Homework 4, Math 451, Spring 2013

due Friday, February 15th

This home covers material from Sections 3.1-3.5 of Mathews and Fink.

1. Matrix-Matrix multiplication

   (a) Present the results of the following code:

   ```
   >> y = [ 1,1; 2,1; 2,2; 1,2; 1,1 ]';
   >> plot( y(1,:), y(2,:), 'o-');
   >> axis([-4,4,-4,4]);
   ```

   (b) Show using appropriate Matlab or Octave plots how multiplication by the rotation matrix

   $$R = \begin{bmatrix} \cos(2\pi/5) & -\sin(2\pi/5) \\ \sin(2\pi/5) & \cos(2\pi/5) \end{bmatrix}$$

   changes the matrix in part a.

   (c) Show using an appropriate plot how multiplication by the matrix

   $$D = \begin{bmatrix} 1/2 & 0 \\ 0 & 1/3 \end{bmatrix}$$

   changes the matrix in part a.

2. Set up the linear algebra problems in matrix-vector form for the following stoichiometry problems and solve for x_1, x_2, and x_3. Check your answers by matrix multiplication.

   (a)

   $$x_1\, Al + x_2\, CuSO_4 \rightarrow x_3\, Cu + 5\, Al_2(SO_4)_3$$

   (b)

   $$x_1\, Pb(NO_3)_2 + x_2\, NaI \rightarrow x_3\, PbI_2 + 4\, NaNO_3$$

3. Solve the following linear systems for $\vec{x}$ by backward or forward substitution by hand and compare the amount of work required to that of previous problems.

   (a)

   $$\begin{bmatrix} 2 & -1 & 4 \\ 0 & 3 & -1 \\ 0 & 0 & 1 \end{bmatrix} x = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix}$$

   (b)

   $$\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} x = \begin{bmatrix} -5/2 \\ 1 \\ 2 \end{bmatrix}$$

4. In class, we presented the following program for backward substitution, which works when our initial matrix A is square and has a pivot in each column.

```
function x = back_sub(R)
        [m,n] = size(R);
        assert(m+1==n);
        for i = m:-1:1
                pivot = R(i,i);
                R(i,:) = R(i,:)/pivot;
                for j = 1:i-1
                        R(j,:) = R(j,:) - R(i,:)*R(j,i);
                end
        end
        x=R(:,n);
return
```

   (a) Determine the exact number of divisions and multiplications used in this algorithm, as a function of the number of rows m in R.

   (b) A FAST implementation of back-substitution would require only m divisions and $m(m-1)/2$ multiplications. Explain where this algorithm is wasting effort.

   (c) This script fails for any matrix in row-echelon-form with a zero-pivot. Correct the function, so that it will return a solution for any matrix in row echelon form (REF) with a pivot in each row. For columns lacking a pivot, you should assign those variables values of zero. So, for example, if

$$R = \begin{bmatrix} 1 & -8 & 2 & 3 \\ 0 & 0 & 2 & 4 \end{bmatrix},$$

   then your code should return $x = [-1, 0, 2]^T$. In addition to handing in your code, please submit your function on Angel so I can test it. **[TCR: To make things easier to test, change your function's output so it returns the full matrix, in reduced row-echelon form. This will also make automated grading easier. Make sure your dot-m function file is named "transform_ref_to_rref.m"]**

5. Scaled partial pivoting is another pivoting method which can be used to stabilize Gaussian elimination. In partial pivoting, we chose our pivot p_{jj} as the maximum entry in column j at or below j such that

$$p_{jj} = \text{absmax}\left\{A_{j,j}, A_{j+1,j} \ldots A_{m,j}\right\},$$

   where absmax() returns the entry with the largest absolute value. In scaled partial pivoting is the same, except that we first divide all the rows on or below the pivot by their largest value. Thus, scaled partial pivoting tries to maximize

$$\text{absmax}\left\{\frac{A_{i,j}}{\text{absmax}\left\{A_{i,j}, A_{i,j+1}, A_{i,j+2} \ldots A_{i,n}\right\}}\right\}$$

   over all $i \geq j$, and uses that value for p_{jj}. Since the entries in the rows change after every step, this has to be redone every time we move to a new pivot.

   Write a pivoting function as a drop-in replacement for the pivoting subroutine of our in-class "elim_with_piv.m" function that performs scaled partial pivoting. Use this to compute the REF of the matrix

$$M = \begin{bmatrix} 1 & 6 & 2 & 0 \\ 1 & 6 & 1 & 1 \\ 2 & 1 & 1 & 0 \\ 5 & 2 & 10 & -1 \end{bmatrix}.$$

   (note: when you do this, only one of your pivots should end up being 1.)