EE 242 Numerical Methods for Electrical Engineering

Project 1: Gaussian Elimination with Partial Pivoting

Project Instructions:

• For this project, you will work alone. No collaborations of any sort will be allowed with others. Any violation, regardless of the scope, will be directly referred to the department's Ethical Commission.

• You will submit your program (fully commented and documented) to CIMS. Late submission penalty is 20% for up to one week after the deadline. No credits will be given for late submissions beyond one week.

• You will write in C or C++. You can use Dev-C++ as a compiler or any other compiler you wish. You can download Dev-C++ from: http://dev-c.en.malavida.com/

• Your project will not only be graded on whether it works or not, but also on whether it has good programming style.

• You should turn in:

– *Source code:* Fully commented. You should be explicit in your comments. An educated person reading your code should clearly understand the purpose of each line. Executable or object files are not accepted. The file name should be **source.cpp**

– *A Readme file:* A short file named **readme.txt** containing information regarding how to compile and run your program including the necessary arguments. If your program is incomplete, this should be indicated in the beginning of the Readme file.

Important: You should upload two files, named source.cpp and readme.txt.

DO NOT upload .zip or .rar files, or you will be penalized.

Project Goals:

In this project, you will be implementing the Gaussian elimination algorithm with partial pivoting together with backward substitution to solve Ax = b, where A is an n by n square matrix.

Your program should read A and b from two input files and output the solution x as a text file.

Programming Details:

Your program should

• read A matrix from A.txt file and b vector from b.txt file. Each line in a file represents a row,

Example: The file "A.txt" contains:

3.14 1.59 2.65 3.58

9.79 3.23 8.46 2.64

3.38 3.27 9.50 2.88

4.19 7.16 9.39 9.37

The file “b.txt” contains:

5.10

5.82

0.97

4.94

• use dynamically allocated memory to store the matrix and the vector,

• print out an error message and quit if it detects that A is singular, (Don’t forget to consider the machine precision while detecting singularity.)

• print out the elements of the solution x (in the correct order) and write them in a text file.

*******The Case of High Condition Numbers*******

• In order not to add a great amount of burden onto this project, you are not obligated to find the condition numbers for every given matrix. However in the case of 2⇥2 matrices, your program is required to output the condition numbers at 1 and infinity. (You can print out these; they are not required in a text file.) In order to showcase how a matrix with a high condition number can cause issues, consider the following example:

A:

1.000 1.000

1.000 1.001

b1:	b2:
2.000	2.000
2.000	2.001

• Solve Ax=b for both vectors and observe how a small change in b affects the result x. Include this example, the results and comment on their inferences in your Readme file.