

# CSCE 410 502

## Homework 8

Keegan Smith

November 11, 2024

### Problem 1

1. LU:

```
1 def LU(A):
2     lower = [[] * len(A)
3     for i in range(0, len(A)):
4         lower[i] = [0] * len(A)
5     for row in range(1, len(A)):
6         for col in range(0, row):
7             coefficient = -A[row][col] / A[col][col]
8             add_rows(A, col, row, -A[row][col] / A[col][col])
9             lower[row][col] = -coefficient
10            lower[col][col] = 1
11    lower[len(A)- 1][len(A)-1] = 1
12    return lower
```

please note that the above function mutates A to be the upper triangular matrix. Only the lower triangular matrix is returned.

2. Cholesky:

```
1 def cholesky(A):
2     lower = [[] * len(A)
3     for i in range(0, len(A)):
4         lower[i] = [0] * len(A)
5     for j in range(0, len(A)):
6         my_sum = 0
7         for k in range(0, j):
8             my_sum += lower[j][k] * lower[j][k]
9         lower[j][j] = (A[j][j] - my_sum) ** .5
10        for i in range(j + 1, len(A)):
11            my_sum = 0
12            for k in range(0, j):
13                my_sum += lower[i][k] * lower[j][k]
14            lower[i][j] = (1 / lower[j][j] * (A[i][j] - my_sum))
15    return lower
```

3. solving:

```
1 def add_rows(A, row_a, row_b, factor): # adds row_a * factor to b
2     for i in range(0, len(A)):
3         A[row_b][i] += factor * A[row_a][i]
4 def display(A):
5     for i in range(0, len(A)):
6         result = ""
7         for j in range(0, len(A)):
```

```
8         result += str(A[i][j]) + "\t"
9     print(result)
10    print()
11    def transpose(A):
12        for i in range(0, len(A)):
13            for j in range(0, i):
14                temp = A[i][j]
15                A[i][j] = A[j][i]
16                A[j][i] = temp
17    def LU(A):
18        lower = [[] * len(A)
19        for i in range(0, len(A)):
20            lower[i] = [0] * len(A)
21        for row in range(1, len(A)):
22            for col in range(0, row):
23                coefficient = -A[row][col] / A[col][col]
24                add_rows(A, col, row, -A[row][col] / A[col][col])
25                lower[row][col] = -coefficient
26                lower[col][col] = 1
27        lower[len(A)-1][len(A)-1] = 1
28        return lower
29    def back_sub(A, b):
30        i = len(A) - 1
31        coefficients = [0] * len(b)
32        while(i >= 0):
33            row_sum = 0
34            j = len(A) - 1
35            while(j > i):
36                row_sum += A[i][j] * coefficients[j]
37                j -= 1
38            b[i] -= row_sum
39            coefficients[i] = b[i] / A[i][i]
40            i -= 1
41        return coefficients
42    def forward_sub(A, b):
43        coefficients = [0] * len(A)
44        for i in range(0, len(A)):
45            sum_row = 0
46            for j in range(0, i):
47                sum_row += A[i][j] * coefficients[j]
48            result = b[i] - sum_row
49            coefficients[i] = result / A[i][i]
50        return coefficients
51
52    def cholesky(A):
53        lower = [[] * len(A)
54        for i in range(0, len(A)):
55            lower[i] = [0] * len(A)
56        for j in range(0, len(A)):
57            my_sum = 0
58            for k in range(0, j):
59                my_sum += lower[j][k] * lower[j][k]
60            lower[j][j] = (A[j][j] - my_sum) ** .5
61            for i in range(j + 1, len(A)):
62                my_sum = 0
63                for k in range(0, j):
64                    my_sum += lower[i][k] * lower[j][k]
65                lower[i][j] = (1 / lower[j][j] * (A[i][j] - my_sum))
66        return lower
67    def test_lu(A, b):
68        lower = LU(A)
69        display(A)
```

```
70     y = forward_sub(lower, b)
71     result = back_sub(A, y)
72     print("LU result: ", result)
73 def test_cholesky(A, b):
74     lower = cholesky(A)
75     y = forward_sub(lower, b)
76     transpose(lower)
77     result = back_sub(lower, y)
78     print("Cholesky result: ", result)
79 def main():
80     test = [
81         [5, -5, 0, 0],
82         [-5, 7, -2, 0],
83         [0, -2, 20, -18],
84         [0, 0, -18, 19]
85     ]
86     test_b = [5, -7, 20, -17]
87     test_lu(test, test_b)
88     test = [
89         [5, -5, 0, 0],
90         [-5, 7, -2, 0],
91         [0, -2, 20, -18],
92         [0, 0, -18, 19]
93     ]
94     test_b = [5, -7, 20, -17]
95     test_cholesky(test, test_b)
96 if __name__ == "__main__":
97     main()
```