CSCE 410 502
Homework 8

Keegan Smith

November 11, 2024

Problem 1

1. LU:

```
def LU(A):
    lower = [[]] * len(A)
    for i in range(0, len(A)):
        lower[i] = [0] * len(A)
    for row in range(1, len(A)):
        for col in range(0, row):
            coefficient = -A[row][col] / A[col][col]
            add_rows(A, col, row, -A[row][col] / A[col][col])
            lower[row][col] = -coefficient
            lower[col][col]=  1
    lower[len(A)- 1][len(A)-1] = 1
    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:

```
def cholesky(A):
    lower = [[]] * len(A)
    for i in range(0, len(A)):
        lower[i] = [0] * len(A)
    for j in range(0, len(A)):
        my_sum = 0
        for k in range(0, j):
            my_sum += lower[j][k] * lower[j][k]
        lower[j][j] = (A[j][j] - my_sum) ** .5
        for i in range(j + 1, len(A)):
            my_sum = 0
            for k in range(0, j):
                my_sum += lower[i][k] * lower[j][k]
            lower[i][j] = (1 / lower[j][j] * (A[i][j] - my_sum))
    return lower
```

3. solving:

```
def add_rows(A, row_a, row_b, factor): # adds row_a * factor to b
    for i in range(0, len(A)):
        A[row_b][i] += factor * A[row_a][i]
def display(A):
    for i in range(0, len(A)):
        result = ""
        for j in range(0, len(A)):
            result += str(A[i][j]) + "\t"
        print(result)
    print()
def transpose(A):
    for i in range(0, len(A)):
        for j in range(0, i):
            temp = A[i][j]
            A[i][j] = A[j][i]
            A[j][i] = temp
def LU(A):
    lower = [[]] * len(A)
    for i in range(0, len(A)):
        lower[i] = [0] * len(A)
    for row in range(1, len(A)):
        for col in range(0, row):
            coefficient = -A[row][col] / A[col][col]
            add_rows(A, col, row, -A[row][col] / A[col][col])
            lower[row][col] = -coefficient
            lower[col][col]=  1
    lower[len(A)- 1][len(A)-1] = 1
    return lower
def back_sub(A, b):
    i = len(A) -1
    coefficients = [0] * len(b)
    while(i >= 0):
        row_sum = 0
        j = len(A) - 1
        while(j > i):
            row_sum += A[i][j] * coefficients[j]
            j -= 1
        b[i] -= row_sum
        coefficients[i] = b[i] / A[i][i]
        i -= 1
    return coefficients
def forward_sub(A, b):
    coefficients = [0] * len(A)
    for i in range(0, len(A)):
        sum_row = 0
        for j in range(0, i ):
            sum_row += A[i][j] * coefficients[j]
        result = b[i] - sum_row
        coefficients[i] = result / A[i][i]
    return coefficients

def cholesky(A):
    lower = [[]] * len(A)
    for i in range(0, len(A)):
        lower[i] = [0] * len(A)
    for j in range(0, len(A)):
        my_sum = 0
        for k in range(0, j):
            my_sum += lower[j][k] * lower[j][k]
        lower[j][j] = (A[j][j] - my_sum) ** .5
        for i in range(j + 1, len(A)):
            my_sum = 0
            for k in range(0, j):
                my_sum += lower[i][k] * lower[j][k]
            lower[i][j] = (1 / lower[j][j] * (A[i][j] - my_sum))
    return lower
def test_lu(A, b):
    lower = LU(A)
    display(A)
    y = forward_sub(lower, b)
    result = back_sub(A, y)
    print("LU result: ", result)
def test_cholesky(A, b):
    lower = cholesky(A)
    y = forward_sub(lower, b)
    transpose(lower)
    result = back_sub(lower, y)
    print("Cholesky result: ", result)
def main():
    test = [
        [5, -5, 0, 0],
        [-5, 7, -2, 0],
        [0, -2, 20, -18],
        [0, 0, -18, 19]
    ]
    test_b = [5, -7, 20, -17]
    test_lu(test, test_b)
    test = [
        [5, -5, 0, 0],
        [-5, 7, -2, 0],
        [0, -2, 20, -18],
        [0, 0, -18, 19]
    ]
    test_b = [5, -7, 20, -17]
    test_cholesky(test, test_b)
if __name__ == "__main__":
    main()
```