CS 621: Assignment 1

Due: Friday, September 20, 2002 (In Lecture)

Scoring on each problem is on a 0-1-2-3 scale. 3 = complete success, 2 = overlooked a small point, 1 = germ of the right idea, 0 = missed the point of the problem. One point will be deducted for insufficiently commented code. Test drivers and related material will be posted on the course website. For each problem submit output and a listing of all the scripts/functions that you had to write/modify in order to produce the output. You are allowed to discuss *background* issues with other students, but the codes you submit must be your own.

Problem 1. Recall *Horner's* nested multiplication scheme that is normally used for polynomial evaluation. To evaluate $p(x) = c_1 + c_2x + \cdots + c_dx^{d-1}$ at $x = z$ we proceed as follows:

```
pval = c(d);
for k=d-1:-1:1
   pval = z*pval + c(k)
end
```

Develop an efficient, fully vectorized saxpy implementation for the following MATLAB function

```
   function y = PolyVec(c,A,p)
% c is a column m-vector
% A is an n-by-n matrix with lower bandwidth p. Assume mp<n
% y  is the first column of c(1)I + c(2)A + c(3)A^2 + ... + c(m)A^{m-1}
```

Use the Horner idea and exploit A's band structure. You may find the discussion in §11.2.4 helpful. Test your code on the script `P1.m` that is available on the website. Submit output and a listing of `PolyVec`.

Problem 2. Suppose $Q \in \mathbb{R}^{n\times n}$ is orthogonal and $v \in \mathbb{R}^n$. Consider the matrix

$$M = \left[\ v \mid Qv \mid Q^2v \mid \cdots \mid Q^{n-1}v\ \right]$$

Think about the structure of the matrix $A = M^TM$. Implement the following MATLAB function so that it performs as specified:

```
   function y = KrylovProd(Q,v,x)
% Q is an n-by-n orthogonal matrix.
% v and x are column n-vectors
% y = M'*Mx  where M = [ v | Qv | Q^{2}v |...|(Q^{n-1}v]
```

Test your code on the script `P2.m` that is available on the website. Submit output and a listing of `KrylovProd`.

Problem 3. For given 2-by-2 real matrix A define the set $S_p(A) = \{\ Av \mid \|\, v\,\|_p = 1\}$. Write a MATLAB function `pNormImage(A,p)` that opens up a new figure window and plots $S_p(A)$. It should include the command `axis equal` so that the x and y axes have the same scale. The plot should be based on a couple of hundred points. Test your code on the script `P3.m` that is available on the website. Submit output and a listing of `pNormImage`. To do this problem you will have to generate a couple of hundred points around the set of all unit p-norm vectors. Aply A to each of these and then "connect the dots" in the range space. You code should be vectorized and efficient.

Problem 4. Define the matrices $W_2, W_4, W_8, \ldots$ as follows

$$W_{2m} = \begin{cases} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} & \text{if } m = 1 \\[2ex] \begin{bmatrix} W_m & W_m \\ W_m & -W_m \end{bmatrix} & \text{if } m > 1 \end{cases}$$

Complete the following MATLAB function so that it performs as specified:

```
  function y = FastProd(x)
% x is a column n-vector and n is a power of 2.
% y = W_{n}*x
```

Hint. Look at the product

$$\begin{bmatrix} W_m & W_m \\ W_m & -W_m \end{bmatrix} \begin{bmatrix} y \\ z \end{bmatrix}$$

where $y, z \in \mathbb{R}^m$. Your implementation should be recursive and involve $O(n \log n)$ flops. Take a look at the Strassen algorithm §1.3.7 for guidance on the design of a recursive matrix procedure. Test your code on the script `P4.m` that is available on the website. Submit output and a listing of `FastProd`.