Experiment -7

Code :-

clc;clear;clf;

disp("Priyanshu | 13592 | 23025558022")

// User inputs for matrix dimensions and values

m = input("Enter the number of rows (equations): ");

n = input("Enter the number of columns (variables): ");

// Initialize matrix A

A = zeros(m, n);

for i = 1:m

for j = 1:n

A(i, j) = input("Enter A(" + string(i) + "," + string(j) + "): ");

end

end

// Initialize vector b

b = zeros(m, 1);

for l = 1:m

b(l) = input("Enter b(" + string(l) + "): ");

end

// Check for diagonal dominance

isDiagonallyDominant = 1; // Assume it is dominant (true)

for i = 1:m

sumOffDiagonal = 0;

for j = 1:n

if i ~= j then

sumOffDiagonal = sumOffDiagonal + abs(A(i, j));

end

end

if abs(A(i, i)) < sumOffDiagonal then

isDiagonallyDominant = 0; // Found a row that is not dominant (false)

break;

end

end

// Display message if not diagonally dominant

if isDiagonallyDominant == 0 then

disp("The matrix A is not diagonally dominant. The method may not converge.");

return; // Exit the program if not dominant

end

// Define parameters

max\_iter = 100; // Maximum number of iterations

tolerance = .005

; // Convergence tolerance

// Initialize solution vectors

x = zeros(n, 1); // Initial guess for x

x\_new = zeros(n, 1); // For storing updated x values

iter = 0; // Initialize iteration counter

// Gauss-Jacobi Iteration with while loop

while iter < max\_iter

iter = iter + 1; // Increment the iteration counter

for i = 1:m

// Calculate the sum for the ith equation

sum1 = 0;

for j = 1:n

if i ~= j then

sum1 = sum1 + A(i, j) \* x(j);

end

end

// Update x\_new(i)

x\_new(i) = (b(i) - sum1) / A(i, i);

end

// Check for convergence

if norm(x\_new - x, "inf") < tolerance then

disp("Converged after " + string(iter) + " iterations");// convert this code to gauss seidal method

break;

end

// Update x with x\_new

x = x\_new;

end

// Display the result

disp("Total iterations: " + string(iter));

if iter >= max\_iter then

disp("Maximum iterations reached without convergence.");

else

disp("Solution:");

disp(x);

end





