Experiment:5
Solving systems using Iterative methods

1. ## Vector and Matrix Norms

   (a) Find the $||.||_1, ||.||_2, ||.||_3$ and $||.||_\infty$ norms of the following vectors

        i. $v_1 = (1, 2, 3)$

        ii. $v_2 = (-4, 5, 6)$

   (b) Find the $||.||_1, ||.||_2$ and $||.||_\infty$ norm of the Matrix

$$A = \begin{bmatrix} \dfrac{53}{360} & -\dfrac{13}{90} & \dfrac{23}{360} \\ -\dfrac{11}{180} & \dfrac{1}{45} & \dfrac{19}{180} \\ -\dfrac{7}{360} & \dfrac{17}{90} & -\dfrac{37}{360} \end{bmatrix}$$

2. ## Iterative Method(s) for solving system of equations

   Creates code in MATLAB that can be called as a function(s) (name it *Gauss_Jacobi.m* and *Gauss_Seidel.m*) that gives the computational solution of system(s) for given tolerance level, and the no. of iterations. Solve the following system(s) of equations using both the above function(s) and check which method converges faster.

   (a) Using Jacobi method and Guass Seidel method find approximating solution with initial guess as $x^{(0)} = (0,0,0)^T$ and tolerance 0.00001.

$$5x - y + z = 10$$
$$2x + 4y = 12$$
$$x + y + 5z = -1$$

   (b) Using Jacobi method and Guass Seidel method find approximations solution starting with $x^{(0)} = (0,0,0,0)^T$ with tolerance 0.001

$$29x + 2y + z = 7$$
$$2x + 6y + z = 5$$
$$x + y + \frac{z}{5} = 4$$

Algorithm for Gauss-Jacobi method

// Input: a $m \times n$ matrix A, $n \times 1$ column vector b, tol, N(Number of iterations), $x^{(0)}$(initial guess)

// Output: Jacobi method won't be applied or $n \times 1$ column vector along with number of iterations consisting computational solution of the system

(a) Create a function which take the matrix A, b, tol, N, $x^{(0)}$ as input

(b) Evaluate size of matrix A and store it in variables m, n

(c) Define zero matrices D, L & U of order $m \times n$ % for creating A=D+L+U, D is diagonal matrix, L is strictly lower triangular matrix and U is strictly upper triangular matrix

(d) `If` a given matrix A is a square matrix

     i. `for` $i < j$ store the values of matrix A to the matrix U;
        `end for`

     ii. `for` $i > j$ store the values of matrix A to the matrix L;
        `end for`

     iii. `for` $i = j$ store the values of matrix A to the matrix D;
        `end for`

(e) Set $s = 1$;

(f) `While` $s <= N$ % for running iterations

     i. $x^{(k+1)} = D^{-1}(b - (L + U)x^{(k)})$

     ii. Define $||x^{(k+1)} - x^{(k)}||_\infty$

     iii. `if` $||x^{(k+1)} - x^{(k)}||_\infty$ is less than tol

     iv. `Print` the solutions and no. of iterations

     v. `return`;
        `end if`

     vi. Set $s = s + 1$

     vii. Repeat the above steps (i)-(vi)

    `end while`

(g) `Print` "Method fails to converge to given tol in given number of iterations"

(h) `else Print` "Jacobi method will not work for given system"
    `end if`

Algorithm for Gauss Seidel method

// Input: a $m \times n$ matrix A, $n \times 1$ column vector b, tol, N(Number of iterations), $x^{(0)}$(initial guess)

// Output: Jacobi method won't be applied or $n \times 1$ column vector along with number of iterations consisting computational solution of the system

(a) Create a function which take the matrix A, b, *tol*, N, $x^{(0)}$ as input

(b) Evaluate size of matrix A and store it in variables m, n

(c) Define zero matrices D, L & U of order $m \times n$ % for creating A=D+L+U, D is diagonal matrix, L is strictly lower triangular matrix and U is strictly upper triangular matrix

(d) `If` a given matrix A is a square matrix

    i. `for` $i < j$ store the values of matrix A to the matrix U;
       `end for`

    ii. `for` $i > j$ store the values of matrix A to the matrix L;
       `end for`

    iii. `for` $i = j$ store the values of matrix A to the matrix D;
       `end for`

(e) Set $s = 1$;

(f) `While` $s <= N$ % for running iterations

    i. $x^{(k+1)} = (D + L)^{-1}(b - Ux^{(k)})$

    ii. Define $||x^{(k+1)} - x^{(k)}||_\infty$

    iii. `if` $||x^{(k+1)} - x^{(k)}||_\infty$ is less than *tol*

    iv. `Print` the solutions and no. of iterations

    v. `return`;
       `end if`

    vi. Set $s = s + 1$

    vii. Repeat the above steps (i)-(vi)

    `end while`

(g) `Print` "Method fails to converge to given *tol* in given number of iterations"

(h) `else Print` "Gauss Seidel method will not work for given system"
    `end if`