Linear Algebra and its Applications

Assignment 4

Name - Anjan Mondal

Roll Number- MDS202208

Import all prerequisites

In [1]:
```
from helper_functions import *
import time
import scipy as sp
import pandas as pd
```

Initialization

Initialization of all data structures for storing the following:

- Running time of both solvers
- Solutions L, U, P, x and the chosen A and b for both solvers

In [2]:
```
n_list=[1,10,100,500,1000,5000,10000]
scipy_LU_rt={} #Running times by scipy's LU factorization algorithm
my_LU_rt={}  # Running times by my implementation LU factorization algorithm

scipy_subst_rt={} #Running times by scipy's Substitution algorithm
my_subst_rt={}   # Running times by my implementation of Substitution algorithm


scipy_solutions={}
my_solutions={}
```

Execution

Execute the solvers and record the running times

In [3]:
```
for n in n_list:

    A=np.random.rand(n,n)
    b=np.random.rand(n)

    start_time = time.time()
    L,U,P=LU_factorize(A)
    my_LU_rt[n]=time.time()-start_time


    start_time=time.time()
    x=substitution(L,U,P,b)
    my_subst_rt[n]=time.time()-start_time
    my_solutions[n]=(L,U,P,x,A,b)


    start_time=time.time()
    p, l, u = sp.linalg.lu(A) # The decomposition is A=plu
    scipy_LU_rt[n]=time.time()-start_time



    lu, piv = sp.linalg.lu_factor(A)
    start_time=time.time()
    x2 = sp.linalg.lu_solve((lu, piv), b)
    scipy_subst_rt[n]=time.time()-start_time


    scipy_solutions[n]=(l,u,p,x2,A,b)
```

Calculate Norms

Calculate the norms of $PA - LU$ and $Ax_0 - b$ using both solvers.

In [5]:
```
my_PA_LU_norms={}
scipy_PA_LU_norms={}
my_Ax_b_norms={}
scipy_Ax_b_norms={}


for n in n_list:
    L,U,P,x,A,b=my_solutions[n]
    my_PA_LU_norms[n]=np.linalg.norm(np.matmul(P,A)-np.matmul(L,U))
    my_Ax_b_norms[n]=np.linalg.norm(np.matmul(A,x)-b)


    L,U,P,x,A,b=scipy_solutions[n]
    scipy_PA_LU_norms[n]=np.linalg.norm(A-np.matmul(P,np.matmul(L,U)))
    scipy_Ax_b_norms[n]=np.linalg.norm(np.matmul(A,x)-b)
```

Time Taken

This table gives us the times taken (in seconds) for LU Factorization and Substitution for both the solvers

The columns labels are set as n for an $n \times n$ matrix

In [6]:
```
all_times={}
col_names=["LU","LU SciPy", "Substitution","Substitution SciPy "]
for n in n_list:
    all_times[n]=[my_LU_rt[n],scipy_LU_rt[n], my_subst_rt[n],scipy_subst_rt[n]]
time_taken=pd.DataFrame(all_times, index=col_names)
```

In [7]:
```
time_taken
```

Out[7]:

	1	10	100	500	1000	5000	10000
LU	0.000000	0.0	0.016196	0.580633	2.726164	128.682333	916.863778
LU SciPy	0.125434	0.0	0.000000	0.015593	0.031233	1.318268	23.225082
Substitution	0.000000	0.0	0.000000	0.000000	0.000000	0.059418	0.397035
Substitution SciPy	0.000000	0.0	0.000000	0.000000	0.000000	0.000000	0.073293

Norms Table

This table gives us the the matrix norms of $PA - LU$ and $Ax_0 - b$ for both solvers.

The columns labels are set as n for an $n \times n$matrix

In [8]:
```
matrix_norms={}
col_names=["PA-LU","PA-LU SciPy", "Ax-b","Ax-b SciPy"]
for n in n_list:
    matrix_norms[n]=[my_PA_LU_norms[n],scipy_PA_LU_norms[n], my_Ax_b_norms[n],scipy
norm_table=pd.DataFrame(matrix_norms, index=col_names)
```

In [9]:
```
norm_table
```

Out[9]:

	1	10	100	500	1000	5000	10000
PA-LU	0.0	5.383795e-16	2.861281e-14	4.760205e-13	1.686134e-12	3.257530e-11	1.191446e-10
PA-LU SciPy	0.0	5.783892e-16	2.667814e-14	4.249832e-13	1.434099e-12	2.458436e-11	8.317461e-11
Ax-b	0.0	3.007864e-15	1.206219e-13	4.624882e-13	4.761288e-12	5.234042e-10	2.340650e-08
Ax-b SciPy	0.0	2.641304e-15	1.550325e-13	7.755543e-13	7.645528e-12	9.339654e-10	6.158627e-08