MAI0199 Computational Linear Algebra

Computer Exercise 5. The Conjugate Gradient Method

1 General information

The assignment consists of a mixture of theoretical exercises and practical programming. At the end of the exercise a written report with well structured solutions to the theoretical questions and also Matlab programs, and graphs or plots that summarize the computational results should be sent by email to `Fredrik.Berntsson@liu.se`. In order to reduce the number of Matlab programs keep old code as comments when you modify a program in an exercise.

2 The Conjugate Gradient Method

The *Conjugate gradient method* is as follows:

$r^{(0)} = b - Ax^{(0)}$, $p_0 := r^{(0)}$.

`for` $j = 1, 2, \ldots$ `do`

$\alpha_j = (r^{(j)}, r^{(j)})/(Ap_j, p_j)$.
$x^{(j+1)} := x^{(j)} + \alpha_j p_j$.
$r^{(j+1)} := r^{(j)} - \alpha_j A p_j$.
$\beta_j := (r^{(j+1)}, r^{(j+1)})/(r^{(j)}, r^{(j)})$.
$p_{j+1} := r_{j+1} + \beta_j p_j$.

`end`

Provided that the matrix A is symmetric and positive definite then the above algorithm converges to the solution of the linear system $Ax = b$.

Exercise 2.1 Write a Matlab function that implements the above formulas. The function should be used as follows:

```
>>[ x , Residuals ]=ConjugateGradient( A , b , x0 , MaxIter , tol );
```

where `Residuals` is a vector of the residuals during the iterations. The stopping criteria should be based on the relative residual $\|r\|_2 \leq$ `tol`$\|b\|_2$. $\square$

Exercise 2.2 Load the `bundle1` test problem in Matlab and calculate the dimension and number of non-zeros for the matrix A.

Calculate the LU decomposition of A. How many non-zeros are there in the L and U matrices? □

The above shows that an iterative solver can potentially be alot more efficient than a direct one.

Exercise 2.3 Use your function to solve the linear system $Ax = b$; using $x_0 = 0$. Terminate the iterations when the relative residual is smaller than tol $= 10^{-10}$. How many iterations are needed? □

Hint The starting residual is quite large hence the relatively large number of iterations. Plot the residuals using `semilogy` to clearly see the convergence speed.

3 Image Reconstruction

Sometimes when looking at an object from a distance we may not see the true shape exactly. The light is diffused and the image may appear "blurred". Also if the object is moving it may appear as if it is spread out in one direction. In *image reconstruction* this is modelled by a point spread function. Using the point spread function we obtain a matrix A such that $b = Ax$, where x is the exact image and b is the image actually recorded by the camera. The image reconstructing problem consists of finding the exact image x by solving the linear system $Ax = b$.

In this exercise the blurring is assumed to be described by a Gaussian function. A single pixel $P(i_0, j_0)$ in the original image will be diffused over a few nearby pixles in a manner described by a function.

$$P_b(i,j) = \exp\left(\frac{-(i-i_0)^2 - (j-j_0)^2}{2\sigma^2}\right).$$

The Matlab function `BlurMatrix` calculates a matrix representing the "blurring part" of the point spread function for a given image size. Type

```
>>[A]=BlurMatrix(N,band,sigma)
```

and an $N^2 \times N^2$ matrix A is computed. The parameters `band` and `sigma` control the shape of the Gaussian. If we store the original image as an $N^2 \times 1$ vector x then the "blurred image" is the vector $b = Ax$.

Load the images `Goose.mat` and display the test image by typing

```
>> load Goose.mat
>> imshow(GooseBW,'InitialMagnification','fit')
```

The vector representing the noisy image is created by

```
>> N=256; x = reshape( GooseBW , N*N , 1 );
>> [A]=BlurMatrix(N,band,sigma);
>> b=A*x+randn(size(x))*10^-4;
```

where random noise representing the errors in the recording device, i.e. the camera, and modelling errors in the point spread function has been added.

Exercise 3.1 Let $N = 256$, $\sigma = 1.9$ and `band`$= 13$ for the blurring. Compute the image we will see in the camera with noise as detailed above. Look at the image. Is the blurring effect clearly visible? □

Hint In order to display the image use `reshape` to transform the vector x back into a quadratic matrix X.

Let b be the vector representing the image recorded by the camera. Since the matrix A is very ill-conditioned we can't solve the linear system $Ax = b$ using Guaussian elimination and expect a good solution. Try this and see what happens. Instead we will use Tikhonov regularization. For a given value of λ minimize,

$$\|Ax^\lambda - b\|_2^2 + \lambda^2\|x^\lambda\|_2^2.$$

This minimization problem can be solved efficiently using the singular value decomposition. This is the goal in the next few exercises.

Exercise 3.2 Show that the normal equations for the above minimization problem are

$$(A^TA + \lambda I)x = A^Tb. \qquad \square$$

Exercise 3.3 Since the normal equations are symmetric and positive definite we can solve the problem $Ax = b$ using the Conjugate Gradient Method. Experiment, in the range $10^2 < \lambda < 10^{-2}$, and compute the Tikhonov solutions x^λ. Look at the resulting images and find a good value for the requlariza tion parameter λ? Is it possible to find a good reconstruction of the image?

Hint Since the product A^TA is dense you can't directly use your Conjugate Gradient function. Instead rewrite the code so that the normal equations are solved, i.e. write a function

```
>>[ x , Residuals ]=ConjugateGradientTikhonov( A , b , x0 , Lambda, tol );
```

Matlab has both `pcg` and `cgls` functions for the same reason. A maximum number of iterations is no longer needed since the regularization parameter λ limmits the condition number for the linear system and hence the convergence will in practice be fast.

Exercise 3.4 Let $B = A^TA + \lambda^2 I$. Show that $(Bp_j, p_j) = \|Ap_j\|_2^2 + \lambda^2\|p_j\|_2^2$. Use the result to simplify the CG algorithm. Verify that you still get the same solution.

Exercise 3.5 The condition number of the normal equations can be approximated

$$\kappa_2(A^TA + \lambda^2 I) = \frac{\sigma_1^2 + \lambda^2}{\sigma_n^2 + \lambda^2} \approx \frac{\sigma_1^2}{\lambda^2}.$$

Set `tol`$= 10^{-8}$ and let λ vary in the interval 10^2 to 10^{-2}. Record the number of CG iterations needed as a function of λ and verify that the convergence rate for CG depends on $\sqrt{\kappa_2}$.