Exercises – Krylov Subspace Methods and Regularization Tools

Per Christian Hansen, DTU Compute

Preparation

Download the Regularization Tools MATLAB software from http://www2.imm.dtu.dk/~pch/Regutools, unpack the zip file, and place all the functions in a directory (preferably the *same* directory as where you put the AIR Tools files). Start MATLAB and go to the directory. Have fun with the exercises!

Exercise 1: Illustration of the CGLS algorithm

The purpose of this exercise is to illustrate the use of regularizing iterations from the CGLS algorithm, which is implemented in Regularization Tools as the function **cgls**. The model problem is a small image deblurring problem which is generated by means of the function **blur** from the same package. Both the exact image and the blurred image are $N \times N$, and they are represented by N^2-vectors by stacking their columns. Use **reshape(x,N,N)** to get from the "stacked" representation of the image to the image itself. To display[1] the images, you can use the function **imagesc** together with the commands **axis image** and **colormap gray**.

1. Choose $N = 64$ and generate the test problem with the call **blur(N,10,1.4)** (the second parameter controls the sparsity of A, and the third parameters controls the amount of blurring). Plot the sharp and the blurred $N \times N$ images. At this stage, do not add noise to the problem.

2. Perform a number of CGLS iterations – note that the CGLS iterates are returned as columns of the output matrix from **cgls**. Plot some of the iterates as images, and notice how the reconstruction gets sharper as the number of iterations increases.

3. Now add noise e to the blurred image with the noise level $|| e ||_2 / || b ||_2 = 0.1$, and repeat the CGLS computations. You should notice that after a certain number of steps, the noise starts to dominate. This illustrates that the number of iterations indeed plays the role of the regularization parameter for CGLS.

Exercise 2: CGLS versus TSVD

In this example we consider a test problem whose matrix A is from the **deriv2** test problem in Regularization Tools, and the exact solution is a "sawtooth function"  constructed such that only every fourth SVD component is nonzero. The following code constructs this solution and the corresponding right-hand side, assuming that the problem dimension n is a multiple of 4:

```
nq = n/4;
x1 = (0:nq-1)' + 0.5;
x2 = (nq:-1:1)' - 0.5;
xex = [x1;x2;-x1;-x2];
bex = A*xex;
```

[1]If the Image Processing Toolbox is available, use **imshow(X,[])** to display image **X**.

Your task is to compare the solutions computed by means of TSVD and CGLS, for $k = 1,\dots,n$, using the functions **tsvd** and **cgls** from Regularization Tools. Use the noise level $|| e ||_2 / || b ||_2 = 10^{-5}$. Plot the error histories for the two methods, and comment on the results. Which method gives the most accurate solution?

Exercise 3: Which is better: CGLS or RRGMRES?

This exercise illustrates the regularizing properties of CGLS and RRGMRES. Your task is to compare the two methods for solving the two test problems **baart** and **i_laplace** from Regularization Tools. We suggest you use the problem dimension $n = 64$, the very small noise level $|| e ||_2 / || b ||_2 = 10^{-8}$, and a total of $k = 16$ iterations for both methods. Since RRGMRES always orthogonalizes the basis vectors for the Krylov subspace $\mathcal{K}_k$ the comparison is most fair if you also use reorthogonalization in CGLS (set **cgls**'s fourth input parameter to **1**).

1. Compare the error histories for the two methods. For each of the two test problems, which iterative method is better?

2 (optional). If you want to study the convergence histories in details, you need to investigate the expansion of the exact solution x^* in the orthonormal basis vectors of the two different Krylov subspaces associated with the two iterative methods. You can compute the first basis as the third output argument from **lanc_b(A,b,16,2)**, where the fourth input argument enforces accurate reorthogonalization of these vectors. The following code computes an accurate basis for the second basis:

```
Khat = A*b; K = Khat/norm(Khat);

for i=2:kmax
   v = A*K(:,i-1);
   for j=1:i-1, v = v - (K(:,j)'*v)*K(:,j); end
   for j=1:i-1, v = v - (K(:,j)'*v)*K(:,j); end
   K(:,i) = v/norm(v);
end
```

Two orthogonalizations are needed in each loop to obtain the necessary accuracy.

Exercise 4: the NCP criterion for regularizing iterations

The implementation of the NCP criterion in regularization Tools, in the function **ncp**, computes the residual vectors given the right-hand side b and the SVD of A. This is not suited for iterative regularization methods, and your task is to write an implementation of the NCP criterion, based on **ncp**, that takes a number of residual vectors as input (e.g., in the form of a matrix whose columns are the residual vectors). Your function must find that residual vector whose NCP is closest to a straight line.

Use your function on the image deblurring test problem from Exercise 1, and compare with that solution which is closest to the exact solution x^*. Does the NCP criterion work for this problem? How does the performance of this criterion depend on the relative noise level and the amount of blurring (controlled by the third input parameter to **blur**)?