

Data assimilation for engineers

2022-2023

Olivier THUAL et Mohamed SAADI
Toulouse INP-ENSEEIH/MFEE

Course:

Data Assimilation (ASID)

Teaching Unit:

Numerical Methods for Scientific Computing in Aerodynamics

Where are we? MFEE/MSN and MFEE/MSN-BD

Teaching Unit: Numerical Methods for Scientific Computing

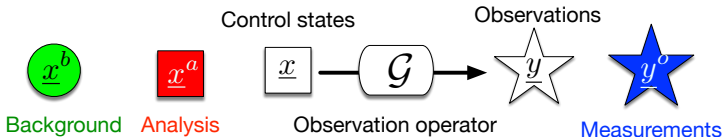
Three courses in this Teaching Unit (5 ECTS):

- Numerical methods: simulation of incompressible flows (35%)
- Numerical methods: simulation of compressible flows (35%)
- Data Assimilation (30%)

What are the learning outcomes?

At the end of the lecture, students should be able to:

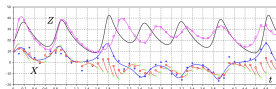
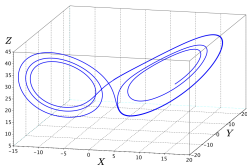
- Describe the basic formalism of data assimilation
- Explain the specification of a cost function on several examples
- Choose several methods to minimize the cost function



At the end of the numerical projects, student should be able to:

- Write computer programs for several methods with several examples
- Compare and comment the performances of several methods
- Achieve convincing forecast for chaotic dynamical systems

Data assimilation for engineers



Toulouse INP - ENSEIHT

"Fluid Mechanics, Energetics and Environment" Department

Year 2022-2023, November 23, 2022

Contents

1 From weather forecast to engineer applications	7
1 How is weather forecasted	8
2 Other examples of data assimilation applications	10
3 What time is it?	13
2 Generic cost function	19
1 Looking for the minimum of a cost function	20
2 Linearization methods	22
3 How will the bore propagate?	24
3 Time dependent models	31
1 The 4D-Var data assimilation	32
2 The Kalman filter data assimilation	36
3 Will the water overflow?	39
4 Example with variational and ensemble methods	47
1 Ensemble methods for three simple examples	48
2 Data assimilation methods for the Lorenz model	51
3 Data assimilation for the advection-diffusion model	55
Bibliographie	59

What is the program of the course?

CM 1	<ul style="list-style-type: none">○ Course presentation① From weather forecast to engineer application <p>Homework A: Exo 1.1 “What time is it?”</p> <ul style="list-style-type: none">② Generic cost function (1/2) <p>Homework B: Exo 2.1 “How will the bore propagate?”</p>
TDM 1	Improvement of Homeworks A and B
CM2 & 3	<ul style="list-style-type: none">② Generic cost function (2/2)③ Time dependent models④ Application projects <p>Homework C: Exo 3.3 and Section 4.1 “Methods for the Lorenz model”</p>
TDM 2& 3	④ Improvements of Homework C
Exam	○ Presentation of the projects by the groups

How will the course be evaluated?

Three Jupyter notebooks to put on Moodle before deadlines

- 1 Introduction, results and conclusion
- 2 Text, equations and figures, including animated gif
- 3 Python code inside the notebook

Homework A : What time is it? (individual 20%)

- Follow exercise 1.1 and go beyond

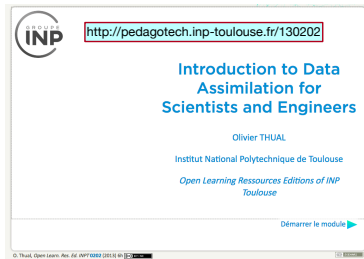
Homework B : How will the bore propagate? (individual 20%)


- Follow exercise 2.1 and go beyond

Homework report C: Methods for the Lorenz model (trinomial 60%)

- Follow exercise 3.3 and reproduce Section 4.1
- Go beyond with innovations

Homeworks A and B: Two simple exercises



 <http://pedagotech.inp-toulouse.fr/130202>

Introduction to Data Assimilation for Scientists and Engineers

Olivier THUAL

Institut National Polytechnique de Toulouse

Open Learning Ressources Editions of INP Toulouse

Démarrer le module ▶

© Thual, Open Learn. Rev. Ed. INPT 0002 (2016) 01


Exos 1.1 and 2.1

- 1 What time is it?
- 2 How will the bore propagate?


Link to the ressource:
Available from web or Moodle

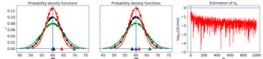


Hands-on for time estimation


 Méthode

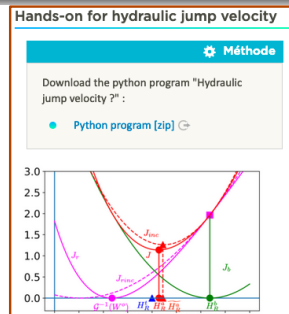
Download the program "What time is it?" :

- Python program [zip] 




From two given value of clock time, which uncertainty is know, the program computes the best estimate of the clock time.


Hands-on instructions:  Simulation

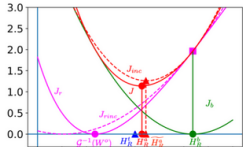


Hands-on for hydraulic jump velocity

 Méthode

Download the python program "Hydraulic jump velocity ?" :

- Python program [zip] 



The graph plots hydraulic jump velocity J_b against a parameter $g^{-1}(W^*)$. It shows several curves: a solid purple curve J_b , a dashed red curve $J_{b,crit}$, a solid green curve J_b , and a dashed blue curve $J_{b,crit}$. Key points on the x-axis are marked with $H_{1,c}$, $H_{1,c}$, $H_{1,c}$, and $H_{1,c}$. A vertical red line connects a point on the dashed red curve to the x-axis. A vertical green line connects a point on the solid green curve to the x-axis.

Homeworks C: A big project



Course major resources



Course' Book: Data assimilation for engineers (15Mo) 14.8Mo

Slides: Data assimilation for engineers 13.5Mo

O. Thual, Introduction to Data Assimilation for Scientists and Engineers, Open Learn. Res. Ed. INPT 0202 (2013) 6h

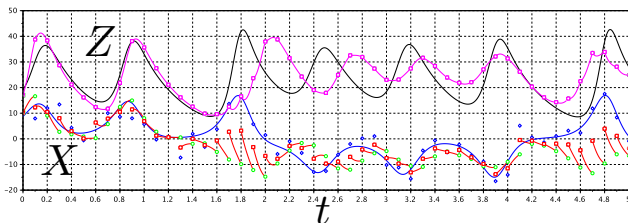
Python programs 24.2Ko

Scilab programs 16.8Ko

Exo 3.3 and Section 4.1

- 1 Mandatory : chains of BLUE, Kalman filter, Incremental 4DVar
- 2 Exploratory : Kalman filter with error model, Ensemble Kalman filter...

Link to the resource:
Available from Moodle



Pedagogical resources

Moodle N7

All documents on ASID Moodle course :

<http://moodle-n7.inp-toulouse.fr/mod/data/view.php?id=759>

Online courses

- O. Thual, Introduction to Data Assimilation for Scientists and Engineers, *Open Learn. Res. Ed. INPT*, **0202** (2013) 6h
<https://www.olivier-thual.fr/130202>
- O. Pannekoucke, Introduction to data assimilation, *Open Learn. Res. Ed. INPT*, **0831** (2013) 6h
<http://pedagotech.inp-toulouse.fr/130831>
- *Open Learn. Res. Ed. INPT*, **0831** (2013) 6h S. Gratton and Ph. Toint, Numerical methods for Data Assimilation, *Open Learn. Res. Ed. INPT*, **0826** (2013) 6h
<http://pedagotech.inp-toulouse.fr/130826>

Outline of the slides

1. From weather forecast to engineer applications

Data assimilation for meteorology, oceanography, hydrology, aerodynamics...
Motivation of the formalism with a simple model: “What time is it?”

2. Generic cost function

Incremental methods on a very simple example: “How will the bore propagate?” Generalization to any dimensions and ensemble methods.

3. Time dependent models

Observation operator including a temporal model. Data assimilation example through a simple model : “Will the water overflow?”

4. Application projects

Comparing four data assimilation methods on a chaotic dynamical system: “Lorenz model” .

Data assimilation for engineers

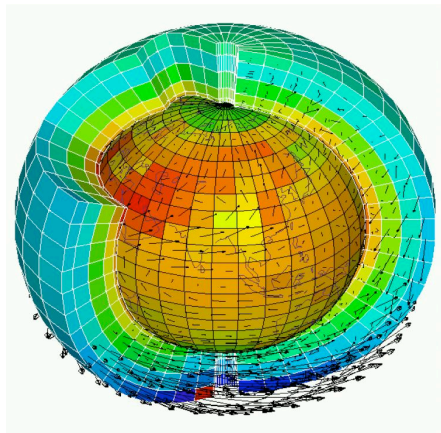
Chapter 1: From weather forecast to engineer applications

Olivier THUAL
Toulouse INP-ENSEEIH/ MFEE

Course:
Data Assimilation (ASID)

Teaching Unit:
Numerical Methods for Scientific Computing in Aerodynamics

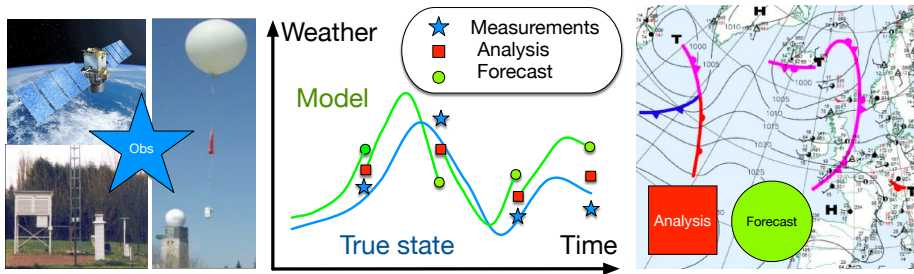
How is weather forecasted?



Atmospheric model:

Fluid mechanics equations for winds, pressure, temperature and humidity.

Data assimilation chain : a minimization problem



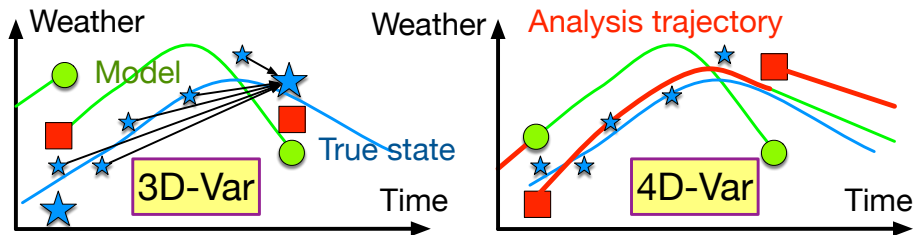
$$J(\blacksquare) = \frac{1}{2} \|\blacksquare - \bullet\|_B^2 + \frac{1}{2} \|\star - \mathcal{G}(\blacksquare)\|_R^2$$

Looking for the present weather : the analysis

A state close to both the previous forecast and the field measurements

Two century of data assimilation

- XVIIIth: planet orbit computations by Gauss and least square method by Legendre.
- XXth: concept of maximum likelihood by Fisher, Kalman filter for the APOLLO program and objective analysis of meteorological fields.
- End of XXth: 3D-Var data assimilation for weather forecast model.

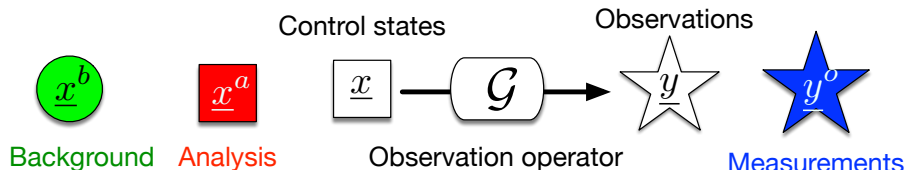


- XXIth: gain of 20% forecast quality at Météo-France with 4D-Var

The standard formalism of data assimilation

The analysis minimizes a cost function

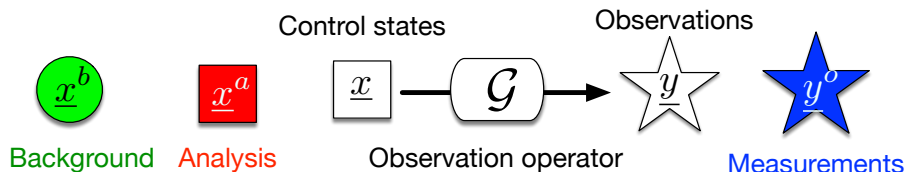
$$J(\underline{x}) = \frac{1}{2} (\underline{x} - \underline{x}^b)^T \underline{\underline{B}}^{-1} (\underline{x} - \underline{x}^b) + \frac{1}{2} [\underline{y}^o - \mathcal{G}(\underline{x})]^T \underline{\underline{R}}^{-1} [\underline{y}^o - \mathcal{G}(\underline{x})]$$



The metrics depends on the uncertainties

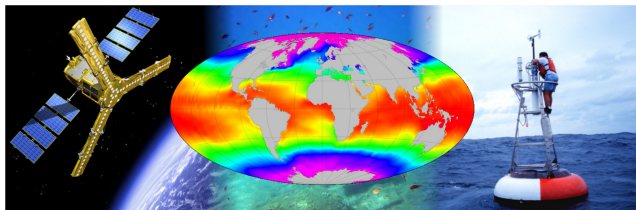
- The $N \times N$ “background error covariance matrix” $\underline{\underline{B}}$
- The $M \times M$ “observation error covariance matrix” $\underline{\underline{R}}$

Data assimilation for meteorology



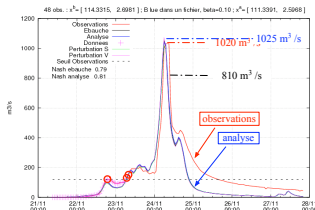
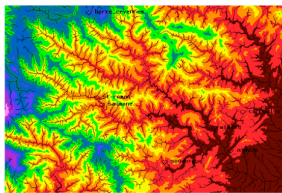
Control space	Observation operator	Observation space
<ul style="list-style-type: none"> • 3D fields: temperature, pressure, humidity, winds • Order ten millions of grid points 	Evolution model of the primitive equations of the atmosphere	<ul style="list-style-type: none"> • Satellite data: surface temperatures, radiances, cloud cover... • Order one million of observations

Data assimilation for oceanography



Control space	Observation operator	Observation space
<ul style="list-style-type: none">• 3D fields: temperature, pressure, salinity, currents• 2D fields: sea surface level• Order ten millions of grid points	Evolution model of the primitive equations of the ocean	<ul style="list-style-type: none">• Satellite data: sea surface temperature, altimetry• In-situ data: temperature, salinity...• Order thousands of observations

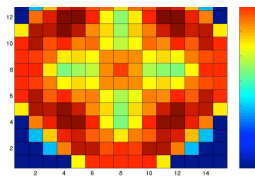
Data assimilation for hydrology



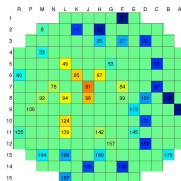
Control space	Observation operator	Observation space
<ul style="list-style-type: none"> ● 1D fields: water height, velocity, temperature ● Model parameters: friction, soil water content... ● Order thousands of grid points 	<p>Shallow water (Saint-Venant) equations</p>	<ul style="list-style-type: none"> ● Satellite data: altimetry ● In-situ data: water height, piezometric height ● Order hundred observations

Data assimilation for nuclear cores

Neutronic flux

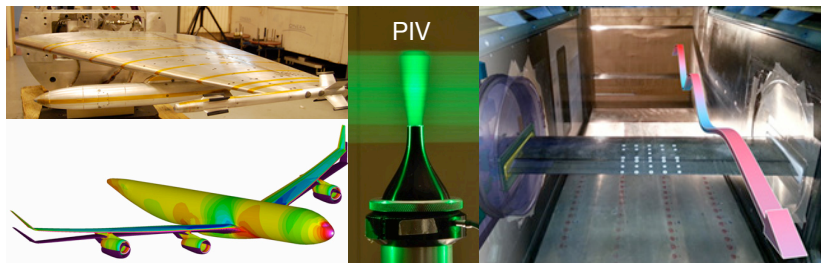


Activity measurements



Control space	Observation operator	Observation space
<ul style="list-style-type: none"> • 3D fields: neutronic flux, temperature, chemical concentration • Parameters: boundary reflections, diffusion coefficients • Order ten millions of grid points 	<p>Equilibrium model of the neutronic flux interacting with the heat transport fluid</p>	<ul style="list-style-type: none"> • Neutronic and thermo-hydraulic sensors • Thousand of observations

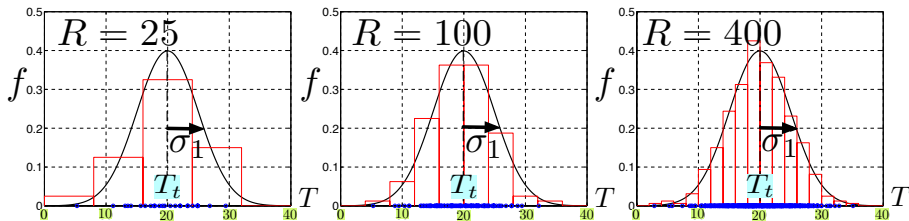
Data assimilation for aerodynamics



Control space	Observation operator	Observation space
<ul style="list-style-type: none">• 3D fields: velocity, pressure• Model parameters: wing friction• Order millions of grid points	Compressible Navier-Stokes equations	<ul style="list-style-type: none">• Aerodynamic flume measurements• PIV, pressure• Order hundred observations

Gaussian random variable

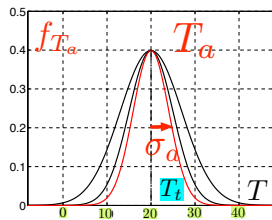
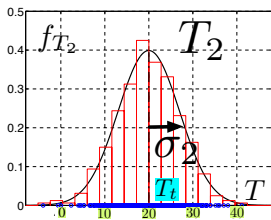
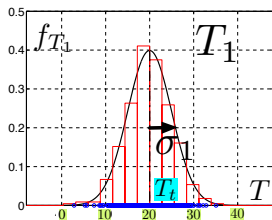
$$f_{T_1}(x) = \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left[-\frac{(T - T_t)^2}{2\sigma_1^2}\right] \implies \langle \Phi(x) \rangle = \int_{-\infty}^{\infty} \Phi(x) f_{T_1}(x) dx$$



$$\text{Mean: } T_t = \langle T_1 \rangle = \int_R T f_{T_1}(T) dT \sim \frac{1}{R} \sum_{r=1}^R T_1^{(r)}$$

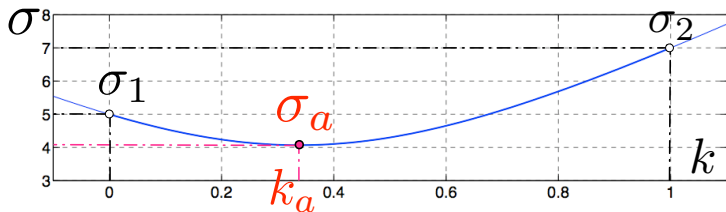
$$\text{Variance: } \sigma_1^2 = \langle T_1'^2 \rangle = \int_R (T - T_t)^2 f_{T_1}(T) dT$$

Analysis with two uncorrelated variables



New random variable $T_a = (1 - k)T_1 + kT_2$ with $\langle T_1' T_2' \rangle = 0$

Best choice: $T_a = \frac{C_1 T_1 + C_2 T_2}{C_1 + C_2}$ with $C_1 = \frac{1}{\sigma_1^2}$, $C_2 = \frac{1}{\sigma_2^2}$



$$C_a = \frac{1}{\sigma_a^2}$$

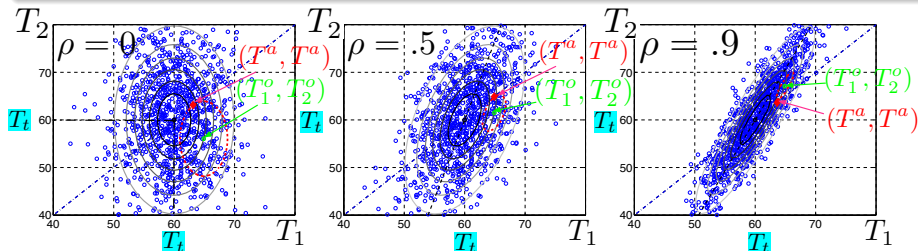
$$C_a = C_1 + C_2$$

Analysis with correlated variables

Analysis when $\langle T_1' T_2' \rangle = \rho \sigma_1 \sigma_2$

The analysis $T_a = (1 - k)T_1^o + kT_2^o$ minimizes the cost function:

$$J(T) = \frac{1}{2} (T_1^o - T, T_2^o - T) \begin{pmatrix} \sigma_1^2 & \rho \sigma_1 \sigma_2 \\ \rho \sigma_1 \sigma_2 & \sigma_2^2 \end{pmatrix}^{-1} \begin{pmatrix} T_1^o - T \\ T_2^o - T \end{pmatrix}$$



Link with the general formalism of data assimilation

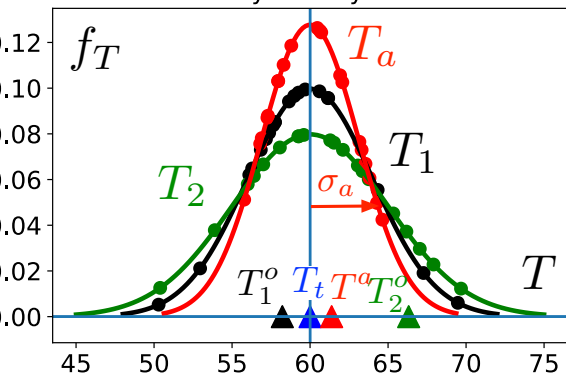
$\underline{x} = (T) \in \mathbf{R}$, $\underline{y}^o = (T_1^o, T_2^o)^T \in \mathbf{R}^2$ and $\mathcal{G}(\underline{x}) = (T, T)^T \in \mathbf{R}^2$.

There is no background and the 2×2 matrix is $\underline{\underline{R}}$.

Homework A: “What time is is?”, Exo 1.1

Hands-on from: <https://www.olivier-thual.fr/130202>

Probability density functions



- 1 Read program.
- 2 Launch program
- 3 Deactivate figures
- 4 Plot σ_a function of R
- 5 Count improvements
- 6 Compute scores
- 7 Go beyond

Data assimilation for engineers

Chapter 2: Generic cost function

Olivier THUAL

Toulouse INP-ENSEEIH/TFEE

Course:

Data Assimilation (ASID)

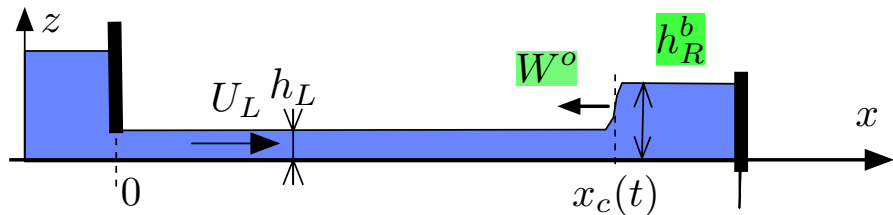
Teaching Unit:

Numerical Methods for Scientific Computing in Aerodynamics

How will the bore propagate?



A very simple geophysical model



Mass conservation:

$$h_L(U_L - W) = h_R(U_R - W) \quad \Rightarrow \quad W = \frac{-h_L U_L}{h_R - h_L}$$

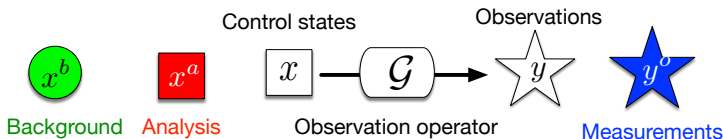
Notations $x \in \mathbb{R}$, $\mathcal{G} : \mathbb{R} \rightarrow \mathbb{R}$, $y \in \mathbb{R}$:

$$x = h_R, \quad y = W, \quad \text{and} \quad y = \mathcal{G}(x) = \frac{-h_L U_L}{x - h_L} = \frac{-q}{x - h_L}$$

Best estimate of the height $x = h_R$

Too much information in: $x = h_R$, $\mathcal{G}(x) = \frac{-q}{x-h_L}$, $y = W$

- We know approximatively $x = x^b$ with an uncertainty error σ_b
- We know approximatively $y = y^o$ with an uncertainty error σ_r
- What is the best estimate x^a of x knowing that $y = \mathcal{G}(x)$?

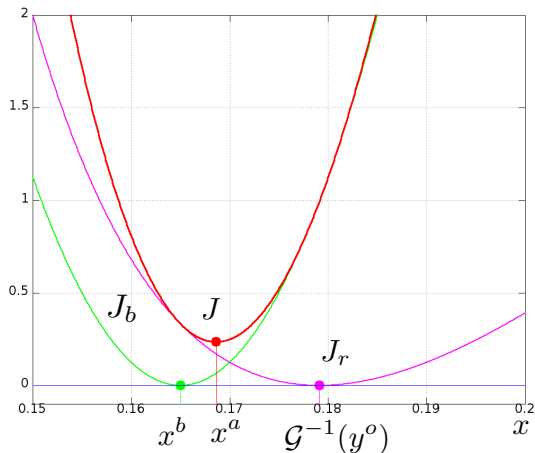


The analysis x^a minimizes the cost function:

$$J(x) = \frac{(x - x^b)^2}{2\sigma_b^2} + \frac{[y^o - \mathcal{G}(x)]^2}{2\sigma_r^2}$$

Plotting the cost function $J(x)$

$$J(x) = \frac{(x - x^b)^2}{2\sigma_b^2} + \frac{[y^o - \mathcal{G}(x)]^2}{2\sigma_r^2} = J_b(x) + J_r(x)$$



$$J_b(x) = \frac{(x - x^b)^2}{2\sigma_b^2}$$

$$J_r(x) = \frac{[y^o - \mathcal{G}(x)]^2}{2\sigma_r^2}$$

$$\text{with } \mathcal{G}(x) = \frac{-q}{x - h_L}$$

Incremental cost function

The cost function to minimize

$$J(x) = \frac{(x - x^b)^2}{2\sigma_r^2} + \frac{[y^o - \mathcal{G}(x)]^2}{2\sigma_r^2} \quad \text{with} \quad \mathcal{G}(x) = \frac{-q}{x - h_L}$$

Linearization of \mathcal{G} around x^b :

$$\mathcal{G}(x) = \mathcal{G}(x^b + \delta x) \sim \mathcal{G}(x^b) + G \delta x \quad \text{with} \quad \delta x = x - x^b$$

$$\text{One can compute} \quad G = \mathcal{G}'(x^b) = q/(x^b - h_L)^2$$

The incremental cost function J_{inc} is an approximation of J :

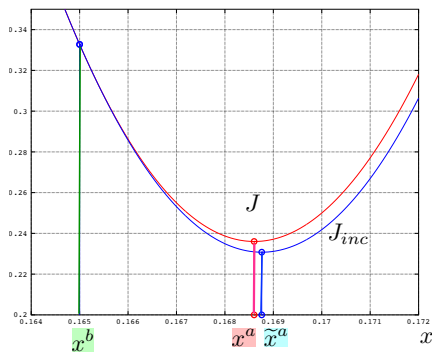
$$J_{inc}(x^b + \delta x) = \frac{(\delta x)^2}{2\sigma_b^2} + \frac{(d - G \delta x)^2}{2\sigma_r^2}$$

where $d = y^o - \mathcal{G}(x^b)$ is the “innovation”

Plotting the incremental cost function J_{inc}

Gradient of the function $J_{inc}(x^b + \delta x) = \frac{(\delta x)^2}{2\sigma_b^2} + \frac{(d - G\delta x)^2}{2\sigma_r^2}$:

$$J'_{inc}(x^b + \delta x) = \frac{\delta x}{\sigma_b^2} + G \frac{G\delta x - d}{\sigma_r^2}$$



J'_{inc} vanishes for: $\tilde{x}^a = x^b + \tilde{\delta x}^a$

with $\tilde{\delta x}^a = K d$ where:

Innovation: $d = y^o - \mathcal{G}(x_b)$

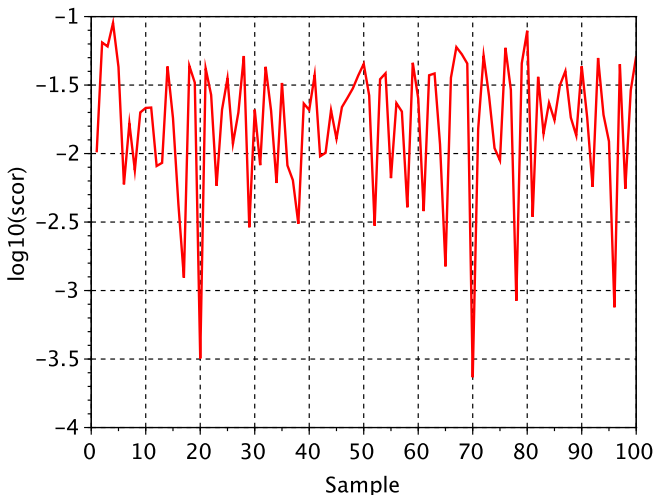
Gain: $K = \left(\frac{1}{\sigma_b^2} + \frac{G^2}{\sigma_r^2} \right)^{-1} \frac{G}{\sigma_r^2}$

or $K = \sigma_b^2 G (G^2 \sigma_b^2 + \sigma_r^2)^{-1}$

Exo 2.1: Example of score for $\mathcal{G}(x) = -q/(x - h_L)$

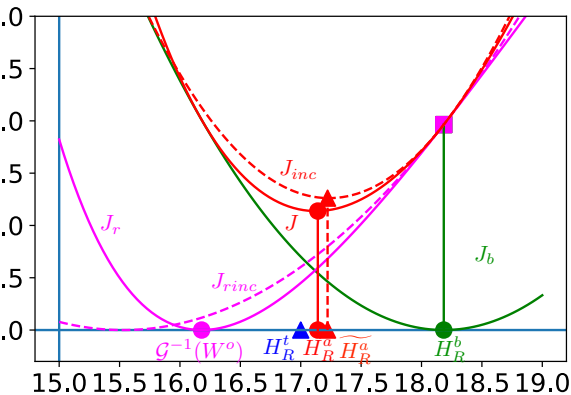
Experimental values

$q = 7$, $h_L = 5$, $h_R = x^t = 17$, $x^b = 18$, $\sigma_b = 1$, $\sigma_r = .03$



Homework B: "How will the bore propagate?", Exo 2.1

Hands-on from: <https://www.olivier-thual.fr/130202>



$$\mathcal{G}(x) = \frac{-q}{x - h_L}$$

$$G = \mathcal{G}'(x^b) = \frac{q}{(x - h_L)^2}$$

- 1 Read program.
- 2 Launch program
- 3 Compute scores
- 4 Replace \mathcal{G}
- 5 Count improvements
- 6 Go beyond

Basic linear algebra

Vectors are $1 \times N$ or $1 \times M$ matrices

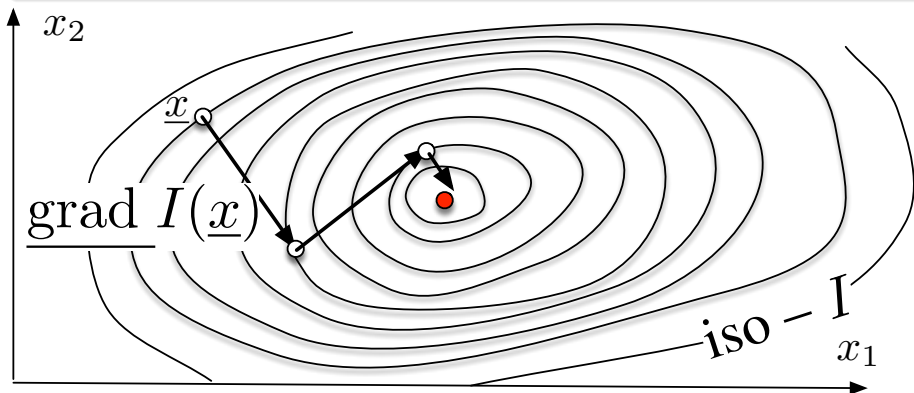
$$\underline{x} = \begin{pmatrix} x_1 \\ \dots \\ x_j \\ \dots \\ x_N \end{pmatrix}, \quad \underline{y} = \begin{pmatrix} y_1 \\ \dots \\ y_i \\ \dots \\ y_M \end{pmatrix}, \quad x \quad \begin{cases} \underline{x}^T = (x_1, \dots, x_j, \dots, x_N) \\ \underline{y}^T = (y_1, \dots, y_i, \dots, y_M) \end{cases}$$

Example of a $M \times N$ matrix considered as a linear operator:

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

Gradient of a scalar function

$$\text{grad } I(\underline{x}) = \left(\frac{\partial I}{\partial x_1}, \frac{\partial I}{\partial x_2}, \dots, \frac{\partial I}{\partial x_j}, \dots, \frac{\partial I}{\partial x_N} \right)^T$$



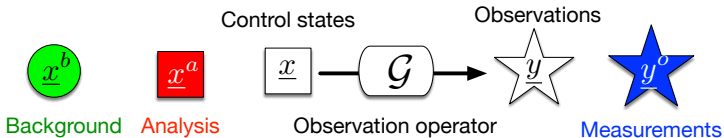
Examples of gradient computations

$\mathbf{l}(\underline{\mathbf{x}})$	$\text{grad } \mathbf{l}(\underline{\mathbf{x}})$
$\underline{\mathbf{u}}^T \underline{\mathbf{x}}$	$\underline{\mathbf{u}}$
$\underline{\mathbf{u}}^T \underline{\mathbf{M}} \underline{\mathbf{x}}$	$\underline{\mathbf{M}}^T \underline{\mathbf{u}}$
$\underline{\mathbf{x}}^T \underline{\mathbf{M}} \underline{\mathbf{x}}$	$(\underline{\mathbf{M}} + \underline{\mathbf{M}}^T) \underline{\mathbf{x}}$
$\underline{\mathbf{x}}^T \underline{\mathbf{H}}^T \underline{\mathbf{S}} \underline{\mathbf{H}} \underline{\mathbf{x}}$	$\underline{\mathbf{H}} (\underline{\mathbf{S}} + \underline{\mathbf{S}}^T) \underline{\mathbf{H}}^T \underline{\mathbf{x}}$

Generic cost function for data assimilation

Searching the analysis \underline{x}^a that minimize the cost function

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



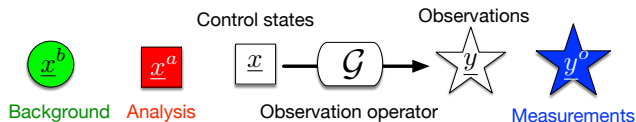
- Control space $\underline{x} \in \mathbf{R}^N$ and Observation space $\underline{y} \in \mathbf{R}^M$
- Background \underline{x}^b known with errors $\underline{\epsilon}^b$
- Measurement \underline{y}^o known with errors $\underline{\epsilon}^o$
- Covariance background error matrix $\underline{\underline{B}}$ with $B_{ij} = \langle \epsilon_i^b \epsilon_j^b \rangle$
- Covariance observation error matrix $\underline{\underline{R}}$ with $R_{ij} = \langle \epsilon_i^o \epsilon_j^o \rangle$

Bayesian approach of data assimilation

Density probability function for \underline{x}^b and \underline{y}^o independantly

$$f^b(\underline{x}) = K^b e^{-J_b(\underline{x})} \quad \text{with} \quad J_b(\underline{x}) = \frac{1}{2} (\underline{x} - \underline{x}^b)^T \underline{\underline{B}}^{-1} (\underline{x} - \underline{x}^b)$$

$$f^o(\underline{y}) = K^o e^{-J_r(\underline{y})} \quad \text{with} \quad J_r(\underline{y}) = \frac{1}{2} (\underline{y}^o - \underline{y})^T \underline{\underline{R}}^{-1} (\underline{y}^o - \underline{y})$$



Density probability function for \underline{x} knowing \underline{y}^o

$$f^{b/o}(\underline{x}) = f^b(\underline{x}) f^o[\mathcal{G}(\underline{x})] = K^o K^b e^{-J(\underline{x})} \quad \text{with}$$

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

Incremental cost function through a linearization of \mathcal{G}

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

Linearization of \mathcal{G} around the background \underline{x}^b :

$$\mathcal{G}(\underline{x}^b + \underline{\delta x}) \approx \mathcal{G}(\underline{x}^b) + \underline{\underline{G}} \underline{\delta x}$$

Incremental cost function

$$J_{inc}(\underline{x}^b + \underline{\delta x}) = \frac{1}{2} \underline{\delta x}^T \underline{\underline{B}}^{-1} \underline{\delta x} + \frac{1}{2} (\underline{d} - \underline{\underline{G}} \underline{\delta x})^T \underline{\underline{R}}^{-1} (\underline{d} - \underline{\underline{G}} \underline{\delta x})$$

where $\underline{d} = \underline{y}^o - \mathcal{G}(\underline{x}^b)$ is the innovation vector

Minimum of the incremental cost function

Knowing the innovation vector $\underline{d} = \underline{y}^o - \underline{\mathcal{G}}(\underline{x})$:

$$J_{inc}(\underline{x}^b + \underline{\delta x}) = \frac{1}{2} \underline{\delta x}^T \underline{\underline{B}}^{-1} \underline{\delta x} + \frac{1}{2} (\underline{d} - \underline{\underline{G}} \underline{\delta x})^T \underline{\underline{R}}^{-1} (\underline{d} - \underline{\underline{G}} \underline{\delta x})$$

Gradient of the incremental cost function

$$\text{grad } J_{inc}(\underline{x}^b + \underline{\delta x}) = \underline{\underline{B}}^{-1} \underline{\delta x} - \underline{\underline{G}}^T \underline{\underline{R}}^{-1} [\underline{d} - \underline{\underline{G}} \underline{\delta x}]$$

The minimum $\underline{\tilde{x}}^a$ is found through $\text{grad } J_{inc}(\underline{x}^a) = \underline{0}$:

$$\underline{\tilde{x}}^a = \underline{x}^b + \underline{\underline{K}} \underline{d}$$

where the gain matrix is: $\underline{\underline{K}} = \left(\underline{\underline{B}}^{-1} + \underline{\underline{G}}^T \underline{\underline{R}}^{-1} \underline{\underline{G}} \right)^{-1} \underline{\underline{G}}^T \underline{\underline{R}}^{-1}$

and is also equal to (SMW identity): $\underline{\underline{K}} = \underline{\underline{B}} \underline{\underline{G}}^T \left(\underline{\underline{G}} \underline{\underline{B}} \underline{\underline{G}}^T + \underline{\underline{R}} \right)^{-1}$

Sherman-Morrison-Woodbury identity

Two expressions of the gain matrix:

$$\underline{K} = (\underline{G}^T \underline{R}^{-1} \underline{G} + \underline{B}^{-1})^{-1} \underline{G}^T \underline{R}^{-1} = \underline{B} \underline{G}^T (\underline{G} \underline{B} \underline{G}^T + \underline{R})^{-1}$$

- If $M < N$: inverse the $M \times M$ matrix $(\underline{G} \underline{B} \underline{G}^T + \underline{R})$
- If $N < M$: inverse the $N \times N$ matrix $(\underline{G}^T \underline{R}^{-1} \underline{G} + \underline{B}^{-1})$

$$\begin{aligned} \underline{G}^T \underline{R}^{-1} (\underline{G} \underline{B} \underline{G}^T + \underline{R}) &= \underline{G}^T \underline{R}^{-1} \underline{G} \underline{B} \underline{G}^T + \underline{G}^T = (\underline{G}^T \underline{R}^{-1} \underline{G} \underline{B} + \underline{I}) \underline{G}^T \\ &= (\underline{G}^T \underline{R}^{-1} \underline{G} \underline{B} + \underline{B}^{-1} \underline{B}) \underline{G}^T = (\underline{G}^T \underline{R}^{-1} \underline{G} + \underline{B}^{-1}) \underline{B} \underline{G}^T \end{aligned}$$

$$(\underline{G}^T \underline{R}^{-1} \underline{G} + \underline{B}^{-1})^{-1} \underline{G}^T \underline{R}^{-1} = \underline{B} \underline{G}^T (\underline{G} \underline{B} \underline{G}^T + \underline{R})^{-1}$$

Data assimilation for engineers

Chapter 3: Time dependent models

Olivier THUAL

Toulouse INP-ENSEEIH/MFEE

Course:

Data Assimilation (ASID)

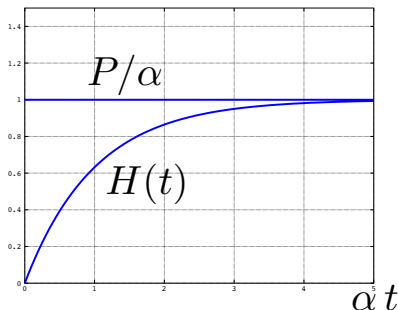
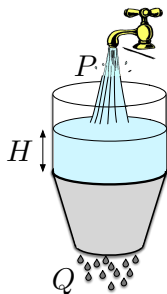
Teaching Unit:

Numerical Methods for Scientific Computing in Aerodynamics

Will the water overflow?

Reservoir model

$$\frac{dH(t)}{dt} = -\alpha H(t) + P \quad \text{with} \quad H(0) = 0$$



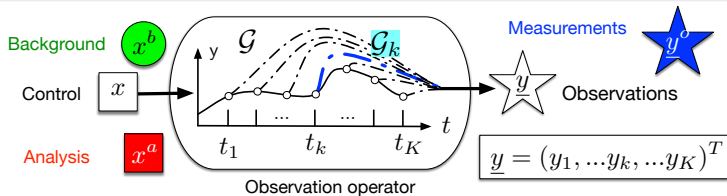
Exact solution

$$H(t) = (P/\alpha) [1 - \exp(-\alpha t)]$$

Assimilation of α

Measurements of $y_k = H(t_k)$ to determine $x = \alpha$

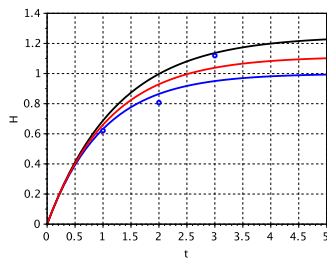
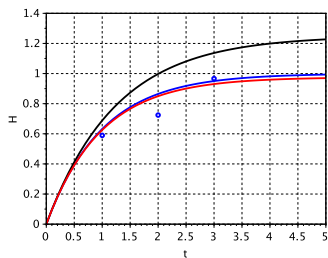
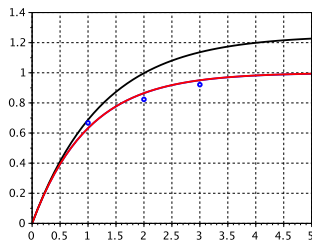
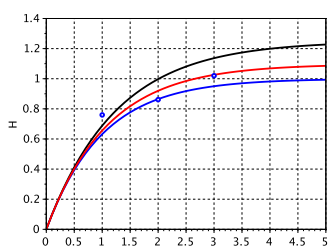
- $\underline{y}^o = (y_1^o, \dots, y_k^o, \dots, y_K^o)^T$
- $\mathcal{G} = (\mathcal{G}_1^T, \dots, \mathcal{G}_k^T, \dots, \mathcal{G}_K^T)^T$ with $\mathcal{G}_k(\alpha) = (P/\alpha)[1 - \exp(-\alpha t_k)]$



Cost function

$$J(\alpha) = \frac{1}{2} \frac{(\alpha - \alpha^b)^2}{\sigma_b^2} + \sum_{k=1}^K \frac{[y_k^o - \mathcal{G}_k(\alpha)]^2}{2\sigma_r^2}$$

Incremental method for the assimilation of α

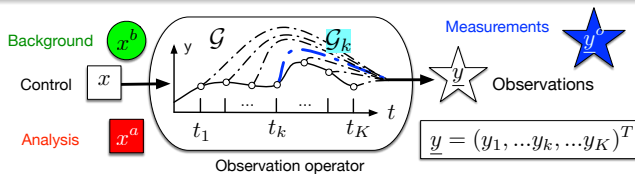


Red: analysis, Blue: true state, Black: background

Assimilation of (α, P)

Measurements of $y_k = H(t_k)$ to determine $\underline{x} = (\alpha, P)$

- $\underline{y}^o = (y_1^o, \dots, y_k^o, \dots, y_K^o)^T$
- $\underline{\mathcal{G}} = (\mathcal{G}_1^T, \dots, \mathcal{G}_k^T, \dots, \mathcal{G}_K^T)^T$ with $\mathcal{G}_k(\alpha, P) = (P/\alpha)[1 - \exp(-\alpha t_k)]$



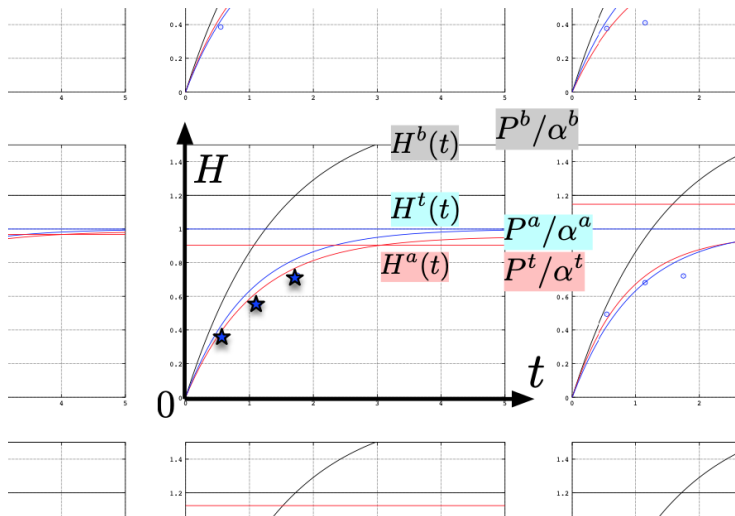
Cost function:

$$J(\alpha, P) = \frac{1}{2}(\alpha - \alpha^b, P - P^b) \underline{\underline{B}}^{-1} \begin{pmatrix} \alpha - \alpha^b \\ P - P^b \end{pmatrix} + \sum_{k=1}^K \frac{[y_k^o - \mathcal{G}_k(\alpha, P)]^2}{2\sigma_r^2}$$

Covariance background error matrix:

$$\underline{\underline{B}} = \begin{pmatrix} \sigma_\alpha^2 & \rho \sigma_\alpha \sigma_P \\ \rho \sigma_\alpha \sigma_P & \sigma_P^2 \end{pmatrix}$$

Incremental method for the assimilation of (α, P)



Red: analysis, Blue:true state, Black: background

The 4D-Var data assimilation

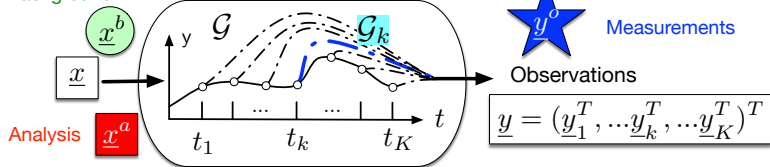
$\underline{x} \in \mathbb{R}^N$ with measurements $\underline{y}^o \in \mathbb{R}^M$

- $\underline{y} = (\underline{y}_1^T, \dots, \underline{y}_k^T, \dots, \underline{y}_K^T)^T$ with $\underline{y}_k = \mathcal{G}_k(\underline{x})$
- $\underline{y}^o = (\underline{y}_1^{oT}, \dots, \underline{y}_k^{oT}, \dots, \underline{y}_K^{oT})^T$

Observation operator:

$$\mathcal{G} = (\mathcal{G}_1^T, \dots, \mathcal{G}_k^T, \dots, \mathcal{G}_K^T)^T$$

Background



Measurements

Observations

$$\underline{y} = (\underline{y}_1^T, \dots, \underline{y}_k^T, \dots, \underline{y}_K^T)^T$$

$$J(\underline{x}) = \frac{1}{2} (\underline{x} - \underline{x}^b)^T \underline{\underline{B}}^{-1} (\underline{x} - \underline{x}^b)$$

$$+ \frac{1}{2} \sum_{k=1}^K \left[\underline{y}_k^o - \mathcal{G}_k(\underline{x}) \right]^T \underline{\underline{R}}_k^{-1} \left[\underline{y}_k^o - \mathcal{G}_k(\underline{x}) \right]$$

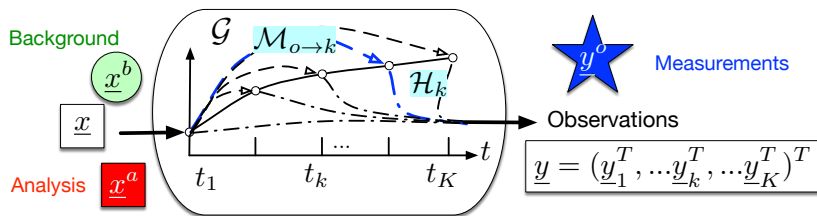
Covariance observation error matrix:

$$\underline{\underline{R}} = \text{diag} (\underline{\underline{R}}_1, \dots, \underline{\underline{R}}_k, \dots, \underline{\underline{R}}_K)$$

Nonlinear Cauchy problem

Dynamical system

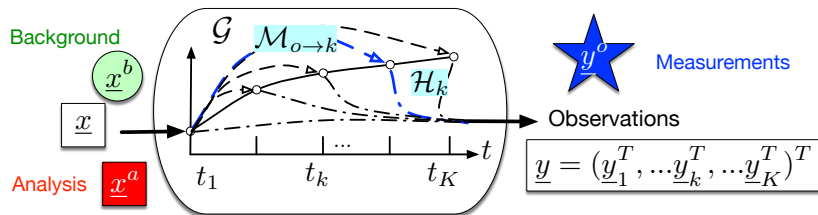
$$\frac{d\underline{X}}{dt} = \mathcal{L}(t, \underline{X}) \quad \text{with} \quad \underline{X}(0) = \underline{x}$$



Control of the initial conditions $\underline{x} = \underline{X}(0)$:

- $\underline{X}(t_k) = \mathcal{M}_{l \rightarrow k} [\underline{X}(t_l)] \implies \underline{X}(t_k) = \mathcal{M}_{0 \rightarrow k}(\underline{x})$.
- $\underline{y}_k = \mathcal{H}_k \mathcal{M}_{0 \rightarrow k}(\underline{x})$ with \mathcal{H}_k nonlinear or linear
- $\mathcal{G} = (\mathcal{G}_1^T, \dots, \mathcal{G}_k^T, \dots, \mathcal{G}_K^T)^T$ with $\mathcal{G}_k = \mathcal{H}_k \mathcal{M}_{0 \rightarrow k}$

The generic 4D-Var cost function



Assuming $\underline{\underline{R}} = \text{diag} (\underline{\underline{R}}_1, \dots, \underline{\underline{R}}_k, \dots, \underline{\underline{R}}_K)$

$$J(\underline{x}) = \frac{1}{2} (\underline{x} - \underline{x}^b)^T \underline{\underline{B}}^{-1} (\underline{x} - \underline{x}^b) + \frac{1}{2} \sum_{k=1}^K \left[\underline{y}_k^o - \mathcal{H}_k \mathcal{M}_{0 \rightarrow k}(\underline{x}) \right]^T \underline{\underline{R}}_k^{-1} \left[\underline{y}_k^o - \mathcal{H}_k \mathcal{M}_{0 \rightarrow k}(\underline{x}) \right].$$

Linear tangent model

Linearization around the background vector \underline{x}^b :

$$\mathcal{H}_k \mathcal{M}_{0 \rightarrow k}(\underline{x}^b + \delta \underline{x}) \sim \mathcal{H}_k \mathcal{M}_{0 \rightarrow k}(\underline{x}^b) + \underline{\underline{H}}_k \underline{\underline{M}}_{0 \rightarrow k} \delta \underline{x}$$

- $\underline{\underline{H}}_k$ is the linearized of \mathcal{H}_k around $\mathcal{M}_{0 \rightarrow k}(\underline{x}^b)$
- $\underline{\underline{M}}_{0 \rightarrow k}$ is the linearized of $\mathcal{M}_{0 \rightarrow k}$ around \underline{x}^b

Linear tangent dynamical system and tangent model $\underline{\underline{M}}_{0 \rightarrow k}$

$$\frac{\partial \underline{U}}{\partial t} = \underline{\underline{L}} \left[\underline{X}^b(t) \right] \underline{U} \quad \text{with} \quad \underline{U}(0) = \delta \underline{x}$$

- $\underline{X}^b(t)$ is the solution of $\frac{d\underline{X}}{dt} = \mathcal{L}(t, \underline{X})$ with $\underline{X}^b(0) = \underline{x}^b$
- $\underline{\underline{L}}$ is such that $\mathcal{L}[\underline{X}^b(t) + \underline{U}] \sim \mathcal{L}[\underline{X}^b(t)] + \underline{\underline{L}} [\underline{X}^b(t)] \underline{U}$
- $\underline{U}(t_k) = \underline{\underline{M}}_{l \rightarrow k} \underline{U}(t_l)$ and thus $\underline{U}(t_k) = \underline{\underline{M}}_{0 \rightarrow k} \delta \underline{x}$

Magnus expansion: $\underline{\underline{M}}_{l \rightarrow k}$ “like” $\exp \left\{ \int_{t_l}^{t_k} \underline{\underline{L}} [\underline{X}^b(t)] dt \right\}$ (add Lie brackets)

The incremental 4D-Var function

Incremental cost function with $\underline{\underline{R}} = \text{diag} (\underline{\underline{R}}_1, \dots, \underline{\underline{R}}_k, \dots, \underline{\underline{R}}_K)$:

$$J_{inc}(\underline{x}^b + \underline{\delta x}) = \frac{1}{2} \underline{\delta x}^T \underline{\underline{B}}^{-1} \underline{\delta x} + \frac{1}{2} \sum_{k=1}^K \left[\underline{d}_k - \underline{\underline{H}}_k \underline{\underline{M}}_{0 \rightarrow k} \underline{\delta x} \right]^T \underline{\underline{R}}_k^{-1} \left[\underline{d}_k - \underline{\underline{H}}_k \underline{\underline{M}}_{0 \rightarrow k} \underline{\delta x} \right]$$

where $\underline{d}_k = \underline{y}_k^o - \mathcal{H}_k \mathcal{M}_{0 \rightarrow k}(\underline{x}^b)$ are the innovation subvectors

- $\underline{\underline{G}} = (\underline{\mathcal{G}}_1^T, \dots, \underline{\mathcal{G}}_k^T, \dots, \underline{\mathcal{G}}_K^T)^T$ with $\underline{\mathcal{G}}_k = \mathcal{H}_k \mathcal{M}_{0 \rightarrow k}$
- Innovation vector: $\underline{d} = \underline{y}^o - \underline{\underline{G}}(\underline{x})$
- $\underline{\underline{G}} = \text{diag} \left(\underline{\underline{H}}_1 \underline{\underline{M}}_{0 \rightarrow 1}, \dots, \underline{\underline{H}}_k \underline{\underline{M}}_{0 \rightarrow k}, \dots, \underline{\underline{H}}_K \underline{\underline{M}}_{0 \rightarrow K} \right)$

$$\tilde{\underline{x}}^a = \underline{x}^b + \underline{\underline{K}} \underline{d} \quad \text{with} \quad \underline{\underline{K}} = \left(\underline{\underline{B}}^{-1} + \underline{\underline{G}}^T \underline{\underline{R}}^{-1} \underline{\underline{G}} \right)^{-1} \underline{\underline{G}}^T \underline{\underline{R}}^{-1}$$

Adjoint model $\underline{\underline{M}}_{k \rightarrow 0}^* = \underline{\underline{M}}_{0 \rightarrow k}^T$

- $\underline{\underline{K}} = (\underline{\underline{B}}^{-1} + \underline{\underline{G}}^T \underline{\underline{R}}^{-1} \underline{\underline{G}})^{-1} \underline{\underline{G}}^T \underline{\underline{R}}^{-1}$
- $\underline{\underline{G}} = \text{diag} \left(\underline{\underline{H}}_1 \underline{\underline{M}}_{0 \rightarrow 1}, \dots, \underline{\underline{H}}_k \underline{\underline{M}}_{0 \rightarrow k}, \dots, \underline{\underline{H}}_K \underline{\underline{M}}_{0 \rightarrow K} \right)$
- $\underline{\underline{G}}^T = \text{diag} \left(\underline{\underline{M}}_{0 \rightarrow 1}^T \underline{\underline{H}}_1^T, \dots, \underline{\underline{M}}_{0 \rightarrow k}^T \underline{\underline{H}}_k^T, \dots, \underline{\underline{M}}_{0 \rightarrow K}^T \underline{\underline{H}}_K^T \right)$

Definition of the adjoint model $\underline{\underline{M}}^*$

New Cauchy problem: $\frac{\partial \tilde{\underline{X}}}{\partial \tau} = \underline{\underline{L}}^T \left[\underline{\underline{X}}^b(t_K - \tau) \right] \tilde{\underline{X}}$ with $\tilde{\underline{X}}(0) = \tilde{\underline{x}}$

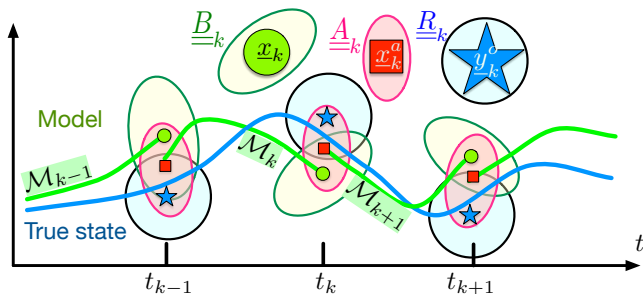
$\tilde{\underline{X}}(\tau_l) = \underline{\underline{M}}_{k \rightarrow l}^* \tilde{\underline{X}}(\tau_k)$ with $\underline{\underline{M}}_{k \rightarrow l}^*$ “like” $\exp \left\{ \int_{\tau_k}^{\tau_l} \underline{\underline{L}}^T \left[\underline{\underline{X}}^b(t_K - \tau) \right] d\tau \right\}$

$\underline{\underline{M}}_{l \rightarrow k}$ “like” $\exp \left\{ \int_{\tau_l}^{\tau_k} \underline{\underline{L}} \left[\underline{\underline{X}}^b(t) \right] dt \right\}$ (Magnus) $\implies \underline{\underline{M}}_{k \rightarrow l}^* = \underline{\underline{M}}_{l \rightarrow k}^T$

Extended Kalman filter

Initialization:

- $\underline{x}_0 = \underline{x}$
- $\underline{A}_0 = \underline{B}_0$



We denote $\mathcal{M}_k = \mathcal{M}_{k-1 \rightarrow k}$ and $\underline{M}_k = \underline{M}_{k-1 \rightarrow k}$:

$$\underline{x}_k = \mathcal{M}_k(\underline{x}_{k-1}^a) \quad \text{and} \quad \underline{B}_k = \underline{M}_k \underline{A}_{k-1} \underline{M}_k^T + \underline{Q}_k$$

$$\underline{x}_k^a = \underline{x}_k + \underline{K}_k \underline{d}_k \quad \text{with}$$

$$\underline{d}_k = \underline{y}_k^o - \mathcal{H}_k(\underline{x}_k) \quad \text{and} \quad \underline{K}_k = \underline{B}_k \underline{H}_k^T (\underline{H}_k \underline{B}_k \underline{H}_k + \underline{R}_k)^{-1},$$

$$\underline{A}_k = (\underline{I} - \underline{K}_k \underline{H}_k) \underline{B}_k$$

where \underline{Q}_k is the covariance model error matrix.

Data assimilation for engineers

Chapter 4: Application projects

Olivier THUAL

Toulouse INP-ENSEEIH/ MFEE

Course:

Data Assimilation (ASID)

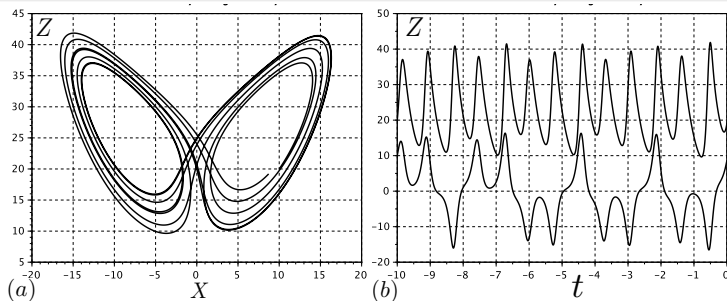
Teaching Unit:

Numerical Methods for Scientific Computing in Aerodynamics

Project: data assimilation for the Lorenz model

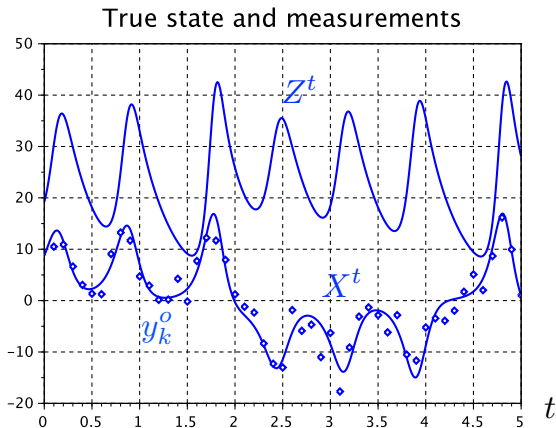
The Lorenz model: $\sigma = 10$, $\rho = 28$ and $\beta = 8/3$

$$\frac{dX}{dt} = -\sigma X + \sigma Y, \quad \frac{dY}{dt} = \rho X - Y - XZ, \quad \frac{dZ}{dt} = XY - \beta Z$$



Transitory: $t \in [-10, 0]$ with $(X_s, Y_s, Z_s) = (10, 15, 20)$ for $t = -10$

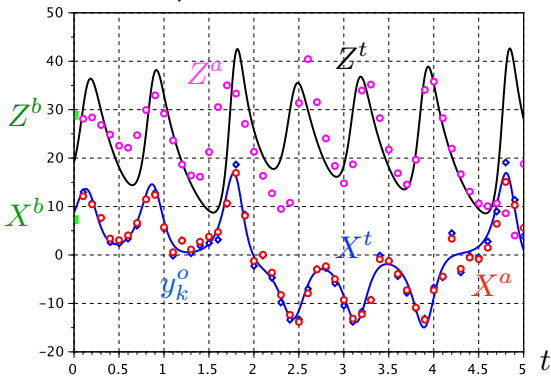
Twin experiments



- True trajectory: $[X^t(t), Y^t(t), Z^t(t)]$ for $t \in [0, t_f]$ with $t_f = 5$
- Background: $(X^b, Y^b, Z^b) = (X_0, Y_0, Z_0) + (\epsilon_X, \epsilon_Y, \epsilon_Z)$
- Measurements: $y_k^o = X^t(t_k) + \epsilon^o$ for $k = 1, \dots, 50$, where $t_k = k \tau$

Chain of BLUE (3D-Var) with \underline{B} constant

Analyzed states for method=1



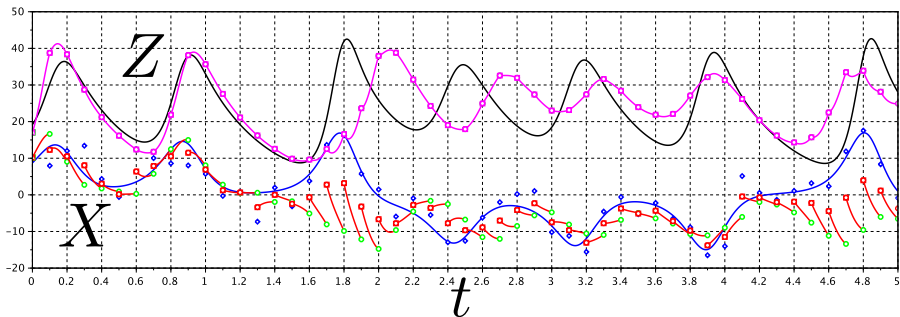
Let $\underline{B} = \sigma_b^2 \underline{I}$ constant:

$$J_k(\underline{x}) = \frac{1}{2}(\underline{x} - \underline{x}_k^b)^T \underline{B}^{-1} (\underline{x} - \underline{x}_k^b)^T + \frac{1}{2\sigma_r^2} (y_k^o - X)^2$$

$$\underline{x}_k^a = \underline{x}_k^b + \underline{K} d_k = \left[X_k^b + \frac{\sigma_b^2}{\sigma_b^2 + \sigma_r^2} (y_k^o - X_k^b), Y_k^b, Z_k^b \right]^T$$

where $d_k = y_k^o - \underline{G} \underline{x}_k^b = y_k^o - X_k^b$ is the innovation

Chain of BLUE (3D-Var) with $\underline{B} = \underline{A}$



$$J(\underline{x}) = \frac{1}{2}(\underline{x} - \underline{x}_k^b)^T \underline{A}_{k-1}^{-1} (\underline{x} - \underline{x}_k^b)^T + \frac{1}{2\sigma_r^2} (y_k^o - X)^2$$

$$\underline{A}_k = (\underline{I} - \underline{K} \underline{G}) \underline{A}_{k-1} \quad \text{for } k = 1, \dots, K \quad \text{with} \quad \underline{A}_0 = \underline{B}$$

Kalman filter with no error model

$$J(\underline{x}) = \frac{1}{2}(\underline{x} - \underline{x}_k^b)^T \underline{\underline{B}}_k^{-1} (\underline{x} - \underline{x}_k^b)^T + \frac{1}{2\sigma_r^2}(y_k^o - X)^2,$$

$$\underline{\underline{A}}_{k-1} = (\underline{I} - \underline{\underline{K}} \underline{\underline{G}}) \underline{\underline{B}}_{k-1} \quad \text{and} \quad \underline{\underline{B}}_k = \underline{\underline{M}}_k \underline{\underline{A}}_{k-1} \underline{\underline{M}}_k^T$$

Tangent model $\underline{\underline{M}}_k$:

$$\frac{d}{dt}(dX) = -\sigma dX + \sigma dY$$

$$\frac{d}{dt}(dY) = \rho dX - dY - dX Z^b(t) - X^b(t) dZ$$

$$\frac{d}{dt}(dZ) = dX Y^b(t) + X^b(t) dY - \beta dZ$$

$\underline{\underline{M}}_k$ obtained by integrating from t_{k-1} to t_k with the respective initial conditions $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$ for (dX, dY, dZ)

Ensemble Kalman filter

Gain matrix: $\underline{\underline{K}}_k = \underline{\underline{B}}_k \underline{\underline{H}}_k^T (\underline{\underline{H}}_k \underline{\underline{B}}_k \underline{\underline{H}}_k^T + \underline{\underline{R}}_k)^{-1}$

R trajectories: $\underline{x}_k^{(r)} = \mathcal{M}_k \left(\underline{x}_{k-1}^{(r)} \right)$ for $r = 1, \dots, R$

Analyses: $\underline{x}_k^{a(r)} = \underline{x}_k^{(r)} + \underline{\underline{K}}_k \underline{d}_k^{(r)}$ with $\underline{y}_k^{(r)} = \mathcal{H}_k \left(\underline{x}_k^{(r)} \right)$ for $r = 1, \dots, R$

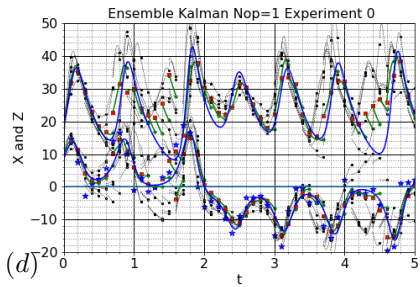
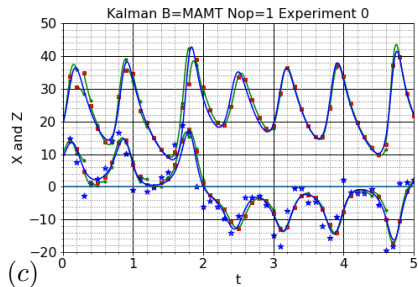
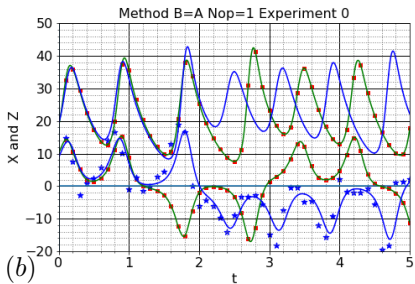
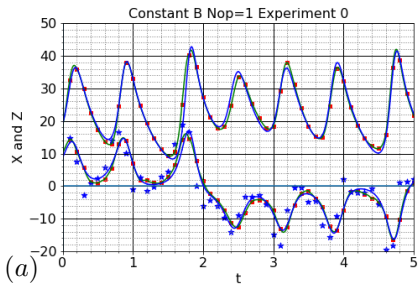
Approximations of the gain matrix

$$\underline{\underline{B}}_k \underline{\underline{H}}_k^T \sim \underline{\underline{A}}_k^{BH^T} = \frac{1}{R-1} \sum_{r=1}^R \left(\underline{x}_k^{(r)} - \underline{x}_k \right) \left(\underline{y}_k^{(r)} - \underline{y}_k \right)^T$$

$$\underline{\underline{H}}_k \underline{\underline{B}}_k \underline{\underline{H}}_k^T \sim \underline{\underline{A}}_k^{HBH^T} = \frac{1}{R-1} \sum_{r=1}^R \left(\underline{y}_k^{(r)} - \underline{y}_k \right) \left(\underline{y}_k^{(r)} - \underline{y}_k \right)^T$$

$$\text{with } \underline{x}_k = \frac{1}{R} \sum_{r=1}^R \underline{x}_k^{(r)} \quad \text{and} \quad \underline{y}_k = \frac{1}{R} \sum_{r=1}^R \mathcal{H}_k \left(\underline{x}_k^{(r)} \right)$$

Comparison of four methods



Homework C : Methods for the Lorenz model (1/2)

Lorenz model

The Lorenz model shows a chaotic dynamics. Write a Python program to simulate the evolution of this model:

- Parameters ($\sigma = 10, \rho = 28, \beta = 10/3$),
- Initial conditions $(x, y, z) = (10, 15, 20)$ for $t = -10$.

Twin experiments

- From the $t = 0$ state of the previous transitory simulation, compute the “true trajectory” for $t \in [0, t_f]$ with $t_f = 5$.
- Consider the values $x(t_i)$ for $t_i = i\tau$ with $i = \{1, 2, \dots, K\}$, with $K = 50$, and $\tau = t_f/K$, perturbed by a random gaussian error of standard deviation $\sigma_r = 1$, as the measurement vector \underline{y}^o .

Homework C : Methods for the Lorenz model (2/2)

Jupyter notebook from:

<https://www.olivier-thual.fr/130202/co/genintro.html>

Comparison of data assimilation methods

At first, program the methods for Section 2.2 of Chapter 4:

- Chain of BLUE with \underline{B} constant
- Chain of BLUE with $\underline{B} = \underline{A}$
- Kalman filter with no error model
- Ensemble Kalman filter

Explorations beyond the basic subject such as:

- New observation operators: $\mathcal{H}(\underline{x}) = Z$, $\mathcal{H}(\underline{x}) = \underline{x}^2 \dots$
- Incremental 4DVar, Kalman filter with model error...
- Choose another chaotic model: Lorenz 84, Belousov-Zhabotinsky...
- ...