Mathematical Models, Analysis and Simulation
Part I, Fall 2008

August 29, 2008

Homework 2, Strang Ch. 1. Max. score 3.0

1. **(2.3)**
   1.6:15,19(p75ff);**1.7:**2,10,11,12(p88);
   from *G.Strang, Intro. to Applied Mathematics*: **1.6:**2,20,29 (pages copied and handed out)

2. **(0.5)**
   This is a modified version of P6.3.4 in *Charles van Loan: Introduction to Scientific Computing.* A and C are given n-by-n matrices, A is nonsingular. g and h are given n-vectors. Develop different algorithms (at least three) for computing n-vectors y and z so that
   $$A^T y + Cz = g, Az = h$$

   Implement them in `Matlab` and measure their run-times. Explain your observations. Choose test matrices of different sizes (say, n = 500, 1000, 1500, 2000) and measure the execution time. The following code snippet gives a hint:

   ```
   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 A's - use `lam = eig(A); plot(lam,'.'); hold on;`.

3. **(0.2)**
   The game changes when C is non-singular, $n \times n$, but A is $m \times n, m < n$. Then h is an m-vector as is y. Why not $m > n$? Eliminate y and show that the resulting system for z has coefficient matrix $AC^{-1}A^T$ - called the Schur complement.