Programming Assignment - 4

Name:

In [3]:
```python
# Import required packages
import numpy as np
```

In [41]:
```python
# You can modify this code to answer the following
'''
Jacobi's iteration method for solving the system of equations Ax=b.
p0 is the initialization for the iteration.
'''
def jacobi(A, b, p0, tol, maxIter=100):
    n=len(A)
    p = p0

    for k in range(maxIter):
        p_old = p.copy() # In python assignment is not the same as copy

        # Update every component of iterant p
        for i in range(n):
            sumi = b[i];
            for j in range(n):
                if i==j: # Diagonal elements are not included in Jacobi
                    continue;
                sumi = sumi - A[i,j] * p_old[j]
            p[i] = sumi/A[i,i]

        rel_error = np.linalg.norm(p-p_old)/n
        # print("Relative error in iteration", k+1,":",rel_error)
        if rel_error<tol:
            print("TOLERANCE MET BEFORE MAX-ITERATION")
            break
    return p;
```

In [24]:
```python
# Example System
A = np.array([[10, -1, 2, 0],
              [-1, 11, -1, 3],
              [2, -1, 10, -1],
              [0, 3, -1, 8]],dtype=float)
b = np.array([6, 25, -11, 15],dtype=float)
```

In [40]:
```python
## What will happen if the followign code runs
#x = jacobi(A,b, np.array([0,0,0,0]),0.00001, 100)

x = jacobi(A,b, np.array([0,0,0,0],dtype=float),0.0000001, 100)
print("The solution is: ",x)
```

```
TOLERANCE MET BEFORE MAX-ITERATION
The solution is:  [ 1.00000003  1.99999996 -0.99999997  0.99999995]
```

- (A) Implement the Gauss-Siedel Iteration in Python. Solve the following system by using this method. Exact answer is (1,2,-1,1). Stopping criteria could be a relative $error < 0.00001$. $$

$$\begin{pmatrix} 10 & -1 & 2 & 0 \\ -1 & 11 & -1 & 3 \\ 2 & -1 & 10 & -1 \\ 0 & 3 & -1 & 8 \end{pmatrix}$$

\begin{pmatrix} x_1\x_2\x_3\x_4

\end{pmatrix}

$$\begin{pmatrix} 6 \\ 25 \\ -11 \\ 15 \end{pmatrix}$$

$$

In [14]:

```python
# Your code here
```

- (B) Implement Successive Over-relaxation in Python and solve the above problem again with $\omega = 1.5$.

In [ ]:

```python
# Your code here
```