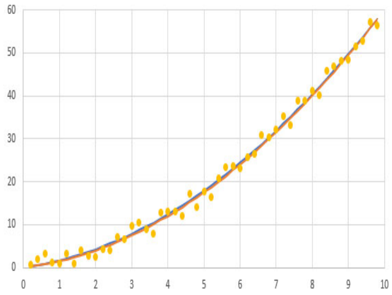

24-780 B—ENGINEERING COMPUTATION

Assigned: Wed. Sept. 17, 2021
Due: Tues. Sept. 21, 2021, 11:59pm

Problem Set 3: Visualizing Curves

In PS02, you created a program that could read a set of coordinate pairs from a file and determine the parameters of a best-fit quadratic of the form:

$$F = a_1 x + a_2 x^2$$

In this assignment, you will build on that success by generating a graphical view of the data and the resulting best-fit curve.

Task 1: Develop data structures

Create a struct type called *SpringCurve* to hold the information for a data set. You should include the coordinate pairs themselves, the parameters a1 and a2, the maximum x and maximum y values, and a color, enumerated as RED (default), ORANGE, GREEN, CYAN, BLUE.

Provide the following function that takes a string and a reference to a SpringCurve and performs the necessary operations to load the data from the file and determine other parameters. Note that you will make use of your old code from PS02 (or use my solution) with some additions/edits:

```
void createCurve(const string &inputFile, SpringCurve &aCurve);
```

Provide a function that prints out the data stored in a SpringCurve.

```
void printCurve(const SpringCurve &aCurve, ostream &output = std::cout);
```

Note that the declaration above includes a default value for the second parameter which means that if the second parameter is omitted in a function call, the default value will be used (i.e., prints to console).

The printing should list all data like this:

```
Max X = 25.78
Max Y = 64.7
Curve param a1 = 1.548
Curve param a2 = 0.7589
Color = RED
0.2   1.25
0.4   2.47
. . .
etc
```

Finally, provide a means of storing an indeterminate number of SpringCurve datasets (a collection).

Task 2: Provide Graphical Display

Use FSsimplewindow libraries to provide a graphical display of a single curve in OpenGL. The window size should be 800x600. Since we tend to think of the origin (0,0) as the lower left corner and then have y-coord go up from there, be sure to allow for this in your graphical output. Based on maximum X and maximum Y, choose an appropriate scale for your display (e.g., one unit of data = 10 pixels) so that everything fits and makes good use of the graphic space. Your display should include both the data points (any shape you like: square, diamond, cross, etc.) and the best-fit curve in the correct color scheme.

Prepare and make use of a function like this:

```
      void drawCurve(SpringCurve &aCurve);
```

Task 3: Add User Interaction

Allow the user to control the display by using the following key strokes on the OpenGL window (provide a “menu” on console window):

```
A -> Add a curve and display it (this will ask the user for a filename)
C -> cycle through possible colors
D -> delete the curve currently displayed from the collection
L -> list all the data for the currently displayed curve on the console
      window
Arrow Left/Right -> cycle through curves in the collection
X -> eXit the program
```

Deliverables

1 file, very appropriately named (or perhaps 3, based on how you feel about header files):

ps03springcurve_yourAndrewID.cpp

or

ps03springcurve_yourAndrewID.h

ps03springcurve_yourAndrewID.cpp

ps03springcurve_main_yourAndrewID.cpp

Upload the file(s) to the class Canvas page before the deadline (Tuesday, Sept.21, 11:59pm).

Hint: Even if you name your file appropriately, be sure to include your full name within the code itself (perhaps as a comment block at the top of the file). It is also appropriate to include date and course info, plus a short description of what the program does. (Think about what part of the text in the assignment write-up can be copy/pasted or adapted for this purpose.)

Learning Objectives

Applying basic data structures (enum, struct, vector)

Introduction to OpenGL

Making use of console input/output

File input and output

Using functions effectively

Searching references (online and/or textbook) for C++ library functions.