P346 (Computational Physics Lab)
Assignment 4
Name: Ratul Das
Roll No.: 1911128

[1]:
```
import mylibrary
```

Question 1

[2]:
```
A = []
b = []

mat_a = open("a.txt", "r+")
mat_ab = open("a_b.txt", "r+")
#calling matrix from txt file

for line in mat_a:
    x = line.split()
    y = []
    for i in range(len(x)):
        y.append(float(x[i]))
    A.append(y)

for line in mat_ab:
    elem = line.split()
    for i in range(len(elem)):
        b.append(float(elem[i]))

crout_result = mylibrary.solveLU(A,b,1)
doolittle_result = mylibrary.solveLU(A,b,2)
```

[3]:
```
print("Result of LU decomposition using Crout's method:")
print("x_1 = " + str(crout_result[0]))
print("x_2 = " + str(crout_result[1]))
print("x_3 = " + str(crout_result[2]))
print("x_4 = " + str(crout_result[3]))
```

```
Result of LU decomposition using Crout's method:
x_1 = 1.0
x_2 = -1.0
x_3 = 1.0
x_4 = 2.0
```

[4]:
```
L_1,U_1 = mylibrary.crout(A)
print("L matrix generated using Crout's method:")
mylibrary.displayMatrix(L_1)
print("U matrix generated using Crout's method:")
```

```
mylibrary.displayMatrix(U_1)
```

```
L matrix generated using Crout's method:
[1.0, 0.0, 0.0, 0.0]
[0.0, 1.0, 0.0, 0.0]
[1.0, 2.0, 2.0, 0.0]
[2.0, 1.0, 3.0, -3.0]
U matrix generated using Crout's method:
[1, 0.0, 1.0, 2.0]
[0.0, 1, -2.0, 0.0]
[0.0, 0.0, 1, -1.0]
[0.0, 0.0, 0.0, 1]
```

[5]:
```
print("Result of LU decomposition using Doolittle's method:")
print("x_1 = " + str(doolittle_result[0]))
print("x_2 = " + str(doolittle_result[1]))
print("x_3 = " + str(doolittle_result[2]))
print("x_4 = " + str(doolittle_result[3]))
```

```
Result of LU decomposition using Doolittle's method:
x_1 = 1.0
x_2 = -1.0
x_3 = 1.0
x_4 = 2.0
```

[6]:
```
L_2,U_2 = mylibrary.doolittle(A)
print("L matrix generated using Doolittle's method:")
mylibrary.displayMatrix(L_2)
print("U matrix generated using Doolittle's method:")
mylibrary.displayMatrix(U_2)
```

```
L matrix generated using Doolittle's method:
[1, 0.0, 0.0, 0.0]
[0.0, 1, 0.0, 0.0]
[1.0, 2.0, 1, 0.0]
[2.0, 1.0, 1.5, 1]
U matrix generated using Doolittle's method:
[1.0, 0.0, 1.0, 2.0]
[0.0, 1.0, -2.0, 0.0]
[0.0, 0.0, 2.0, -2.0]
[0.0, 0.0, 0.0, -3.0]
```

Question 2

[7]:
```
B = []

mat_b = open("b.txt", "r+")
#calling matrix from txt file
```

```
for line in mat_b:
    x = line.split()
    y = []
    for i in range(len(x)):
        y.append(float(x[i]))
    B.append(y)

print("Inverse of the matrix is: ")
B_inv = mylibrary.LU_inverse(B)
mylibrary.displayMatrix(B_inv)
```

```
Inverse of the matrix is:
[-0.25000000000000006, 1.6666666666666667, -1.8333333333333333, 0.3333333333333333]
[0.08333333333333337, -0.6666666666666667, 0.8333333333333333, 0.0]
[0.16666666666666666, -0.3333333333333333, -0.3333333333333333, 0.0]
[-0.08333333333333333, 0.6666666666666666, 0.16666666666666666, 0.0]
```

[8]:
```
#we verify the obtained inverse
P = mylibrary.productMatrix(B_inv,B)
mylibrary.displayMatrix(P)
```

```
[1.0, 0.009999999999999787, 0.010000000000000009, 0.009999999999999787]
[0.0, 1.01, 0.010000000000000009, 0.009999999999999787]
[0.0, 0.010000000000000009, 1.01, 0.009999999999999898]
[0.0, 0.010000000000000009, 0.010000000000000009, 1.0100000000000002]
```

Question 3

[9]:
```
C = []
b = []

mat_c = open("c.txt", "r+")
mat_cb = open("c_b.txt", "r+")
#calling matrices from txt files

for line in mat_c:
    x = line.split()
    y = []
    for i in range(len(x)):
        y.append(float(x[i]))
    C.append(y)

for line in mat_cb:
    elem = line.split()
    for i in range(len(elem)):
        b.append(float(elem[i]))
```

```python
L = mylibrary.choleskydecomp(C)
L_T = mylibrary.transpose(L)
cholesky_result = mylibrary.solveCholesky(L, L_T, b)

print("Solution using Cholesky decomposition is: ")
print("x_1 = " + str(cholesky_result[0]))
print("x_2 = " + str(cholesky_result[1]))
print("x_3 = " + str(cholesky_result[2]))
print("x_4 = " + str(cholesky_result[3]))
```

```
Solution using Cholesky decomposition is:
x_1 = 0.1
x_2 = 0.2
x_3 = 0.3
x_4 = 0.4
```