Math 426 Homework 4

Muhideen Ogunlowo

22.1 $C = \begin{bmatrix} I & A \\ -I & B \end{bmatrix}$

$$C^2 = C \times C$$

$$= \begin{bmatrix} I & A \\ -I & B \end{bmatrix} \begin{bmatrix} I & A \\ -I & B \end{bmatrix}$$

$$= \begin{bmatrix} I^2 - AI & AI + BA \\ -I^2 - BI & BA + B^2 \end{bmatrix}$$

$$C^3 = C^2 \times C$$

$$= \begin{bmatrix} I^2 - AI & AI + BA \\ -I^2 - BI & BA + B^2 \end{bmatrix} \begin{bmatrix} I & A \\ -I & B \end{bmatrix}$$

$$\begin{bmatrix} I^3 - AI^2 - AI^2 - BAI & AI^2 - A^2 I + BAI + B^2 A \\ -I^3 - BI^2 - BAI - BI^2 & -AI^2 - BAI - BI^2 - B^2 I \end{bmatrix}$$

$$= \begin{bmatrix} I(I^2 - 2AI - BA) & I(AI - A^2 + BA) + B^2 A \\ I(-I^2 - BI - BA - BI) & I(-AI - BA - BI - B^2) \end{bmatrix}$$

2.5.1b

$f(n) = n^a$

$g(n) = n^b$

for $a \le b$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{n^a}{n^b}$$

for $a \le b$

$\frac{f(n)}{g(n)}$ is bounded for all sufficiently large n for $a \le b$

So $n^a = O(n^b)$ if $a \le b$ which is True

2.5.~~1~~2d

$$n+\sqrt{n} \sim n+2\sqrt{n}$$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \frac{n+\sqrt{n}}{n+2\sqrt{n}}$$

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \frac{\cancel{n}(1+1/\sqrt{n})}{\cancel{n}(1+2/\sqrt{n})} = \lim_{n\to\infty} \frac{(1+0)}{(1+0)}$$

$$= 1$$

$= 1$

therefore

$n + \sqrt{n} \sim n + 2\sqrt{n}$

which is true

2c) $\sin(n) \sim n$

$\Rightarrow \lim_{n \to \infty} \frac{\sin(n)}{n} = 1$

but we know that

$\lim_{n \to 0} \frac{\sin n}{n} = 1$

not $\lim_{n \to \infty} \frac{\sin n}{n}$

$\sin n$ is not asymptotic to n

So c is false

2.5.4

There are $n \times n$ elements in the output matrix of the multiplication of two $n \times n$ matrices. Each of them is obtained by n multiplications (1 element from the first matrix and 1 from the second) then summing up). Since there are n products, we add $n-1$ of them to the first one. So the number of operation for one element in the output matrix is n multiplication and $n-1$ addition, meaning $2n-1$ Flops. Then for all elements we have

$n \times n \times (2n-1)$ flops

$= 2n^3 - n^2$

For large values of n, n^2 becomes insignificant, so the amount of flops needed in two $n \times n$ matrix multiplication

is $\approx 2n^3$

2.5.6

To find the exact result for (2.5.4)

use the given expressions for

$p = 1$, $p = 2$

$$\sum_{k=1}^{n} (k) = \frac{n(n+1)}{2} \qquad \sum_{k=1}^{n} (k^2) = \frac{n(n+1)(2n+1)}{6}$$

$$Q_n = \sum (k^p) = \sum (k)^2 - \sum (k^2)$$

$$Q_n = (n(n+1))^2 - \frac{(n(n+1)(2n+1)}{6}$$

$$Q_n = (n(n+1))^2 - \frac{(n(n+1)(2n+1)}{6}$$

Muhideen Ogunlowo Math426 Homework 4

Question 2.2. 3

```
u=[1;3;5;7;9;11]; %matrix u
v=[-60;-50;-40;-30;-20;-10]; %matrix v
P=transpose(u); %transpose of u
```

```
P = 1×6
     1     3     5     7     9    11
```

```
b=transpose(v); %transpose of v
```

```
b = 1×6
   -60   -50   -40   -30   -20   -10
```

```
r= P*v % inner product of u transpose and v
```

```
r =
  -910
```

```
q= b*u % inner product of v transpose and u
```

```
q =
  -910
```

```
s= u*b % outer product of u and v transpose
```

```
s = 6×6
   -60   -50   -40   -30   -20   -10
  -180  -150  -120   -90   -60   -30
  -300  -250  -200  -150  -100   -50
  -420  -350  -280  -210  -140   -70
  -540  -450  -360  -270  -180   -90
  -660  -550  -440  -330  -220  -110
```

```
z= v*P % outer product of v and u transpose
```

```
z = 6×6
   -60  -180  -300  -420  -540  -660
   -50  -150  -250  -350  -450  -550
   -40  -120  -200  -280  -360  -440
   -30   -90  -150  -210  -270  -330
   -20   -60  -100  -140  -180  -220
   -10   -30   -50   -70   -90  -110
```

Question 2.2.8 a

```
%Assume v is a 3 X 1 vector withe the values [1; 8; 13]
%Assume w is a 3 X 1 vector with the value [2; 5; 7]
v=[1; 8; 13];
w=[2; 5; 7];
g=transpose(v) %transpose of v
```

```
g = 1×3
     1     8    13
```

```
h=transpose(w) % transpose of w
```

```
h = 1×3
     2     5     7
```

```
j=g*w % inner product of v transpose and w
```

```
j = 133
```

```
k=h*v % inner product of w transpose and v
```

```
k = 133
```

As shown by the output, $v^T w$ and $w^T v$ are the same

Question 2.2.8(b)

For the counter example we find the output for vw^T and wv^T

```
f=v*h % outer product of v and w transpose
```

```
f = 6×3
  -120  -300  -420
  -100  -250  -350
   -80  -200  -280
   -60  -150  -210
   -40  -100  -140
   -20   -50   -70
```

```
l=w*g % outer product of w and v transpose
```

```
l = 3×3
     2    16    26
     5    40    65
     7    56    91
```

For the counter example, we can see that vw^T and wv^T are not the same

Question 2.3.3 a

```
A=[-2,0,0;1,-1,0; 3,2,1];
L=tril(A)
```

```
L = 3×3
    -2     0     0
     1    -1     0
     3     2     1
```

```
b=[-4;2;1];
x= forwardsub(L,b)
```

```
x = 3×1
     2
     0
    -5
```

Question 2.3.6

```
format short g
alpha= 0.1;
betas= 10.^(1:12)
```

```
betas = 1×12
          10         100        1000       10000       1e+05       1e+06 ···
```

```
baseU = eye(5)+ diag([-1 -1 -1 -1],1);
for K = 1 : length(betas)
    U = baseU;
    beta = betas(K)
    U(1,[4 5]) = [ alpha-beta, beta ]
    x_exact = ones(5, 1);
    b = [alpha;0;0;0;1];
    x=backsub(U,b)
end
```

```
beta =
    10
U = 5×5
           1          -1           0        -9.9          10
           0           1          -1           0           0
           0           0           1          -1           0
           0           0           0           1          -1
           0           0           0           0           1
x = 5×1
         0.1
           0
           0
           0
           0
beta =
   100
U = 5×5
           1          -1           0       -99.9         100
           0           1          -1           0           0
           0           0           1          -1           0
           0           0           0           1          -1
           0           0           0           0           1
x = 5×1
         0.1
           0
           0
           0
           0
beta =
        1000
U = 5×5
           1          -1           0      -999.9        1000
           0           1          -1           0           0
           0           0           1          -1           0
```

```
           0             0             0             1            -1
           0             0             0             0             1
x = 5×1
         0.1
           0
           0
           0
           0
beta =
       10000
U = 5×5
           1            -1             0       -9999.9         10000
           0             1            -1             0             0
           0             0             1            -1             0
           0             0             0             1            -1
           0             0             0             0             1
x = 5×1
         0.1
           0
           0
           0
           0
beta =
      100000
U = 5×5
           1            -1             0        -1e+05         1e+05
           0             1            -1             0             0
           0             0             1            -1             0
           0             0             0             1            -1
           0             0             0             0             1
x = 5×1
         0.1
           0
           0
           0
           0
beta =
     1000000
U = 5×5
           1            -1             0        -1e+06         1e+06
           0             1            -1             0             0
           0             0             1            -1             0
           0             0             0             1            -1
           0             0             0             0             1
x = 5×1
         0.1
           0
           0
           0
           0
beta =
    10000000
U = 5×5
           1            -1             0        -1e+07         1e+07
           0             1            -1             0             0
           0             0             1            -1             0
           0             0             0             1            -1
           0             0             0             0             1
x = 5×1
         0.1
           0
           0
           0
           0
```

```
beta =
   100000000
U = 5×5
           1           -1            0      -1e+08       1e+08
           0            1           -1           0           0
           0            0            1          -1           0
           0            0            0           1          -1
           0            0            0           0           1
x = 5×1
         0.1
           0
           0
           0
           0
beta =
       1e+09
U = 5×5
           1           -1            0      -1e+09       1e+09
           0            1           -1           0           0
           0            0            1          -1           0
           0            0            0           1          -1
           0            0            0           0           1
x = 5×1
         0.1
           0
           0
           0
           0
beta =
       1e+10
U = 5×5
           1           -1            0      -1e+10       1e+10
           0            1           -1           0           0
           0            0            1          -1           0
           0            0            0           1          -1
           0            0            0           0           1
x = 5×1
         0.1
           0
           0
           0
           0
beta =
       1e+11
U = 5×5
           1           -1            0      -1e+11       1e+11
           0            1           -1           0           0
           0            0            1          -1           0
           0            0            0           1          -1
           0            0            0           0           1
x = 5×1
         0.1
           0
           0
           0
           0
beta =
       1e+12
U = 5×5
           1           -1            0      -1e+12       1e+12
           0            1           -1           0           0
           0            0            1          -1           0
           0            0            0           1          -1
           0            0            0           0           1
```

```
x = 5×1
         0.1
           0
           0
           0
           0
```

Question 2.4.2

```
%Assume T(3,-1) = P
%Assume T (-3,1)= Q
%Assume R(pi/5)= R
P=[1 0 0;0 1 0;3 -1 1] ;
Q=[1 0 0;0 1 0;-3 1 1];
R=[cos(pi/5) sin(pi/5) 0; -sin(pi/5) cos(pi/5) 0; 0 0 1];
A= P*R*Q;
z=[2; 2; 1];
b=A*z
```

```
b = 3×1
      2.7936
     0.44246
      4.9383
```

```
[L,U]= lufact(A)
```

```
L = 3×3
           1            0            0
    -0.72654            1            0
    0.018339       1.5724            1
U = 3×3
     0.80902      0.58779            0
           0       1.2361            0
           0            0            1
```

```
A-L*U;
x=backsub(U,z)
```

```
x = 3×1
      2.4721
       1.618
           1
```

```
x-z
```

```
ans = 3×1
     0.47214
    -0.38197
           0
```

Question 2.4.7 a

```
%Question 2.4.7 a
A= [2 0 4 3; -4 5 -7 -10; 1 15 2 -4.5; -2 0 2 -13];
[L,U] = lufacto(A)
```

```
L = 4×4
         1          0          0          0
        -2          1          0          0
       0.5          3          1          0
        -1          0         -2          1
U = 4×4
     2     0     4     3
     0     5     1    -4
     0     0    -3     6
     0     0     0     2
```

Question 2.4.7 b

```
L(4,3) = A(4,3) / A(3,3)
```

```
L = 4×4
         1          0          0          0
        -2          1          0          0
       0.5          3          1          0
        -1          0          1          1
```

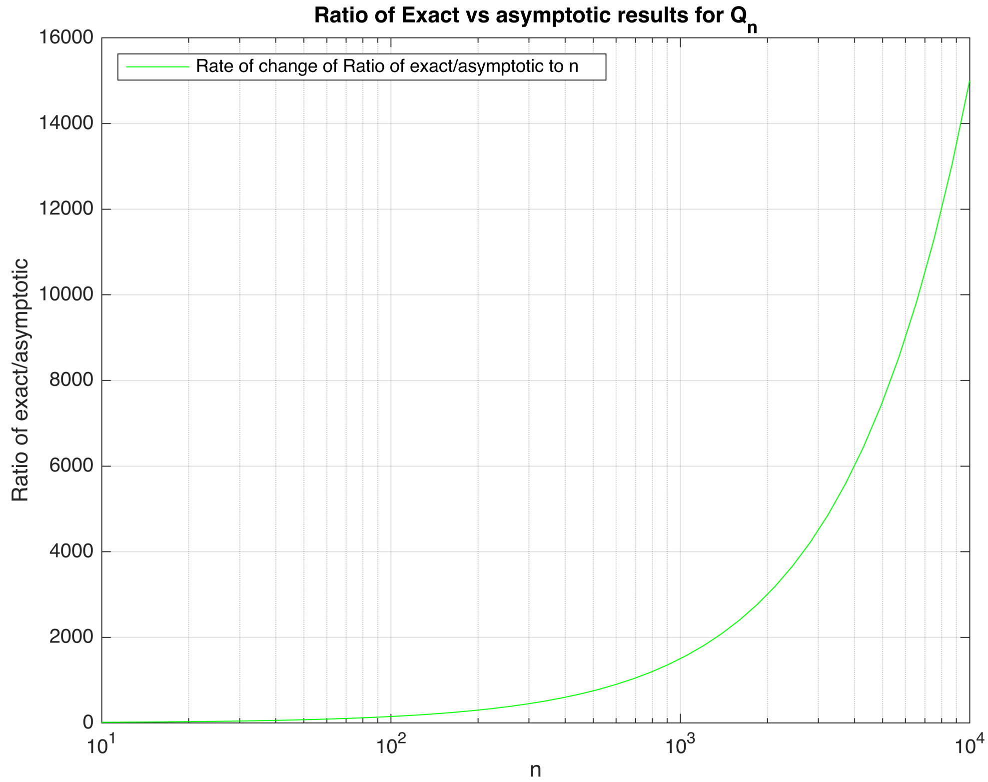
```
A(4,3) = A(4,3) - L(4,3) * A(3,3)
```

```
A = 4×4
         2          0          4          3
        -4          5         -7        -10
         1         15          2       -4.5
        -2          0          0        -13
```

Question 2.5.6 c

```
%Question 2.5.6c
%Define range of values using logspace
n=logspace(1,4);
x_result=(n.^2.*(n+1).^2)-(n.*(n+1).*(2.*n + 1)/6);
asymp_result=(2*n.^3)/3;
%ratio of exact result vs asymptotic result
rat_io = x_result./asymp_result;

%Create a semi-log plot
semilogx(n,rat_io,'g');
%label axes
xlabel('n')
ylabel('Ratio of exact/asymptotic')
legend('Rate of change of Ratio of exact/asymptotic to n' ,'Location',
'Northwest')
title('Ratio of Exact vs asymptotic results for Q_n')
grid on
```



Question 2.3.3 a, 2.3.6, 2.4.2 and 2.4.7

```
function x = forwardsub(L,b) %Question 2.3.3a
%FORWARDSUB Solve a lower triangular linear system
%Input:
% L = lower triangular matrix (n by n)
% b = right-hand side vector (n by 1)
% Output:
%Solution of Lx=b (n by 1 vector)
n =  length(L);
x= zeros(n,1);
for i = 1:3
    x(i) =( b(i) - L(i, 1:i-1)*x(1:i-1) ) / L(i,i);
end
end

%Question 2.3.6
function x = backsub(U,b)
%FORWARDSUB Solve a lower triangular linear system
%Input:
% U = Upper triangular matrix (n by n)
% b = right-hand side vector (n by 1)
% Output:
%Solution of Ux=b (n by 1 vector)
n =  length(U);
```

```
x= zeros(n,1);
for i = 1:3
    x(i) =( b(i) - U(i, 1:i-1)*x(1:i-1) ) / U(i,i);
end
end

%Question 2.4.2
function [L,U] = lufact(A)
% LUFACT LU factorization
% Input:
% A square matrix
% Output:
% (L,U) unit lower triangular and upper triangular such that LU=A
n=length(A);
L= eye(n); %ones on diagonal
%Gaussian elimination
for j= 1:n-1
    for i = j+1:n
        L(i,j) = A(i,j) / A(j,j);
        A(i,j:n) = A(i,j:n) - L(i,j)*A(j,j:n);
    end
end
U=triu(A);
end

%Question 2.4.7
function [L, U] = lufacto(A)
    % LUFACT LU factorization
    % Input:
    % A square matrix
    % Output:
    % (L, U) unit lower triangular and upper triangular such that LU = A
    n = length(A);
    L = eye(n); % ones on diagonal
    % Gaussian elimination
    for j = 1:n-1
        L(j+1:n, j) = A(j+1:n, j) / A(j, j);
        A(j+1:n, j:n) = A(j+1:n, j:n) - L(j+1:n, j) * A(j, j:n);
    end
U = triu(A);
end
```