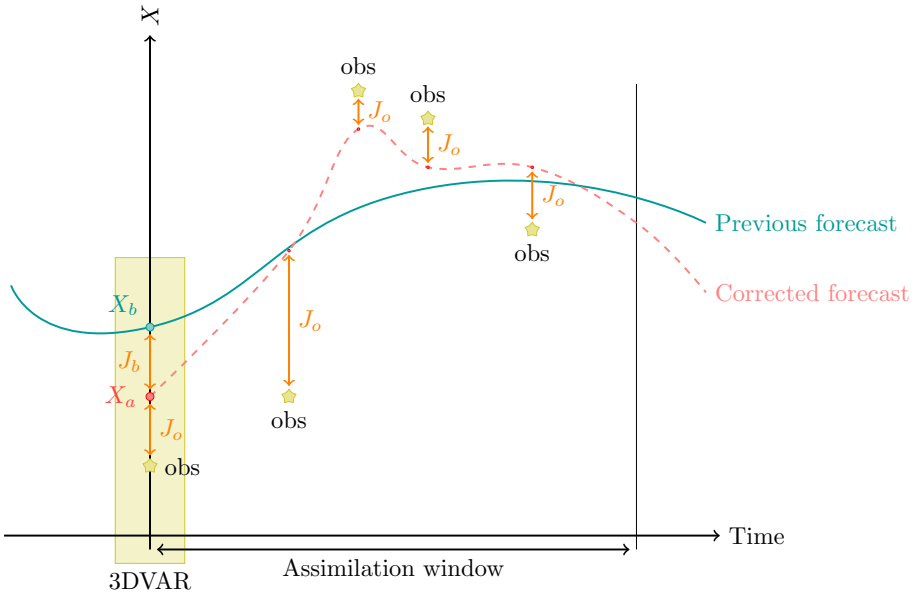
Data Assimilation

Mark Asch - CSU/IMU/2023



Outline of the course (I)

Adjoint methods and variational data assimilation (4h)

1. Introduction to data assimilation: setting, history, overview, definitions.

2. Optimization methods.

3. Adjoint method.

4. Variational data assimilation methods:

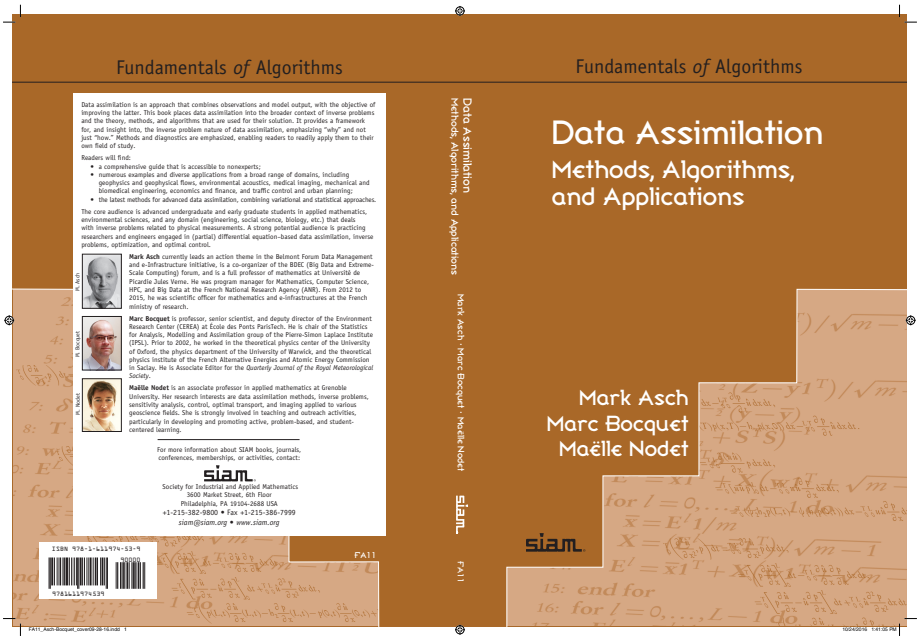
   (a) 3D-Var,
   (b) 4D-Var.

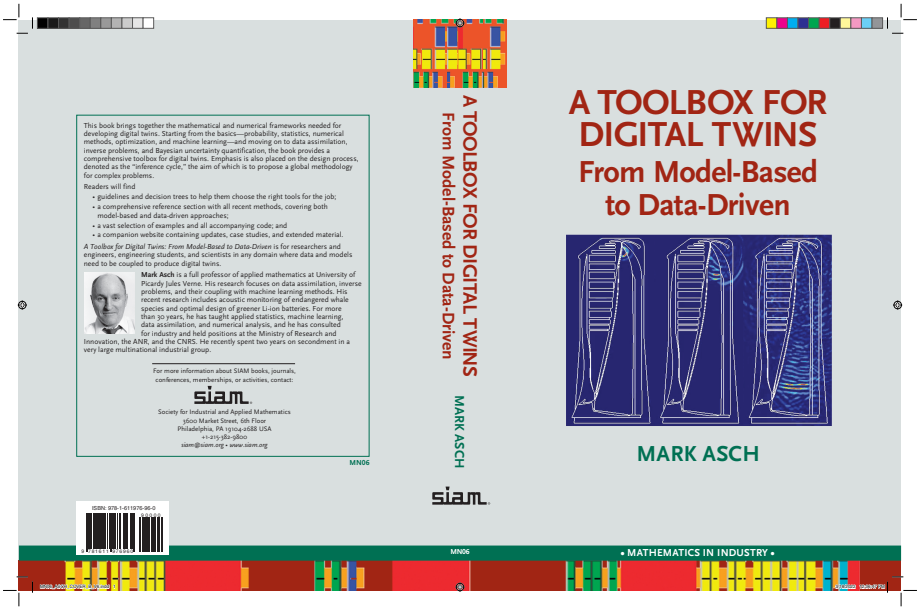
Outline of the course (II)

Statistical estimation, Kalman filters and sequential data assimilation (4h)

1. Introduction to statistical DA.
2. Statistical estimation.
3. The Kalman filter.
4. Nonlinear extensions and ensemble filters.

Reference Textbooks





INTRODUCTION

What is data assimilation?

- Simplest view: a method of combining observations with model output.

- Why do we need data assimilation? Why not just use the observations? (cf. Regression)

  $\Rightarrow$ We want to predict the future!
  - $\rightarrow$ For that we need models.
  - $\rightarrow$ But when models are not constrained periodically by reality, they are of little value.
  - $\rightarrow$ Therefore, it is necessary to fit the model state as closely as possible to the observations, before a prediction is made.

Definition 1. Data assimilation (DA) is the approximation of the true state of some physical system at a given time, by combining time-distributed observations with a dynamic model in an optimal way.

Data assimilation methods

There are two major classes of methods:

1. Variational methods where we explicitly minimize a cost function using optimization methods.

2. Statistical methods where we compute the best linear unbiased estimate (BLUE) by algebraic computations using the Kalman filter.

- They provide the same result in the linear case, which is the only context where their optimality can be rigorously proved.

- They both have difficulties in dealing with non-linearities and large problems.

- The error statistics that are required by both, are in general poorly known.

Introduction: approaches

- DA is an approach for solving a specific class of inverse, or parameter estimation problems, where the parameter we seek is the initial condition.

- Assimilation problems can be approached from many directions (depending on your background/preferences):

  $\Rightarrow$ control theory;
  $\Rightarrow$ variational calculus;
  $\Rightarrow$ statistical estimation theory;
  $\Rightarrow$ probability theory,
  $\Rightarrow$ stochastic differential equations.

- Newer approaches (see Advanced Course): nudging methods, reduced methods, ensemble methods and hybrid methods that combine variational and statistical approaches, Machine/Deep Learning based approaches.

Introduction: applications

1. Navigation: important application of the Kalman filter.

2. Remote sensing: satellite data.

3. Geophysics: seismic exploration, geo-prospection, earthquake prediction.

4. Air and noise pollution, source estimation

5. Weather forecasting.

6. Climatology. Global warming.

7. Epidemiology.

8. Forest fire evolution.

9. Finance.

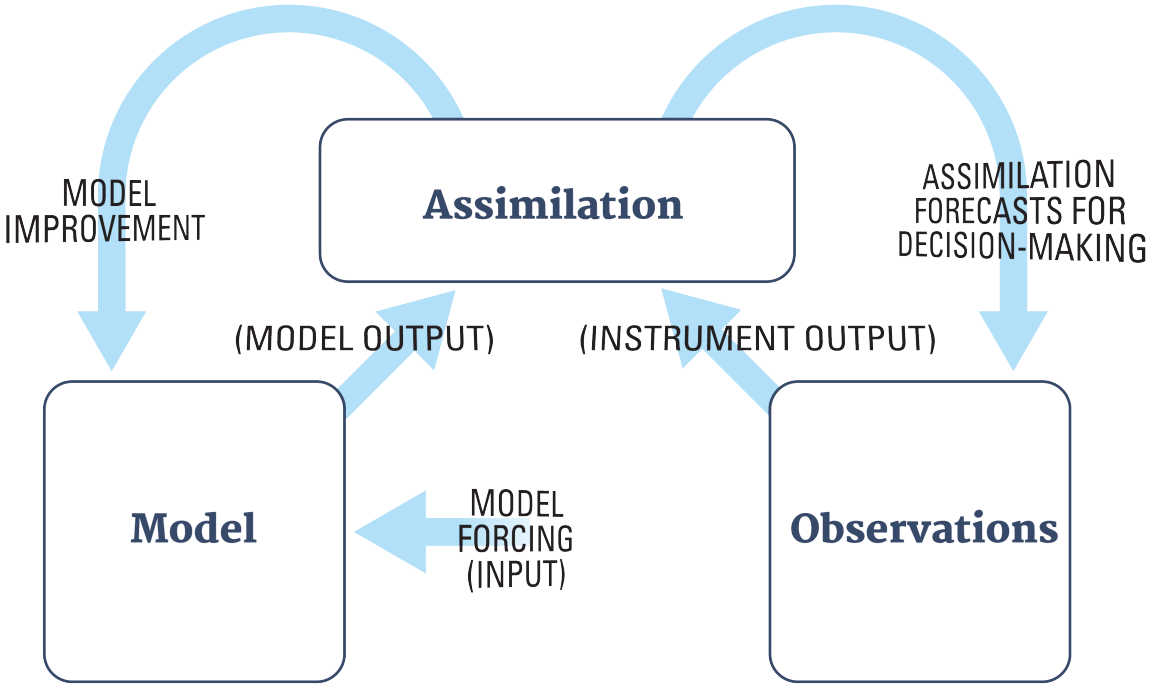
Introduction: nonlinearity...

The problems of data assimilation (in particular) and inverse problems in general arise from:

1. The nonlinear dynamics of the physical model equations.

2. The nonlinearity of the inverse problem.

Introduction: iterative process...



- Closely related to (see Advanced Course)

  $\Rightarrow$ the inference cycle

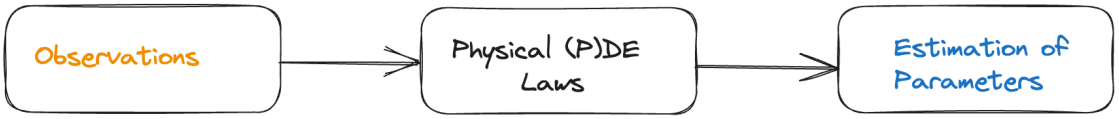
  $\Rightarrow$ machine learning...

FORWARD AND INVERSE PROBLEMS

Forward and Inverse problems

Direct Problem:

Inverse Problem:



- Consider a parameter-dependent dynamical system,

$$\frac{dz}{dt} = g(t, z; \theta), \qquad z(t_0) = z_0,$$

with g known, $\theta \in \Theta$, $z(t) \in \mathbb{R}^k$.

Forward: Given θ, z_0, find $z(t)$ for $t \geq t_0$.

Inverse: Given $z(t)$ for $t \geq t_0$, find $\theta \in \Theta$.

Observations

- Observation equation:

$$f(t, \theta) = \mathcal{H} z(t, \theta),$$

  where $\mathcal{H}$ is the observation operator—to account for the fact that observations are never completely known (in space-time).

- Usually we have a finite number of discrete (space-time) observations

$$\{\tilde{y}_j\}_{j=1}^{n},$$

  where

$$\tilde{y}_j \approx f(t_j, \theta).$$

Model-driven and data-driven inverse problems

- Model-driven:

$$\tilde{y}_j = f(t_j, \theta)$$

- Data-driven:

$$\tilde{y}_j = f(t_j, \theta) + \varepsilon_j,$$

where ε_j is error and requires that we introduce variability/uncertainty into the modeling and analysis.

Well-posedness

1. Existence

2. Uniqueness

3. Continuous dependence of solutions on observations.

✔ The existence and uniqueness together are also known as "identifiability".

✔ The continuous dependence is related to the "stability" of the inverse problem.

Well-posedness (mathematical)

Definition 2. Let X and Y be two normed spaces and let $K : X \to Y$ be a linear or nonlinear map between the two. The problem of finding x given y such that

$$Kx = y$$

is well-posed if the following three properties hold:

WP1 Existence—for every $y \in Y$ there is (at least) one solution $x \in X$ such that $Kx = y$.

WP2 Uniqueness—for every $y \in Y$ there is at most one $x \in X$ such that $Kx = y$.

WP3 Stability—the solution x depends continuously on the data y in that for every sequence $\{x_n\} \subset X$ with $Kx_n \to Kx$ as $n \to \infty$, we have that $x_n \to x$ as $n \to \infty$.

- This concept of ill-posedness will help us to understand and distinguish between direct and inverse

models.

- It will provide us with basic comprehension of the methods and algorithms that will be used to solve inverse problems.

- Finally, it will assist us in the analysis of "what went wrong?" when we attempt to solve the inverse problems.

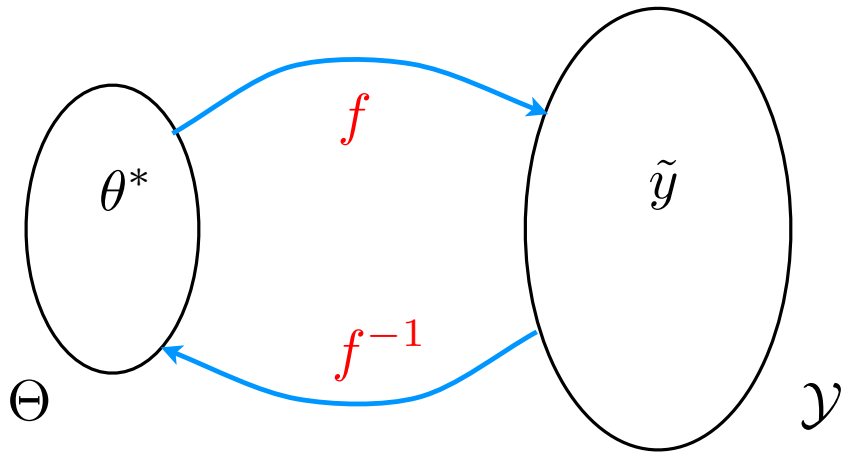
Ill-posedness of inverse problems

All inverse problems are notoriously ill-posed!!!

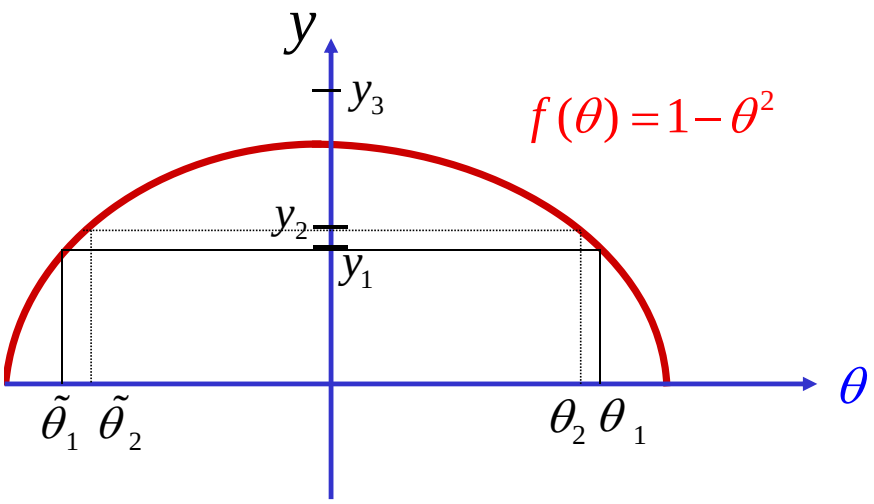
Simplest case: one observation $\tilde{y}$ for $f(\theta)$ and we need to find the pre-image

$$\theta^* = f^{-1}(\tilde{y})$$

for a given $\tilde{y}$.



Simplest case



- ✘ Non-existence: there is no θ_3 such that $f(\theta_3) = y_3$
- ✘ Non-uniqueness: $y_j = f(\theta_j) = f(\widetilde{\theta}_j)$ for $j = 1, 2$.
- ✘ Lack of continuity of inverse map: $|y_1 - y_2|$ small $\not\Rightarrow \left|f^{-1}(y_1) - f^{-1}(y_2)\right| = \left|\theta_1 - \tilde{\theta}_2\right|$ small.

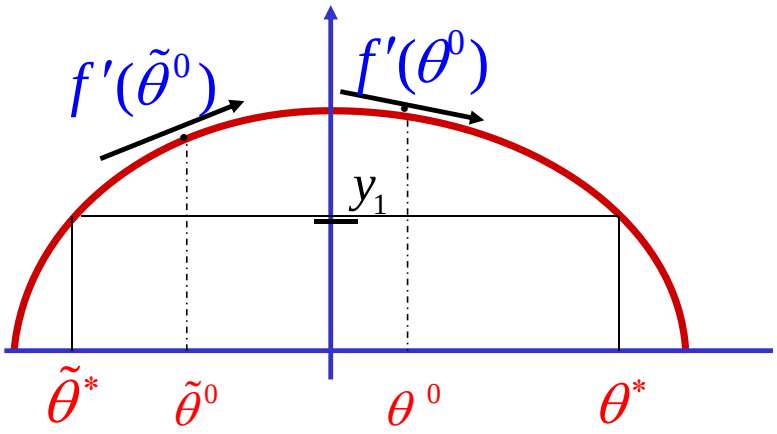
Why is this so important???

Couldn't we just apply a good least squares algorithm (for example) to find the best possible solution?

- Define $J(\theta) = |y_1 - f(\theta)|^2$ for a given y_1
- Apply a standard iterative scheme, such as direct search or gradient-based minimization, to obtain a solution
- Newton's method:

$$\theta^{k+1} = \theta^k - \left[J'(\theta^k)\right]^{-1} J(\theta^k)$$

Newton iterations



- $\theta^{k+1} = \theta^k - \left[J'(\theta^k)\right]^{-1} J(\theta^k), \quad J(\theta) = |y_1 - f(\theta)|^2$
- $J'(\theta) = 2\,(y_1 - f(\theta))\,(-f'(\theta))$

✘ $J'(\theta^0) = 2(-)(--) < 0 \quad \Rightarrow \quad \theta^1 > \theta^0$, etc.

✘ $J'(\tilde{\theta}^0) = 2(-)(-+) > 0 \quad \Rightarrow \quad \tilde{\theta}^1 < \tilde{\theta}^0$, etc.

What went wrong?

- ✘ This behavior is not the fault of steepest descent algorithms.
- ✘ It is a manifestation of the inherent ill-posedness of the problem.
- ✘ How to fix this problem is the subject of much research over the past 50 years!!!
- ✔ Many remedies (fortunately) exist....
  - ✔ explicit and implicit constrained optimizations
  - ✔ regularization and penalization
  - ✔ machine learning...

Example: Tikhonov regularization

Idea is to replace the ill-posed problem for $J(\theta) = |y_1 - f(\theta)|^2$ by a "nearby" problem for

$$J_\beta(\theta) = |y_1 - f(\theta)|^2 + \beta\,|\theta - \theta_0|^2$$

where β is "suitably chosen" regularization/penalization parameter—see below for details.

✔ When it is done correctly, TR provides convexity and compactness.

✘ Even when done correctly, it *modifies the problem* and new solutions may be far from the original ones.

✘ It is not trivial to regularize correctly or even to know if you have succeeded...

Examples of ill-posedness

We illustrate these with 2 numerical examples.

- ✔ Duffing's equation.
- ✔ Estimation of seismic travel time.

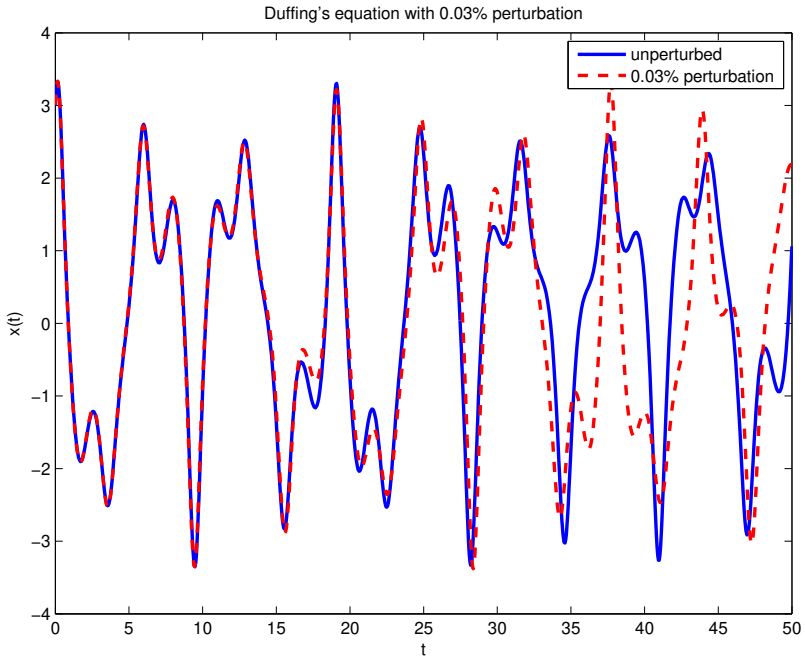
Nonlinearity: Duffing

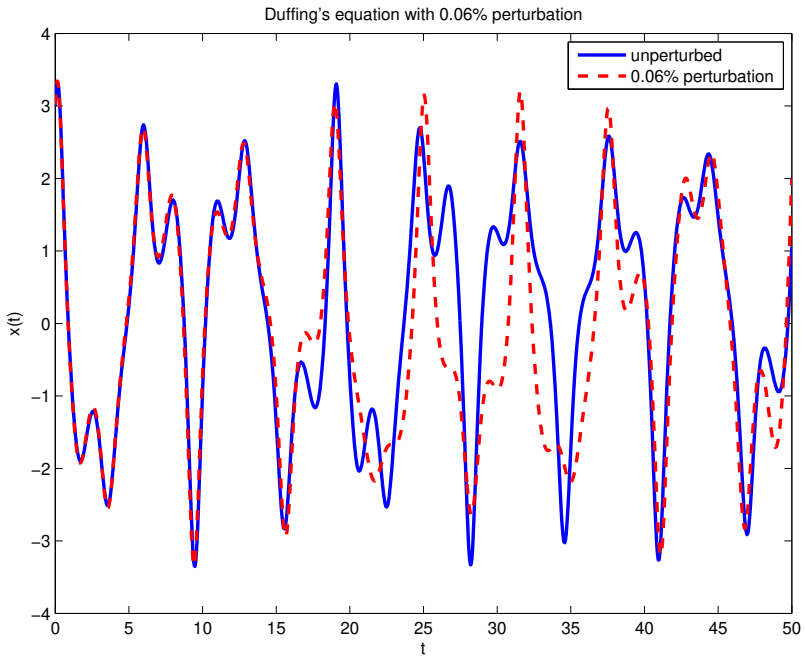
The highly nonlinear Duffing's equation,

$$\ddot{x} + 0.05\dot{x} + x^3 = 7.5\cos t$$

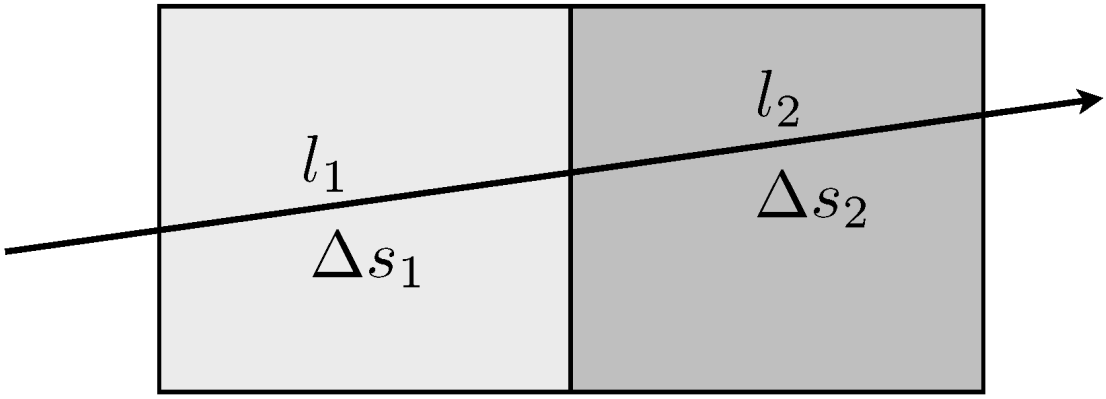
exhibits great sensitivity to the initial conditions. We can observe that two very closely spaced initial states lead to a large discrepancy in the trajectories.

- let $x(0) = 3$ and $\dot{x}(0) = 4$ be the true initial state;
- introduce an error of 0.03% - here we have an accurate forecast until $t = 35$;
- introduce an error of 0.06% - - here we only have an accurate forecast until $t = 20$.


Duffing's equation with 0.03% perturbation
unperturbed
0.03% perturbation
x(t)
t
4
3
2
1
0
−1
−2
−3
−4
0
5
10
15
20
25
30
35
40
45
50


Duffing's equation with 0.06% perturbation
unperturbed
0.06% perturbation
x(t)
t
4
3
2
1
0
−1
−2
−3
−4
0
5
10
15
20
25
30
35
40
45
50

Non uniqueness: seismic travel-time tomography



A signal seismic ray passes through a 2-parameter block model.

- unknowns are the 2 block slownesses (inverse of seismic velocity) $(\Delta s_1, \Delta s_2)$

- data is the observed travel time of the ray, Δt_1

- model is the linearized travel time equation,

$$\Delta t_1 = l_1 \Delta s_1 + l_2 \Delta s_2$$

  where l_j is the length of the ray in the j-th block.

Clearly we have one equation for two unknowns and hence there is no unique solution.

Inverse Problems: General Formulation

- All inverse problems share a common formulation.
- Let the model parameters[1] be a vector (in general, a multivariate random variable), $\mathbf{m}$, and the data be $\mathbf{d}$,

$$\mathbf{m} = (m_1, \ldots, m_p) \in \mathcal{M},$$
$$\mathbf{d} = (d_1, \ldots, d_n) \in \mathcal{D},$$

where

$\Rightarrow$ $\mathcal{M}$ and $\mathcal{D}$ are the corresponding model parameter space and data space.

[1] Applied mathematicians often call the equation $G(m) = d$ a mathematical model and m the parameters. Other scientists call G the forward operator and m the model. We will adopt the more mathematical convention, where m will be referred to as the model parameters, G the model and d the data.

- The mapping $G\colon \mathcal{M} \to \mathcal{D}$ is defined by the direct (or forward) model

$$\mathbf{d} = g(\mathbf{m}), \tag{1}$$

where

$\Rightarrow$ $g \in G$ is an operator that describes the "physical" model and can take numerous forms, such as algebraic equations, differential equations, integral equations, or linear systems.

- Then we can add the observations or predictions, $\mathbf{y} = (y_1, \ldots, y_r)$, corresponding to the mapping from data space into observation space, $H : \mathcal{D} \to \mathcal{Y}$, and described by

$$\mathbf{y} = h(\mathbf{d}) = h\left(g(\mathbf{m})\right),$$

where

$\Rightarrow$ $h \in H$ is the observation operator, usually some projection into an observable subset of $\mathcal{D}$.

- Note that, in addition, there will be some random noise in the system, usually modeled as additive noise, giving the more realistic, stochastic direct model

$$\mathbf{d} = g(\mathbf{m}) + \epsilon, \tag{2}$$

where

$\Rightarrow$ ϵ is a random vector.

Inverse Problems: Classification

- We can now classify inverse problems as:

  $\Rightarrow$ deterministic inverse problems that solve (1) for $\mathbf{m}$,

  $\Rightarrow$ statistical inverse problems that solve (2) for $\mathbf{m}$.

- The first class will be treated by linear algebra and optimization methods.

- The latter can be treated by a Bayesian (filtering) approach, and by weighted least-squares, maximum likelihood and DA techniques

- Both classes can be further broken down into:

  $\Rightarrow$ Linear inverse problems, where (1) or (2) are linear equations. These include linear systems—that are often the result of discretizing (partial) differential equations—and integral equations.

$\Rightarrow$ Nonlinear inverse problems where the algebraic or differential operators are nonlinear.

- Finally, since most inverse problems cannot be solved explicitly, computational methods are indispensable for their solution—see [Asch2022]

- . Also note that we will be inverting here between the model and data spaces, that are usually both of high dimension and thus this model-based inversion will invariably be computationally expensive.

- This will motivate us to employ

  $\Rightarrow$ reduced order methods based on suitable projections,

  $\Rightarrow$ inversion between the data and observation spaces in a purely data-driven approach, using machine learning methods (see Advanced Course)

Tikhonov Regularization: Introduction

- Tikhonov regularization (TR) is probably the most widely used method for regularizing ill-posed, discrete and continuous inverse problems.

- Note that the LASSO and ridge regression methods—see ML Lectures—are special cases of TR,

- The theory is the subject of entire books...

- Recall:

  $\Rightarrow$ the objective of TR is to reduce, or remove, ill-posedness in optimization problems by modifying the objective function.

  $\Rightarrow$ the three sources of ill-posedness: non-existence, non-uniqueness and sensitivity to perturbations.

  $\Rightarrow$ TR, in principle, addresses and alleviates all three sources of ill-posedness and is thus a vital tool for the solution of inverse problems.

Tikhonov Regularization: Formulation

- The most general TR objective function is

$$\mathcal{T}_\alpha(\mathbf{m};\mathbf{d}) = \rho\left(G(\mathbf{m}),\mathbf{d}\right) + \alpha J(\mathbf{m}),$$

  where

  $\Rightarrow$ ρ is the *data discrepancy functional* that quantifies the difference between the model output and the measured data;

  $\Rightarrow$ J is the *regularization functional* that represents some desired quality of the sought for model parameters, usually smoothness;

  $\Rightarrow$ α is the *regularization parameter* that needs to be *tuned*, and determines the relative importance of the regularization term.

- Each domain, each application and each context will require *specific choices* of these three items,

and often we will have to rely either on previous experience, or on some sort of numerical experimentation (trial-and-error) to make a good choice.

- In some cases there exist empirical algorithms, in particular for the choice of α.

Tikhonov Regularization: Discrepancy

The most common *discrepancy functions* are:

- *least-squares*,

$$\rho_{\mathrm{LS}}(\mathbf{d}_1, \mathbf{d}_2) = \frac{1}{2}\|\mathbf{d}_1 - \mathbf{d}_2\|_2^2,$$

- *1-norm*,

$$\rho_1(\mathbf{d}_1, \mathbf{d}_2) = |\mathbf{d}_1 - \mathbf{d}_2|,$$

- *Kullback-Leibler distance*,

$$\rho_{\mathrm{KL}}(d_1, d_2) = \langle d_1, \log(d_1/d_2) \rangle \,,$$

where d_1 and d_2 are considered here as probability density functions. This discrepancy is valid in the Bayesian context.

Tikhonov Regularization: Regularization

The most common *regularization functionals* are derivatives of order one or two.

- *Gradient smoothing*:

$$J_1(\mathbf{m}) = \frac{1}{2}\|\nabla \mathbf{m}\|_2^2,$$

  where ∇ is the gradient operator of first-order derivatives of the elements of $\mathbf{m}$ with respect to each of the independent variables.

- *Laplacian smoothing*:

$$J_2(\mathbf{m}) = \frac{1}{2}\|\nabla^2 \mathbf{m}\|_2^2,$$

  where $\nabla^2 = \nabla \cdot \nabla$ is the Laplacian operator defined as the sum of all second-order derivatives of $\mathbf{m}$ with respect to each of the independent variables.

TR: Computing the Regularization Parameter

- Once the data discrepancy and regularization functionals have been chosen, we need to tune the regularization parameter, α.

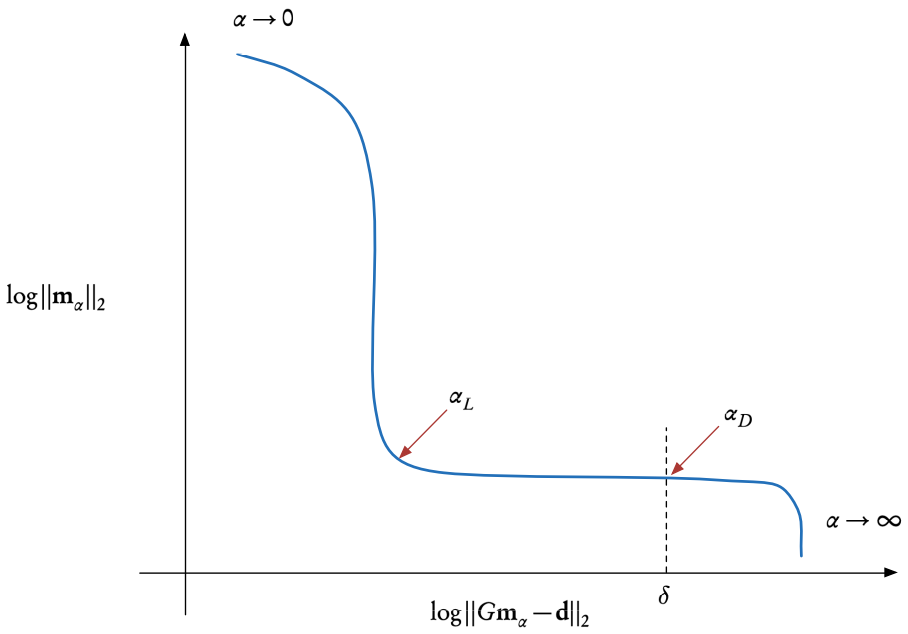
- We have here, similarly to the bias-variance trade-off of ML Lectures, a competition between the discrepancy error and the magnitude of the regularization term.

  $\Rightarrow$ We need to choose, the best compromise between the two.

- We will briefly present three frequently used approaches:

  1. L-curve method.
  2. Discrepancy principle.
  3. Cross-validation and LOOCV.

TR: Computing the Regularization Parameter (II)



- The L-curve criterion is an empirical method for picking a value of α.

  $\Rightarrow$ Since $e_m(\alpha) = \|\mathbf{m}\|_2$ is a strictly decreasing function of α and $e_d(\alpha) = \|G\mathbf{m} - \mathbf{d}\|_2$ is a strictly increasing one,

  $\Rightarrow$ we plot $\log e_m$ against $\log e_d$ we will always obtain an L-shaped curve that has an "elbow" at

the optimal value of $\alpha = \alpha_L$, or at least at a good approximation of this optimal value—see Figure .

$\Rightarrow$ This trade-off curve gives us a visual recipe for choosing the regularization parameter, reminiscent of the bias-variance trade off

$\Rightarrow$ The range of values of α for which one should plot the curve has to be determined by either trial-and-error, previous experience, or a balancing of the two terms in the TR functional.

- The **discrepancy principle**

$\Rightarrow$ choose the value of $\alpha = \alpha_D$ such that the residual error (first term) is equal to an *a priori* bound, δ, that we would like to attain.

$\Rightarrow$ On the L-curve, this corresponds to the intersection with the vertical line at this bound, as shown in Figure.

$\Rightarrow$ A good approximation for the bound is to put $\delta = \sigma\sqrt{n}$, where σ^2 is the variance and n the number of observations.[2] This can be thought

[2] This is strictly valid under the hypothesis of i.i.d. Gaussian noise.

of as the noise level of the data.

$\Rightarrow$ The discrepancy principle is also related to regularization by the truncated singular value decomposition (TSVD), in which case the truncation level implicitly defines the regularization parameter.

- **Cross-validation**, as we explained in ML Lectures, is a way of using the observations themselves to estimate a parameter.

  $\Rightarrow$ We then employ the classical approach of either LOOCV or k-fold cross validation, and choose the value of α that minimizes the RSS (Residual Sum of Squares) of the test sets.

  $\Rightarrow$ In order to reduce the computational cost, a generalized cross validation (GCV) method can be used.

DATA ASSIMILATION

Definitions and notation

- ✔ Analysis is the process of approximating the true state of a physical system at a given time
- ✔ Analysis is based on:
  - ✔ observational data,
  - ✔ a model of the physical system,
  - ✔ background information on initial and boundary conditions.
- ✔ An analysis that combines time-distributed observations and a dynamic model is called data assimilation.

Standard notation

- A discrete model for the evolution of a physical (atmospheric, oceanic, etc.) system from time t_k to time t_{k+1} is described by a dynamic, state equation

$$\mathbf{x}^f(t_{k+1}) = M\left[\mathbf{x}^f(t_k),\right] \tag{3}$$

  $\Rightarrow$ $\mathbf{x}$ is the model's state vector of dimension n,
  $\Rightarrow$ M is the corresponding dynamics operator (finite difference or finite element discretization).

- The error covariance matrix associated with $\mathbf{x}$ is given by $\mathbf{P}$ since the true state will differ from the simulated state (3) by random or systematic errors.

- Observations, or measurements, at time t_k are defined by

$$\mathbf{y}_k^{\mathrm{o}} = H_k\left[\mathbf{x}^t(t_k)\right] + \varepsilon_k,$$

$\Rightarrow$ H is an observation operator

$\Rightarrow$ ε is a white noise process zero mean and covariance matrix $\mathbf{R}$ (instrument errors and representation errors due to the discretization)

$\Rightarrow$ observation vector $\mathbf{y}_k^{\mathrm{o}} = \mathbf{y}^{\mathrm{o}}(t_k)$ has dimension p_k (usually $p_k \ll n.$)

- Subscripts are used to denote the discrete time index, the corresponding spatial indices or the vector with respect to which an error covariance matrix is defined.

- Superscripts refer to the nature of the vectors/matrices in the data assimilation process:

  $\Rightarrow$ "a" for analysis,

  $\Rightarrow$ "b" for background (or 'initial/first guess'),

  $\Rightarrow$ "f" for forecast,

  $\Rightarrow$ "o" for observation and

  $\Rightarrow$ "t" for the (unknown) true state.

Standard notation - continuous system

- Now let us introduce the continuous system. In fact, continuous time simplifies both the notation and the theoretical analysis of the problem. For a finite-dimensional system of ordinary differential equations, the sate and observation equations become

$$\dot{\mathbf{x}}^{\mathrm{f}} = \mathcal{M}(\mathbf{x}^{\mathrm{f}}, t)$$

and

$$\mathbf{y}^{\mathrm{o}}(t) = \mathcal{H}(\mathbf{x}^{\mathrm{t}}, t) + \boldsymbol{\epsilon},$$

where $\dot{(\,)} = \mathrm{d}/\mathrm{d}t$, $\mathcal{M}$ and $\mathcal{H}$ are nonlinear operators in continuous time for the model and the observation respectively.

- This implies that $\mathbf{x}$, $\mathbf{y}$, and $\boldsymbol{\epsilon}$ are also continuous-in-time functions.

- For PDEs, where there is in addition a dependence on space, attention must be paid to the function spaces, especially when performing variational analysis.

- With a PDE model, the field (state) variable is commonly denoted by $\boldsymbol{u}(\mathbf{x}, t)$, where $\mathbf{x}$ represents the space variables (no longer the state variable as above!), and the model dynamics is now a nonlinear partial differential operator,

$$\mathcal{M} = \mathcal{M}\left[\partial_{\mathbf{x}}^{\alpha}, \boldsymbol{u}(\mathbf{x}, t), \mathbf{x}, t\right]$$

  with $\partial_{\mathbf{x}}^{\alpha}$ denoting the partial derivatives with respect to the space variables of order up to $|\alpha| \leq m$ where m is usually equal to two and in general varies between one and four.

CONCLUSIONS

Introduction: conclusions

Data assimilation requires not only the observations and a background, but also knowledge of:

- ✔ error statistics (background, observation, model, etc.)
- ✔ physics (forecast model, model relating observed to retrieved variables, etc.).

The challenge of data assimilation is in combining our stochastic knowledge with our physical knowledge.

Codes

Various open-source repositories and codes are available for both academic and operational data assimilation.

1. DARC: `https://research.reading.ac.uk/met-darc/` from Reading, UK.

2. DAPPER: `https://github.com/nansencenter/DAPPER` from Nansen, Norway.

3. DART: `https://dart.ucar.edu/` from NCAR, US, specialized in ensemble DA.

4. OpenDA: `https://www.openda.org/`.

5. Verdandi: `http://verdandi.sourceforge.net/` from INRIA, France.

6. PyDA: `https://github.com/Shady-Ahmed/PyDA`, a Python implementation for academic use.

7. Filterpy: `https://github.com/rlabbe/filterpy`, dedicated to KF variants.

8. EnKF; `https://enkf.nersc.no/`, the original Ensemble KF from Geir Evensen.

References

1. K. Law, A. Stuart, K. Zygalakis. *Data Assimilation. A Mathematical Introduction*. Springer, 2015.

2. G. Evensen. *Data assimilation, The Ensemble Kalman Filter*, 2nd ed., Springer, 2009.

3. A. Tarantola. *Inverse problem theory and methods for model parameter estimation.* SIAM. 2005.

4. O. Talagrand. Assimilation of observations, an introduction. *J. Meteorological Soc. Japan*, **75**, 191–209, 1997.

5. F.X. Le Dimet, O. Talagrand. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus,* **38**(2), 97–110, 1986.

6. J.-L. Lions. Exact controllability, stabilization and perturbations for distributed systems. *SIAM Rev.*, **30**(1):1–68, 1988.

7. J. Nocedal, S.J. Wright. *Numerical Optimization*. Springer, 2006.

8. F. Tröltzsch. *Optimal Control of Partial Differential Equations*. AMS, 2010.