HOMEWORK 2
for Mathmatical Models, Analysis and Simulation, Fall 2010
Report due Mon Sep 20, 2010.
Maximum score 3.0 pts.

Read chapter 1 and section 2.3 of Strang.
Solve the following problems:

1) **1.7**: 10,11
   2.3: 7,8

2) Show that all eigenvalues are zero for a nilpotent matrix. (A square matrix $\mathbf{A}$ is nilpotent if $\mathbf{A}^k = 0$ for some positive integer k).

3) This is a modified version of P6.3.4 in Charles van Loan: Introduction to Scientific Computing. $\mathbf{A}$ and $\mathbb{C}$ are given n-by-n matrices, $\mathbf{A}$ is nonsingular. g and h are given n-vectors. Develop different algorithms (at least three) for computing n-vectors $\mathbf{y}$ and $\mathbf{z}$ so that

$$\mathbf{A}^T\mathbf{y} + C\mathbf{z} = g, \quad \mathbf{Az} = h.$$

   Implement them in Matlab and measure their run-times. Choose test matrices of different sizes (say, $n = 500$, 1000, 1500, 2000) and measure the execution time. **Explain** your observations.

   The following code snippet shows how to obtain timing resluts :

```
tstart = cputime;
% insert your code to be timed
tend = cputime;
duration = tend-tstart;
```

   A random square matrix can be generated by rand which gives (pseudo-) random numbers uniformly distributed in $[0, 1]$:

```
n = 100;
A = 2*eye(n)+rand(n);
C = rand(n);
```

   Note Is A sure to be non-singular? No, but the probability of finding a singular one is zero. You may find it amusing to plot the spectrum of a large number of As - use `lam = eig(A); plot(lam,'.'); hold on;` .

4 In this exercise we will use the SVD for image compression. Go to the course homepage and download `basketpic.mat`. Load it into matlab.

This is a 256×256 matrix. It contains integers between 0 and 255, building up a gray scale picture. To view it, do:

`imagesc(basketpic);`

and possibly, if it displays in color:

`colormap(gray);`

Now, convert it into a matrix of type double:

`Amat=double(basketpic);`

i) Do a singular value decomposition of the matrix, using Matlab's `svd` command. ($\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$)

ii) Define the Frobenius norm of a matrix:

$$\|A\|_{Fro} = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} |A_{ij}|^2}.$$

Verify that for our matrix (to some precision in Matlab):

$$\|A\|_{Fro} = \sqrt{\sum_{i=1}^{n} \sigma_i^2}$$

with $n = 256$. Show that this is true for a general $m \times n$ matrix $\mathbf{A}$. (Hint: What is the trace of $\mathbf{A}^T\mathbf{A}$?).

iii) Plot the singular values against the position in the diagonal of $\mathbf{\Sigma}$ (i.e. your x-axis should include $i = 1, 2, \ldots, 256$). Use the `semilogy` command to get a log-scale on the y-axis.

iv) Define $\mathbf{A}_k$ to be the rank k approximation of $\mathbf{A}$, i.e.

$$\mathbf{A}_k = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T + \ldots + \sigma_k \mathbf{u}_k \mathbf{v}_k^T = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$$

Define $\mathbf{A}_k$ for different values of k, and use the commands above to view the resulting image. How large must the rank k be before the image is almost perfect to your eyes?

v) Define the relative error in the Frobenius norm:

$$E_k = \frac{\|\mathbf{A}_k - \mathbf{A}\|}{\|\mathbf{A}\|}$$

Pick 4 different values of k. Plot the resulting images for these four values of k in one figure using Matlab's `subplot` command. For each of them, display E_k.