Table of Contents

Assignment

Gauss Seidel Iterator with rounding

```
% Name         : Mohamed Mafaz
% Roll Number : AM25M009
% Department   : Applied Mechanics

clc;
clear;

A = [1/1   1/2  1/3 1/4;     % A Matrix
     1/2   1/3  1/4 1/5;
     1/3   1/4  1/5 1/6;
     1/4   1/5  1/6 1/7];

B = [25/12; 77/60; 57/60; 319/420];   % B Matrix

X = zeros(1, length(B));      % Initial Guess
tolerance = 1e-12;
```

Part 1 (Preprocessing)

```
function [] = Diag_dom(A)
    diag_dom = 0;
    for j = 1:length(A)
        sum = 0;
        for i = 1:length(A)
            if i ~= j
                sum = sum + abs(A(j, i)); % abs value of sum of non diagonal
elements
            end
        end
        if sum > abs(A(j, j))
            diag_dom = diag_dom + 1;
        end
    end

    if diag_dom > 0
        fprintf('Matrix is NOT Diagonally dominant :(\n\n')
    else
        fprintf('Matrix is Diagonally dominant :)\n\n')
    end
```

```
end

Diag_dom(A)
```

Matrix is NOT Diagonally dominant :(

Part 2 (Gauss-Seidel Function with rounding)

```
function [loops_taken, relative_error, X] = Gauss_Sadel(A, B, X, tolerance,
sig)
    relative_error = Inf;
    loops_taken = 0;

    while (relative_error > tolerance)
        X_old = X;
        for j = 1:length(A)
            sum = 0;
            for i = 1:length(A)
                if i ~= j
                    % multiply then round
                    temp = A(j,i) * X(i);
                    temp = round(temp, sig, 'significant');
                    sum = round(sum + temp, sig, 'significant');  % Rounding
of significant digits
                end
            end
            % numerator and division with rounding
            num = round(B(j) - sum, sig, 'significant');
            X(j) = round(num / A(j,j), sig, 'significant');
        end

        % relative error (use rounding too)
        relative_error = max(abs(X - X_old) ./ (X + 1e-9));
        loops_taken = loops_taken + 1;

        % Printing progress
        % fprintf("Loop: %d   |", loops_taken)
        % for i = 1:length(X)
        %     fprintf("   X_%d: %.6f   |   ", i, X(i))
        % end
        % fprintf("\n")
    end
    if relative_error < tolerance
        fprintf("Loop: %d   |", loops_taken)
        for i = 1:length(X)
            fprintf("   X_%d: %.6f   |   ", i, X(i))
        end
    end
end
```

Loop: 37 | X_1: 0.963000 | X_2: 1.080000 | X_3: 1.090000 | X_4: 0.847000 | Loops: 37 | Relative error: 0.000000e+00

```
=== Gauss-Seidel with 6 significant digits ===
Loop: 2568    |    X_1: 1.000510    |       X_2: 0.994419    |       X_3:
1.013210    |       X_4: 0.991508    |    Loops: 2568 | Relative error:
0.000000e+00
```

Part 3 (Post Processing)

```
% 3 sig
fprintf("\n=== Gauss-Seidel with 3 significant digits ===\n")
[loops3, rel3, X3] = Gauss_Sadel(A, B, X, tolerance, 3);
fprintf("Loops: %d | Relative error: %e\n", loops3, rel3)

% 6 sig
fprintf("\n=== Gauss-Seidel with 6 significant digits ===\n")
[loops6, rel6, X6] = Gauss_Sadel(A, B, X, tolerance, 6);
fprintf("Loops: %d | Relative error: %e\n", loops6, rel6)
```

```
=== Gauss-Seidel with 3 significant digits ===
```

Published with MATLAB® R2025a