

Numerical derivative

1 INTRODUCTION

A number of metrological processes do not directly measure the quantity of interest, but its gradient. This is the case, for example, of deflectometric measurements, which measure the shape gradient of the surface of a mirror.

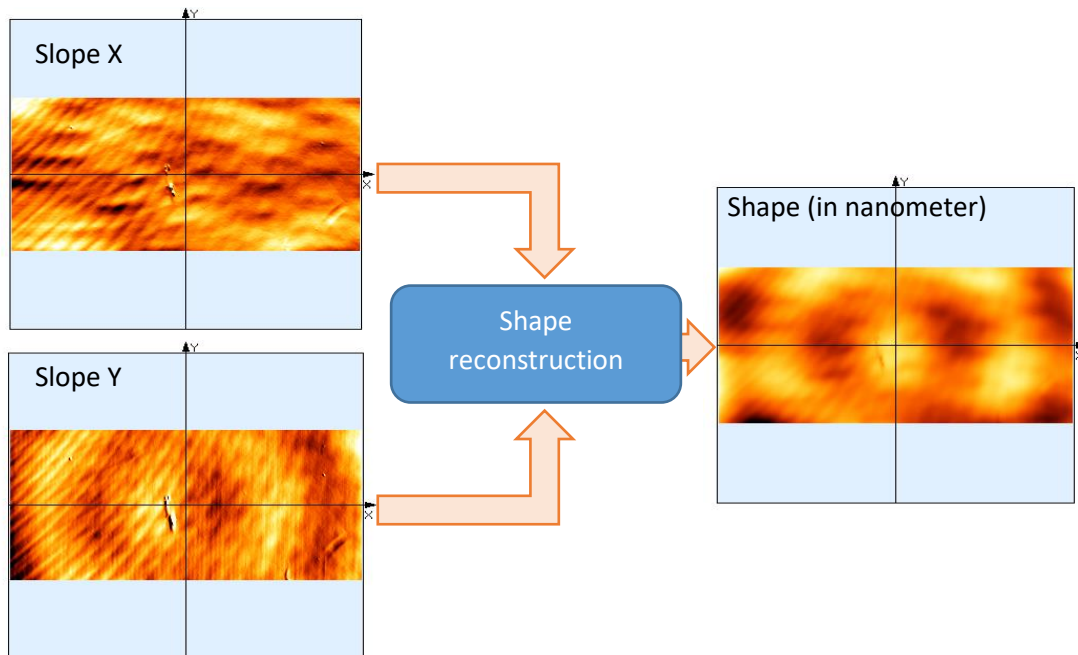


Figure 1: Example of shape reconstruction. Data from a freeform mirror (spacial application).

Therefore, a reconstruction algorithm is necessary to estimate the shape of the surface of a mirror. During this project, you will code a shape reconstruction algorithm, which will allow you to become familiar with the notion of ill-conditioned problem.

You will implement a solution to this problem by a Bayesian regularization method, and then analyze its behavior on simple cases before applying it to data from a deflectometric measurement on a satellite mirror.

2 DIRECT PROBLEM :

The problem we are trying to solve can be defined as an inverse problem:

$$\vec{\nabla}\phi(x, y) \rightarrow \phi(x, y)$$

where $\phi(x, y)$ is the shape function describing the height of the surface for any point (x, y) .

With its corresponding direct problem:

$$\phi(x, y) \rightarrow \vec{\nabla}\phi(x, y)$$

Before any implementation of shape reconstruction, we will start this project designing solving the direct problem, i.e. a program that can estimate the gradient of any input shape. The gradient of a surface is defined as:

$$\vec{\nabla}\phi = \left(\frac{\partial\phi}{\partial x}, \frac{\partial\phi}{\partial y} \right) = \left(\lim_{h \rightarrow +\infty} \frac{\phi(x+h, y) - \phi(x, y)}{h}, \lim_{h \rightarrow +\infty} \frac{\phi(x, y+h) - \phi(x, y)}{h} \right)$$

The obvious difficulty lies in the fact that we need an infinitely sampled signal to apply the analytical definition of the gradient. To bypass this problem, numerical derivatives are estimated from closest neighbor values, and for a discrete sampled surface shape $\phi(x, y) = \phi_{i,j}$ we obtain:

$$\vec{\nabla}\phi = \left(\frac{\phi_{i+1,j} - \phi_{i-1,j}}{2h_0}, \frac{\phi_{i,j+1} - \phi_{i,j-1}}{2h_0} \right)$$

Where h_0 is the lateral step between two values of $\phi_{i,j}$.

- What is the definition of a convolution? Why is a convolutive formalism adapted to derivative estimation?
- Demonstrate that if we vectorize $\phi_{i,j} = \phi_k, (k = (N-1) \times j + i)$:

$$\frac{\partial\phi}{\partial x} = \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2h_0} = \phi * h_x$$

Where $\vec{h} = \left(0, \frac{1}{2h_0}, 0, \dots, 0, -\frac{1}{2h_0} \right)$

- What would be the form of $\frac{\partial\phi}{\partial y}$?
- Why would a Fourier basis be adapted to a convolution formalism?
- Using the numpy function `numpy.fft2`, create a function that estimate the gradient of any input image.
- Compare the results on an image of your choice between your algorithm and the function `numpy.gradient.z`. What do you notice? How can you explain this difference?

3 NOISE STUDY

- Generate a Gaussian white noise matrix of standard deviation $\sigma = 1$.
- What is the spatial power spectral density of this noise?
- Using the `numpy.fft2` function, calculate the power spectral density of the white noise.
- Is the Parseval theorem valid in this case?
- Estimate the gradient of this noise, and calculate his power spectral density.
- What do you notice? Calculate the transfer function of your derivative algorithm.