Homework 3 Writeup

Christian Diangco
AMATH 301 B

Problem 1

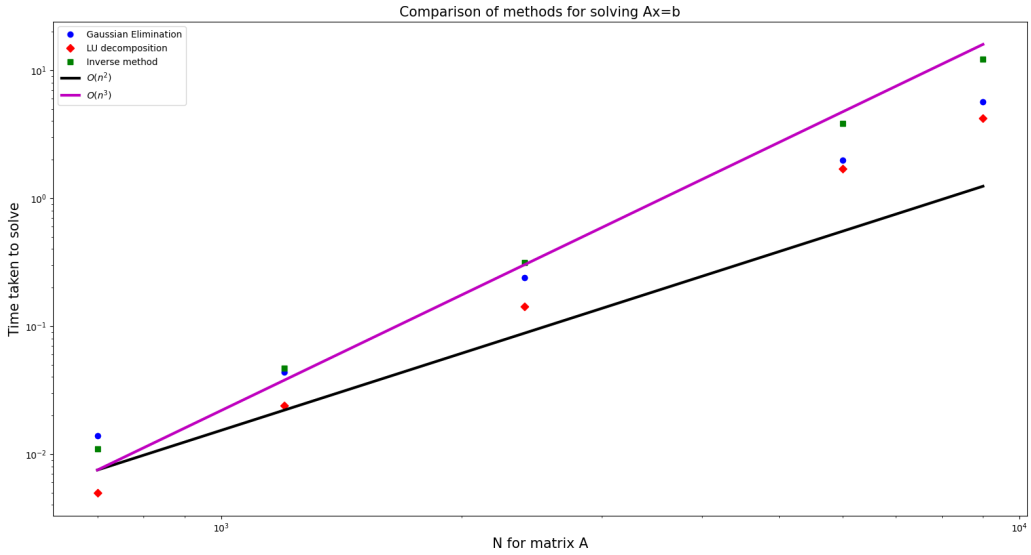
	Time for 100 vectors	||r||
Backslash time	12.982279539108276	3.2951747373141015e-12
LU time	0.4807145595550537	3.0655888679991147e-12
inv time	0.5116353034973145	3.2803350523239547e-11

From the table, the **inverse method** has the most residual error. If I had to get the right answer quickly, as a matter of life or death, I would use the **LU method**. It has the smallest residual error and is the fastest of the three.

```
import numpy as np
import scipy.linalg
import matplotlib.pyplot as plt
import time

N = 2000
offDiag = np.ones(N-1)
A = -1*np.diag(np.ones(N)) + 4*np.diag(offDiag,1) + 4*np.diag(offDiag,-1)

start = time.time()
r1 = 0
for i in range(100):
    b = np.random.rand(N,1)
    x = scipy.linalg.solve(A,b)
    r1 += np.linalg.norm(A @ x - b)
stop1 = time.time() - start

start = time.time()
r2 = 0
LU, piv = scipy.linalg.lu_factor(A)
for i in range(100):
    b = np.random.rand(N,1)
    x = scipy.linalg.lu_solve((LU,piv), b)
    r2 += np.linalg.norm(A @ x - b)
stop2 = time.time() - start

start = time.time()
r3 = 0
A_inv = scipy.linalg.inv(A)
for i in range(100):
    b = np.random.rand(N,1)
    x = A_inv @ b
    r3 += np.linalg.norm(A @ x - b)
stop3 = time.time() - start
```

Problem 2

Of the three methods, LU decomposition is the fastest. The inverse method starts out faster than Gaussian Elimination, but as N increases, Gaussian Elimination becomes faster with an increasing difference in time between the two.




```
n = np.array([700., 1200., 2400., 6000., 9000.])
ge_solve = np.empty([5])
lu_solve = np.empty([5])
inv_solve = np.empty([5])
v = 0
for i in range(5):
    A = np.random.rand(int(n[i]),int(n[i]))
    b = np.random.rand(int(n[i]),1)

    start = time.time()
    scipy.linalg.solve(A,b)
    stop = time.time() - start
    ge_solve[i] = stop

    start = time.time()
    LU, piv = scipy.linalg.lu_factor(A)
    scipy.linalg.lu_solve((LU,piv), b)
    stop = time.time() - start
    lu_solve[i] = stop

    start = time.time()
    scipy.linalg.inv(A) @ b
    stop = time.time() - start
    inv_solve[i] = stop

fig,ax = plt.subplots()
ax.loglog(n, ge_solve, "bo", label="Gaussian Elimination")
ax.loglog(n, lu_solve, "rD", label="LU decomposition")
ax.loglog(n, inv_solve, "gs", label="Inverse method")
ax.loglog(n, 10**-2.125 * n**2 / n[0]**2, "k", label="$O(n^2)$", linewidth=3)
ax.loglog(n, 10**-2.125 * n**3 / n[0]**3, "m", label="$O(n^3)$", linewidth=3)
ax.set_title("Comparison of methods for solving Ax=b", fontsize=15)
ax.set_xlabel("N for matrix A", fontsize=15)
ax.set_ylabel("Time taken to solve", fontsize=15)
ax.legend()
plt.show()
```