligned} \mathcal{H}\Delta\mathbf{m} &= -\mathcal{G} \\ \mathbf{A}^T\mathbf{A}\Delta\mathbf{m} &= -\mathbf{A}^T\mathbf{A}\mathbf{m_0} + \mathbf{A}^T\mathbf{d}_{\mathbf{obs}} \end{aligned} \tag{19}$$

  we have $\Delta\mathbf{m} = \mathbf{m} - \mathbf{m_0}$

$$\begin{aligned} \mathbf{A}^T\mathbf{A}\mathbf{m} - \mathbf{A}^T\mathbf{A}\mathbf{m_0} &= -\mathbf{A}^T\mathbf{A}\mathbf{m_0} + \mathbf{A}^T\mathbf{d}_{\mathbf{obs}} \\ \mathbf{A}^T\mathbf{A}\mathbf{m} &= \mathbf{A}^T\mathbf{d}_{\mathbf{obs}} \end{aligned} \tag{20}$$

Linear Case : minimization

Solving the system :

$$\mathbf{A}^T\mathbf{A}\mathbf{m} = \mathbf{A}^T\mathbf{d_{obs}} \quad (21)$$

$$\mathbf{m} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{d_{obs}} = \mathbf{A}^\dagger\mathbf{d_{obs}} \quad (22)$$

Or the "Normal equation system" :

$$\mathbf{A}^T\mathbf{A}\Delta\mathbf{m} = -\mathbf{A}^T(\mathbf{A}\mathbf{m_0} - \mathbf{d_{obs}}) \quad (23)$$

can be tackled from different techniques

- Linear algebra techniques for Symmetric Positive Definite (SPD) matrices
  - Cholesky factorization : $\mathbf{A}^T\mathbf{A} = \mathbf{L}\mathbf{L}^T$
  - Conjugate gradient
- Singular Value Decomposition (SVD)

Linear Case : minimization

Solving the system :

$$\mathbf{A}^T\mathbf{A}\mathbf{m} = \mathbf{A}^T\mathbf{d_{obs}} \quad (21)$$

$$\mathbf{m} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{d_{obs}} = \mathbf{A}^\dagger\mathbf{d_{obs}} \quad (22)$$

Or the "Normal equation system" :

$$\mathbf{A}^T\mathbf{A}\Delta\mathbf{m} = -\mathbf{A}^T(\mathbf{A}\mathbf{m_0} - \mathbf{d_{obs}}) \quad (23)$$

can be tackled from different techniques

- Linear algebra techniques for Symmetric Positive Definite (SPD) matrices
  - Cholesky factorization : $\mathbf{A}^T\mathbf{A} = \mathbf{L}\mathbf{L}^T$
  - Conjugate gradient
- Singular Value Decomposition (SVD)

Attention, solving the 1st or the 2nd equation is really different ! !

Linear case : uncertainties and regularization

- Generally, the previous normal equation is difficult to solve
  - $\mathbf{A}^T\mathbf{A}$ is ill-condition or even singular
    $\rightarrow$ can not be solved with Cholesky or CG...
  - storage of $\mathbf{A}$ can be expensive
  - SVD is difficult to tackle for medium to large matrices

Linear case : uncertainties and regularization

- Generally, the previous normal equation is difficult to solve
  - $\mathbf{A}^T\mathbf{A}$ is ill-condition or even singular
    $\rightarrow$ can not be solved with Cholesky or CG...
  - storage of $\mathbf{A}$ can be expensive
  - SVD is difficult to tackle for medium to large matrices
- Regularization is added to the observation misfit :

$$C = \frac{1}{2}(\mathbf{Am} - \mathbf{d_{obs}})^T(\mathbf{Am} - \mathbf{d_{obs}}) + \mathbf{R}(\mathbf{m}) \tag{24}$$

$\mathbf{R}(\mathbf{m})$ should be design to improve the well-posedness of the problem

Linear case : uncertainties and regularization

- Generally, the previous normal equation is difficult to solve
  - $\mathbf{A}^T\mathbf{A}$ is ill-condition or even singular
    $\rightarrow$ can not be solved with Cholesky or CG...
  - storage of $\mathbf{A}$ can be expensive
  - SVD is difficult to tackle for medium to large matrices
- Regularization is added to the observation misfit :

$$C = \frac{1}{2}(\mathbf{Am} - \mathbf{d_{obs}})^T(\mathbf{Am} - \mathbf{d_{obs}}) + \mathbf{R}(\mathbf{m}) \tag{24}$$

  $\mathbf{R}(\mathbf{m})$ should be design to improve the well-posedness of the problem
- We can also add prior and uncertainty information, as covariance matrices :

$$C = \frac{1}{2}(\mathbf{Am}-\mathbf{d_{obs}})^T\mathbf{C_d}^{-1}(\mathbf{Am}-\mathbf{d_{obs}})+\frac{1}{2}(\mathbf{m}-\mathbf{m_{prior}})^T\mathbf{C_m}^{-1}(\mathbf{m}-\mathbf{m_{prior}}) \tag{25}$$

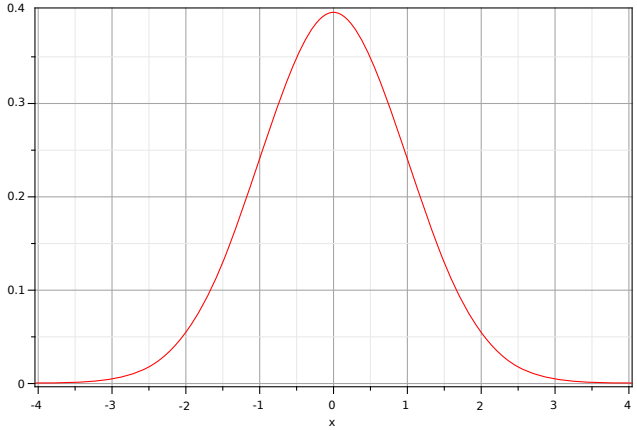
  where
  - $\mathbf{C_d}$ is the covariance matrix on the data
  - $\mathbf{C_m}$ is the covariance matrix on the model parameters
  - $\mathbf{m_{prior}}$ is the prior model parameters vector

Linear case : Few words about statistics

- All the previous developments are done with the Euclidean norm l^2 norm
- The norm choice relies on a choice of **probabilities density function** (pdf) for each error (observation, model)

Linear case : Few words about statistics

- All the previous developments are done with the Euclidean norm l^2 norm
- The norm choice relies on a choice of **probabilities density function** (pdf) for each error (observation, model)
- For Gaussian errors and linear problems, we can show that the solution of l^2 problem gives the maximum likelihood solution of the pdf
- For non-Gaussian errors, the optimal norm should be adapted to the statistics
- In practice, l^2 norm and therefore Gaussian statistics are often assumed



You can refer to the book of Tarantola (1987) for a complete view on the statistical point

Linear case : covariance

Covariance matrices allow to include uncertainties and correlations between data/model parameters

- The diagonal elements contain the variance : $C_{d_{i,i}} = var(e_i) = \sigma_i^2$
- The off-diagonal elements contain the covariance : $C_{d_{i,j}} = covar(e_i, e_j)$
  - for data, observations are generally uncorrelated $covar(e_i, e_j) = 0$
  - for model parameters, covariance represent the local correlation between parameters.
    Forms in $\exp\left(\frac{|x|}{l_{ref}}\right)$ or $\exp\left(\frac{x^2}{l_{ref}^2}\right)$ to enforce smoothness in the model for example

Linear case : solution with covariance

- From the cost function with covariance

$$C = \frac{1}{2}(\mathbf{Am}-\mathbf{d_{obs}})^T \mathbf{C_d}^{-1}(\mathbf{Am}-\mathbf{d_{obs}})+\frac{1}{2}(\mathbf{m}-\mathbf{m_{prior}})^T \mathbf{C_m}^{-1}(\mathbf{m}-\mathbf{m_{prior}}) \quad (26)$$

- we have the gradient

$$\mathcal{G} = \mathbf{A}^T \mathbf{C_d}^{-1}(\mathbf{Am} - \mathbf{d_{obs}}) + \mathbf{C_m}^{-1}(\mathbf{m} - \mathbf{m_{prior}}) \quad (27)$$

- the Hessian :

$$\mathcal{H} = \mathbf{A}^T \mathbf{C_d}^{-1}\mathbf{A} + \mathbf{C_m}^{-1} \quad (28)$$

Linear case : solution with covariance

- From the cost function with covariance

$$C = \frac{1}{2}(\mathbf{Am}-\mathbf{d_{obs}})^T \mathbf{C_d}^{-1}(\mathbf{Am}-\mathbf{d_{obs}})+\frac{1}{2}(\mathbf{m}-\mathbf{m_{prior}})^T \mathbf{C_m}^{-1}(\mathbf{m}-\mathbf{m_{prior}}) \quad (29)$$

- the solution is given by

$$\mathbf{m} = \left(\mathbf{A}^T \mathbf{C_d}^{-1}\mathbf{A} + \mathbf{C_m}^{-1}\right)^{-1}\left(\mathbf{A}^T \mathbf{C_d}^{-1}\mathbf{d_{obs}} + \mathbf{C_m}^{-1}\mathbf{m_{prior}}\right)$$

Linear case : solution with covariance

- From the cost function with covariance

$$C = \frac{1}{2}(\mathbf{Am}-\mathbf{d_{obs}})^T \mathbf{C_d}^{-1}(\mathbf{Am}-\mathbf{d_{obs}})+\frac{1}{2}(\mathbf{m}-\mathbf{m_{prior}})^T \mathbf{C_m}^{-1}(\mathbf{m}-\mathbf{m_{prior}}) \quad (29)$$

- the solution is given by

$$\begin{aligned} \mathbf{m} &= \left(\mathbf{A}^T \mathbf{C_d}^{-1}\mathbf{A} + \mathbf{C_m}^{-1}\right)^{-1} \left(\mathbf{A}^T \mathbf{C_d}^{-1}\mathbf{d_{obs}} + \mathbf{C_m}^{-1}\mathbf{m_{prior}}\right) \\ &= \mathbf{m_{prior}} + \left(\mathbf{A}^T \mathbf{C_d}^{-1}\mathbf{A} + \mathbf{C_m}^{-1}\right)^{-1} \mathbf{A}^T \mathbf{C_d}^{-1} \left(\mathbf{d_{obs}} - \mathbf{Am_{prior}}\right) \end{aligned}$$

Linear case : solution with covariance

- From the cost function with covariance

$$C = \frac{1}{2}(\mathbf{Am}-\mathbf{d_{obs}})^T \mathbf{C_d}^{-1}(\mathbf{Am}-\mathbf{d_{obs}})+\frac{1}{2}(\mathbf{m}-\mathbf{m_{prior}})^T \mathbf{C_m}^{-1}(\mathbf{m}-\mathbf{m_{prior}}) \quad (29)$$

- the solution is given by

$$\begin{aligned} \mathbf{m} &= \left(\mathbf{A}^T \mathbf{C_d}^{-1}\mathbf{A} + \mathbf{C_m}^{-1}\right)^{-1}\left(\mathbf{A}^T \mathbf{C_d}^{-1}\mathbf{d_{obs}} + \mathbf{C_m}^{-1}\mathbf{m_{prior}}\right) \\ &= \mathbf{m_{prior}} + \left(\mathbf{A}^T \mathbf{C_d}^{-1}\mathbf{A} + \mathbf{C_m}^{-1}\right)^{-1}\mathbf{A}^T \mathbf{C_d}^{-1}\left(\mathbf{d_{obs}} - \mathbf{Am_{prior}}\right) \\ &= \mathbf{m_{prior}} + \mathbf{C_m}\mathbf{A}^T\left(\mathbf{AC_mA}^T + \mathbf{C_d}\right)^{-1}\left(\mathbf{d_{obs}} - \mathbf{Am_{prior}}\right) \end{aligned} \quad (30)$$

Linear case : solution with covariance

- From the cost function with covariance

$$C = \frac{1}{2}(\mathbf{Am}-\mathbf{d_{obs}})^T \mathbf{C_d}^{-1}(\mathbf{Am}-\mathbf{d_{obs}})+\frac{1}{2}(\mathbf{m}-\mathbf{m_{prior}})^T \mathbf{C_m}^{-1}(\mathbf{m}-\mathbf{m_{prior}}) \quad (29)$$

- the solution is given by

$$\begin{aligned} \mathbf{m} &= \left(\mathbf{A}^T\mathbf{C_d}^{-1}\mathbf{A}+\mathbf{C_m}^{-1}\right)^{-1}\left(\mathbf{A}^T\mathbf{C_d}^{-1}\mathbf{d_{obs}}+\mathbf{C_m}^{-1}\mathbf{m_{prior}}\right) \\ &= \mathbf{m_{prior}}+\left(\mathbf{A}^T\mathbf{C_d}^{-1}\mathbf{A}+\mathbf{C_m}^{-1}\right)^{-1}\mathbf{A}^T\mathbf{C_d}^{-1}\left(\mathbf{d_{obs}}-\mathbf{Am_{prior}}\right) \\ &= \mathbf{m_{prior}}+\mathbf{C_m}\mathbf{A}^T\left(\mathbf{AC_mA}^T+\mathbf{C_d}\right)^{-1}\left(\mathbf{d_{obs}}-\mathbf{Am_{prior}}\right) \quad (30) \end{aligned}$$

- the posterior covariance matrix (should tends to small value)

$$\begin{aligned} \tilde{\mathbf{C}}_\mathbf{m} &= \left(\mathbf{A}^T\mathbf{C_d}^{-1}\mathbf{A}+\mathbf{C_m}^{-1}\right)^{-1} \\ &= \mathbf{C_m}-\mathbf{C_m}\mathbf{A}^T\left(\mathbf{AC_mA}^T+\mathbf{C_d}\right)^{-1}\mathbf{AC_m} \quad (31) \end{aligned}$$

Linear case : solution with covariance

- From the cost function with covariance

$$C = \frac{1}{2}(\mathbf{Am}-\mathbf{d_{obs}})^T \mathbf{C_d}^{-1}(\mathbf{Am}-\mathbf{d_{obs}})+\frac{1}{2}(\mathbf{m}-\mathbf{m_{prior}})^T \mathbf{C_m}^{-1}(\mathbf{m}-\mathbf{m_{prior}}) \quad (29)$$

- the solution is given by

$$\begin{aligned} \mathbf{m} &= \left(\mathbf{A}^T\mathbf{C_d}^{-1}\mathbf{A}+\mathbf{C_m}^{-1}\right)^{-1}\left(\mathbf{A}^T\mathbf{C_d}^{-1}\mathbf{d_{obs}}+\mathbf{C_m}^{-1}\mathbf{m_{prior}}\right) \\ &= \mathbf{m_{prior}}+\left(\mathbf{A}^T\mathbf{C_d}^{-1}\mathbf{A}+\mathbf{C_m}^{-1}\right)^{-1}\mathbf{A}^T\mathbf{C_d}^{-1}\left(\mathbf{d_{obs}}-\mathbf{Am_{prior}}\right) \\ &= \mathbf{m_{prior}}+\mathbf{C_m}\mathbf{A}^T\left(\mathbf{AC_mA}^T+\mathbf{C_d}\right)^{-1}\left(\mathbf{d_{obs}}-\mathbf{Am_{prior}}\right) \end{aligned} \quad (30)$$

- the posterior covariance matrix (should tends to small value)

$$\begin{aligned} \tilde{\mathbf{C}}_\mathbf{m} &= \left(\mathbf{A}^T\mathbf{C_d}^{-1}\mathbf{A}+\mathbf{C_m}^{-1}\right)^{-1} \\ &= \mathbf{C_m}-\mathbf{C_m}\mathbf{A}^T\left(\mathbf{AC_mA}^T+\mathbf{C_d}\right)^{-1}\mathbf{AC_m} \end{aligned} \quad (31)$$

- link with the resolution operator (should tends to identity)

$$\mathbf{R} = \mathbf{C_m}\mathbf{A}^T\left(\mathbf{AC_mA}^T+\mathbf{C_d}\right)^{-1}\mathbf{A} = \mathbf{I}-\tilde{\mathbf{C}}_\mathbf{m}\mathbf{C_m}^{-1} \quad (32)$$

Example : adjustment of a linear function

- $y(x_i)$ the observed values for each x_i
- $ax_i + b$ the model $\rightarrow$ the two parameters to adjust are a and b
- the misfit function is $C = \sum_i 0.5(y(x_i) - (ax_i + b))^2$
- we want this misfit function minimum
  $\rightarrow$ the gradient of C should be zero
- the gradient is

$$\begin{aligned} \nabla_a C &= -\sum_i x_i y(x_i) + \sum_i x_i^2 a + \sum_i x_i b = 0 \\ \nabla_b C &= -\sum_i y(x_i) + \sum_i x_i a + \sum_i x_i b = 0 \end{aligned} \tag{33}$$

which gives

$$\begin{aligned} \begin{bmatrix} \sum_i x_i^2 & \sum_i x_i \\ \sum_i x_i & \sum_i 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} &= \begin{bmatrix} \sum_i x_i y(x_i) \\ \sum_i y(x_i) \end{bmatrix} \\ \rightarrow \begin{bmatrix} a \\ b \end{bmatrix} &= \begin{bmatrix} \sum_i x_i^2 & \sum_i x_i \\ \sum_i x_i & \sum_i 1 \end{bmatrix}^{-1} \begin{bmatrix} \sum_i x_i y(x_i) \\ \sum_i y(x_i) \end{bmatrix} \end{aligned} \tag{34}$$

- exercice : redo the same with error (covariance matrix) on measurments

Outline

1. Introduction
2. Presentation of the inversion problem
3. Recalls on Linear Algebra
4. Inversion
   - Linear case
   - Non-linear case
   - Sum-up
5. Link with Data Assimilation
6. Project suggestion

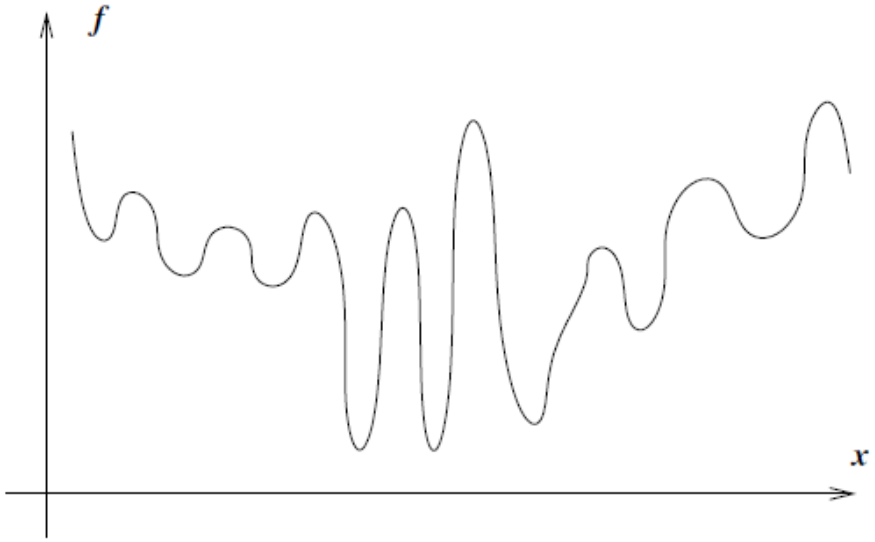
Non-Linear Case : presentation

- In some cases, the relation between the model parameters and the state is **non-linear**, we have

$$\mathbf{d} = \mathbf{g}(\mathbf{m}) \tag{35}$$

- two general ways to tackle the inverse problem
  1. **global optimization** : allows "strongly" non-linear problems but small parameters number ($\approx \mathcal{O}(10^0) - \mathcal{O}(10^2)$)
  2. **local optimization** : only for weakly non-linear problems, small to large number of parameters ($> \mathcal{O}(10^6)$)

  $\rightarrow$ for large number of parameters and strongly non-linear problems, no tractable methods...



Non-Linear Case : global optimization

- The aim of global optimization is to sample the model space in order to find the global minimum of the problem
- Several methods exist, based on different exploration strategies (see the book of Sen & Stoffa, 1995)
  - grid search : systematic research
  - Monte Carlo method : random exploration
  - Simulated annealing : "driven" random exploration
  - Genetic algorithm
  - Neighborhood algorithm
  - ...

Non-Linear Case : global optimization

- The aim of global optimization is to sample the model space in order to find the global minimum of the problem
- Several methods exist, based on different exploration strategies (see the book of Sen & Stoffa, 1995)
  - grid search : systematic research
  - Monte Carlo method : random exploration
  - Simulated annealing : "driven" random exploration
  - Genetic algorithm
  - Neighborhood algorithm
  - ...
- all required lots of forward simulations

Non-Linear Case : local optimization

- Local optimization for non-linear problem is based on successive linearization of the problem
- A first order Taylor development of the misfit in the vicinity of the starting model gives

$$C(\mathbf{m_0} + \Delta\mathbf{m}) = C(\mathbf{m_0}) + \frac{\partial C(\mathbf{m_0})}{\partial \mathbf{m}} \Delta\mathbf{m} + \mathcal{O}(\mathbf{m}^2) \tag{36}$$

Non-Linear Case : local optimization

- Local optimization for non-linear problem is based on successive linearization of the problem
- A first order Taylor development of the misfit in the vicinity of the starting model gives

$$C(\mathbf{m_0} + \Delta\mathbf{m}) = C(\mathbf{m_0}) + \frac{\partial C(\mathbf{m_0})}{\partial \mathbf{m}} \Delta\mathbf{m} + \mathcal{O}(\mathbf{m}^2) \tag{36}$$

- Taking the derivative with respect to the model parameters **m** gives

$$\frac{\partial C(\mathbf{m})}{\partial \mathbf{m}} = \frac{\partial C(\mathbf{m_0})}{\partial \mathbf{m}} + \frac{\partial^2 C(\mathbf{m_0})}{\partial \mathbf{m}^2} \Delta\mathbf{m} \tag{37}$$

Non-Linear Case : local optimization

- Local optimization for non-linear problem is based on successive linearization of the problem
- A first order Taylor development of the misfit in the vicinity of the starting model gives

$$C(\mathbf{m_0} + \Delta\mathbf{m}) = C(\mathbf{m_0}) + \frac{\partial C(\mathbf{m_0})}{\partial \mathbf{m}} \Delta\mathbf{m} + \mathcal{O}(\mathbf{m}^2) \tag{36}$$

- Taking the derivative with respect to the model parameters **m** gives

$$\frac{\partial C(\mathbf{m})}{\partial \mathbf{m}} = \frac{\partial C(\mathbf{m_0})}{\partial \mathbf{m}} + \frac{\partial^2 C(\mathbf{m_0})}{\partial \mathbf{m}^2} \Delta\mathbf{m} \tag{37}$$

- The minimum of the misfit function is reached when the gradient vanishes

$$\Delta\mathbf{m} = -\left[\frac{\partial^2 \mathbf{C}(\mathbf{m_0})}{\partial \mathbf{m^2}}\right]^{-1} \frac{\partial \mathbf{C}(\mathbf{m_0})}{\partial \mathbf{m}}. \tag{38}$$

Non-Linear Case : local optimization

- Local optimization for non-linear problem is based on successive linearization of the problem
- A first order Taylor development of the misfit in the vicinity of the starting model gives

$$C(\mathbf{m_0} + \Delta\mathbf{m}) = C(\mathbf{m_0}) + \frac{\partial C(\mathbf{m_0})}{\partial \mathbf{m}} \Delta\mathbf{m} + \mathcal{O}(\mathbf{m}^2) \tag{36}$$

- Taking the derivative with respect to the model parameters **m** gives

$$\frac{\partial C(\mathbf{m})}{\partial \mathbf{m}} = \frac{\partial C(\mathbf{m_0})}{\partial \mathbf{m}} + \frac{\partial^2 C(\mathbf{m_0})}{\partial \mathbf{m}^2} \Delta\mathbf{m} \tag{37}$$

- The minimum of the misfit function is reached when the gradient vanishes

$$\Delta\mathbf{m} = -\left[\frac{\partial^2 \mathbf{C}(\mathbf{m_0})}{\partial \mathbf{m^2}}\right]^{-1} \frac{\partial \mathbf{C}(\mathbf{m_0})}{\partial \mathbf{m}}. \tag{38}$$

$\rightarrow$ we retrieve the Normal equation system !

Non-Linear Case : local optimization

- based on the least-square norm, we have the cost function

$$C = \frac{1}{2}(\mathbf{g}(\mathbf{m}) - \mathbf{d}_{\mathbf{obs}})^T(\mathbf{g}(\mathbf{m}) - \mathbf{d}_{\mathbf{obs}}) \tag{39}$$

- the gradient :

$$\mathcal{G} = \frac{\partial \mathbf{g}(\mathbf{m})}{\partial \mathbf{m}}^T (\mathbf{g}(\mathbf{m}) - \mathbf{d}_{\mathbf{obs}}) = \mathbf{J}^T(\mathbf{g}(\mathbf{m}) - \mathbf{d}_{\mathbf{obs}}) \tag{40}$$

where $\mathbf{J}$ is the **sensitivity** or **Fréchet** derivative matrix
$\rightarrow$ the linearization or the non-linear forward problem operator $\mathbf{g}(\mathbf{m})$

Non-Linear Case : local optimization

- based on the least-square norm, we have the cost function

$$C = \frac{1}{2}(\mathbf{g}(\mathbf{m}) - \mathbf{d_{obs}})^T(\mathbf{g}(\mathbf{m}) - \mathbf{d_{obs}}) \tag{39}$$

- the gradient :

$$\mathcal{G} = \frac{\partial \mathbf{g}(\mathbf{m})}{\partial \mathbf{m}}^T(\mathbf{g}(\mathbf{m}) - \mathbf{d_{obs}}) = \mathbf{J}^T(\mathbf{g}(\mathbf{m}) - \mathbf{d_{obs}}) \tag{40}$$

  where **J** is the **sensitivity** or **Fréchet** derivative matrix
  $\rightarrow$ the linearization or the non-linear forward problem operator $\mathbf{g}(\mathbf{m})$

- the Hessian :

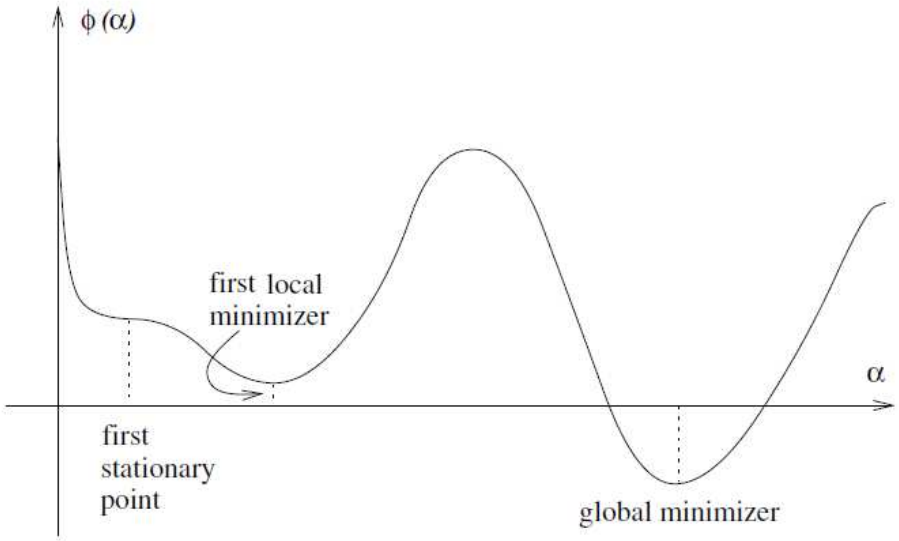
$$\mathcal{H} = \mathbf{J}^T\mathbf{J} + \frac{\partial \mathbf{J}}{\partial \mathbf{m}}^T(\Delta\mathbf{d}..\Delta\mathbf{d}) \tag{41}$$

Non-Linear Case : local optimization

- the non-linear problem is solved as an iterative linear problem...
- ... but we retrieve also the issues of the linear problem
  - ill-posed normal equation
  - large storage of $\mathbf{J}^T\mathbf{J}$

Non-Linear Case : local optimization

- the non-linear problem is solved as an iterative linear problem...
- ... but we retrieve also the issues of the linear problem
  - ill-posed normal equation
  - large storage of $\mathbf{J}^T\mathbf{J}$
- ...but also other issue
  - $\mathbf{J}$ can be expensive to compute
  - $\mathbf{m_0}$ must be in the global minimum valley



Non-Linear Case : local optimization

- least-square norm with covariance matrices and regularization

$$C = \frac{1}{2}(\mathbf{g}(\mathbf{m}) - \mathbf{d_{obs}})^T \mathbf{C_d}^{-1}(\mathbf{g}(\mathbf{m}) - \mathbf{d_{obs}}) + \frac{1}{2}(\mathbf{m} - \mathbf{m_{prior}})^T \mathbf{C_m}^{-1}(\mathbf{m} - \mathbf{m_{prior}}) \tag{42}$$

- the gradient :

$$\mathcal{G} = \mathbf{J}^T \mathbf{C_d}^{-1}(\mathbf{g}(\mathbf{m}) - \mathbf{d_{obs}}) + \mathbf{C_m}^{-1}(\mathbf{m} - \mathbf{m_{prior}}) \tag{43}$$

Non-Linear Case : local optimization

- least-square norm with covariance matrices and regularization

$$C = \frac{1}{2}(\mathbf{g}(\mathbf{m}) - \mathbf{d_{obs}})^T \mathbf{C_d}^{-1}(\mathbf{g}(\mathbf{m}) - \mathbf{d_{obs}}) + \frac{1}{2}(\mathbf{m} - \mathbf{m_{prior}})^T \mathbf{C_m}^{-1}(\mathbf{m} - \mathbf{m_{prior}}) \tag{42}$$

- the gradient :

$$\mathcal{G} = \mathbf{J}^T \mathbf{C_d}^{-1}(\mathbf{g}(\mathbf{m}) - \mathbf{d_{obs}}) + \mathbf{C_m}^{-1}(\mathbf{m} - \mathbf{m_{prior}}) \tag{43}$$

- the Hessian (linear part only, second order neglected here) :

$$\mathcal{H} = \mathbf{J}^T \mathbf{C_d}^{-1} \mathbf{J} + \mathbf{C_m}^{-1} \tag{44}$$

Non-Linear Case : local optimization

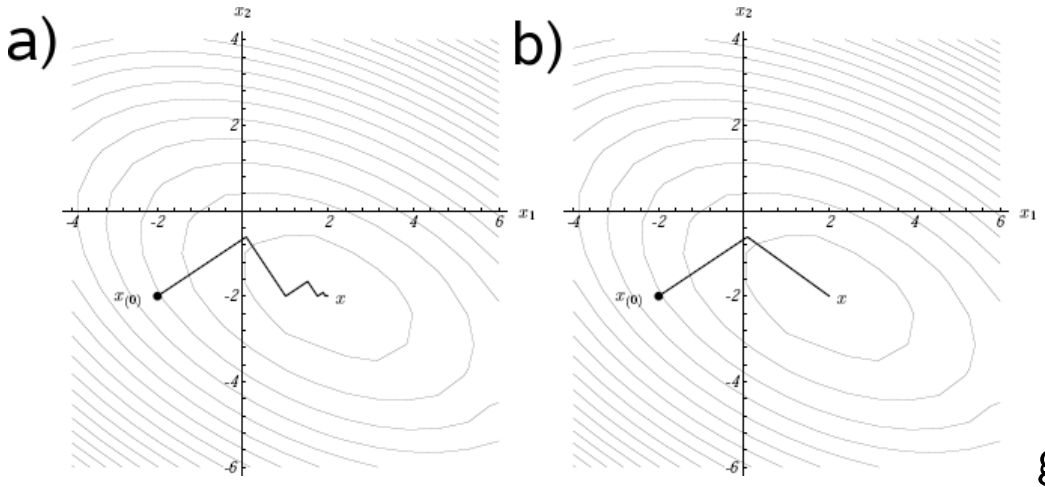
Optimization methods

- for large scale problems, the gradient $\mathcal{G}$ is generally computed from the adjoint method (see Plessix, 2006, for a review in geophysics)
  $\rightarrow$ avoid the explicit computation of $\mathbf{J}$

Non-Linear Case : local optimization

Optimization methods

- for large scale problems, the gradient $\mathcal{G}$ is generally computed from the adjoint method (see Plessix, 2006, for a review in geophysics)
  $\rightarrow$ avoid the explicit computation of $\mathbf{J}$
- based on $\mathcal{G}$, classic optimization schemes for realistic problems rely on (see the book of Nocedal & Wright, 1999)
  - gradient
  - non-linear conjugate gradient (NLCG)
  - preconditioned gradient or NLCG
  - Quasi-Newton : cheap estimation of $\mathcal{H}$ or $\mathbf{H}^{-1}$
  - Gauss or Full-Newton : requires $\mathcal{H}$, quite expensive and not often used

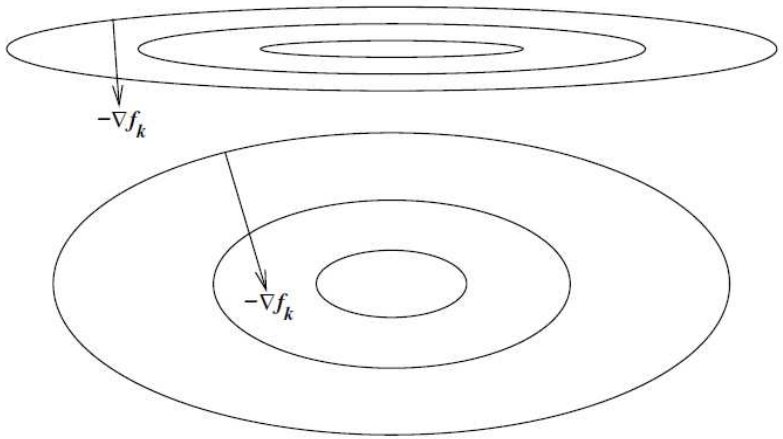


gradient *vs* CG

Non-Linear Case : local optimization

Optimization methods

- for large scale problems, the gradient $\mathcal{G}$ is generally computed from the adjoint method (see Plessix, 2006, for a review in geophysics)
  $\rightarrow$ avoid the explicit computation of $\mathbf{J}$
- based on $\mathcal{G}$, classic optimization schemes for realistic problems rely on (see the book of Nocedal & Wright, 1999)
  - gradient
  - non-linear conjugate gradient (NLCG)
  - preconditioned gradient or NLCG
  - Quasi-Newton : cheap estimation of $\mathcal{H}$ or $\mathbf{H}^{-1}$
  - Gauss or Full-Newton : requires $\mathcal{H}$, quite expensive and not often used



hessian effects

Outline

1. Introduction
2. Presentation of the inversion problem
3. Recalls on Linear Algebra
4. Inversion
   - Linear case
   - Non-linear case
   - Sum-up
5. Link with Data Assimilation
6. Project suggestion

Sum-up

- inversion is often a difficult task
- in the linear case, a linear system is solved (or SVD)
- in the non-linear case
  - global optimization schemes can be used for small number of parameters and "cheap" forward problems
  - local optimization schemes can be used on locally linearized problems (with a good knowledge of the solution...)
- for both linear and non-linear problems, a good knowledge of the physics and prior information should be used to design efficient regularization
  $\rightarrow$ improvement of well-posedness

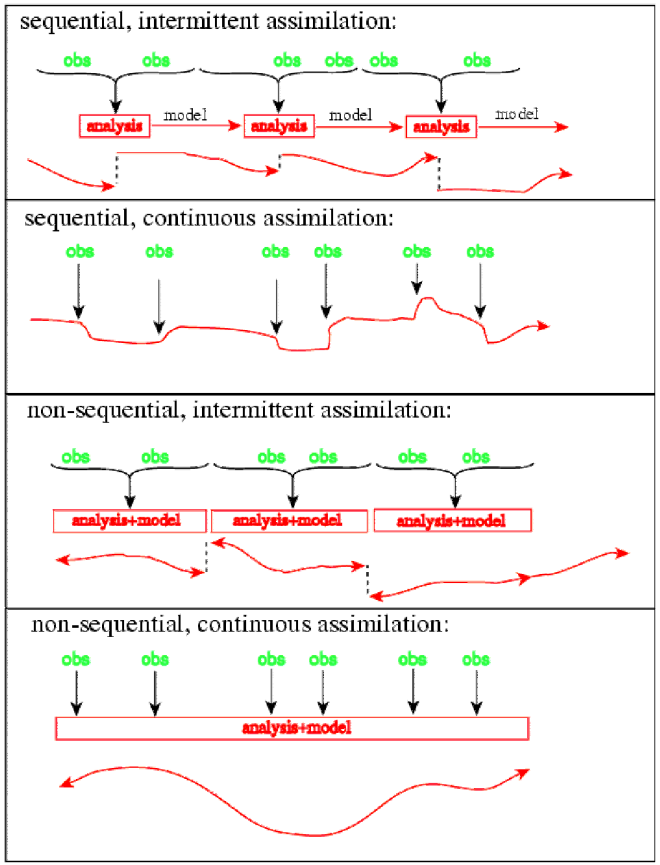
Outline

1. Introduction
2. Presentation of the inversion problem
3. Recalls on Linear Algebra
4. Inversion
5. Link with Data Assimilation
6. Project suggestion

Presentation

"Determine and predict the state of a dynamic system from local measurements of the state"

- determine the full system state : **analysis** → underconstrained inverse problem
- predict the future system state : **forecast** → modeling in the future



- sequential : real-time assimilation
- non-sequential : retrospective analysis
- intermittent : more adapted to discrete world with finite number of analysis
- continuous : more realistic, but continuous analysis...

Notations and definitions

Despite the fact that **analysis** is an inverse problem, specific notations and terms are used in assimilation...
Remark : we no more try to find the model parameters **m**, but the state vector of system : **x**

- State vector
  - $\mathbf{x}_t$ (dim n) : the **true** state vector, is the best possible representation of reality
  - $\mathbf{x}_b$ (dim n) : the **background** state vector before analysis. Equivalent to starting/prior model
  - $\mathbf{x}_a$(dim n) : the **analysis** state vector, the final state after analysis

Notations and definitions

Despite the fact that **analysis** is an inverse problem, specific notations and terms are used in assimilation...
Remark : we no more try to find the model parameters **m**, but the state vector of system : **x**

- State vector
  - $\mathbf{x}_t$ (dim n) : the **true** state vector, is the best possible representation of reality
  - $\mathbf{x}_b$ (dim n) : the **background** state vector before analysis. Equivalent to starting/prior model
  - $\mathbf{x}_a$(dim n) : the **analysis** state vector, the final state after analysis
- Control variable (solution of the analysis inverse problem)
  - $\Delta\mathbf{x}$ (dim n) : the **analysis increment** vector, defined as $\mathbf{x}_a = \mathbf{x}_b + \Delta\mathbf{x}$

Notations and definitions

Despite the fact that **analysis** is an inverse problem, specific notations and terms are used in assimilation...
Remark : we no more try to find the model parameters **m**, but the state vector of system : **x**

- State vector
  - $\mathbf{x}_t$ (dim n) : the **true** state vector, is the best possible representation of reality
  - $\mathbf{x}_b$ (dim n) : the **background** state vector before analysis. Equivalent to starting/prior model
  - $\mathbf{x}_a$(dim n) : the **analysis** state vector, the final state after analysis
- Control variable (solution of the analysis inverse problem)
  - $\Delta\mathbf{x}$ (dim n) : the **analysis increment** vector, defined as $\mathbf{x}_a = \mathbf{x}_b + \Delta\mathbf{x}$
- Observations
  - $\mathbf{y}$ (dim p) : the **observation vector**
  - $\mathcal{H}$ (operator from nxp) : the **observation operator**, that extracts synthetic observations $\mathbf{y}_{\mathbf{synth}}$ from the state vector : $\mathbf{y}_{\mathbf{synth}} = \mathcal{H}(\mathbf{x})$. Its linearized version is $\mathbf{H}$

Notations and definitions

- Departure (residuals)
  - $\mathbf{y} - \mathcal{H}(\mathbf{x}_b)$: the **innovation** vector
  - $\mathbf{y} - \mathcal{H}(\mathbf{x}_a)$: the **analysis residual** vector

Notations and definitions

- Departure (residuals)
  - $\mathbf{y} - \mathcal{H}(\mathbf{x}_b)$: the **innovation** vector
  - $\mathbf{y} - \mathcal{H}(\mathbf{x}_a)$: the **analysis residual** vector
- Covariance matrices
  - **B** (dim nxn) : covariance matrix related to **background errors**, difference between background state and true state. (="model covariance matrix" in inversion)
  - **R** (dim pxp) : covariance matrix related to **observation errors** (instrumental, errors in $\mathcal{H}$, discretization effects) (="data covariance matrix" in inversion)
  - **A** (dim nxn) : covariance matrix related to **analysis errors**, difference between $\mathbf{x}_a$ and $\mathbf{x}_t$ (="posterior covariance matrix" in inversion)

Least-square solution

- the Best Linear Unbiased Estimate (BLUE) analysis gives us the solution

$$\mathbf{x}_a = \mathbf{x}_b + \mathbf{K}(\mathbf{y} - \mathcal{H}(\mathbf{x}_b)) \tag{45}$$

with $\mathbf{K}$ the gain matrix defined as

$$\mathbf{K} = \mathbf{B}\mathbf{H}^T(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R})^{-1} \tag{46}$$

Least-square solution

- the Best Linear Unbiased Estimate (BLUE) analysis gives us the solution

$$\mathbf{x}_a = \mathbf{x}_b + \mathbf{K}(\mathbf{y} - \mathcal{H}(\mathbf{x}_b)) \tag{45}$$

with $\mathbf{K}$ the gain matrix defined as

$$\mathbf{K} = \mathbf{B}\mathbf{H}^T(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R})^{-1} \tag{46}$$

- the "more" classical least-square problem (variational approach) based on the misfit function gives us the same solution

$$C = \frac{1}{2}(\mathbf{y} - \mathcal{H}(\mathbf{x}_b))^T\mathbf{R}^{-1}(\mathbf{y} - \mathcal{H}(\mathbf{x}_b)) + \frac{1}{2}(\mathbf{x}_a - \mathbf{x}_b)^T\mathbf{B}^{-1}(\mathbf{x}_a - \mathbf{x}_b) \tag{47}$$

and

$$\Delta\mathbf{x} \quad = \quad (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{B}^{-1})^{-1}\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{y} - \mathcal{H}(\mathbf{x}_b))$$

Least-square solution

- the Best Linear Unbiased Estimate (BLUE) analysis gives us the solution

$$\mathbf{x}_a = \mathbf{x}_b + \mathbf{K}(\mathbf{y} - \mathcal{H}(\mathbf{x}_b)) \tag{45}$$

with $\mathbf{K}$ the gain matrix defined as

$$\mathbf{K} = \mathbf{B}\mathbf{H}^T(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R})^{-1} \tag{46}$$

- the "more" classical least-square problem (variational approach) based on the misfit function gives us the same solution

$$C = \frac{1}{2}(\mathbf{y} - \mathcal{H}(\mathbf{x}_b))^T\mathbf{R}^{-1}(\mathbf{y} - \mathcal{H}(\mathbf{x}_b)) + \frac{1}{2}(\mathbf{x}_a - \mathbf{x}_b)^T\mathbf{B}^{-1}(\mathbf{x}_a - \mathbf{x}_b) \tag{47}$$

and

$$\begin{aligned} \Delta\mathbf{x} &= (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{B}^{-1})^{-1}\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{y} - \mathcal{H}(\mathbf{x}_b)) \\ &= \mathbf{B}\mathbf{H}^T(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R})^{-1}(\mathbf{y} - \mathcal{H}(\mathbf{x}_b)) \end{aligned} \tag{48}$$

(same as inversion before)

Least-square solution

- the Best Linear Unbiased Estimate (BLUE) analysis gives us the solution

$$\mathbf{x}_a = \mathbf{x}_b + \mathbf{K}(\mathbf{y} - \mathcal{H}(\mathbf{x}_b)) \tag{45}$$

with $\mathbf{K}$ the gain matrix defined as

$$\mathbf{K} = \mathbf{B}\mathbf{H}^T(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R})^{-1} \tag{46}$$

- the "more" classical least-square problem (variational approach) based on the misfit function gives us the same solution

$$C = \frac{1}{2}(\mathbf{y} - \mathcal{H}(\mathbf{x}_b))^T\mathbf{R}^{-1}(\mathbf{y} - \mathcal{H}(\mathbf{x}_b)) + \frac{1}{2}(\mathbf{x}_a - \mathbf{x}_b)^T\mathbf{B}^{-1}(\mathbf{x}_a - \mathbf{x}_b) \tag{47}$$

and

$$\begin{aligned} \Delta\mathbf{x} &= (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{B}^{-1})^{-1}\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{y} - \mathcal{H}(\mathbf{x}_b)) \\ &= \mathbf{B}\mathbf{H}^T(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R})^{-1}(\mathbf{y} - \mathcal{H}(\mathbf{x}_b)) \end{aligned} \tag{48}$$

(same as inversion before)

- the analysis errors covariance matrix is given by

$$\mathbf{A} = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{B} \tag{49}$$

Least-square solution : few remarks

- For non-linear problems, as for inversion, a local linearization should be performed to use the linear framework (or global optimization)
- The previous equations gives the best solution associated to Gaussian errors model and l^2 norm optimization
- It remains however computationally expensive to implement strictly these equations for large scale problems... (Kalman filter)
- Several algorithms are therefore used in practice to tackle realistic problems
  - Optimal Interpolation
  - 3D-VAR
  - 4D-VAR
  - the Kalman filter
  - Ensemble Kalman filter and variants

Sum-up

- Sequential assimilation is an iterative two-steps procedure
  - Analysis : an inverse problem
  - Forecast : a forward problem
- for large scales problems, the analysis encounters the same issues than inversion problems, in which case the full least-square solution is not tractable...

Outline

1. Introduction
2. Presentation of the inversion problem
3. Recalls on Linear Algebra
4. Inversion
5. Link with Data Assimilation
6. Project suggestion

Linear and non-linear deconvolution

The deconvolution problem starts from the convolutional model of signal

$$g(t) = e(t) * f(t) + n(t) \tag{50}$$

where

- $g(t)$ is the observed signal
- $e(t)$ is the impulse response of the studied system (a sensor, Earth...)
- $f(t)$ is the input signal put on the studied system, assumed or not to be well known
- $n(t)$ is an observation noise

Linear and non-linear deconvolution

The deconvolution problem starts from the convolutional model of signal

$$g(t) = e(t) * f(t) + n(t) \tag{50}$$

where

- $g(t)$ is the observed signal
- $e(t)$ is the impulse response of the studied system (a sensor, Earth...)
- $f(t)$ is the input signal put on the studied system, assumed or not to be well known
- $n(t)$ is an observation noise

The project will focus on :

1. Define a linear inverse problem to retrieve $e(t)$
2. Define a prior some prior on $e(t)$ to regularize the problem, ending with linear or non-linear problem to solve
3. Application to seismic data, possibly with multiple observation (and associated other regularization terms)

Nocedal, J. & Wright, S. J. (1999). *Numerical Optimization*. New York, US : Springer.

Plessix, R. E. (2006). A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. *Geophysical Journal International*, 167(2):495–503.

Sen, M. K. & Stoffa, P. L. (1995). *Global Optimization Methods in Geophysical Inversion*. Elsevier Science Publishing Co.

Tarantola, A. (1987). *Inverse problem theory : methods for data fitting and model parameter estimation*. Elsevier, New York.