Mock test 1 - 24th April, Ichim Ștefan - 234/1

1

Analizarea complexitatii metodei lui Gauss pt n = 4

A, b - matrici cu valori intre [100, 400]

nr operatii -> no_operations = cate operatii aritmetice se realizeaza

timp -> time-milliseconds = cat dureaza toate operatiile aritmetice

```
clear;
close all;
clc;

n = 4;
fprintf("n = %d\n", n);
```

```
n = 4
```

```
A = randi([100, 400], n)
```

```
A = 4×4
   254   160   337   133
   366   222   195   141
   276   325   260   304
   146   348   127   249
```

```
b = randi([100, 400], n, 1)
```

```
b = 4×1
   157
   248
   144
   116
```

```
[new_A, new_b, x, no_operations, time_milliseconds] = scaled_partial_gaussian_factorization_con

%new_A
%new_b
A \ b
```

```
ans = 4×1
    0.5804
    0.5745
    0.0288
   -0.6921
```

```
x
```

```
x = 4×1
    0.5804
    0.5745
    0.0288
   -0.6921
```

```
no_operations
```

```
no_operations = 64
```

```
time_milliseconds
```

```
time_milliseconds = 3
```

2

a)

```
n = 10;
A = diag(3 * ones(1,n)) + diag(-1 * ones(1, n-1), 1) + diag(-1 * ones(1, n-1), -1);
for i = 1:n
    if A(i, n - i + 1) == 0
        A(i, n - i + 1) = 1/2;
    end
end
A
```

```
A = 10×10
    3.0000   -1.0000         0         0         0         0         0         0···
   -1.0000    3.0000   -1.0000         0         0         0         0         0
         0   -1.0000    3.0000   -1.0000         0         0         0    0.5000
         0         0   -1.0000    3.0000   -1.0000         0    0.5000         0
         0         0         0   -1.0000    3.0000   -1.0000         0         0
         0         0         0         0   -1.0000    3.0000   -1.0000         0
         0         0         0    0.5000         0   -1.0000    3.0000   -1.0000
         0         0    0.5000         0         0         0   -1.0000    3.0000
         0    0.5000         0         0         0         0         0   -1.0000
    0.5000         0         0         0         0         0         0         0
```

```
b = zeros(n,1);
b(1) = 2.5;
b(n) = 2.5;

b(n / 2) = 1;
b(n / 2 + 1) = 1;

for i = 2:(n/2)-1
    b(i) = 1.5;
    b(n - i + 1) = 1.5;
end
b
```

```
b = 10×1
    2.5000
    1.5000
    1.5000
    1.5000
    1.0000
    1.0000
    1.5000
    1.5000
    1.5000
    2.5000
```

LUP factorization

```
[L, U, P, ~] = LUPFactorization(A, b, n);
L
```

```
L = 10×10
     1     0     0     0     0     0     0     0     0     0
   NaN     1     0     0     0     0     0     0     0     0
   NaN   NaN     1     0     0     0     0     0     0     0
   NaN   NaN   NaN     1     0     0     0     0     0     0
   Inf   NaN   NaN   NaN     1     0     0     0     0     0
   NaN   NaN   NaN   NaN   NaN     1     0     0     0     0
   NaN   NaN   NaN   NaN   NaN   NaN     1     0     0     0
   Inf   NaN   NaN   NaN   NaN   NaN   NaN     1     0     0
   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN     1     0
  -Inf   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN     1
```

```
U
```

```
U = 10×10
         0   -1.0000    3.0000   -1.0000         0         0         0    0.5000···
         0       NaN       NaN       NaN       NaN       NaN       NaN       NaN
         0         0       NaN       NaN       NaN       NaN       NaN       NaN
         0         0         0       NaN       NaN       NaN       NaN       NaN
         0         0         0         0       NaN       NaN       NaN       NaN
         0         0         0         0         0       NaN       NaN       NaN
         0         0         0         0         0         0       NaN       NaN
         0         0         0         0         0         0         0       NaN
         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0
```

```
P
```

```
P = 10×10
     0     0     1     0     0     0     0     0     0     0
     0     0     0     0     0     1     0     0     0     0
     0     0     0     0     0     0     0     0     1     0
     0     0     0     0     0     0     0     1     0     0
     0     0     0     0     0     0     0     0     0     1
     0     0     0     0     1     0     0     0     0     0
     0     0     0     0     0     0     1     0     0     0
     1     0     0     0     0     0     0     0     0     0
     0     0     0     1     0     0     0     0     0     0
     0     1     0     0     0     0     0     0     0     0
```

Cholesky factorization

```
[L, x] = CholeskyFactorization(A, b, n);
L
```

```
L = 10×10
    1.7321         0         0         0         0         0         0         0···
   -0.5774    1.6330         0         0         0         0         0         0
         0   -0.6124    1.6202         0         0         0         0         0
         0         0   -0.6172    1.6183         0         0         0         0
         0         0         0   -0.6179    1.6181         0         0         0
         0         0         0         0   -0.6180    1.6180         0         0
         0         0         0    0.3090    0.1180   -0.5730    1.6007         0
         0         0    0.3086    0.1177    0.0449    0.0172   -0.6446    1.5726
         0    0.3062    0.1157    0.0441    0.0169    0.0064   -0.0075   -0.6655
    0.2887    0.1021    0.0386    0.0147    0.0056    0.0021   -0.0025   -0.0099
```

```
L * L'
```

```
ans = 10×10
    3.0000   -1.0000         0         0         0         0         0         0···
   -1.0000    3.0000   -1.0000         0         0         0         0         0
         0   -1.0000    3.0000   -1.0000         0         0         0    0.5000
         0         0   -1.0000    3.0000   -1.0000         0    0.5000    0.0000
         0         0         0   -1.0000    3.0000   -1.0000    0.0000    0.0000
         0         0         0         0   -1.0000    3.0000   -1.0000   -0.0000
         0         0         0    0.5000    0.0000   -1.0000    3.0000   -1.0000
         0         0    0.5000    0.0000    0.0000   -0.0000   -1.0000    3.0000
         0    0.5000    0.0000   -0.0000   -0.0000   -0.0000   -0.0000   -1.0000
    0.5000   -0.0000   -0.0000    0.0000    0.0000    0.0000    0.0000   -0.0000
```

b)

```
n = 100;
A = diag(3 * ones(1,n)) + diag(-1 * ones(1, n-1), 1) + diag(-1 * ones(1, n-1), -1);
for i = 1:n
    if A(i, n - i + 1) == 0
        A(i, n - i + 1) = 1/2;
    end
end
A;
b = zeros(n,1);
b(1) = 2.5;
b(n) = 2.5;

b(n / 2) = 1;
b(n / 2 + 1) = 1;

for i = 2:(n/2)-1
    b(i) = 1.5;
    b(n - i + 1) = 1.5;
end
b;
```

LUP factorization

```
[L, U, P, x] = LUPFactorization(A, b, n);
A \ b
```

```
ans = 100×1
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
      ⋮
```

```
x
```

```
x = 100×1
```

```
    NaN
    NaN
    NaN
    NaN
    NaN
    NaN
    NaN
    NaN
    NaN
    NaN
      ⋮
```

Cholesky factorization

```
[L, x] = CholeskyFactorization(A, b, n);
A \ b
```

```
ans = 100×1
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
      ⋮
```

```
x
```

```
x = 100×1
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
      ⋮
```