CS 370 Winter 2013: Assignment 4

Instructor: Yuying Li	Office: DC3623	e-mail: *yuying@uwaterloo.ca*
Lec 001 :	MWF 8:30-9:20 MC2054	OH (Li):Tues 2-3pm
TAs:		
Nian Ke	knian@uwaterloo.ca	DC3594
Mark Prosser	rmprosse@uwaterloo.ca	DC2302B
Haofan Zhang	h267zhan@uwaterloo.ca	DC3594

Web Site: www.student.cs.uwaterloo.ca/~cs370/

Due April 5, 2013, 5:00 PM. Submit to the Assignment Boxes, 3rd Floor MC, except for the Matlab content which is to be submitted electronically. Please attach a cover page, which you can download at the course website, to your paper assignment.

1. **(12 marks)** (LU factorization, Pivoting)

   (a) Find $A = LU$ factorization by hand **without** using (row) partial pivoting for the following matrix

   $$A = \begin{bmatrix} 10 & 20 & 1 \\ 1 & 1.99 & 6 \\ 0 & 50 & 1 \end{bmatrix}$$

   What is the largest magnitude l_{ij}?

   (b) Find $PA = LU$ factorization by hand **with** using (row) partial pivoting for the matrix A. What is the largest magnitude l_{ij} now?

   (c) Solve $Ax = b$ using the factorization in (b) for

   $$b = \begin{bmatrix} 12 \\ 13 \\ 2 \end{bmatrix}.$$

2. **(12 marks)** (Computation of Periodic Spline)

   For periodic data, the boundary conditions for a cubic spline interpolant can be requiring that both the first-order and second-order derivatives are equivalent at the end points x_1 and x_n, i.e.,

   $$S_1'(x_1) = S_{n-1}'(x_n) \quad \text{and} \quad S_1''(x_1) = S_{n-1}''(x_n).$$

   (a) Using above boundary conditions and (2.19) on page 25 in the Course Notes, derive the linear equations which determine the required derivative values $s = [s_1; \cdots; s_n]$ for the cubic spline interpolant. The first equation corresponds to the condition $S_1''(x_1) = S_{n-1}''(x_n)$, and the second to the $(n-1)^{th}$ equation from (2.19) (for $i = n-1$). Show your work.

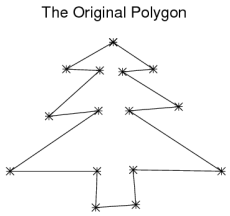
   (b) Write a Matlab function $[a, b, c, d]$=**MySpline**(x, y) to generate $[a, b, c, d]$ representing such a cubic spline interpolant, using the provided Matlab functions $[L, U, p]$ = **lutx**(A) and x = **bslashtx**(A, b) to perform LU factorization and triangular solves respectively. The preamble for $[a, b, c, d]$=**MySpline**(x, y) is given below

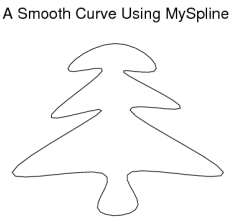
function $[a, b, c, d] =$ **MySpline**(x, y)
%
% Pre:
% x, y column n-vectors. $n \geq 4$ and $x(1) < \cdots < x(n)$
%
% Post:
% a, b, c, d column $(n-1)$-vectors that define the spline
%
% On $[x(i), x(i+1)]$, the spline $S(x)$ is specified by the cubic
%
% $$S_i(x) = a(i) + b(i)(x - x(i)) + c(i)(x - x(i))^2 + d(i)(x - x(i))^3$$
%
% $S(x)$ satisfies the conditions:
%
% $$S_i'(x(i+1)) = S_{i+1}'(x(i+1)), \quad i = 1 \cdots, n-2$$
% $$S_i''(x(i+1)) = S_{i+1}''(x(i+1)), \quad i = 1 \cdots, n-2$$
% $$S_1'(x(1)) = S_{n-1}'(x(n))$$
% $$S_1''(x(1)) = S_{n-1}''(x(n))$$
%

(c) Use the script below to read a sequence of points denoting the vertices of a polygon.

```
figure('position',get(0,'screensize'))
axes('position',[0 0 1 1])
[x,y]=ginput;
⋮
v=axis;
clf;
```

Using the provided functions **pwCEval** and **Locate**, graph in subplot(2,1,1) and subplot(2,1,2) the polygon (piecewise linear curve) and the smooth curve generated by **MySpline** respectively. You should use the arc length parameterizations and a refinement of 10 to plot the smooth curve. Here is a sample output.





(d) **(Bonus 4 marks)** Implement an efficient $O(n)$ method to solve the linear equations in your **MySpline** by modifying the provided Matlab functions **TriDiLU, LBiDiSol** and **UBidiSol**. Assume that pivoting is not necessary here. Generate another plot as in (c) but using the revised code.

Please submit the following files **to the dropbox folder on Learn**:

- `MySpline.m` - your periodic spline function
- `Q2.m` - your other Matlab code
- `plots.png` - the plots of the polygon and smooth curve

If you are doing the **bonus** question, please submit your code and plots to the **bonus dropbox** on Learn.

3. **(8 marks)** (LU factorization)

Assume that A, B, and C are $n \times n$ matrices, with A nonsingular, I is the $n \times n$ identity matrix, and b an $n \times 1$ vector. Let x be given by

$$x = A^{-1}(B + I)(A^{-1} + C)b$$

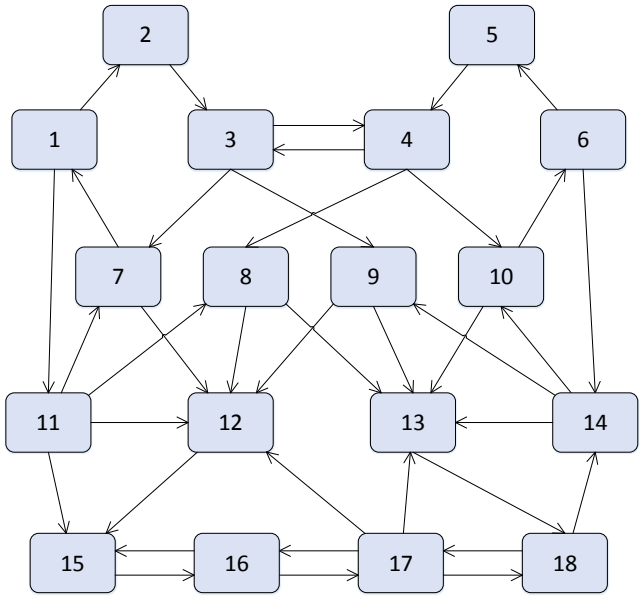
For simplicity, assume you do not need to consider permutations.

(a) Describe an efficient algorithm to compute x. Hint: Do not compute A^{-1}. Use an LU factorization of A, and compute the solution from right to left.

(b) How many flops are required by your method? Provide the leading term including the coefficient.

4. **(10 marks)** (Page Rank)

The following figure shows a the connection graph of a set of 18 web pages.



(a) Write a Matlab script to construct the connectivity matrix G for the graph shown above and compute the pagerank. To compute the pagerank, write a Matlab function $x = \mathbf{MyPageRank}(G, \alpha)$ by solving

$$(I - \alpha P)x = \frac{1 - \alpha}{R}e$$

where $P = GD$, D is the diagonal matrix with the diagonal equal to the 1./deg (deg is the vector of out-degrees of the nodes), R equal the total number of nodes, and e is the vector of all ones.

The inputs are the connectivity matrix G and the probability α of following a link to a page. The output is the pagerank.

Your matlab function **MyPageRank** needs to first compute an LU factorization with pivoting using the provided Matlab function $[L, U, p] = \mathbf{lutx}(A)$, then use provided $x = \mathbf{bslashtx}(A, b)$ to perform triangular solves.

Use Matlab command **spy** to graph the sparsity pattern (connectivity). of G.

(b) Use **MyPageRank** to compute the pagerank of the 18 web pages for $\alpha = 0.85$. Using matlab command **bar**, graph the components of the vector x as well as the final ranking. Make sure that your plots have titles and labels.

(c) Use **MyPageRank** to compute the pagerank of the 18 web pages for $\alpha = 0.5$. Generate pagerank plots again. Which page ranks increase, and which decrease?

Please submit the following files **to the dropbox folder on Learn**:

- `MyPageRank.m` - your pagerank function
- `Q4.m` - your other Matlab code
- `spy.png, pageranks_b.png, pageranks_c.png` - the required plots
- `Q4.txt` - your answer to part (c) and any other comments.