GV08 Optimisation. Exercise 1

Introduction

This unmarked exercise is designed to prepare you for the marked course work. It repeats some of the basic Matlab techniques on examples taken from the lecture notes.

Tasks

1. Minimum $p-$Norm Solutions to Underdetermined Problems

In Section 2.1,1, the linear equation $x_1 + 2x_2 = 5$ is used to introduce the concept of minimum norm solutions of underdetermined problems:

$$\text{minimize} \quad \Phi = \sum_i |x_i|^p, \qquad \text{subject to} \quad Ax = b,$$

where in our case, $A = (1\ 2)$, $b = 5$.
Familiarize yourself with the concept of an *anonymous function* or *function handle* in Matlab and define an anonymous function `Phi(x,p)` that computes Φ as defined above for a given p and a vector x (of arbitrary length).
For a given p, the above optimization problem can be solved by calling Matlab's `fmincon` function (study its documentation to figure out the correct order of arguments). Compute the solutions for $p = 1, 1.5, 2, 2.5, \ldots, 4$.
Familiarize yourself with Matlab's 2D graphics (MATLAB > Graphics > 2-D and 3-D Plots > Line Plots) to plot your results as shown in Figure 1: The blue line represents the space of all solutions to $x_1 + 2x_2 = 5$, while the red asterisks represent the computed minimum norm solutions for different p. The axis should have the same limits and scaling.
To export the line plot, consult the documentation (MATLAB > Graphics > Printing and Saving > Save Figure for Document or Presentation) and use the functions `print` or `saveas` to save the graphic in a vector format (depended on which program you use to write your report, choose `.eps`, `.pdf` or `.svg`). Clip the exported image if necessary (see "Tools").
While the *Moore-Penrose generalised inverse* can be computed as `A' * ((A * A') \ b)` or `pinv(A)*b`, and yields the $p = 2$ minimum norm solution, Matlab's canonical operator to solve linear equations, the *backslash operator* `A\b`, yields a different solution. Which one?

2. Singular Value Decomposition For a given integer n define spatial grids x and y by

$$x_j = (j-1)/(2n-1), \quad j = 1, \ldots, 2n, \qquad y_i = (i-1)/(n-1), \quad i = 1, \ldots, n \tag{1}$$

using `linspace`. We want to implement a discrete convolution of a 1D function f defined on grid x with a Gaussian kernel ($\sigma = 0.2$) followed by an evaluation on the grid y:
Construct the corresponding matrix A as

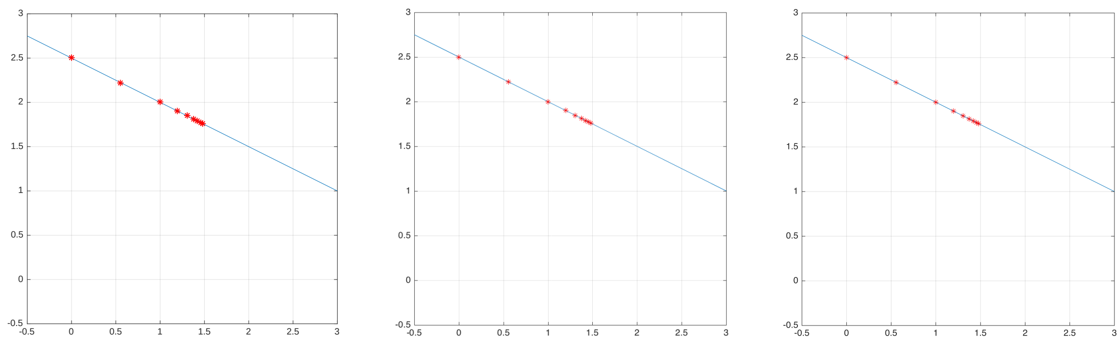
$$A_{i,j} = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - x_j)^2}{2\sigma^2}\right) \tag{2}$$

Firstly, we want to visualize A as an image for $n = 100$. One way is to use Matlab's `imagesc` function to plot A and and use `saveas` to export the plot as a `.png`. Often, it is advantageous to export images directly into a raster graphic format. This can be done using Matlab's `imwrite` function, but requires a rescaling of A and the explicit definition of a *colormap*:

```
>> Aimg = ceil(A/max(A(:))*256);
>> colorMap = parula(256);
>> imwrite(Aimg,colorMap,'Aimage2.png')
```

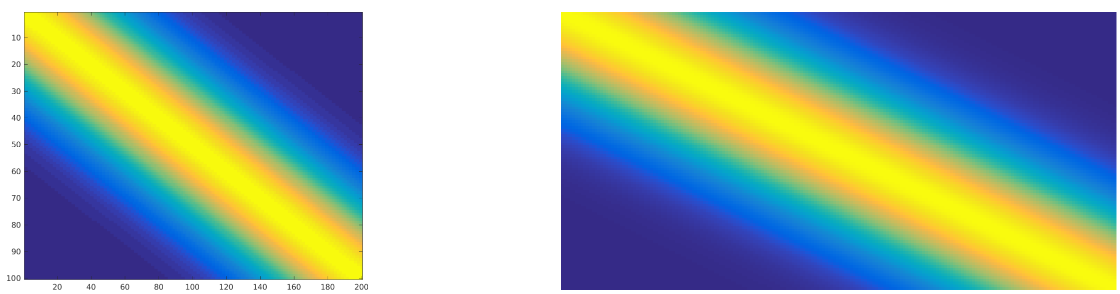
Consult the documentation to comprehend the above commands (cf. Figure 2).
Use Matlab's `svd` function to compute the SVD (cf. Section 2.3.1) of A for $n = 10$ and verify $A = UWV^T$. Construct the pseudoinverse $W^\dagger$ of W as a *sparse matrix* by using the `spdiags` function (familiarize yourself with the concept of sparse matrices). Use the function `spy` to check

whether $W^\dagger$ has the desired form and verify that $A^\dagger = VW^\dagger U^T$ by using Matlab's `pinv` function to compute $A^\dagger$.
Repeat the last steps for $n = 20$. What do you observe? Choose $n = 100$ again and plot the first 9 columns of V, the last 9 columns of V and $\log(\mathrm{diag}(W))$. Compare with Exercise 3, page 16.



(a) pdf, 4kB (b) png, 12kB (c) png, high res, 76kB

Figure 1: Example for a line plot exported as a pdf vector graphic or as a png raster graphic format (low and high resolution) and embedded into this LaTex document (which is compiled into a pdf document). Zoom into the subfigures to examine the differences.



(a) exported by imagesc.m, 36.1kB (b) exported by imwrite.m, 2.6kB

Figure 2: Example for an image exported to png via the imagesc.m function or the direct export via imwrite.m.

Tools

Gimp (www.gimp.org/) is a free, cross-platform raster-graphics editor. To crop an image, simply load it to Gimp, select "Image" > "Autocrop Image" and overwrite the image. Inkscape (inkscape.org) is a free, cross-platform vector-graphics editor. One way to crop a pdf is to open it in Inkscape, select "File" > "Document Preferences", check "Show canvas border" (if not already checked) and set canvas size to "Custom". Modify the bounding box and overwrite the pdf. PDFCrop (pdfcrop.sourceforge.net/) provides a convenient, automated way to crop pdfs but is a command line tool. It comes with most installations of LaTex.

Report

In the marked course work, you will have to write short reports, in which you describe the steps you took, present the results you obtained and draw conclusions. As the reports will be marked, we recommend that you practice for them by writing a short report for this exercise.