Lab 05
MATH 3180: Numerical Analysis

Jackson Cole

October 04, 2018

Contents

CSCI/MATH 3180
Lab Assignment #5

Part I. Use Maple to find the solutions for the linear systems given in class.

Part II. Use Visual Studio 2015 to find the solutions for the linear systems.

1. Create a C++ console application project in Visual Studio 2015 and name your project YourLastName5.
2. Write a program that implements 1) the Naïve Gaussian Elimination and 2) Gaussian Elimination with Partial Pivoting for linear systems.
3. All floating point arithmetic will be double precision.
4. Input to the main program: Name of a data file that contains a sequence of augmented matrices each of which represents a linear system
5. Program output:
   For each linear system
   - Original augmented matrix
   - Upper triangular matrix obtained by the Naïve Gaussian Elimination
   - Solution from the Naïve Gaussian Elimination
   - Upper triangular matrix obtained by the Gaussian Elimination with Partial Pivoting
   - Solution from the Gaussian Elimination with Partial Pivoting
6. Analyze your output and write a short report including the following
   - Description of your experiment
   - For each of the data sets
     - Program input
     - Program output
   - Your conclusion/findings
   - Save your report as **YourLastNameReport5**.

Submission

1. Delete the following from your Visual Studio project folder.
   - *Debug* sub-folder
   - *Debug* sub-sub-folder under your project folder(second level down)
   - *sdf* file.
2. Save the following in a compressed (zipped) folder.
   PartI :
   YourLastNameMaple5.mw
   PartII :
   YourLastName5 -- *main project folder*
   Note: A data file must be in the project folder.
   YourLastNameReport5 -- *report on the experiment*
3. Submit the compressed folder to D2L.
4. **Confirm** your submission.
   - **Download** the zipped folder which you have submitted and **check the contents**.
   - Multiple submissions are allowed, but the last submission will be graded.

NOTE: LABS MUST BE INDEPENDENT WORK.

LAB #5 EVALUATION RUBRIC

<table>
<tr><td>Part I</td><td></td><td>Use Maple to find the solutions for the linear systems.</td><td>__/1.5</td></tr>
<tr><td rowspan="5">Part II</td><td>1</td><td>Solve the assigned problem using methods described in program description.
See Requirements #1 through #5</td><td>__/2</td></tr>
<tr><td>2</td><td>Compilation/Execution
✓ Compile without errors.
✓ Execute without crashing.
✓ Work for all data and produce correct answers.
✓ The program output well formatted and properly labeled.</td><td>__/4</td></tr>
<tr><td>3</td><td>Main Comment Block includes the following.
file name due date author course #
program description input output</td><td>__/0.5</td></tr>
<tr><td>4</td><td>Documentation, indentation, and white space usage
✓ Meaning variable names are used and they are briefly described.
✓ Each section of statements in the program is well documented.
✓ Proper INDENTATION is used to make the program easier to read.
✓ WHITE SPACES are used in appropriate places for readability.</td><td>__/0.5</td></tr>
<tr><td>5</td><td>Contents of report
• Brief description of your experiment
• For each of the data sets
o Program input
o Program output
• Your conclusion/findings</td><td>__/1.5</td></tr>
<tr><td>Submission</td><td></td><td>Contents of zipped folder
✓ Zip folder contains the following.
❖ Maple worksheet.
❖ Zip folder contains the project folder and the report.
❖ Project folder contains a data file.
✓ The project folder does NOT contain the following.
❖ Debug sub-folder
❖ Debug sub-sub-folder
❖ .sdf file</td><td>__/0.5</td></tr>
<tr><td colspan="3">TOTAL</td><td>__/10</td></tr>
</table>

```
2
3     2     4
-6    -4    6

2
6     -3    6
-2    1     -2

2
0     2     4
1     -1    5

3
1     1     2     4
1     1     0     2
0     1     1     0

3
3     2     -5    0
2     -3    1     0
1     4     -1    4

3
3     4     3     10
1     5     -1    7
6     3     7     15

4
6     -2    2     4     16
12    -8    6     10    26
3     -13   9     3     -19
-6    4     1     -18   -34

4
1     -1    2     1     1
3     2     1     4     1
5     8     6     3     1
4     2     5     3     -1

5
9     3     2     0     7     35
7     6     9     6     4     58
2     7     7     8     2     53
0     9     7     2     2     37
7     3     6     4     3     39
```

0.1 Description of Experiment

In this experiment, our goal was to implement a Naive Gaussian Elimination method as well as a method for Gaussian Elimination with Partial Pivoting in C++. Both of these methods also require implementation of back substitution given a general matrix. We were also asked to read in data from a data file of a given format (this data is shown in the next section, but is read in from a data file in practice). We were to implement these methods to work for all data and for other data files as well.

0.2 Program Input

```
2
3       2       4
-6      -4      6

2
6       -3      6
-2      1       -2

2
0       2       4
1       -1      5

3
1       1       2       4
1       1       0       2
0       1       1       0

3
3       2       -5      0
2       -3      1       0
1       4       -1      4

3
3       4       3       10
1       5       -1      7
6       3       7       15

4
6       -2      2       4       16
12      -8      6       10      26
3       -13     9       3       -19
-6      4       1       -18     -34

4
1       -1      2       1       1
3       2       1       4       1
5       8       6       3       1
4       2       5       3       -1

5
9       3       2       0       7       35
7       6       9       6       4       58
2       7       7       8       2       53
0       9       7       2       2       37
7       3       6       4       3       39
```

0.3 Program Output

```
You may now enter the path to a data file or choose the
default by pressing enter. The default is './data.txt'.
>
********************************************************************************
Original Augmented Matrix:
   3.000   2.000   |   4.000
  -6.000  -4.000   |   6.000

Naive Gaussian Elimination
> Upper triangular matrix obtained:
   3.000   2.000   |   4.000
   0.000   0.000   |  14.000
> SOLUTION:
x1 = -inf
x2 = inf

Gaussian Elimination with Partial Pivoting
> Upper triangular matrix obtained:
   3.000   2.000   |   4.000
   0.000   0.000   |  14.000
> SOLUTION:
x1 = -inf
x2 = inf
********************************************************************************

********************************************************************************
Original Augmented Matrix:
   6.000  -3.000   |   6.000
  -2.000   1.000   |  -2.000

Naive Gaussian Elimination
> Upper triangular matrix obtained:
   6.000  -3.000   |   6.000
   0.000   0.000   |   0.000
> SOLUTION:
x1 = nan
x2 = nan

Gaussian Elimination with Partial Pivoting
> Upper triangular matrix obtained:
   6.000  -3.000   |   6.000
   0.000   0.000   |   0.000
> SOLUTION:
x1 = nan
x2 = nan
********************************************************************************

********************************************************************************
Original Augmented Matrix:
   0.000   2.000   |   4.000
   1.000  -1.000   |   5.000

Naive Gaussian Elimination
> Upper triangular matrix obtained:
   0.000   2.000   |   4.000
     nan    -inf   |    -inf
```

```
> SOLUTION:
x1 = nan
x2 = nan

Gaussian Elimination with Partial Pivoting
> Upper triangular matrix obtained:
   1.000  -1.000   |   5.000
   0.000   2.000   |   4.000
> SOLUTION:
x1 = 7.000
x2 = 2.000
********************************************************************************

********************************************************************************
Original Augmented Matrix:
   1.000   1.000   2.000   |   4.000
   1.000   1.000   0.000   |   2.000
   0.000   1.000   1.000   |   0.000

Naive Gaussian Elimination
> Upper triangular matrix obtained:
   1.000   1.000   2.000   |   4.000
   0.000   0.000  -2.000   |  -2.000
   0.000     nan     inf   |     inf
> SOLUTION:
x1 = nan
x2 = nan
x3 = nan

Gaussian Elimination with Partial Pivoting
> Upper triangular matrix obtained:
   1.000   1.000   0.000   |   2.000
   0.000   1.000   1.000   |   0.000
   0.000   0.000   2.000   |   2.000
> SOLUTION:
x1 = 3.000
x2 = -1.000
x3 = 1.000
********************************************************************************

********************************************************************************
Original Augmented Matrix:
   3.000   2.000  -5.000   |   0.000
   2.000  -3.000   1.000   |   0.000
   1.000   4.000  -1.000   |   4.000

Naive Gaussian Elimination
> Upper triangular matrix obtained:
   3.000   2.000  -5.000   |   0.000
   0.000  -4.333   4.333   |   0.000
   0.000   0.000   4.000   |   4.000
> SOLUTION:
x1 = 1.000
x2 = 1.000
x3 = 1.000

Gaussian Elimination with Partial Pivoting
```

```
> Upper triangular matrix obtained:
   2.000  -3.000   1.000   |   0.000
   0.000   6.500  -6.500   |   0.000
   0.000   0.000   4.000   |   4.000
> SOLUTION:
x1 = 1.000
x2 = 1.000
x3 = 1.000
********************************************************************************

********************************************************************************
Original Augmented Matrix:
   3.000   4.000   3.000   |  10.000
   1.000   5.000  -1.000   |   7.000
   6.000   3.000   7.000   |  15.000

Naive Gaussian Elimination
> Upper triangular matrix obtained:
   3.000   4.000   3.000   |  10.000
   0.000   3.667  -2.000   |   3.667
   0.000   0.000  -1.727   |   0.000
> SOLUTION:
x1 = 2.000
x2 = 1.000
x3 = -0.000

Gaussian Elimination with Partial Pivoting
> Upper triangular matrix obtained:
   6.000   3.000   7.000   |  15.000
   0.000   4.500  -2.167   |   4.500
   0.000   0.000   0.704   |   0.000
> SOLUTION:
x1 = 2.000
x2 = 1.000
x3 = 0.000
********************************************************************************

********************************************************************************
Original Augmented Matrix:
   6.000  -2.000   2.000   4.000   |  16.000
  12.000  -8.000   6.000  10.000   |  26.000
   3.000 -13.000   9.000   3.000   | -19.000
  -6.000   4.000   1.000 -18.000   | -34.000

Naive Gaussian Elimination
> Upper triangular matrix obtained:
   6.000  -2.000   2.000   4.000   |  16.000
   0.000  -4.000   2.000   2.000   |  -6.000
   0.000   0.000   2.000  -5.000   |  -9.000
   0.000   0.000   0.000  -3.000   |  -3.000
> SOLUTION:
x1 = 3.000
x2 = 1.000
x3 = -2.000
x4 = 1.000

Gaussian Elimination with Partial Pivoting
```

```
> Upper triangular matrix obtained:
   3.000 -13.000   9.000   3.000   | -19.000
   0.000  24.000 -16.000  -2.000   |  54.000
   0.000   0.000   4.333 -13.833   | -22.500
   0.000   0.000   0.000  -0.462   |  -0.462
> SOLUTION:
x1 = 3.000
x2 = 1.000
x3 = -2.000
x4 = 1.000
********************************************************************************

********************************************************************************
Original Augmented Matrix:
   1.000  -1.000   2.000   1.000   |   1.000
   3.000   2.000   1.000   4.000   |   1.000
   5.000   8.000   6.000   3.000   |   1.000
   4.000   2.000   5.000   3.000   |  -1.000

Naive Gaussian Elimination
> Upper triangular matrix obtained:
   1.000  -1.000   2.000   1.000   |   1.000
   0.000   5.000  -5.000   1.000   |  -2.000
   0.000   0.000   9.000  -4.600   |   1.200
   0.000   0.000   0.000  -0.667   |  -3.000
> SOLUTION:
x1 = -7.233
x2 = 1.133
x3 = 2.433
x4 = 4.500

Gaussian Elimination with Partial Pivoting
> Upper triangular matrix obtained:
   5.000   8.000   6.000   3.000   |   1.000
   0.000  -2.600   0.800   0.400   |   0.800
   0.000   0.000  -1.154  -0.077   |  -3.154
   0.000   0.000   0.000   2.000   |   9.000
> SOLUTION:
x1 = -7.233
x2 = 1.133
x3 = 2.433
x4 = 4.500
********************************************************************************

********************************************************************************
Original Augmented Matrix:
   9.000   3.000   2.000   0.000   7.000   |  35.000
   7.000   6.000   9.000   6.000   4.000   |  58.000
   2.000   7.000   7.000   8.000   2.000   |  53.000
   0.000   9.000   7.000   2.000   2.000   |  37.000
   7.000   3.000   6.000   4.000   3.000   |  39.000

Naive Gaussian Elimination
> Upper triangular matrix obtained:
   9.000   3.000   2.000   0.000   7.000   |  35.000
   0.000   3.667   7.444   6.000  -1.444   |  30.778
   0.000   0.000  -6.303  -2.364   2.939   |  -7.939
```

```
   0.000   0.000   0.000  -8.500   0.288   | -24.346
   0.000   0.000   0.000   0.000  -0.681   |  -2.724
> SOLUTION:
x1 = 0.000
x2 = 1.000
x3 = 2.000
x4 = 3.000
x5 = 4.000

Gaussian Elimination with Partial Pivoting
> Upper triangular matrix obtained:
   7.000   3.000   6.000   4.000   3.000   |  39.000
   0.000   9.000   7.000   2.000   2.000   |  37.000
   0.000   0.000   0.667   1.333   0.333   |   6.667
   0.000   0.000   0.000   5.143   5.857   |  38.857
   0.000   0.000   0.000   0.000  -5.574   | -22.296
> SOLUTION:
x1 = 0.000
x2 = 1.000
x3 = 2.000
x4 = 3.000
x5 = 4.000
********************************************************************************
```

0.4 Conclusion/Findings

We see clearly from several of the earlier matrices in this experiment that the Naive Gaussian Elimination method indeed fails as a result of division by 0. In many cases, we recieve a `nan` value for several of our solutions. This problem is remedied by employing the method of partial pivoting during Gaussian Elimination with Partial Pivoting. In any case, the resulting solutions from either method are obtained by simply doing back substitution, which in both cases is an identical process.

Given the power of modern computing, it seems as though there would not be very many cases where performing Naive Gaussian Elimination in practice would be beneficial, as the Gaussian Elimination with Partial Pivoting provides better results in every case.