



I MAGNET BRAZIL

Fundamentals of data assimilation

Alexandre Fournier



Équipe de Géomagnétisme
Institut de Physique du Globe de Paris
Sorbonne Paris Cité, Université Paris Diderot,
INSU/CNRS (UMR 7154), F-75005 Paris, France
fournier@ipgp.fr
www.ipgp.fr/~fournier



2 Most of these working notes (chapters 1 to 3) have been made possible thanks to the generous
3 input of my friend and colleague Emmanuel Cosme. Emmanuel is a physical oceanographer
4 working at the Université of Grenoble, www-meom.hmg.inpg.fr/Web/pages-perso/Cosme/.
5 On his webpage you will find much more material related to the foundations and applications
6 of data assimilation.

7 I designed the labs specifically for the workshop, and as a result they were tested that week
8 for the first time. They rely on the the `matlab` software, and can be run as well using its
9 open-source clone, `octave` (on my ubuntu 9.04 linux computer, I noticed a substantial drop
10 of performance using `octave`, though). I deserve the full credit for any mistake in the notes,
11 labs, codes, . . .

12 I would like to thank the crew of teaching assistants who kindly volunteered to help me out
13 and made my life a whole lot easier during this nice week in Búzios: Hagay Amit, Nicolas
14 Gillet, Andy Jackson, Saulo Martins, Sabrina Sanchez, and Jakub Velínský. Thanks also to
15 the students for their active participation and the keen interest they showed.



16 Additions since the workshop (as of *August30*, 2011) :

- 17 • some general references in the introductory chapter
- 18 • an appendix where the discrete adjoint equation is derived



19 “One day he came home with a little bird in his hand and I said to him: ‘Look, it’s just like
20 you. It flies around a lot, but it’s no good for anything. It’s a garrincha (little bird)’. The name
stuck for the rest of his life.” Rosa dos Santos, Garrincha’s elder sister

21 Contents

22	I Working notes	5
23	1 Introduction	7
24	1.1 What is data assimilation?	7
25	1.2 A scalar example	8
26	1.3 Notations	10
27	1.4 Useful references	13
28	2 Stochastic estimation	14
29	2.1 Basics of probability and statistics	14
30	2.2 The two pillars of estimation theory	16
31	2.3 Optimal estimates	17
32	2.4 The BLUE	18
33	2.5 The Gaussian case	19
34	3 The Kalman filter	21
35	3.1 Introduction	21
36	3.2 The Kalman filter algorithm	22
37	3.3 Implementation issues	24
38	4 Variational assimilation	29
39	II Labs	33
40	1 The vibrating string: forward modelling	34
41	1.1 Introduction	34
42	1.2 The CFL criterion	36
43	1.3 Resolution / dispersion	36

44	1.4	Generating observations	38
45	2	An optimal interpolation scheme applied to the vibrating string	41
46	2.1	Statistical ingredients	41
47	2.2	Algorithm	42
48	2.3	Points to address	45
49	3	The Kalman filter applied to the vibrating string	47
50	3.1	Statistical ingredients	47
51	3.2	Algorithm	48
52	3.3	Points to address	51
53	III	Appendix	56
54	A	Derivation of the discrete adjoint equation	57

55

Part I

56

Working notes

58 Chapter 1

59 Introduction

60 1.1 What is data assimilation?

61 The basic purpose of data assimilation is to combine different sources of information in order
62 to produce the best possible estimate of the state of a system. These sources generally
63 consist of observations of the system and of physical laws describing its behaviour, often
64 represented in the form of a numerical model. Why not simply use observations? First
65 because observations are often too sparse or partial in geophysics. Some extra information
66 is needed to interpolate the information contained in the observations to unobserved regions
67 or quantities. A numerical model naturally performs this task. Second, because observations
68 are contaminated by errors (they are noised). Combining (by means of the model) several
69 noised data can be an efficient way of to filter out at least part of the noise and to provide a
70 more accurate estimate (“accuracies are added”, see below).

71 The problem of data assimilation can be tackled using different mathematical approaches:
72 signal processing, control theory, estimation theory, ... Stochastic methods, such as the pop-
73 ular Kalman filter, are based on estimation theory. On the other hand, variational methods
74 (3D-Var, 4D-Var...) are rooted in control theory.

75 The historical development of data assimilation for geophysical systems can hardly be dis-
76 connected from meteorology. Data assimilation is indeed a mandatory step if one wishes to
77 provide a weather prediction system with a good initialization (an initial condition), and un-
78 til the early nineteen-nineties data assimilation was mostly used for this purpose. Today, its
79 application is generalized to many other fields (atmospheric chemistry, oceanic biochemistry,
80 glaciology, physical oceanography, geomagnetism, stellar magnetism, seismology...), and for
81 a variety of purposes :

- 82 • the estimation of the trajectory of a system to study its variability (reanalyses)
- 83 • the identification of systematic errors in numerical models
- 84 • the estimation of unobserved field variables (e.g. the magnetic field inside Earth’s core)
- 85 • the estimation of parameters (e.g. a structural Earth model in seismology)
- 86 • the optimization of observation networks

1.2 A scalar example

Following for instance Ghil and Malanotte-Rizzoli (1991), assume we have two distinct measurements, $y_1 = 1$ and $y_2 = 2$, of the same unknown quantity x . What estimation of its true value can we make ?

1.2.1 First approach

We seek x which minimizes $(x - 1)^2 + (x - 2)^2$, and we find the estimate $\hat{x} = 3/2 = 1.5$ (this is the least-squares solution). This solution has the following problems:

- it is sensitive to any change of units. If $y_1 = 1$ is a measurement of x and $y_2 = 4$ is a measurement of $2x$, then minimizing $(x - 1)^2 + (2x - 4)^2$ leads to $\hat{x} = 9/5 = 1.8$.
- it does not reflect the quality of the various measurements.

1.2.2 Reformulation in a statistical framework

We define

$$Y_i = x + \epsilon_i, \tag{1.1}$$

where the observation errors ϵ_i satisfy the following hypotheses

- $E(\epsilon_i) = 0$ (unbiased measurements)
- $\text{Var}(\epsilon_i) = \sigma_i^2$ (accuracy is known)
- $\text{Covar}(\epsilon_1, \epsilon_2) = 0$, i.e. $E(\epsilon_1\epsilon_2) = 0$, errors are independent.

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

- linear: $\hat{X} = \alpha_1 Y_1 + \alpha_2 Y_2$
- unbiased: $E(\hat{X}) = x$
- of minimum variance: $\text{Var}(\hat{X})$ minimal (optimal accuracy)

This estimator is called the **BLUE**: Best Linear Unbiased Estimator. To compute the α_i we use the unbiased hypothesis

$$E(\hat{X}) = x = (\alpha_1 + \alpha_2)x + \alpha_1 E(\epsilon_1) + \alpha_2 E(\epsilon_2) = (\alpha_1 + \alpha_2)x, \tag{1.2}$$

so that $\alpha_1 + \alpha_2 = 1$, or $\alpha_2 = 1 - \alpha_1$. Next we compute the variance of \hat{X} .

$$\begin{aligned} \text{Var}(\hat{X}) &= E\left[\left(\hat{X} - x\right)^2\right] = E\left[\left(\alpha_1\epsilon_1 + \alpha_2\epsilon_2\right)^2\right] \\ &= \alpha_1^2 E(\epsilon_1^2) + 2\alpha_1\alpha_2 E(\epsilon_1\epsilon_2) + \alpha_2^2 E(\epsilon_2^2) \\ &= \alpha_1^2\sigma_1^2 + \alpha_2^2\sigma_2^2 \\ &= \alpha_1^2\sigma_1^2 + (1 - \alpha_1)^2\sigma_2^2. \end{aligned}$$

112 Our estimator \hat{X} has to minimize this quantity. Computing α_1 such that

$$113 \quad \frac{d}{d\alpha_1} \text{Var}(\hat{X}) = 0 \quad (1.3)$$

114 yields

$$115 \quad \alpha_1 = \frac{\sigma_2^2}{\sigma_2^2 + \sigma_1^2}. \quad (1.4)$$

116 It follows that

$$117 \quad \hat{X} = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} y_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} y_2. \quad (1.5)$$

118 Note that we get the same result if we try to minimize the functional

$$119 \quad \mathcal{J}(x) = \frac{1}{2} \left[\frac{(x - y_1)^2}{\sigma_1^2} + \frac{(x - y_2)^2}{\sigma_2^2} \right]. \quad (1.6)$$

120 Comments:

- 121 • This statistical approach solves the problem of sensitivity to units and it incorporates
- 122 measurement accuracies.
- 123 • The accuracy of the estimator is given by the second derivative of \mathcal{J}

$$124 \quad \left. \frac{d^2 \mathcal{J}}{dx^2} \right|_{x=\hat{X}} = \frac{1}{\text{Var}(\hat{X})} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}, \quad (1.7)$$

125 so that “accuracies are added”.

- 126 • If we consider that $y_1 = x^b$ is a first guess of x (with standard deviation $\sigma_b = \sigma_1$)
- 127 and $y_2 = y$ is an additional observation (with std dev $\sigma = \sigma_2$), then we can rearrange
- 128 Eq. (1.5) as

$$129 \quad \hat{X} = x^b + \frac{\sigma_b^2}{\sigma^2 + \sigma_b^2} (y - x^b). \quad (1.8)$$

130 The quantity $y - x^b$ is called the **innovation**. It contains the additional information
131 provided by y with respect to x^b .

132 1.2.3 Data assimilation methods

133 There are two classes of methods

- 134 • statistical methods: direct computation of the BLUE thanks to algebraic computations
- 135 (the Kalman filter);

136 • variational methods: minimization of the functional J (4DVar).

137 Shared properties:

138 • they provide the same result (in the linear case);

139 • their optimality can only be demonstrated in the linear case;

140 Shared difficulties:

141 • accounting for non-linearities

142 • dealing with large problems

143 • error statistics are required but sometimes only poorly known

144 [Courtier \(1997\)](#) provides a concise and elegant discussion of the two classes of methods and
145 discusses their equivalence.

146 1.3 Notations

147 There exists some sort of standard notations, summarized by [Ide et al. \(1997\)](#).

148 • \mathbf{x} state vector

149 • \mathbf{x}^t true state

150 • \mathbf{x}^b background state

151 • \mathbf{x}^a analyzed state

152 Superscripts denote vector types, subscripts refer to space or time. In the following: unless
153 otherwise noted, all vectors will be column vectors. If \mathbf{a} and \mathbf{b} are two column vectors of
154 equal size n , with the superscript T denoting transposition, then

$$\mathbf{a}^T \mathbf{b} \quad \text{is their scalar product} = \sum a_i b_i, \quad (1.9)$$

$$\mathbf{a} \mathbf{b}^T \quad \text{is a matrix of coefficients } a_i b_j, (i, j) \in \{1, \dots, n\}^2. \quad (1.10)$$

155 1.3.1 Discretization and true state

156 Most of the time, our goal will be to estimate as accurately as possible a geophysical field that
157 varies continuously in space and time. This real, continuous (and possibly multivariate) field
158 is denoted by \mathcal{x} . For the one-dimensional (1D) vibrating string problem we will be dealing
159 with in our labs, \mathcal{x} comprises the transverse displacement $y(x, t)$ and velocity $\partial_t y(x, t)$ along
160 the vibrating string.

161 Numerical models are often used for the estimation. Numerical models operate in a discrete
162 world and only handle discrete representations of physical fields. Therefore we will try to

163 estimate a projection of the real state \boldsymbol{x} onto a discrete space. Let $\mathbf{\Pi}$ denote the associated
 164 projector, and \mathbf{x}^t be the projection of \boldsymbol{x}

$$165 \quad \mathbf{x}^t = \mathbf{\Pi}(\boldsymbol{x}). \quad (1.11)$$

166 \mathbf{x}^t is called the true state (see above); this is the state we wish to estimate in practice. In our
 167 labs, the true state will consist of the value of the displacement and velocity fields, discretized
 168 on a finite difference grid.

169 In a data assimilation problem, one deals with **dynamical** models that compute the time
 170 evolution of the simulated state. Let \boldsymbol{x}_i and \boldsymbol{x}_{i+1} be the real (continuous) states at two
 171 consecutive observation times, i being a time index. These two states are related by a causal
 172 link (the physical model)

$$173 \quad \boldsymbol{x}_{i+1} = g(\boldsymbol{x}_i). \quad (1.12)$$

174 Projecting this equality into the discrete world, we get

$$175 \quad \mathbf{x}_{i+1}^t = \mathbf{\Pi}[g(\boldsymbol{x}_i)]. \quad (1.13)$$

176 The dynamical model g is not strictly known, even though we hopefully know most of the
 177 physics involved in it (in our vibrating string problem, our model will be exactly known).
 178 This physics is represented in the discrete world by our numerical model \mathcal{M} , which operates
 179 on discrete states such as \mathbf{x}^t . Introducing this model into Eq. (1.13), we get

$$180 \quad \mathbf{x}_{i+1}^t = \mathcal{M}_{i,i+1}(\mathbf{x}_i^t) + \boldsymbol{\eta}_{i,i+1}, \quad (1.14)$$

181 in which

$$182 \quad \boldsymbol{\eta}_{i+1} = \mathbf{\Pi}[g(\boldsymbol{x}_i)] - \mathcal{M}_{i,i+1}(\mathbf{x}_i^t). \quad (1.15)$$

183 The **model error** $\boldsymbol{\eta}_{i+1}$ term accounts for the errors in the numerical models (e.g. misrep-
 184 resentation of some physical processes) and for the errors due to the discretization. The
 185 covariance matrix \mathbf{Q}_{i+1} of the model error is given by

$$186 \quad \mathbf{Q}_{i+1} = \text{Covar}(\boldsymbol{\eta}_{i+1}) = \text{E} \left[(\boldsymbol{\eta}_{i+1} - \langle \boldsymbol{\eta}_{i+1} \rangle) (\boldsymbol{\eta}_{i+1} - \langle \boldsymbol{\eta}_{i+1} \rangle)^T \right], \quad (1.16)$$

187 where $\langle \boldsymbol{\eta}_{i+1} \rangle = \text{E}(\boldsymbol{\eta}_{i+1})$ is the average error.

188 1.3.2 Observations

189 The real, continuous field \boldsymbol{x} results in a signal \boldsymbol{y} in the space of observations. This involves a
 190 mapping \boldsymbol{h}

$$191 \quad \boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}). \quad (1.17)$$

192 Despite its simplicity, this equation can not be used in practice. First, we do not have access
 193 to the real y : the observed field \mathbf{y}^o is contaminated with measurement errors, denoted by $\boldsymbol{\epsilon}^\mu$.
 194 Accordingly,

$$195 \quad \mathbf{y}^o = \mathcal{h}(\boldsymbol{x}) + \boldsymbol{\epsilon}^\mu. \quad (1.18)$$

196 Second, \mathcal{h} , which represents the physics of the measurement process (which might be exactly
 197 known), is a continuous mapping. In practice, this physics is represented by a numerical
 198 operator \mathcal{H} , which is applied to the discrete state we wish to estimate, \mathbf{x}^t . Incorporating \mathcal{H}
 199 and $\boldsymbol{\Pi}$ in Eq. (1.18) yields

$$200 \quad \mathbf{y}^o = \mathcal{H}(\mathbf{x}^t) + \underbrace{\mathcal{h}(\boldsymbol{x}) - \mathcal{H}[\boldsymbol{\Pi}(\boldsymbol{x})]}_{\boldsymbol{\epsilon}^r} + \boldsymbol{\epsilon}^\mu, \quad (1.19)$$

201 where $\boldsymbol{\epsilon}^r$ is often termed the error of representativeness (Lorenç, 1986), which includes the
 202 errors related to the representation of the physics in \mathcal{H} and those errors due to the projection
 203 $\boldsymbol{\Pi}$ of the real state \boldsymbol{x} onto the discrete state space (due for instance to numerical interpolation).
 204 The sum of the measurement error and the error of representativeness is the **observation**
 205 **error**

$$206 \quad \boldsymbol{\epsilon}^o = \boldsymbol{\epsilon}^\mu + \boldsymbol{\epsilon}^r. \quad (1.20)$$

207 This allows us to write the final form of the equation relating the discrete true state \mathbf{x}^t and
 208 the observations

$$209 \quad \mathbf{y}^o = \mathcal{H}(\mathbf{x}^t) + \boldsymbol{\epsilon}^o. \quad (1.21)$$

210 The covariance matrix of the observation error $\boldsymbol{\epsilon}^o$ is defined by

$$211 \quad \mathbf{R} = \text{Covar}(\boldsymbol{\epsilon}^o) = \text{E} \left[(\boldsymbol{\epsilon}^o - \langle \boldsymbol{\epsilon}^o \rangle) (\boldsymbol{\epsilon}^o - \langle \boldsymbol{\epsilon}^o \rangle)^T \right]. \quad (1.22)$$

212 In our labs, we will be dealing with synthetic data and we will artificially introduce observation
 213 errors $\boldsymbol{\epsilon}^o$ (the statistics of which we will assume to be Gaussian).

214 1.3.3 A priori (background) information

215 It can be that we have some a priori knowledge of the state \mathbf{x}^t , under the form of a vector \mathbf{x}^b
 216 having the same dimension as \mathbf{x}^t . This is the **background state**. Following a similar logic,
 217 the background error is defined as

$$218 \quad \boldsymbol{\epsilon}^b = \mathbf{x}^b - \mathbf{x}^t. \quad (1.23)$$

219 Often the estimate of the background state comes from a model simulation. In this case, the
 220 background is a **forecast** and is rather denoted by \mathbf{x}^f , with forecast error $\boldsymbol{\epsilon}^f$.

221 The covariance \mathbf{P}^b of the background error is given by

$$222 \quad \mathbf{P}^b = \text{Covar}(\boldsymbol{\epsilon}^b) = \text{E} \left[(\boldsymbol{\epsilon}^b - \langle \boldsymbol{\epsilon}^b \rangle) (\boldsymbol{\epsilon}^b - \langle \boldsymbol{\epsilon}^b \rangle)^T \right]. \quad (1.24)$$

1.3.4 Analysis

The result of the assimilation process is often called the analysis, and is denoted by \mathbf{x}^a . The analysis error is defined by

$$\boldsymbol{\epsilon}^a = \mathbf{x}^a - \mathbf{x}^t, \quad (1.25)$$

while the covariance matrix of the analysis error $\boldsymbol{\epsilon}^a$ is defined by

$$\mathbf{P}^a = \text{Covar}(\boldsymbol{\epsilon}^a) = \text{E} \left[(\boldsymbol{\epsilon}^a - \langle \boldsymbol{\epsilon}^a \rangle) (\boldsymbol{\epsilon}^a - \langle \boldsymbol{\epsilon}^a \rangle)^T \right]. \quad (1.26)$$

An important comment: the problem is entirely set-up once the physical model and the observations have been chosen, and the covariances (and possibly the background) defined. All the physics has been introduced at this stage. The remaining part (the production of the analysis) is technical.

1.4 Useful references

At this stage it might be timely to provide the reader with general references on data assimilation. My favorite book on the topic is “Discrete Inverse and State Estimation Problems”, by Wunsch (2006), which provides a very personal and powerful account of adjoint methods and their application in geophysical fluid dynamics (oceanography). In her book entitled “Atmospheric Modelling, Data Assimilation and Predictability”, E. Kalnay (2003) has two comprehensive and very well-written chapters on the basics and applications of data assimilation techniques to atmospheric dynamics. Last, but not least, Evensen (2009) provides a very complete treatment of data assimilation techniques, with a strong and useful emphasis on the basics and applications of the ensemble Kalman filter he invented (we will briefly touch on this in Sect. 3.3.4.2).

For a start, I would highly recommend the review paper by Talagrand (1997), “Assimilation of observations, an introduction” which provides an extremely concise and well-written overview of the topic.

In addition, if you are looking for references related to the geophysical inverse problem in general, Parker (1994) and Tarantola (2005) provide two very personal, insightful, and sometimes contradictory views on how we should go about making inference on the Earth based on a finite number of noisy observations and on physical laws governing its behaviour.

252 Chapter 2

253 Stochastic estimation

254 2.1 Basics of probability and statistics

255 2.1.1 Probability

256 2.1.1.1 Random experiment

257 A random experiment is mathematically described by

- 258 • the set Ω of all possible outcomes of an experiment, the result of which can not be
259 perfectly anticipated;
- 260 • the subsets of Ω , called events;
- 261 • a probability function, P : a numerical expression of a state of knowledge. P is such
262 that, for any disjoint events A and B ,

$$0 \leq P(A) \leq 1, \tag{2.1}$$

$$P(\Omega) = 1, \tag{2.2}$$

$$P(A \cup B) = P(A) + P(B). \tag{2.3}$$

263 Here, \cup means .OR. In the next paragraph, \cap will mean .AND.

264 2.1.1.2 Conditional probability

265 When two events A and B are not independent, knowing that B has occurred changes our
266 state of knowledge on A . This writes

$$267 P(A|B) = \frac{P(A \cap B)}{P(B)}. \tag{2.4}$$

2.1.2 Real random variables

The outcome of a random experiment is called a random variable. A random variable can be an integer (the number of tries scored by the French rugby team, whose games often resemble random experiments), or a real number (e.g. the lifetime of a Buzz Lightyear action figure).

2.1.2.1 Probability density function

For a real random variable x , being equal to a given number is not strictly speaking an event. Only the inclusion into an interval is an event. This defines the **probability density function**, also known as pdf

$$P(a \leq x \leq b) = \int_a^b p(x) dx. \quad (2.5)$$

2.1.2.2 Joint and conditional pdf

If x and y are two real random variables, $p(x, y)$ is the joint pdf of x and y . The conditional pdf $p(x|y)$ writes

$$p(x|y) = \frac{p(x, y)}{p(y)}. \quad (2.6)$$

2.1.2.3 Expectation and variance

A pdf is seldom known completely. In most instances, only some of its properties are determined and handled. The two main properties are the expectation and the variance. The expectation of a random variable x , characterized by a pdf p is given by

$$E(x) = \langle x \rangle = \int_{-\infty}^{+\infty} xp(x) dx. \quad (2.7)$$

The variance is given by

$$\text{Var}(x) = E \left[(x - \langle x \rangle)^2 \right] = \int_{-\infty}^{+\infty} (x - \langle x \rangle)^2 p(x) dx. \quad (2.8)$$

The standard deviation σ is **the square root of the variance**.

2.1.2.4 The Gaussian distribution

The random variable x has a Gaussian (or normal) distribution with parameters μ and σ^2 , denoted by $x \sim \mathcal{N}(\mu, \sigma^2)$ when

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[-\frac{(x - \mu)^2}{2\sigma^2} \right]. \quad (2.9)$$

This Gaussian pdf has the following properties

- 294 • the parameters μ and σ^2 are its expectation and variance, respectively;
- 295 • If $x_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$ and $x_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$ are two independent variables, then $x_1 + x_2$ is
296 also Gaussian and $x_1 + x_2 \sim \mathcal{N}(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)$;
- 297 • if $a \in \mathbb{R}$ and $x \sim \mathcal{N}(\mu, \sigma^2)$, then $ax \sim \mathcal{N}(a\mu, a^2\sigma^2)$.

298 2.1.3 Real random vectors

299 Real random vectors are vectors whose components are real random variables. The pdf of a
300 vector is the joint pdf of its components.

301 2.1.3.1 Expectation and variance

302 The expectation vector is the vector of the expected values of the components. The second
303 moment of the distribution is the covariance matrix. If \mathbf{x} denotes the random vector, the
304 covariance matrix is defined by

$$305 \quad \mathbf{P} = \mathbb{E} \left[(\mathbf{x} - \langle \mathbf{x} \rangle) (\mathbf{x} - \langle \mathbf{x} \rangle)^T \right]. \quad (2.10)$$

306 A covariance matrix is symmetric positive definite. The terms appearing on its diagonal are
307 the variances of the vector components. The off-diagonal terms are covariances. If x_i and x_j
308 denote two different components of \mathbf{x} , their covariance is

$$309 \quad P_{ij} = \text{Covar}(x_i, x_j) = \mathbb{E} \left[(x_i - \langle x_i \rangle) (x_j - \langle x_j \rangle)^T \right] \quad (2.11)$$

310 and their correlation is

$$311 \quad \rho(x_i, x_j) = \frac{\text{Covar}(x_i, x_j)}{\sqrt{\text{Var}(x_i) \text{Var}(x_j)}}. \quad (2.12)$$

312 2.1.3.2 The multivariate Gaussian distribution

313 The random vector \mathbf{x} of size n has a Gaussian (or normal) distribution with parameters $\boldsymbol{\mu}$
314 and \mathbf{P} , denoted by $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{P})$, if

$$315 \quad p(\mathbf{x}) = \frac{1}{(2\pi)^{n/2} (\det \mathbf{P})^{1/2}} \exp \left\{ -\frac{1}{2} \left[(\mathbf{x} - \boldsymbol{\mu})^T \mathbf{P}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right] \right\}. \quad (2.13)$$

316 Here $\boldsymbol{\mu}$ and \mathbf{P} are the expectation and the covariance matrix of \mathbf{x} , respectively; $\det \mathbf{P}$ is the
317 determinant of \mathbf{P} . The components of \mathbf{x} are said to be jointly Gaussian.

318 2.2 The two pillars of estimation theory

319 If one has to remember only two formulas from this section, these are

320 1. Bayes' theorem

$$321 \quad p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{y})}. \quad (2.14)$$

322 2. The marginalization rule

$$323 \quad p(\mathbf{y}) = \int p(\mathbf{x}, \mathbf{y}) d\mathbf{x} = \int p(\mathbf{y}|\mathbf{x})p(\mathbf{x}) d\mathbf{x}. \quad (2.15)$$

324 where

- 325 • $p(\mathbf{y}|\mathbf{x})$ is the measurement model (or likelihood);
- 326 • $p(\mathbf{x})$ is the prior distribution;
- 327 • $p(\mathbf{y})$ is the marginal distribution (or evidence).

328 2.3 Optimal estimates

329 The optimal estimate of the random vector \mathbf{x} given the observation \mathbf{y} is the vector of values
 330 which best reflects what a realisation of \mathbf{x} can be, having \mathbf{y} at hand. Optimality is subjective,
 331 and several criteria can be proposed in order to define it. For the sake of illustration we
 332 present three such estimators below (although the rest of the material discussed this week
 333 will only have to do with the minimum variance estimator).

334 2.3.1 Minimum variance estimation

335 The estimate we seek is such that the spread around it is minimal. The measure of the spread
 336 is the variance. If $p(\mathbf{x}|\mathbf{y})$ is the pdf of \mathbf{x} , having \mathbf{y}^o at hand, the minimum variance estimate
 337 $\hat{\mathbf{x}}_{mv}$ is the solution of

$$338 \quad \nabla_{\hat{\mathbf{x}}} \mathcal{J}(\hat{\mathbf{x}}) = \mathbf{0}, \quad (2.16)$$

339 where

$$340 \quad \mathcal{J}(\hat{\mathbf{x}}) = \int (\mathbf{x} - \hat{\mathbf{x}})^T (\mathbf{x} - \hat{\mathbf{x}}) p(\mathbf{x}|\mathbf{y}) d\mathbf{x} \quad (2.17)$$

341 and the gradient is defined as

$$342 \quad \nabla_{\hat{\mathbf{x}}} = [\partial_{\hat{x}_1}, \dots, \partial_{\hat{x}_i}, \dots, \partial_{\hat{x}_n}] \quad (2.18)$$

343 (This is a row vector.) We can show that the solution is the expectation of the pdf, that is

$$344 \quad \hat{\mathbf{x}}_{mv} = \mathbf{E}[\mathbf{x}|\mathbf{y}]. \quad (2.19)$$

345 2.3.2 Maximum a posteriori estimation

346 The estimate is defined at the most probable vector of \mathbf{x} given \mathbf{y} , i.e., the vector that maximizes
347 the conditional pdf $p(\mathbf{x}|\mathbf{y})$. $\hat{\mathbf{x}}_{map}$ is such that

$$348 \quad \left. \frac{\partial p(\mathbf{x}|\mathbf{y})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{map}} = \mathbf{0}. \quad (2.20)$$

349 With a Gaussian pdf, the minimum variance and the maximum a posteriori estimators are
350 the same.

351 2.3.3 Maximum likelihood estimation

352 The estimate is defined as the most probable vector of \mathbf{y} given \mathbf{x} , i.e., the vector which
353 maximizes the conditional pdf $p(\mathbf{y}|\mathbf{x})$. $\hat{\mathbf{x}}_{ml}$ is such that

$$354 \quad \left. \frac{\partial p(\mathbf{y}|\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{ml}} = \mathbf{0}. \quad (2.21)$$

355 The ML estimator can be interpreted as the MAP estimator without any prior information
356 $p(\mathbf{x})$.

357 2.4 The best linear unbiased estimate (BLUE)

358 We now return to the BLUE, which we already introduced based on the simple scalar example
359 of Sect. 1.2. We aim at estimating the true state \mathbf{x}^t of a system, assuming that a background
360 estimate \mathbf{x}^b and partial observations \mathbf{y}^o are given. We assume that these two pieces of infor-
361 mation are unbiased and that their uncertainties are known in the form of covariance matrices
362 \mathbf{P}^b and \mathbf{R} , respectively (recall paragraphs 1.3.3 and 1.3.2). The observation operator \mathcal{H} is
363 assumed linear (denoted by \mathbf{H}). All together we have the following pieces of information

$$\mathbf{H}, \text{ such that } \mathbf{y}^o = \mathbf{H}\mathbf{x}^t + \boldsymbol{\epsilon}^o, \quad (2.22)$$

$$\mathbf{x}^b = \langle \mathbf{x}^t \rangle, \quad (2.23)$$

$$\mathbf{P}^b = \langle \boldsymbol{\epsilon}^b \boldsymbol{\epsilon}^{bT} \rangle, \quad (2.24)$$

$$\langle \boldsymbol{\epsilon}^o \rangle = \mathbf{0}, \quad (2.25)$$

$$\mathbf{R} = \langle \boldsymbol{\epsilon}^o \boldsymbol{\epsilon}^{oT} \rangle. \quad (2.26)$$

364 The best estimate (or analysis) \mathbf{x}^a is sought as a linear combination of the background estimate
365 and the observation

$$366 \quad \mathbf{x}^a = \mathbf{A}\mathbf{x}^b + \mathbf{K}\mathbf{y}^o, \quad (2.27)$$

367 where \mathbf{A} and \mathbf{K} are to be determined in order to make the estimation optimal (you can think
368 of them as the generalization of the coefficients α_1 and α_2 in the simple scalar example of

369 Sect. 1.2). How do we define optimality? Given the information at hand, a wise choice is
 370 to seek an unbiased estimate, with minimum variance. Reintroducing $\boldsymbol{\epsilon}^a = \mathbf{x}^a - \mathbf{x}^t$ we seek
 371 (\mathbf{A}, \mathbf{K}) such that

$$\mathbf{E}(\boldsymbol{\epsilon}^a) = \mathbf{0}, \quad (2.28)$$

$$\text{Tr}(\mathbf{P}^a) \text{ minimum}, \quad (2.29)$$

372 where $\text{Tr}(\cdot)$ denotes the trace (sum of the diagonal elements, here the variance of each com-
 373 ponent of \mathbf{x}^a). One can show that

$$\begin{aligned} \mathbf{A} &= \mathbf{I} - \mathbf{K}\mathbf{H}, \\ \mathbf{K} &= \mathbf{P}^b \mathbf{H}^T (\mathbf{H}\mathbf{P}^b \mathbf{H}^T + \mathbf{R})^{-1}, \end{aligned}$$

374 in which \mathbf{K} is called the Kalman gain matrix ¹.

375 The a posteriori error covariance matrix \mathbf{P}^a can also be computed. The final form of the
 376 update equations writes

$$\mathbf{K} = \mathbf{P}^b \mathbf{H}^T (\mathbf{H}\mathbf{P}^b \mathbf{H}^T + \mathbf{R})^{-1}, \quad (2.30)$$

$$\mathbf{x}^a = \mathbf{x}^b + \mathbf{K} (\mathbf{y}^o - \mathbf{H}\mathbf{x}^b), \quad (2.31)$$

$$\mathbf{P}^a = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}^b, \quad (2.32)$$

377 where \mathbf{I} is the identity matrix.

378 These equations constitute the best linear unbiased estimate (BLUE) equations, under the
 379 constraint of minimum variance. They are the backbone of sequential data assimilation
 380 methods (soon to come).

381 2.5 The Gaussian case

382 If we know that both the prior and observation pieces of information are adequately rep-
 383 resented by Gaussian pdfs, we may apply Bayes' theorem to compute the a posteriori pdf.
 384 With

$$\mathbf{x}^t \sim \mathcal{N}(\mathbf{x}^b, \mathbf{P}^b),$$

$$p(\mathbf{x}^t) = \frac{1}{(2\pi)^{n/2} \det \mathbf{P}^{b1/2}} \exp \left\{ -\frac{1}{2} \left[(\mathbf{x}^t - \mathbf{x}^b)^T \mathbf{P}^{b-1} (\mathbf{x}^t - \mathbf{x}^b) \right] \right\}, \quad (2.33)$$

$$\mathbf{y}^o \sim \mathcal{N}(\mathbf{H}\mathbf{x}^b, \mathbf{R}),$$

$$p(\mathbf{y}^o | \mathbf{x}^t) = \frac{1}{(2\pi)^{n/2} \det \mathbf{R}^{1/2}} \exp \left\{ -\frac{1}{2} \left[(\mathbf{y}^o - \mathbf{H}\mathbf{x}^t)^T \mathbf{R}^{-1} (\mathbf{y}^o - \mathbf{H}\mathbf{x}^t) \right] \right\}. \quad (2.34)$$

¹ We have assumed so far that we were dealing with real-valued variables. When dealing with complex-valued fields, everything holds, provided one replaces the transpose operator T by a transpose conjugate operator, often denoted by a dagger \dagger .

385 Bayes' theorem provides us with the a posteriori pdf

$$386 \quad p(\mathbf{x}^t | \mathbf{y}^o) \propto \exp(-\mathcal{J}), \quad (2.35)$$

387 with

$$388 \quad \mathcal{J}(\mathbf{x}^t) = \frac{1}{2} \left[(\mathbf{x}^t - \mathbf{x}^b)^T \mathbf{P}^{b-1} (\mathbf{x}^t - \mathbf{x}^b) + (\mathbf{y}^o - \mathbf{H}\mathbf{x}^t)^T \mathbf{R}^{-1} (\mathbf{y}^o - \mathbf{H}\mathbf{x}^t) \right]. \quad (2.36)$$

389 We can show that this last equation can be rewritten as

$$390 \quad \mathcal{J}(\mathbf{x}^t) = \frac{1}{2} \left[(\mathbf{x}^t - \mathbf{x}^a)^T \mathbf{P}^{a-1} (\mathbf{x}^t - \mathbf{x}^a) \right] + \beta, \quad (2.37)$$

391 with

$$\mathbf{P}^a = \left[\mathbf{P}^{b-1} + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \right]^{-1}, \quad (2.38)$$

$$\mathbf{x}^a = \mathbf{P}^a \left[\mathbf{P}^{b-1} \mathbf{x}^b + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{y}^o \right], \quad (2.39)$$

392 and β a vector independent of \mathbf{x}^t . With the help of the Sherman-Morrison formula (aka the
393 matrix inversion lemma according to Wunsch (2006), page 29)

$$394 \quad [\mathbf{A} + \mathbf{U}\mathbf{D}\mathbf{V}]^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U} [\mathbf{D}^{-1} + \mathbf{V}\mathbf{A}^{-1}\mathbf{U}]^{-1} \mathbf{V}\mathbf{A}^{-1}, \quad (2.40)$$

395 we can show that these are the BLUE equations (2.30-2.32). The a posteriori pdf defined
396 by Eq. (2.35) is thus Gaussian, with parameters given by the BLUE equations. Since the
397 BLUE provides the same result as the application of Bayes' theorem, it is the best estimator
398 (in the case of Gaussian pdfs and of a linear observation operator, though). In passing we
399 can recognize in Eq. (2.36) the cost function used in the static variational method termed
400 3D-Var. When it minimizes this cost function, the 3D-Var algorithm computes the Maximum
401 A Posteriori estimate of the Gaussian pdf, which is identical to the Minimum Variance esti-
402 mate found by the BLUE. We can take from this that when statistics are Gaussian and the
403 observation operator is linear, every method, whatever its name, will yield the same optimal
404 solution, which, depending on the philosophy followed, will be given different interpretations.

405

Chapter 3

The Kalman filter

3.1 Introduction

The system is now dynamical. Instead of a unique estimation, we set out to estimate a series of states \mathbf{x}_i^t (a sequence of real random vectors), where the index i refers to a discrete time index (when observations are made). The situation is summarized in Fig 3.1.

We assume the following a priori knowledge:

- the initial condition \mathbf{x}_0^t is Gaussian-distributed with mean \mathbf{x}_0^b and covariance \mathbf{P}_0^b ;
- a linear dynamical model \mathbf{M} describes the evolution of the state of the system we are interested in;
- the model errors (recall Sect. 1.3.1) $\boldsymbol{\eta}_i$ are Gaussian with zero mean (they are unbiased) and covariance \mathbf{Q}_i ;
- the model errors are white (i.e. uncorrelated) in time $\mathbb{E}(\boldsymbol{\eta}_i \boldsymbol{\eta}_j^T) = \mathbf{0}$ if $i \neq j$;
- Observation errors $\boldsymbol{\epsilon}_i^o$ are Gaussian, with zero mean and covariance matrix \mathbf{R}_i ;
- observation errors are white in time $\mathbb{E}(\boldsymbol{\epsilon}_i^o \boldsymbol{\epsilon}_j^{oT}) = \mathbf{0}$ if $i \neq j$;
- Errors of different kinds are independent

$$\mathbb{E}(\boldsymbol{\eta}_i \boldsymbol{\epsilon}_j^{oT}) = \mathbb{E}(\boldsymbol{\eta}_i \boldsymbol{\epsilon}_0^{bT}) = \mathbb{E}(\boldsymbol{\epsilon}_i \boldsymbol{\epsilon}_0^{bT}) = \mathbf{0}.$$

Under these many conditions, the Kalman filter provides the estimate of the states \mathbf{x}_i^t , conditioned by the past and present observations $\mathbf{y}_1^o, \dots, \mathbf{y}_i^o$; in terms of pdf, this amounts to considering

$$p(\mathbf{x}_i | \mathbf{y}_{1:i}^o),$$

where $\mathbf{y}_{1:i}^o = \{\mathbf{y}_1^o, \dots, \mathbf{y}_i^o\}$.

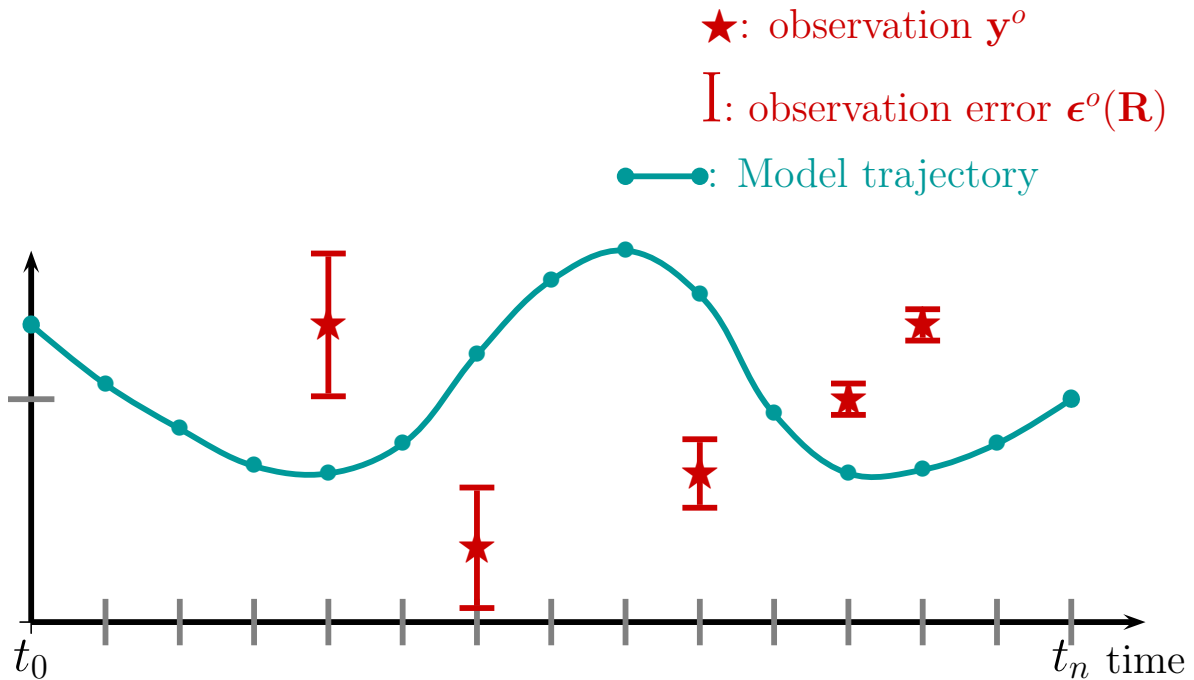


Figure 3.1: Assimilation starts with an unconstrained model trajectory over the time window of interest. It aims at correcting this initial model trajectory in order to provide an optimal fit to the available observations (the stars), given their error bars.

3.2 The Kalman filter algorithm

The Kalman filter algorithm is sequential and decomposed into two steps:

1. A **forecast**
2. An **analysis** (or observational update)

3.2.1 The forecast step

We start from some previously analyzed state \mathbf{x}_i^a (or from the initial condition \mathbf{x}_0 if $i = 0$), characterized by the Gaussian pdf $p(\mathbf{x}_i^a | \mathbf{y}_{1:i}^o)$ of mean \mathbf{x}_i^a and covariance matrix \mathbf{P}_i^a .

An estimate of \mathbf{x}_{i+1}^t is provided by the dynamical model. This defines the **forecast**. As seen in Sect. 1.3.1, we have

$$\mathbf{x}_{i+1}^f = \mathbf{M}_{i,i+1} \mathbf{x}_i^a, \text{ and} \quad (3.1)$$

$$\mathbf{P}_{i+1}^f = \mathbf{M}_{i,i+1} \mathbf{P}_i^a \mathbf{M}_{i,i+1}^T + \mathbf{Q}_{i+1}. \quad (3.2)$$

The forecast error ϵ_{i+1}^f results from the addition of two contributions (see Fig. 3.2): the propagation of the a priori error by the model, and the model error itself.

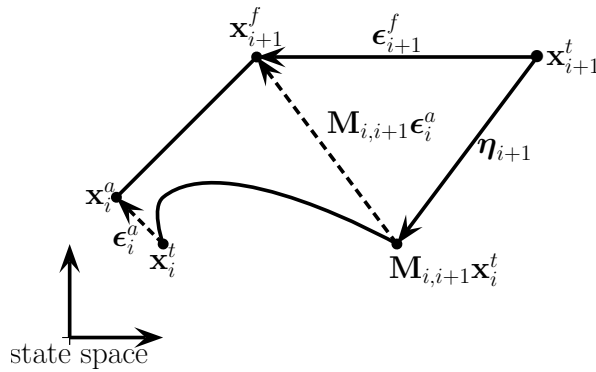


Figure 3.2: The forecast error ϵ_{i+1}^f has two sources: One is related to the propagation of the a priori error by the model (dashed arrow), and the other is related to the model itself: η_{i+1} quantifies the physics which the model does not account for properly. After Brasseur (2006).

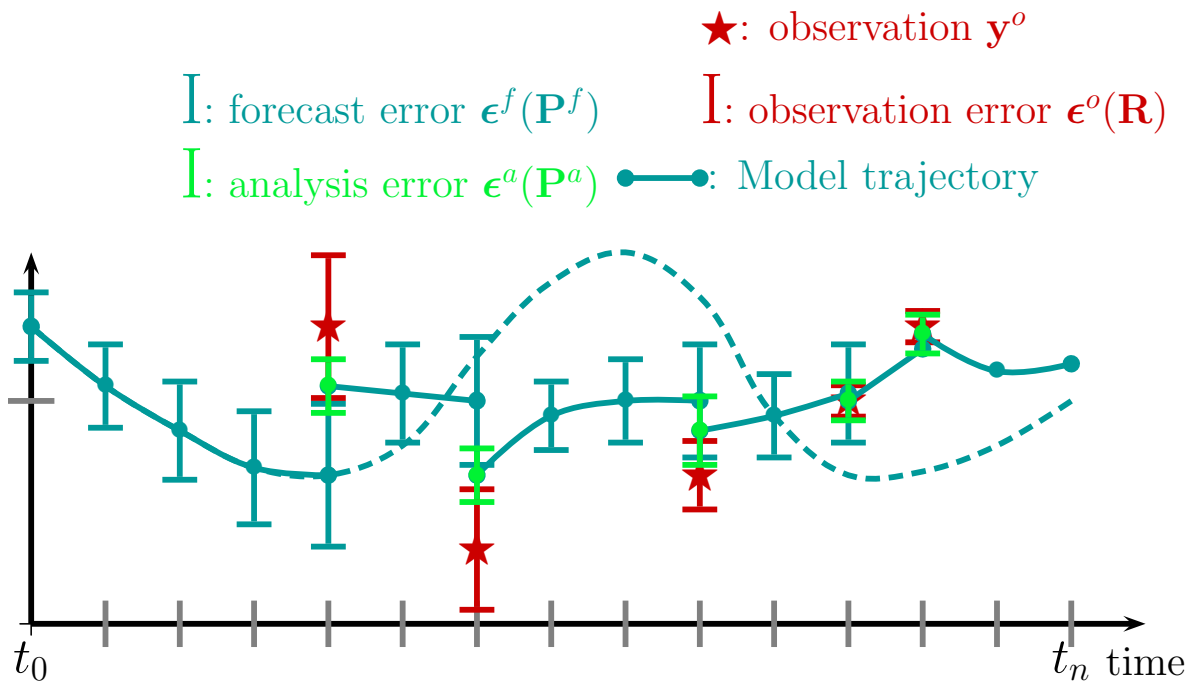


Figure 3.3: The sequential approach to data assimilation. Starting from the initial time, the model trajectory follows the initial forecast, and is characterized by a growth of the forecast error. As soon as the first observation is available, the analysis is performed (green bullet), and the associated error decreases (green error bar). The same cycle is repeated anytime an observation is available, with the assimilated trajectory deviating from the initial guess (the dashed line).

3.2.2 Analysis step

At time t_{i+1} , $p(\mathbf{x}_{i+1}|\mathbf{y}_{1:i}^o)$ is known through the mean \mathbf{x}_{i+1}^f and covariance matrix \mathbf{P}_{i+1}^f , and, again, the assumption of a Gaussian distribution. The analysis step consists of updating this pdf using the observation available at time t_{i+1} and to find $p(\mathbf{x}_{i+1}|\mathbf{y}_{1:i+1}^o)$. This comes down to re-deriving the BLUE equations of paragraph 2.4, this time in a dynamical context. Therefore “all” we have to do is compute

$$\mathbf{K}_{i+1} = \mathbf{P}_{i+1}^f \mathbf{H}_{i+1}^T \left(\mathbf{H}_{i+1} \mathbf{P}_{i+1}^f \mathbf{H}_{i+1}^T + \mathbf{R}_{i+1} \right)^{-1}, \quad (3.3)$$

$$\mathbf{x}_{i+1}^a = \mathbf{x}_{i+1}^f + \mathbf{K}_{i+1} \left(\mathbf{y}_{i+1}^o - \mathbf{H}_{i+1} \mathbf{x}_{i+1}^f \right), \quad (3.4)$$

$$\mathbf{P}_{i+1}^a = (\mathbf{I} - \mathbf{K}_{i+1} \mathbf{H}_{i+1}) \mathbf{P}_{i+1}^f, \quad (3.5)$$

The principle of the Kalman filter is illustrated in Fig. 3.3.

3.3 Implementation issues

3.3.1 Definition of covariance matrices and filter divergence

In case the input statistical information is mis-specified, the filter might end up underestimating the variances of the state errors, ϵ_i^a . Too much confidence is put on the state estimation and too little confidence is put on the information contained in the observations. The effects of the analysis is minimized, and the gain happens to be too small. In the most extreme case, observations are simply rejected. This is a **filter divergence**. We will see how we can get such a behaviour when we consider our vibrating string toy problem.

Very often filter divergence is easy to diagnose: state error variances are small, and the time sequence of innovations is biased. The fix is not as easy to make as the diagnostic. The main rule to follow is not to underestimate model errors. If possible, it is better to use an adaptive scheme to tune them on-the-fly.

3.3.2 Size / Optimal interpolation

The first limitation to the straightforward application of the Kalman filter is related to the size of the problem. If n denotes the size of the state vector, the state covariance matrix is $n \times n$. Since its propagation by means of the model is n times for expensive than a model step, it becomes rapidly out of reach when n increases (not to mention its storage).

If the storage is not an issue, but the computational cost of propagating \mathbf{P}^a is one, a possibility is to resort to a frozen covariance matrix

$$\mathbf{P}_i^a = \mathbf{P}^b \quad \forall t_i.$$

This defines the class of methods known as Optimal Interpolation (OI)¹.

¹Although the method is not really optimal, see e.g. Brasseur (2006)

467 Under this simplifying hypothesis, the two-step assimilation cycles defined above becomes:

1. Forecast:

$$\mathbf{x}_{i+1}^f = \mathbf{M}_{i,i+1} \mathbf{x}_i^a, \quad (3.6)$$

$$\mathbf{P}_{i+1}^f = \mathbf{P}^b. \quad (3.7)$$

468 2. Analysis:

$$\mathbf{x}_{i+1}^a = \mathbf{x}_{i+1}^f + \mathbf{K}_{i+1} (\mathbf{y}_{i+1}^o - \mathbf{H} \mathbf{x}_{i+1}^f), \quad (3.8)$$

$$\mathbf{P}_{i+1}^a = \mathbf{P}^b. \quad (3.9)$$

with $\mathbf{K}_{i+1} = \mathbf{P}^b \mathbf{H}_{i+1}^T (\mathbf{H}_{i+1} \mathbf{P}^b \mathbf{H}_{i+1}^T + \mathbf{R}_{i+1})^{-1}$.

469 There are at least two approaches to form the static covariance matrix \mathbf{P}^b .

470 1. The analytical formulation: The covariance matrix is formed from a vector of variances
471 and a correlation matrix \mathbf{C}

$$472 \quad \mathbf{P}^b = \mathbf{D}^{1/2} \mathbf{C} \mathbf{D}^{1/2}, \quad (3.10)$$

473 where \mathbf{D} is a diagonal matrix holding the variances and \mathbf{C} is a correlation matrix to be
474 defined. One example is (Brasseur, 2006, and references therein)

$$475 \quad C_{mn} = \left(1 + al + \frac{1}{3} a^2 l^2 \right) \exp(-al), \quad (3.11)$$

476 where a is a tunable parameter and l is the distance between the grid points m and n .
477 We will deal with a so defined background covariance matrix when we assimilate data
478 recorded along the vibrating string and see how the choice of the tunable parameter
479 can affect the behaviour of the assimilating scheme (see lab 2). However, we should
480 make it clear that such an approach is mostly relevant to multi-dimensional problems,
481 as it allows information to be spread from points where observations are made to points
482 where they are missing (think of satellite observation of sea surface height, for instance).

483 2. The second approach consists of taking an ensemble of N_e snapshots of the state vector
484 from a model free run, and to build the first and second statistical moments, \mathbf{x}^b and
485 \mathbf{P}^b , from this collection of snapshots. In practice we compute

$$\mathbf{x}^b = \frac{1}{N_e} \sum_{e=1}^{N_e} \mathbf{x}_e, \quad (3.12)$$

$$\mathbf{P}^b = \frac{1}{N_e - 1} \sum_{e=1}^{N_e} (\mathbf{x}_e - \mathbf{x}^b) (\mathbf{x}_e - \mathbf{x}^b)^T. \quad (3.13)$$

486 The static approach suffers from the fact that if a correction is applied along a certain direction
487 in state space during an update, the error statistics are not modified accordingly (by virtue of
488 Eq.(3.9) above). During the next update, the same level of correction might be applied along
489 the very same direction, whereas it might not be needed. The static approach is therefore
490 more suitable if two successive assimilation cycles are separated by a long enough time, so
491 that the corresponding dynamical states are decorrelated enough.

492 3.3.3 Evolution of the state error covariance matrix

493 In principle, Eq. (3.2) generates a symmetric matrix. Its practical implementation may not.
 494 Numerical truncation errors may lead to an asymmetric covariance matrix and a subsequent
 495 collapse of the filter. A remedy is to add an extra step to enforce symmetry, such as

$$496 \quad \mathbf{P}_{i+1}^f = \frac{1}{2} \left(\mathbf{P}_{i+1}^f + \mathbf{P}_{i+1}^{fT} \right).$$

497 Another possibility is to use the square root decomposition of the covariance matrix. Since
 498 \mathbf{P}^a is symmetric positive definite, it can be written as

$$499 \quad \mathbf{P}^a = \mathbf{S}^a \mathbf{S}^{aT},$$

500 where \mathbf{S}^a is a $n \times n$ matrix. This decomposition is not unique. For instance, in Lab 3, we will
 501 use a Cholesky factorization which will provide us with a lower triangular matrix \mathbf{S}^a . This is
 502 how this can be coded in matlab

```
503 function matPf=forward_matP(matP,NmatP,deltat,halfdeltat,halfdeltat2,beta,N)
504 matU=chol(matP);
505 matL=matU';
506 ibeg_disp = 1;
507 iend_disp = N-1;
508 ibeg_velo = iend_disp+1;
509 iend_velo = ibeg_velo+N-2;
510 ML=zeros(2*(N-1),2*(N-1));
511 for jcol=1:2*(N-1) % Each column is propagated
512     y=matL(ibeg_disp:iend_disp,jcol);
513     yd=matL(ibeg_velo:iend_velo,jcol);
514     [y_new,yd_new]=forward_it_neat(y,yd,deltat,halfdeltat,halfdeltat2,beta,N);
515     MP(:,jcol)=[y_new' yd_new' ]';
516 end
517 matPf=MP*(MP');
```

518 The propagation of the covariance matrix is then performed by first computing $\mathbf{M}_{i,i+1} \mathbf{S}_i^a$, and
 519 then by assembling $\mathbf{P}_{i+1}^f = (\mathbf{M}_{i,i+1} \mathbf{S}_i^a)(\mathbf{M}_{i,i+1} \mathbf{S}_i^a)^T + \mathbf{Q}_{i+1}$. (Note that we will neglect model
 520 errors in our labs, effectively taking $\mathbf{Q}_{i+1} = \mathbf{0}$.)

521 3.3.4 Nonlinearities

522 Nonlinearities are ubiquitous in geophysical fluid dynamics, and the cause of a great deal of
 523 concern for the data assimilation practitioner. Nonlinearities are likely to spoil the Gaussianity
 524 of statistics. In addition, the model can no longer be represented by a matrix, and its transpose
 525 is no longer defined. This statement also applies to a nonlinear observation operator. A way
 526 to proceed with nonlinearities is provided by the Extended Kalman Filter (EKF), which relies
 527 on a local linearization about the current model trajectory. This linearization is of course
 528 valid only in the case of a weakly nonlinear system.

529 **3.3.4.1 The extended Kalman filter (EKF)**

530 When the dynamical model \mathcal{M} and/or the observation operator \mathcal{H} are (weakly) nonlinear,
531 the Kalman filter can be extended by resorting to the **tangent linear** approximation of \mathcal{M}
532 and \mathcal{H} , denoted by \mathbf{M} and \mathbf{H} , respectively. The two-step filter assimilation cycle now writes :

1. Forecast:

$$\mathbf{x}_{i+1}^f = \mathcal{M}_{i,i+1}(\mathbf{x}_i^a), \text{ (nonlinear forecast)} \quad (3.14)$$

$$\mathbf{P}_{i+1}^f = \mathbf{M}_{i,i+1} \mathbf{P}_i^a \mathbf{M}_{i,i+1}^T + \mathbf{Q}_{i+1}. \text{ (linear forecast)} \quad (3.15)$$

533 2. Analysis:

$$\mathbf{x}_{i+1}^a = \mathbf{x}_{i+1}^f + \mathbf{K}_{i+1} \left[\mathbf{y}_{i+1}^o - \mathcal{H}_{i+1}(\mathbf{x}_{i+1}^f) \right], \quad (3.16)$$

$$\mathbf{P}_{i+1}^a = (\mathbf{I} - \mathbf{K}_{i+1} \mathbf{H}_{i+1}) \mathbf{P}_{i+1}^f. \quad (3.17)$$

with $\mathbf{K}_{i+1} = \mathbf{P}_{i+1}^f \mathbf{H}_{i+1}^T \left(\mathbf{H}_{i+1} \mathbf{P}_{i+1}^f \mathbf{H}_{i+1}^T + \mathbf{R}_{i+1} \right)^{-1}$.

534 **3.3.4.2 The ensemble Kalman filter (EnKF)**

535 The Kalman filter is only optimal in the case of Gaussian statistics and linear operators, in
536 which case the first two moments (the mean and the covariances) suffice to describe the pdf
537 entering the estimation problem. Practitioners report that its linearized extension to nonlinear
538 problems, the EKF, only works for moderate deviations from linearity and Gaussianity (e.g.
539 [Miller et al., 1994](#)). The ensemble Kalman filter ([Evensen, 1994, 2009](#)) is a method which has
540 been designed to deal with strong nonlinearities and non-Gaussian statistics, whereby the pdf
541 is described by an ensemble of N_e time-dependent states, $\mathbf{x}_{i,e}$.

542 A given cycle still consists of a forecast followed by an analysis, which relies on the good
543 old BLUE. The statistical information needed by the BLUE (see below) is provided by the
544 ensemble, at the exception of the observation errors in Eq. 3.24 below, which are random
545 drawings from the Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{R}_i)$. This is necessary for consistency with the
546 observation error covariance matrix.

1. Forecast:

$$\mathbf{x}_{i,e}^f = \mathcal{M}_{i-1,i}(\mathbf{x}_{i-1,e}^a) + \boldsymbol{\eta}_{i,e}, \quad e = \{1, \dots, N_e\}. \quad (3.18)$$

2. Analysis:

$$\langle \mathbf{x}_i^f \rangle = \frac{1}{N_e} \sum_{e=1}^{N_e} \mathbf{x}_{i,e}^f, \quad (3.19)$$

$$\mathbf{P}_i^f = \frac{1}{N_e - 1} \sum_{e=1}^{N_e} \left(\mathbf{x}_{i,e}^f - \langle \mathbf{x}_i^f \rangle \right) \left(\mathbf{x}_{i,e}^f - \langle \mathbf{x}_i^f \rangle \right)^T, \quad (3.20)$$

$$547 \quad \mathbf{H}_i \mathbf{P}_i^f = \frac{1}{N_e - 1} \sum_{e=1}^{N_e} \left[\mathcal{H}_i(\mathbf{x}_{i,e}^f) - \mathcal{H}_i(\langle \mathbf{x}_i^f \rangle) \right] \left[\mathbf{x}_{i,e}^f - \langle \mathbf{x}_i^f \rangle \right]^T \quad (3.21)$$

$$\mathbf{H}_i \mathbf{P}_i^f \mathbf{H}_i^T = \frac{1}{N_e - 1} \sum_{e=1}^{N_e} \left[\mathcal{H}_i(\mathbf{x}_{i,e}^f) - \mathcal{H}_i(\langle \mathbf{x}_i^f \rangle) \right] \left[\mathcal{H}_i(\mathbf{x}_{i,e}^f) - \mathcal{H}_i(\langle \mathbf{x}_i^f \rangle) \right]^T \quad (3.22)$$

$$\mathbf{K}_i = \left(\mathbf{H}_i \mathbf{P}_i^f \right)^T \left[\mathbf{H}_i \mathbf{P}_i^f \mathbf{H}_i^T + \mathbf{R}_i \right]^{-1}, \quad (3.23)$$

$$\mathbf{y}_{i,e}^o = \mathbf{y}_i^o + \boldsymbol{\epsilon}_e^o, \quad e = \{1, \dots, N_e\}, \quad (3.24)$$

$$\mathbf{x}_{i,e}^a = \mathbf{x}_{i,e}^f + \mathbf{K}_i \left[\mathbf{y}_{i,e}^o - \mathcal{H}_i(\mathbf{x}_{i,e}^f) \right], \quad e = \{1, \dots, N_e\}. \quad (3.25)$$

548 The problem of storing the state covariance matrix \mathbf{P}^a is solved, since “only” N_e state vectors
549 need be stored.

550 I have never applied the EnKF myself: a detailed description of its implementation can be
551 found in the book written by its inventor, Geir Evensen, along with a comprehensive list of
552 related publications (appendix B of the book).

553 A note of caution: the update phase (3.25) is still linear, as the Kalman gain matrix is
554 produced using Eq. (3.23).

Chapter 4

Variational assimilation

Here is a slightly modified excerpt taken from our review paper (Fournier et al., 2010).

Unlike sequential assimilation (which emanates from estimation theory), variational assimilation is rooted in optimal control theory. The analyzed state is not defined as the one maximizing a certain pdf, but as the one minimizing a functional \mathcal{J} of the form

$$\mathcal{J}(\mathbf{x}) = \frac{1}{2} \left\{ \sum_{i=0}^n [\mathcal{H}_i \mathbf{x}_i - \mathbf{y}_i^o]^T \mathbf{R}_i^{-1} [\mathcal{H}_i \mathbf{x}_i - \mathbf{y}_i^o] + [\mathbf{x} - \mathbf{x}^b]^T \mathbf{P}^{b-1} [\mathbf{x} - \mathbf{x}^b] \right\}, \quad (4.1)$$

in which $\mathbf{x}_i = \mathcal{M}_{i,i-1} \cdots \mathcal{M}_{1,0} \mathbf{x}$, the sought \mathbf{x} being the best estimate of the initial state of the core, \mathbf{x}_0 . This objective function is defined over the entire time window of interest. It is the sum of two terms. The first one measures the distance between the observations and the predictions of the model. It is weighted by the confidence we have in the observations. The second term is analogous to the various norms which are added when solving the kinematic core flow problem; it evaluates the distance between the initial condition and an a priori background state \mathbf{x}^b . That stabilizing term is weighted by the confidence we have in the definition of the background state, described by the background error covariance matrix \mathbf{P}^{b1} . Defining a background state for the core is no trivial matter. But one may substitute (or supplement) the corresponding term in equation 4.1 by (with) another stabilizing term, typically a norm, as was done by Talagrand and Courtier (1987) and Courtier and Talagrand (1987) in their early numerical experiments with the vorticity equation on the sphere.

The goal of variational data assimilation is to minimize \mathcal{J} by adjusting its control variables (or parameters), usually the initial condition \mathbf{x}_0 (if everything else is held fixed, see Fig. 4.1), as implied by our formulation in equation 4.1. Iterative minimization requires the computation of the sensitivity (gradient) of \mathcal{J} with respect to its control vector, which writes $(\nabla_{\mathbf{x}_0} \mathcal{J})^T$ (the transpose is needed since $\nabla_{\mathbf{x}_0} \mathcal{J}$ is by definition a row vector, recall Eq. 2.18). The size of the problem (the size of the state vector n) precludes a brute force calculation of the gradient (which would imply n realizations of the forward model over $[t_0, t_n]$). Fortunately, as pointed out early on by Le Dimet and Talagrand (1986) and Talagrand and Courtier (1987), a much more affordable method exists: The so-called adjoint method, which is based on the

¹Ide et al. (1997), and many others, use \mathbf{B} to denote that matrix, a notation which is preempted in our case by the magnetic induction.

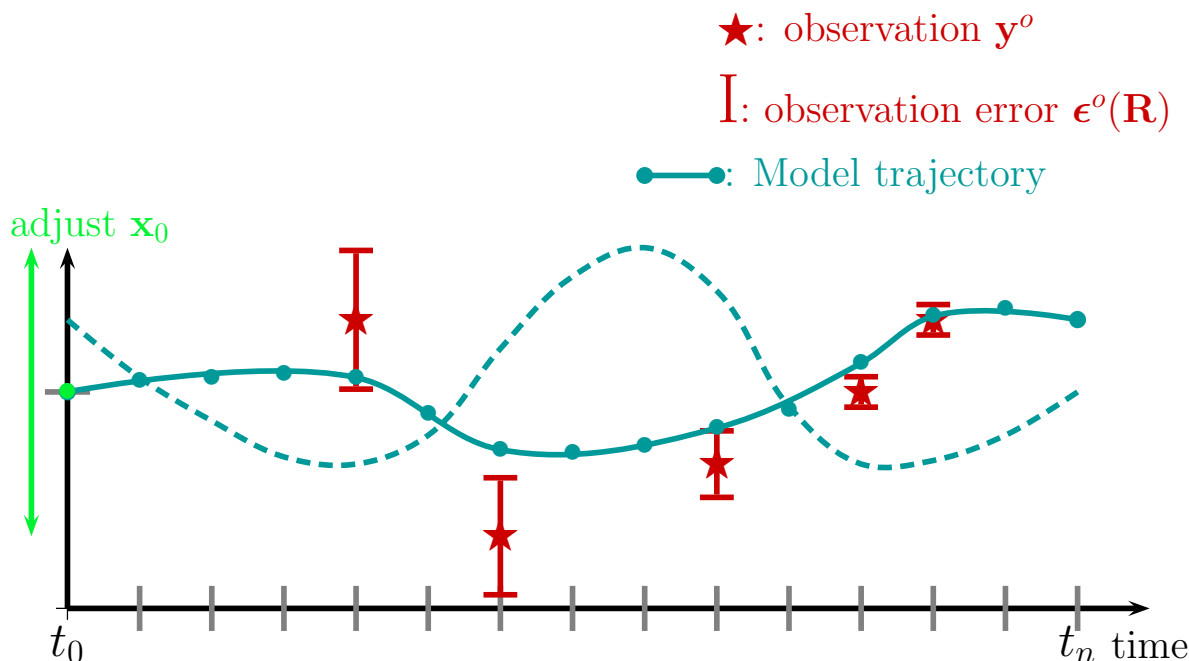


Figure 4.1: The variational approach to data assimilation. After adjustment of the initial condition \mathbf{x}_0 (the green bullet on the $t = t_0$ axis) by means of an iterative minimization algorithm, the model trajectory is corrected over the entire time window, in order to provide an optimal fit to the data (in a generalized least squares sense). The dashed line corresponds to the initial (unconstrained) guess of the model trajectory introduced in Fig. 3.1.

583 integration of the so-called adjoint equation backward in time

$$584 \quad \mathbf{a}_{i-1} = \mathbf{M}_{i-1,i}^T \mathbf{a}_i + \mathbf{H}_{i-1}^T \mathbf{R}_{i-1}^{-1} (\mathcal{H}_{i-1} \mathbf{x}_{i-1} - \mathbf{y}_{i-1}^o) + \delta_{i1} \mathbf{P}^{b-1} (\mathbf{x}_{i-1} - \mathbf{x}^b), \quad n \geq i \geq 1, (4.2)$$

585 starting from $\mathbf{a}_{n+1} = \mathbf{0}$, where \mathbf{a} is the adjoint field, and δ is the Kronecker symbol. The
 586 initial value of the adjoint field provides the sensitivity we seek: $(\nabla_{\mathbf{x}_0} \mathcal{J})^T = \mathbf{a}_0$ (e.g. Fournier
 587 et al., 2007); a derivation of Eq. (4.2) is provided in Appendix A. Note that when writing
 588 equation 4.2, we assumed for simplicity that observations were available at every model time-
 589 step.

590 Equation 4.2 indicates that over the course of the backward integration, the adjoint field
 591 is fed with innovation vectors. Those vectors have an observational component $(\mathcal{H}_{i-1} \mathbf{x}_{i-1} -$
 592 $\mathbf{y}_{i-1}^o)$, and a departure-to-background component $(\mathbf{x}_0 - \mathbf{x}^b)$ for the initial condition, these two
 593 contributions being weighted by the statistics introduced above. The adjoint model \mathbf{M}^T in
 594 equation 4.2 is the adjoint of the tangent linear model \mathbf{M} introduced previously in the context
 595 of the extended Kalman filter (Sec. 3.3.4.1). The adjoint model has a computational cost
 596 similar to that of the forward model, and makes it possible to use an iterative minimization
 597 algorithm suitable for large-scale problems.

598 A few comments on the adjoint method are in order:

- 599 • It demands the implementation of the adjoint model \mathbf{M}^T : the rules to follow for deriving
 600 (and validating) the tangent linear and adjoint codes from an existing forward code are

well documented in the literature (e.g. Talagrand, 1991; Giering and Kaminski, 1998), and leave no room for improvisation. Still, this process is rather convoluted. It requires expertise and deep knowledge of the forward code to begin with. The best situation occurs when the forward code is written in a modular fashion, bearing in mind that its adjoint will be needed in the future, and by casting as many operations as possible in terms of matrix-matrix or matrix-vector products (for a one-dimensional illustration with a spectral-element, non-linear magnetohydrodynamic model, see Fournier et al., 2007). The task of coding an adjoint by hand can still become beyond human reach in the case of a very large model. One might then be tempted to resort to an automated differentiation algorithm. Automated differentiation (AD) is a very active field of research²: several operational tools are now available, some of which have been tested on geophysical problems by Sambridge et al. (2007).

- The discrete adjoint equation 4.2 is based on the already discretized model of core dynamics. An alternative exists, which consists first in deriving the adjoint equation at the continuous level, and second in discretizing it, using the same machinery as the one used to discretize the forward model. In most instances, both approaches to the adjoint problem yield the same discrete operators. When in doubt, though, in the case of a minimization problem, one should take the safe road and derive the adjoint of the already discretized problem: This guarantees that the gradient injected in the minimization algorithm is exactly the one corresponding to the discrete cost function (equation 4.1), up to numerical roundoff error. Since the efficiency of a minimization algorithm grows in proportion to its sensitivity to errors in the gradient, any error in the gradient could otherwise result in a suboptimal solution.
- The adjoint approach is versatile. Aside from the initial state \mathbf{x}_0 , one can declare static model parameters (static fields, material properties) adjustable, and add them to the control vector.
- In the case of a non-linear problem, the forward trajectory \mathbf{x}_i , $i \in \{0, \dots, n\}$, is needed to integrate the adjoint equation. The storage of the complete trajectory may cause memory issues (even on parallel computers), which are traditionally resolved using a so-called checkpointing strategy. The state of the system is stored at a limited number of discrete times, termed checkpoints. Over the course of the backward integration of the adjoint model, these checkpoints are then used to recompute local portions of the forward trajectory on-the-fly, whenever those portions are needed (e.g. Hersbach, 1998).
- On a more general note, adjoint methods have gained some popularity in solid Earth geophysics over the past few years, a joint consequence (again) of the increase in computational power and the availability of high-quality satellite, or ground-based, data. Adjoint methods are now applied to problems related to the structure and evolution of the deep Earth: Electromagnetic induction (Kelbert et al., 2008; Kuvshinov et al., 2010), mantle convection (Bunge et al., 2003; Liu and Gurnis, 2008; Liu et al., 2008), and seismic wave propagation (Tromp et al., 2005; Fichtner et al., 2006; Tromp et al., 2008), building in that last case on the theoretical work of Tarantola (1984, 1988).

²www.autodiff.org

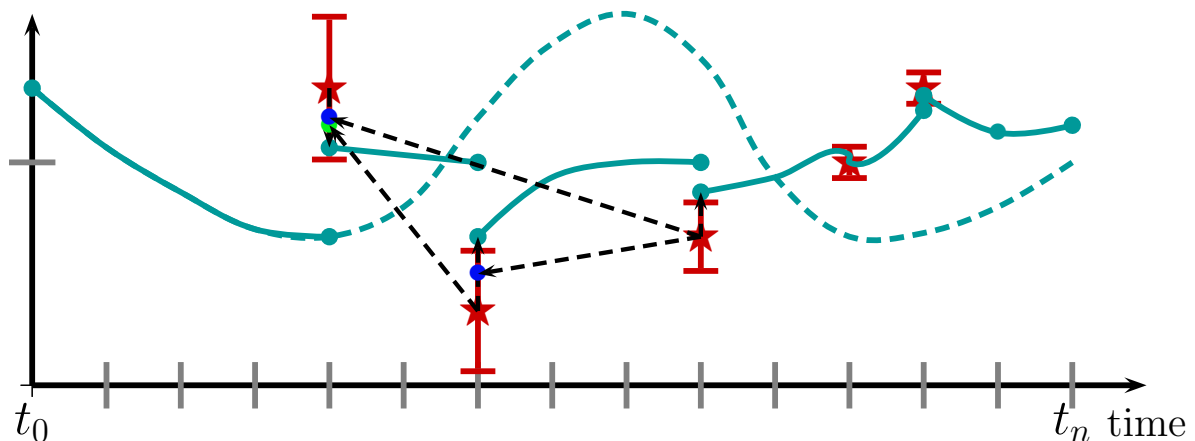


Figure 4.2: Principle of sequential smoothing. The state to observation difference (the innovation) at analysis time can be used to retrospectively correct the past products of analysis.

642 The application of a variational approach to time-dependent problems has been generically
 643 labeled as the 4D-Var approach to data assimilation (e.g. [Courtier, 1997](#)), and is commonly
 644 referred to as 4D-Var. As such, the standard 4D-Var suffers from two drawbacks: It assumes
 645 that the model is perfect ($\boldsymbol{\eta} = \mathbf{0}$), and it does not provide direct access to the statistics of
 646 the analysis error - notice its absence in Fig. 4.1. An alternative approach to the “strong
 647 constraint” assumption ($\boldsymbol{\eta} = \mathbf{0}$) consists in adding a term quantifying the model error in
 648 the definition of the cost function, a term whose weight is controlled by an a priori forecast
 649 error covariance. This more general “weak constraint” approach ([Sasaki, 1970](#)) has been suc-
 650 cessfully introduced and implemented (under the name “method of representers”) in physical
 651 oceanography during the past fifteen years ([Egbert et al., 1994](#); [Bennett, 2002](#), and references
 652 therein).

653 From a general perspective, the advantages of a variational approach are its flexibility re-
 654 garding the definition and identification of control variables, and its natural ability to handle
 655 time-dependent observation operators (and possibly time-correlated errors). It is also well-
 656 suited for the reanalysis of past data records (hindcasting), since the state at a given time
 657 is estimated using the past and future observations available over the entire time window
 658 (see Fig. 4.1). Note, however, that hindcasting is also possible if one resorts to sequential
 659 smoothers (see Fig. 4.2), of the kind described by e.g. [Cohn et al. \(1994\)](#), and applied in an
 660 oceanic context by e.g. [Cosme et al. \(2010\)](#).

661

Part II

662

Labs

Lab 1

The vibrating string: forward modelling

1.1 Introduction

A good understanding of the forward model is mandatory before any practice of data assimilation. The goal of this first lab is to get familiar with the numerical model we will deal with in the practicals, which has to do with the description of transverse motion along a vibrating string. This is a classical problem in physics which is of musical and academic interest. It allows one to derive the one-dimensional wave equation and to introduce the notion of normal modes. A nice description of this problem can be found in the textbook by [French \(1971\)](#), chapters 6 and 7.

As shown in Fig. 1.1, the physical system consists of a string of length L , and mass per unit length μ , which is stretched in the x direction up to a certain tension T . We assume that the string is held fixed at both ends. Tension is precisely the restoring force responsible for wave motion. Under the assumptions that the string is non-elastic, and that the angle characterizing the deviation of its shape from the horizontal direction remains small, one can show that the transverse displacement y obeys the one-dimensional wave equation (aka d'Alembert's equation)

$$\partial_t^2 y - c^2 \partial_x^2 y = 0, \tag{1.1}$$

$$y(x = 0, t) = y(x = L, t) = 0. \tag{1.2}$$

in which c is the wave speed,

$$c = \sqrt{\frac{T}{\mu}}. \tag{1.3}$$

For this second order initial value problem, both the initial displacement $y(\cdot, t = 0)$ and the transverse velocity $\partial_t y(\cdot, t = 0)$ need be specified. Using the length of the string L as the length scale, and the travel time $\tau = L/c$ as the time scale, we can conveniently rewrite the

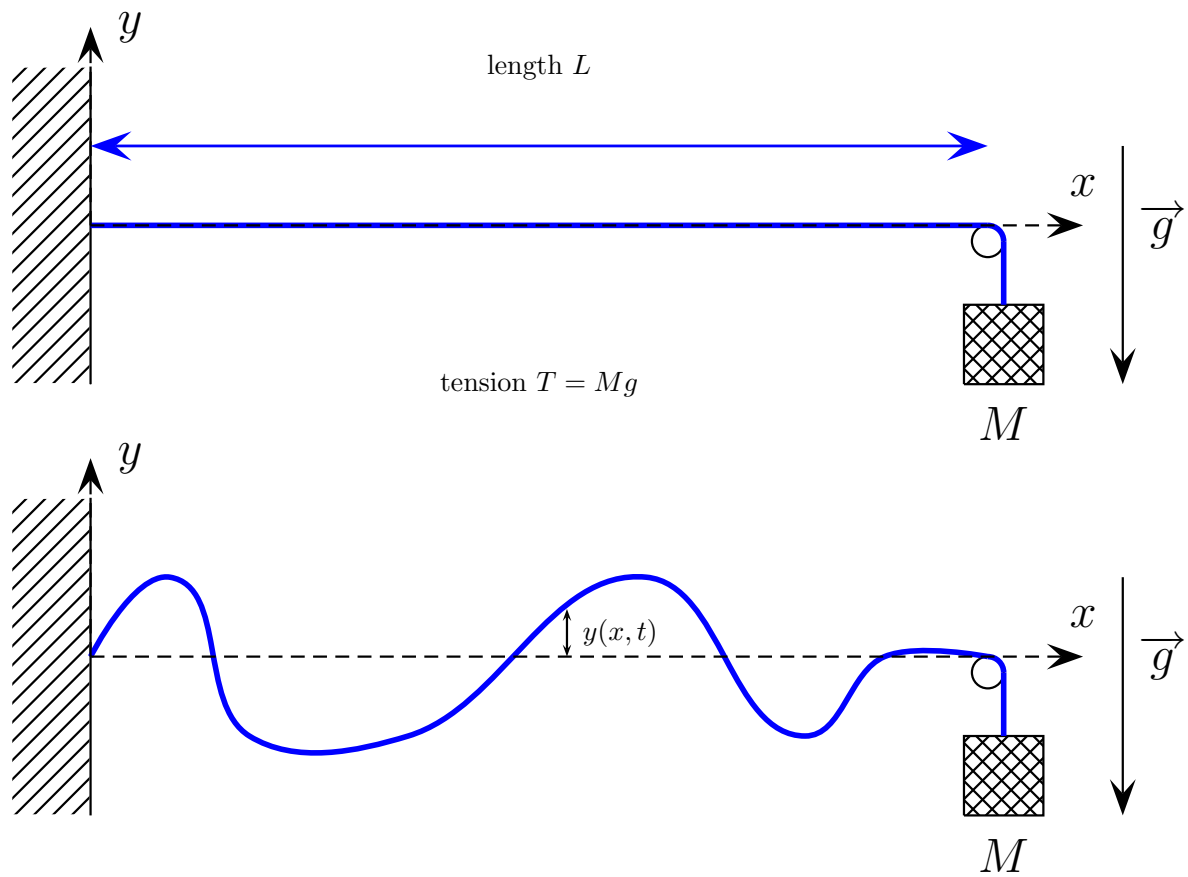


Figure 1.1: The vibrating string in its equilibrium configuration (top), and away from this configuration (bottom, with vertical exaggeration). Both ends of the string are held fixed.

686 same problem in a non-dimensional form

$$\partial_t^2 y - \partial_x^2 y = 0, \quad (1.4)$$

$$y(x = 0, t) = y(x = 1, t) = 0, \quad (1.5)$$

$$+\text{initial } y \text{ and } \partial_t y. \quad (1.6)$$

687 In the following, we will focus on this form of the problem. We approximate its solution using
 688 the finite difference method in both space and time. In space, the segment $[0, 1]$ is divided
 689 into N segments of equal length $h = 1/N$. In time, we resort to an explicit, second-order
 690 Newmark scheme¹, with a time-step size denoted by Δt .

691 1.2 The Courant-Friedrichs-Lewy (CFL) stability condition

692 Since it is explicit, the time scheme is subject to the Courant-Friedrichs-Lewy stability con-
 693 dition: the value of Δt should not exceed a critical value Δt_{\max}

$$694 \quad \Delta t \leq \Delta t_{\max} = \text{cte} \times \frac{h}{c}, \quad (1.7)$$

695 in which c is the wave speed (1 in our dimensionless paradise), h is the grid spacing introduced
 696 above, and cte is a constant whose value depends on the time scheme.

697 **Q1:** Using the `param_lab1_q1.m` parameter file and running the `script_lab1_q1.m` matlab
 698 script, determine empirically the value of the prefactor cte (trial and error). In other words,
 699 how does Δt_{\max} vary with h , for a simple enough initial condition (here the fundamental
 700 mode of vibration of the string)? A simple inspection of the solution tells you if it is stable
 701 or not. Feel free to automate the procedure if you feel so inclined.

702 Fill the following table

703	h	0.1	0.01	0.001	0.0001
	Δt_{\max}				

704 **Q2:** Find out if the value of cte depends on the initial condition, by selecting a different
 705 normal mode `imode` as the initial condition for displacement, and by varying its amplitude
 706 `amp0`, using the `param_lab1_q2.m` parameter file and running the `script_lab1_q2.m` matlab
 707 script.

708 At this stage, we are able to determine ab initio the maximum admissible value for the time
 709 step Δt , given a spatial chosen resolution (characterized by $h = 1/N$). This is what we will
 710 do automatically in the following.

711 1.3 Resolution / dispersion

712 Depending on the spatial scale one wishes to resolve, it is important to know which grid
 713 resolution to prescribe. We will assume a Gaussian initial profile for the displacement, and

¹ of the kind used for seismic wave propagation modelling, see e. g. [Komatitsch and Vilotte \(1998\)](#); [Chaljub and Valette \(2004\)](#)

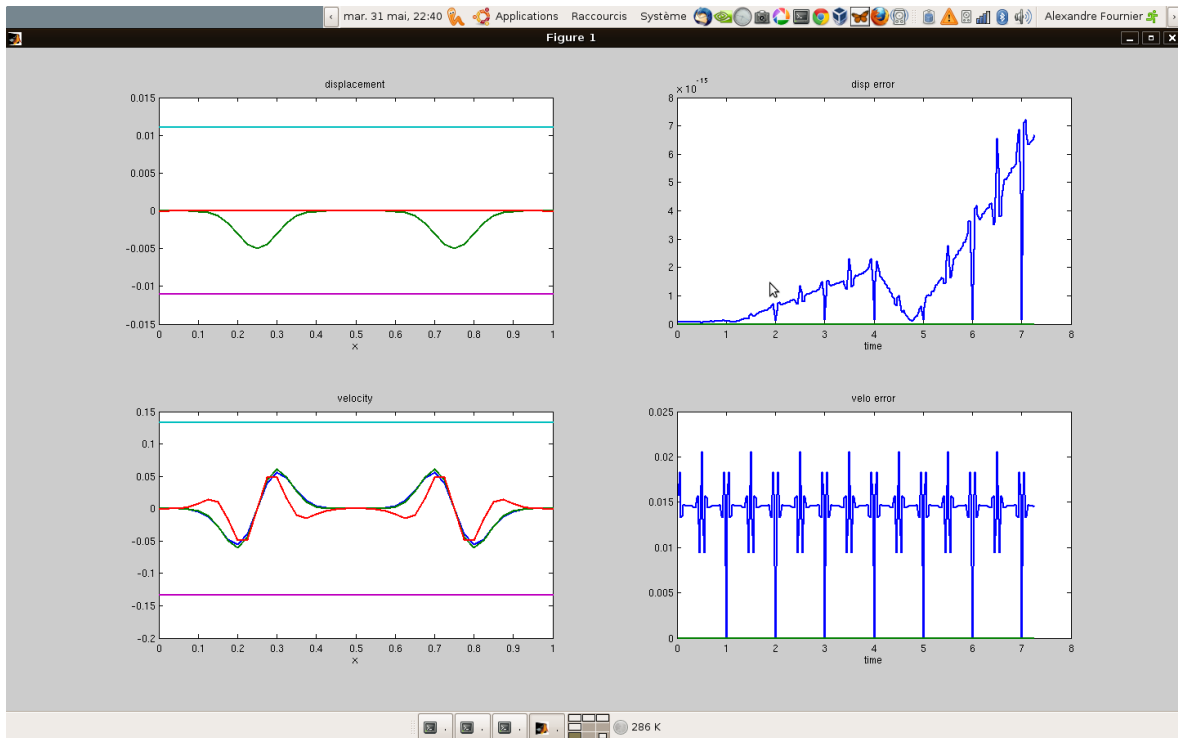


Figure 1.2: A typical screenshot obtained while answering **Q3**.

714 a zero initial velocity. The Gaussian profile is characterized by a standard deviation σ_0 (our
715 characteristic length scale)

$$716 \quad y(x, t = 0) = A_0 \exp \left[-\frac{(x - x_0)^2}{2\sigma_0^2} \right]. \quad (1.8)$$

717 **Q3:** Use `param_lab1_q3.m` to change the value of σ_0 (`sigma0`) and the grid resolution (`N`).
718 Running `script_lab1_q3` will allow you to visualize the propagation of the numerical solution
719 and to compare it with an analytical solution, constructed using the catalog of normal modes
720 of the string. The plot will also display the difference between the two multiplied by a factor
721 of 10. For a given value of σ_0 the goal is to find the minimum N_{\min} such that the numerical
722 solution still looks okay compared to the reference one, after the equivalent of 5 travel times.

723 The arrays `L2_diff_velo` and `L2_diff_disp` contains, for each discrete time t_i , the absolute
724 error (in a L_2 sense) of the numerical solution compared with the analytical one. They
725 are plotted to the right of the displacement and velocity fields. Figure 1.2 shows a typical
726 screenshot.

727 Use these tools to fill the following table

σ_0	0.1	0.05	0.02	0.01	0.001
N_{\min}					
$1/N_{\min}$					

728

729 Knowing which resolution is needed for a given σ_0 will allow us to neglect the model error $\boldsymbol{\eta}$
 730 in our assimilation experiments.

731 NB: the phenomenon you see appearing when the resolution is not good enough is called
 732 numerical dispersion. Its effects are stronger as the wave travels a longer distance (I have
 733 chosen 7.25 travel times as the duration of integration in my script. Feel free to modify it).

734 1.4 Generating observations

735 [parameter file `param_lab1_q4.m` and script `script_lab1_q4.m`]

736 In labs 2 and 3, assimilation will be performed by assimilating displacement and velocity time
 737 series recorded by an array of receivers, located along the string. For that purpose, we define
 738 an observation operator \mathbf{H} which will generate these time series from the knowledge of the
 739 displacement and velocity fields on the finite difference grid,

$$740 \quad x_k = k \times h. \quad (1.9)$$

741 The state vector \mathbf{x}_i will refer to the column vector containing the values of the displacement
 742 and velocity fields on the finite difference grid at any discrete time t_i

$$743 \quad \mathbf{x}_i = \begin{bmatrix} y_{1,i} \\ \vdots \\ y_{k,i} \\ \vdots \\ y_{N-1,i} \\ \dot{y}_{1,i} \\ \vdots \\ \dot{y}_{k,i} \\ \vdots \\ \dot{y}_{N-1,i} \end{bmatrix}, \quad (1.10)$$

744 where $y_{k,i} = y(x = x_k, t = t_i)$ (same for $\dot{y} = \partial_t y$).

745 The endpoints 0 and 1 ($k = 0$ and $k = N$) are not included since the value of the fields these
 746 endpoints is specified by the boundary conditions.

747 The receivers are located between position $x=xrecleft$ and $x=xrecright$, and equally
 748 spaced every `deltaxrec`. Observations will be noised. The noise level can vary (or not)
 749 from one receiver to the next. This noise will be Gaussian, with a standard deviation called
 750 `sigmao_disp` and `sigmao_velo` for displacement and velocity, respectively. The standard
 751 deviation is to be understood as a fraction of the maximum amplitude of the displacement
 752 (velocity). For instance, if the initial velocity has an amplitude A_0 , setting `sigmao_disp` to
 753 0.01 will generate a Gaussian noise of amplitude $0.01A_0$.

754 **Q4:** Play around with the various parameters to get a feeling for what the timeseries actually
 755 look like. An example is shown in Fig. 1.3.

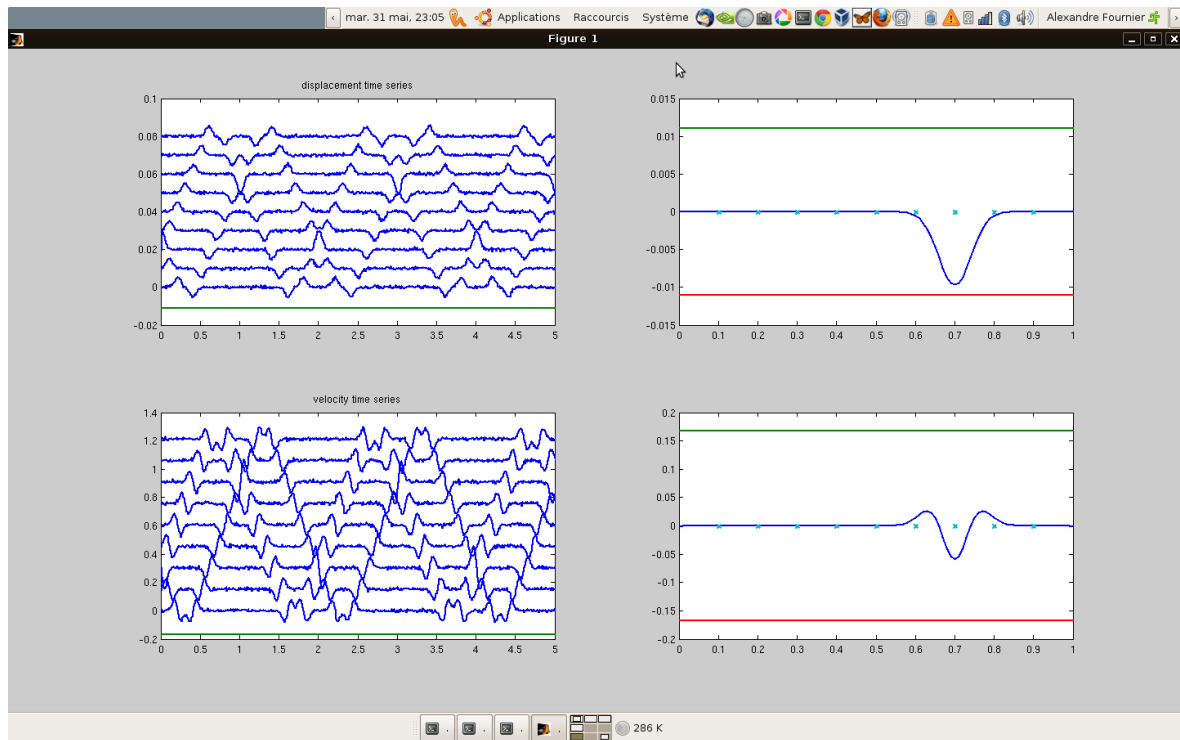


Figure 1.3: An example obtained using the following parameters

```

N=100;           % number of segments dividing the interval
duration=5;     % duration of the simulation
                % Initial condition: zero velocity, Gaussian profile
x0=0.3;         % location of the Gaussian peak (between 0 and 1)
sigma0=0.04 ;  % Gaussian standard dev
amp0=0.01 ;    % amplitude of the Gaussian
xrecleft=0.1;  % leftmost coordinate of receiver arrays
xrecright=0.9; % rightmost coordinate of receiver arrays
deltaxrec=0.1 ; % spacing between receivers
sigmao_disp=0.04; %
sigmao_velo=0.04; %
uniform_array=true;%

```

756 **Appendix: analytical solution based on normal modes** Fourier analysis allows us to
757 derive the transverse displacement $y(x, t)$ at any time by expanding the initial displacement
758 in a series of sine functions (the normal modes of vibration of the string). For an initial
759 displacement $\alpha(x)$ and a zero initial velocity, we find

$$760 \quad y(x, t) = \sum_{n=1}^{+\infty} \alpha_n \sin(n\pi x) \cos(\omega_n t), \quad (1.11)$$

761 in which

$$762 \quad \omega_n = n\pi \quad (1.12)$$

763 and

$$764 \quad \alpha_n = 2 \int_0^1 \alpha(x) \sin(n\pi x) dx. \quad (1.13)$$

765 (Recall that we work in a dimensionless world.)

766 Lab 2

767 An optimal interpolation scheme 768 applied to the vibrating string

769 The goal of this lab is to get familiar with the working of a sequential assimilation scheme
770 running a so-called twin experiment: A true, reference model trajectory \mathbf{x}_i^t is generated, and is
771 used to construct a catalog of synthetic observations. These observations are then assimilated
772 in order to correct a second model trajectory, which differs from the first one (the true one).
773 In our case it will differ because we will assume a different initial condition, $\mathbf{x}_0 \neq \mathbf{x}_0^t$.

774 Twin experiments (also called OSSE, Observing System Assimilation Experiments) are a
775 logical first step when implementing an assimilation scheme, since they allow to develop an
776 understanding for the behaviour of the scheme, without the additional complexity which may
777 arise from the inability of the forward model to represent some of the physics expressed in the
778 observations. Today we will run these twin experiments using our vibrating string toy model,
779 and we will resort to an optimal interpolation assimilation scheme, of the kind described in
780 Sect. 3.3.2.

781 2.1 Statistical ingredients

782 We need to begin by specifying the statistical bits of information needed by the scheme, in the
783 form of the covariance matrices \mathbf{P}^b and \mathbf{R} of background and observation error, respectively.

784 2.1.1 Model

785 On the account of our perfect control of both the physics we are interested in and its numerical
786 approximation (thanks to lab1), we will neglect modelling errors (as introduced for the first
787 time in the notes in Eq. 1.15)

$$788 \quad \boldsymbol{\eta} = \mathbf{0}. \tag{2.1}$$

789 The background covariance matrix is next defined as in Eq. (3.10)

$$790 \quad \mathbf{P}^b = \mathbf{D}^{1/2} \mathbf{C} \mathbf{D}^{1/2}, \tag{2.2}$$

791 where \mathbf{D} is a diagonal matrix holding the variances. We will assume the same value of the
792 variance for every grid point, taking

$$793 \quad \mathbf{D}^{1/2} = \left[\begin{array}{c|c} \sigma_{md}\mathbf{I} & \mathbf{0} \\ \hline \mathbf{0} & \sigma_{mv}\mathbf{I} \end{array} \right], \quad (2.3)$$

794 where σ_{md} and σ_{mv} are the model standard deviation for displacement and velocity, respec-
795 tively, (which we will be able to vary), and \mathbf{I} is the identity matrix of size half the size of
796 the state vector (recall its definition, Eq. 1.10 in lab1). In the codes, they are called `sigmand`
797 and `sigmamv`. As is the case for the observation standard deviations, these quantities will be
798 defined as a fraction of the maximum amplitude of the initial displacement A_0 for the first
799 one, and as a fraction of the maximum expected velocity for the second one¹.

800 \mathbf{C} is a correlation matrix, defined by

$$801 \quad C_{mn} = \left(1 + al + \frac{1}{3}a^2l^2 \right) \exp(-al), \quad (2.4)$$

802 where a is a tunable parameter and l is the distance between the grid points m and n . a can
803 be defined as the inverse of the correlation lengthscale,

$$804 \quad a = l_c^{-1}, \quad (2.5)$$

805 The value of l_c will we a free parameter as well, and it will appear as `lcorr` in the matlab
806 codes.

807 2.1.2 Observations

808 Displacement and/or velocities are recorded by the array of receivers (whose properties are
809 described in lab 1, Sect. 1.4). We will assume that the observation errors are time independent,
810 and that their distribution follows a Gaussian pdf. The observation error covariance matrix
811 \mathbf{R} is diagonal, of size $2s$, s being the number of receivers (`nrec` in the code). The diagonal
812 elements of \mathbf{R} hold the variances characterizing the Gaussian noise affecting each receiver.
813 The standard deviation of the noise affecting each receiver ir will be denoted by $\sigma_{ir,d}^o$ and
814 $\sigma_{ir,v}^o$ for displacement and velocity, respectively.

815 2.2 Algorithm

816 2.2.1 Preparation of data

817 In this lab, all the parameters are defined in `param_lab2.m`.

818 We set the duration of the experiment (`duration` in matlab) to a value T . Once \mathbf{x}_0^t is specified
819 (we will again assume a Gaussian profile for displacement, and zero velocity), we know from
820 lab1 which resolution N is needed to generate a a clean (that is, not contaminated by a

¹That is, $A_0/(\sigma_0 e^{1/2})$ in the case of a Gaussian displacement of amplitude A_0 and standard deviation σ_0 , and zero velocity.

821 substantial level of numerical error) true trajectory \mathbf{x}_i^t . (Note that once we choose N , the
822 timestep Δt is automatically computed.)

823 The specification of the standard deviation of the observation error σ_{ir} allows us to generate
824 observations on-the-fly, while computing the sequence of \mathbf{x}_i^t , by the now well-known formula

$$825 \quad \mathbf{y}_i^o = \mathbf{H}\mathbf{x}_i^t + \boldsymbol{\epsilon}_i^o, \quad (2.6)$$

826 in which every component $\epsilon_{ir,i}^o$ of the vector of size s $\boldsymbol{\epsilon}_i^o$ is drawn randomly following a normal
827 distribution with zero mean and standard deviation σ_{ir}^o (this applies to the displacement
828 and/or the velocity).

829 The observation operator \mathbf{H} is constructed once and for all once the grid resolution, and the
830 location of the receivers, are prescribed. Receivers start to operate at time `time0_obs`.

831 2.2.2 Initialization

832 [All these steps are taken care of in the `init_matrices_lab2.m` script]

- 833 • With l_c , σ_{md} and σ_{mv} specified, construct the frozen \mathbf{P}^b .
- 834 • With the $\sigma_{ir,d}^o$ and $\sigma_{ir,v}^o$ already specified, form the block diagonal matrix

$$835 \quad \mathbf{R} = \text{blkdiag} \left(\left[\begin{array}{ccccc} \sigma_{1,d}^o & 0 & \dots & \dots & 0 \\ 0 & \ddots & \ddots & 0 & \vdots \\ \vdots & \ddots & \sigma_{ir,d}^o & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 & \sigma_{s,d}^o \end{array} \right]^2, \left[\begin{array}{ccccc} \sigma_{1,v}^o & 0 & \dots & \dots & 0 \\ 0 & \ddots & \ddots & 0 & \vdots \\ \vdots & \ddots & \sigma_{ir,v}^o & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 & \sigma_{s,v}^o \end{array} \right]^2 \right) \quad (2.7)$$

- 836 • Finally, assemble the Kalman gain matrix

$$837 \quad \mathbf{K} = \mathbf{P}^b \mathbf{H}^T \left(\mathbf{H} \mathbf{P}^b \mathbf{H}^T + \mathbf{R} \right)^{-1}. \quad (2.8)$$

838 2.2.3 Assimilation cycle

839 Pick your wrong initial condition, \mathbf{x}_0^f . (In practice a Gaussian profile again for the displace-
840 ment, and zero for the velocity. Setting the amplitude of the Gaussian to 0 amounts to
841 choosing a zero initial condition for the displacement.)

842 While time has not reached its final value T , do the following

- 843 1. given \mathbf{x}_i^a , use the model to compute the forecast \mathbf{x}_{i+1}^f
- 844 2. If the observations \mathbf{y}_{i+1}^o exist, perform the analysis $\mathbf{x}_{i+1}^a = \mathbf{K} \left(\mathbf{y}_{i+1}^o - \mathbf{H}\mathbf{x}_{i+1}^f \right)$

845 The modest size of the problem conveniently allows us to store the entire true and analysed
846 trajectory, \mathbf{x}_i^t , and \mathbf{x}_i^a .

847 The construction of the trajectory is done in practice by running the `run_OI_lab2.m` script.

848 2.2.4 A description of the tools at your disposal for this lab

849 Compared to lab 1, there is more flexibility as to what you can choose to do within this lab.
850 Here is a summary of the commands which might come in handy:

- 851 • `param_lab2.m` is the parameter file
- 852 • `create_data_lab2.m` generates the true trajectory \mathbf{x}_i^t and the synthetic observations
- 853 • `init_matrices_lab2.m` creates \mathbf{P}^b , \mathbf{R} and \mathbf{K}
- 854 • `run_OI_lab2.m` generates the predicted trajectory \mathbf{x}_i^a
- 855 • `doall_lab2.m` executes `create_data_lab2.m`, `init_matrices_lab2.m`, and `run_OI_lab2.m`
856 in this order
- 857 • `compute_chi2_disp.m` computes the following quantities for the displacement field

$$\chi_2\text{disp}_i = (\mathbf{y}_i^o - \mathbf{H}\mathbf{x}_i)^T \mathbf{R}^{-1} (\mathbf{y}_i^o - \mathbf{H}\mathbf{x}_i) \quad \forall i \in \{1, \dots, N_t\}, \quad (2.9)$$

$$\chi_2\text{disp} = \frac{1}{N_t} \sum_i \chi_2\text{disp}_i. \quad (2.10)$$

858 where we understand that the various vectors and matrices are restricted to their dis-
859 placement component. In this formula, N_t is the number of time steps.

- 860 • `compute_chi2_velo.m` does the same for the velocity field. Note that by summing the
861 two, you will get the total misfit to the data.
- 862 • `plot_traj_disp` plots the time space evolution of the displacement component of the
863 true trajectory, the predicted one, and the difference between the two. An example is
864 shown in Fig. 3.1.
- 865 • `plot_traj_velo` does the same for the velocity
- 866 • `compute_l2diff_state` computes, over the last time unit of the simulation (ie, one
867 travel time) the quantity

$$\sqrt{\int_{T-1}^T (\mathbf{x}^t - \mathbf{x}^a)^T (\mathbf{x}^t - \mathbf{x}^a) dt}, \quad (2.11)$$

869 which is returned under the name `l2diff`. The script also returns `l2diff_d` and
870 `l2diff_v`, which are the displacement and velocity restrictions of the first one, nor-
871 malized by A_0 and $A_0/(\sigma_0 \exp(1/2))$, respectively.

- 872 • `irec= something` followed by `plot_traces` will plot the traces recorded at receiver
873 `irec` (\mathbf{y}^o), and their prediction ($\mathbf{H}\mathbf{x}^a$). These traces can be useful in diagnosing a filter
874 divergence (see below).

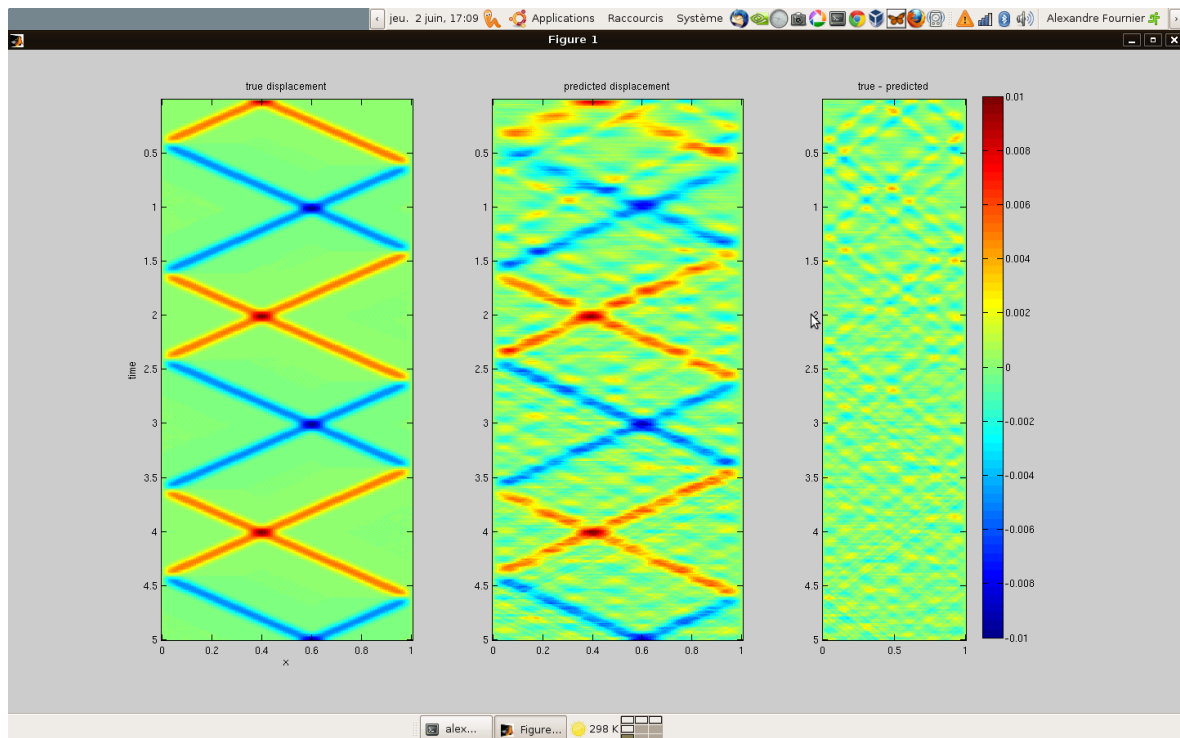


Figure 2.1: A screenshot showing the result of the command `plot_traj_disp`.

2.3 Points to address

875

876 Use these tools to try and answer the following questions, or others that might come to mind
 877 while answering these. I suggest you begin with the parameters listed in `param_lab2.m` to get
 878 familiar with these routines (in particular, we shall ascribe the same level of noise to every
 879 receiver).

880 **Q1:** Is the estimate of the displacement field any better when the velocity is observed as
 881 well? Everything else being fixed, does your answer depend on the ratio σ_v^o/σ_d^o ? How? And
 882 why?

883 **Q2:** All other parameters remaining fixed, what is the optimal correlation length l_c in
 884 terms of getting the smallest misfit? The smallest difference (in a L_2 sense) between the two
 885 trajectories over the last cycle of oscillation?

886 **Q3:** Are there values of l_c which make the scheme unstable? Does these values depend
 887 on the grid spacing h (or equivalently, the standard deviation σ_0 which defines the initial
 888 displacement)? How?

889 **Q4:** Is the correlation length l_c relevant at all for this one-dimensional problem?

890 **Q5:** Filter divergence is said to occur when one has too much confidence in the numerical
 891 model compared to the confidence in the observations. Assuming that you observe only the
 892 velocity, find the corresponding critical σ_{mv}/σ_v^o ratio.

893 **Q6:** You are running out of funds. All that is left in your pocket can buy you either
894 an ‘array’ of 5 receivers ‘mono’ (which record only the displacement) or 2 receivers ‘stereo’
895 (which have the ability to record both the displacement and the velocity, the accuracy on the
896 velocity being 100 times better, looking at the standard deviation). You do not know where
897 the maximum of the initial Gaussian displacement is located. On the other hand, you know
898 the value of the initial $\sigma_0 = 0.02$. Assuming that the hard disk of your receivers are such
899 that you can not record for more that 2 dimensionless time units, find empirically an optimal
900 deployment for your 1D array (which must have, as usual, a constant spacing), in the sense
901 of producing the best 1D profile of displacement after you’ve recorded for 2 time units. Is is
902 better to go ‘mono’ or ‘stereo’?

903 Notes after the fact: we focussed on these issues

- 904 • Filter divergence
- 905 • The more data the better? Up to which point?

906 Lab 3

907 The Kalman filter applied to the 908 vibrating string

909 This lab is also concerned with twin experiments of the kind described at length in lab 2.
910 The main difference stands in the use of the Kalman filter (KF) for the assimilation algo-
911 rithm (Sect. 3.2 in the notes), as opposed to the optimal interpolation (OI) algorithm. This
912 implies that the forecast error covariance matrix will not be frozen anymore, and that it will
913 evolve following the model trajectory. The only subjective part which remains concerns its
914 initialization: the remainder is taken care of by the algorithm.

915 This comes with a price, in terms of computer resources. The propagation of \mathbf{P}^a makes the
916 calculation n times more expensive, if n denotes the size of the state vector \mathbf{x} .

917 The settings are essentially the same as those from lab 2.

918 3.1 Statistical ingredients

919 3.1.1 Model

920 Again, we neglect model errors

$$921 \quad \boldsymbol{\eta} = \mathbf{0}. \tag{3.1}$$

922 Our initial covariance matrix \mathbf{P}_0^a is defined in a way similar to the way we defined the back-
923 ground covariance matrix \mathbf{P}^b in lab 2. We keep a correlation matrix in order to see how things
924 are improved when we use the KF algorithm, as opposed to an OI algorithm.

$$925 \quad \mathbf{P}_0^a = \mathbf{D}^{1/2} \mathbf{C} \mathbf{D}^{1/2}, \tag{3.2}$$

926 where \mathbf{D} is a diagonal matrix holding the variances. Again, we will assume the same value of
927 the variance for every grid point, taking

$$928 \quad \mathbf{D}^{1/2} = \left[\begin{array}{c|c} \sigma_{md} \mathbf{I} & \mathbf{0} \\ \hline \mathbf{0} & \sigma_{mv} \mathbf{I} \end{array} \right], \tag{3.3}$$

929 where σ_{md} and σ_{mv} are the model standard deviation for displacement and velocity, respec-
 930 tively, and \mathbf{I} is the identity matrix of size half the size of the state vector. In the codes, they
 931 are called again `sigmamd` and `sigmamv`. As is the case for the observation standard devia-
 932 tions, these quantities will be defined as a fraction of the maximum amplitude of the initial
 933 displacement A_0 for the first one, and as a fraction of the maximum expected velocity for the
 934 second one.

935 \mathbf{C} is a correlation matrix, defined by

$$936 \quad C_{mn} = \left(1 + al + \frac{1}{3}a^2l^2 \right) \exp(-al), \quad (3.4)$$

937 where a is a tunable parameter and l is the distance between the grid points m and n . a can
 938 be defined as the inverse of the correlation lengthscale,

$$939 \quad a = l_c^{-1}, \quad (3.5)$$

940 The value of l_c is a free parameter as well, and still appears as `lcorr` in the matlab codes.

941 3.1.2 Observations

942 Our observations are strictly the same: displacement and/or velocities are recorded by the
 943 array of receivers (whose properties are described in lab 1, Sect. 1.4). We will assume that
 944 the observation errors are time independent, and that their distribution follows a Gaussian
 945 pdf. The observation error covariance matrix \mathbf{R} is diagonal, of size $2s$, s being the number
 946 of receivers (`nrec` in the code). The diagonal elements of \mathbf{R} hold the variances characterizing
 947 the Gaussian noise affecting each receiver. The standard deviation of the noise affecting each
 948 receiver ir will be denoted by $\sigma_{ir,d}^o$ and $\sigma_{ir,v}^o$ for displacement and velocity, respectively.

949 3.2 Algorithm

950 3.2.1 Preparation of data

951 All the parameters are defined in `param_lab3.m`.

952 We set the duration of the experiment (`duration` in matlab) to a value T . Once \mathbf{x}_0^t is specified
 953 (we will again assume a Gaussian profile for displacement, and zero velocity), we know from
 954 lab1 which resolution N is needed to generate a a clean (that is, not contaminated by a
 955 substantial level of numerical error) true trajectory \mathbf{x}_i^t (once we choose N , the timestep Δt is
 956 automatically computed).

957 The specification of the standard deviation of the observation error σ_{ir} allows us to generate
 958 observations on-the-fly, while computing the sequence of \mathbf{x}_i^t , by the now well-known formula

$$959 \quad \mathbf{y}_i^o = \mathbf{H}\mathbf{x}_i^t + \boldsymbol{\epsilon}_i^o, \quad (3.6)$$

960 in which every component $\epsilon_{ir,i}^o$ of the vector of size s $\boldsymbol{\epsilon}_i^o$ is drawn randomly following a normal
 961 distribution with zero mean and standard deviation σ_{ir}^o (this applies to the displacement
 962 and/or the velocity).

963 The observation operator \mathbf{H} is constructed once and for all once the grid resolution, and the
 964 location of the receivers, are prescribed. Receivers start to operate at time `time0_obs`.

965 3.2.2 Initialization

966 [These two steps are taken care of in the `init_matrices_lab3.m` script]

- 967 • With l_c , σ_{md} and σ_{mv} specified, construct the initial \mathbf{P}_0^a .
- 968 • With the $\sigma_{ir,d}^o$ and $\sigma_{ir,v}^o$ already specified, form the block diagonal matrix

$$969 \quad \mathbf{R} = \text{blkdiag} \left(\left[\begin{array}{ccccc} \sigma_{1,d}^o & 0 & \dots & \dots & 0 \\ 0 & \ddots & \ddots & 0 & \vdots \\ \vdots & \ddots & \sigma_{ir,d}^o & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 & \sigma_{s,d}^o \end{array} \right]^2, \left[\begin{array}{ccccc} \sigma_{1,v}^o & 0 & \dots & \dots & 0 \\ 0 & \ddots & \ddots & 0 & \vdots \\ \vdots & \ddots & \sigma_{ir,v}^o & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 & \sigma_{s,v}^o \end{array} \right]^2 \right) .(3.7)$$

970 3.2.3 Assimilation cycle

971 In addition to your guess \mathbf{P}_0^a , pick your wrong initial condition, \mathbf{x}_0^f . (Our good old Gaussian
 972 profile for the displacement, and zero for the velocity. Again, setting the amplitude of the
 973 Gaussian to 0 amounts to choosing a zero initial condition for the displacement.)

974 While time has not reached its final value T , do the following

- 975 1. given \mathbf{x}_i^a and \mathbf{P}_i^a , use the model \mathbf{M} to compute

- (a) the forecast state

$$977 \quad \mathbf{x}_{i+1}^f = \mathbf{M}\mathbf{x}_i \quad (3.8)$$

- (b) the forecast error covariance

$$979 \quad \mathbf{P}_{i+1}^f = \mathbf{M}\mathbf{P}_i^a\mathbf{M}^T \quad (3.9)$$

- 980 2. If the observations \mathbf{y}_{i+1}^o exist, perform the analysis

- (a) compute the Kalman gain matrix

$$982 \quad \mathbf{K}_{i+1} = \mathbf{P}_{i+1}^f \mathbf{H}^T \left(\mathbf{H}\mathbf{P}_{i+1}^f \mathbf{H}^T + \mathbf{R} \right)^{-1} . \quad (3.10)$$

- (b) compute

$$984 \quad \mathbf{x}_{i+1}^a = \mathbf{K}_{i+1} \left(\mathbf{y}_{i+1}^o - \mathbf{H}\mathbf{x}_{i+1}^f \right) \quad (3.11)$$

- (c) compute

$$986 \quad \mathbf{P}_{i+1}^a = (\mathbf{I} - \mathbf{K}_{i+1}\mathbf{H}) \mathbf{P}_{i+1}^f \quad (3.12)$$

987 Again, the modest size of the problem conveniently allows us to store the entire true and analysed
 988 trajectory, \mathbf{x}_i^t , and \mathbf{x}_i^a .

989 The construction of the trajectory is done in practice by running the `run_KF_lab3.m` script.

3.2.4 A description of the tools at your disposal for this lab

The commands are similar to the ones used in lab 2:

- `param_lab3.m` is the parameter file
- `create_data_lab3.m` generates the true trajectory \mathbf{x}_i^t and the synthetic observations
- `init_matrices_lab3.m` creates \mathbf{P}^b , \mathbf{R} and \mathbf{K}
- `run_KF_lab3.m` generates the predicted trajectory \mathbf{x}_i^a
- `doall_lab3.m` executes `create_data_lab3.m`, `init_matrices_lab3.m`, and `run_OI_lab3.m` in this order
- `compute_chi2_disp.m` computes the following quantities for the displacement field

$$\chi_2\text{disp}_i = (\mathbf{y}_i^o - \mathbf{H}\mathbf{x}_i)^T \mathbf{R}^{-1} (\mathbf{y}_i^o - \mathbf{H}\mathbf{x}_i) \quad \forall i \in \{1, \dots, N_t\}, \quad (3.13)$$

$$\chi_2\text{disp} = \frac{1}{N_t} \sum_i \chi_2\text{disp}_i. \quad (3.14)$$

where we understand that the various vectors and matrices are restricted to their displacement component. In this formula, N_t is the number of time steps.

- `compute_chi2_velo.m` does the same for the velocity field. Note that by summing the two, you will get the total misfit to the data.
- `plot_traj_disp` plots the time space evolution of the displacement component of the true trajectory, the predicted one, and the difference between the two. An example is shown in Fig. 3.1.
- `plot_traj_velo` does the same for the velocity
- `compute_l2diff_state` computes, over the last time unit of the simulation (ie, one travel time) the quantity

$$\sqrt{\int_{T-1}^T (\mathbf{x}^t - \mathbf{x}^a)^T (\mathbf{x}^t - \mathbf{x}^a) dt}, \quad (3.15)$$

which is returned under the name `l2diff`. The script also returns `l2diff_d` and `l2diff_v`, which are the displacement and velocity restrictions of the first one, normalized by A_0 and $A_0/(\sigma_0 \exp(1/2))$, respectively.

- `irec= something` followed by `plot_traces` will plot the traces recorded at receiver `irec` (\mathbf{y}^o), and their prediction ($\mathbf{H}\mathbf{x}^a$). These traces can be useful in diagnosing a filter divergence.
- `plot_variances.m` is a new routine which plots the diagonal elements of $\mathbf{P}_i^a = \mathbf{E}(\boldsymbol{\epsilon}_i^a \boldsymbol{\epsilon}_i^{aT})$ for displacement and velocity as a function of time, normalised using their first value, and using a logarithmic scale. The plots reflect how the pointwise errors vary as a function of time. For instance, you can certainly guess by inspecting Fig. 3.1 where the receivers are located.

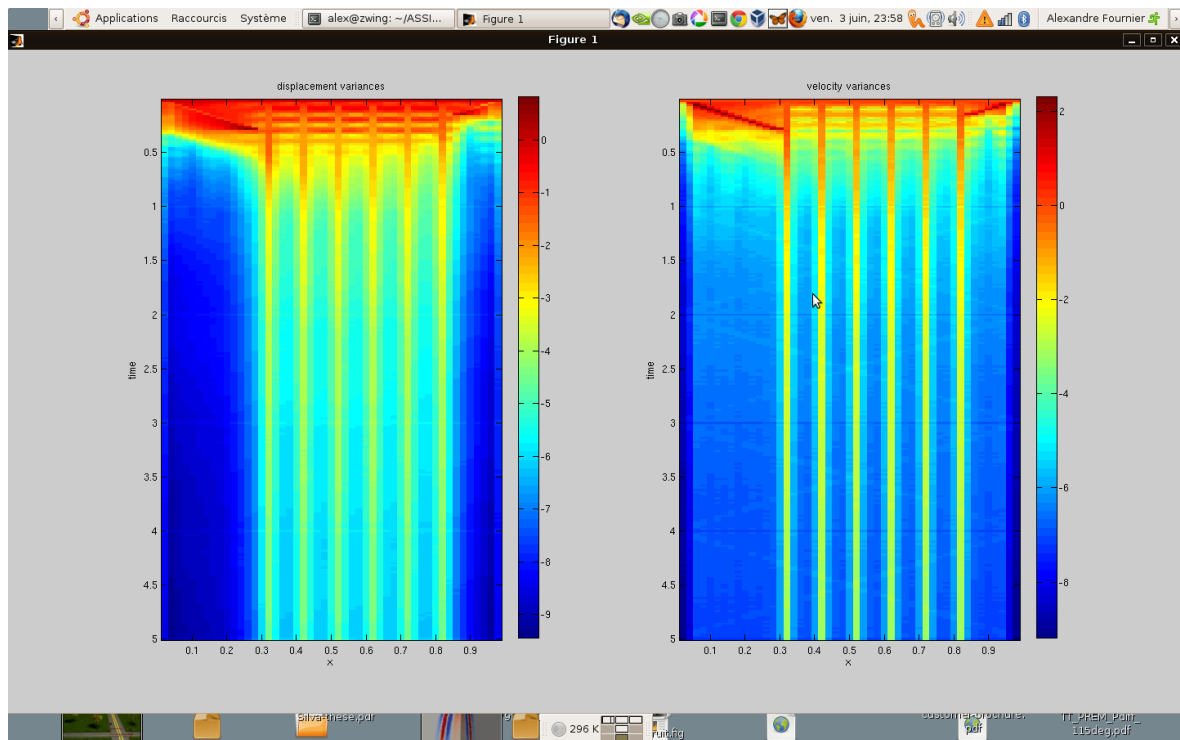


Figure 3.1: A screenshot showing the result of the command `plot_variances`, after application of the Kalman filter algorithm to correct a model trajectory.

1021 3.3 Points to address

1022 There is a lot of flexibility. First get used to running the filter and to diagnosing its behaviour
 1023 (using traces of records, variances, L_2 quantities). Setting $l_c = 0$, try again to find a situation
 1024 where filter divergence occurs. Still using $l_c = 0$, try to find a situation which illustrates the
 1025 benefit of using the KF filter as opposed to the OI scheme (this illustration should not depend
 1026 on our exact knowledge of the true state we are seeking).

1027 Notes after the fact: we focussed again on these issues

- 1028 • Filter divergence
- 1029 • The more data the better? Up to which point?

1030 Cláudio Paulo found a nice set-up for which the Kalman filter clearly outperforms the OI
 1031 scheme

```

1032           % PARAMETER FILE for lab2
1033           %
1034           %   True trajectory
1035           %   -----
1036 duration=10;           % duration of the simulation
1037           % Initial condition: zero velocity, Gaussian profile for displacement
1038 x0=0.7;               % location of the Gaussian peak (between 0 and 1)
1039 sigma0=0.02 ;        % Gaussian standard dev
  
```

```
1040 amp0=0.01 ; % amplitude of the Gaussian
1041 N=100; % number of segments dividing the interval (depends on sigma0 , cf. lab1)
1042 %
1043 % Observations
1044 % -----
1045 xreclft=0.1; % leftmost coordinate of receiver arrays
1046 xrecrigh=0.3; % rightmost coordinate of receiver arrays
1047 deltaxrec=0.1; % spacing between receivers
1048 sigmao_disp=0.0004; % represents a fraction of amp0 (the maximum displacement)
1049 sigmao_velo=0.04; % represents a fraction of the maximum velocity
1050 uniform_array=true;% keep it true (always)
1051 observ_disp= true;% logical (record disp or not?)
1052 observ_velo= false;% logical (record velo or not?)
1053 time0_obs = 0.; % t0 obs
1054 %
1055 % Model statistics
1056 % -----
1057 lcorr=0.01; % correlation length
1058 sigmamd=0.5; % Standard deviation for displacement
1059 sigmamv=0.5; % Standard deviation for velocity
1060 %
1061 % Forecast trajectory: initial condition for displacement
1062 % -----
1063 x0f=0.2; % location of the Gaussian peak (between 0 and 1)
1064 sigma0f=0.04 ; % Gaussian standard dev
1065 amp0f=0.01 ; % amplitude of the Gaussian
```

1066 Bibliography

- 1067 Bennett, A. (2002). *Inverse modeling of the Ocean and Atmosphere*. Cambridge: Cambridge
1068 University Press.
- 1069 Brasseur, P. (2006). Ocean data assimilation using sequential methods based on the Kalman
1070 filter. In E. Chassignet and J. Verron (Eds.), *Ocean Weather Forecasting: An Integrated
1071 View of Oceanography*, pp. 271–316. Springer.
- 1072 Bunge, H.-P., C. Hagelberg, and B. Travis (2003). Mantle circulation models with variational
1073 data assimilation: inferring past mantle flow and structure from plate motion histories and
1074 seismic tomography. *Geophysical Journal International* 152(2), 280–301.
- 1075 Canet, E. (2009). *Modèle dynamique et assimilation de données de la variation scalaire du
1076 champ magnétique terrestre*. Ph. D. thesis, Université Joseph-Fourier.
- 1077 Chaljub, E. and B. Valette (2004). Spectral element modelling of three-dimensional wave
1078 propagation in a self-gravitating earth with an arbitrarily stratified outer core. *Geophysical
1079 Journal International* 158(1), 131–141.
- 1080 Cohn, S., N. Sivakumaran, and R. Todling (1994). A Fixed-Lag Kalman Smoother for Ret-
1081 rospective Data Assimilation. *Monthly Weather Review* 122(12), 2838–2867.
- 1082 Cosme, E., J.-M. Brankart, J. Verron, P. Brasseur, and M. Krysta (2010). Implementation of
1083 a reduced-rank, square-root smoother for high resolution ocean data assimilation. *Ocean
1084 Modelling* 33(1-2), 87–100.
- 1085 Courtier, P. (1997). Variational Methods. *Journal of the Meteorological Society of
1086 Japan* 75(1B), 211–218.
- 1087 Courtier, P. and O. Talagrand (1987). Variational assimilation of meteorological observations
1088 with the adjoint vorticity equation. II: Numerical results. *Quarterly Journal of the Royal
1089 Meteorological Society* 113(478), 1329–1347.
- 1090 Egbert, G. D., A. F. Bennett, and M. G. G. Foreman (1994). TOPEX/POSEIDON tides
1091 estimated using a global inverse model. *Journal of Geophysical Research* 99(C12), 24821–
1092 24852.
- 1093 Evensen, G. (1994). Sequential data assimilation with a nonlinear quasi-geostrophic model
1094 using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Re-
1095 search* 99(C5), 10143–10162.
- 1096 Evensen, G. (2009). *Data assimilation: The ensemble Kalman filter* (2 ed.). Berlin: Springer.

- 1097 Fichtner, A., H. P. Bunge, and H. Igel (2006). The adjoint method in seismology I. Theory.
1098 *Physics of the Earth and Planetary Interiors* 157(1-2), 86–104.
- 1099 Fournier, A., C. Eymin, and T. Alboussière (2007). A case for variational geomagnetic
1100 data assimilation: insights from a one-dimensional, nonlinear, and sparsely observed MHD
1101 system. *Nonlinear Processes in Geophysics* 14, 163–180.
- 1102 Fournier, A., G. Hulot, D. Jault, W. Kuang, A. Tangborn, N. Gillet, E. Canet, J. Aubert,
1103 and F. Lhuillier (2010). An introduction to data assimilation and predictability in geomag-
1104 netism. *Space Science Reviews* 155(1-4), 247–291. 10.1007/s11214-010-9669-4.
- 1105 French, A. P. (1971). *Vibrations and waves*. The M.I.T. Introductory Physics Series. London:
1106 Norton.
- 1107 Ghil, M. and P. Malanotte-Rizzoli (1991). Data assimilation in meteorology and oceanogra-
1108 phy. *Advances in geophysics* 33, 141–266.
- 1109 Giering, R. and T. Kaminski (1998). Recipes for adjoint code construction. *ACM Transactions*
1110 *on Mathematical Software* 24(4), 437–474.
- 1111 Hersbach, H. (1998). Application of the adjoint of the WAM model to inverse wave modeling.
1112 *Journal of geophysical research* 103(C 5), 10469–10487.
- 1113 Ide, K., P. Courtier, M. Ghil, and A. C. Lorenc (1997). Unified notation for data assimilation:
1114 Operational, sequential and variational. *Journal of the meteorological society of Japan* 75,
1115 181–189.
- 1116 Kalnay, E. (2003). *Atmospheric modeling, data assimilation, and predictability*. Cambridge:
1117 Cambridge University Press.
- 1118 Kelbert, A., G. Egbert, and A. Schultz (2008). Non-linear conjugate gradient inversion for
1119 global EM induction: resolution studies. *Geophysical Journal International* 173(2), 365–
1120 381.
- 1121 Komatitsch, D. and J. P. Vilotte (1998). The spectral-element method: an efficient tool to
1122 simulate the seismic response of 2D and 3D geological structures. *Bulletin of the Seismo-*
1123 *logical Society of America* 88(2), 368–392.
- 1124 Kuvshinov, A., J. Velínský, P. Tarits, A. Semenov, O. Pankratov, L. Tøffner-Clausen, Z. Mar-
1125 tinec, N. Olsen, T. J. Sabaka, and A. Jackson (2010). Level 2 products and performances
1126 for mantle studies with swarm. Technical report, ESA.
- 1127 Le Dimet, F.-X. and O. Talagrand (1986). Variational algorithms for analysis and assimilation
1128 of meteorological observations: Theoretical aspects. *Tellus* 38(2), 97–110.
- 1129 Liu, L. and M. Gurnis (2008). Simultaneous inversion of mantle properties and initial condi-
1130 tions using an adjoint of mantle convection. *Journal of Geophysical Research* 113(B8405).
- 1131 Liu, L., S. Spasojevic, and M. Gurnis (2008). Reconstructing Farallon Plate Subduction
1132 Beneath North America Back to the Late Cretaceous. *Science* 322(5903), 934–938.
- 1133 Lorenc, A. C. (1986). Analysis methods for numerical weather prediction. *Quarterly Journal*
1134 *of the Royal Meteorological Society* 112(474), 1177–1194.

- 1135 Miller, R. N., M. Ghil, and F. Gauthiez (1994). Advanced data assimilation in strongly
1136 nonlinear dynamical systems. *Journal of the Atmospheric Sciences* 51(8), 1037–1056.
- 1137 Parker, R. L. (1994). *Geophysical inverse theory*. Princeton, NJ: Princeton University Press.
- 1138 Sambridge, M., P. Rickwood, N. Rawlinson, and S. Sommacal (2007). Automatic differenti-
1139 ation in geophysical inverse problems. *Geophysical Journal International* 170(1), 1–8.
- 1140 Sasaki, Y. (1970). Some basic formalisms in numerical variational analysis. *Monthly Weather*
1141 *Review* 98(12), 875–883.
- 1142 Talagrand, O. (1991). The use of adjoint equations in numerical modelling of the atmospheric
1143 circulation. In A. Griewank and G. G. Corliss (Eds.), *Automatic Differentiation of Algo-*
1144 *rithms: Theory, Implementation, and Application*, Philadelphia, PA, pp. 169–180. Society
1145 for Industrial and Applied Mathematics.
- 1146 Talagrand, O. (1997). Assimilation of observations, an introduction. *Journal of the Meteorolo-*
1147 *gical Society of Japan* 75(1B), 191–209.
- 1148 Talagrand, O. and P. Courtier (1987). Variational assimilation of meteorological observations
1149 with the adjoint vorticity equation. I: Theory. *Quarterly Journal of the Royal Meteorological*
1150 *Society* 113(478), 1311–1328.
- 1151 Tarantola, A. (1984). Inversion of seismic reflection data in the acoustic approximation.
1152 *Geophysics* 49(8), 1259–1266.
- 1153 Tarantola, A. (1988). Theoretical background for the inversion of seismic waveforms including
1154 elasticity and attenuation. *Pure and Applied Geophysics* 128(1), 365–399.
- 1155 Tarantola, A. (2005). *Inverse problem theory and methods for model parameter estimation*.
1156 Philadelphia, PA: Society for Industrial Mathematics.
- 1157 Tromp, J., D. Komatitsch, and Q. Liu (2008). Spectral-element and adjoint methods in
1158 seismology. *Communications in Computational Physics* 3(1-32).
- 1159 Tromp, J., C. Tape, and Q. Liu (2005). Seismic tomography, adjoint methods, time reversal
1160 and banana-doughnut kernels. *Geophysical Journal International* 160(1), 195–216.
- 1161 Wunsch, C. (2006). *Discrete inverse and state estimation problems*. Cambridge: Cambridge
1162 University Press.

1163

Part III

1164

Appendix

Appendix A

Derivation of the discrete adjoint equation

This derivation is rather standard, and can be found in review papers and textbooks (Tala-
grand, 1997; Bennett, 2002; Wunsch, 2006), as well as in the thesis of Canet (2009), in the
specific context of the geomagnetic secular variation. We start with the definition of the misfit
function (Eq. (4.1)), expressed directly as a function of the initial condition \mathbf{x}_0

$$\mathcal{J}(\mathbf{x}_0) = \frac{1}{2} \left\{ \sum_{i=0}^n [\mathcal{H}_i \mathbf{x}_i - \mathbf{y}_i^o]^T \mathbf{R}_i^{-1} [\mathcal{H}_i \mathbf{x}_i - \mathbf{y}_i^o] + [\mathbf{x}_0 - \mathbf{x}^b]^T \mathbf{P}^{b-1} [\mathbf{x}_0 - \mathbf{x}^b] \right\}, \quad (\text{A.1})$$

in which $\mathbf{x}_i = \mathcal{M}_{i,i-1} \cdots \mathcal{M}_{1,0} \mathbf{x}_0$. Any infinitesimal change $\delta \mathbf{x}_0$ in the initial condition \mathbf{x}_0 will
result in a change in \mathbf{x}_i , $\delta \mathbf{x}_i$, which writes to first order

$$\delta \mathbf{x}_i = \mathbf{M}_{i,i-1} \cdots \mathbf{M}_{1,0} \delta \mathbf{x}_0, \quad (\text{A.2})$$

or, in a more compact form,

$$\delta \mathbf{x}_i = \left[\prod_{j=i}^{j=1} \mathbf{M}_{j,j-1} \right] \delta \mathbf{x}_0, \quad (\text{A.3})$$

where $\mathbf{M}_{j,j-1}$ is the tangent linear operator, the Jacobian matrix of local partial derivatives
of the components of \mathbf{x}_j with respect to those of \mathbf{x}_{j-1} . Introducing in a similar manner the
tangent linear approximation \mathbf{H}_i of the observation operator \mathcal{H}_i , we find that a change in the
initial condition $\delta \mathbf{x}_0$ results in a variation of the objective function $\delta \mathcal{J}$ given by

$$\begin{aligned} \delta \mathcal{J} &= \frac{1}{2} \left\{ \sum_{i=0}^n \delta \mathbf{x}_i^T \mathbf{H}_i^T \mathbf{R}_i^{-1} [\mathcal{H}_i \mathbf{x}_i - \mathbf{y}_i^o] + \sum_{i=0}^n [\mathcal{H}_i \mathbf{x}_i - \mathbf{y}_i^o]^T \mathbf{R}_i^{-1} \mathbf{H}_i \delta \mathbf{x}_i \right\} \\ &+ \frac{1}{2} \left\{ \delta \mathbf{x}_0^T \mathbf{P}^{b-1} [\mathbf{x}_0 - \mathbf{x}^b] + [\mathbf{x}_0 - \mathbf{x}^b]^T \mathbf{P}^{b-1} \delta \mathbf{x}_0 \right\}. \end{aligned} \quad (\text{A.4})$$

Because of the symmetry of both \mathbf{R} and \mathbf{P}^b , it is easy to show that

$$\delta \mathbf{x}_i^T \mathbf{H}_i^T \mathbf{R}_i^{-1} [\mathcal{H}_i \mathbf{x}_i - \mathbf{y}_i^o] = [\mathcal{H}_i \mathbf{x}_i - \mathbf{y}_i^o]^T \mathbf{R}_i^{-1} \mathbf{H}_i \delta \mathbf{x}_i \quad (\text{A.5})$$

1183 and that

$$\delta \mathbf{x}_0^T \mathbf{P}^{b-1} [\mathbf{x}_0 - \mathbf{x}^b] = [\mathbf{x}_0 - \mathbf{x}^b]^T \mathbf{P}^{b-1} \delta \mathbf{x}_0. \quad (\text{A.6})$$

1184 Using these two equalities along with the compact notation introduced in Eq. (A.3) yields

$$\delta \mathcal{J} = \left\{ \sum_{i=0}^n [\mathcal{H}_i \mathbf{x}_i - \mathbf{y}_i^o]^T \mathbf{R}_i^{-1} \mathbf{H}_i \prod_{j=i}^{j=1} \mathbf{M}_{j,j-1} \delta \mathbf{x}_0 \right\} + [\mathbf{x}_0 - \mathbf{x}^b]^T \mathbf{P}^{b-1} \delta \mathbf{x}_0. \quad (\text{A.7})$$

1186 Reminding ourselves that $\nabla_{\mathbf{x}_0} \mathcal{J}$ (a row vector) is defined by $\delta \mathcal{J} = \nabla_{\mathbf{x}_0} \mathcal{J} \delta \mathbf{x}_0$, we see that

$$\nabla_{\mathbf{x}_0} \mathcal{J} = \left\{ \sum_{i=0}^n [\mathcal{H}_i \mathbf{x}_i - \mathbf{y}_i^o]^T \mathbf{R}_i^{-1} \mathbf{H}_i \prod_{j=i}^{j=1} \mathbf{M}_{j,j-1} \right\} + [\mathbf{x}_0 - \mathbf{x}^b]^T \mathbf{P}^{b-1}. \quad (\text{A.8})$$

1188 A correction (update) of the initial condition will require to take the transpose of this row
1189 vector

$$\nabla_{\mathbf{x}_0} \mathcal{J}^T = \left\{ \sum_{i=0}^n \prod_{j=1}^{j=i} \mathbf{M}_{j,j-1}^T \mathbf{H}_i^T \mathbf{R}_i^{-1} [\mathcal{H}_i \mathbf{x}_i - \mathbf{y}_i^o] \right\} + \mathbf{P}^{b-1} [\mathbf{x}_0 - \mathbf{x}^b]. \quad (\text{A.9})$$

1191 Let us rewrite this equation in the more inductive following form

$$\begin{aligned} \nabla_{\mathbf{x}_0} \mathcal{J}^T &= \mathbf{M}_{0,1}^T \{ \mathbf{M}_{1,2}^T [\dots [\mathbf{M}_{n-1,n}^T \mathbf{H}_n^T \mathbf{R}_n^{-1} [\mathcal{H}_n \mathbf{x}_n - \mathbf{y}_n^o]] \dots + \mathbf{H}_1^T \mathbf{R}_1^{-1} [\mathcal{H}_1 \mathbf{x}_1 - \mathbf{y}_1^o]] \\ &+ \mathbf{H}_0^T \mathbf{R}_0^{-1} [\mathcal{H}_0 \mathbf{x}_0 - \mathbf{y}_0^o] \} + \mathbf{P}^{b-1} [\mathbf{x}_0 - \mathbf{x}^b]. \end{aligned} \quad (\text{A.10})$$

1192 If one introduces the auxiliary adjoint field \mathbf{a}_i , subject to the terminal condition $\mathbf{a}_{n+1} = \mathbf{0}$,
1193 and whose backward time evolution is governed by

$$\mathbf{a}_{i-1} = \mathbf{M}_{i-1,i}^T \mathbf{a}_i + \mathbf{H}_{i-1}^T \mathbf{R}_{i-1}^{-1} (\mathcal{H}_{i-1} \mathbf{x}_{i-1} - \mathbf{y}_{i-1}^o) + \delta_{i1} \mathbf{P}^{b-1} (\mathbf{x}_{i-1} - \mathbf{x}^b), \quad n \geq i \geq 1, \quad (\text{A.11})$$

1195 the inductive form (A.10) shows why the column vector sought simply writes

$$\nabla_{\mathbf{x}_0} \mathcal{J}^T = \mathbf{a}_0. \quad (\text{A.12})$$

1197 ($\delta_{i1} = 1$ if $i = 1$, 0 otherwise.)