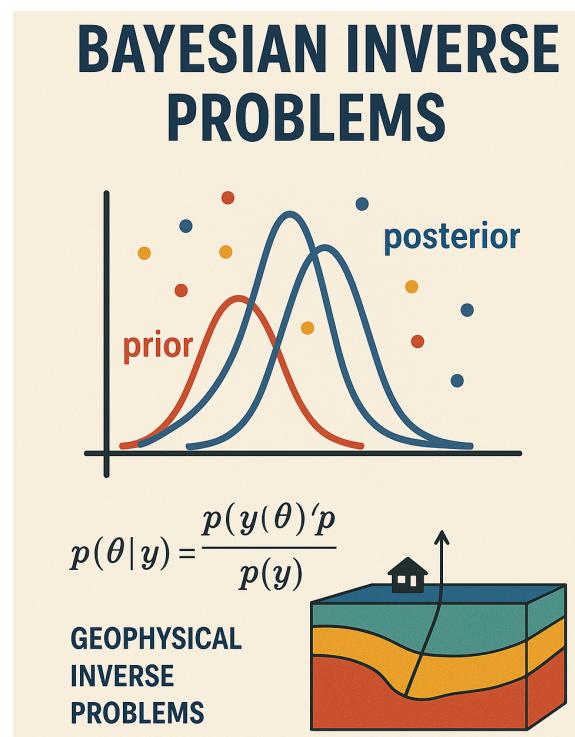


Inverse Problems and Data Assimilation

Mark Asch - MAKUTU/2025



Outline of the course

1. Introduction to inverse problems and data assimilation: overview, setting, history, definitions, examples.
2. Bayesian inverse problems.
 - (a) Bayesian inference.
 - (b) Bayesian/Statistical inversion theory.
3. Posterior Estimation methods.
 - (a) Monte Carlo methods.
 - (b) McMC and variants for posterior estimation.
 - (c) V.I. for posterior estimation.
4. Statistical estimation, Kalman filters and sequential data assimilation.
 - (a) Introduction to statistical DA.

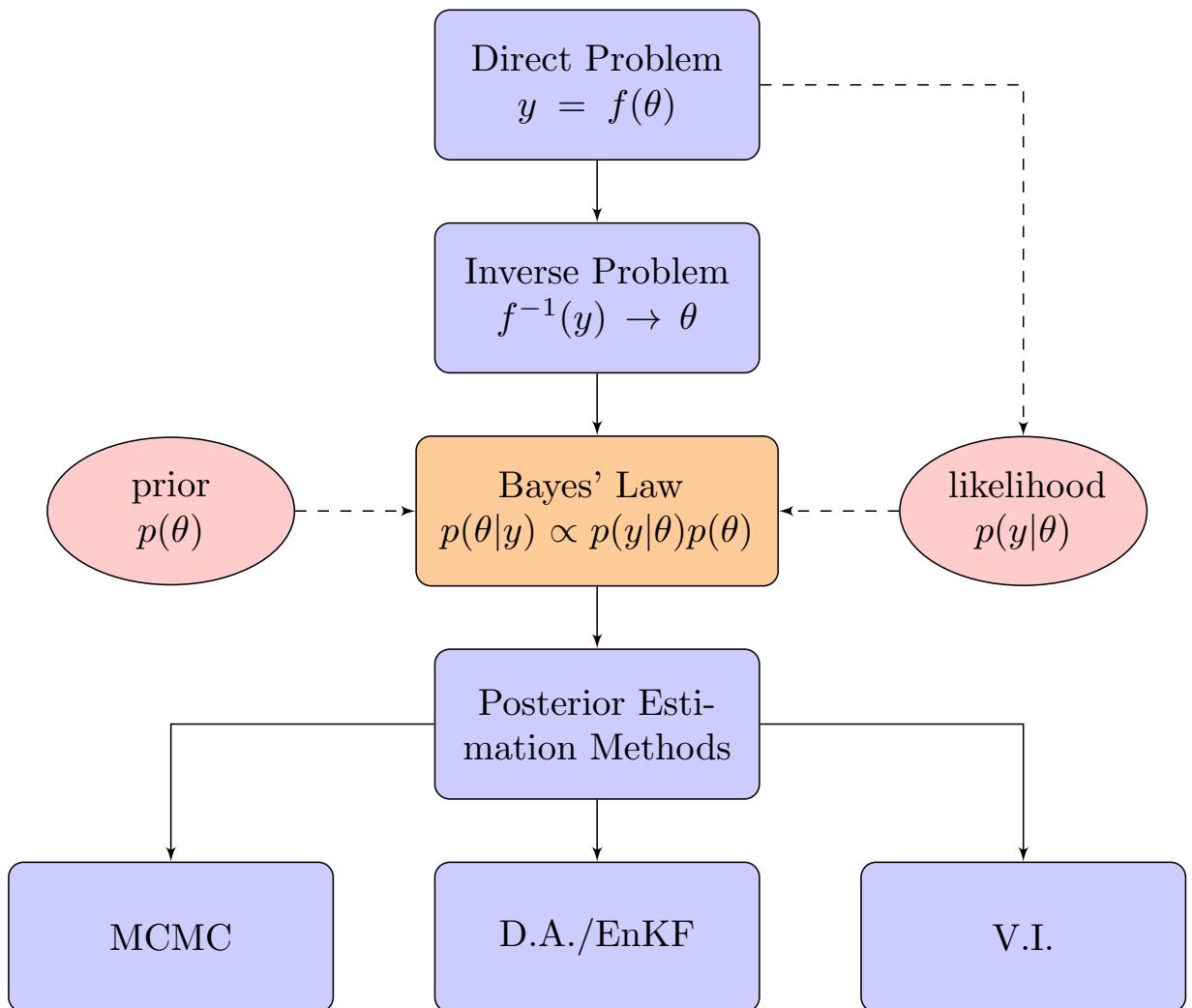
- (b) The Kalman filter and Ensemble KF.
- (c) Ensemble Kalman Inversion (EKI).

2 Reference Textbooks



INTRODUCTION

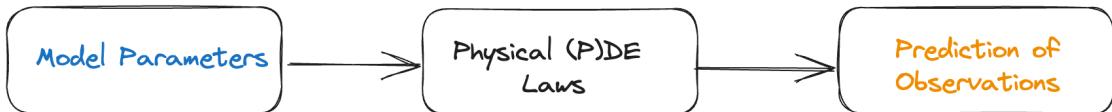
OVERVIEW



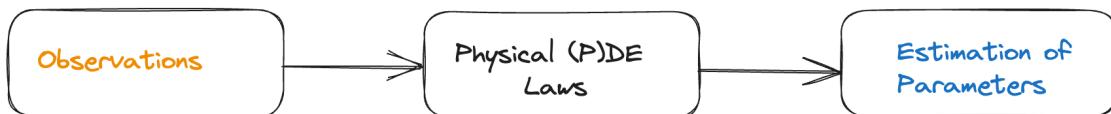
FORWARD AND INVERSE PROBLEMS

Classical Forward and Inverse Problems

Direct Problem:



Inverse Problem:



- Consider a parameter-dependent dynamical system,

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

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 $\theta \in \Theta$.

Stochastic Forward and Inverse Problems

- All real-life systems are subject to noise—in the “physics”, in the measurements, in the model, in the simulations.
- Bayes’ Theorem provides a posterior probability density that captures ALL the uncertainty.

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 1. Let X and Y be two normed spaces and let $K : X \rightarrow Y$ 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 $Kx_n \rightarrow Kx$ as $n \rightarrow \infty$, we have that $x_n \rightarrow x$ as $n \rightarrow \infty$.

- 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.

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} ,

$$\mathbf{m} = (m_1, \dots, m_p) \in \mathcal{M},$$

$$\mathbf{d} = (d_1, \dots, d_n) \in \mathcal{D},$$

where

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

¹Applied 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 $G: \mathcal{M} \rightarrow \mathcal{D}$ is defined by the **direct** (or forward) model

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

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, $\mathbf{y} = (y_1, \dots, y_r)$, corresponding to the mapping from data space into observation space, $H: \mathcal{D} \rightarrow \mathcal{Y}$, and described by

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

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

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

where

$\Rightarrow \epsilon$ is a random vector.

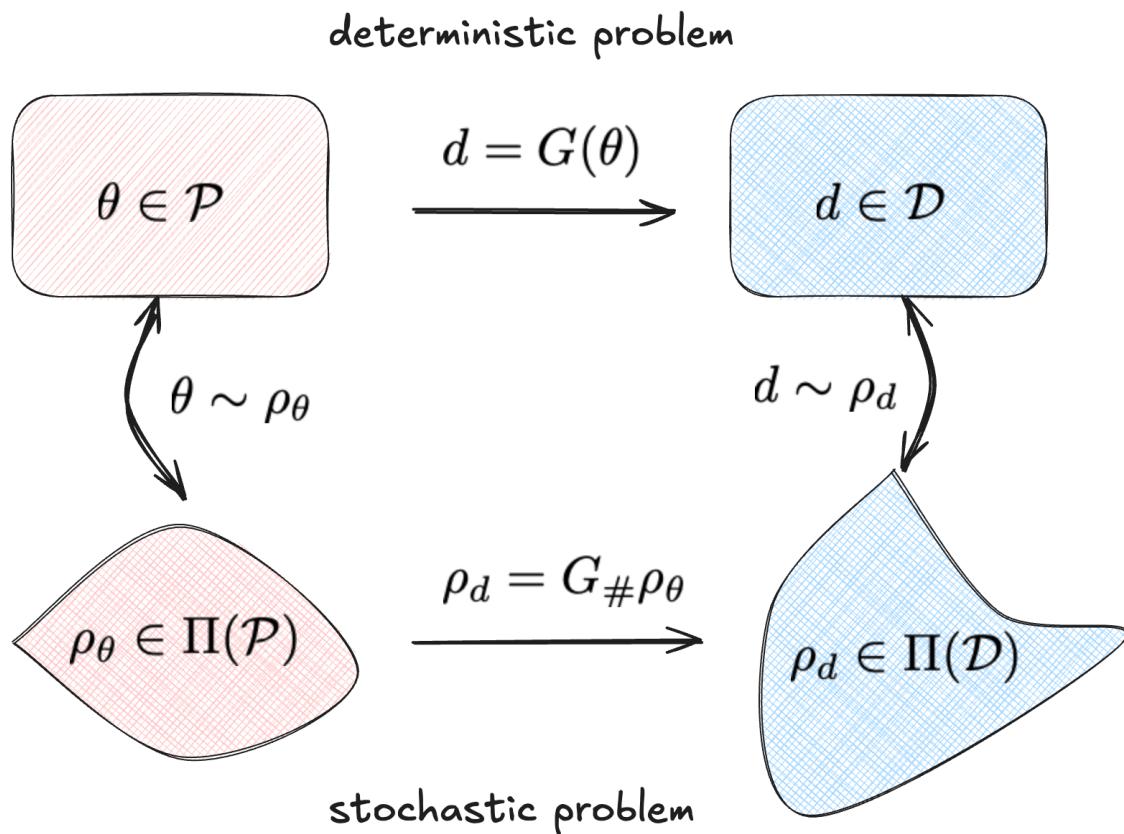
Inverse Problems: Classification

- We can now classify inverse problems as:
 - ⇒ deterministic inverse problems that solve (1) for \mathbf{m} ,
 - ⇒ statistical/stochastic 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 also by
 - ⇒ weighted least-squares,
 - ⇒ maximum likelihood,
 - ⇒ DA,
 - ⇒ optimal transport techniques.
- Both classes can be further broken down into:

- ⇒ **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.
 - ⇒ **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
 - ⇒ **reduced order methods** based on suitable projections,
 - ⇒ inversion between the data and observation spaces in a purely data-driven approach, using

machine learning methods...

Deterministic vs. Stochastic Inverse Problems



Remark 1. The stochastic inverse case could be considered as an **optimal transport** problem!

DETERMINISTIC INVERSE PROBLEMS FOR DE's

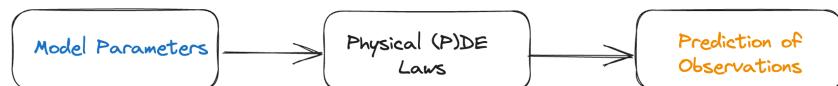
(DIP)

DIP Overview

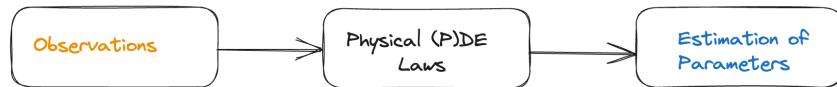
Definition 2. *Deterministic inverse problems* (DIP) involve recovering unknown parameters, coefficients, or initial/boundary conditions of a differential equation from observations of its solution.

- Unlike forward (direct) problems where we compute solutions from a given, complete problem specification, inverse problems work “backwards” from observed data to infer the underlying model parameters.

Direct Problem:



Inverse Problem:



DIP Mathematical Framework

- The Direct (Forward) Problem

Let u denote the state variable (solution) and θ denote parameters/coefficients. The direct problem is:

Given: Parameter vector $\theta \in \Theta \subset \mathbb{R}^p$

Find: Solution $u \in U$ satisfying

$$\mathcal{L}(u; \theta) = f \quad \text{in } \Omega,$$

$$\mathcal{B}(u; \theta) = g \quad \text{on } \partial\Omega,$$

where

- \mathcal{L} is a differential operator (ODE or PDE) ,
- \mathcal{B} represents boundary/initial conditions,

- Ω is the spatial domain,
- U is an appropriate function space (e.g., $H^1(\Omega)$, $C^2(\Omega)$).

We denote the **parameter-to-solution map** (forward operator)

$$\mathcal{G} : \Theta \rightarrow U, \quad \theta \mapsto u(\theta).$$

- The **Inverse Problem**

Given: Observations $d \in \mathbb{R}^m$ of the solution

Find: Parameter $\theta^* \in \Theta$ such that the solution matches observations that are related to the solution through an **observation operator** $\mathcal{H} : U \rightarrow \mathbb{R}^m$

$$d = \mathcal{H}(u(\theta^*)) + \eta,$$

where η represents measurement noise, taken as zero here.

- The inverse problem seeks to determine θ^* from knowledge of d , which amounts to inverting the composed operator

$$\mathcal{M} = \mathcal{H} \circ \mathcal{G} : \Theta \rightarrow \mathbb{R}^m.$$

DIP Optimization Formulation

Since inverse problems are typically **ill-posed** (in Hadamard's sense: existence, uniqueness, or stability may fail), we reformulate as an optimization problem.

- Least Squares Formulation

⇒ Minimize the **data misfit functional**

$$J(\theta) = \frac{1}{2} \|d - \mathcal{G}(u(\theta))\|^2,$$

where $\|\cdot\|$ is typically the ℓ^2 norm in \mathbb{R}^m .

- Tikhonov Regularization

⇒ To ensure well-posedness and incorporate prior information

$$J_\alpha(\theta) = \frac{1}{2} \|d - \mathcal{G}(u(\theta))\|^2 + \frac{\alpha}{2} \|\mathcal{R}(\theta - \theta_0)\|^2,$$

where

- $\alpha > 0$ is the **regularization parameter** (see APPENDIX for details),
 - \mathcal{R} is a regularization operator (e.g., identity, gradient, Laplacian),
 - θ_0 is a prior estimate.
-

- Optimization Methods
⇒ Gradient-Based Methods

1. Steepest Descent:

$$\theta^{k+1} = \theta^k - \beta_k \nabla J(\theta^k)$$

2. Gauss-Newton Method: Linearize $\mathcal{M}(\theta) \approx \mathcal{M}(\theta^k) + \mathcal{M}'(\theta^k)(\theta - \theta^k)$ and solve

$$(\mathcal{M}'(\theta^k)^T \mathcal{M}'(\theta^k) + \alpha I) \delta \theta^k = \mathcal{M}'(\theta^k)^T (d - \mathcal{M}(\theta^k))$$

$$\theta^{k+1} = \theta^k + \delta\theta^k$$

3. Adjoint Method (for PDE-constrained optimization): Compute gradients efficiently by solving an adjoint equation (avoids computing full Jacobian)—see below.

- Derivative-Free Methods
 - ⇒ Ensemble Kalman Filter (EnKF).
 - ⇒ Genetic algorithms.
 - ⇒ Bayesian optimization.

Example: ODE Inverse Problem

- Direct Problem: Consider a first-order decay model

$$\frac{du}{dt} = -\theta u(t), \quad t \in [0, T]$$

$$u(0) = u_0$$

where $\theta > 0$ is the unknown decay rate. The analytical solution is

$$u(t; \theta) = u_0 e^{-\theta t}$$

- Inverse Problem

⇒ Given: Measurements d_i at times t_i for $i = 1, \dots, m$

$$d_i = u(t_i; \theta^*) + \eta_i$$

⇒ Find: θ^*

- Optimization Problem

$$\min_{\theta > 0} J(\theta) = \frac{1}{2} \sum_{i=1}^m (d_i - u_0 e^{-\theta t_i})^2$$

- Gradient Computation

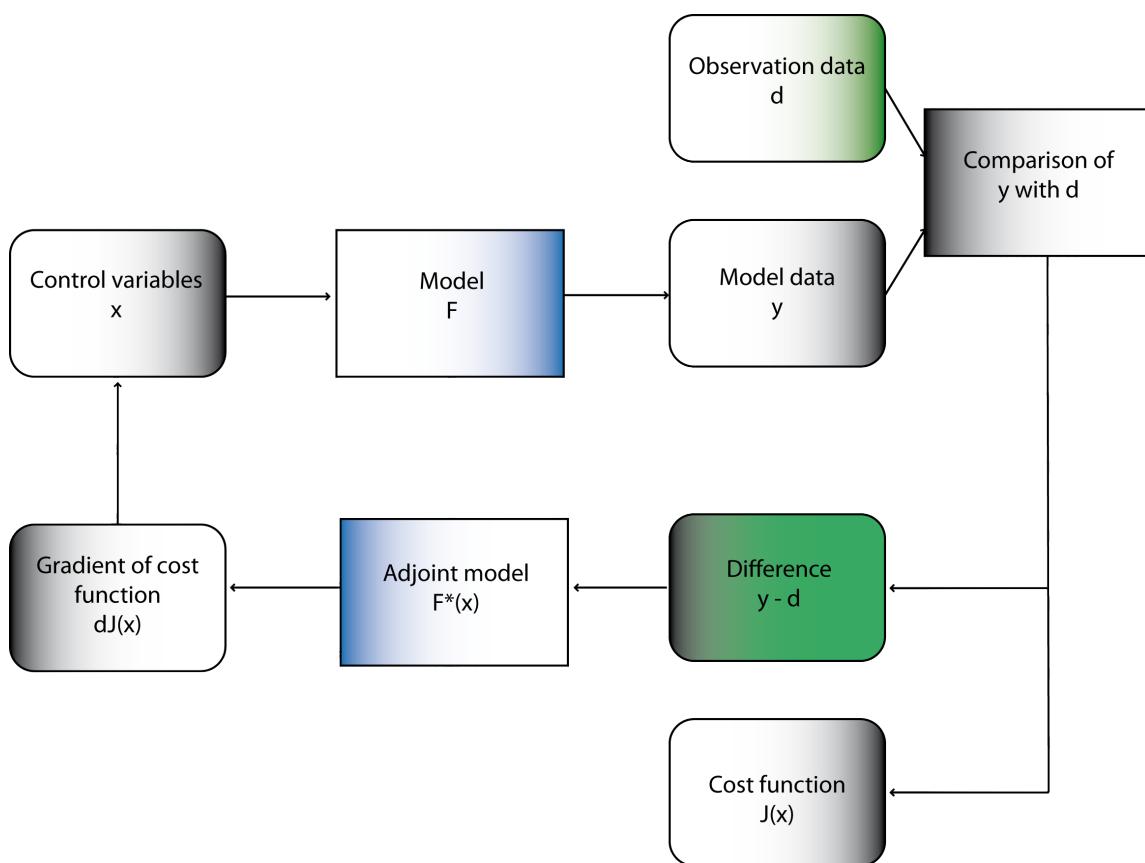
$$\frac{dJ}{d\theta} = - \sum_{i=1}^m (d_i - u_0 e^{-\theta t_i}) \cdot u_0 t_i e^{-\theta t_i}$$

- Solution: Set to zero and solve, gives a nonlinear equation for θ , typically solved iteratively (e.g., modified Newton's method).

ADJOINT METHOD FOR DIP

Adjoint Methods (I)

- A very general approach for solving inverse problems... including Machine Learning!
- Variational DA is based on an adjoint approach.



Adjoint Methods (II) - definition

Definition 3. *An adjoint method is a general mathematical technique, based on variational calculus, that enables the computation of the gradient of an objective, or cost functional with respect to the model parameters in a very efficient manner.*

- This method will be addressed in the Practical session.
- See also the lecture material in [11_DA_adj.pdf](#)

BAYESIAN INFERENCE

Bayes' Law

- **Independence and Conditioning:**

Definition 4 (Independence). Two events, A and B , are **independent** if

$$P(A \cap B) = P(A)P(B).$$

Definition 5 (Conditional Probability). The **conditional probability** of A on B is the probability that A occurs provided (or knowing) that B has occurred, and is given by

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

- It follows that if A and B are **mutually independent**, then

$$P(A | B) = P(A), \quad P(B | A) = P(B).$$

- We can now state **Bayes' Theorem**
 - ⇒ in discrete form, for events
 - ⇒ in continuous form, for probability distributions

Theorem 1 (Bayes' Theorem for Two Events). *Let A and B be two events in Ω , then*

$$P(A | B) = \frac{P(B | A)P(A)}{P(B)}$$

or

$$P(B | A) = \frac{P(A | B)P(B)}{P(A)}.$$

- A more general situation is where Ω is subdivided into a **partition of events**, such as blood types in a human population.

Theorem 2 (Bayes' Theorem for a Partition). *Let A_1, A_2, \dots, A_k be a partition of Ω with $P(A_i) > 0$ for each i . If $P(B) > 0$, then, for each $i = 1, \dots, k$,*

$$P(A_i | B) = \frac{P(B | A_i)P(A_i)}{P(B)},$$

where the total probability

$$P(B) = \sum_{j=1}^k P(B \mid A_j)P(A_j).$$

Bayesian Inference

In the most general case, where we want to perform **Bayesian inference** for the **estimation of parameters** (an inverse problem!), we simply replace the probabilities by the corresponding **density functions**.

- Then Bayesian inference is performed in **three steps**:
 1. Choose a probability density $f(\theta)$, called the **prior distribution**, that expresses our beliefs, or prior experimental or historical knowledge, about a parameter θ before we see any data.
 2. Choose a statistical model $f(x | \theta)$ that reflects our beliefs about x given θ . Notice that this is expressed as a conditional probability, called the **likelihood function**, and not as a joint probability function.
 3. After observing data x_1, \dots, x_n , update our beliefs and calculate the **posterior distribution** $f(\theta | x_1, \dots, x_n)$.

- Let us look more closely at the three components of Bayes' Law.

Definition 6 (Prior Distribution). For a given statistical model that depends on a parameter θ , considered as random, the distribution assigned to θ before observing the other random variables of interest is called the *prior distribution*. This is just the marginal distribution of the parameter.

Definition 7. [Posterior Distribution] For a statistical inference problem, with parameter θ and random sample X_1, \dots, X_n , the conditional distribution of θ given $X_1 = x_1, \dots, x_n = X_n$ is called the *posterior distribution* of θ .

Definition 8 (Likelihood Function). Suppose that X_1, X_2, \dots, X_n have a joint density function

$$f(X_1, X_2, \dots, X_n | \theta).$$

Given the observations $X_1 = x_1, X_2 = x_2, \dots,$

$X_n = x_n$, the likelihood function of θ is

$$L(\theta) = L(\theta \mid x_1, x_2, \dots, x_n) = f(x_1, x_2, \dots, x_n \mid \theta).$$

If the X_i are i.i.d. with density $f(X_i \mid \theta)$, , then the joint density is a product and

$$L(\theta \mid x_1, x_2, \dots, x_n) = \prod_{i=1}^n f(x_i \mid \theta).$$

- We point out the following properties of the likelihood:
 - ⇒ The likelihood is **not** a probability density function and can take values outside the interval $[0, 1]$.
 - ⇒ Likelihood is an important concept in both frequentist and Bayesian statistics.
 - ⇒ Likelihood is a measure of the extent to which a sample provides **support** for particular values of a parameter in a parametric model—this will be very important when we will deal with parameter estimation, and **inverse problems** in general.

- ⇒ The likelihood measures the support (**evidence**) provided by the data for each possible value of the parameter. This means that if we compute the likelihood function at two points, $\theta = \theta_1$, $\theta = \theta_2$, and find that $L(\theta_1 | x) > L(\theta_2 | x)$, then the sample observed is more likely to have occurred if $\theta = \theta_1$. We say that θ_1 is a more plausible value for θ than θ_2 .
- ⇒ For i.i.d. random variables, the **log-likelihood** is usually used, since it reduces the product to a sum.

Bayes' Theorem

- We now formulate the general version of **Bayes' Theorem**.

Theorem 3. Suppose that n random variables, X_1, \dots, X_n , form a random sample from a distribution with density, or probability function in the case of a discrete distribution, $f(x | \theta)$. Suppose also that the unknown parameter, θ , has a prior pdf $f(\theta)$. Then the posterior pdf of θ is

$$f(\theta | x) = \frac{f(x_1 | \theta) \cdots f(x_n | \theta) f(\theta)}{f_n(x)}, \quad (3)$$

where $f_n(x)$ is the marginal joint pdf of X_1, \dots, X_n .

- In this theorem,
 - ⇒ the *prior*, $f(\theta)$, represents the credibility of, or belief in the values of the parameters

we seek, without any consideration of the data/observations;

- ⇒ the *posterior*, $f(\theta | x)$, is the credibility of the parameters with the data taken into account;
- ⇒ $f(x | \theta)$, considered as a function of θ , is the *likelihood* function, which is the probability that the data/observation could be generated by the model with a given value of the parameter;
- ⇒ the denominator, called the *evidence*, $f_n(x)$, is the *total probability* of the data taken over all the possible parameter values, also called the *marginal likelihood*, or the marginal, and can be considered as a normalization factor;
- ⇒ the posterior distribution is thus proportional to the product of the likelihood and the prior distribution, or, in applied terms,

$$f(\text{parameter} | \text{data}) \propto f(\text{data} | \text{parameter}) f(\text{parameter}).$$

- What can one do with the posterior distribution thus obtained? The answer is a lot of things, in fact a **complete quantification of the incertitude** of the parameter's estimation is possible. We can compute:

- ⇒ Point estimates by summarizing the center of the posterior. Typically, these are the posterior mean or the posterior mode.
- ⇒ Interval estimates for a given level α —see below.
- ⇒ Estimates of the probability of an event, such as $P(a < \theta < b)$ or $P(\theta > b)$.
- ⇒ Posterior quantiles.

Recap of Bayesian Setting

We seek the **posterior density** of the parameters θ

$$\pi(\theta) := p(\theta|y) \propto \mathcal{L}(y, f(\theta))p(\theta).$$

Ingredients:

- Parameters $\theta \in \mathbb{R}^d$, data $y \in \mathbb{R}^n$.
- Prior density $p(\theta): \mathbb{R}^d \rightarrow \mathbb{R}^+$.
- Forward model $f: \mathbb{R}^d \rightarrow \mathbb{R}^n$.
 - ⇒ Can be a black-box function.
 - ⇒ Each evaluation is **expensive**.
- Likelihood function $\mathcal{L}: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^+$.
 - ⇒ $\mathcal{L}(y, f(\theta)) = p(y|\theta)$
 - ⇒ Each evaluation requires an evaluation of f .

EXAMPLES

Inverse Problems - Application Domains

- *Computational imaging*—recovering the true image from a blurred and noisy observation.
- *Geophysics*—inferring the conductivity of the subsurface from measurements at wells; inferring the geological substructure from measurements on the earth's surface, or the seafloor surface.
- *Machine learning*—building an underlying model from observed data points.
- *Others*: non-destructive testing, astrophysics, medicine, weather prediction (DA), . . .

Inverse Problem Examples

Shared Mathematical Formulation

Inverse Problem

Given observation data $y \in Y$, determine the unknown $u \in U$ such that

$$y = \mathcal{G}(u) + \eta,$$

where \mathcal{G} is a mathematical model of a process and η is random (observational) noise.

Inverse Problem Examples

Example 1: Imaging

- **Goal:** reconstruct an image u given a noisy, partial observation y .



Inverse Problem Examples

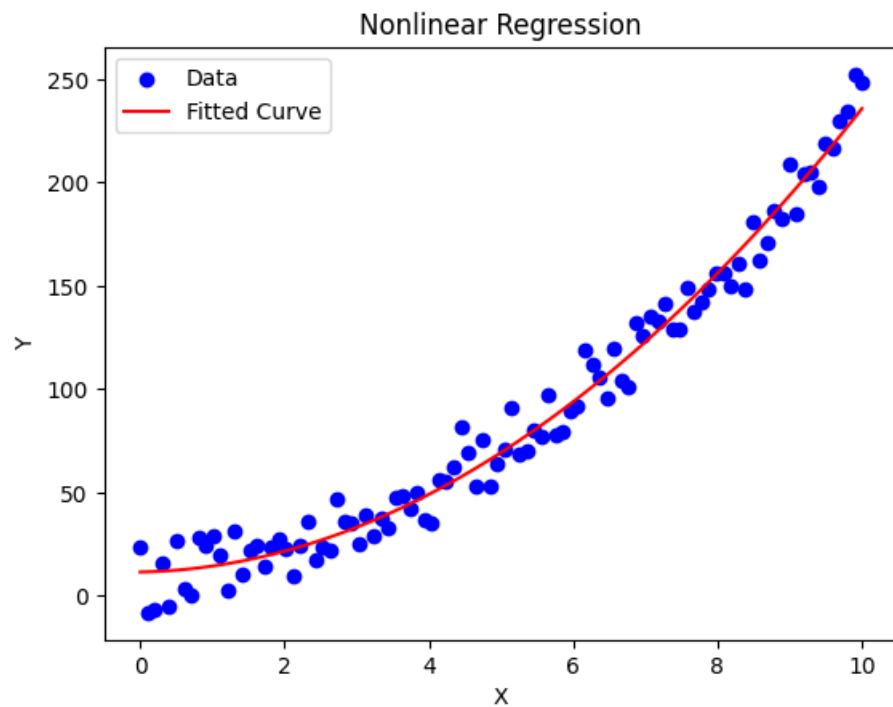
Example 1: Imaging

- **Goal:** reconstruct an image u given a noisy, partial observation y .
- **Unknown:** $u \in \mathbb{R}^{d_u}$, the pixel values of the image.
- **Map:** $\mathcal{G} = G \in \mathbb{R}^{d_y \times d_u}$, linear in many imaging problems, where G incorporates mechanisms such as
 - ⇒ blurring (\rightarrow averaging over a neighbourhood of pixels)
 - ⇒ Fourier transform (\rightarrow observations in frequency domain)
 - ⇒ mask (\rightarrow partial observations)
- Observations: $y = \mathcal{G}(u) + \eta \in \mathbb{R}^{d_y}$.

Inverse Problem Examples

Example 2: Regression

- **Goal:** reconstruct a function f given noisy point values (measurements) $\{f(x_i)\}$.



Inverse Problem Examples

Example 2: Regression

- **Goal:** reconstruct a function f given noisy point values (measurements) $\{f(x_i)\}$.
- **Unknown:** $u \in \mathbb{R}^{d_u}$, coefficients in a (polynomial) basis expansion

$$f(x; u) = \sum_{j=1}^{d_u} \textcolor{red}{u_j} \phi_j(x),$$

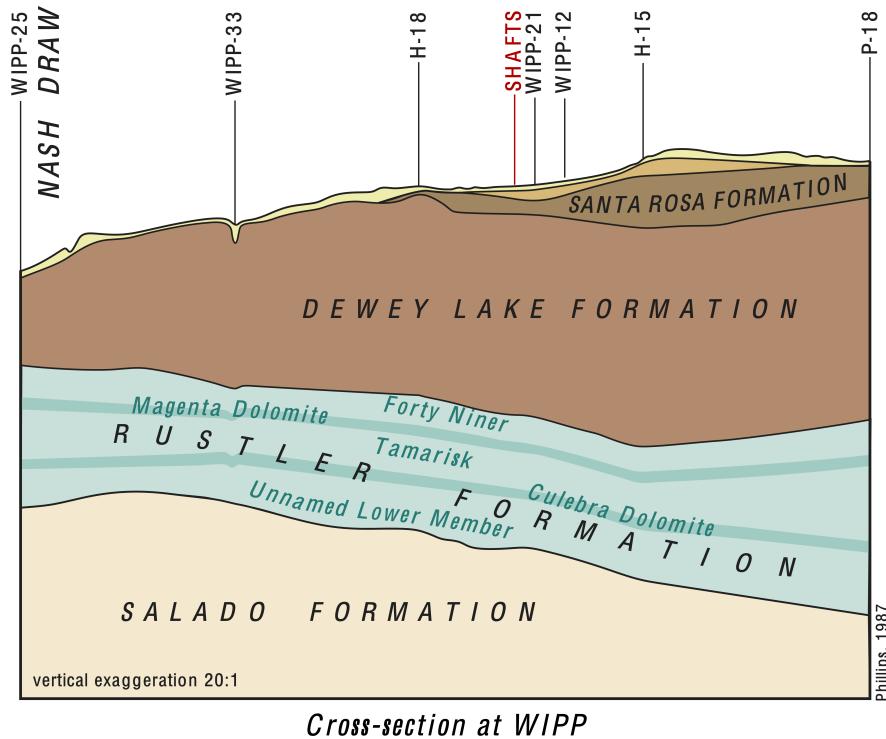
where $\{\phi_j\}_{j=1}^{d_u}$ are linearly independent, eg.
 $\phi_j(x) = x^{j-1}$.

- **Map:** \mathcal{G} implicitly defined by $u \mapsto \{f(x_i; u)\}_{i=1}^{d_y}$ and $\mathcal{G} = G$ is linear (Vandermonde matrix) with $a_{ij} = \phi_j(x_i)$.
- Observations: $y = \{f(x_i; u) + \eta_i\}_{i=1}^{d_y} \in \mathbb{R}^{d_y}$.

Inverse Problem Examples

Example 3: Porous Medium Flow

- **Goal:** reconstruct the hydraulic conductivity k of the subsurface given noisy measurements of the water pressure $\{p(x_i)\}$.



Inverse Problem Examples

Example 3: Porous Medium Flow

- **Goal:** reconstruct the hydraulic conductivity k of the subsurface given noisy measurements of the water pressure $\{p(x_i)\}$.
- **Unknown:** $u \in \mathbb{R}^{d_u}$, coeff's in a basis expansion

$$f(x; u) = \phi_0(x) + \sum_{j=1}^{d_u} \color{red}{u_j} \phi_j(x),$$

where $\{\phi_j\}_{j=1}^{d_u}$ are linearly independent and ϕ_0 ensures positivity of k .

- **Map:** \mathcal{G} implicitly defined by $u \mapsto \{p(x_i; u)\}_{i=1}^{d_y}$, where p is the solution of (Darcy)

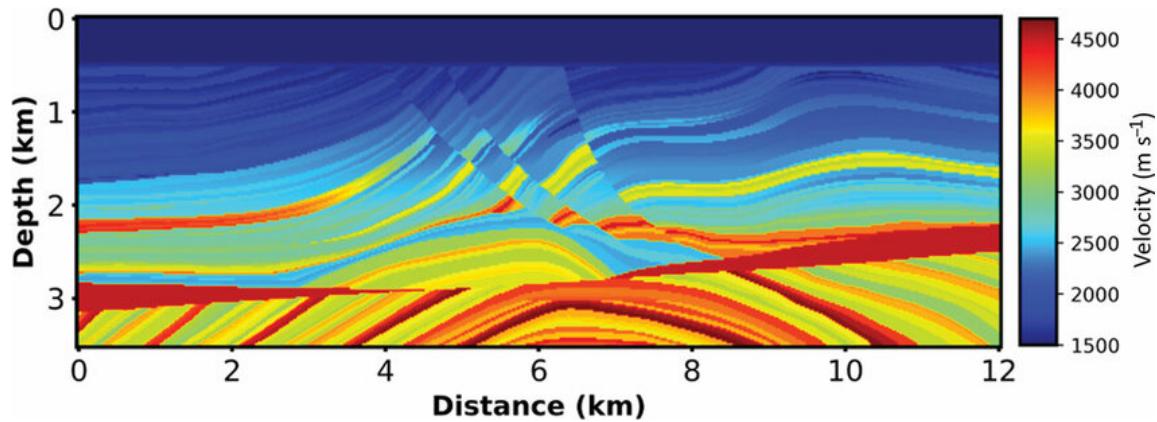
$$-\nabla \cdot (k(x; u) \nabla p(x; u)) = h(x)$$

- Observations: $y = \{p(x_i; u) + \eta_i\}_{i=1}^{d_y} \in \mathbb{R}^{d_y}$.

Inverse Problem Examples

Example 4: Full Wave Inversion (FWI)

- **Goal:** reconstruct the subsurface properties ρ, κ given noisy measurements of the acoustic/elastic pressure at given times $\{p(x_i, t_j)\}$.



Inverse Problem Examples

Example 4: Full Wave Inversion (FWI)

- **Goal:** reconstruct the subsurface properties ρ, κ given noisy measurements of the acoustic/elastic pressure at given times $\{p(x_i, t_j)\}$.
- **Unknown:** $u \in \mathbb{R}^{d_u}$, coeff's in a basis expansion

$$f(x; u) = \phi_0(x) + \sum_{j=1}^{d_u} \mathbf{u}_j \phi_j(x),$$

- **Map:** \mathcal{G} implicitly defined by $u \mapsto \{p(x_i, t_n; u)\}_{i=1}^{d_y}$, where p is the solution of

$$\rho(x; u) \frac{\partial^2 p}{\partial t^2} - \nabla \cdot (\kappa(x; u) \nabla p(x; u)) = h(x)$$

- Observations: $y_n = \{p(x_i, t_n; u) + \eta_i\}_{i=1}^{d_y} \in \mathbb{R}^{d_y}$.

DIP vs BIP

Deterministic/Bayesian IP Standoff

Our test problem is the estimation of an unknown (or badly known) coefficient, m , in an elliptic partial differential equation. This is a **protoypical problem** setting, and can be extended to a large number of other pde's, contexts and problems.

- Let $\Omega \subset \mathbb{R}^n$, $n \in \{1, 2, 3\}$ be an open, bounded domain, and consider the following **deterministic inverse problem** (DIP):

$$\min_m J(m) := \frac{1}{2} \int_{\Omega} (u - u_d)^2 dx + \frac{\gamma}{2} \int_{\Omega} |\nabla m|^2 dx, \quad (4)$$

where u is the solution of a Poisson, boundary-value problem,

$$\begin{aligned} -\nabla \cdot (\exp(m) \nabla u) &= f \text{ in } \Omega, \\ u &= 0 \text{ on } \partial\Omega. \end{aligned} \quad (5)$$

- ⇒ $m \in \mathcal{M}_{\text{ad}} := \{m \in H^1(\Omega) \cap L^\infty(\Omega)\}$ the unknown coefficient field,
- ⇒ u_d denotes (possibly noisy) measured data,
- ⇒ $f \in H^{-1}(\Omega)$ a given force, and
- ⇒ $\gamma \geq 0$ the regularization parameter.

DIP

Numerical Solution

- Simulations will show the computational cost as function of problem size N of
 - ⇒ (1) direct simulations, and
 - ⇒ (2) deterministic inverse probem.
- Direct solution could be seen as a **random differential equation** (a differential equation with random coefficients) approach, where the parameter is perturbed randomly, but
 - ⇒ there is no propagation of uncertainty;
 - ⇒ we have no guarantee that the ensemble is representative;
 - ⇒ we require a very large number of simulations to obtain reliable (but, see above 2 points) statistics.
- Some advantages of the RDE approach:

- ⇒ extremely simple and non-intrusive (requires no modification of the direct code);
- ⇒ can aid falsification (in a Bayesian approach) where we would like to ascertain whether realizations fall within a given distribution or not.

DIP

Computational Scaling

- Computational cost as a function of the problem size².

| N | #dof | Time Fwd (s) | Time DIP (s) |
|------|-----------------|--------------|----------------|
| 32 | 1024 | 0.0277 | 3.2 |
| 64 | 4096 | 0.0446 | 10.9 |
| 128 | 16384 | 0.1336 | 61.9 |
| 256 | 65536 | 0.6138 | 368.1 |
| 512 | 262144 | 3.6356 | ≈ 3000 |
| 1024 | 1×10^6 | 27.112 | |

Table 1: CPU time for deterministic Poisson equation (left), deterministic inverse problem (right) - factor of 500X between direct and inverse.

²Villa, U. and Petra, N. and Ghattas, O. *hIPPYlib: an Extensible Software Framework for Large-scale Deterministic and Bayesian Inversion.* 2016. <http://hippylib.github.io>

- The results (Table 1 and Figure 1) show a cubic (power of 3) exponential growth.

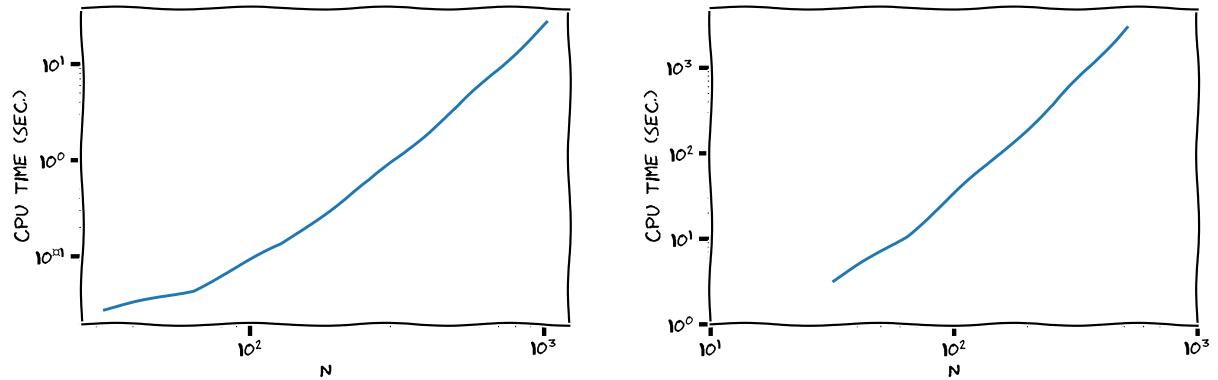


Figure 1: CPU time as function of problem size: direct problem (left), inverse problem (right).

- The computational cost for a **direct solution** scales as $\mathcal{O}(N^2)$.
- The computational cost for an **inverse problem solution** is approximately $500\times$ that for the direct problem, i.e. $\mathcal{O}(N^3)$.

BIP

Problem Formulation

- Determining the permeability of an **unknown medium** (subsurface rock, battery cell, etc.) is enormously important in a range of different applications. Among these applications are:
 - ⇒ the prediction of transport of radioactive waste from underground waste repositories,
 - ⇒ the forecast of geothermal production,
 - ⇒ the optimization of oil recovery from underground fields,
 - ⇒ the electrochemical behaviour of battery cells.
- **Darcy's law** is an excellent model of the pressure field as a function of the permeability,

$$\begin{aligned}-\nabla \cdot (k \nabla p) &= 0, \quad x \in D, \\ p &= h, \quad x \in \partial D.\end{aligned}$$

- The **inverse problem** is: find the permeability k from observations/measurements of the pressure at points in the interior of D .

BIP

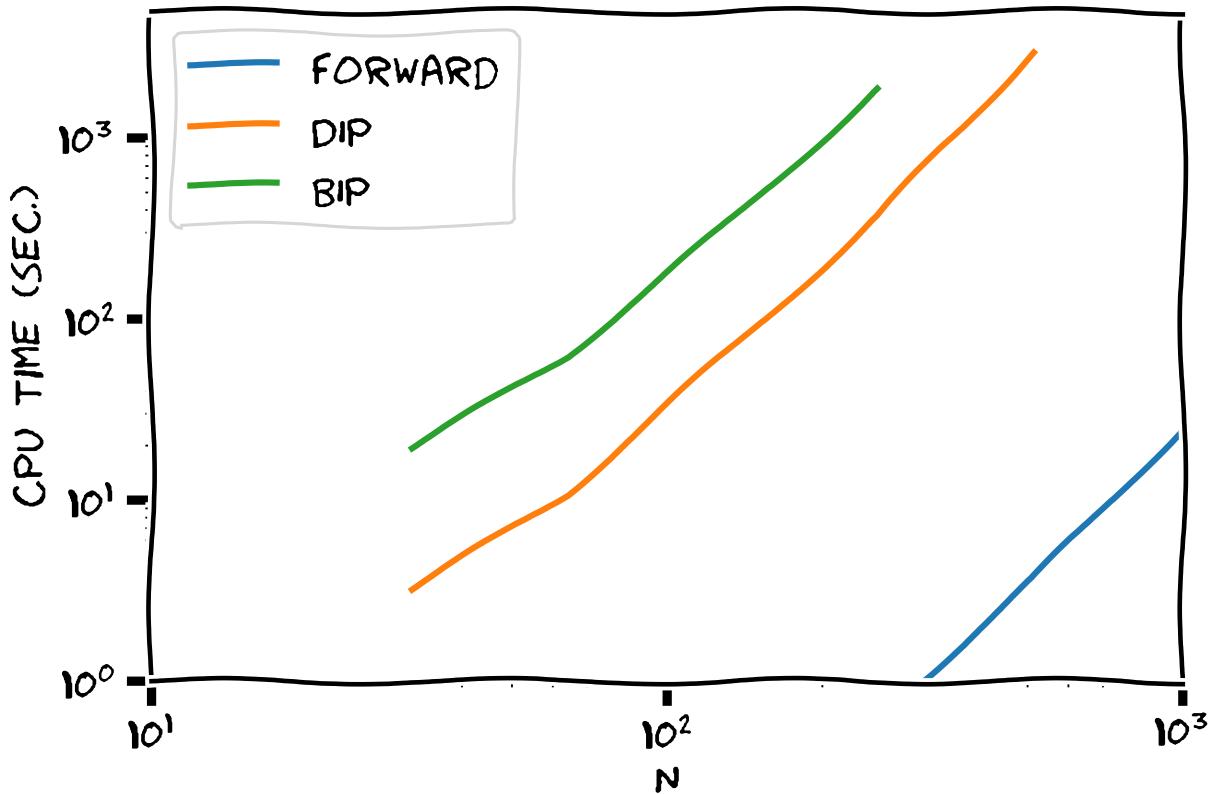
Numerical Solution

| N | #dof | Time Fwd | Time DIP | Time BIP (s) |
|------|-----------------|----------|----------------|--------------|
| 32 | 1024 | 0.03 | 3.2 | 19.3 |
| 64 | 4096 | 0.04 | 10.9 | 63.1 |
| 128 | 16384 | 0.13 | 61.9 | 317.0 |
| 256 | 65536 | 0.61 | 368.1 | 1865.8 |
| 512 | 262144 | 3.64 | ≈ 3000 | |
| 1024 | 1×10^6 | 27.1 | | |

Table 2: CPU times for MAP solution of Bayesian inverse problem

- Method used here [Ghattas, et al] is state-of-the-art, scalable, adjoint-based algorithm, based on a stochastic Newton McMC method with MAP-based Hessian.
- Comparison of CPU times for forward and inverse

problems.



- **Remark:** Cost of BIP is 5-50X DIP, which is $2\text{--}500\text{--}25\,000\times$ the direct problem (depending on the required posterior statistics).

CONCLUSIONS

DA: 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.

IP: conclusions

- Ill-posedness must always be addressed.
- Statistical inversions, based on Bayesian analysis are one approach for bypassing the ill-posedness, but introduce new theoretical and computational challenges.
- Kalman filters have a long tradition and a strong theoretical basis and they can solve the Bayesian inverse problem.
- Ensemble Kalman filters can be used as a basis for the formulation and solution of more general inverse problems.

Post-Bayes Research

- 3 major limiting assumptions of Bayesian approach:
[J. Knoblauch, UCL]
 - ⇒ (A1) Model is well-specified.
 - ⇒ (A2) Prior is well-specified.
 - ⇒ (A3) Computational feasibility.
- Generalized Variational Inference (see also V.I. in Lecture #3) can overcome all three...

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.
9. Ghattas Group. <http://users.ices.utexas.edu/~omar/>

APPENDIX

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:
 - ⇒ the objective of TR is to reduce, or remove, ill-posedness in optimization problems by modifying the objective function.
 - ⇒ the three sources of ill-posedness: non-existence, non-uniqueness and sensitivity to perturbations.
 - ⇒ 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

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

where

- ⇒ ρ is the *data discrepancy functional* that quantifies the difference between the model output and the measured data;
- ⇒ J is the *regularization functional* that represents some desired quality of the sought for model parameters, usually smoothness;
- ⇒ 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*,

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

- *1-norm*,

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

- *Kullback-Leibler distance*,

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

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*:

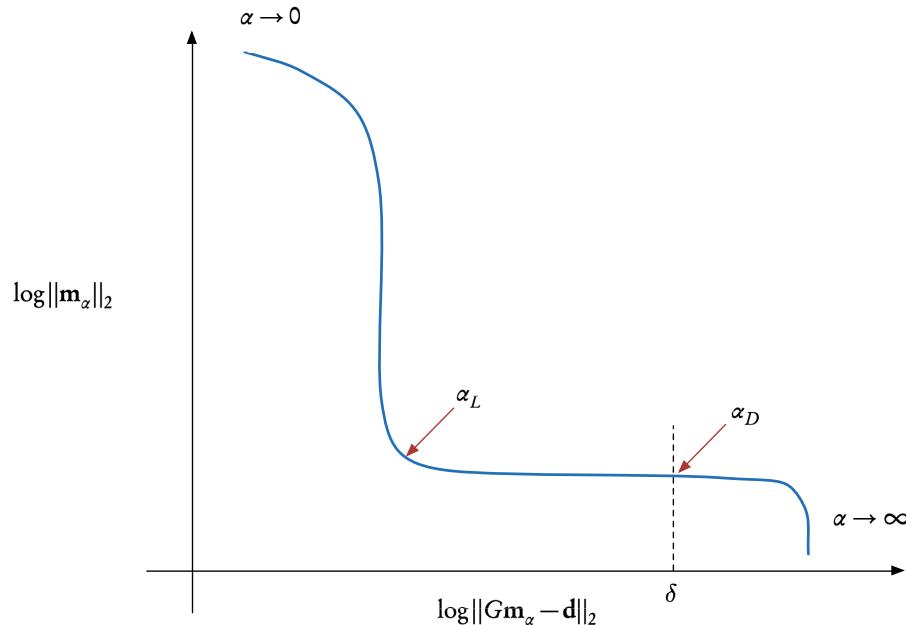
$$J_2(\mathbf{m}) = \frac{1}{2} \|\nabla^2 \mathbf{m}\|_2^2,$$

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.
⇒ 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 .
 - ⇒ 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,
 - ⇒ 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 .

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

- ⇒ 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.
- ⇒ On the L-curve, this corresponds to the intersection with the vertical line at this bound, as shown in Figure.
- ⇒ 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.³ This can be thought

³This is strictly valid under the hypothesis of i.i.d. Gaussian noise.

of as the noise level of the data.

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

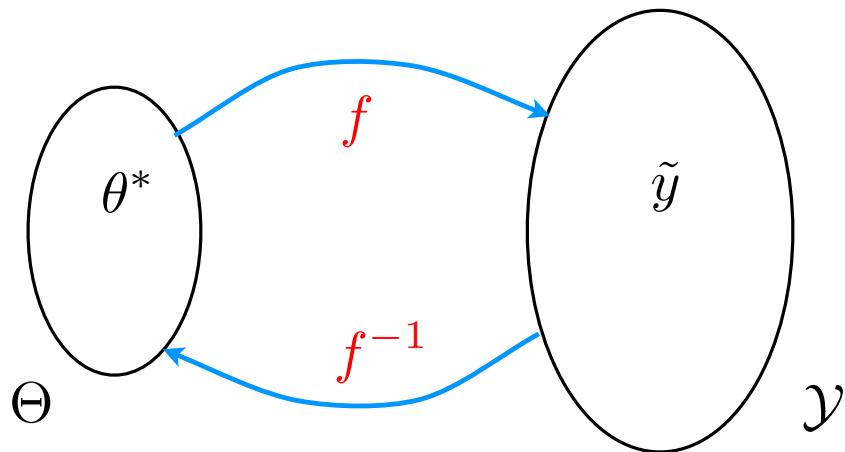
III-posedness of inverse problems

All inverse problems are notoriously ill-posed!!!

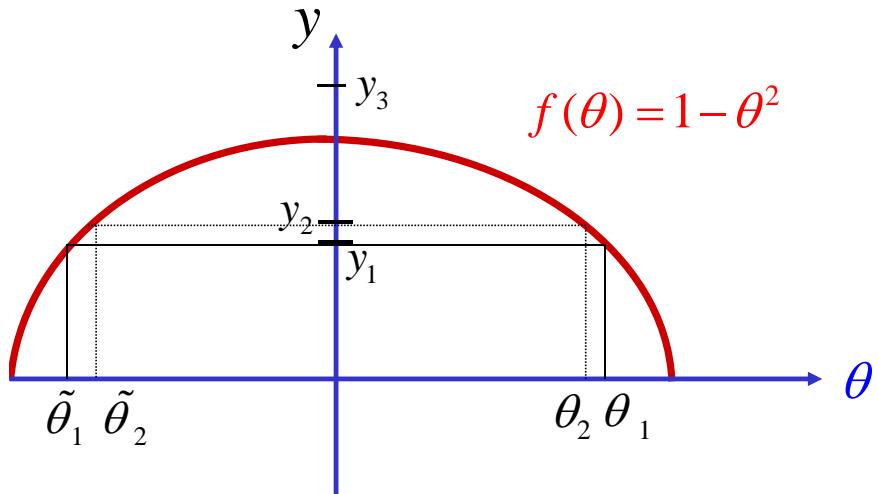
Simplest case: one observation \tilde{y} for $f(\theta)$ and we need to find the pre-image

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

for a given \tilde{y} .



Simplest case



- ✖ Non-existence: there is no θ_3 such that $f(\theta_3) = y_3$
- ✖ Non-uniqueness: $y_j = f(\theta_j) = f(\tilde{\theta}_j)$ for $j = 1, 2$.
- ✖ Lack of continuity of inverse map: $|y_1 - y_2|$ small
 $\Rightarrow |f^{-1}(y_1) - f^{-1}(y_2)| = |\theta_1 - \tilde{\theta}_2|$ small.

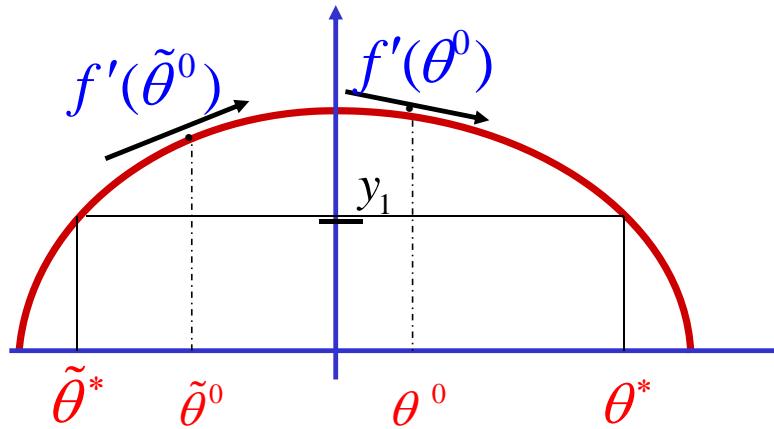
Why is this so important???

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:

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

Newton iterations



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

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

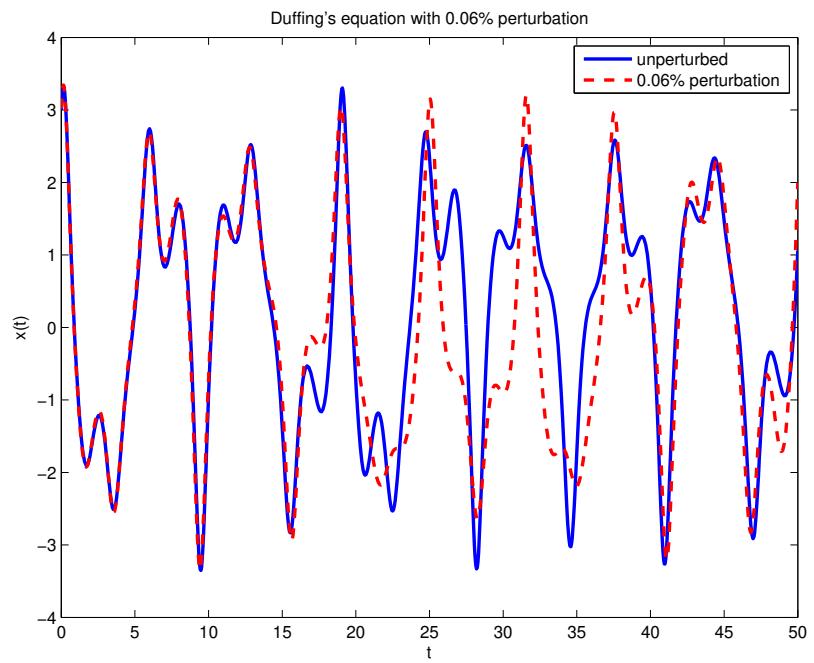
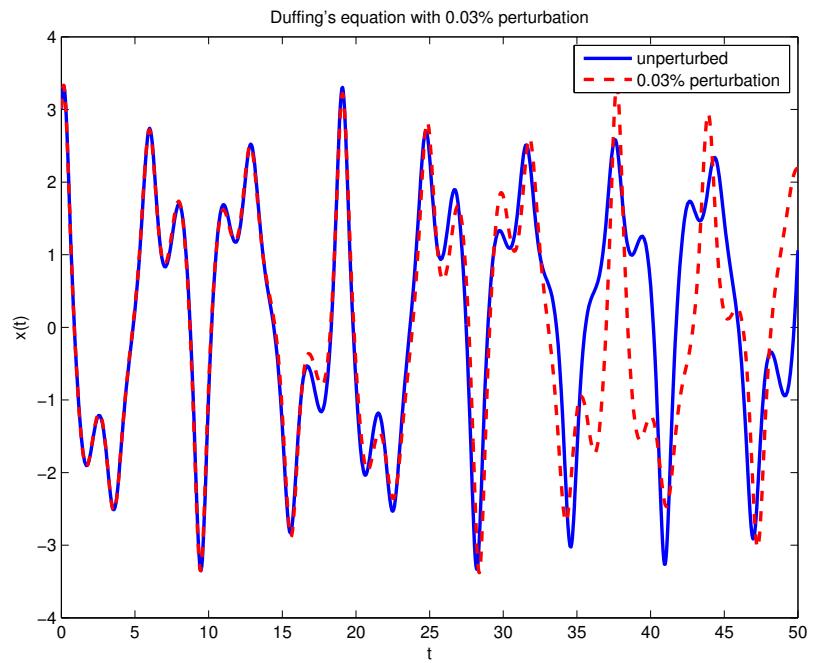
Nonlinearity: Duffing

The highly nonlinear Duffing's equation,

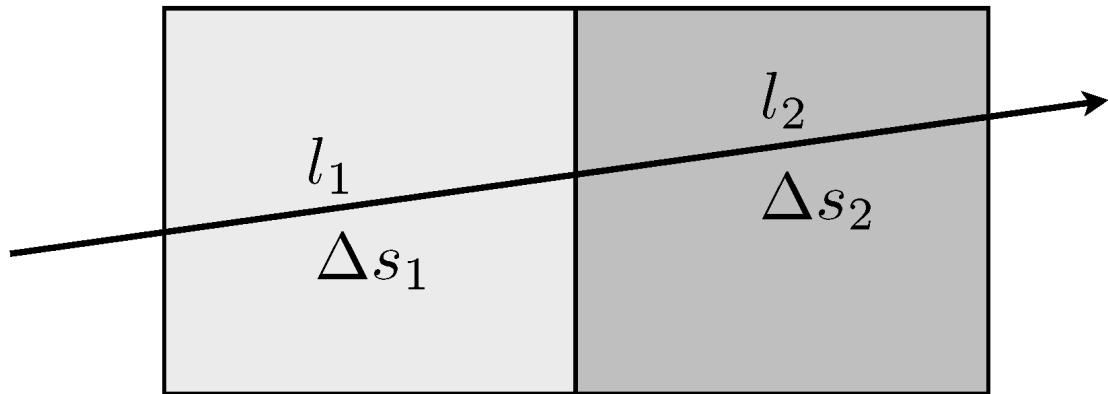
$$\ddot{x} + 0.05\dot{x} + x^3 = 7.5 \cos t$$

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 $t = 20$.



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.

DATA ASSIMILATION

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

Definition 9. 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.

DA: 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):
 - ⇒ control theory;
 - ⇒ variational calculus;
 - ⇒ statistical estimation theory;
 - ⇒ probability theory,
 - ⇒ stochastic differential equations.
- Newer approaches: nudging methods, reduced methods, ensemble methods and hybrid methods that combine variational and statistical approaches, **Machine/Deep Learning based approaches**.

DA: applications

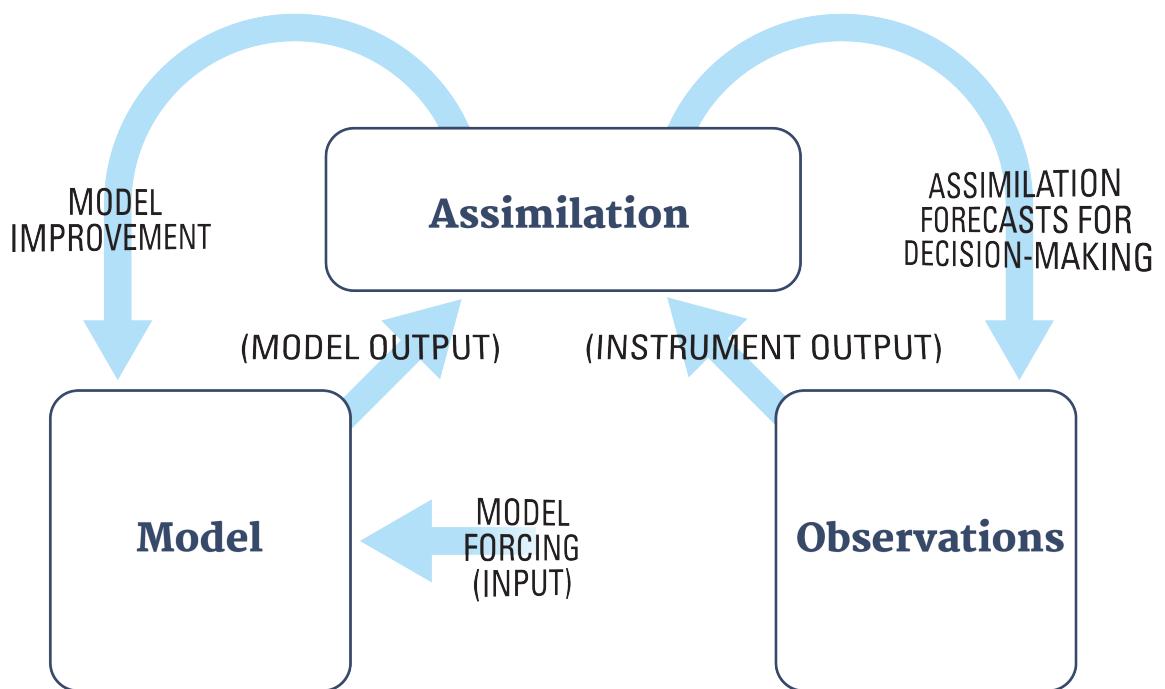
1. Navigation: important application of the Kalman filter.
2. Remote sensing: satellite data.
3. Geophysics: seismic exploration, geo-prospection, earthquake prediction.
4. Air and noise pollution, source estimation.
5. Weather forecasting.
6. Climatology. Global warming.
7. Epidemiology.
8. Forest fire evolution.
9. Finance.

DA & IP: 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
 - ⇒ the inference cycle
 - ⇒ machine learning...

DA: 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

$$\mathbf{x}^f(t_{k+1}) = M [\mathbf{x}^f(t_k),] \quad (6)$$

⇒ \mathbf{x} is the model's state vector of dimension n ,
⇒ 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 (6) by random or systematic errors.
- Observations, or measurements, at time t_k are defined by

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

- ⇒ H is an **observation operator**
- ⇒ ε is a **white noise process** zero mean and covariance matrix \mathbf{R} (instrument errors and representation errors due to the discretization)
- ⇒ observation vector $\mathbf{y}_k^o = \mathbf{y}^o(t_k)$ 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:
 - ⇒ “a” for **analysis**,
 - ⇒ “b” for **background** (or ‘initial/first guess’),
 - ⇒ “f” for **forecast**,
 - ⇒ “o” for **observation** and
 - ⇒ “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 state and observation equations become

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

and

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

where $(\cdot) = d/dt$, \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 $\mathbf{u}(\mathbf{x}, t)$, 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**,

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

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.

DA 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.