MATH 3340 - Scientific Computing Assignment 3

Libao Jin

March 10, 2021

Please note that the deadline will be enforced as per the previous homework. Remember that you are allowed to work in teams of two on this assignment. You are encouraged to prepare your work in LaTeX; a template will be provided to help you put it all together. If you choose to submit a hard copy, you may submit only one copy for a team, indicating the names of both contributors. Online submission is encouraged, however, in that case both members of a team should submit the PDF file containing their work and showing both their names.

Instruction

1. Go to `https://www.overleaf.com` and sign in (required).
2. Open template, click *Menu* (up left corner), then *Copy Project*.
3. Go to `LaTeX/meta.tex` (the file `meta.tex` under the folder `LaTeX`) to change the section and your name, e.g.,
   - change title to `\title{MATH 3340-01 Scientific Computing Homework 3}`
   - change author to `\author{Albert Einstein \& Carl F. Gauss}`
4. For Problem 1, 2, 3, you need to write function/script files, store results to output files, and save graphs to figure files. Here are suggested names for function files, script files, output files, and figure files:

Problem	Function File	Script File	Output File	Figure File
1	`jacobi.m`	`hw3_p1.m`	`hw3_p1.txt`	`hw3_p1.pdf`
2	`gaussSeidel.m`	`hw3_p2.m`	`hw3_p2.txt`	`hw3_p2.pdf`
3		`hw3_p3.m`	`hw3_p3.txt`	`hw3_p3.pdf`

   Once finished, you need to upload these files to the folder `src` on Overleaf. If you have different filenames, please update the filenames in `\lstinputlisting{../src/your_script_name.m}` accordingly. You can code in the provided files in hw3.zip, and use the MATLAB script `save_results.m` to generate the output files and store the graphs to `.pdf` files automatically (the script filenames should be exactly same as listed above).
5. Recompile, download, and submit the generated PDF.
6. You may find LaTeX.Mathematical.Symbols.pdf and the second part of Lab 01 Slides and Lab 02 Slides helpful.

Problem 1. Solve, using a MATLAB code, the following system:

$$\begin{cases} 5x - y + z + w = 9 \\ x + 7y + 2z + 2w = 3 \\ 2x + y + 5z + w = 7 \\ x - y + z + 4w = 6 \end{cases}$$

Use the Jacobi method with with a tolerance of 10^{-5} for the norm of the residual. Arrange your results in a table of the form so that you can see how x, y, z and w change with each iteration. You

Table 1: caption

iteration	x	y	z	w
0	0	0	0	0
1	1	2	3	4
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

can create such a table using `disp` in a loop; you may also use variants of `printf` (in C Language), i.e., `fprintf` or `sprintf` if you are familiar with these functions, but do not use the MATLAB `table` command as it does not do quite what is expected here. Also, as obvious from the table above, start with the zero guess. Moreover, plot the norm of the residual versus the iteration number; use a logarithmic scale on the vertical axis (the residual axis). Turn in your code and the output as described. The code can be organized as:

(a) a function file that implements Jacobi's method; and
(b) a script that calls the function with with the appropriate inputs and processes the results.

Your plot should have a plot title, axes labels and a legend. Use the `help plot` command and the class notes to investigate the various options available when plotting.

Solution.

- Function file `jacobi.m`

```
function [x, iters, res] = jacobi(A, b, x0, maxIter, tol)
%JACOBI: Jacobi Method
% Syntax: [x, iters] = jacobi(A, b, x0, maxIter, tol)
% Inputs:
%   A       = coefficient matrix (matrix)
%   b       = right-hand side vector (column vector)
%   x0      = solution to the linear system (column vector)
%   maxIter = maximum of number of iterations (scalar)
%   tol     = tolerance (scalar)
% Outputs:
%   x     = solution to the linear system (matrix, i-th column is the solution in the i-th
    iteration)
%   iters = number of iterations performed (vector)
%   res   = norm of residuals (vector)
%
% Author: Libao Jin
```

```
% Date: 10/04/2020

D = diag(diag(A));
R = A - D;
x = [x0];
iters = [0];
res = [norm(b - A * x0)];
i = 0;
while i <= maxIter && res(end) > tol
    x0 = D \ (b - R * x0);
    res = [res norm(b - A * x0)];
    x = [x x0];
    i = i + 1;
    iters = [iters i];
end

end
```

- Script file `hw3_p1.m`

```
% MATH 3340, Fall 2020
% Homework 3, Problem 1
% Author: Libao Jin
% Date: 10/04/2020

clc; clear; figure(1);
% change default text interpreter to LaTeX
set(groot, 'defaultTextInterpreter','latex');
set(groot, 'defaultAxesTickLabelInterpreter','latex');
set(groot, 'defaultLegendInterpreter','latex')

A = [5 -1 1 1; 1 7 2 2; 2 1 5 1; 1 -1 1 4];
b = [9; 3; 7; 6];
x0 = [0; 0; 0; 0];
maxIter = 100;
tol = 1e-5;
[x, iters, res] = jacobi(A, b, x0, maxIter, tol);

fprintf('%10s %10s %10s %10s %10s\n', 'iterations', 'x', 'y', 'z', 'w');
for i = 1:length(x)
    fprintf('%10d %10.4f %10.4f %10.4f %10.4f\n', i - 1, x(1, i), x(2, i), x(3, i), x(4, i))
        ;
end

semilogy(iters, res);
xlabel('Iteration Number');
ylabel('Norm of Residual');
title('Jacobi method');
grid on;
grid minor;
```

- Output file `hw3_p1.txt`

```
iterations          x          y          z          w
         0     0.0000     0.0000     0.0000     0.0000
```

```
          1    1.8000     0.4286    1.4000    1.5000
          2    1.3057    -0.6571    0.2943    0.8071
          3    1.4483    -0.0727    0.8477    0.9357
          4    1.4288    -0.2879    0.6481    0.9078
          5    1.4312    -0.2201    0.7045    0.9088
          6    1.4333    -0.2368    0.6898    0.9110
          7    1.4325    -0.2336    0.6918    0.9100
          8    1.4329    -0.2337    0.6917    0.9105
          9    1.4328    -0.2339    0.6915    0.9104
         10    1.4328    -0.2338    0.6916    0.9105
         11    1.4328    -0.2338    0.6915    0.9104
         12    1.4328    -0.2338    0.6915    0.9104
         13    1.4328    -0.2338    0.6915    0.9104
```

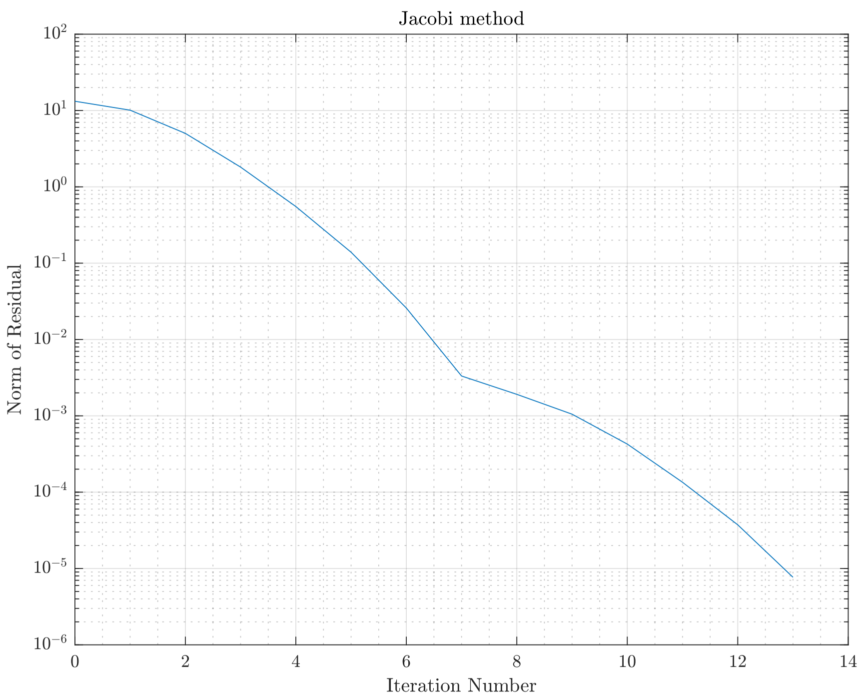

Figure 1

□

Problem 2. Solve again the system above, in MATLAB, using the Gauss-Seidel method. Produce the same results as for the previous problem.

Solution.

- Function file `gaussSeidel.m`

```
function [x, iters, res] = gaussSeidel(A, b, x0, maxIter, tol)
%GAUSSSEIDEL: Gauss-Seidel Method
% Syntax: [x, iters] = gaussSeidel(A, b, x0, maxIter, tol)
% Inputs:
%   A       = coefficient matrix (matrix)
%   b       = right-hand side vector (column vector)
%   x0      = solution to the linear system (column vector)
%   maxIter = maximum of number of iterations (scalar)
%   tol     = tolerance (scalar)
% Outputs:
%   x     = solution to the linear system (matrix, i-th column is the solution in the i-th
    iteration)
%   iters = number of iterations performed (vector)
%   res   = norm of residuals (vector)
%
% Author: Libao Jin
% Date: 10/04/2020


L = tril(A);
U = triu(A, 1);
x = [x0];
iters = [0];
res = [norm(b - A * x0)];
i = 0;
while i <= maxIter && res(end) > tol
    x0 = L \ (b - U * x0);
    res = [res norm(b - A * x0)];
    x = [x x0];
    i = i + 1;
    iters = [iters i];
end

end
```

- Script file `hw3_p2.m`

```
% MATH 3340, Fall 2020
% Homework 3, Problem 2
% Author: Libao Jin
% Date: 10/04/2020

clc; clear; figure(2);

% change default text interpreter to LaTeX
set(groot, 'defaultTextInterpreter','latex');
set(groot, 'defaultAxesTickLabelInterpreter','latex');
set(groot, 'defaultLegendInterpreter','latex')
```

```

A = [5 -1 1 1; 1 7 2 2; 2 1 5 1; 1 -1 1 4];
b = [9; 3; 7; 6];
x0 = [0; 0; 0; 0];
maxIter = 100;
tol = 1e-5;
[x, iters, res] = gaussSeidel(A, b, x0, maxIter, tol);

fprintf('%10s %10s %10s %10s %10s\n', 'iterations', 'x', 'y', 'z', 'w');
for i = 1:length(x)
    fprintf('%10d %10.4f %10.4f %10.4f %10.4f\n', i, x(1, i), x(2, i), x(3, i), x(4, i));
end

semilogy(iters, res);
xlabel('Iteration Number');
ylabel('Norm of Residual');
title('Gauss-Seidel method');
grid on;
grid minor;
```

- Output file `hw3_p2.txt`

```
iterations          x          y          z          w
         1     0.0000     0.0000     0.0000     0.0000
         2     1.8000     0.1714     0.6457     0.9314
         3     1.5189    -0.2390     0.6540     0.8970
         4     1.4420    -0.2206     0.6879     0.9124
         5     1.4358    -0.2338     0.6899     0.9101
         6     1.4332    -0.2333     0.6914     0.9105
         7     1.4330    -0.2338     0.6915     0.9104
         8     1.4329    -0.2338     0.6915     0.9105
         9     1.4328    -0.2338     0.6915     0.9104
        10     1.4328    -0.2338     0.6915     0.9104
```

□

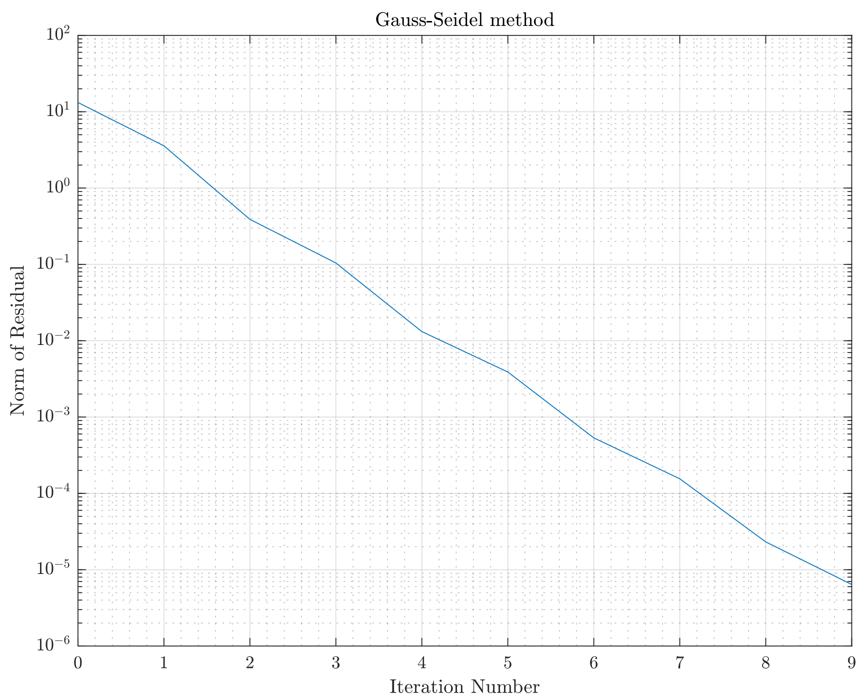
Gauss-Seidel method
Norm of Residual
Iteration Number
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
0 1 2 3 4 5 6 7 8 9

Figure 2

Problem 3. Solve the twenty-one systems of equations:

$$\begin{cases} 7x + y + 2z = 0.01m^2 - 2m \\ x - 5y + 2z = 2 - m \\ 2x + y + 5z = 9 \end{cases}$$

obtained by setting, in turn, the value of the quantity m on the right-hand side to all integers in between, and including, $m = 0$ and $m = 20$. This time use the LU-decomposition of the system matrix that is implemented in MATLAB by the `lu` function. Perform the decomposition only once, then use the lower- and upper-triangular factors repeatedly to find each successive solution. Turn in the code and a plot of the first and second components of the solution (that is, the values of x and y) as a function of thee right-hand side parameter m.

Solution.

- Script file `hw3_p3.m`

```matlab
% MATH 3340, Fall 2020
% Homework 3, Problem 3
% Author: Libao Jin
% Date: 10/04/2020
clc; clear; figure(3); hold on;

% change default text interpreter to LaTeX
set(groot, 'defaultTextInterpreter','latex');
set(groot, 'defaultAxesTickLabelInterpreter','latex');
set(groot, 'defaultLegendInterpreter','latex')

A = [7 1 2; 1 -5 2; 2 1 5];
[L, U] = lu(A);
x = [];

for m = 0:20
    b = [0.01 * m^2 - 2 * m; 2 - m; 9];
    z = L \ b;
    x(:, m + 1) = U \ z;
end

fprintf('%9s %10s %10s %10s\n', 'm', 'x', 'y', 'z');
for i = 1:size(x, 2)
    fprintf('%9d %10.4f %10.4f %10.4f\n', i - 1, x(1, i), x(2, i), x(3, i));
end

m = 0:20;
plot(m, x(1, :), 'o-');
plot(m, x(2, :), '^--');
xlabel('$m$');
ylabel('$x$ and $y$');
title('System of Linear Equations');
legend('$x$', '$y$', 'Location', 'best');
grid on;
grid minor;
```

- Output file `hw3_p3.txt`

```
         m          x          y          z
         0    -0.6071     0.2738     1.9881
         1    -0.9448     0.4465     2.0886
         2    -1.2793     0.6193     2.1879
         3    -1.6105     0.7922     2.2858
         4    -1.9386     0.9652     2.3824
         5    -2.2634     1.1384     2.4777
         6    -2.5850     1.3117     2.5717
         7    -2.9034     1.4851     2.6643
         8    -3.2186     1.6586     2.7557
         9    -3.5305     1.8322     2.8458
        10    -3.8393     2.0060     2.9345
        11    -4.1448     2.1798     3.0220
        12    -4.4471     2.3538     3.1081
        13    -4.7463     2.5279     3.1929
        14    -5.0421     2.7021     3.2764
        15    -5.3348     2.8765     3.3586
        16    -5.6243     3.0510     3.4395
        17    -5.9105     3.2255     3.5191
        18    -6.1936     3.4002     3.5974
        19    -6.4734     3.5751     3.6743
        20    -6.7500     3.7500     3.7500
```

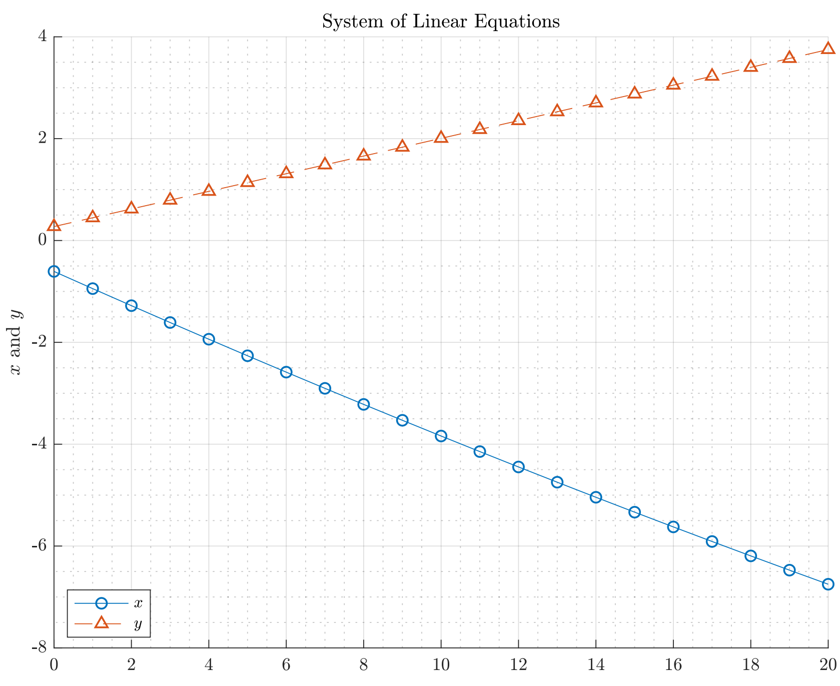

Figure 3

□