Simulation Assignment 1 Report

ECE 204
September 29, 2023
Punit Shah and Dylan Nogueira

Codes

PART A

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ECE 204 Simulation Assignment 1%
% Punit Shah and Dylan Nogueira  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

A = load('A.txt');
B = load('B.txt');

% disp(AB);

% disp(AB);

% create a new matrix with the vector
AB = [A,B];
% finding the row size and column size to use in the for loops
[rsize, csize] = size(A);

%forward elimination section
iterationnum = 0;
for i = 1:csize

   % set the current maxval to the diaganol values within AB
   maxval = AB(i,i);
   % absolute value the current maximum value
   maxval = abs(maxval);
   row = i;
   col = i;

   % iterate through the rest of the matrix to find if there is a new max
   % value
   for p = i:rsize
      for d = i:i
         if abs(AB(p,d)) > maxval
            maxval = abs(AB(p,d));

            % if there is a new maxval, set the row and col values to
            % p and d respectively, else leave them as i and i
            row = p;
            col = d;
         end
      end
   end
```

```
    % disp(row);

    % if i is not the "row" value, swap the rows
    if i ~= row
        AB([i,row],:) = AB([row,i],:);
        % disp(AB);
    end

    % set the new value of AB(i,:) to a new value divided by AB(i,i)
    AB(i,:) = AB(i,:)/AB(i,i);

    % disp(AB);

    %
    for j = i+1:rsize
        % stores the value of AB(j,i) into the variable del, that will be
        % used as the factor to delete the other value in the matrix
        del = AB(j,i);
        % deletes the values within the matrix, and updates the values
        % within the row using the del factor
        AB(j,:) = (AB(j,:) - del* AB(i,:));
        % round the value to 5 decimal places
        AB(j,:) = round(AB(j,:),5);
        iterationnum = iterationnum + 1;
        % display(iterationnum);
        % disp(AB);
    end
end

disp("iteration number = ");
disp(iterationnum);
disp('final');
disp(AB);

% back substitution section

% creates a new matrix of all zeros that we can write to as a clean state
empty = zeros(csize,1);

% start the for loop from csize up to 1
for i = csize:-1:1
    % set empty(i) as the last element in the row, or the element in the
    % vector with the same row number thereby storing a known value
    empty(i) = AB(i, end);
```

```
   for j = i+1:csize
      % subtract the coefficient of the values in AB(i,j) multiplied by
      % empty(j) from the current known values stored in empty(i),
      % thereby performing back substitution
      empty(i) = empty(i) - AB(i,j) * empty(j);
      % round the values in empty to 5 decimal places
      empty(i) = round(empty(i),5);
   end
end

% display the solved values
disp(empty)
```

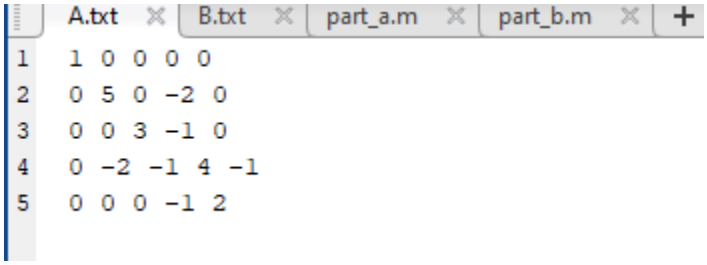
PART B

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ECE 204 Simulation Assignment 1%
% Punit Shah and Dylan Nogueira  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


A = load('A.txt');
B = load('B.txt');

% finding the row size and column size to use in the for loops
[rsize,csize] = size(A);
% disp(rsize);

% setting the hard-stop maximum error obtained when performing gauss
% seidel, select the desired maximum error value.
maxerror = 0.01;
% maxerror = 0.001;
% maxerror = 0.0001;

% setting a counter for the number of iterations performed
iterationnum = 0;

% initializing a current error counter that is greater than the max error
% so the while loop will run. it will be updated within the while loop
errorcurr = 1;

% creates a new matrix of all zeros that we can write to as a clean slate
new = zeros(1,rsize);

% creats a new matrix with the vector
AB = [A,B];

% diaganolize the matrix by finding what the element in the diaganol is,
% and divide each value in the row by that value. thus creating a diaganol
% of 1s
for i = 1:csize
   temp = (A(i,i));
   % create another for loop to run through all the elements in AB, while
   % taking the values from A
   for j = 1:rsize+1
      AB(i,j) = (AB(i,j)) / temp;
   end
end
```

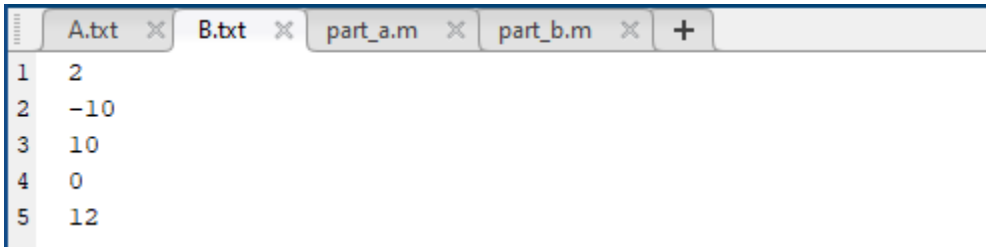
```
% check to see if diaganolization worked
disp(AB);

% while loop to keep the program running until the current error is less
% than the max error
while (errorcurr > maxerror)
   % set the new value into old
   old = new;


   for i = 1:rsize
      % create a new variable we can write to as a clean slate
      sumval = 0;

      % calculate the sum of terms before the current value
      for j = 1:i-1
         sumval = sumval + A(i,j) * new(j);
      end

      % calculate the sum of terms after the current value
      for j = i+1:rsize
         sumval = sumval + A(i,j) * old(j);
      end

      % update the vector "new" with the new values and chop the values
      % to 5 significant digits.
      new(i) = round((1/A(i,i))*(B(i)-sumval),5);
   end

   % increase the iteration number for each run through the while loop
   iterationnum = iterationnum + 1;

   % set the new current error value to check if while loop condition is
   % still valid
   errorcurr = abs((new-old)/new);
end

% display the iteration number
disp("iteration number = ");
disp(iterationnum);

% display the finalized matrix
disp(new);
```

Pictures

A.txt (Matrix)



B.txt (Vector)



PART A

A.txt | B.txt | part_a.m | part_b.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ECE 204 Simulation Assignment 1%
% Punit Shah and Dylan Nogueira  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

A = load('A.txt');
B = load('B.txt');

% disp(AB);

% disp(AB);

% create a new matrix with the vector
AB = [A,B];
% finding the row size and column size to use in the for loops
[rsize, csize] = size(A);

%forward elimination section
iterationnum = 0;
for i = 1:csize

    % set the current maxval to the diaganol values within AB
    maxval = AB(i,i);
    % absolute value the current maximum value
    maxval = abs(maxval);
    row = i;
    col = i;

    % iterate through the rest of the matrix to find if there is a new max
    % value
    for p = i:rsize
        for d = i:i
            if abs(AB(p,d)) > maxval
                maxval = abs(AB(p,d));

                % if there is a new maxval, set the row and col values to
                % p and d respectively, else leave them as i and i
                row = p;
                col = d;
            end
        end
    end
```

A.txt | B.txt | part_a.m | part_b.m

```
41 -            end
42 -        end
43
44          % disp(row);
45
46          % if i is not the "row" value, swap the rows
47 -        if i ~= row
48 -            AB([i,row],:) = AB([row,i],:);
49              % disp(AB);
50 -        end
51
52          % set the new value of AB(i,:) to a new value divided by AB(i,i)
53 -        AB(i,:) = AB(i,:)/AB(i,i);
54
55          % disp(AB);
56
57          %
58 -        for j = i+1:rsize
59              % stores the value of AB(j,i) into the variable del, that will be
60              % used as the factor to delete the other value in the matrix
61 -            del = AB(j,i);
62              % deletes the values within the matrix, and updates the values
63              % within the row using the del factor
64 -            AB(j,:) = (AB(j,:) - del* AB(i,:));
65              % round the value to 5 decimal places
66 -            AB(j,:) = round(AB(j,:),5);
67 -            iterationnum = iterationnum + 1;
68              % display(iterationnum);
69              % disp(AB);
70 -        end
71 -    end
72
73 -    disp("iteration number = ");
74 -    disp(iterationnum);
75 -    disp('final');
76 -    disp(AB);
77
78      % back substitution section
79
80      % creates a new matrix of all zeros that we can write to as a clean state
81 -    empty = zeros(csize,1);
82
83      % start the for loop from csize up to 1
```

```
82
83      % start the for loop from csize up to 1
84 -    for i = csize:-1:1
85          % set empty(i) as the last element in the row, or the element in the
86          % vector with the same row number thereby storing a known value
87 -        empty(i) = AB(i, end);
88
89 -        for j = i+1:csize
90              % subtract the coefficient of the values in AB(i,j) multiplied by
91              % empty(j) from the current known values stored in empty(i),
92              % thereby performing back substitution
93 -            empty(i) = empty(i) - AB(i,j) * empty(j);
94              % round the values in empty to 5 decimal places
95 -            empty(i) = round(empty(i),5);
96 -        end
97 -    end
98
99      % display the solved values
100 -   disp(empty)
```

PART B

A.txt | B.txt | part_a.m | part_b.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ECE 204 Simulation Assignment 1%
% Punit Shah and Dylan Nogueira  %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


A = load('A.txt');
B = load('B.txt');

% finding the row size and column size to use in the for loops
[rsize,csize] = size(A);
% disp(rsize);

% setting the hard-stop maximum error obtained when performing gauss
% seidel, select the desired maximum error value.
maxerror = 0.01;
% maxerror = 0.001;
% maxerror = 0.0001;

% setting a counter for the number of iterations performed
iterationnum = 0;

% initializing a current error counter that is greater than the max error
% so the while loop will run. it will be updated within the while loop
errorcurr = 1;

% creates a new matrix of all zeros that we can write to as a clean slate
new = zeros(1,rsize);

% creats a new matrix with the vector
AB = [A,B];

% diaganolize the matrix by finding what the element in the diaganol is,
% and divide each value in the row by that value. thus creating a diaganol
% of 1s
for i = 1:csize
    temp = (A(i,i));
    % create another for loop to run through all the elements in AB, while
    % taking the values from A
    for j = 1:rsize+1
        AB(i,j) = (AB(i,j)) / temp;
    end
end
```

```
43 -    end
44
45      % check to see if diaganolization worked
46 -    disp(AB);
47
48      % while loop to keep the program running until the current error is less
49      % than the max error
50 -    while (errorcurr > maxerror)
51          % set the new value into old
52 -        old = new;
53
54
55 -        for i = 1:rsize
56              % create a new variable we can write to as a clean slate
57 -            sumval = 0;
58
59              % calculate the sum of terms before the current value
60 -            for j = 1:i-1
61 -                sumval = sumval + A(i,j) * new(j);
62 -            end
63
64              % calculate the sum of terms after the current value
65 -            for j = i+1:rsize
66 -                sumval = sumval + A(i,j) * old(j);
67 -            end
68
69              % update the vector "new" with the new values and chop the values
70              % to 5 significant digits.
71 -            new(i) = round((1/A(i,i))*(B(i)-sumval),5);
72 -        end
73
74          % increase the iteration number for each run through the while loop
75 -        iterationnum = iterationnum + 1;
76
77          % set the new current error value to check if while loop condition is
78          % still valid
79 -        errorcurr = abs((new-old)/new);
80 -    end
```

```
76
77          % set the new current error value to check if while loop condition is
78          % still valid
79 -        errorcurr = abs((new-old)/new);
80 -    end
81
82      % display the iteration number
83 -    disp("iteration number = ");
84 -    disp(iterationnum);
85
86      % display the finalized matrix
87 -    disp(new);
```

Command Window Outcomes

PART A

```
Command Window
>> part_a
iteration number =
    10

final
    1.0000         0         0         0         0    2.0000
         0    1.0000         0   -0.4000         0   -2.0000
         0         0    1.0000   -0.3333         0    3.3333
         0         0         0    1.0000   -0.3488   -0.2326
         0         0         0         0    1.0000    7.1268

    2.0000
   -1.0986
    4.0845
    2.2535
    7.1268
```

PART B

maxerror = 0.01 (1%)

```
Command Window
>> part_b
    1.0000         0         0         0         0    2.0000
         0    1.0000         0   -0.4000         0   -2.0000
         0         0    1.0000   -0.3333         0    3.3333
         0   -0.5000   -0.2500    1.0000   -0.2500         0
         0         0         0   -0.5000    1.0000    6.0000

iteration number =
     6

    2.0000   -1.1255    4.0621    2.2260    7.1130

fx >>
```

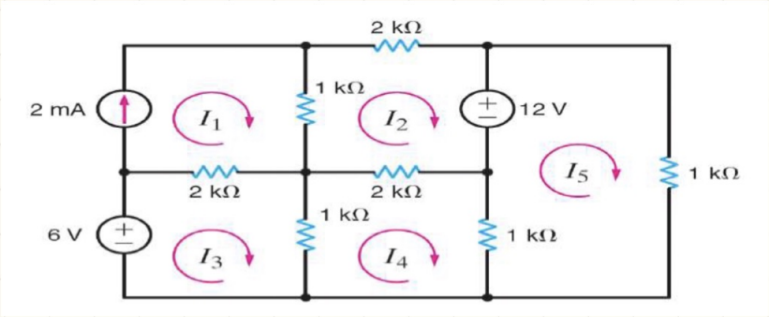
maxerror = 0.001 (0.1%)

```
Command Window
>> part_b
    1.0000         0         0         0         0    2.0000
         0    1.0000         0   -0.4000         0   -2.0000
         0         0    1.0000   -0.3333         0    3.3333
         0   -0.5000   -0.2500    1.0000   -0.2500         0
         0         0         0   -0.5000    1.0000    6.0000

iteration number =
     8

    2.0000   -1.1031    4.0808    2.2489    7.1245

fx >>
```

maxerror = 0.0001 (0.01%)

```
Command Window
>> part_b
    1.0000         0         0         0         0    2.0000
         0    1.0000         0   -0.4000         0   -2.0000
         0         0    1.0000   -0.3333         0    3.3333
         0   -0.5000   -0.2500    1.0000   -0.2500         0
         0         0         0   -0.5000    1.0000    6.0000

iteration number =
    11

    2.0000   -1.0989    4.0842    2.2532    7.1266

fx >>
```

Work For Matrices



$I_1 = 0.002A$ (1)

(2) $= 12 + 1000(I_2 - 0.002) + 2000 I_2 + 2000(I_2 - I_4) = 0$

$12 + 1000 I_2 - 2 + 2000 I_2 + 2000 I_2 - I_4 = 0$

$5600 I_2 + 12 - 2 - 2000 I_4 = 0$

$5000 I_2 - 2000 I_4 = -10$ (2)

(3) $= -6 + 2000(I_3 - I_1) + 1000(I_3 - I_4) = 0$

$-2000(0.002) + 3000 I_3 + 1000 I_4 = 6$ (3)

$3000 I_3 - 1000 I_4 = 10$

(4) $= 1000(I_4 - I_3) + 2000(I_4 - I_2) + 1000(I_4 - I_5) = 0$

$4000 I_4 - 1000 I_3 - 2000 I_2 - 1000 I_5 = 0$ (4)

(5) $= -12 + 1000(I_5) + 1000(I_5 - I_4) = 0$

$2000 I_5 - 1000 I_4 = 12$ (5)

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & -2 & 0 \\ 0 & 0 & 3 & -1 & 0 \\ 0 & -2 & -1 & 4 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} 2 \\ -10 \\ 10 \\ 0 \\ 12 \end{bmatrix}$$