Introduction

This coursework is designed to allow you to learn about inversion, regularisation and optimisation of relatively large-scale problems.

In directory

`http://www.cs.ucl.ac.uk/staff/S.Arridge/teaching/optimisation/CW2/`

you will find a number of standard test images.

In the following we give a guideline, which tasks you should be able to finish in the lab session of each week.

Tasks

Week 1: 31st January

---

1. Convolution and deconvolution

We want to perform convolution, such that

$$g = Af_{true} + n$$

where A denotes two-dimensional convolution with a Gaussian with standard deviation σ and n denotes additional noise of standard deviation θ. Make σ and θ variables of the process.

a.) Read a (grayscale) image of your choice and convert it to a float, normalise it such that the values are in $[0, 1]$ and display the image.

b.) Set up a convolution mapping: Convolution can be done explicitly by setting up a matrix A as we have done in the last coursework for 1D. This takes far too much memory in 2D. Instead write a function that takes in an image f and outputs the blurred image Af.

c.) Deconvolve using normal equations, i.e. find f_α as the solution to

$$(A^\mathsf{T} A + \alpha I) f_\alpha = A^\mathsf{T} g.$$

You can solve the above problem using a Krylov solver such as preconditioned conjugate gradients (PCG) or GMRES. Since the explicit matrix representation of A is infeasibly large, pass the solver instead a function that computes $(A^\mathsf{T} A + \alpha I)f$:

$\mathtt{z} = \mathtt{ATA}(\mathtt{f}, \alpha)$

which performs the two step process

$\mathtt{y} = \mathtt{A}(\mathtt{f})$
$\mathtt{z} = \mathtt{A}^\mathtt{T}(\mathtt{y}) + \alpha\mathtt{f}$

d.) Rather than using the normal equations, numerical analysis suggests that it is preferable to solve the augmented equations

$$\begin{pmatrix} A \\ \sqrt{\alpha} I \end{pmatrix} f = \begin{pmatrix} g \\ 0 \end{pmatrix},$$

which can be done by a least squares solver (lsqr). Compare the performance to the one you used in c.), in terms of number of iterations required to achieve convergence.

Week 2: 7th February

---

2. Choose a regularisation parameter α

Use the following two methods to choose an optimal value for α for the solution in task 1:

i.) Discrepency Principle

ii.) L-Curve

See lectures for details. Comment on the difference in value obtained and the results using these two values.

3. Using a regularisation term based on the spatial derivative

In Question 1, we solve the regularised least square problem

$$f_\alpha = \arg\min_f ||Af - g||_2^2 + \alpha||f||_2^2,$$

where $||f||_2$ is chosen to be the regulariser. In this Question, we are going to use a new regulariser $||Df||_2$ to penalise the gradient instead. So the new problem will be

$$f_\alpha = \arg\min_f ||Af - g||_2^2 + \alpha||Df||_2^2.$$

a.) You need to construct the gradient operator

$$D = \begin{pmatrix} \nabla_x \\ \nabla_y \end{pmatrix},$$

which can either be done as an explicit sparse matrix, or you can implement it as a function like the forward convolution.

b.) Solve the gradient regularised problem with both solvers from exercise 1, that means repeat 1c.) and 1d.) with the new term.

For the Krylov solver, remember to derive the needed normal equation first. In both cases, remember to include the transpose D^T properly.

c.) Chose a value for α, explain your choice.

Week 3: 14 February

4. Construct an *anisotropic* derivative filter

Rather than using the isotropic regulariser on the gradient, we can add weights and use an anisotropic regulariser $||\sqrt{\gamma}Df||_2^2$. With the *anisotropic* derivative filter, we turn to solve

$$f_\alpha = \arg\min_f ||Af - g||_2^2 + \alpha||\sqrt{\gamma}Df||_2^2$$

Here γ is termed the *diffusivity*. You should make γ a diagonal matrix with values between 0 and 1. Places where $\gamma = 0$ will not be smoothed by the regularisation term. You would ideally set γ based on the values of the edges in f_{true}, but since this is not known (it is what you are trying to find!) you should use the edges in the data. Note that after defining γ it is fixed for the optimisation procedure; in part 5) we will consider varying it during optimisation.

An example diffusivity is the Perona-Malik function

$$\gamma(f) = \exp(-|Df|/T) \quad = \exp(-\sqrt{(\nabla_x f)^2 + (\nabla_y f)^2}/T)$$

for some threshold T based on the maximum expected edge values in the image; this can be estimated from the norm of the image gradient. Note that $\sqrt{\cdot}$ here is an element-wise operation on the diagonal matrix γ, so to calculate $\sqrt{\gamma}D$ we use

$$\sqrt{\gamma}D := \begin{pmatrix} \sqrt{\gamma}\nabla_x \\ \sqrt{\gamma}\nabla_y \end{pmatrix}.$$

Note : The quantities γ, $|Df|$, $\nabla_x f$, $\nabla_y f$ are all defined *at each pixel*. It may help to display them as images to aid your understanding of their meaning. The square operation on vectors $\nabla_x f$ and $\nabla_y f$ are also element-wise, as well as the square root operation and exponential operation.

5. Iterative deblurring

Repeat the deblurring process of Task 4 *iteratively*:

i.) Take an initial blurred image f_0; Set $i = 0$

ii.) Compute the diffusivity $\gamma(f_i)$.

iii.) Compute a solution f_{i+1} by following the process in Task 4.

iv.) Increase $i = i + 1$; Repeat from ii.)

Choose how to decide on the number of such iterations to use, i.e. when to terminate the loop.

Report

Write one report for all parts. Explain your method and present your results and figures. Make sure that you provide an answer to all questions. The total length of the report would normally be between 6-10 pages. Submit your report as a PDF file using Moodle. Code can be uploaded separately.