ELEN2004: Software Development - I
Course Project 2015

Last Updated: March 22, 2015

1 Introduction

This project concerns writing a C++ code that solves a system of linear equations using the Gaussian elimination method with back-substitution. The project is described in 4 sections and an Appendix. The Appendix describes the mathematical steps involved in solving, in a simplistic algorithmic manner, a system of linear equations. As 2nd year engineering students, you should know how the method works. However, if you need to refresh your understanding of how the *Gaussian elimination* algorithm works, Appendix A is a reproduction of the text from *Wolfram Mathworld*[1] that explains it. The next Section 2 gives some background and Section 3 describes the problem and the solution you are expected to present. Section 4 also discusses what is to be submitted for marking, how it should be submitted, and how your project will be evaluated.

1.1 Important information

Submission Deadline: The deadline for the submission of final project deliverables is **17:59hr, Monday April 13th, 2015.** Late submissions will be penalised as described in the School's policy on late submissions.

Grading: The project contributes 30% to the final mark of the course.

Collaboration: The project should be done as a pair. Both partners, except in the case where only one individual is in a pair, will be awarded the same mark or score from assessing the work of the pair.

Cheating and plagiarism: Cheating and plagiarism will not be tolerated - this applies to both source code and the written reports. Suspected cases of cheating or plagiarism will be dealt with in accordance with the process established in the School and may be referred to the Legal Office of the University for further investigation.

Planning and Time Management: You will be required to plan and manage your time to accomplish the goal of the project, i.e., writing a program that solves well conditioned system of linear equations. Establish a disciplined approach right from the start of the project (hint: make use of an engineering notebook as well as a tool such as a spreadsheet for logging time). Between you and your partner. Track the time spent on:

[1] http://mathworld.wolfram.com/GaussianElimination.html

- Reading and understanding (refreshing your understanding) of how the Gaussian elimination algorithm works. This should be easy from your 1st year algebra classes.
- Writing a step-by-step procedure for the Gaussian elimination process.
- Writing a pseudo-code for the algorithm
- Writing and testing your code. Your code should include ample comments to explain what it does.
- Generating a final run of your code
- Writing your final report.

2 Background

The Gaussian elimination method involves applying a elementary row operations to the *augmented matrix* of the coefficients of a system of linear equations. The result is a transformed matrix upper triangular matrix referred to as an *echelon* form.

2.1 Fundamental Ideas

Given a system of m equations in n variables or unknowns, one picks the first equation and subtracts suitable multiples of it from the remaining $m - 1$ equations. In each case choose the multiple so that the subtraction cancels or eliminates the same variable, say x_1. The result is that the remaining $m - 1$ equations contain only $n - 1$ unknowns and x_1 no longer appears.

Next the first equation is set aside and above process is repeated again with the remaining $m - 1$ equations in $n - 1$ unknowns. The process is then continuously repeated such that each cycle reduces the number of variables and the number of equations. The process stops when either:

- There remains one equation in one variable. In that case, there is a unique solution and back-substitution is used to find the values of the other variables.
- There remain variables but no equations. In that case there is no unique solution.
- There remain equations but no variables (i.e. the lowest row(s) of the augmented matrix contain only zeros on the left side of the vertical line). This indicates that either the system of equations is inconsistent or redundant. In the case of inconsistency the information contained in the equations is contradictory. In the case of redundancy, there may still be a unique solution and back-substitution can be used to find the values of the other variables.

We will assume, for the algorithm to be developed, that $\mathbf{m = n}$ and we always have consistent equations. Details of the process are described in the Appendix A.

2.2 Some Basic Required Knowledge

You'll need to have some basic understanding of how to write C++ programs and more specifically how to:

- split your programs into a set of *functions()*
- use one-dimensional and 2-dimensional arrays.

- how to write a test driver for the functions.
- how to do input and output respectively from and to files.

3 Problem Definition

Given a system of m equations in n variables, $m = n$ in our case, we are required to write:

A) A C++ program that solves the system of equations using the Gaussian elimination method. For a system of equations of the form

$$\begin{bmatrix} a_{00} & a_{01} & \cdots & a_{0(k-1)} \\ a_{10} & a_{11} & \cdots & a_{1(k-1)} \\ \vdots & \vdots & \ddots & \vdots \\ a_{(k-1)0} & a_{(k-1)1} & \cdots & a_{(k-1)(k-1)} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{k-1} \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{k-1} \end{bmatrix}$$

B) A step-by-step description of how the code you implement works,

C) A pseudo-code of each of your implemented user defined functions.

The input will be presented in the form
k

$$\begin{array}{ccccc} a_{00} & a_{01} & \cdots & a_{0(k-1)} & b_0 \\ a_{10} & a_{11} & \cdots & a_{1(k-1)} & b_1 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ a_{(k-1)0} & a_{(k-1)1} & \cdots & a_{(k-1)(k-1)} & b_{(k-1)} \end{array}$$

An input file will consist of such groups of coefficients preceded by a single value of k. The program should be split into functions which independently do the following,

- functRead(): reads the value of k and the subsequent values of coefficients of a_{ij}, and b_j one row at a time.
- functGauss(): processes the augmented matrix into a triangular echelon form.
- backSubstitute(): derives the solution of the variables by back substitution.
- functResult(): writes out the solution obtained in the form

$$\begin{array}{ccc} \text{x0} & = & \ldots \\ \text{x1} & = & \ldots \\ \vdots & \vdots & \vdots \\ \text{x(k-1)} & = & \ldots \end{array}$$

- int main(): a main function that calls the preceding defined functions to read each input group and derive a solution for a group of input values.

4 Deliverables and Submission

The deliverables and what has to be submitted should be zipped as a single file. Choose a suitable name for the zipped file. This should contain:

i) All the source code developed for the project,

ii) Output generated for the input data given,

iii) A copy of your project documentation.

4.1 Your Test Data

The test data is available from SAKAI webpage of the course as ***testData.dat***. Submission should be through the SAKAI's course submission page for projects. Submission will be opened in the first week of April.

4.2 Your C++ Source Code

Your source code should be well commented and most importantly, the first two lines must specify the Student Numbers of the pair delivering the code.

4.3 Project Documentation

As part of your submission, you should include a copy of your project documentation. This should be submitted in PDF and sectionalized as per the **blue book** format:

- *A title page*
- *Introduction*
- *Problem Definition - Your understanding, analysis and method of solution*
- *Design Approach - Pseudo-Code of essential functions, Planned time management, etc.*
- *Results*
- *Discussion of Results - limitations, accuracy, etc.*
- *Conclusion*
- *References - Any references you consulted for the project*

You may include in your conclusion any special technique you may have used to speed up your calculations or any ideas that you would recommend for future projects.

A Gaussian Elimination From Woolfram MathWorld

[2] Gaussian elimination is a method for solving matrix equations of the form

$$\mathbf{Ax} = \mathbf{b}$$

To perform Gaussian elimination starting with a system of equations

$$\begin{bmatrix} a_{00} & a_{01} & \dots & a_{0k-1} \\ a_{10} & a_{11} & \dots & a_{1k-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k-10} & a_{k-11} & \dots & a_{k-1k-1} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{k-1} \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{k-1} \end{bmatrix}$$

compose the *augmented matrix equation*

$$\left[\begin{array}{cccc|c} a_{00} & a_{01} & \dots & a_{0k-1} & b_0 \\ a_{10} & a_{11} & \dots & a_{1k-1} & b_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{k-10} & a_{k-11} & \dots & a_{k-1k-1} & b_{k-1} \end{array}\right] \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{k-1} \end{bmatrix}$$

Here, the *column vector* in the variables $\mathbf{x}$ is carried along for labelling the matrix rows. Now, perform *elementary row operations* to put the augmented matrix into the *upper triangular* form

$$\left[\begin{array}{cccc|c} a'_{00} & a'_{01} & \dots & a'_{0k-1} & b'_0 \\ 0 & a'_{11} & \dots & a'_{1k-1} & b'_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & a'_{k-1k-1} & b'_{k-1} \end{array}\right]$$

Solve the equation of the $(k-1)$th row for x_{k-1}, then substitute back into the equation of the $(k-2)$nd row to obtain a solution for $x_{(k-2)}$, etc., according to the formula

$$x_i = \frac{1}{a'_{ii}} \left(b'_i - \sum_{j=i+1}^{k-1} a'_{ij} x_j \right)$$

A matrix that has undergone Gaussian elimination is said to be in *echelon form.*
For example, consider the matrix equation

$$\begin{bmatrix} 9 & 3 & 4 \\ 4 & 3 & 4 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 7 \\ 8 \\ 3 \end{bmatrix}$$

In augmented form, this becomes

$$\left[\begin{array}{ccc|c} 9 & 3 & 4 & 7 \\ 4 & 3 & 4 & 8 \\ 1 & 1 & 1 & 3 \end{array}\right] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

[2] http://mathworld.wolfram.com/GaussianElimination.html

Switching the first and third rows (without switching the elements in the right-hand column vector) gives

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 3 \\ 4 & 3 & 4 & 8 \\ 9 & 3 & 4 & 7 \end{array}\right] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Subtracting 9 times the first row from the third row gives

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 3 \\ 4 & 3 & 4 & 8 \\ 0 & -6 & -5 & -20 \end{array}\right] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Subtracting 4 times the first row from the second row gives

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 3 \\ 0 & -1 & 0 & -4 \\ 0 & -6 & -5 & -20 \end{array}\right] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Finally, adding -6 times the second row to the third row gives

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 3 \\ 0 & -1 & 0 & -4 \\ 0 & 0 & -5 & 4 \end{array}\right] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Restoring the transformed matrix equation gives

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & -1 & 0 \\ 0 & 0 & -5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ -4 \\ 4 \end{bmatrix}$$

which can be solved immediately to give $x_3 = -4/5$, back-substituting to obtain $x_2 = 4$ and $x_1 = -1/5$.