Intro Inverse Modelling

Thomas Kaminski (http://FastOpt.com)

Thanks to:

Simon Blessing (FastOpt), Ralf Giering (FastOpt), Nadine Gobron (JRC),

Wolfgang Knorr (QUEST), Thomas Lavergne (Met.No), Bernard Pinty (JRC),

Peter Rayner (LSCE), Marko Scholze (QUEST), Michael Voßbeck (FastOpt)

4th Earth Observation Summer School, Frascati, August 2008

Overview

BP1: Basic concepts: from remote sensing measurements to surface albedo estimates (1)

BP2: Basic concepts: from remote sensing measurements to surface albedo estimates (2)

TK1: Introduction to inverse modelling

TK2: Intro tangent and adjoint code construction

TK3: Demo: BRF inverse package and Carbon Cycle Data Assimilation System

BP3: Monitoring land surfaces: applications of inverse packages
of both surface BRF and albedo models

FastOpt

References and Definitions

- A. Tarantola, Inverse Problem Theory and Methods for Model Parameter Estimation, SIAM, (1987, 2004)

- I. G. Enting, Inverse Problems in Atmospheric Constituent Transport, C.U.P. (2002)

Definitions

Name	Symbol	Description
Parameters	$\vec{p}$	Quantities not changed by model, i.e. process parameters, boundary conditions
State variables	$\vec{v}(t)$	Quantities altered by model from time step to time step
Control variables	$\vec{x}$	Quantities exposed to optimisation, a combination of subsets of $\vec{p}$ and $\vec{v}(t=0)$
Observables	$\vec{o}$	Measurable quantities
Observation operator	$\mathbf{H}$	Transforms $\vec{v}$ to $\vec{o}$
Model	$\mathbf{M}$	Predicts $\vec{o}$ given $\vec{p}$ and $\vec{v}(t=0)$, includes $\mathbf{H}$
Data	$\vec{d}$	Measured values of $\vec{o}$

Statement of Problem

Given a model $\mathbf{M}$, a set of measurements $\vec{d}$ of some observables $\vec{o}$, and prior information on some control variables $\vec{x}$, produce an updated description of $\vec{x}$.

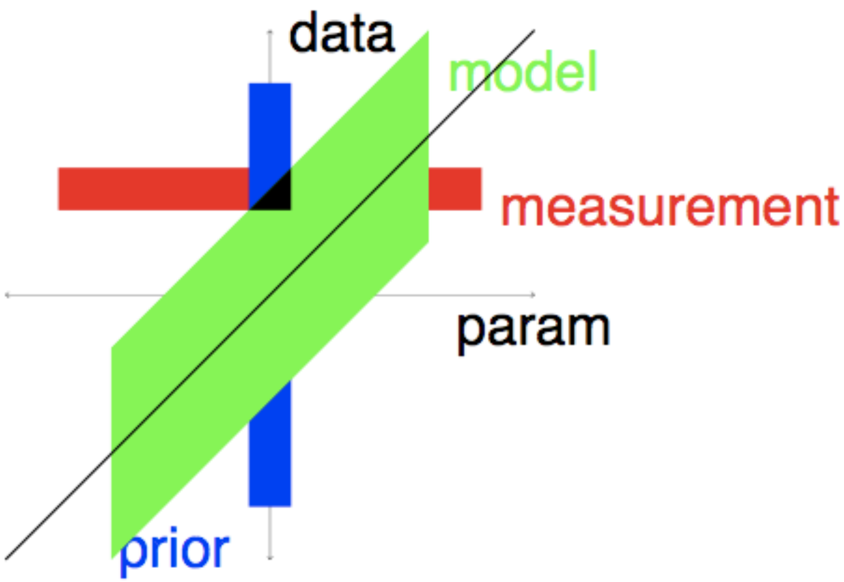
$\vec{x}$ may include parts of $\vec{p}$ and $\vec{v}(t=0)$

$$\vec{o} = \mathbf{M}(\vec{x})$$

Information and PDFs

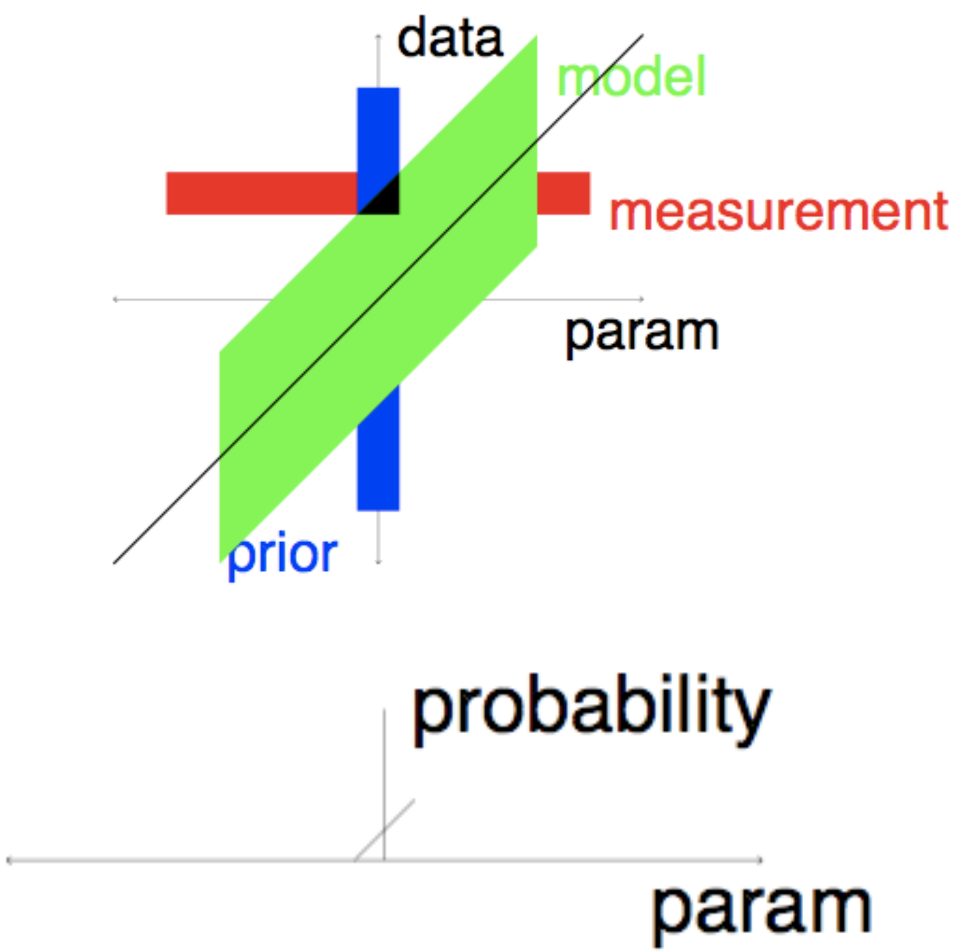
- We seek true value but all input information is approximate
- Treat model, measurements and priors as PDFs describing distribution of true value
- Data and prior error easy
- Model error is difference between actual and perfect simulation *for a given* $\vec{x}$
- Arises from errors in equations and errors in those parameters not included in $\vec{x}$

Combining PDFs



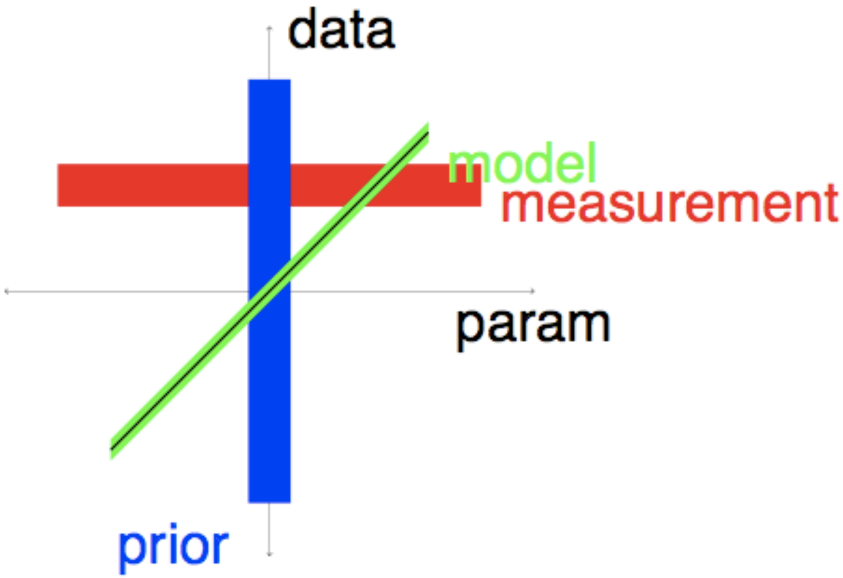
- Operate in joint parameter and data space
- Estimates are combination of prior, measurement and model (black triangle)
- Estimate is multiplication of PDFs
- Only involves forward models
- Parameter estimate projection of PDF onto X-axis

Summary statistics from PDF



- Combination of PDFs (upper) and posterior PDF for parameters (lower)
- Can calculate all summary statistics from posterior PDF
- Maximum Likelihood Estimate (MLE) maximum of PDF

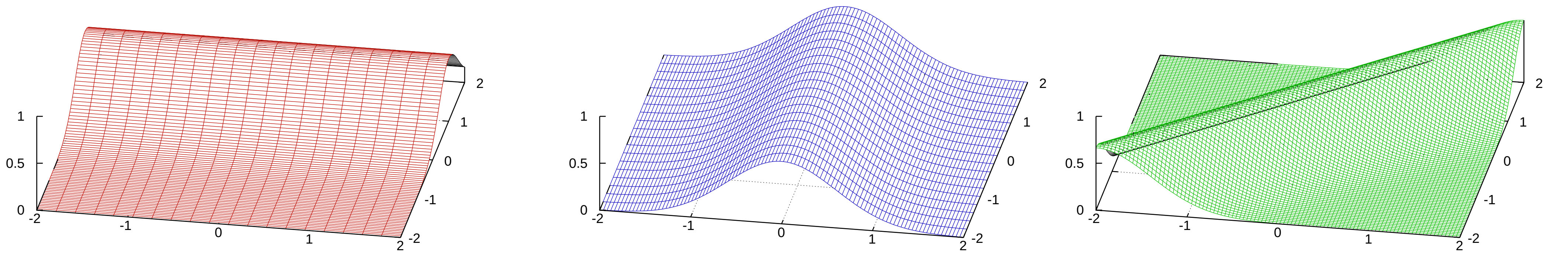
Tightening Model Constraint



- Note no viable solution
- If we cannot treat model as soft constraint we must inflate data uncertainty
- Model error hard to characterise

Gaussian Case

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$



Data:

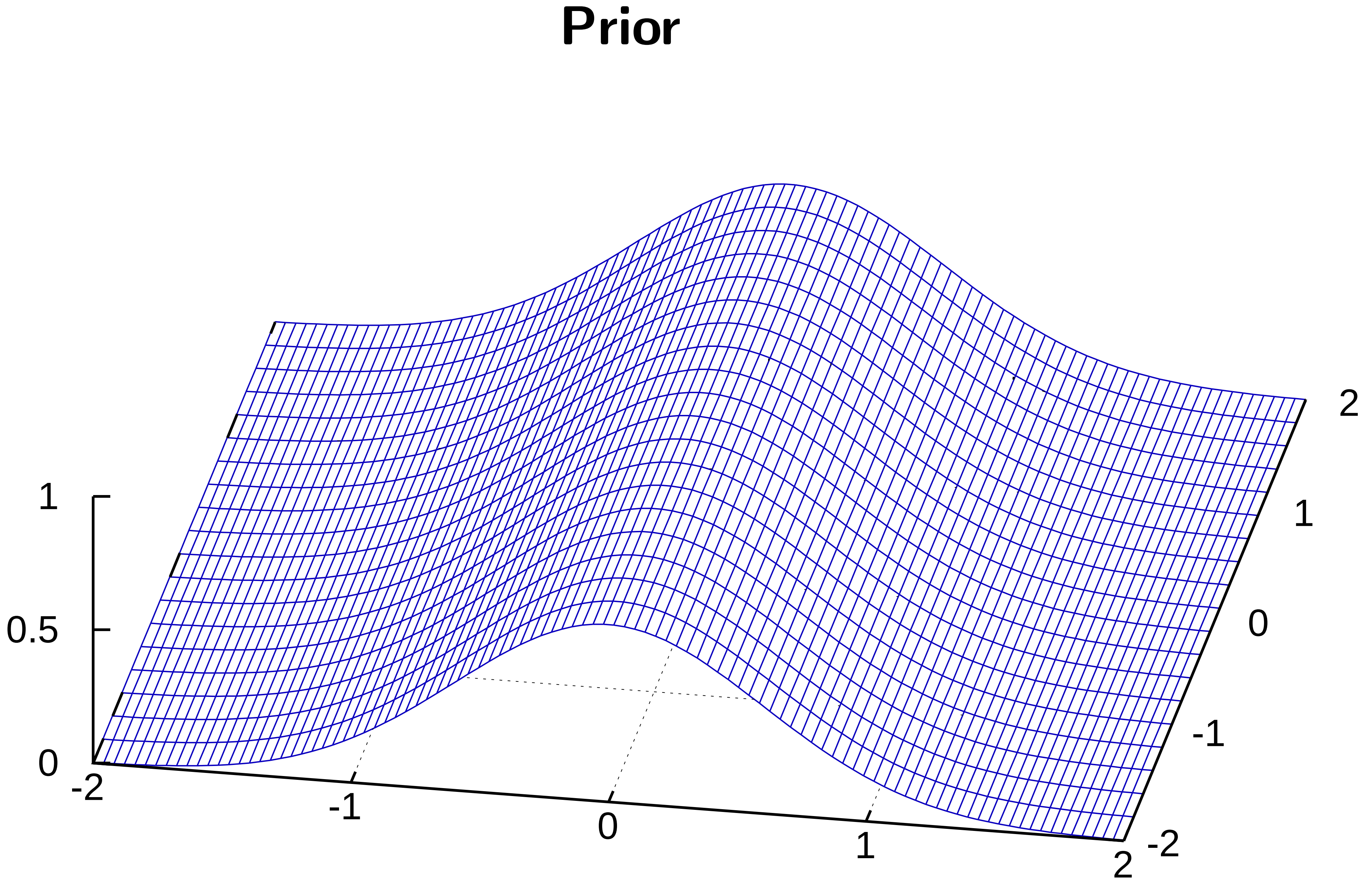
$$P_d \propto e^{\frac{-(d-1)^2}{2\cdot 0.6^2}}$$

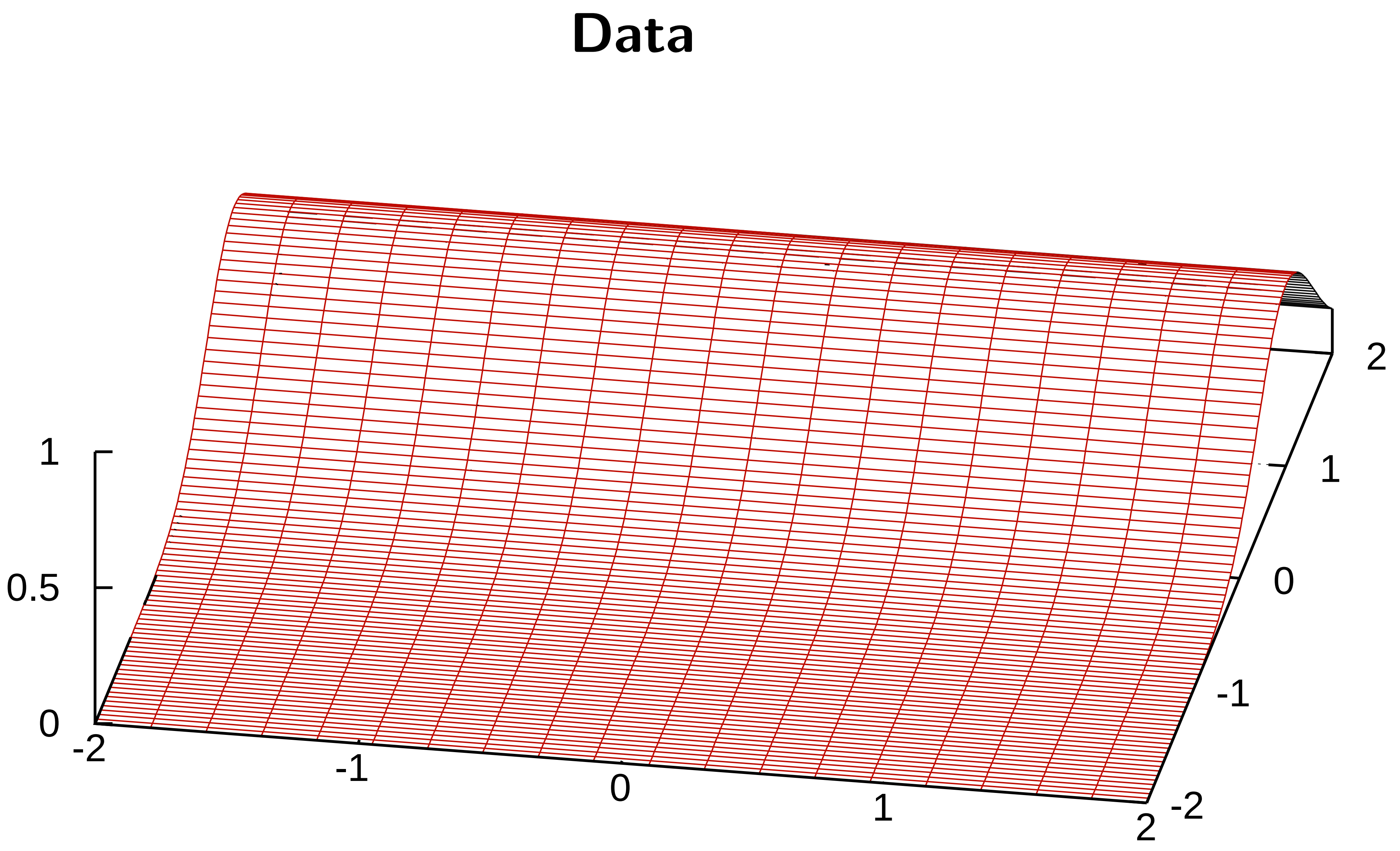
Parameters:

$$P_x \propto e^{\frac{-x^2}{2\cdot 0.6^2}}$$

Model:

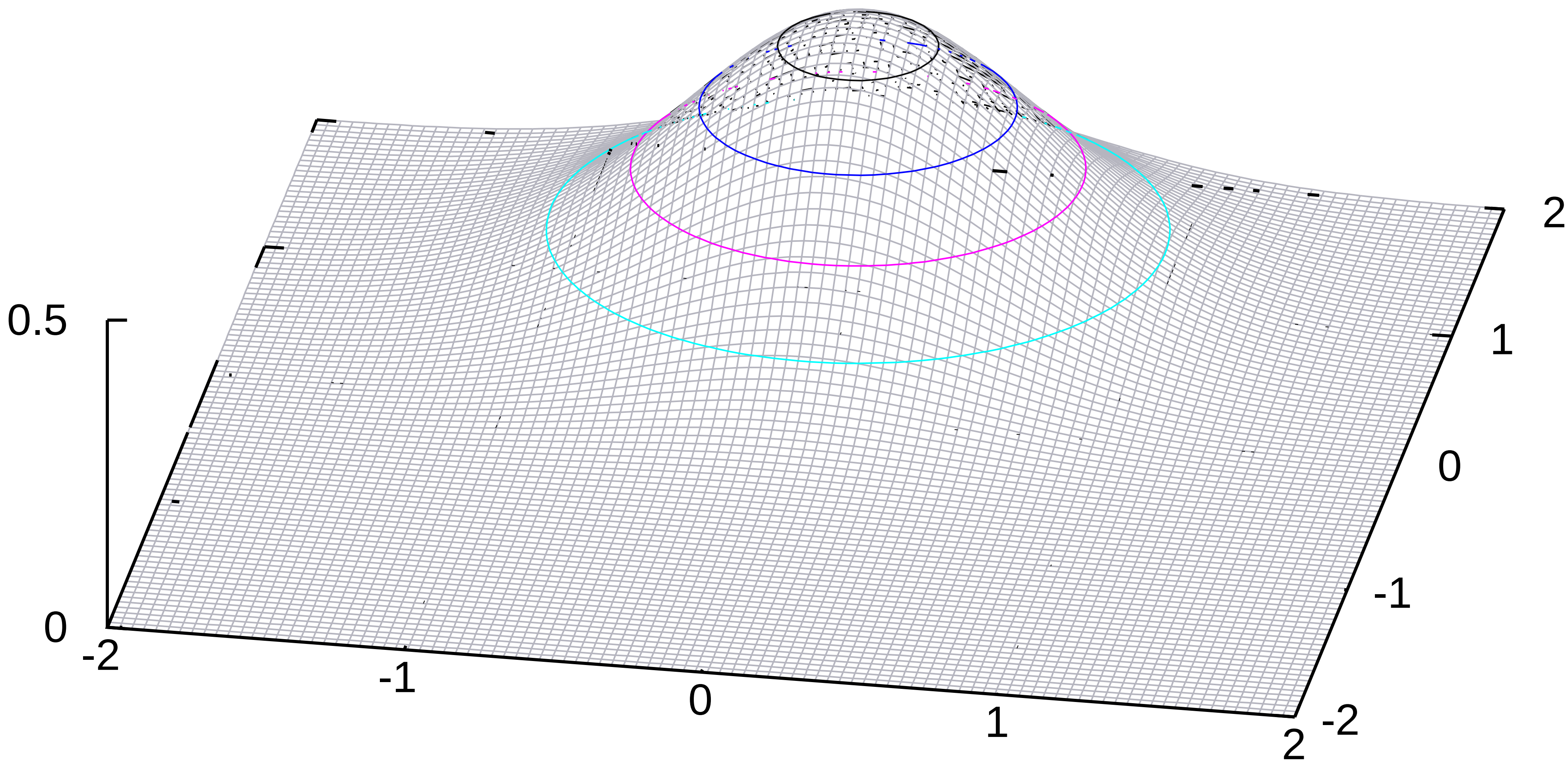
$$P_M \propto e^{\frac{-(d-x)^2}{2\cdot 0.6^2}}$$


Prior
1
0.5
0
-2
-1
0
1
2
-2
-1
0
1
2

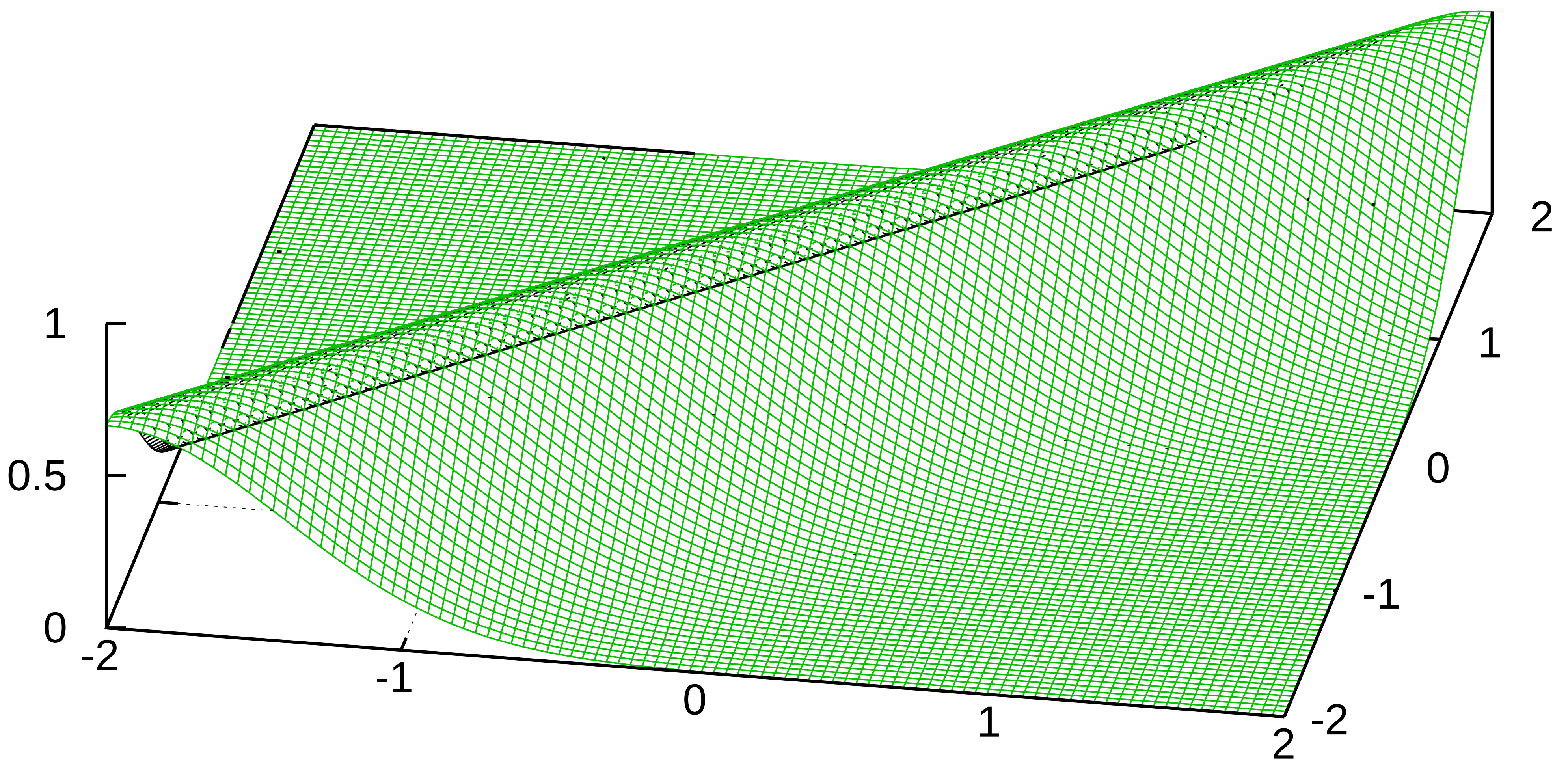


Data
1
0.5
0
-2
-1
0
1
2
-2
-1
0
1
2

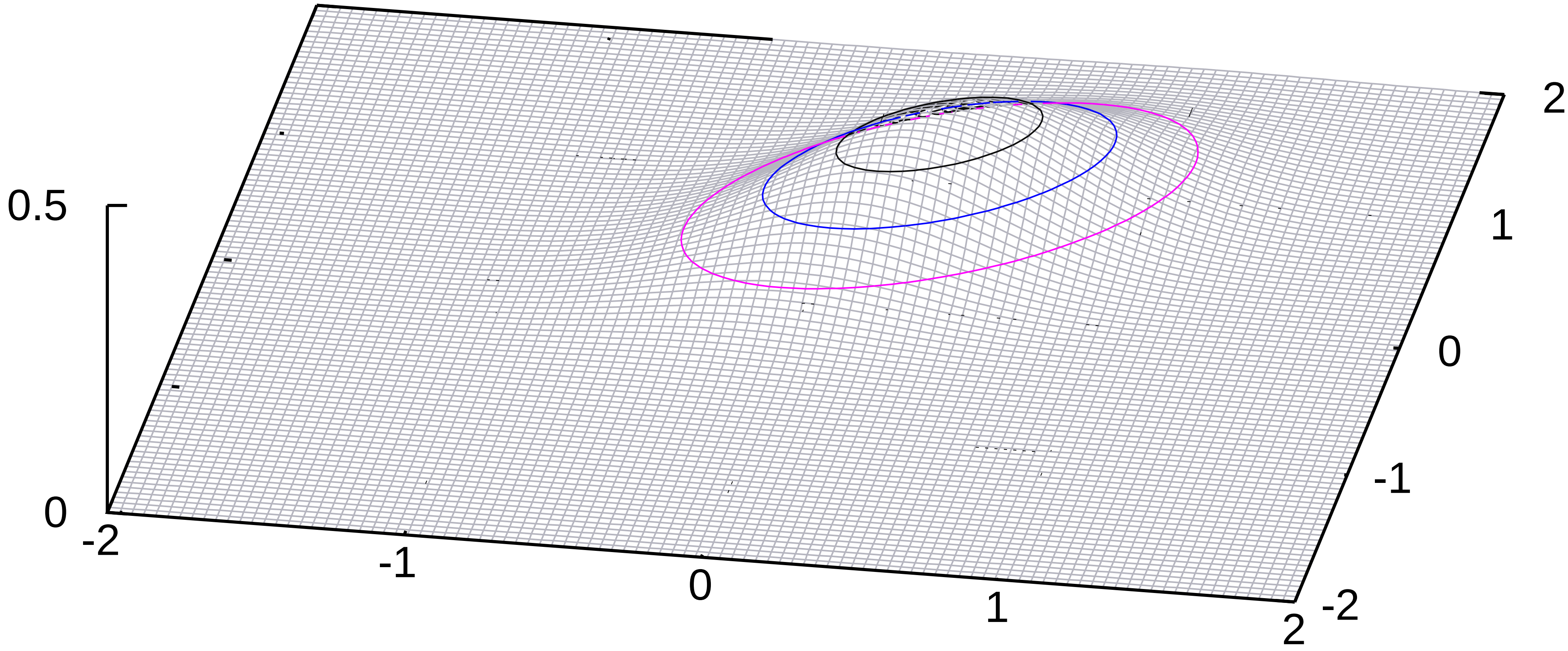
Prior plus Data



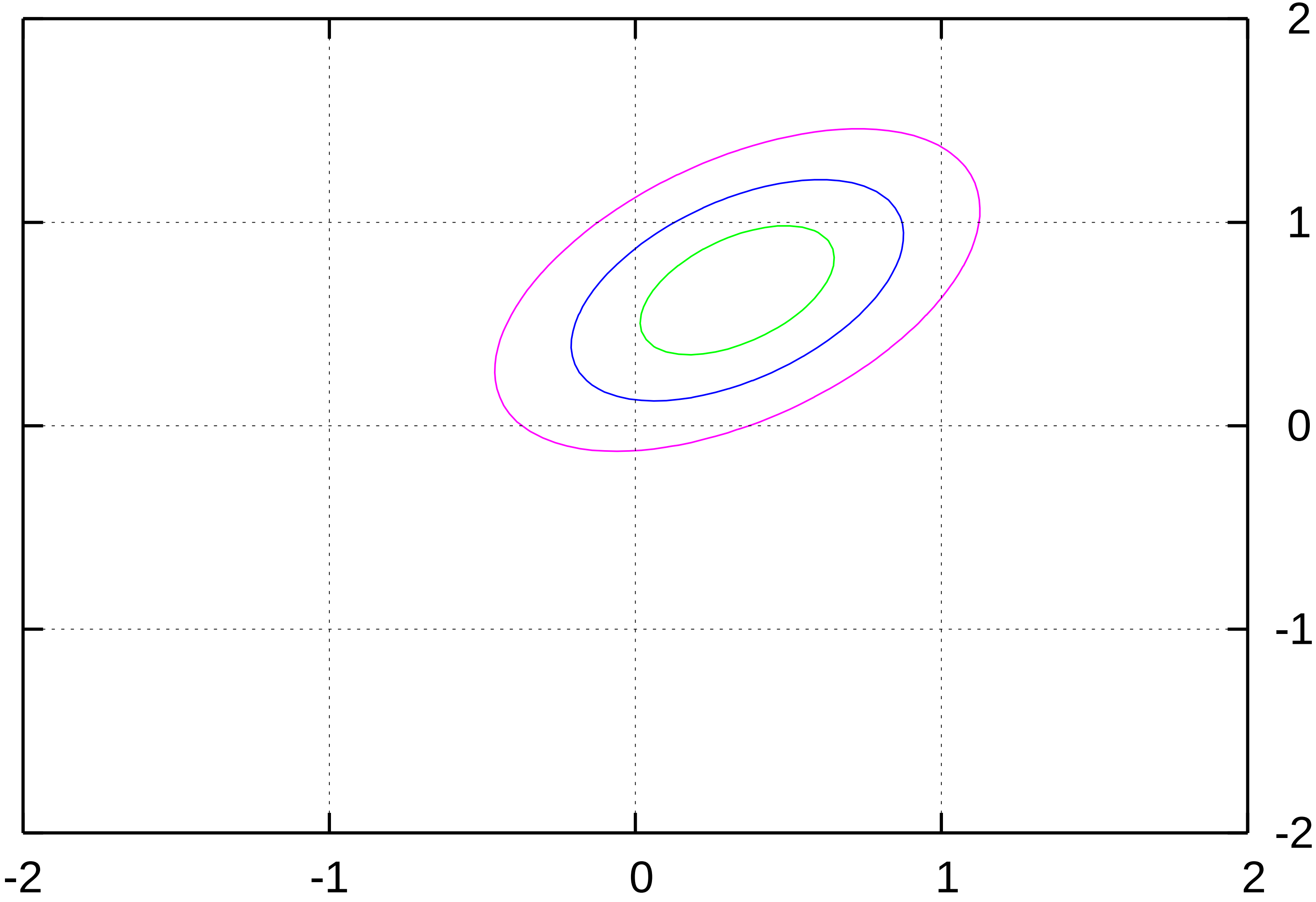
Model



Prior plus Data plus Model



Prior plus Data plus Model



PDFs and cost functions

- MLE most common quantity
- Maximise PDF numerically
- Most common $P(x) \propto e^{-\frac{1}{2}J}$ so maximising $P \leftrightarrow$ minimising J
- J usually called the cost (or misfit or objective) function
- Exponentials convenient because multiplying exponentials $\leftrightarrow$ adding exponents

Perfect and imperfect models

Imperfect Model

- Include observables ($\vec{o}$) in control variables

$$J = \frac{1}{2}\left(\frac{(\vec{x}-\vec{x}_0)^2}{\sigma_x^2} + \frac{(\vec{o}-\vec{d})^2}{\sigma_{obs}^2} + \frac{(\mathbf{M}(\vec{x})-\vec{o})^2}{\sigma_M^2}\right)$$

- $\vec{x}_0$ prior estimate for $\vec{x}$, σ standard deviations.
- Allows explicit model error
- Different from weak constraint 4D-Var in which model state added to control variables

Perfect Model

- Observables not included in control vector

$$J = \frac{1}{2}\left(\frac{(\vec{x}-\vec{x}_0)^2}{\sigma_x^2} + \frac{(\mathbf{M}(\vec{x})-\vec{d})^2}{\sigma_d^2}\right)$$

- If model imperfect, carried model error in σ_d:
  $\sigma_d^2 = \sigma_{obs}^2 + \sigma_M^2$
- This is the form usually used in data assimilation
- For linear $\mathbf{M}$ there is a closed solution
- Otherwise minimised by iterative algorithm, usually using gradients
- Can calculate posterior σ_x by rearranging PDF

Covariances

$$J = \frac{1}{2}\left(\frac{(\vec{x}-\vec{x}_0)^2}{\sigma_x^2} + \frac{(\mathbf{M}(\vec{x})-\vec{d})^2}{\sigma_d^2}\right)$$

Takes the more general form

$$J = \frac{1}{2}\Big((\vec{x}-\vec{x}_0)^T\mathbf{C}(\vec{x}_0)^{-1}(\vec{x}-\vec{x}_0) + (\mathbf{M}(\vec{x})-\vec{d})^T\mathbf{C}(\vec{d})^{-1}(\mathbf{M}(\vec{x})-\vec{d})\Big)$$

- Off-diagonal terms represent correlation in uncertainty
- Persistent instrumental error
- Model error usually correlated
- Errors in initial conditions from previous forecast

Linear Gaussian case

$$J = \frac{1}{2}\left((\vec{x}-\vec{x}_0)^T\mathbf{C}(\vec{x}_0)^{-1}(\vec{x}-\vec{x}_0) + (\mathbf{M}\vec{x}-\vec{d})^T\mathbf{C}(\vec{d})^{-1}(\mathbf{M}\vec{x}-\vec{d})\right)$$

$$\vec{x}_{opt} = \vec{x}_0 + \mathbf{C}(\vec{x}_0)\mathbf{M}^T\left(\mathbf{M}\mathbf{C}(\vec{x}_0)\mathbf{M}^T + \mathbf{C}(\vec{d})\right)^{-1}(\vec{d}-\mathbf{M}\vec{x}_0)$$

$$\begin{aligned}\mathbf{C}(\vec{x})^{-1} &= \mathbf{C}(\vec{x}_0)^{-1} + \mathbf{M}^T\mathbf{C}(\vec{d})^{-1}\mathbf{M} \\ &= \frac{d^2J(x_{opt})}{dx^2}\end{aligned}$$

- $\mathbf{C}(\vec{x})$ does not depend on values, only uncertainties and model

How to solve it?
1. Minimisation

Efficient minimisation algorithms use J(x) and the gradient of J(x) in an iterative procedure.

Typically the prior value is used as starting point of the iteration.

The gradient is helpful as it always points uphill.

The adjoint is used to provide the gradient efficiently.

<u>Example: Newton algorithm for minimisation</u>

Gradient: $g(x) = dJ/dx(x)$

Hessian: $H(x) = dg/dx(x) = d^2J/dx^2(x)$

At the minimum, x_{min}: $g(x_{min}) = 0$, hence:

$$g(x) = g(x) - g(x_{min}) \sim H(x)\,(x - x_{min})$$

rearranging yields:

$$(x_{min} - x) \sim - H^{-1}(x)\, g(x)$$

Smart gradient algorithms use an approximation of H(x)

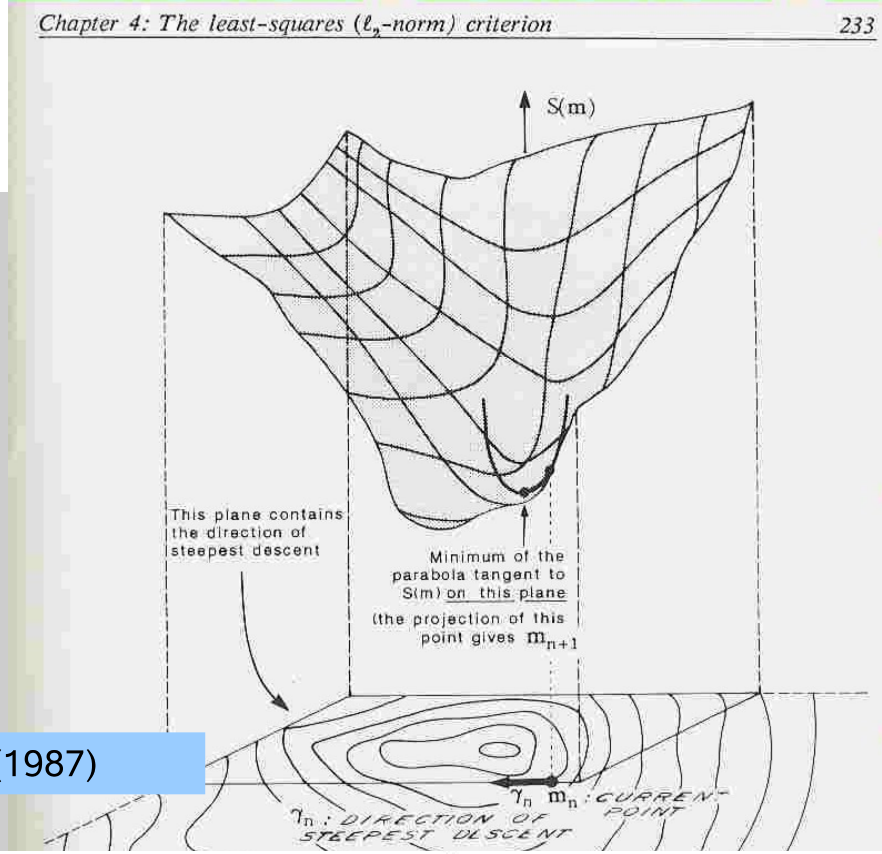


Figure: *Tarantola* (1987)

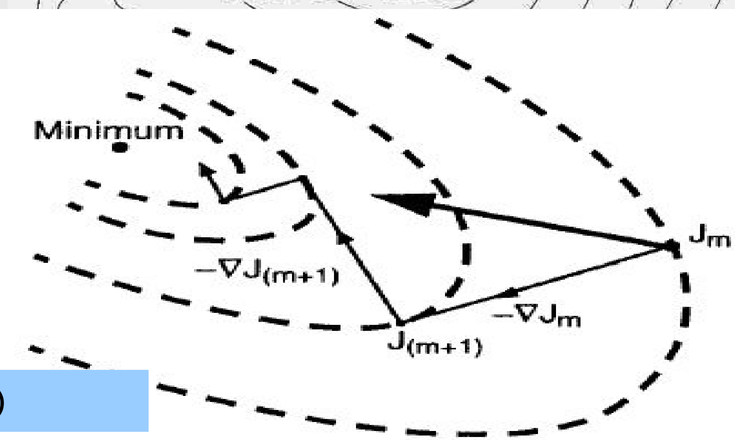


Figure: *Fischer* (1996)

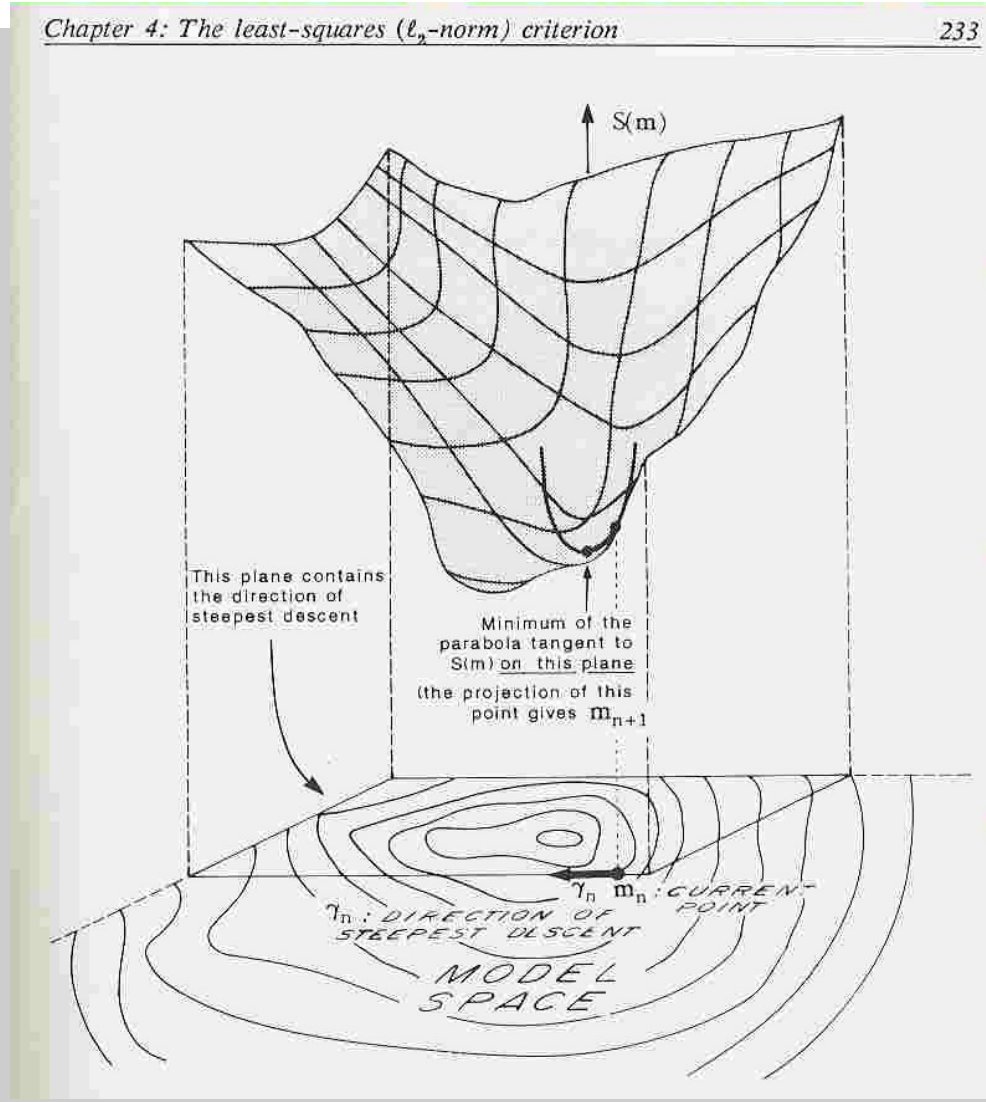
FastOpt

How to solve it?
2. Error bars

Hessian quantifies the curvature of the cost function

Use Hessian at minimum to approximate $C(x)^{-1}$

Figure taken from Tarantola (1987) -->



Exercise

Do algebra for simple case

- Gaussian
- 1 control variable
- 1 observation
- Model: identity