LinAlg Assignment Solutions

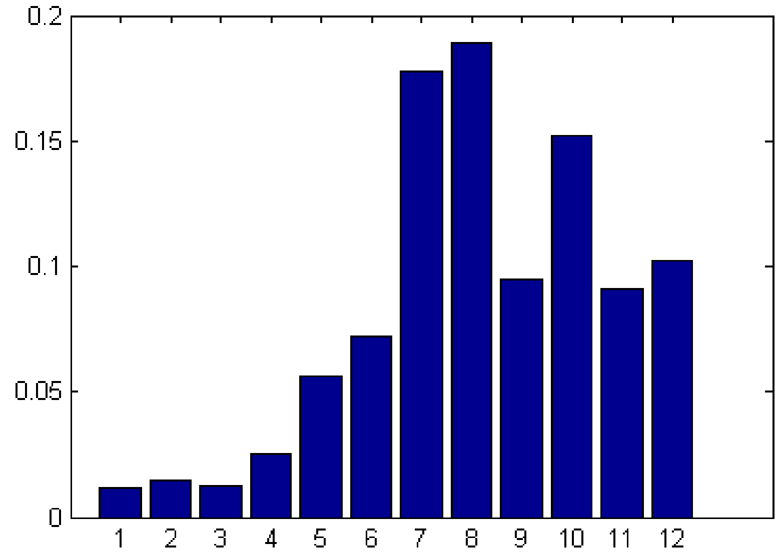
Q1 PageRank Solution

```
function [x, iters] = PageRank(G, alpha)

    outlinks = sum(G);
    d = (outlinks==0);
    D = sparse( diag(1./(outlinks+d)) ); % sparse

    P = G * D; % Scale columns by 1/deg(j)
    R = size(P,1);

    e = ones(R,1);

    x = ones(R,1) ./ R; % Initial prob. vector

    counter = 0;
    err = 1000;
    maxIter = 10000;

    while err > 1e-8 && counter<maxIter

        % Multiply by M
        % Do (d*x) before e*d and remove e*p in the last term (it equals 1)
        Mx = alpha*(P*x + e*(d*x)/R) + (1-alpha)/R*e;

        err = norm(x-Mx, inf); % Infinity norm
        % Could also use max function.

        x = Mx;

        counter = counter + 1;
    end
```

```
iters = counter;

if counter==maxIter
   disp('Failed to converge');
end
```

Q2 Trading Net Solutions

```
%% Trade network script

% Declare G as a sparse matrix
G = sparse(12,12); % sparse

% Set non-zero values of G

G(2,1) = 38;
G(3,1) = 38;
G(5,1) = 24;
G(1,2) = 6;
G(3,2) = 41;
G(6,2) = 53;
G(1,3) = 47;
G(2,3) = 29;
G(4,3) = 24;
G(3,4) = 8;
G(5,4) = 42;
G(6,4) = 50;
G(1,5) = 9;
G(4,5) = 4;
G(6,5) = 9;
G(7,5) = 39;
G(12,5) = 39;
G(2,6) = 9;
G(4,6) = 28;
G(5,6) = 19;
G(8,6) = 22;
G(12,6) = 22;
G(5,7) = 13;
G(8,7) = 17;
G(9,7) = 23;
G(10,7) = 27;
G(12,7) = 20;
G(6,8) = 15;
G(7,8) = 21;
G(9,8) = 21;
G(10,8) = 10;
G(11,8) = 18;
G(12,8) = 15;
G(7,9) = 24;
```



```
G(8,9) = 24;
G(10,9) = 32;
G(11,9) = 20;
G(7,10) = 30;
G(8,10) = 40;
G(9,10) = 5;
G(11,10) = 25;
G(8,11) = 33;
G(9,11) = 7;
G(10,11) = 60;
G(5,12) = 6;
G(6,12) = 18;
G(7,12) = 47;
G(8,12) = 29;

%% Page Rank algorithm, with alpha = 1

[p, iters] = PageRank(G, 1.);


%% Display Bar Chart

figure(1);
bar(p);
```

c) Member "H" is the most influential member of the trade network.

LU Solution

(a)

$$A = \begin{bmatrix} 1 & 1 & 3 & 0 \\ 2 & 1 & -1 & 1 \\ -1 & 2 & 3 & -1 \\ 3 & -1 & -1 & 2 \end{bmatrix} \qquad b = \begin{bmatrix} 4 \\ 1 \\ 4 \\ -3 \end{bmatrix}$$

(b)

```
Swap rows (1) and (4)

Process column 1

U =
    3.0000   -1.0000   -1.0000    2.0000
         0    1.6667   -0.3333   -0.3333
         0    1.6667    2.6667   -0.3333
         0    1.3333    3.3333   -0.6667

L =
    1.0000         0         0         0
    0.6667    1.0000         0         0
   -0.3333         0    1.0000         0
    0.3333         0         0    1.0000

Process column 2

U =
    3.0000   -1.0000   -1.0000    2.0000
         0    1.6667   -0.3333   -0.3333
         0         0    3.0000   -0.0000
         0         0    3.6000   -0.4000

L =
    1.0000         0         0         0
    0.6667    1.0000         0         0
   -0.3333    1.0000    1.0000         0
    0.3333    0.8000         0    1.0000


Swap rows (3) and (4)

Process column 3

U =
    3.0000   -1.0000   -1.0000    2.0000
         0    1.6667   -0.3333   -0.3333
         0         0    3.6000   -0.4000
         0         0         0    0.3333

L =
    1.0000         0         0         0
    0.6667    1.0000         0         0
    0.3333    0.8000    1.0000         0
   -0.3333    1.0000    0.8333    1.0000
```

```
Process column 4

U =
    3.0000   -1.0000   -1.0000    2.0000
         0    1.6667   -0.3333   -0.3333
         0         0    3.6000   -0.4000
         0         0         0    0.3333


L =
    1.0000         0         0         0
    0.6667    1.0000         0         0
    0.3333    0.8000    1.0000         0
   -0.3333    1.0000    0.8333    1.0000

P =
     0     0     0     1
     0     1     0     0
     1     0     0     0
     0     0     1     0
```

(c) Solve L z = P b using forward-substitution

$$Pb = \begin{bmatrix} -3 \\ 1 \\ 4 \\ 4 \end{bmatrix}$$

```
z =
   -3.0000
    3.0000
    2.6000
   -2.1667

Now solve U x = z using back-substitution

x =
    3.5000
    0.5000
   -0.0000
   -6.5000
```

GE Pivot Solution

(a)

```
>> A = [4 3 1 ; -8 -5.999 6 ; 4 8 11]
A =
   4.0000    3.0000    1.0000
  -8.0000   -5.9990    6.0000
   4.0000    8.0000   11.0000

>> b = [1 ; 0 ; 4]
b =
    1
    0
    4

>> x = A \ b
x =
  0.11248437109277
  0.10002500625156
  0.24998749687422
```

(b) GE without row pivoting

$$\left[\begin{array}{ccc|c} 4 & 3 & 1 & 1 \\ -8 & -5.999 & 6 & 0 \\ 4 & 8 & 11 & 4 \end{array}\right]$$

$$\downarrow$$

$$\begin{array}{c} \\ ② - \frac{-8}{4}① \\ ③ - \frac{4}{4}① \end{array} \left[\begin{array}{ccc|c} 4 & 3 & 1 & 1 \\ 0 & 0.001 & 8 & 2 \\ 0 & 5 & 10 & 3 \end{array}\right]$$

$fl(-5.999 + 2(3)) = 0.001$

$$\downarrow$$

$fl(10 - 5000 \cdot 8) = fl(10 - 40000)$

$= fl(-39990) = -39990$

$$\begin{array}{c} \\ \\ ③ - \frac{5}{0.001}② \end{array} \left[\begin{array}{ccc|c} 4 & 3 & 1 & 1 \\ 0 & 0.001 & 8 & 2 \\ 0 & 0 & -39990 & -9997 \end{array}\right]$$

$fl(3 - 5000 \cdot 2) = fl(3 - 10000)$

$= fl(9997)$

$= -9997$

$= -9997$

Back substitute

row 3 $\Rightarrow$ $x_3 = fl\left(\frac{-9997}{-39990}\right) = fl(0.2499875\ldots) = 0.2500$

row 2 $\left\{ \begin{array}{l} 0.001\, x_2 = 2 \ominus (8 \otimes 0.2500) \\ \qquad = 2 \ominus 2 = 0 \Rightarrow x_2 = 0 \end{array} \right\}$ no rounding needed here

row 1 $\left\{ \begin{array}{l} 4x_1 = 1 \ominus (\cancel{3 \otimes (0)}^{\,0} \oplus 1 \otimes 0.2500) \\ 4x_1 = 1 \ominus 0.2500 \Rightarrow x_1 = fl\left(\frac{0.7500}{4}\right) = 0.1875 \end{array} \right.$

$$\therefore \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0.1875 \\ 0 \\ 0.2500 \end{bmatrix}$$

(c) GE with row pivoting

$$\left[\begin{array}{ccc|c} 4 & 3 & 1 & 1 \\ -8 & -5.999 & 6 & 0 \\ 4 & 8 & 11 & 4 \end{array}\right]$$

$\downarrow$ swap ① & ②

$$\left[\begin{array}{ccc|c} -8 & -5.999 & 6 & 0 \\ 4 & 3 & 1 & 1 \\ 4 & 8 & 11 & 4 \end{array}\right]$$

$3 \ominus \frac{1}{2} \otimes (-5.999) = 3 \oplus fl(-2.9995)$

$= 3 \oplus (-3.000)$

$= 0$

$\downarrow$

$$\begin{array}{r} \\ ② - \frac{4}{-8}\,① \\ ③ - \frac{4}{-8}\,① \end{array} \left[\begin{array}{ccc|c} -8 & -5.999 & 6 & 0 \\ 0 & 0 & 4 & 1 \\ 0 & 5 & 14 & 4 \end{array}\right]$$

$8 \oplus \frac{1}{2} \otimes (-5.999)$

$= 8 \oplus (-3.000)$

$= 5.000$

$\downarrow$ swap ② & ③

$$\left[\begin{array}{ccc|c} -8 & -5.999 & 6 & 0 \\ 0 & 5 & 14 & 4 \\ 0 & 0 & 4 & 1 \end{array}\right]$$

Back Sub

row 3 $\Rightarrow$ $4x_3 = 1 \Rightarrow \boxed{x_3 = 0.2500}$

row 2 $\Rightarrow$ $5x_2 + 14x_3 = 4 \Rightarrow 5x_2 = 4 \ominus (14 \otimes 0.25)$

$= 4 \ominus fl(3.5)$

$= 0.5$

$\therefore \boxed{x_2 = 0.1}$

row 1 $\Rightarrow$ $-8x_1 - 5.999x_2 + 6x_3 = 0$

$-8x_1 = 5.999 \otimes (0.1) \quad 6 \otimes (0.25)$

$$-8x_1 = fl(0.5999) \ominus fl(1.5)$$

$$-8x_1 = fl(-0.9001) = -0.9001$$

$$x_1 = fl(0.1125125\ldots)$$

$$\boxed{x_1 = 0.1125}$$

(d) Summary

Exact:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0.11248437109277 \\ 0.10002500625156 \\ 0.24998749687422 \end{bmatrix}$$

Not Pivoting:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0.1875 \\ 0 \\ 0.25 \end{bmatrix}$$

With Pivoting:

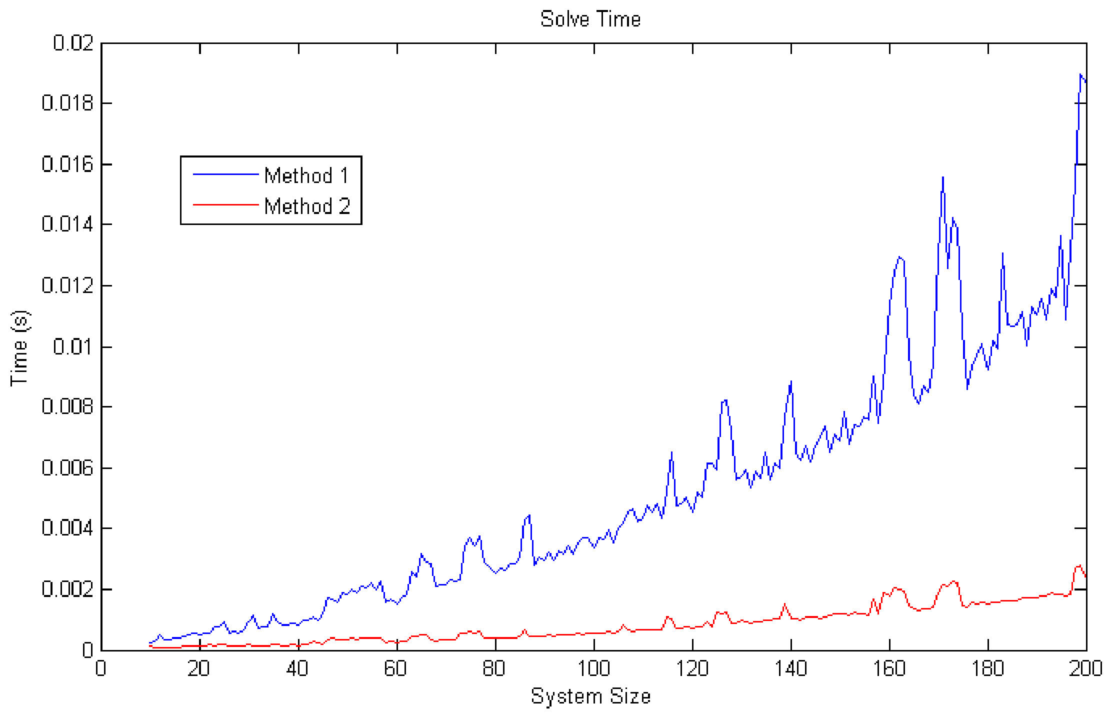
$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0.1125 \\ 0.1 \\ 0.25 \end{bmatrix}$$

Thus, the solution with pivoting is more accurate.

Solve Time Solutions

```
% solve_time.m
% Solve NxN systems using 2 methods, for various sizes of N.

Ns = 10:1:200;
M = 20;

% Store the timing results here
t_method1 = zeros(size(Ns));
t_method2 = zeros(size(Ns));

idx = 1;

% Loop over N-values (for both methods)
% Generate random A and B

for N = Ns
   % Generate random matrices
   A = rand(N);
   B = rand(N,M);
   X = zeros(size(B)); % allocate space for X

   % === Method 1 ===
   tic;
   % Solve M individual vector system using backslash
   for col = 1:M
      X(:,col) = A \  B(:,col);
   end
   %X = A \ B;  % This method is not what I want.
   t_method1(idx) = toc;  % Use tic and toc

   % === Method 2 ===
```

```
    tic;
    [L, U, P] = lu(A);  % LU factorization once
    PB = P*B;    % Apply permutation matrix
    for col = 1:M
        %{
        % This uses Matlab's triangular methods (not necessary)
        optsL.LT = true;
        z = linsolve(L, PB(:,col), optsL);
        optsU.UT = true;
        X(:,col) = linsolve(U, z, optsU);
        %}
        z = L \ PB(:,col);  % \ Solve L and U systems
        X(:,col) = U \ z;   % /
    end
    t_method2(idx) = toc;

    idx = idx + 1;
end

%% Plot
% Plot both lines on one axis
% Labeled axes, curves look correct
figure(1);
plot(Ns, t_method1, 'b');
hold on;
plot(Ns, t_method2, 'r');
legend('Method 1', 'Method 2', 'Location', 'NorthWest');
xlabel('System Size');
ylabel('Time (s)');
title('Solve Time');
hold off;
```


Solve Time
Method 1
Method 2
Time (s)
System Size
0.02
0.018
0.016
0.014
0.012
0.01
0.008
0.006
0.004
0.002
0
0
20
40
60
80
100
120
140
160
180
200