Numerical Computing HW4

Yunfan Gao

March 5, 2021

Problem 1

a)

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

$x_3 = 2$

$x_2 = 1$

$2x_2 - 2 - 2 = -2$

$x_1 = 1$

b)

$$\left[\begin{array}{ccc|c} 1 & -2 & -1 & 2 \\ 0 & 3 & 1 & 4 \\ 2 & -1 & 1 & 2 \end{array}\right] \rightarrow \left[\begin{array}{ccc|c} 1 & 2 & -1 & 2 \\ 0 & 3 & 1 & 4 \\ 0 & -5 & 3 & -2 \end{array}\right] \rightarrow \left[\begin{array}{ccc|c} 1 & 2 & -1 & 2 \\ 0 & 3 & 1 & 4 \\ 0 & 0 & \frac{14}{3} & \frac{14}{3} \end{array}\right] \qquad \begin{array}{l} x_1 = 1 \\ x_2 = 1 \\ x_3 = 1 \end{array}$$

Problem 2

$$\begin{bmatrix} 3 & 1 & 2 \\ 6 & 3 & 4 \\ 3 & 1 & 5 \end{bmatrix} \rightarrow \begin{bmatrix} 3 & 1 & 2 \\ 0 & 1 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

$$LU = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 & 1 & 2 \\ 0 & 1 & 0 \\ 0 & 0 & 3 \end{bmatrix} = A$$

b)

$$\begin{bmatrix} 4 & 2 & 0 \\ 4 & 4 & 2 \\ 2 & 2 & 3 \end{bmatrix} \rightarrow \begin{bmatrix} 4 & 2 & 0 \\ 0 & 2 & 2 \\ 0 & 1 & 3 \end{bmatrix} \rightarrow \begin{bmatrix} 4 & 2 & 0 \\ 0 & 2 & 2 \\ 0 & 0 & 2 \end{bmatrix}$$

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

Problem 3

$$\begin{bmatrix} 3 & 1 & 2 \\ 6 & 3 & 4 \\ 3 & 1 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 3 \end{bmatrix}$$

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} 3 & 1 & 2 \\ 0 & 1 & 0 \\ 0 & 0 & 3 \end{bmatrix}}_{U} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 3 \end{bmatrix}$$

$$Lc = b \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 3 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 3 \end{bmatrix}$$

$c_1 = 0$ $c_2 = 1$ $c_3 = 3$

$$\begin{bmatrix} 3 & 1 & 2 \\ 0 & 1 & 0 \\ 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 3 \end{bmatrix}$$

$x_1 = -1$
$x_2 = 1$
$x_3 = 1$

Problem 4

U x=c
$Kn^2 = 1000 \cdot 500^2$
$Ax = b$
$\frac{2kn^3}{3} = \frac{2 \cdot 1 \cdot 5000^3}{3}$
$\frac{\frac{2kn^3}{3}}{kn^2} = \frac{\frac{2 \cdot 1 \cdot 5000^3}{3}}{1000 \cdot 500^2} = 333.33$

Problem 5

```
clear all;close all;clc
%set up the problem%
n = 100;
a = zeros(n,n);
for m = 1:n
    for k = 1:n
        if m>k
            a(m,k) = 0;
        else
            a(m,k) = m + 2*k;
```

```
        end
    end
end
x = ones(n,1);
b = a*x;

solvedx = Backsub(n,a,b)
%back substitution%
function sx=Backsub(n,a,b)
for i = n:-1:1
    s = b(i);
    for j = i+1:n
        s = s - a(i,j)*sx(j);
    end
    sx(i) = s/a(i,i);
end
end
```

With output solvedx $= (1, \cdots, 1)^T$

Problem 6

The naive Gaussian function

```

%2.1 (a)
%{
a = [2 -3;5 -6];
b = [2;8];
n = 2;
x = naiveGaussian(n,a,b);
%}

%2.1(b)

%{
a = [1 2; 2 3];
b = [-1;1];
n = 2;
x = naiveGaussian(n,a,b);
%}

%problem 6 b)
%a) n =2 b) n = 5 c) n = 10
%{
n = 2;
```

```
a = zeros(n,n);
for m = 1:n
    for k = 1:n
        a(m,k) = 1/(m+k-1);
    end
end
b = ones(n,1);
x = naiveGaussian(n,a,b);
%}
function x=naiveGaussian(n,a,b)

%using vectorization to solve naive Gaussian
L = zeros(n,n);
for j = 1 : n-1
    if abs(a(j,j))<eps
        error('zero pivot encountered');
    end
    L(j+1:n,j) = a(j+1:n,j)/a(j,j);
    a(j+1:n,j+1:n) = a(j+1:n,j+1:n) - L(j+1:n,j)*a(j,j+1:n); % making first column zero
    b(j+1:n) = b(j+1:n )- L(j+1:n,j)*b(j);
end

%back substitution
for i = n:-1:1
    s = b(i);
    for j = i+1:n
        s = s - a(i,j)*x(j);
    end
    x(i) = s/a(i,i);
end

end
```

Input and output for 2.1 a)

```
>> a = [2 -3;5 -6];
b = [2;8];
n = 2;
x = naiveGaussian(n,a,b)

x =

     4     2
```

Input and output for 2.1 b)

```
>> a = [1 2; 2 3];
b = [-1;1];
n = 2;
x = naiveGaussian(n,a,b)

x =

     5    -3
```

Input and output for n = 2

```
>> n = 2;
a = zeros(n,n);
for m = 1:n
    for k = 1:n
        a(m,k)  = 1/(m+k-1);
    end
end
b = ones(n,1);
x = naiveGaussian(n,a,b)

x =

  -2.000000000000001   6.000000000000002
```

Input and output for n = 5

```
>> n = 5;
a = zeros(n,n);
for m = 1:n
    for k = 1:n
        a(m,k)  = 1/(m+k-1);
    end
end
b = ones(n,1);
x = naiveGaussian(n,a,b)

x =

   1.0e+03 *

   0.004999999999994  -0.119999999999913   0.629999999999665  -1.119999999999537   0.629999999999788
```

Input and output for n = 10

```
>> n = 10;
a = zeros(n,n);
for m = 1:n
    for k = 1:n
        a(m,k) = 1/(m+k-1);
    end
end
b = ones(n,1);
x = naiveGaussian(n,a,b)

x =

   1.0e+06 *

  Columns 1 through 6

  -0.000009997364824   0.000989771860948  -0.023755133779705   0.240195714290297  -1.261048597183709   3.783198501115927

  Columns 7 through 10

  -6.725765489566751   7.000357237862889  -3.937735417590592   0.923673408496481
```