

Lab4: Working with File Streams

Objective

The objective of this lab is to learn to work with data files:

- Input (read/retrieve) data from a file using input file stream object *ifstream*
- Output (write/store) information into a file using output file stream object *ofstream*
- Use file operations: *open*, *close*, *<< (write)*, *>> (read)*, and *eof (end of file)*
- Process information saved using other applications such as *MatLab*
- Reinforce all the programming concepts learned in Labs 1-3.

Description

For this lab, you will revisit the three (3) problems from Lab 3:

1. **Taylor:** Generate the sine and cosine of a set of angles (0 to 90 degrees in increments of 5) using the Taylor series, as specified in Lab3. Save each angle and its corresponding sine and cosine values in a file arranged as a table with three columns.
2. **Projectile:** Compute the distance and height of a projectile as a function of time as specified in Lab 3. Compute this for a fixed height and velocity, where the angle alpha varies, and is input from a file.
3. **Stick Figure:** Draw a stick figure as specified in Lab 3 and another fun figure of your choice using the same principles. Save both figures in a single output file.

For each solution you will develop an algorithm, code the algorithm in C++, develop and test it in the Visual C++ environment.

Visual C++ Environment

A **project** in the VC++ environment is a program. A **workspace** is a folder in which all project-related information is stored. When you create a project you may create it in a new workspace or add it to an existing workspace. Refer to the handout "Working with VC++ programs" for more details. Do the following:

1. Create a workspace called Lab4.
2. Create a project for each program and add it to the Lab4 workspace. Create project names as follows:
 - a. Taylor4
 - b. Projectile4
 - c. StickFigure4
 - d. FunFigure4

Template for Program Header

Place the following code at the beginning of every source code file that you submit for this class.

```

/*****
* NAME: your name
* PERSON NUMBER: your person number
* PROGRAM: Lab name
* PURPOSE: 1-2 line summary of the purpose of the lab
* DATE: Date of last update
* PLATFORM: Microsoft Visual C++ 6.0 Pro
* Course & Section:
*****/

```

On-line submission of your code

All source code (.cpp) and data files created by you and generated by the programs that you implement will need to be submitted using the on-line command which is available on-line on the course website (www.cse.buffalo.edu/~terril/eas230). A summary of the procedure for submission is as follows:

For example, from your command-prompt, do the following:

- ssh unix.eng.buffalo.edu
 - enter your username and password
- Change directories to where your files are located as shown in the example below,
 - cd ~/eas230/Lab4/Taylor4
- Run the following command for each file
 - submit_eas230 <file>
 - where <file> is the name of the file you need to submit.

Problem 1: Taylor Series and File Output

20%

You will work on *incremental development* of the Taylor series discussed in Lab3. We used the approximation given by Taylor Series for sine and cosine, for accuracy to 6 decimal places, we stop iterating when the term is insignificant as quantified by the following expression:

$$\left| \frac{x^n}{n!} \right| < 0.0000005$$

Your computation will “iteratively” add terms after making sure it is significant using the expression given above. For iterating we suggest using the “while” control structure.

The Taylor series approximation for sine and cosine are given below:

$$\sin x = x - (x^3/3!) + (x^5/5!) - (x^7/7!) + (x^9/9!) \dots$$

$$\cos x = 1 - (x^2/2!) + (x^4/4!) - (x^6/6!) + (x^8/8!) \dots$$

In this lab (Lab4) you will compute in a “loop” the sine and cosine for a range of x values ($0 \leq x \leq 90$) in increments of 5. Your program should output two sets (tables) of information. First set {x, sin(x)} pairs and the second set {x, cos(x)} pairs. Write the output to the monitor and save it to the file “**SinCos4.out**”.

Algorithm

Write an algorithm and include it as a “preamble” comment in your program.

Code

Create a new project called **Taylor4** and add it to the workspace **Lab4**. Create a C++ source file named **Taylor4.cpp**.

Submission

Submit the following files: **Taylor4.cpp** and **SinCos4.out**.

Problem 2: File Input and Output and Plotting Using Matlab 50%

We will “extend” the projectile problem solved in Lab 3 to repeat the computation of height and distance for various values of the parameter: angle of launch (alpha in the description below).

- You are to make values of “alpha” available in an “input” file named “**Alpha4.dat**”. Use alpha values between 10 and 60 degrees. You may use any text editor (xemacs, emacs, vi, or notepad) to create the file with alpha values and save it in the Lab4/Projectile4 folder.
- Assume that the program does not know the number of data points in the input file. You will have to use the function eof() (end of file) to determine the end of data in the input file. (A while loop with eof() is a good choice for a control structure.)
- Assume that z0 is 10.5 and v0 is 6.5. These can be assigned inside the program (as opposed to being read from the input file).
- For each value of alpha, the program will compute a set of r(t) and z(t) and write the set to a data file called “**Plot4.out**”. This is different than Lab3, where the program only generated one set of r(t) and z(t). The format output file you generate will be such that it can be fed into Matlab without any preprocessing. It will have four columns (no column headings needed), angle (in degrees), time, distance and height. A sample data set is shown below:

```
.....
15  0.3  1.88356 10.5632
15  0.31  1.94634 10.5501
15  0.32  2.00913 10.5361
15  0.33  2.07191 10.521
.....
```

```
.....
30  0.52  2.92717 10.8637
30  0.53  2.98346 10.8447
30  0.54  3.03975 10.8247
30  0.55  3.09604 10.8037
30  0.56  3.15233 10.7818
.....
```

- Use MatLab to plot the data. The Matlab program **Lab4.m** will be given to you. You will copy this program into your Projectile4 folder. Open the Lab4.m file using Matlab and run it. The details of using MatLab will be given to you in another document and during recitations. You will start Matlab from the Unix command line.
- You will be plotting two graphs: (i) a 2D one of distance vs height and (iii) a 3D one of distance, time and height. Save the plots as “**Matlab41.jpg**”, “**MatLab42.jpg**” and submit them online along with other files.

The projectile problem from Lab3 is repeated below for your reference:

This problem deals with equations dealing with motion of objects in two dimensions. In particular we will look at projectile motion as discussed in the web resource:

<http://www.glenbrook.k12.il.us/gbssci/phys/Class/vectors/u3l2b.html>. We are interested in the height and distance of the projectile with reference to time. The formulae for these two are readily available at another web resource: <http://www.egwald.com/geometry/projectile3d.php> and are repeated below. Assume that the projectile is fired from a gun that is mounted on a cliff above the sea.

z_0 = height of the gun above the x-y plane in meters.

v_0 = muzzle velocity = initial velocity of the projectile in meters/sec.

alpha = the angle between the horizontal (the x-y plane) and the muzzle of the gun in radians.

$z(t)$ = the height of the projectile t seconds after being fired (meters).

$r(t)$ = the distance of the projectile from the gun after t seconds (meters).

g = pull of gravity (9.81 meters/sec).

Then the parametric equations of motion of (r(t), z(t)) are:

$$\begin{aligned} r(t) &= v_0 * \cos(\alpha) * t \\ z(t) &= -1/2 * g * t^2 + v_0 * \sin(\alpha) * t + z_0 \end{aligned}$$

You will input z_0 , v_0 , and alpha and compute distance and height of the at various time intervals until the projectile reaches ground. Your program should repeatedly compute the distance and height and output them in a tabular form. Use a while loop control structure. Choose your intervals appropriately after executing the correct version of

