Problem 3

```
clc;
clear;
close all;

fprintf("Problem 3\n\n");

epsilon = 1e-7;

A = [4 -1 0 -1 0 0; ...
    -1 4 -1 0 -1 0; ...
    0 -1 4 -1 0 -1; ...
    -1 0 -1 4 -1 0; ...
    0 -1 0 -1 4 -1; ...
    0 0 -1 0 -1 4];

b = [2; 1; 2; 2; 1; 2];

D = diag(diag(A));
L_plus_U = A-D;

% Perform Gauss-Jacobi iteration
GJ_iters = 0;
x0 = ones(6, 1);
abs_err = Inf;

while abs_err >= epsilon
    x1 = (D\eye(size(D)))*(b - L_plus_U*x0);
    GJ_iters = GJ_iters + 1;
    abs_err = norm(x0-x1, 2);
    x0 = x1;
end

GJ_abs_err = abs_err;

fprintf("Gauss-Jacobi iterations required to get absolute error
 between successive iterates below %1.0e: %d\n", ...
        epsilon, GJ_iters);

% Perform Gauss-Seidel iteration
L        = tril(A);
U_strict = triu(A, 1);
GS_iters = 0;
x0 = ones(6, 1);
abs_err = Inf;

while abs_err >= epsilon
    x1 = (L\eye(size(L)))*(b - U_strict*x0);
    GS_iters = GS_iters + 1;
    abs_err = norm(x0-x1, 2);
    x0 = x1;
```

```
end

GS_abs_err = abs_err;

fprintf("Gauss-Seidel iterations required to get absolute error
 between successive iterates below %1.0e: %d\n", ...
        epsilon, GS_iters);

% Perform SOR iteration
L_strict = tril(A, -1);
omega = 1.6735;
% If we pick omega = 1.2, SOR out-performs the GJ and GS iterations,
 i.e.,
% it takes 14 iterations for the absolute error of successive iterates
 (in
% the 2-norm) to fall below 10^(-7)
% omega = 1.2;
SOR_iters = 0;
x0 = ones(6, 1);
abs_err = Inf;

while abs_err >= epsilon
    x1 = ((D+omega*L_strict)\eye(size(D)))*(omega*b - (omega*U_strict
 + (omega-1)*D)*x0);
    SOR_iters = SOR_iters + 1;
    abs_err = norm(x0-x1, 2);
    x0 = x1;
end

SOR_abs_err = abs_err;

fprintf("SOR iterations required to get absolute error between
 successive iterates below %1.0e: %d\n", ...
        epsilon, SOR_iters);

% Find error estimates for the three iteration methods
spectral_radius_B_GJ = max(abs(eig(-inv(D)*L_plus_U)));
spectral_radius_B_GS = max(abs(eig(inv(L)*U_strict)));
spectral_radius_B_SOR = max(abs(eig(inv(D
+omega*L_strict)*(omega*U_strict + (omega-1)*D))));

fprintf("\n");
fprintf("Spectral radius of B matrix for Gauss-Jacobi iteration: %0.5f
\n", spectral_radius_B_GJ);
fprintf("Spectral radius of B matrix for Gauss-Seidel iteration: %0.5f
\n", spectral_radius_B_GS);
fprintf("Spectral radius of B matrix for SOR iteration: %0.5f\n",
 spectral_radius_B_SOR);
fprintf("\n");

fprintf("\n");
fprintf("Error bound for Gauss-Jacobi iteration: %0.16f\n", ...
        (spectral_radius_B_GJ/(1-spectral_radius_B_GJ))*GJ_abs_err);
fprintf("Error bound for Gauss-Seidel iteration: %0.16f\n", ...
```

```
         (spectral_radius_B_GS/(1-spectral_radius_B_GS)*GS_abs_err));
fprintf("Error bound for SOR iteration: %0.16f\n", ...
        (spectral_radius_B_SOR/(1-
spectral_radius_B_SOR))*SOR_abs_err);
fprintf("\n");
```

```
Problem 3

Gauss-Jacobi iterations required to get absolute error between
 successive iterates below 1e-07: 40
Gauss-Seidel iterations required to get absolute error between
 successive iterates below 1e-07: 22
SOR iterations required to get absolute error between successive
 iterates below 1e-07: 49

Spectral radius of B matrix for Gauss-Jacobi iteration: 0.68301
Spectral radius of B matrix for Gauss-Seidel iteration: 0.48058
Spectral radius of B matrix for SOR iteration: 0.72573


Error bound for Gauss-Jacobi iteration: 0.0000001669161786
Error bound for Gauss-Seidel iteration: 0.0000000847837315
Error bound for SOR iteration: 0.0000002139901023
```

Problem 4

```
clear;
fprintf("\nProblem 4\n");

omegas = 0.8:0.1:1.3;

% Compute spectral radii of the iteration matrix given in problem 4
 for
% various omega values
fprintf("\n");
for i=1:length(omegas)
    iter_matrix = [(1-omegas(i)), (1/2)*omegas(i); ...
                   (1/2)*omegas(i)*(1-omegas(i)),
 (1/4)*(omegas(i))^(2)+(1-omegas(i))];
    rho_iter_matrix = max(abs(eig(iter_matrix)));
    str = sprintf("Spectral radius for iteration matrix (omega =
 %0.2f): %0.16f", ...
            omegas(i), abs(rho_iter_matrix));
    disp(str);
end
fprintf("\n");
```

```
Problem 4

Spectral radius for iteration matrix (omega = 0.80):
 0.4759591794226542
```

```
Spectral radius for iteration matrix (omega = 0.90):
 0.3758969653329253
Spectral radius for iteration matrix (omega = 1.00):
 0.2500000000000000
Spectral radius for iteration matrix (omega = 1.10):
 0.1000000000000001
Spectral radius for iteration matrix (omega = 1.20):
 0.2000000000000000
Spectral radius for iteration matrix (omega = 1.30):
 0.3000000000000000
```

Published with MATLAB® R2021a