

First Year Computing (C++ Course)

Dr John Joss Chinn

Week 8.

Cubic Splines, Assignment, File Input and Output in C++

Last week we looked at Natural Cubic Splines and how they may be used to fit a smooth curve through a set of x,y data points. This week we will carry on with that and take a look at what we need to do to put the information into a suitable form so that the matrix can be solved using our Gaussian Elimination with Partial Pivoting code.

Last week we ended with a matrix problem of the form

$$\begin{pmatrix} 2(h_0 + h_1) & h_1 & 0 & 0 \\ h_1 & 2(h_1 + h_2) & h_2 & 0 \\ 0 & h_2 & 2(h_2 + h_3) & h_3 \\ 0 & 0 & h_3 & 2(h_3 + h_4) \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix} = \begin{pmatrix} \frac{3}{h_1}(a_2 - a_1) - \frac{3}{h_0}(a_1 - a_0) \\ \frac{3}{h_2}(a_3 - a_2) - \frac{3}{h_1}(a_2 - a_1) \\ \frac{3}{h_3}(a_4 - a_3) - \frac{3}{h_2}(a_3 - a_2) \\ \frac{3}{h_4}(a_5 - a_4) - \frac{3}{h_3}(a_4 - a_3) \end{pmatrix}$$

	x		y
x ₀	0	y ₀	0
x ₁	1	y ₁	2
x ₂	4	y ₂	1
x ₃	6	y ₃	4
x ₄	9	y ₄	5
x ₅	11	y ₅	4

Now we need to insert actual numbers for the h_i and the a_i so as to fill in the matrix on the left hand side and the column vector on the right hand side. Continuing with our example, we had the following x,y values, given in the table. Well, with a_i = y_i and h_i = x_{i+1} - x_i this is actually quite easy. I have got my code to do this for me and to produce an augmented matrix. This looks like the following,

8	3	0	0	-7
3	10	2	0	5.5
0	2	10	3	-3.5
0	0	3	10	-2.5

You could do it by hand but it is quite simple to read in pairs of data points from an external text file and then write a piece of code to form the augmented matrix.

Once the augmented matrix is formed then we use the Gaussian Elimination with Partial Pivoting routine to solve it. Hence all the fuss over the last three weeks with Gaussian Elimination! Having found the values of the c_i (c₁, c₂, c₃ and c₄) and knowing that c₀ = 0 and c₅ = 0, then we can find all of the b_i and the d_i, from equations (14) and (17) from last weeks notes, on page 67

$$d_i = \frac{(c_{i+1} - c_i)}{3h_i} \tag{14}$$

$$b_i = \frac{1}{h_i}(a_{i+1} - a_i) - \frac{h_i}{3}(2c_i + c_{i+1}). \tag{17}$$



Department of Mechanical Engineering, UMIST

Computer Programming in C++

Dr John Chinn

Coursework Assignment: Spline Curves

Task

Write a computer program in C++ to calculate the natural cubic spline curves through a set of x,y data. The data is given in the table below. The suggested way to go about this is as follows:

x	y
0	0
1	1.25
5	1.25
6	1.3
9	2.62
10	2.81
11	1.12
11.87	1.31
12.12	0.87
12.34	1.25
13	1
14	0.94
14.75	0

- (1) Put this set of data points in an external text file.
- (2) Write a piece of code to input the data points from the text file.
- (3) Write a piece of code to manipulate the data points into the augmented matrix (what size will this be?).
- (4) Splice your code with the Gaussian Elimination source code to solve the augmented matrix, for the c_i values.
- (5) Write a piece of code to use the c_i values to find the d_i and b_i values.
- (6) Write a piece of code to calculate the cubic equations for about eight data points between each of these given data points. (i.e. you need enough of your own points to form a graph between each pair of these given data points).
- (7) Write a piece of code to output all of your data points to an external text file.
- (8) Input your text file into Excel and plot the graphs.

Specification

To indicate the intended structure of your computer program I would suggest that it ought to include the following, as a minimum:-

- (1) The ability to read in two columns of data, x and y values, from a simple text file.
- (2) The ability to detect the end of the data file.
- (3) Use of arrays.
- (4) Construction of the augmented matrix.
- (5) Implementation of the Gaussian Elimination code to solve the matrix for the c_i .
- (6) Calculation of the coefficients c_i , d_i and b_i .
- (7) Form the cubic spline equations.
- (8) Use the cubic spline equations to put, say ten, data points between each pair of given x,y values. (7 and 8 will probably go together).
- (9) Output all of the values into an external text file
- (10) Read the values into Microsoft Excel and plot a graph.
- (11) Your graph should indicate the original data points (read that file in as well).
- (12) Your program should make use of functions and pointers.
- (13) Your program should trap errors.

You may wish to extend your own specification beyond the basic requirements. Although this course is too brief to cover such things as structures and classes and windows application building, you may wish to include these in your program. However, I really will be happy with your graphs done in Excel.

Contents of the Assignment Report

Your report ought to include the following:-

- (1) A listing of your source code.
- (2) A full explanation of the workings of your program, indicating your understanding of its operation.
- (3) Try to indicate other uses for your program. Explain how the program might be improved, even just on an explanatory level.
- (4) Try to structure your report to have an Overview (which ought to include what the report is about and what it has achieved (~150 words), Introduction (what you are trying to achieve), Program listing and explanation and a Summary (include the things indicated in item (3), above). You may want to have a bibliography, if you have referred to any books in order to complete the work.

Assessment

To assess your work the following aspects will be considered:-

- Engineering Design
 - Your source code listing should be logical and structured
 - Testing; you may want to test your program with other data.
 - Functionality; does it perform the required tasks?
- Implementation
 - Method of programming, your approach to solving this assignment.
 - Inventiveness and originality.
- Programming aspects
 - Your understanding of the C++ language.
 - Choice and use of control structures; decisions, loops etc.
 - Use of comments and program layout.
- Marking scheme
 - (1) Solutions that give the correct graphs between the data points, without error, will gain a **pass** grade.
 - (2) A report that gives the correct graphs between the data points and an adequate write-up will gain a **merit** grade.
 - (3) A **distinction** will be given for a report with a good program, a good write-up and some indication of thoughts, ideas and/or implementations for an improved program design.

Completion Date

Please complete and submit the report for this assignment by Monday 14th May 2001. Your examinations are between Monday 14th and Thursday 31st. You are strongly advised to do little bits towards this assignment in your free time. **If you leave it until the last minute then you will not have time to do it.** If you have any problems then please contact me, or one of the tutors.

File Input and Output in C++

Introduction

For your assignment you have been asked to read and write to external text files. One of the main uses of C++ programs is to read in data, manipulate it in some way and then to create output data. Let us look at some example source code.

File Input and Output Programs

This is a simple program to write some entries to a text file:-

```
//file1.cpp
#include<iostream.h>
/*header needed for input and out put files*/
#include<fstream.h>

void main()
{
    /*open an output file in the same directory
    as your *.cpp source file*/
    ofstream out("Inventory.txt");

    /*put stuff in the file*/
    out<<"Radios "<<39.95<<endl;
    out<<"Toasters "<<19.95<<endl;
    out<<"Mixers "<<24.80<<endl;
    out<<"Tellys "<<206.50<<endl;
    out<<"Fridges "<<125.99<<endl;

    /*close the file*/
    out.close();
    getch();
}
```

The line `ofstream out("Inventory.txt");` opens a file for output. The file is in the same directory (folder) as your source code, `file1.cpp`. You could also use `ofstream out("c:\work\Inventory.txt")` to send the file to whatever directory you choose. The lines `out<<"Radios "<<39.95<<endl` put text entries into the file `Inventory.txt`. You can view this file using Notepad or Word. Not surprisingly, it looks like this

```
Radios 39.95
Toasters 19.95
Mixers 24.8
Tellys 206.5
Fridges 125.99
```

Ok, so that is how you write to an external file. What about reading from a text file. The following is a simple C++ program to read the file that you created with the program above:-

```
//file2.cpp
#include<iostream.h>
#include<fstream.h>

void main()
{
    /*open input file
    ifstream in("Inventory.txt");
```

```

//error handling routine
if(!in)
{
    cout<<"Cannot open file"<<endl;
    return 1;
}

char item[20];
float cost;

in>>item>>cost;
cout<<item<<" "<<cost<<endl;
in>>item>>cost;
cout<<item<<" "<<cost<<endl;
in>>item>>cost;
cout<<item<<" "<<cost<<endl;
in>>item>>cost;
cout<<item<<" "<<cost<<endl;
in>>item>>cost;
cout<<item<<" "<<cost<<endl;
in>>item>>cost;
cout<<item<<" "<<cost<<endl;
cout<<endl;

in.close();
getch();
}

```

This piece of code reads in the text file, Inventory.txt, which was created using file.cpp above. The lines `in>>item>>cost;` reads in the two text elements, item (i.e. radio etc) and cost (i.e. 39.95 etc). The lines `cout<<item<<" "<<cost<<endl;` just prints the item and the cost to the screen. There is an error handling routine to check if the file exists. More about error handling later in the course. For now, just try to ensure that when you read in files that you put them in the right directory in the first place. The problem with file2.cpp is that these lines, `in>>item>>cost` and `cout<<item<<" "<<cost<<endl` are repeated. So how can you write a piece of code that reads in a text file and stops when it reaches the end? Look at the following code

```

//file3.cpp
#include<iostream>
#include<conio.h>
#include<fstream.h>

void main()
{
    ifstream in("number2.txt");

    float x,y;
    int i=0;

    while(!in.eof())
    {
        i++;
        in>>x>>y;
        cout<<i<<"\t"<<x<<"\t"<<y<<endl;
    }

    in.close();
    getch();
}

```

4.0	2.6
4.1	2.0
5.5	4.0
8.0	4.0
9.0	4.1
10.0	6.5
12.0	7.2
15.0	7.0
16.5	7.6
17.0	8.8
19.5	8.9
21.0	10.4
24.0	10.6
25.0	12.2

This code reads in two columns of numbers and outputs them to the screen. The loop `while(!in.eof())` contains the keyword `eof`, meaning “end of file”. So, while not end of file, read stuff in. The file `number2.txt`, that I used is just on the left here. In theory, `file3.cpp` should be able to read in any text file containing two columns of floats. Note that `file3.cpp` does not contain the error handling routine from `file2.cpp`. It would be a good idea to put it in.

The following is another simple file input output program. It reads in a text file and converts all of the white space characters (space bars) to the ‘|’ symbol in a new file. Call the project “convert”. Note that you can copy a listing of this program from www.osborne.com. Go to this URL and then click downloads and find “Teach Yourself C++” (by Herbert Schildt). Save the file to disk, unzip it and you will find that this is listing 12 in the file `Chap9.lst`. You can access this file using Word, say. Just copy and paste it using the short cut keys.

```
// Convert spaces to |s.
#include <iostream.h>
#include <fstream.h>
using namespace std;

int main(int argc, char *argv[])
{
    if(argc!=3) {
        cout << "Usage: CONVERT <input> <output>\n";
        return 1;
    }

    ifstream fin(argv[1]); // open input file
    ofstream fout(argv[2]); // create output file

    if(!fout) {
        cout << "Cannot open output file.\n";
        return 1;
    }
    if(!fin) {
        cout << "Cannot open input file.\n";
        return 1;
    }

    char ch;

    fin.unsetf(ios::skipws); // do not skip spaces
    while(!fin.eof()) {
        fin >> ch;
        if(ch==' ') ch = '|';
        if(!fin.eof()) fout << ch;
    }

    fin.close();
    fout.close();

    return 0;
}
```

Try using something like the following, to test the program on

```
Mary had a little lamb  
its fleece was white as snow  
and every where that Mary went  
the lamb was sure to go
```

Write the file in notepad and save it as `mary.txt`, in the same folder as the executable `convert.exe`. To run this program, open up a dos box (dos prompt), change to the directory which contains the executable `convert.exe`. For example, type `cd \work\convert`. Then type `convert mary.txt mary1.txt`. This will convert the file `mary.txt` to `mary1.txt`, which will look like the following

```
Mary|had|a|little|lamb  
its|fleece|was|white|as|snow  
and|every|where|that|Mary|went  
the|lamb|was|sure|to|go
```

Next week we will look at C++ “pointers”. These are necessary to break up code containing arrays into functions.

Tutorial

(1) Try adding the following lines to `Inventory.txt`

```
Heaters 201.70  
Hairdryers 23.20
```

and then run `file2.cpp` again.

- (2) See if you can convert `file3.cpp` to read in the six sets of data in the table on page 68, for the spline curves. Do put in the error handling routine from `file2.cpp`.
- (3) Try to write some code to work out the elements of the augmented matrix on page 68. Splice this in with the code from tutorial question (2). In other words, read the `x,y` data points from a file and then write code to create the augmented matrix array.
- (4) You may as well go for it! Try to run the Gaussian Elimination code with the matrix from tutorial question (3). Make sure that you have completed question (3) from the Week 6 tutorial, i.e. have added the back substitution routine to `gauss2.cpp`.